Interner Bericht
DESY FH1T-92-03
July 1992

# Simulation neuronaler Netzwerke und Test des Z-Kammer-Triggers am H1-Detektor

von

L. Larsson

# Simulation neuronaler Netzwerke und Test des Z-Kammer-Triggers am H1-Detektor

Experimentelle Diplomarbeit von
LARS LARSSON

II. Institut für Experimentalphysik
Universität Hamburg
Dezember 1991

**Abstract**

The scope of this report was to investigate by computer simulation how good one can separate physics events from background events with an artificial neural network in the z-chamber trigger of H1. The neural network used for simulations was a backpropagation (BPG) network.

The signals from the z-trigger are histograms being characteristic for the two classes of detected events. In a conventional way of solution one has to define analytical rules describing the characteristics of the event classes, and try to develop hardware which resembles these rules. This may require a complex hardware to do the separation in real time - order of hundreds of nano seconds for the z-trigger. In contrast to a conventional solution, a neural network can solve a pattern recognition problem by training the network with examples of data, and the results are direct portable to an equivalent hardware neural network, which is fast enough for real time signal processing.

The results gained with the neural network are compatible or better compared with those of an elaborated computer algorithm. A typical result of the simulation is that $\approx 90\%$ of the physics events were classified correctly allowing a background rate of $\approx 100\ Hz$.

# Inhaltsverzeichnis

# Abbildungsverzeichnis

# Tabellenverzeichnis

**Heute geht die äußere Welt, das heißt die Welt der Physik, über diejenige der Psychologie hinaus.**

*Salvador Dalí*

# Kapitel 1

# Einleitung

*In der Hochenergiephysik stellt sich in extremer Weise das Problem, daß der überwiegende Teil gemessener Signale Untergrundereignissen zuzuordnen ist und der physikalischen Zielsetzung nicht dienlich ist. Daher sind effiziente Filtersysteme - Trigger - notwendig, um die wenigen interessanten Ereignisse zu separieren.*

*Künstliche neuronale Netze - simulierte Verbunde von Nervenzellen - haben in diesem Zusammenhang für die Hochenergiephysik an Bedeutung gewonnen, insbesondere da in den letzten Jahren hochintegrierte Schaltungen verfügbar wurden, welche die direkte Anwendung für die zeitkritische Signalverarbeitung erlauben. Diese Möglichkeit wurde durch Computersimulationen im Rahmen der vorliegenden Arbeit untersucht.*

**HERA:** In diesem Kapitel werden die physikalische Zielsetzung und Grundlagen des Experiments beschrieben. Beginnend mit der Definition der kinematischen Variablen erfolgt mit Hilfe des Parton-Modells die Beschreibung der elementaren Reaktionen der *tiefinelastischen Lepton-Hadron-Streuung* im Rahmen des Standard-Modells. Abschließend wird kurz auf die experimentelle Situation eingegangen, und den sich daraus ergebenden meßtechnischen Konsequenzen, die für die Thematik der vorliegenden Arbeit von Bedeutung sind.

**H1-Detektor:** In diesem Kapitel wird der Aufbau des H1-Detektors dargestellt, der neben dem ZEUS-Detektor für die Untersuchung der Streuexperimente zur Verfügung stehen wird. Daran anschließend werden die für die Arbeit wichtigen *zentralen Driftkammern* beschrieben.

**Trigger:** Zunächst erfolgt eine kurze Beschreibung des H1-Gesamt-Trigger-Systems, sowie die Einführung einiger wichtiger Bezeichnungen. Die anschließende Darstellung der Funktionsweise und Funktionseinheiten des *Z-Kammer-Triggers* zeigt den Ansatzpunkt für die mit dem neuronalen Netzwerk durchgeführten Simulationen auf.

**Neuronale Netzwerke:** Die Darstellung der, für die vorliegende Arbeit wichtigen, theoretischen Grundlagen neuronaler Netzwerke erfolgt in diesem allgemein gehaltenen Kapitel. Es beginnt mit einer kurzen Beschreibung der für die mathematische Modellbildung relevanten Mechanismen des biologischen Vorbildes. Der Schwerpunkt des Kapitels liegt in der genauen Beschreibung des in der Literatur als *Backpropagation-Netzwerk* bezeichneten Modells, welches für die Simulationen verwendet wurde. Die neuronalen Netzwerke denen historisch in Bezug auf die Entwicklung der unterschiedlichen Modelle eine besondere Bedeutung zukommt werden kurz vorgestellt. Seit einiger Zeit gibt es integrierte Schaltkreise zur Implementation neuronaler Netzwerke (Neuro-Chips), die für zeitkritische Aufgaben geeignet sind. Die Anforderungen, die dabei zu stellen sind, werden beschrieben. Der Vergleich mit dem konventionellen Computer soll wesentliche Unterschiede zwischen diesen informationsverarbeitenden Systemen deutlich machen.

**Simulation:** Die Beschreibung des Simulationsablaufs, der verwendeten Daten, der vom konventionellen Teil des Z-Triggers zur Verfügung gestellten Signale und die Zuführung dieser Signale zur Verarbeitung durch das neuronale Netzwerk erfolgt in diesem Kapitel.

**Ergebnisse:** In diesem Kapitel werden die für eine zukünftige Implementation eines neuronalen Netzwerks in Kombination mit dem Z-Kammer-Trigger relevanten Simulationsergebnisse dargestellt, interpretiert und diskutiert.

**Z-Trigger-Testprogramm:** Die großen Datenmengen, die in der Hochenergiephysik zu verarbeiten sind, erfordern komplexe Elektronik und leistungsfähige Computer. Daher muß sich der Physiker verstärkt der Lösung technischer Probleme widmen, die nicht direkt eine physikalische Zielsetzung verfolgen, was bei der Entwicklung eines Testprogramms für die Elektronik des Z-Triggers deutlich wurde. Das Programm wurde zum Test der ersten Prototypen verwendet. Da diese zum Zeitpunkt der Programmentwicklung noch nicht verfügbar waren, war die strukturierte Programmierung für die spätere Fehlerkorrektur in Prototyp und Programm von entscheidender Bedeutung. Die durch die Programmentwicklung gesammelten Erfahrungen bezüglich der Z-Trigger-Elektronik waren für die simulierte Auswertung der Signale des Z-Triggers durch das neuronale Netz sehr wertvoll.

# Kapitel 2

# HERA

Mit HERA[1] wird beim DESY[2] in Hamburg ein Instrument verfügbar, das einen neuen kinematischen Bereich für Streuexperimente zur Erforschung der Wechselwirkung von Leptonen mit Hadronen und speziell der Protonenstruktur erschließt.

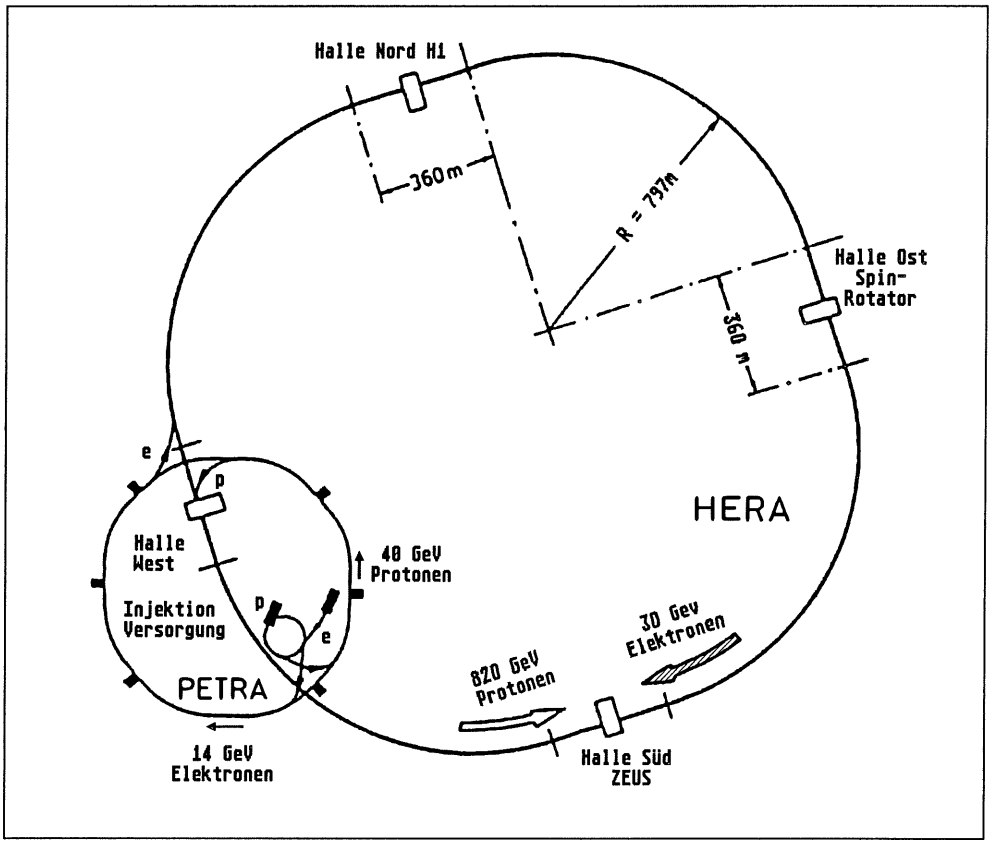


Abbildung 2.1: **Das Speicherringsystem HERA** *[Desy81].*

Für die Streuexperimente werden 30 $GeV$-Elektronen mit 820 $GeV$-Protonen zur Kollision gebracht, was einer Schwerpunktenergie $\sqrt{s} \approx 314$ $GeV$ entspricht. Für Messungen stehen zunächst zwei Detektoren - H1 und ZEUS - zur Verfügung.

[1] **H**adron **E**lektron **R**ing **A**nlage
[2] **D**eutsches **E**lektronen **SY**nchrotron

HERA weist grundsätzlich vier Wechselwirkungszonen auf, an zwei derer sich die Detektoren H1 und ZEUS befinden. Das Speicherringsystem HERA hat eine Länge von $6,3\ km$ und besteht aus zwei getrennten Ringsystemen für Elektronen und Protonen. In den Ringen laufen 220 Teilchenpakete (*engl. Bunches*) mit einer Kollisionsperiode von 96 $ns$ um.

## 2.1 Tiefinelastische Lepton-Hadron-Streuung

Das Elektron $e^-$ unterliegt der elektromagnetischen und schwachen Wechselwirkung. Das Proton, bestehend aus zwei $u$- und einem $d$-Quark, unterliegt zusätzlich der starken Wechselwirkung. Die Quantenelektodynamik (QED) beschreibt die elektromagnetische Wechselwirkung als Austausch von Photonen $\gamma$. Die schwache Wechselwirkung wird als Austausch von $Z^0$-Teilchen oder $W^+$, $W^-$-Teilchen beschrieben. Die Quantenchromodynamik (QCD) beschreibt die starke Wechselwirkung als Austausch von Gluonen $g$. Die Quarks unterliegen allen drei Wechselwirkungen. Die geschlossene Beschreibung dieser Wechselwirkungen erfolgt im Rahmen des Standard-Modells. Bei der $e^-p$-Streuung kommt sowohl Austausch von $\gamma$ und $Z^0$ (neutraler Strom) als auch von $W^+$ $W^-$ (geladener Strom) vor.

### 2.1.1 Kinematik

Die Gesamtkinematik eines Ereignisses wird, bei gegebener Schwerpunktenergie $\sqrt{s}$, durch zwei unabhängige Variablen, die Energie $E_l$ und den Winkel $\theta_l$ des gestreuten Leptons, charakterisiert. Die tiefinelastische Streuung des Elektrons $e$ mit dem Proton $p$ wird qualitativ durch die Reaktionsgleichung

$$e + p \longrightarrow l + H. \tag{2.1}$$

beschrieben. Dabei stellt $l$ das gestreute Leptonsystem und $H$ das gestreute Hadronsystem dar. Gemessen werden können alle wechselwirkenden Reaktionsprodukte. Mit dem Viererimpuls des Elektrons $p_e$, des gestreuten Leptons $p_l$ und des Protons $P$ ist das invariante Massenquadrat

$$s \equiv (p_e + P)^2 \quad \simeq 4E_eE_p. \tag{2.2}$$

Die Näherung $s \simeq 4E_eE_p$ ist, aufgrund der hohen bei HERA erreichten kinetischen Energien, verglichen mit der Ruhemasse $m_e \approx 0,5MeV$ des Elektrons und des Protons $m_p \approx 1GeV$ gerechtfertigt [Ing87a], [Phy0488]. Die ausgetauschten Bosonen weisen einen (raumartigen) Viererimpuls $q = p_e - p_l$ mit positivem Impulstransferquadrat

$$Q^2 \equiv -q^2 = -(p_e - p_l)^2 \quad \simeq 4E_eE_l \sin^2\frac{\theta_l}{2} \tag{2.3}$$

auf. Die Energie $\nu$ des Stroms der Austausch-Bosonen zwischen Proton und Elektron wird durch den Ausdruck

$$m_p \cdot \nu \equiv P \cdot q \quad \simeq 2E_p(E_e - E_l \cos^2\frac{\theta_l}{2}) \tag{2.4}$$

bestimmt. Konventionsgemäß werden die dimensionslosen Bjorken'schen Skalenvariablen $x$ und $y$, mit $0 \leq (x, y) \leq 1$, eingeführt

$$x \equiv \frac{Q^2}{2P \cdot q} = \frac{Q^2}{2m_p \nu} \simeq \frac{E_e E_l \sin^2 \frac{\theta_l}{2}}{E_p(E_e - E_l \cos^2 \frac{\theta_l}{2})} \tag{2.5}$$

und

$$y \equiv \frac{P \cdot q}{P \cdot p_e} = \frac{2P \cdot q}{s} = \frac{\nu}{\nu_{max}} \simeq \frac{E_e - E_l \cos^2 \frac{\theta_l}{2}}{E_e} \tag{2.6}$$

Die Variable $x$ beschreibt im Parton-Modell den Anteil des gestoßenen Partons am Gesamtimpuls des Protons, und $y$ den Energietransfer vom Elektron auf den Kern, was dem Anteil des Energieverlusts des Elektrons im Laborsystem entspricht. Für die Größen $Q^2$, $s$, $x$ und $y$ gilt dabei folgende Relation

$$Q^2 = sxy. \tag{2.7}$$

Die invariante Masse des Hadronensystems ist

$$W^2 \equiv (q + P)^2 = Q^2 \frac{1 - x}{x} + m_p^2 \quad \text{mit} \quad m_p^2 \leq W^2 \leq s. \tag{2.8}$$

Da die Gesamtkinematik durch zwei unabhängige Variablen bestimmt ist, können zwei beliebige der aufgeführten Größen als unabhängige Variable zur Beschreibung eines Wirkungsquerschnitts dienen. Üblich sind jedoch $(x, Q^2)$ oder $(x, y)$.

## 2.2 Parton-Modell

Die Definition der lorentzinvarianten Größen beschränkt sich auf die Beschreibung der Kinematik. Das Parton-Modell beschreibt die *tiefinelastische Streuung* des Elektrons und Protons als *elastische Streuung* des Elektrons mit *genau einem Parton* (s. Abb. 2.2). Das Proton wird dabei als aus mehreren *punktförmigen Partonen* bestehendes Teilchen angenommen [Per87].

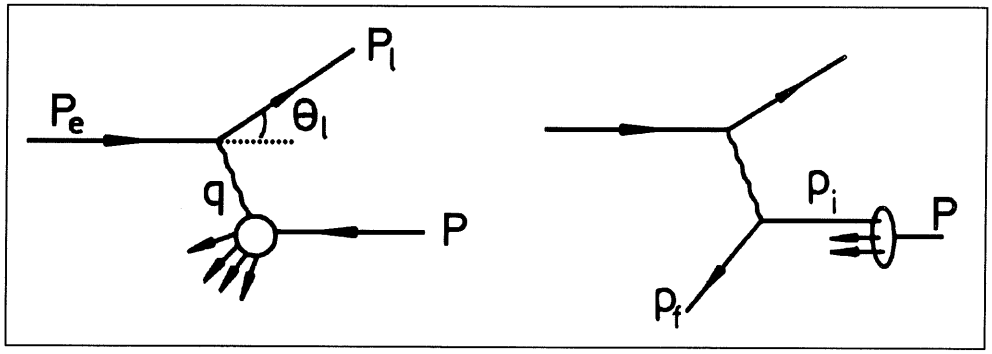


Abbildung 2.2: **Tiefinelastische Streuung** *im Parton-Modell (links) und im Quark-Parton-Modell (rechts) [Ing87a]. Das Parton wandelt sich in Hadronen um, die Emissionen von Teilchen aufgrund von Sekundärreaktionen in kleine Raumwinkel zur Folge haben (Jets, s. Abb 2.5)*

Im Quark-Parton-Modell wird die tiefinelastische Streuung des Elektrons und Protons als Streuung des Elektrons an einem Quark des Protons beschrieben (s. Abb. 2.2). Die Partonen sind nicht frei nachweisbar sondern wandeln sich in Hadronen und Mesonen um, die nach Sekundärreaktionen als sogenannte *Jets* (s. Abb. 2.5) nachgewiesen werden können.

## 2.3 Standard-Modell

Das Standard-Modell liefert für die tiefinelastische *ep*-Streuung Ausdrücke für einen differentiellen Wirkungsquerschnitt, der die Wechselwirkung durch Strukturfunktionen beschreibt, die ihrerseits die gesamten Wechselwirkungen beinhalten. Dabei wird, in Abhängigkeit der für die Kopplung verantwortlichen Austauschteilchen, zwischen *neutralem Strom,* $\gamma, Z^0$ (NC = neutral current[3]) und *geladenem Strom,* $W^+$, $W^-$ (CC = charged current[4]) unterschieden. Für kleine $Q^2$ überwiegt $\gamma$, bei $Q^2 \approx 10^4 GeV^2$ ist der Wirkungsquerschnitt für $\gamma$ und $Z^0$ gleich. Bei geladenen Strömen erfolgt die Kopplung durch $W^+$ und $W^-$. Die elektroschwache Wechselwirkung stellt eine Mischung der verschiedenen Kopplungen dar.

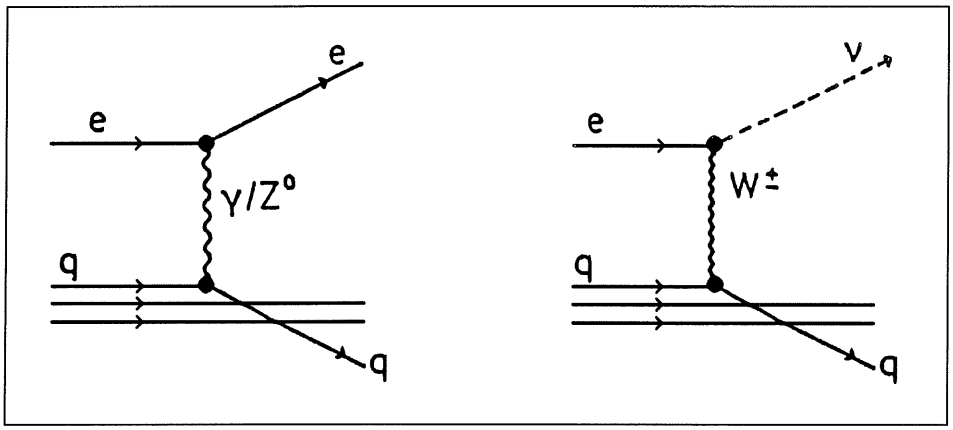


Abbildung 2.3: **NC- und CC-Ereignis** *[Ing87b]. Die Diagramme beschreiben die Lepton-Hadron-Streuung ohne QCD-Korrekturen höherer Ordnung, welche sich als Änderung der Wirkungsquerschnitte äußern. Bei einem NC-Ereignis erfolgt die Kopplung über* $\gamma/Z^0$.

Die Messung von Wirkungsquerschnitten erlaubt so die Überprüfung der Theorie, und gegebenenfalls die Modifikation. Abweichungen können Anhaltspunkte für neue Wechselwirkungen oder neue Teilchen sein [HERA87a], [HERA87b].

### Neutrale Ströme

Der differentielle Wirkungsquerschnitt wird als Funktion von $x$ und $Q^2$ für neutrale Ströme der elektroschwachen Wechselwirkung mit folgenden Ausdruck

$$\frac{d^2\sigma_{NC}(e^{\mp})}{dxdQ^2} = \frac{4\pi\alpha^2}{xQ^4}\left[y^2 xF_1(x,Q^2) + (1-y)F_2(x,Q^2) \pm (y - \frac{y^2}{2})xF_3(x,Q^2)\right] \tag{2.9}$$

[3] *engl. neutral current* ≈ neutraler Strom
[4] *engl. charged current* ≈ geladener Strom

als Funktion der Strukturfunktionen $F_1$, $F_2$ und $F_3$ beschrieben. Sie beschreiben die Kopplung der Quarks durch die Gluonen, sowie die Verteilung der Impulsanteile der Quarks im Proton. Dabei ist $\alpha$ die Feinstrukturkonstante. Bei Vernachlässigung der Masse und des inneren Transversalimpulses der Quarks gilt die Callan-Gross-Relation $2xF_1 = F_2$ [Loh86].

### Geladene Ströme

Für geladene Ströme bei der Elektron-Proton-Streuung wird der differentielle Wirkungsquerschnitt durch den Ausdruck

$$\frac{d^2\sigma_{CC}(e^-p)}{dx\ dQ^2} = \frac{(1-\lambda)\pi\alpha^2}{4\sin^4\theta_W(Q^2+M_W^2)^2}\sum_{i,j}\left[|V_{u_id_j}|^2u_i(x,Q^2)+(1-y)^2|V_{u_jd_i}|^2\bar{d}_i(x,Q^2)\right] \quad (2.10)$$

beschrieben. Die Variable $\lambda$ ($\pm 1$) gibt die Polarisation des Elektronenstrahls an, $u$ und $d$ bezeichnen die verschiedenen Quark-Sorten (flavours[5]). Die $V_{ij}$ sind die Elemente der Kobayashi-Maskawa-Matrix, welche im Standard-Modell die Quark-Kopplung über die vierdimensionale Stromdichte

$$J^\mu = G_F^{1/2}M_W 2^{-1/4}\cdot V_{ij}J_{ij}^\mu \quad (2.11)$$

der W-Bosonen beschreiben. Dabei ist $G_F$ die Fermi-Kopplungskonstante der schwachen Wechselwirkung, $M_W$ die W-Masse, und $\theta_W$ der Weinberg-Winkel, der die elektroschwache Wechselwirkung als Mischung von neutralem und geladenem Strom beschreibt [Per87].

### Strom-Gluon-Fusion

Bei der Elektron-Proton-Streuung kann es, insbesondere bei kleinem $x$, zur Reaktion zwischen einem Austauschteilchen mit einem Gluon des Protons kommen (s. Abb. 2.4). Diese Reaktion erlaubt die direkte Bestimmung der Gluon-Strukturfunktion [Barb87] und Gluon-Dichte [Coo87]. Da die Reaktion für die Produktion schwerer Quark-Antiquark-Paare verantwortlich ist, besitzt sie auch für die Erforschung der Physik schwerer Quarks große Bedeutung.

Die allgemeine Beschreibung der Strom-Gluon-Fusion erfolgt mit Hilfe von Parton-Strukturfunktionen $f_1, f_2, f_3$. Der Wirkungsquerschnitt für die Fusion eines (virtuellen) $\gamma$-Quant mit einem Gluon des Protons $\gamma g \rightarrow q\bar{q}$ wird für jedes Quark-Paar durch

$$\frac{d\sigma}{dQ^2dx} = \frac{4\pi\alpha^2}{xQ^4}[1-\frac{Q^2}{sx}+\frac{Q^4}{2s^2x}]\cdot\{\frac{\alpha_s(Q^2)}{\pi}e_q^2\int_{ax}^1 dx_gG(x_g,Q^2)f_2(\frac{x}{x_g},Q^2)\} \quad (2.12)$$

beschrieben. Dabei ist $G(x_g,Q^2) = x_gg(x_g,Q^2)$ die Gluon-Strukturfunktion, $g$ die Gluon-Dichte, $a$ ein von der Quark-Masse abhängiger Schwellenfaktor und

$$x_g \simeq x(1+\frac{M_{q\bar{q}}^2}{Q^2}), \quad (2.13)$$

wobei $M_{q\bar{q}}$ die invariante Masse des $q\bar{q}$–Paars ist.

[5] *engl. flavour* ≈ Geschmack

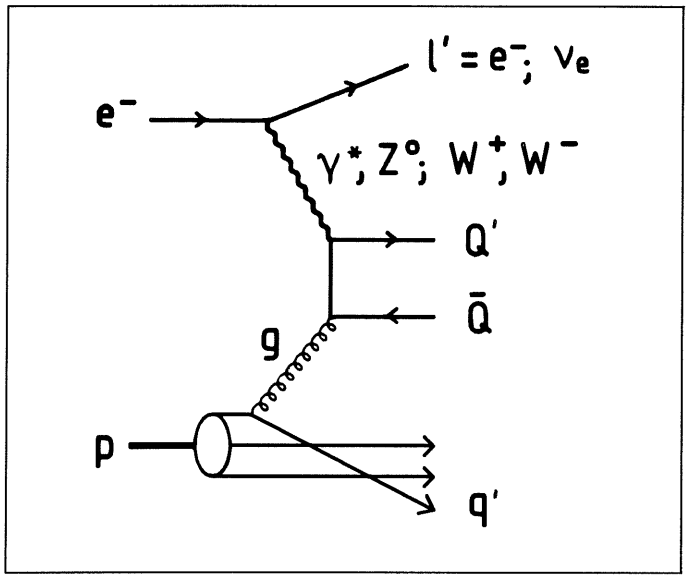


Abbildung 2.4: **Strom-Gluon-Fusion** *[Glü87].*

### 2.3.1 Skaleninvarianz

Bjorken stellte 1967 die Hypothese auf, daß die Strukturfunktionen der inelastischen Streuung bei großen $Q^2 \to \infty$ und $\nu \to \infty$ nicht mehr von zwei Variablen sondern nur noch von der Variablen $x$ abhängen. Da $x$ dimensionslos ist, gibt es dann bei großen $Q^2$ und $\nu$ keine Massenskala mehr, was als *Skaleninvarianz* bezeichnet wird. Die physikalische Deutung wurde 1969 durch Feynman im Parton-Modell gegeben [Per87], wobei die Skaleninvarianz dadurch erklärt wird, daß das Proton aus punktförmigen Partonen aufgebaut angenommen wird, und daß bei großen Impulsüberträgen $Q^2$ und Energieüberträgen $\nu$ aufgrund der kleinen Wellenlänge $\lambda \sim 1/\sqrt{Q^2}$ des virtuellen Photons und der kurzen Wechselwirkungsdauer $\tau \sim 1/\nu$ eine Streuung an einem einzelnen Parton erfolgt [Schm88].

Das in der Bjorken'sche Hypothese postulierte Skalenverhalten konnte experimentell bestätigt werden [Per87], durch Messung des Verlaufs der Strukturfunktionen als Funktion von $Q^2$ und $x$. Dabei wurde überwiegend $F_2$ gemessen, da diese den größten Beitrag am Wirkungsquerschnitt liefert. Die QCD, die der Wechselwirkung der Quarks untereinander Rechnung trägt, zeigt sich bei hohen $Q^2$ und bei sehr kleinen $x$ ($10^{-4}$) als Abweichung vom Skalenverhalten, was mit HERA erforscht werden soll [HERA87a], [HERA87b], um zu klären, ob das Standard-Modell auch dabei noch Gültigkeit besitzt.

## 2.4 Experimentelle Situation

Die inelastische Streuung erlaubt durch Messung von Wirkungsquerschnitten die Bestimmung von Parametern des Standard-Modells. Sie ist in der Lage, die Quark-Verteilung im Inneren des Proton aufzulösen, im Gegensatz zur elastischen Streuung, die nur eine gemittelte Verteilung liefert [Schm88]. Bei Vernachlässigung der Bindungsenergie der Quarks im Proton können diese als *quasi-freie* Teilchen erforscht werden. Mesonische und hadronische Zwischenzustände liefern weitere experimentelle Grundlagen der Quark-Physik.

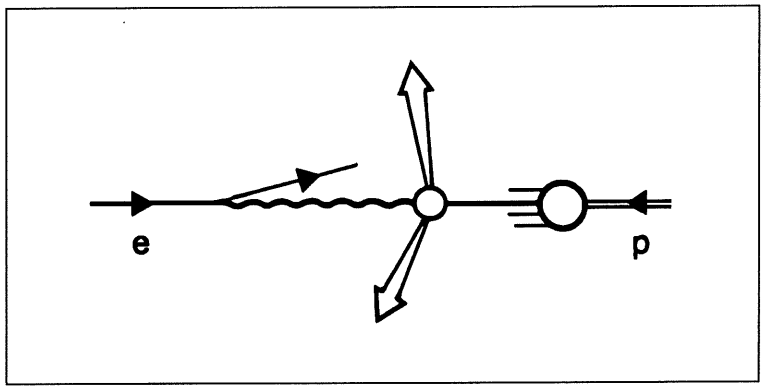


Abbildung 2.5: **Jet** *[Stir87]. Die Messung der Ereignis-Topologie und der im Detektor deponierten Energien erlaubt die Rekonstruktion der Kinematik. Meist werden Emissionen vieler Teilchen in kleine Raumwinkel beobachtet, die als Jets bezeichnet werden.*

Dazu ist die Rekonstruktion der kinematischen Variablen und die Identifikation der unterschiedlichen Reaktionen anhand der aus den verschiedenen Reaktionen hervorgehenden Fragmente erforderlich. So erfassen die Detektoren H1 und ZEUS, abgesehen vom Strahlrohr, fast den gesamten $4\pi-$Raumwinkelbereich. So können Reaktionsfragmente, die in das Strahlrohr emittiert werden nicht nachgewiesen werden.

Das gestreute Lepton wird im allgemeinen unter Winkeln $0^0$ bis $90^0$ - also im Zentral- bis Vorwärtsbereich (Richtung des Protonimpulses) - liegen. Der aus der Zerstörung des Protons bei der tiefinelastischen Streuung resultierende Jet (Zuschauer-Jet) geht zum größten Teil im Strahlrohr verloren. Die sehr energiereichen Strom-Jets (CC, bei großem $Q^2$) gehen sehr stark in den zentralen Bereich des Detektors. Neutral-Strom-Jets (NC) gehen vorwiegend in den Vorwärtsbereich [Ing87b].

Mit HERA kann bei kleinen $Q^2$ und kleinen $x$ viel Neues erforscht werden, insbesondere die $\gamma g$-Fusion, bei der die Energien der resultierenden Jets sehr klein ist, so daß man diese Ereignisse nicht anhand von Energieverteilungen sondern anhand ihrer Topologie mit Hilfe von Spurkammern (s. Kapitel 3) identifizieren muß [Barb87].

Beide Detektoren setzen sich aus für bestimmte Aufgaben spezialisierten Subdetektoren zusammen. Die Impulsdifferenz zwischen Proton und Elektron erfordert, im Gegensatz zu $e^+e^--$Experimenten, einen asymmetrischen Aufbau [TPH86].

### 2.4.1 Physikalische und technische Daten

Aus in Tabelle 2.1 aufgeführten Parametern ergeben sich meßtechnische Konsequenzen. So erfordert die Kollisionsperiode von 96 $ns$ Elektronik, die in der Lage ist die mit dieser korrelierten Ereignisraten schnell genug zu verarbeiten. Die große longitudinale Ausdehnung der Protonen-Pakete führt bei HERA zu einer ausgedehnten Vertex-Region ($\pm$ 30$cm$), was speziell für den Z-Kammer-Trigger einen hohen technischen Aufwand nach sich zieht.

| | | | |
|---|---|---|---|
| Elektronenenergie | : | 30 | $GeV$ |
| Protonenenergie | : | 820 | $GeV$ |
| Schwerpunktenergie | : | 314 | $GeV$ |
| Teilchenpakete | : | 220 | |
| Elektronen / Paket | : | $3,5 \cdot 10^{10}$ | |
| Protonen / Paket | : | $1,0 \cdot 10^{11}$ | |
| Elektronenpaketlänge | : | 8,3 | $mm$ |
| Protonenpaketlänge | : | 150,0 | $mm$ |
| Kollisionsperiode | : | 96 | $ns$ |
| Luminosität | : | $1,5 \cdot 10^{31}$ | $cm^{-2}s^{-1}$ |
| Photoproduktion (gesamt, sichtbar) | : | $10^3$, $10^2$ | $Hz$ |
| NC-Rate für $Q^2 \geq 3^2$ | : | 3 | $Hz$ |
| NC-Rate für $Q^2 \geq 5000^2$ | : | $10^{-4}$ | $Hz$ |
| CC-Rate über alle $Q^2$ | : | $3 \cdot 10^{-3}$ | $Hz$ |
| CC-Rate für $Q^2 \geq 5000^2$ | : | $5 \cdot 10^{-4}$ | $Hz$ |
| $\gamma$–Gluon ($ep \rightarrow ep + c\bar{c}$ oder $J/\Psi$) | : | $\approx 10^{-2}$ | $Hz$ |
| Protonenverlustrate | : | $3 \cdot 10^5$ | $s^{-1}m^{-1}$ |
| Elektronenverlustrate | : | $1 \cdot 10^3$ | $s^{-1}m^{-1}$ |
| Proton-Restgas ($10^{-9}$ $Torr$) | : | $9 \cdot 10^3$ | $s^{-1}m^{-1}$ |
| Synchroton-Strahlung ($\pm$ 2,5$m$) | : | $1 \cdot 10^8$ | $s^{-1}$ |
| Kosmische Strahlung | : | $3 \cdot 10^3$ | $s^{-1}$ |

Tabelle 2.1: **HERA-Daten** *[TPH86], [Phy0488].*

### Physik und Untergrund

Die geringen Wirkungsquerschnitte $\sigma$ der physikalisch relevanten Reaktionen führen auch bei der mit HERA erreichbaren Luminosität $L$ zu geringen Ereignis-Raten $N \approx \sigma \cdot L$ von bestenfalls einigen 100 $Hz$. Manche Ereignisse werden jedoch nur mit Bruchteilen von $Hz$ auftreten. Um aussagekräftige Ergebnisse zu erhalten, müssen viele physikalisch relevante Ereignisse für die Analyse zur Verfügung stehen. Dementgegen stehen Untergrund-Raten bis in den $100kHz$–Bereich, hervorgerufen durch Wechselwirkung von defokussierten Protonen und Elektronen mit dem Strahlrohr[6], sowie Wechselwirkung mit Restgas[7], durch Synchroton-Strahlung und kosmische Strahlung. Die beobachteten Untergrundraten hängen jedoch sehr stark von dem verwendeten Trigger ab. So ist ein Trigger, der auf Spuren in Driftkammern anspricht, relativ unempfindlich gegenüber durch Photonen hervorgerufenem Untergrund, da sich solcher in Form von vereinzelten Signalen zeigt (Hits), denen in der Regel keine Spuren zugeordnet werden können, es sei denn aufgrund zufälliger Koinzidenzen [TPH86].

Technisch ergibt sich das Problem, daß aufgrund der hohen Datenmenge, die bei einem Ereignis anfällt (*ca.* 100 $kByte$), nur etwa 5 Ereignisse pro Sekunde zur späteren *rechenintensiven* Auswertung gespeichert werden [TPH86].

So ergibt sich die Notwendigkeit leistungsfähiger Trigger-Systeme, die eine hohe Untergrundunterdrückung bei geringem Verlust physikalisch relevanter Ereignisse erreichen.

[6] *engl. beam wall* $\approx$ Strahl-Wand
[7] *engl. beam gas* $\approx$ Strahl-Gas

# Kapitel 3

# H1-Detektor

Die Anforderungen, die an den Detektor gestellt werden, ergeben sich aus den primären physikalischen Zielsetzungen. Dabei soll die Erforschung neuer Physik nicht durch eine zu starke Spezialisierung auf die erwartete Physik erschwert werden.

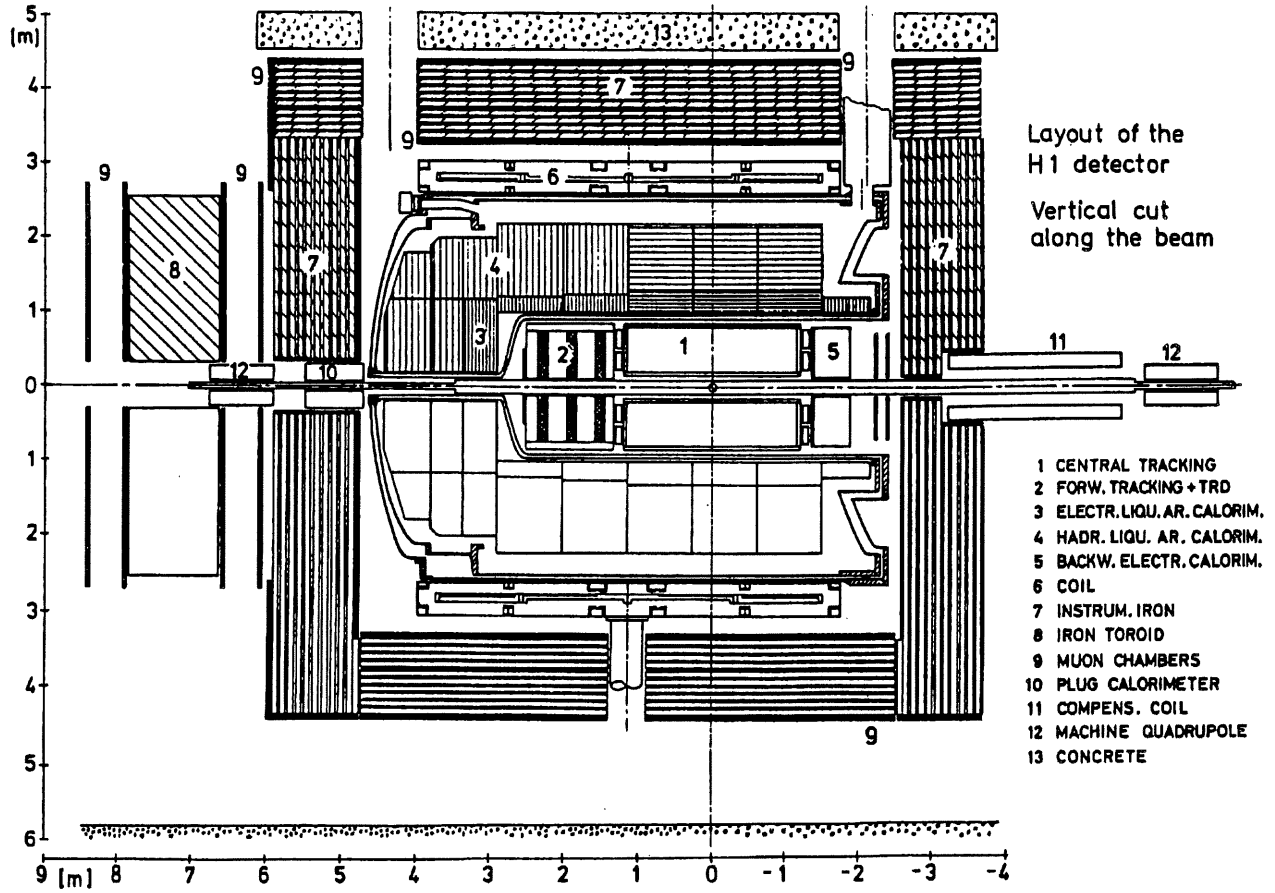


Abbildung 3.1: **H1 in RZ-Projektion** *[TPH86].*

Der Detektor muß in der Lage sein, möglichst das gesamte Ereignis zu erfassen. Nichtwechselwirkende Teilchen, wie Neutrinos, erlauben nur den indirekten Nachweis durch Messung von Energiefluß und Impulsen der auslaufenden Teilchen eines Ereignisses.

## 3.1 Aufbau

Die Rekonstruktion eines Ereignisses erfolgt durch Verknüpfung der von den verschiedenen Detektorkomponenten zur Verfügung gestellten Informationen. Im Inneren des Detektors, nahe am Strahlrohr befinden, sich verschiedene Driftkammern (s. Abb. 3.1) zur Spurmessung. Darauf folgen weiter außerhalb zunächst ein elektromagnetisches Kalorimeter gefolgt von einem hadronischen Kalorimeter zur raumwinkelauflösenden Energieflußmessung. Die Driftkammern und Kalorimeter sind von einer supraleitenden Spule umgeben, die ein weitgehend homogenes Magnetfeld ($1,2\ T$, $\Delta B_z < \pm 3\%$) parallel zur Strahlachse für Impulsmessung und Teilchenidentifikation anhand von Spurkrümmungen erzeugt. All diese Detektorkomponenten sind von einem Eisenjoch umgeben, das ergänzend zur Messung von hadronischem Energiefluß aus dem Haupkalorimeter, als auch zur Messung und Identifikation von Myonen mit Hilfe von Streamer-Kammern [TPH86] dient.

## 3.2 Driftkammern

Driftkammern erlauben die räumliche Erfassung von Spuren ionisierender Teilchen. Sie basieren auf dem direkten Nachweis von Elektronen und Ionen, die bei dem Durchgang ionisierender Strahlung durch eine gasgefüllte Kammer entstehen, in der zwecks Ladungstrennung und Nachweis ein elektrisches Feld herrscht.

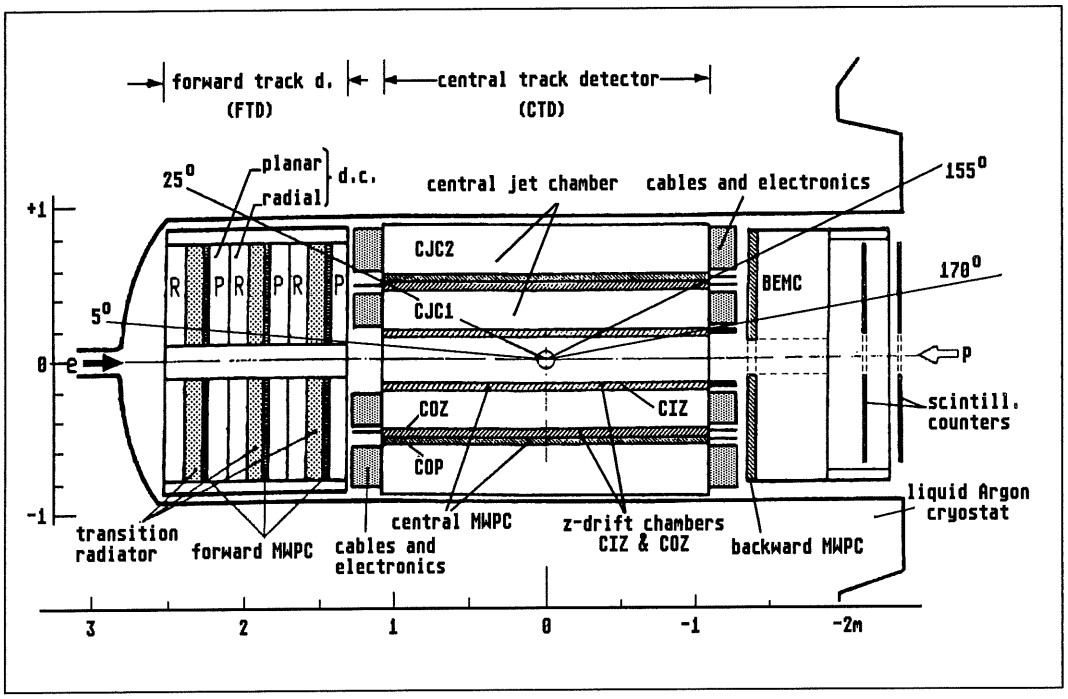


Abbildung 3.2: **H1-Spurkammern** *[TPH86]. Vergrößert dargestellt sind, neben anderen Detektorkomponenten, die inneren Driftkammern des H1-Detektors (s. Abb. 3.1). Der Z-Trigger wertet die Signale der als CIZ und COZ bezeichneten Driftkammern aus.*

### 3.2.1 Funktionsweise

Bei Abwesenheit eines elektrischen Feldes führt die thermische Bewegung nach erfolgter Ionisation schnell zur Rekombination der Ladungsträger. Das Vorhandensein eines elektrischen Feldes bewirkt jedoch die Trennung der Elektronen und Ionen. Die sehr viel schnelleren Elektronen werden am Signaldraht[1] (Anode) nachgewiesen [Leo87].

Bei geeignetem elektrischen Feld wird ein näherungsweise linearer Zusammenhang zwischen Ionisationsort und Driftzeit erreicht, so daß die Messung der Driftzeit dann in verhältnismäßig einfacher Weise eine Ortsmessung ermöglicht. Die Orte, die gleichen Driftzeiten zuzuordnen sind, werden als *Isochronen* bezeichnet (s. Abb. 3.3). Indem mehrere Meßdrähte in einer Driftkammer integriert werden, kann die Richtung von Spursegmenten innerhalb einer Driftzelle bestimmt werden.

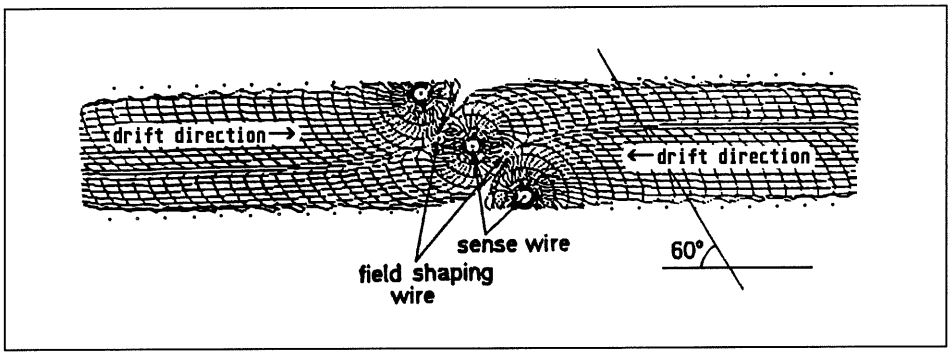


Abbildung 3.3: **Feldlinien und Isochronen** *[TPH86]. Das Driftkammersignal stellt eine gemischte Ort-Zeit-Information dar: Die ansprechende Driftkammer liefert entsprechend ihres sensitiven Bereichs eine grobe Ortsinformation, die gemessene Driftzeit liefert eine feine Ortsinformation innerhalb der Kammer.*

#### Gasmischung

Die Driftgeschwindigkeit ist, neben der elektrischen Feldstärke $E$ und dem Druck $p$ in der Kammer, von den verwendeten Gasen und deren Mischungsverhältnis abhängig [Leo87].

#### Gasverstärkung

In der Nähe des Meßdrahts kommt es aufgrund der höheren elektrischen Feldstärken zur Ionisation der den Draht umgebenden Gasmoleküle. Die dabei freiwerdenden Elektronen führen zu weiteren Ionisationen. Es kommt zu einer lawinenartigen Verstärkung der Primärelektronen. Diese Gasverstärkung ist abhängig von der verwendeten Gasmischung [Leo87].

#### Lorentzwinkel

Das Vorhandensein eines Magnetfelds führt aufgrund der Lorentzkraft $\vec{F} = q \cdot (\vec{v} \times \vec{B})$ zu einer Drehung des durch das elektrische Feld bestimmten Driftweges um den als *Lorentzwinkel* bezeichneten Winkel, sofern die Driftwege nicht parallel zu den Magnetfeldlinien verlaufen.

[1] *engl. sense wire* $\approx$ Meßdraht

### Links-Rechts-Ambiguität

Wenn eine Teilchenspur nicht eine das elektrische Feld begrenzende Kathodenebene schneidet, ist es nicht möglich das betreffende Spursegment eindeutig einer bestimmten Seite der Meßdrahtebene zuzuordnen (s. Abb. 3.3). Diese Uneindeutigkeit wird als *Links-Rechts-Ambiguität* bezeichnet.

## 3.3 Die Z-Vertex-Driftkammern CIZ und COZ

Der Z-Kammer-Trigger wertet die Signale der zentralen inneren Z-Kammer (CIZ=**c**entral **i**nner **z**-chamber) und der zentralen äußeren Z-Kammer (COZ=**c**entral **o**uter **z**-chamber) aus. Beide Driftkammern sind prinzipiell in ähnlicher Weise aufgebaut (s. Abb. 3.4). Wesentlich ist dabei, daß die Driftwege parallel zur Strahlachse verlaufen und daher die genaue Bestimmung der Z-Koordinaten von Spursegmenten ermöglichen, insbesondere weil sich beide Kammern, verglichen mit den anderen Driftkammern, nahe am Strahl befinden. Sowohl die äußere als auch die innere Z-Kammer weisen eine Auflösung der Z-Koordinate von $\approx 350\mu m$ auf [TPH86].

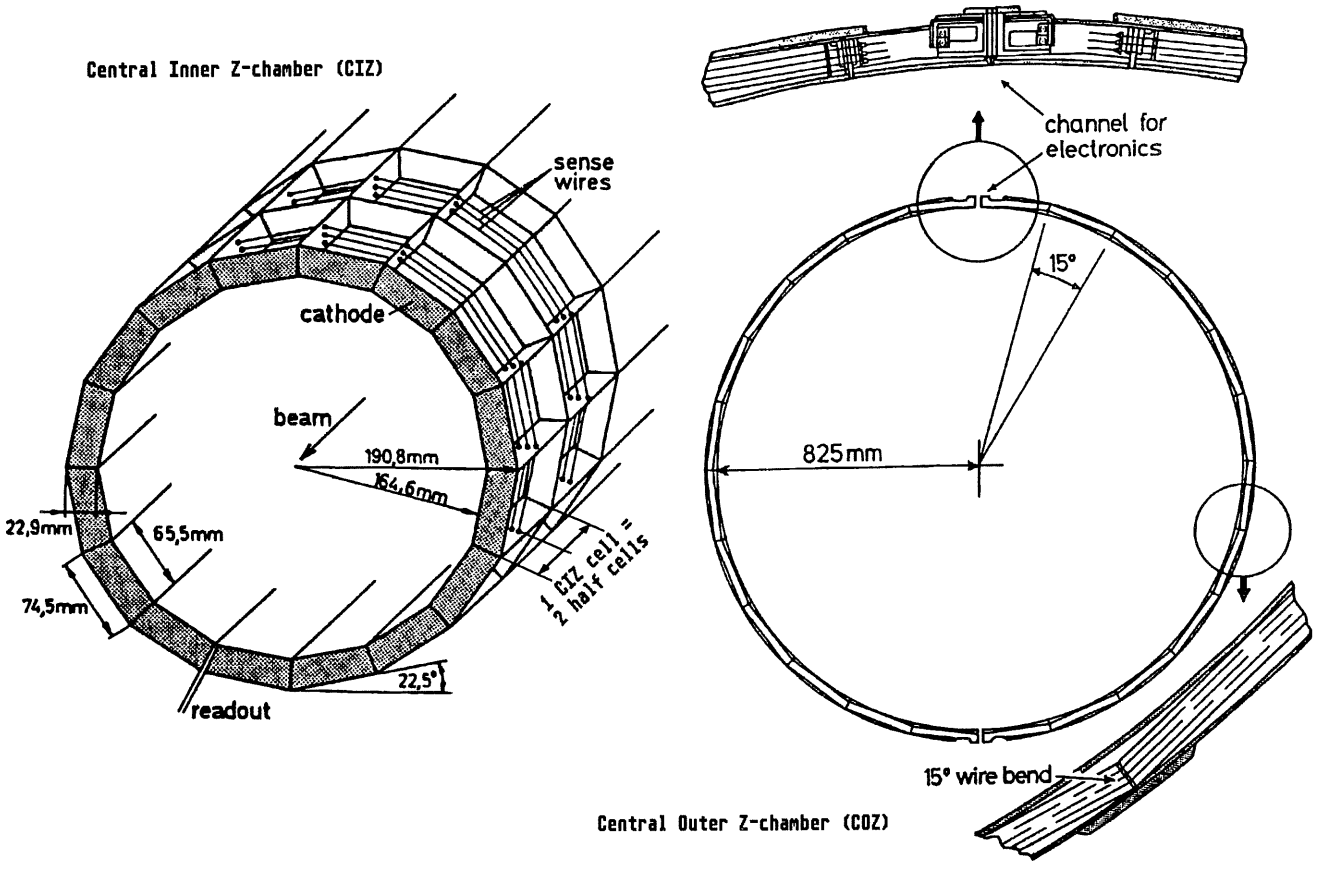


Abbildung 3.4: **Die Z-Kammern CIZ & COZ** *gleichen sich in ihrem Aufbau [TPH86]. Die Driftwege verlaufen näherungsweise auf konzentrischen Zylinderoberflächen parallel zur Strahlachse. Sowohl CIZ als auch COZ setzen sich aus separaten Driftzellen ($n_{CIZ}$ = 15, $n_{COZ}$ = 24) zusammen. Durch die Meßdrahtebene werden sie funktional in zwei Halbzellen unterteilt.*

# Kapitel 4

# Trigger

Die Aufgabe eines Triggers ist es, anhand von Detektorsignalen zu entscheiden, ob die vorliegenden Signale von einem Untergrundereignis herrühren, oder ob sie einem potentiell physikalisch interessanten Ereignis zuzuordnen sind, so daß die dauerhafte Speicherung zur späteren genauen Analyse auf dem Großrechner lohnt. An ein Trigger-System werden komplementäre Anforderungen gestellt, um die wenigen physikalisch relevanten Ereignisse aus der überwiegenden Menge der Untergrund-Ereignisse herauszufiltern. Die hohen Datenraten erfordern sehr schnelle Elektronik zur Verarbeitung der Signale.

## 4.1 Das H1-Gesamt-Trigger-System

Das Gesamt-Trigger-System des H1-Detektors baut sich aus einzelnen Sub-Triggern auf, die zunächst unabhängig von den anderen Trigger-Systemen nur die Signale bestimmter Detektorkomponenten auswerten. Das Gesamt-Trigger-System ist in mehrere Filterstufen eingeteilt, denen jeweils unterschiedlich viel Zeit für die Klassifizierung eines Ereignisses zur Verfügung steht (s. Tab. 4.1). Entsprechend unterscheiden sich die Komponenten der Trigger-Stufen in Aufbau und Komplexität. Durch Verknüpfung der einzelnen Sub-Trigger-Systeme wird letztlich das Signal zum Auslesen der gesamten Detektordaten eines Ereignisses gewonnen [TPH86].

### Trigger-Filterstufen

Die verschiedenen Trigger-Stufen[1] verarbeiten unterschiedlich hohe Eingangsraten. Jede Trigger-Stufe muß eine Ausgangsrate liefern, die nur so hoch ist, daß sie von der folgenden Stufe verarbeitet werden kann, dabei darf die Totzeit jeweils einige % nicht überschreiten.

| Level | Ausgangsrate | Verarbeitungszeit | Realisierung |
|---|---|---|---|
| L1 | $\approx 10^4\ Hz$ | $1\ \mu s$ | Festverdrahtete Logik |
| L2 | $\approx 10^3\ Hz$ | $10\ \mu s$ | Festverdrahtete Logik |
| L3 | $\approx 50\ Hz$ | $1\ ms$ | $\mu$Prozessor |
| L4 | $\approx 5\ Hz$ | $10\ ms$ | $\mu$Prozessoren |

Tabelle 4.1: **Trigger-Stufen** *[TPH86]. Die angegebenen Ausgangsraten und Verarbeitungszeiten sind grobe Richtwerte, für die nach Möglichkeit kleinere Werte angestrebt werden.*

[1] *engl. level* $\approx$ Ebene $\rightarrow$ L1, L2, L3 und L4

Damit die höheren Trigger-Stufen hinreichend viel Zeit zur Bildung der Trigger-Bedingung zur Verfügung haben, werden die Signale potentiell interessanter Ereignisse über einen gewissen Zeitraum - Totzeit - zwischengespeichert, bis die Bewertung abgeschlossen ist. Dagegen sind L1-Trigger-Systeme totzeitfrei, da diese in der Lage sind, laufend neu eintreffende Detektorsignale zu speichern, auch wenn die Bewertung zuvor eingetroffener Signale noch nicht abgeschlossen ist. Diese Technik wird als *Pipelining* bezeichnet.

### Sub-Trigger-Systeme

Die Sub-Trigger-Systeme lassen sich in *Spur-* und *Energie-Trigger* einteilen. Diese Einteilung ergibt sich aus der physikalisch bedingten Spezialisierung einzelner Sub-Detektoren. So können die Kalorimeter zwar den Raumwinkel von Jets auflösen ($< 10\ mrad$ für $\theta < 20^0$), sind jedoch nicht für die genaue Bestimmung des Wechselwirkungspunkts geeignet, da die Messung auf der Energieabsorption durch (elektromagnetische bzw. hadronische) Aufschauerung der Primärteilchen beruht. Zudem sind sie verglichen mit den inneren Driftkammern relativ weit vom Wechselwirkungspunkt entfernt. Die (inneren) Driftkammern sind sehr gut für die Messung der Topologie eines Ereignisses geladener Teilchen und des absoluten Impulses anhand des Krümmungsradius $\rho[m] = p[GeV/c]/0,3 \cdot B[T]$ [Per87] geeignet. Sie erlauben die genaue Bestimmung des Wechselwirkungspunkts ($\Delta z_{CIZ}, \Delta z_{COZ} \approx 350\mu m$).

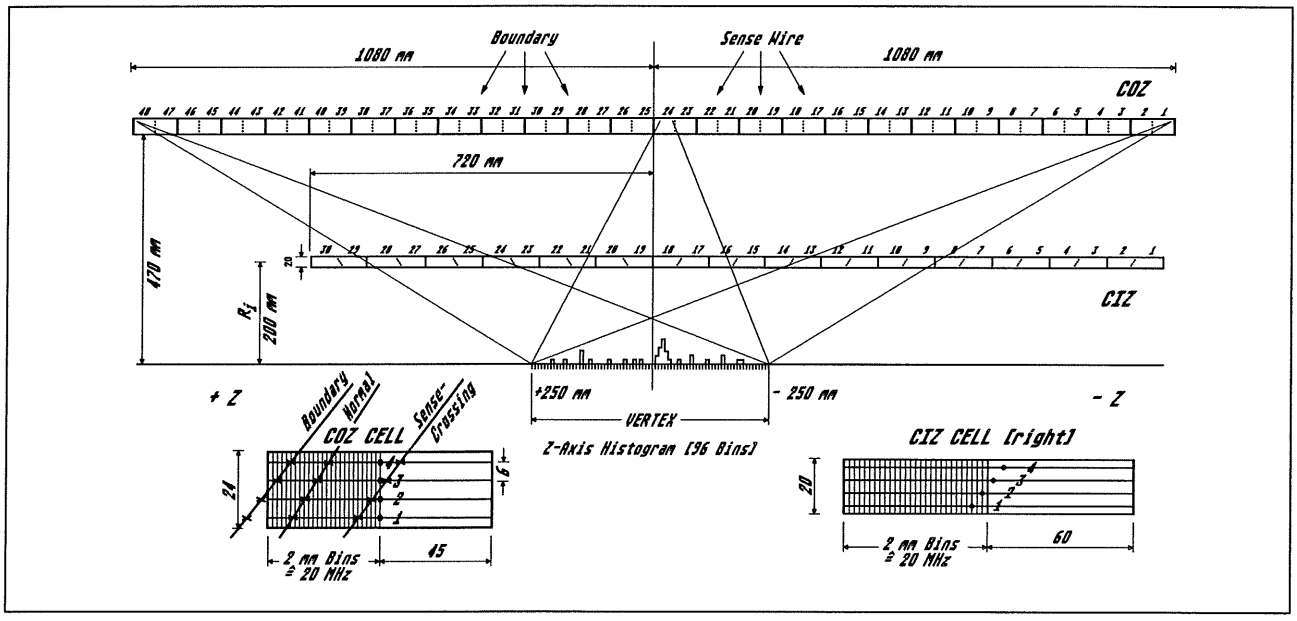


Abbildung 4.1: **H1-Z-Trigger und Driftkammern CIZ und COZ** *in RZ-Projektion [Zim90a]. Die Driftwege in den Kammern verlaufen näherungsweise parallel zur Strahlachse auf konzentrischen Zylinderoberflächen. CIZ und COZ sind in Z-Richtung in Teilkammern unterteilt, die als Driftzellen bezeichnet werden. In jeder Driftzelle befinden sich vier Meßdrähte, die polygonal - quasi konzentrisch - um die Strahlachse gespannt sind. Unten links ist eine COZ-Driftzelle mit drei charakteristischen Spursegmenten dargestellt, unten rechts ist eine CIZ-Driftzelle (rechtsseitig, für negative z, geneigte Meßdrahtebene) dargestellt.*

## 4.2 Der Z-Kammer-Trigger

Der Z-Kammer-Trigger, oder kurz Z-Trigger, wertet die Signale der zentralen inneren Z-Kammer CIZ und der zentralen äußeren Z-Kammer COZ aus (s. Abb. 3.4). Der Z-Trigger ist für den Betrieb als L1- oder L2-Trigger vorgesehen und der Klasse der Spur-Trigger zuzuordnen. Zunächst dachte man, daß es genügen würde, nur auf Energie zu triggern, unterstützt durch eine grobe Vertex-Bestimmung. Doch gibt es auch physikalisch interessante Ereignisse bei kleinen absorbierten Energien. So wurden zunächst Trigger mit scharfer Spurdefinition in der $R\Phi$-Ebene konstruiert. Untergrund-Simulationen zeigten jedoch, daß der Untergrund am besten in der $Rz$-Projektion zu erkennen ist. Daraus ergab sich die Notwendigkeit, den sehr genauen ($\Delta z \approx 5\ mm$) Z-Trigger zu bauen [Beh90].

#### Driftkammer-Signale

Aufgrund der vorliegenden Symmetrie bieten sich Zylinderkoordinaten $(r, \theta, \phi)$ zur parametrisierten Beschreibung einer Teilchenspur an. Bei HERA ist dabei die Richtung der Z-Achse in Richtung des Protonen-Impulses definiert. Die Kammern CIZ und COZ liefern so jeweils vom Winkel $\phi$ unabhängige Koordinaten $z_1, z_2, z_3$ und $z_4$ eines Spursegments bei Radien $r_1, r_2, r_3$ und $r_4$, die durch die (mittleren) Abstände der (polygonalen) Meßdrähte bestimmt sind.

#### Funktionsprinzip

Aus der Koinzidenz von gefundenen Spursegmenten in CIZ und COZ lassen sich auf der Strahlachse Vertices der den Spursegmenten zuzuordnenden Teilchenspuren finden. Die Anzahl der Spuren, die einem Punkt der Wechselwirkungszone zuzuordnen sind, ergibt in der Gesamtheit das *Z-Vertex-Histogramm* (s. Abb. 4.1), wie es von Teilen des Z-Triggers erzeugt wird. Es ist, bis zu einem gewissen Grade, für ein Ereignis charakteristisch und erlaubt so die Separation von Physik und Untergrund. Jedes in CIZ oder COZ gefundene Spursegment zeigt entweder auf einen Punkt der Wechselwirkungszone, oder auf einen Punkt außerhalb der Wechselwirkungszone. So werden die in den Driftzellen gefundenen Spursegmente gezählt, und je nach Richtung einem Bin in den sogenannten *Zell-Histogrammen* zugeordnet. Dabei wird in Abhängigkeit von der Richtung der Spursegmente zwischen den Vertex-Zell-Histogrammen und den Untergrund-Zell-Histogrammen unterschieden, jeweils separat für CIZ und COZ.

### 4.2.1 Funktionseinheiten

Der Z-Trigger teilt sich funktional und konstruktiv in Eingangslogik (*input card*), Kombinationslogik (*combination card*) und Histogrammlogik (*histogram card*) auf (s. Abb. 4.2). Die Eingangslogik dient zunächst zur Synchronisation der (digitalisierten) Kammersignale von CIZ und COZ. Die Hauptaufgabe der Eingangslogik besteht im Finden von Spursegmenten innerhalb der Driftzellen von CIZ und COZ sowie der Verknüpfung der Signale benachbarter Halbzellen und Driftzellen und der Bildung der Zell-Histogramme. Die Kombinationslogik erzeugt aus Koinzidenzen zwischen den in CIZ und COZ gefundenen Spursegmenten das Z-Vertex-Histogramm. Diese werden der Histogrammlogik zusammen mit den von der Eingangslogik bestimmten Zell-Histogrammen von CIZ und COZ zugeführt. Die Histogrammlogik bewertet die Signale und liefert gegebenenfalls das Z-Trigger-Signal, welches an den H1-Gesamt-Trigger weitergeleitet wird.

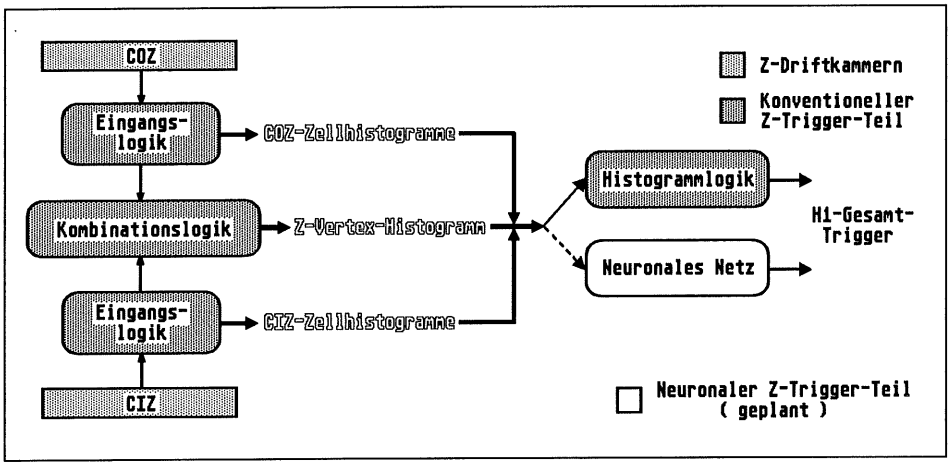


Abbildung 4.2: **Funktionseinheiten des Z-Triggers**

### 4.2.2 Trigger-Masken

Die Definition von Koinzidenzen erfolgt mit Hilfe von logischen UND-Gattern. Eine derart definierte Koinzidenz wird als *Trigger-Maske* bezeichnet. Die Definition eines von der Eingangslogik zu erkennenden Spursegments wird als *Primärmaske* bezeichnet. Die Definition einer Koinzidenz zwischen Spursegmenten von CIZ und COZ, die zu einträgen im Z-Vertex-Histogramm führen, wird als *Sekundärmaske* bezeichnet.

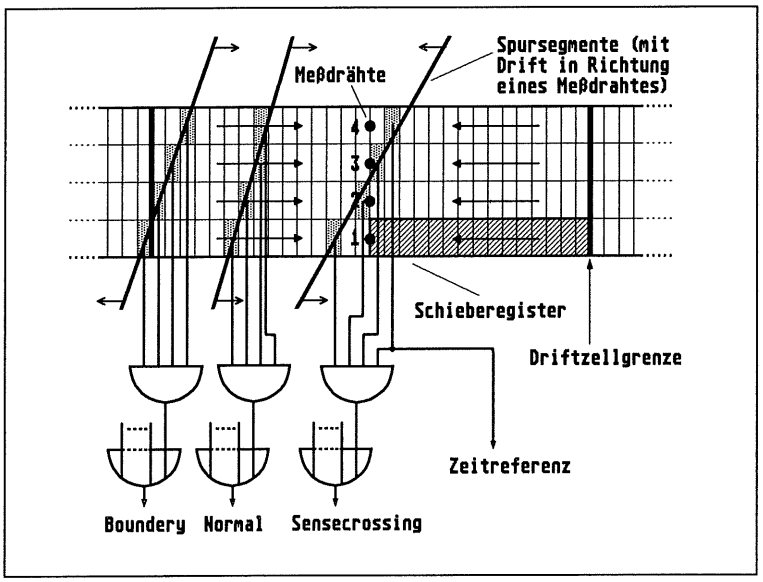


Abbildung 4.3: **Primärmasken** *der Eingangslogik.*

### Primärmasken

Die Driftkammersignale werden den *Primärschieberegistern* zugeführt, die synchron mit der halben Kollisionsperiode ($\approx 20\ MHz$) getaktet werden. Logische UND-Gatter liefern bei Koinzidenz zwischen Bits der Primärschieberegister, die ein der Vertex-Region zuzuordnendes Spursegment repräsentieren, ein Signal (s. Abb. 4.4). Das ständige Abtasten der Meßdrähte verbunden mit der permanenten Koinzidenzbestimmung ermöglicht die totzeitfreiheie Verarbeitung. Diese Logik ist für CIZ und COZ prinzipiell gleich. Spursegmente, die dem gleichen Ort in der Vertex-Region zuzuordnen sind, werden durch logische ODER-Gatter verknüpft. Masken, die weder die Meßdrahtebene noch die Kathoden-Ebene der Driftzellen schneiden, werden als *Normal-Masken*, die Meßdrahtebene schneidende als *Sense-Crossing-Masken*, die Kathodenebene schneidende als *Boundary-Masken* bezeichnet. Diese Unterscheidung in der Nomenklatur liegt in der unterschiedlichen Verarbeitung in der Kombinationslogik begründet.

### Sekundärmasken

Die Kombinationslogik erzeugt durch Verknüpfung der von der Eingangslogik in CIZ und COZ gefundenen Spursegmenten das Z-Vertex-Histogramm [Zim90c]. Die Summation der Z-Vertex-Histogramm-Bins erfolgt durch analoge Addition der Bin-Signale, da eine digitale Summation eine zu umfangreiche Schaltung erfordern würde, und zu hohe Signallaufzeiten aufweisen würde.

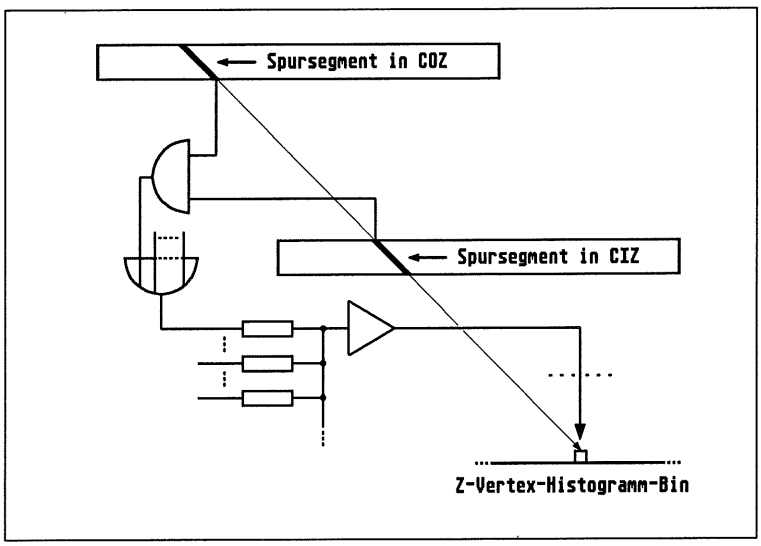


Abbildung 4.4: **Sekundärmaske** *der Kombinationslogik in schematischer Darstellung.*

## 4.2.3 Histogrammanalyse

Die vorerst konventionell aufgebaute Histogramm-Karte liefert durch Analyse des Z-Vertex-Histogramms und der Zell-Histogramme das eigentliche Z-Trigger-Signal. Die Histogrammanalyse wird möglicherweise in einer späteren Ausbaustufe mit Hilfe neuronaler Netzwerke durchgeführt werden. Diese Möglichkeit wurde im Rahmen der vorliegenden Arbeit durch Simulation untersucht (s. Kapitel 6 & 7).

### Z-Vertex-Histogramm

Für die Bewertung eines Ereignisses hoher Multiplizität wird eine Signifikanzbedingung für gefundene Spitzen im Histogramm in analytischer Weise definiert und durch analoge Elektronik näherungsweise nachgebildet. Die Signifikanz $s$ einer Spitze des Z-Vertex-Histogramms ist durch den Ausdruck

$$s = \frac{p - b}{\sqrt{p}} \tag{4.1}$$

definiert. Dabei ist $p$ die Höhe der größten Spitze plus der Höhe des zweitgrößten dazu benachbarten Bins und $b$ die Summe über alle Z-Vertex-Histogramm-Bins, ausgenommen $\pm 2$ Bins um das $p$-Bin, dividiert durch zwei mal die Anzahl der Bins. Für $S > 1,5$ wird ein Ereignis als physikalisch interessant klassifiziert. Es wird angestrebt, Signifikanzen von $> 1,3$ zuzulassen, um Zwei-Spur-Ereignisse triggern zu können und so einen Verlust von interessanten Ereignissen niedriger Multiplizität zu vermeiden, wenn zum Beispiel als Reaktionsprodukte Lepton-Antilepton-Paare $l\bar{l}$ entstehen, wie etwa bei $ep \rightarrow ep + c\bar{c}$ $(J/\Psi)$ und $c\bar{c}$ $(J/\Psi) \rightarrow e^+e^-$ oder $\mu^+\mu^-$ oder $\gamma\gamma \rightarrow e^+e^-$ oder $\mu^+\mu^-$ [Phy0488].

### Zell-Histogramme

Die Einbeziehung der Zell-Histogramme in die Bewertung eines Ereignisses erlaubt die Verwerfung, wenn der größere Anteil der in CIZ und COZ gefundenen Spursegmente Bereichen außerhalb der Vertex-Region zuzuordnen ist. Das wirkt der Vortäuschung eines signifikanten Histogramm-Bins des Z-Vertex-Histogramms aufgrund zufälliger Koinzidenzen entgegen. Zudem sind Ereignisse niedriger Multiplizität bei ausschließlicher Bewertung des Z-Vertex-Histogramms nur schwer separierbar, da Untergrundereignisse bevorzugt $2 - 3$ Einträge im Z-Vertex-Histogramm aufweisen.

### Neuronale Klassifizierung

Die analytische Formulierung der Regeln zur Klassifizierung eines Ereignisses anhand der vorliegenden Histogramme, die in der Simulation durch geeigneten Programm-Code erfolgen kann, erfordert einen hohen Aufwand an Elektronik. Wie im Verlauf der vorliegenden Arbeit noch deutlich wird, ist ein geeignetes neuronales Netzwerk ebenfalls in der Lage, derartige Klassifizierungsregeln zu repräsentieren. Im Gegensatz zum konventionellen Ansatz erfolgt dabei jedoch nicht die explizite Formulierung analytischer Regeln, sondern das neuronale Netzwerk wird durch Training mit Beispieldaten für eine vorgesehene Aufgabe präpariert. Da die Informationsverarbeitung eines neuronalen Netzwerks ausschließlich von den in der Trainingsphase iterativ bestimmten Parametern definiert wird, ist die Implementation des neuronalen Netzwerks von den Klassifizierungsregeln unabhängig. Zudem lassen sich neuronale Netzwerke in Form von hochparallel arbeitenden elektronischen Schaltkreisen realisieren und dann für zeitkritische Aufgaben einsetzen (s. Kapitel 5).

In der konventionell aufgebauten L1-Trigger-Stufe wird nur die Summe der Untergrund-Zell-Histogramme zur Entscheidnung herangezogen. Comuptersimulationen[2] haben aber gezeigt, daß man die Untergrundunterdrückung etwa um einen Faktor 4 verbessern kann, wenn man die Feinstruktur aller verfügbaren Histogramme des Z-Triggers auswertet.

[2]Behrend, H.-J., DESY

## 4.3 Elektronische Realisierung

Die ausgedehnte Vertex-Region führt zu *ca.* 300/*Zelle* Primär- und Sekundärmasken, aufgrund der vielen Koinzidenzmöglichkeiten zwischen den Spursegmenten von CIZ und COZ. Daher ist die Elektronik des Z-Triggers weitgehend mit programmierbaren Logikbausteinen ($\approx$ 1500 Chips), sogenannten LCA[3] der Firma Xilinx, realisiert [Xilinx89]. Diese Chips beinhalten einige tausend elementare Gatterfunktionen, die zu sogenannten Logik-Zellen zusammengefaßt sind. Mit LCA lassen sich komplexe digitale Schaltungen in flexibler Weise realisieren. Der Entwurf erfolgt dabei mit Hilfe eines LCA-Entwicklungssystems, das auf IBM-PC und UNIX-Workstations läuft. Die Konfiguration und Verschaltung der Logik-Zellen erfolgt durch Laden der LCA mit Konfigurationsdaten. So ist gegebenenfalls eine Änderung des Z-Trigger-Masken-Satzes möglich, ohne die Baugruppen selbst verändern zu müssen.

Die Ansteuerung des Z-Triggers erfolgt über einen VMEbus [VME86]. Über diesen werden die LCA mit den Konfigurationsdaten geladen. Zur Logikverifikation und Überwachung (Monitoring) können über den VMEbus Testsignale geschrieben und gelesen werden.

### VMEbus

Der VMEbus stellt ein genormtes Bus-System für 16-Bit und 32-Bit Mikroprozessoren dar [VME85]. Für den Industriestandard "VMEbus" stehen Baugruppen wie Mikrocomputer, Speichermodule, digitale wie analoge Ein-Ausgabe-Module, etc. zur Verfügung. Der VMEbus stellt gleichsam einen "verlängerten" Bus des von der Firma *Motorola Semiconductors* entwickelten 16-Bit Mikroprozessors MC 68000 dar [Mot86], zu dem mittlerweile aufwärtskompatible 32-Bit-Prozessoren verfügbar sind (MC 68020, MC 68030, MC 68040). Stellvertretend für die Gesamtserie wird häufig die Bezeichnung 68K oder 680x0 verwendet. Die Macintosh-Computer der Firma *Apple* basieren auf dieser Prozessorfamilie.

[3] Logic Cell Array

# Kapitel 5

# Neuronale Netzwerke

Auf biologischen oder physikalischen Modellen aufbauende künstliche neuronale Netzwerke ähneln in ihrer Architektur, ihren Eigenschaften und Fähigkeiten bezüglich der Informationsverarbeitung eher dem Gehirn als dem Computer. Dabei ist die Topologie künstlicher, neuronaler Netzwerke gegenüber dem biologischen Vorbild (s. Abb. 5.1) stark idealisiert. Und obwohl künstliche neuronale Netzwerke jeweils nur Teilaspekte berücksichtigen, leisten sie doch einen Beitrag zum Verständnis des Gehirns. Zudem erschließen sie neue technische Anwendungsgebiete, die unter Umständen mit konventionellen Lösungsansätzen nur schwer zugänglich sind.

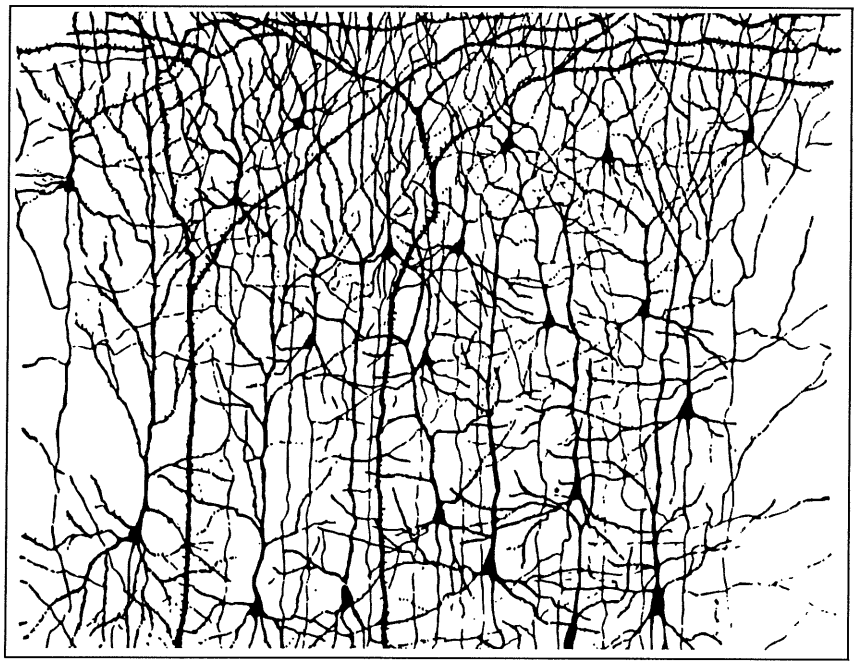

Abbildung 5.1: **Schnitt durch Hirnrinde** *[McCeRum86b].*

Einige Vorteile künstlicher neuronaler Netzwerke gegenüber konventionellen Rechnerarchitekturen verwischen, solange sie auf solchen nur simuliert werden. Stärken zeigen sich jedoch auch dabei, wenn sich eine Problembeschreibung mit analytischen Methoden als schwierig erweist. Die Informationsverarbeitung neuronaler Netzwerke läßt sich gut elektronisch nachbilden, so daß sich die Stärken des biologischen Vorbildes mit der hohen Verarbeitungsgeschwindigkeit elektronischer Schaltungen verbinden lassen.

## 5.1 Das biologische Vorbild

Aus neurologischen Beobachtungen, etwa bei Erkrankungen oder Verletzungen des Gehirns, stammen Erkenntnisse über die Hirnphysiologie. So wird das Gehirn in funktionale Einheiten aufgeteilt, die bestimmten Aufgaben zugeordnet werden können. Der innere Teil des Gehirns ist im wesentlichen für die Steuerung lebenswichtiger Grundfunktionen des Organismus verantwortlich. In der Hirnrinde, dem Cortex (s. Abb. 5.1), lassen sich Bereiche lokalisieren, die höheren Funktionen wie der Motorik und Sensorik [GuN80] zugeordnet werden können.

Mikrobiologische Untersuchungen zeigen Unterschiede im anatomisch-mikroskopischen Aufbau, d.h. in der Netzstruktur bestimmter Regionen der Großhirnrinde, die als Brodmannsche Felder bezeichnet werden [McCeRum86b]. Elektrische und chemische Untersuchungen an einzelnen Nervenzellen und Nervenfasern sowie kleinen Verbunden dieser liefern Erkenntnisse über die Mechanismen des Informationsaustausches der Zellen untereinander.

Die Psychologie, insbesondere die kognitive Psychologie, macht mit Experimenten zum Gedächtnis, Verhalten oder zur Wahrnehmung des visuellen Systems die besonderen Fähigkeiten des Gehirns als Ganzes deutlich [AndJ88]. Dabei versucht die Psychologie höhere mentale Prozesse mehr phänomenologisch zu beschreiben als sie auf mikroskopische Mechanismen zurückzuführen.

### 5.1.1 Neuronen, Axone, Dendriten und Synapsen

Das Gehirn des Menschen besteht aus etwa $10^{10} \ldots 10^{11}$ Nervenzellen - Neuronen, wobei jede durchschnittlich mit $10^4$ anderen Nervenzellen vernetzt ist. Daraus ergeben sich $10^{14} \ldots 10^{15}$ mögliche Verbindungen. Die Verknüpfung der Neuronen untereinander weist dabei grundsätzlich die in Abbildung 5.2 dargestellte Struktur auf.

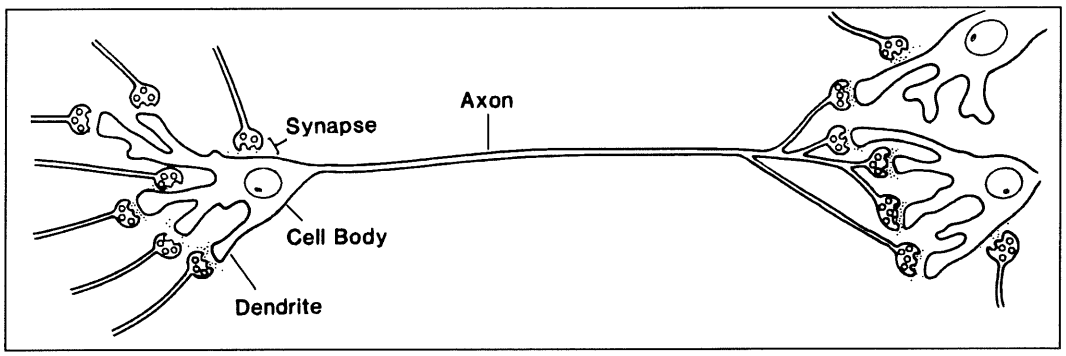


Abbildung 5.2: **Neuron, Axon, Dendriten, Synapsen** *[McCeRum86b]. Die Neuronen stellen zusammen mit den Axonen, Dendriten und Synapsen die funktionalen Einheiten der Informationsverarbeitung dar. Man findet sie in künstlichen neuronalen Netzwerken im Modell berücksichtigt.*

Das biologische neuronale Netzwerk weist in seiner Gesamtheit unterschiedliche Topologien auf, in der sich rückgekoppelte (*feed back*[1]) und nicht rückgekoppelte (*feed forward*[2]) Teilnetze finden lassen, die ihrerseits wieder untereinander teilweise vernetzt sind.

[1] *engl. feed back* ≈ rückwärts (wieder) zuführen
[2] *engl. feed forward* ≈ vorwärts zuführen

Die Attribute rückgekoppelt und nicht rückgekoppelt beziehen sich auf den Informationsfluß im neuronalen Netzwerk. Im Fall von *Feedforward*-Netzen fließt Information nur vom Eingang zum Ausgang, bei *Feedback*-Netzwerken wird Ausgabeinformation, oder ein Teil davon, wieder dem Eingang zugeführt.

#### Das Neuron

Die Zahl der Neuronen ist praktisch kurz nach der Geburt festgelegt, da sich diese nicht, wie andere Zellen, durch Teilung vermehren können. Sie bilden die elementaren, informationsverarbeitenden Schaltelemente der Netzstruktur. Ihre Funktion besteht im gewichteten Aufsummieren von Eingangssignalen. Überschreitet diese Summe eine Schwelle, so liefert das betreffende Neuron einen kurzzeitigen Ausgangsimpuls, das sogenannte Aktionspotential.

#### Das Axon

Das Ausgangssignal eines Neurons wird über das Axon zu anderen Neuronen weitergeleitet. Das Axon stellt eine röhrenförmige Leitungsbahn mit einem Durchmesser von $5 \ldots 100 \mu m$ und einer Länge bis zu $1m$ dar. Der Signaltransport erfolgt auf elektrochemischem Wege. Die für diesen aktiven Signaltransport benötigte Energie wird durch die Oxidation von Nährstoffen bereitgestellt. Die aktive Signalleitung wirkt der ohmschen Bedämpfung entlang des Axons entgegen, und ermöglicht so erst lange Signalwege. Neben marklosen Nervenfasern gibt es markhaltige, die gegenüber den marklosen eine bis zu hundertfache Signalgeschwindigkeit von bis zu 120 $m/s$ aufweisen.

#### Die Dendriten

Die dünn verzweigten Ausläufer der Neuronen werden als Dendriten bezeichnet. Sie stellen die Eingänge der Neuronen dar. Im Durchschnitt liegt die Zahl der Dendriten eines Neurons etwa bei $10^4$, kann aber bis zu $2 \cdot 10^5$ betragen.

#### Die Synapsen

Die Synapsen stehen als verzweigte Endstücke der Axone mit den Dendriten der Neuronen in Verbindung. Die Übertragung des (elektrischen) Nervensignals über den synaptischen Spalt zwischen der Synapse und dem Dendriten erfolgt auf chemischem Wege über sogenannte Neurotransmitter. Ein Transmitter wirkt entweder erregend oder hemmend. Es wird funktionell zwischen erregenden (exzitatorischen) und hemmenden (inhibitatorischen) Synapsen unterschieden.

### 5.1.2 Funktionsweise biologischer Nervenzellen

Im Ruhezustand besteht zwischen dem Inneren und Äußeren einer Nervenzelle, an der Zellmembran, ein Ruhepotential von etwa $-70\ mV$. Die über die Dendriten zum Neuron gelangenden elektrischen Nervensignale bewirken eine Depolarisation der Zellmembran hin zu positiven Werten. Überschreitet die Summe der Nervensignale eine für ein bestimmtes Neuron charakteristische Schwelle, so kommt es zur vollständigen Depolarisation der Zellmembran bis zu etwa $+30\ mV$, dem Aktionspotential. Man sagt: Das Neuron *feuert*. Der zeitliche Verlauf eines solchen Aktionspotentials, gemessen an einer Nervenfaser, ist in Abbildung 5.3 dargestellt.

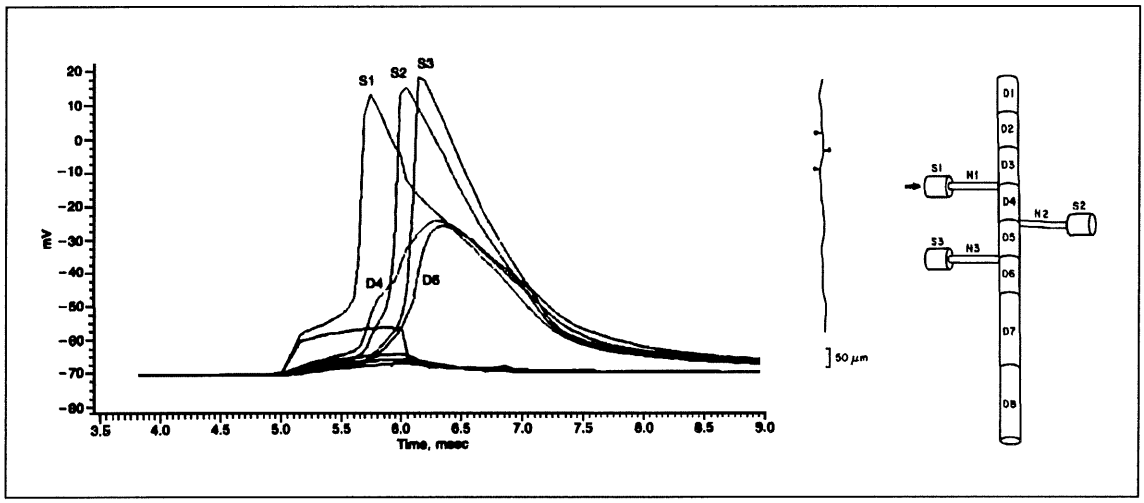


Abbildung 5.3: **Aktionspotential** *[McCeRum86b]. Gemessene Reaktion an Dendriten einer Nervenfaser, die mit D4 und D6 bezeichnet sind, auf die sich zeitlich überlappenden Signale an Synapsen S1, S2 und S3.*

### Ruhepotential

Die Zellulärflüssigkeit enthält hauptsächlich $Na^+$ , $Ka^+$ und $Cl^-$ Ionen, die für das elektrische Verhalten der Nervenzellen entscheidend sind. In der Zellmembran befinden sich Ionenpumpen, welche $Na^+$ aus der und $Ka^+$ in die Zelle pumpen (s. Abb. 5.4). Dabei werden jedoch mehr $Na^+$ aus der Zelle als $Ka^+$ in die Zelle gepumpt, was das negative Ruhepotential zur Folge hat. Die zum Pumpen benötigte Energie wird durch Stoffwechselreaktionen freigesetzt. Im Zellinneren befinden sich neben $Cl^-$ , die durch die Zellmembran diffundieren können, negative Aminosäuren und Protein-Ionen, die aufgrund ihrer Größe die Zelle nicht verlassen können. Neben den Ionenpumpen befinden sich in der Zellmembran getrennte Kanäle für Natriumionen und für Kaliumionen. Diese Ionenkanäle sind im Ruhezustand geschlossen.

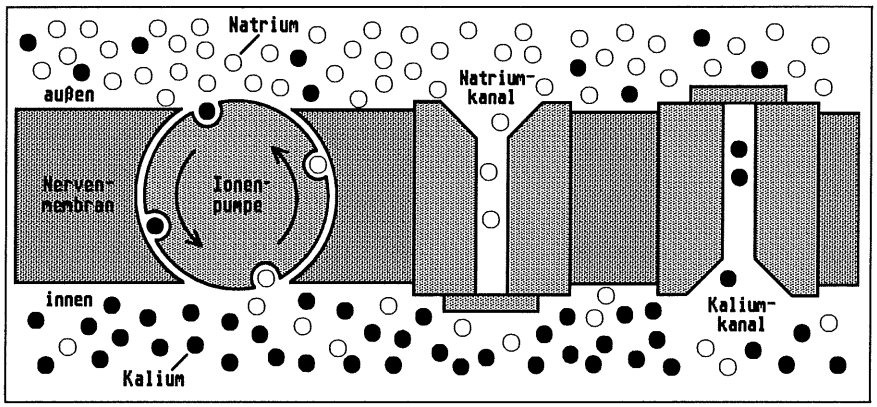


Abbildung 5.4: **Zellmembran** *[GuN80]. Schematische Darstellung der Zellmembran eines Neurons, mit den für das elektrische Verhalten wichtigsten Funktionselementen.*

### Aktionspotential

Die Ionenverteilung zu beiden Seiten der Membran ist für das Ruhepotential an der Zellmembran verantwortlich. Die Depolarisation der Zellmembran bewirkt zunächst eine Erhöhung der Öffnungswahrscheinlichkeit der Natriumkanäle. Wird eine bestimmte Schwelle überschritten, erreicht die Membran einen positiven Rückkopplungszustand, wodurch die Durchlässigkeit für die $Na^+$ weiter ansteigt, so daß $Na^+$ bis zum vollständigen Konzentrationsausgleich in die Zelle einströmt. Ist dieses Einströmen beendet, öffnen sich die Kaliumkanäle, und die $Ka^+$ strömen aus. Die schlagartige Änderung der Ionenkonzentrationen gegenüber dem Ruhezustand bewirken das impulsartige Aktionspotential. Die Ionenpumpen stellen die Konzentrationsverhältnisse des Ruhezustandes nach einiger Zeit, der *Refraktärzeit*, wieder her. Die Ionenkanäle stellen gleichsam elektrische Ventile dar. Neben elektrisch gesteuerten findet man, insbesondere in den Synapsen, auch chemisch gesteuerte Ionenkanäle.

### Aktivität

Die Aktionspotentiale aller Neuronen sind in ihrer Impulsamplitude gleich, bedingt durch die Art der elektrochemischen Mechanismen an den Zellmembranen. Der Informationsgehalt der Nervensignale liegt in der Frequenz ihres Auftretens und nicht in ihrer Amplitude, und wird als *Aktivität* des Neurons bezeichnet. Selbst längere Leitungswege führen daher nicht zu einer Verfälschung der Information. Die auftretenden Impulsfrequenzen liegen im Bereich von $0,1 \ldots 1kHz$. Der zeitliche Abstand zwischen zwei Aktionspotentialen ist durch die Refraktärzeit bestimmt. Obwohl die Verarbeitungsgeschwindigkeit einzelner Neuronen im Bereich von Millisekunden liegt, erreicht das Gehirn durch die Parallelität der Informationsverarbeitung, bei bestimmten Aufgaben, eine überaus hohe Verarbeitungsgeschwindigkeit.

### Signaltransport

Die Weiterleitung des Aktionspotentials über das Axon beruht in ähnlicher Weise auf den beschriebenen elektrochemischen Mechanismen. Die Signalleitung erfolgt primär weg vom Neuron. Dennoch bewirkt eine länger anhaltende, gleichzeitige erhöhte Aktivität direkt verbundener Neuronen langfristig eine Veränderung der synaptischen Kopplungsstärken. Dieser Umstand ist im Zusammenhang mit dem Lernen von Bedeutung, siehe Abschnitt 5.1.3.

Funktional stellt das Neuron einen Verstärker mit nichtlinearer Ausgangscharakteristik dar. Es liefert ein Aktionspotential, welches über das Axon zu anderen Neuronen weitergeleitet wird, wenn die Summe seiner gewichteten Eingänge einen Schwellwert überschreitet. Die Informationsspeicherung im neuronalen Netzwerk verbirgt sich in den Kopplungsstärken zwischen Synapsen und Dendriten. Lernen bedeutet aus der mikrobiologischen Sicht das Verändern dieser synaptischen Kopplungsstärken in der Weise, daß dabei eine Adaption an eine bestimmte Aufgabenstellung stattfindet.

Die dargestellten mikrobiologischen Mechanismen sind in [Net87], [GuN80], [McCeRum86b] und [NNW90] detailliert beschrieben.

## 5.1.3 Die Hebb'sche Lernregel

Im Jahr 1949 formulierte Donald O. Hebb ein neurophysiologisches Postulat bezüglich des Lernens, die Hebb'sche Lernregel. Diese Lernregel ist wörtlich in der Originalarbeit [Hebb49] zu finden. Sie lautet sinngemäß:

*Wenn ein Axon einer Zelle A nahe genug an einer Zelle B ist, um diese anzuregen, oder zu ihrer Anregung beizutragen, so kommt es zu einem Wachstumsprozeß oder einer Stoffwechselveränderung in einer oder beiden Zellen, so daß die Effizienz des Einflusses des Feuerns von A auf B verstärkt wird.*

Diese Lernregel erfaßt das Lernen von der mikrobiologischen Seite. Wesentlich dabei ist die Tatsache, daß es sich bei dieser Regel um eine lokale Lernregel handelt, da nur die Kenntnis der lokalen Aktivitäten zweier benachbarter Nervenzellen für die Anwendung der Lernregel notwendig ist. Viele Lernalgorithmen künstlicher neuronaler Netzwerke weisen ebenfalls diesen lokalen Charakter auf. Sie stellen gleichsam eine Erweiterung der Hebb'schen Lernregel dar.

## 5.2 Mathematische Modellbildung

Die in Abschnitt 5.1 beschriebenen Mechanismen des biologischen Vorbildes neuronaler Netzwerke lassen sich für die mathematische Modellbildung heranziehen. Dabei werden Vereinfachungen vorgenommen, deren Rechtfertigung sich zunächst einmal aus dem funktionsfähigen Netzwerkmodell selbst ergibt. Ein ausformuliertes mathematisches Modell erlaubt dann die Untersuchung dieses Modells mit den Mitteln der Mathematik. Dazu sei jedoch bemerkt, daß ein verifiziertes mathematisches Modell nicht zwingend einen Beitrag zum Verständnis des biologischen Vorbildes liefert vielmehr dient es eher dem Zweck, durch einen verfeinerten Formalismus, die Handhabung eines künstlichen neuronalen Netzwerks zu ermöglichen oder zu erleichtern.

Die biologische Zielsetzung muß andere Schwerpunkte aufweisen, wie etwa die möglichst nahe Nachbildung des biologischen Neurons, mit elektronischen Mitteln [MfNN90], oder die Berücksichtigung der komplexeren Topologie biologischer neuronaler Netzwerke (Jordansche Netze [Jor86]).

Das von Hopfield eingeführte gleichnamige Modell schlägt eine Brücke zwischen dem (biologischen) neuronalen Netzwerk und dem Spinglasmodell der Festkörperphysik [Hop82]. Daraus ergibt sich die Möglichkeit, die in der Festkörperphysik eingehend untersuchten Mechanismen, Begriffe und mathematischen Beschreibungsmöglichkeiten für die Anwendung auf neuronale Netzwerke nutzbar zu machen.

### 5.2.1 Modell des Neurons

Die Aktivität eines Neurons, die Rate des Feuerns, läßt sich im mathematischen Modell durch eine reelle Zahl $y = -1 \ldots +1$ beschreiben. Das Feuern des Neurons läßt sich dabei als Wahrscheinlichkeit interpretieren, ein Ausgangssignal in Abhängigkeit von der gewichteten Summe der Eingangssignale zu liefern. Die Aktivität des Neurons $i$ wird durch folgende Gleichung beschrieben

$$y_i = g(\theta_i + \sum_{j=1}^{n} \omega_{ij} \cdot x_j). \tag{5.1}$$

Dabei ist $x_j$ der $j$-te Eingang des Neurons $i$ gewichtet mit der synaptischen Kopplungsstärke $\omega_{ij}$, wobei $\omega$ Werte zwischen $-1 \ldots +1$ annehmen kann, also von hemmend (inhibitatorisch) bis erregend (exzitatorisch), siehe Abschnitt 5.1.1. Die Größe $\theta_i$ beschreibt die Schwelle des Neurons $i$. Die Rechenvorschrift zur Verknüpfung der Eingangsaktivitäten $x_j$ wird in der Literatur auch als *Propagierungsfunktion* bezeichnet. Neben der in Gleichung 5.1 dargestellten werden auch

andere Funktionen verwendet, die keine direkte Analogie zum biologischen Vorbild aufweisen, für bestimmte technische Anwendungen jedoch von größerem Nutzen sind als das Abbild der Biologie.

### Aktivierungsfunktion

Die Funktion $g$ beschreibt, in Analogie zum biologischen System, die Wahrscheinlichkeit des Feuerns eines Neurons. Sie wird als *Aktivierungsfunktion* oder *Transferfunktion* bezeichnet und hat folgende Form

$$g(x) = \frac{1}{1+\exp(-x/T)} \qquad \text{wobei} \qquad \lim_{T\to 0} g(x) = \begin{cases} 0 & \text{für } x \leq 0 \\ +1 & \text{für } x > 0. \end{cases} \tag{5.2}$$

Der Parameter $T$ (Temperatur) bestimmt die Steilheit der Aktivierungsfunktion $g$ und wird meist auf den Wert $T = 1$ gesetzt. Die Form dieser Funktion läßt sich direkt aus elektrochemischen Messungen an Membranen von Nervenzellen ableiten [Stry90]. Die Funktion $g$ ist in Abbildung 5.5 skizziert. Für $T \to 0$ geht die sigmoide Funktion $g$ in die *Heavysidesche Sprungfunktion* über.

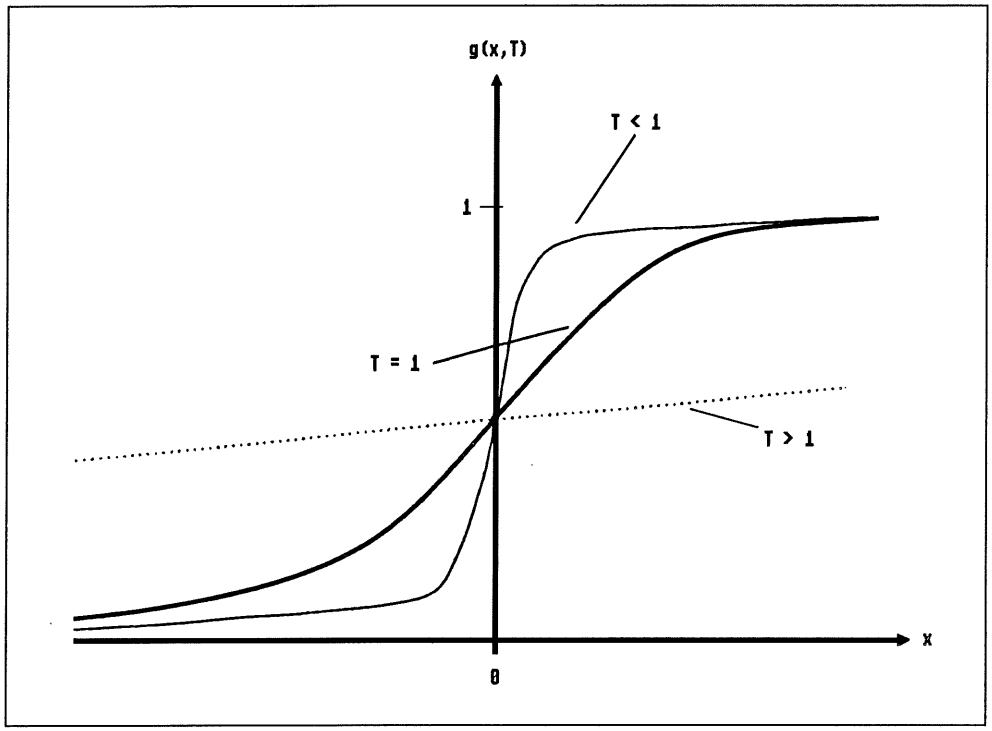


Abbildung 5.5: **Aktivierungsfunktion des Neurons** *[NNW90]. Eine unstetige Aktivierungsfunktion findet man bei manchen künstlichen neuronalen Netzen. Im biologischen System zeigt sie jedoch den sigmoiden Verlauf, bedingt durch die komplexen elektrochemischen Prozesse der Kopplung zwischen Synapse und Dendrit.*

Prinzipiell kann jedes Neuron eines Netzwerks eine andere Aktivierungsfunktion aufweisen. Meist wird jedoch dieselbe Aktivierungsfunktion für alle Neuronen eines Netzwerks verwendet.

Anstelle der dargestellten, an die Biologie anlehnenden Funktion $g$, werden in künstlichen neuronalen Netzen auch andere Funktionen als Aktivierungsfunktionen verwendet. Für die Auswahl einer bestimmten Funktion zur Anwendung auf ein gegebenes Problem gibt es keine allgemeingültigen Regeln, so daß empirisch vorgegangen werden muß. Eine Richtungsweisung läßt sich unter Umständen durch Vergleich der gegebenen Aufgabenstellung mit bestehenden Lösungsansätzen ähnlicher Probleme gewinnen. Beispiele anderer Aktivierungsfunktionen, die nicht in Analogie zur Biologie stehen, sind

$$\begin{array}{rcl} g(x) & = & \begin{cases} 0 & \text{für } x < 0 \\ ax + b & \text{für } x \geq 0 \end{cases} \\ g(x) & = & \begin{cases} -1 & \text{für } x < 0 \\ +1 & \text{für } x \geq 0 \end{cases} \\ g(x) & = & \tanh(x) \\ g(x) & = & \exp(x) \\ g(x) & = & \sin(x) \\ & \vdots & \end{array} \tag{5.3}$$

Für die mathematische Beschreibung ist der stetige, sigmoide Verlauf der Aktivierungsfunktion $g$ bezüglich der Konvergenz des Lernens eines mehrschichtigen neuronalen Netzes von zentraler Bedeutung, da die Veränderung der synaptischen Gewichte $\omega$ dabei in Abhängigkeit von der Ableitung der Aktivierungsfunktion $g$ erfolgt, siehe Abschnitt 5.4.

### Schwelle und Bias

Im Fall der Gleichung 5.1 muß beim Lernen neben den Gewichten $\omega$ zusätzlich die Schwelle $\theta$ verändert werden, woraus sich zunächst die Notwendigkeit ergeben würde, eine eigene Lernregel für die Schwellen $\theta_i$ aufzustellen. Auf diese läßt sich durch Einführen eines zusätzlichen Eingangs $x_0$ des Neurons $i$, welcher mit einem Neuron konstanter Aktivität von eins verbunden ist, verzichten. Dadurch kann die Schwelle des Neurons $i$ quasi durch Verändern des zusätzlichen Gewichtes $\omega_{i0}$ modifiziert werden. Die Gleichung 5.1 nimmt dann die folgende Form an

$$y_i = g(\sum_{j=0}^{n} \omega_{ij} \cdot x_j) \qquad \text{mit} \qquad x_0 = 1. \tag{5.4}$$

Das Neuron mit der konstanten Aktivität von eins wird in der Literatur als *Bias*[3] bezeichnet. Die Schwelle $\theta$, bzw. $\omega_{i0} \cdot x_0$, dient zusammen mit der nichtlinearen Aktivierungsfunktion $g$ zur Diskriminierung, das heißt Klassifizierung, der gewichtet aufsummierten Eingangsaktivitäten.

## 5.3 Lernen und Ausführen

Man unterscheidet bei der Anwendung neuronaler Netze zwischen der *Lernphase* und der *Ausführungsphase*, analog der Unterscheidung zwischen Programmierung und Ausführung im Fall konventioneller Rechner.

[3] *engl. bias* ≈ Einfluß, beeinflussen

### 5.3.1 Lernmodus

Bevor ein neuronales Netz eine Aufgabe wahrnehmen kann, muß es für diese Aufgabe trainiert werden. Dabei wird jedoch der Lösungsweg nicht explizit vorgegeben (programmiert), sondern es werden einem Netzwerk entweder Eingabedaten zusammen mit dazu korrelierten Ausgabedaten angeboten, oder es wird dem Netz eine Bewertungsfunktion vorgegeben. Das neuronale Netz verändert dann solange die die Informationsverarbeitung bestimmenden Netzwerkparameter nach einem Lernalgorithmus, bis die Verarbeitung der angebotenen Information nahe genug an Vorgaben heranreicht.

### 5.3.2 Ausführungsmodus

Wenn ein Netzwerk hinreichend gut trainiert ist, so reagiert es auf angebotene, nicht gelernte, Eingabedaten mit dazu angemessenen Ausgaben. Dieses hängt jedoch sehr stark von dem für die Aufgabe verwendeten Netzwerktyp, der darin realisierten Netztopologie, der Auswahl der Lerndaten sowie der Anzahl der Lernschritte ab. Die Fähigkeit neuronaler Netze auf eine unbekannte, also nicht zuvor gelernte, Eingabe in angemessener Weise zu reagieren, wird als *Generalisation* bezeichnet, was bedeutet, daß ähnliche Eingabedaten zu ähnlichen Ausgaben führen. Das Attribut *ähnlich* bedeutet dabei, daß sich die Aktivitäten der Eingangs- bzw. Ausgangsneuronen nur an wenigen Ein- bzw. Ausgängen geringfügig unterscheiden.

## 5.4 Das Backpropagation-Netzwerk

Das Backpropagation-Netzwerk, abgekürzt **BPG**, wurde im Jahr 1986 zusammen mit dem gleichnamigen Lernalgorithmus von McCelland und Rumelhart vorgestellt [McCeRum86a]. Es handelt sich bei diesem Netzwerk um ein nichtrückgekoppeltes Netzwerk (*feed forward*) mit einer oder mehreren sogenannten verdeckten Neuronenschichten, die in der Literatur im allgemeinen als *Hidden-Layer*[4] bezeichnet werden.

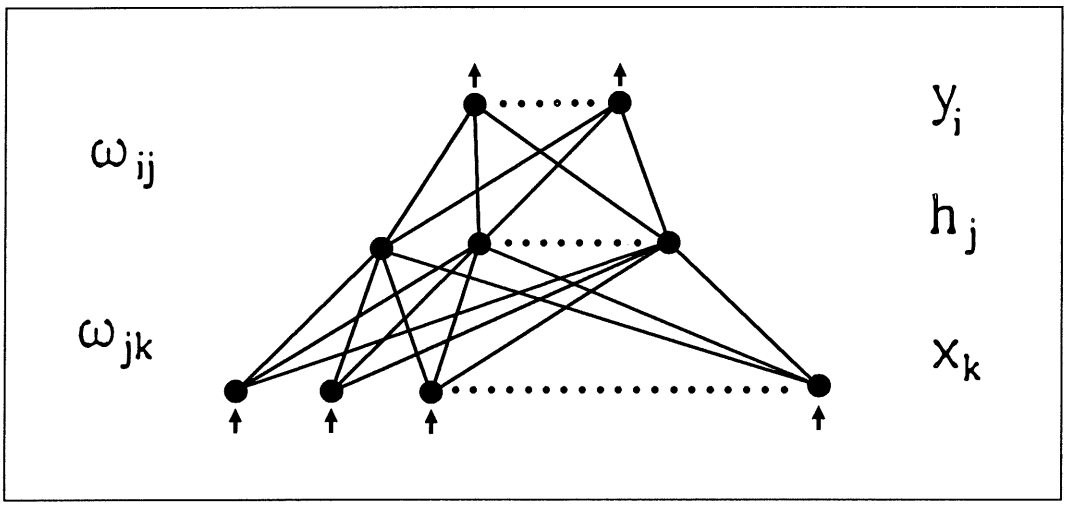


Abbildung 5.6: **BPG-Netz mit einer verdeckten Schicht** *[LUTP90-8].*

[4] *engl. hidden layer* ≈ verdeckte Schicht

Das BPG-Netz eignet sich besonders gut zur Klassifizierung von Mustern. Gerade diese Aufgabenstellung findet man bei einem Trigger: Anhand von Detektordaten soll entschieden werden, ob die vorliegenden Daten von einem physikalisch relevanten Ereignis stammen, oder ob es sich um ein Untergrundereignis handelt, welches nicht ausgewertet wird.

Das an den Eingängen des neuronalen Netzes anliegende Muster läßt sich als Spaltenvektor $\vec{x}$ darstellen, dessen Komponenten die an den Eingangsneuronen anliegenden Aktivitäten sind. Die vom Netz berechneten Aktivitäten der Ausgänge lassen sich ebenfalls als Spaltenvektor $\vec{y}$ darstellen

$$\vec{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_k \end{pmatrix} \quad \text{und} \quad \vec{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_i \end{pmatrix}. \tag{5.5}$$

Ein am Eingang des Netzwerks anliegender Vektor $\vec{x}$ wird vom Netzwerk auf einen Ausgangsvektor $\vec{y}$ abgebildet. Die Abbildungsvorschrift steckt in den synaptischen Gewichten $\omega$, welche die Kopplungsstärke der Neuronen untereinander bestimmen. Der Eingangsvektor stellt ein zu klassifizierendes Muster dar, der Ausgangsvektor die Klassifizierung oder Eigenschaft. Die Abbildung eines Eingangsvektors $\vec{x}$ über die verdeckte(n) Neuronenschicht(en) auf einen Ausgangsvektor $\vec{y}$ erfolgt über eine Funktion $F$ in der Art

$$\vec{y} = F(\vec{x}). \tag{5.6}$$

Diese Abbildung stellt die numerische Näherung der Funktion $F$ aus einem Satz $M$ bekannter Lernmuster

$$(\vec{x}^{(p)}; \vec{y}^{(p)}) \quad \text{mit} \quad p = 1, \ldots, M \tag{5.7}$$

dar. Jedes einzelne Neuron $n_i$ berechnet dabei eine gewichtete Summe $a$ seiner Eingänge mit anschließender Schwellwertbildung durch eine Aktivierungsfunktion der Art

$$g(a) = \frac{1}{1 + \exp(-a)}, \tag{5.8}$$

wobei für die Hidden-Neuronen $h_j$ bzw. Ausgangs-Neuronen $y_i$ gilt

$$\begin{aligned} h_j &= g(a_j) \\ y_i &= g(a_i) \end{aligned} \tag{5.9}$$

mit

$$\begin{aligned} a_j &= \theta_j + \textstyle\sum_k \omega_{jk} \cdot x_k \\ a_i &= \theta_i + \textstyle\sum_j \omega_{ij} \cdot x_j. \end{aligned} \tag{5.10}$$

Diese Abbildungsvorschriften lassen sich mit Hilfe von Matrizenmultiplikationen in kompakter Weise formulieren

$$\vec{y} = g(\Omega_{ij} \cdot \vec{h}) \qquad \text{und} \qquad \vec{h} = g(\Omega_{jk} \cdot \vec{x}) \tag{5.11}$$

mit

$$\Omega_{ij} = \begin{pmatrix} \omega_{11} & \dots & \omega_{1j} \\ \vdots & \ddots & \vdots \\ \omega_{i1} & \dots & \omega_{ij} \end{pmatrix} \qquad \text{und} \qquad \Omega_{jk} = \begin{pmatrix} \omega_{11} & \dots & \omega_{1k} \\ \vdots & \ddots & \vdots \\ \omega_{j1} & \dots & \omega_{jk} \end{pmatrix}. \tag{5.12}$$

Dabei stellen die Matrixelemente die synaptischen Gewichte der Neuronen dar. Ohne die Aktivierungsfunktion $g$ würde ein Eingangsvektor $\vec{x}$ ausschließlich durch Matrixmultiplikation auf einen Ausgangsvektor $\vec{y}$ abgebildet werden. Es ist eine elementare Eigenschaft der Matrixmultiplikation, einen Satz orthogonaler Vektoren auf einen beliebigen Satz anderer Vektoren abzubilden. Ohne die *nichtlineare* Aktivierungsfunktion $g$ ließen sich die Matrixelemente $\omega$ analytisch bestimmen. Nur könnte ein solches Netzwerk nur zueinander orthogonale Eingangsmuster unterscheiden (klassifizieren), und wäre somit für komplexe Aufgabenstellungen ungeeignet. Auch hätte die versteckte Neuronenschicht keine Bedeutung, da sich die Abbildung $\vec{y} = F(\vec{x})$, aufgrund der Assoziativität der Matrixmultiplikation durch eine einzige darstellen ließe.

### 5.4.1 Der BPG-Lernalgorithmus

Lernen bedeutet die Einstellung der synaptischen Gewichte in der Weise, daß die Korrelation zwischen Eingabevektoren $\vec{x}^{(p)}$ und Ausgabevektoren (Eigenschaftsvektoren) $\vec{y}^{(p)}$ vom Netzwerk, gemäß der Abbildungsvorschrift $\vec{x} = F(\vec{y})$, bis auf einen festgelegten Fehler durchgeführt wird. Entscheidend ist dabei, daß zum Lernen andere Muster verwendet werden, als in der Anwendungsphase des Netzwerks. Die Funktion $F$ stellt, aufgrund der Aktivierungsfunktion $g$, ein nichtlineares Gleichungssystem dar. Der BPG-Algorithmus ist ein Verfahren zur numerischen Lösung dieses Gleichungssystems. Es ist verwandt mit dem *Gradientenverfahren* zur Lösung nichtlinearer Gleichungssysteme [Bron81], bei dem die Variationsparameter des Gleichungssystems immer in Richtung des *steilsten Gradienten* in Bezug zur Fehleränderung adaptiert werden.

#### Mittlerer quadratischer Fehler

Vor dem Lernen werden die $\omega_{ij}$ und $\omega_{jk}$ mit zufälligen Werten, im Bereich $-0.1 \dots + 0.1$ vorbesetzt. Beim Lernen mit dem BPG-Algorithmus werden die Gewichte $\omega$ dann so verändert, daß, für eine *konstante* Zahl von $M$ Lernmustern, der mittlere quadratische Fehler

$$E = \frac{1}{2} \sum_p \sum_i (y_i^{(p)} - t_i^{(p)})^2 \tag{5.13}$$

minimal wird. Dabei stellt $t_i$ die *gewünschte* Ausgabe des $i$-ten Ausgangsneurons dar, wogegen $y_i$ die vom Netz *berechnete* Ausgabe des $i$-ten Neurons darstellt. Mit $p$ werden die $M$ Lernmuster indiziert. Mit zunehmender Anzahl von Lernzyklen werden die Differenzen zwischen den $y_i$ und $t_i$ minimal, so daß mittlere quadratische Fehler $E$ minimiert wird. Neben dieser von McCelland und Rumelhart vorgeschlagenen Fehlerfunktion können auch andere verwendet werden.

### Lernzyklus

Ein Lernzyklus wird in der folgenden Weise durchgeführt: Zunächst wird ein Lernmuster $(x_k^{(p)}; y_i^{(p)})$ zufällig aus dem Satz der $M$ Lernmuster ausgewählt. Dann werden gemäß Gleichung 5.11 zunächst die Aktivitäten der verdeckten Neuronen $h_j$ berechnet, anschließend die der Ausgangsneuronen $y_i$. Die Adaption der Kopplungsfaktoren zwischen den Neuronenschichten erfolgt in der entgegengesetzten Richtung. Zuerst erfolgt die Korrektur der $\omega_{ij}$ zwischen der Ausgangsschicht und der verdeckten Neuronenschicht nach

$$\Delta\omega_{ij} = -\eta\frac{\partial E}{\partial\omega_{ij}} = -\eta\delta_i h_j + \alpha\Delta\omega_{ij}^{old} \tag{5.14}$$

mit

$$\delta_i = (y_i - t_i)g'(a_i), \tag{5.15}$$

anschließend erfolgt die Korrektur der $\omega_{jk}$ zwischen den Neuronen der verdeckten Schicht und der Eingangsschicht nach

$$\Delta\omega_{jk} = -\eta\frac{\partial E}{\partial\omega_{jk}} = -\eta\delta_j x_k + \alpha\Delta\omega_{jk}^{old} \tag{5.16}$$

mit

$$\delta_j = \sum_i \omega_{ij}\delta_i g'(a_j). \tag{5.17}$$

Dabei ist $g'$ die erste Ableitung der Aktivierungsfunktion $g$. Die Größe $\eta$ wird als *Lernparameter* bezeichnet. Ein typischer Wert ist $\eta \approx 0,001$. Zur Dämpfung möglicher Oszillationen von $E$ dienen die Terme $\alpha\Delta\omega_{ij}^{old}$ und $\alpha\Delta\omega_{jk}^{old}$, die als *Impulsterme* bezeichnet werden. Die Größe $\alpha$ wird als *Dämpfungsparameter* bezeichnet und hat einen typischen Wert von $\alpha \approx 0,5$. Die Werte $\Delta\omega_{ij}^{old}$ und $\Delta\omega_{jk}^{old}$ stellen die im vorangegangenen Lernzyklus errechneten Korrekturen dar.

### Fehlersignal

Für die Veränderung der $\omega_{ij}$ zwischen der Ausgangsschicht und der verdeckten Neuronenschicht steht das *Fehlersignal* $\delta_i$ direkt zur Verfügung. Dieses ist für die Gewichte $\omega_{jk}$ zwischen der verdeckten Schicht und der Eingangsschicht zunächst nicht der Fall, da für die Neuronen der verdeckten Schicht nicht bekannt ist, welche Aktivität diese zu liefern haben. Der BPG-Algorithmus überwindet dieses Problem durch *rekursive* Berechnung des Fehlersignals $\delta_j$ in Termen $\delta_i$. Der beschriebene Lernzyklus wird so oft wiederholt, bis hinreichend viele der $M$ Lernmuster, bis auf einen festgelegten Fehler, richtig klassifiziert werden.

### Fehlerrückverfolgung

Die Rückverfolgung - das *Zurückpropagieren* - und Korrektur der Fehler von der Ausgabeschicht zurück zur Eingabeschicht, nach vorheriger Abbildung eines Eingangsvektors auf einen Ausgangsvektor in der entgegengesetzten Richtung, hat dem BPG-Algorithmus den Namen gegeben. Eine direkte Analogie zwischen dem BPG-Algorithmus und dem Lernen biologischer Systeme läßt sich nicht finden. Dennoch weist der BPG-Algorithmus den in der Hebb'schen

Lernregel formulierten lokalen Charakter auf. Die Gleichung 5.17 stellt gleichsam eine Erweiterung der aus der Hebb'schen Regel abgeleiteten Delta-Regel 5.15 und 5.14 dar.

Das Verfahren ist prinzipiell für rückkopplungsfreie neuronale Netze mit beliebig vielen verdeckten Schichten geeignet. Gerade diese Möglichkeit eröffnet den BPG-Netzen die Anwendung auf komplexe Problemstellungen bei der Klassifizierung von Mustern, die mit den Vorläufern des BPG-Netzes prinzipiell nicht möglich sind.

### 5.4.2 Konvergenz

Ziel des Lernens ist das Minimieren der Funktion $E$ (Gleichung 5.13) bis unter eine festgelegte Schranke. Im Idealfall weist die Funktion $E$ nur ein globales Minimum auf, welches vom BPG-Algorithmus gefunden werden kann. Wie auch beim, dem Algorithmus verwandten, Gradientenverfahren folgt das System einer Kurve auf der Fehlerfläche in Richtung des steilsten Gradienten. Im Fall lokaler Minima besteht die Gefahr, daß der Algorithmus nur ein solches findet. Liegt der Fehler dabei unter der festgelegten Schranke, so hat der BPG-Algorithmus *eine mögliche Lösung* jedoch *nicht die beste Lösung* gefunden.

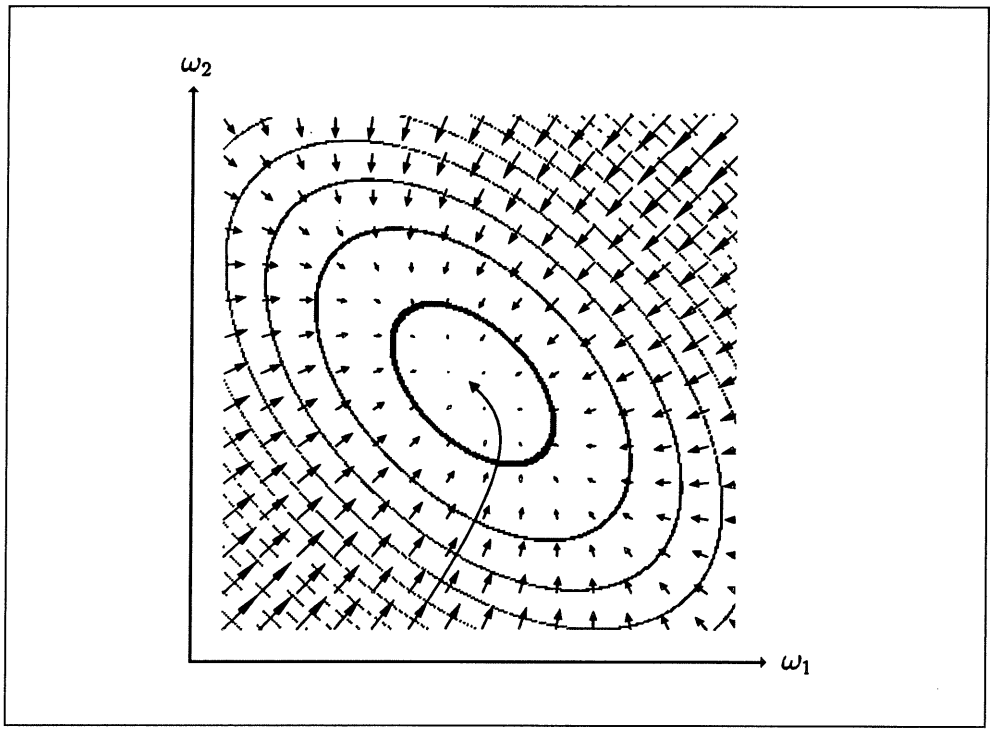


Abbildung 5.7: **Fehlerfläche mit globalem Minimum** *[McCeRum88].*

Die Fehlerfläche stellt ein $n$-dimensionales Gebirge dar, wobei $n$ gleich der Anzahl aller Kopplungskonstanten $\omega$ des neuronalen Netzwerks ist. In Abbildung 5.7 ist eine Fehlerfläche mit einem globalen Minimum für das ODER-Problem[5] skizziert [McCeRum88]. Schon das ODER-Problem erfordert eine nichtlineare Aktivierungsfunktion $g$, verdeckte Schichten sind jedoch nicht erforderlich.

[5]Das Netzwerk hat einen Ausgang und zwei Eingänge. Es bildet die logische Funktion eines digitalen ODER-Gatters nach. Die Lernmuster sind daher: 00→0, 01→1, 10→1, 11→1

In mehrschichtigen neuronalen Netzwerken mit wenigen Hidden-Neuronen ist die Gefahr vieler lokaler Minima besonders groß [McCeRum88]. Als Beispiel ist in Abbildung 5.8 die Fehlerfläche für ein dreischichtiges Netzwerk mit einem Eingangs-Neuron, einem Hidden-Neuron und einem Ausgangs-Neuron skizziert, zur Lösung des Identitätsproblems[6].

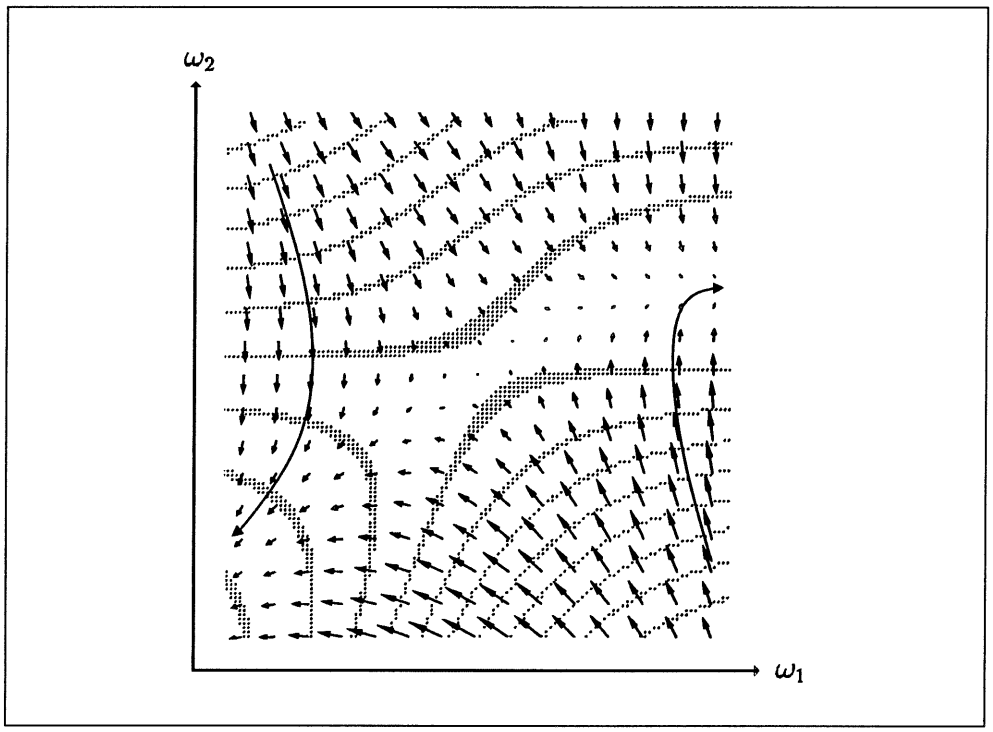


Abbildung 5.8: **Fehlerfläche mit lokalen Minima** *[McCeRum88].*

Der BPG-Lernalgorithmus erfordert die der ersten Ableitung des Fehlers $E$ proportionale Änderung der $\omega$. Die Proportionalitätskonstante ist der Lernparameter $\eta$. Genaugenommen müßten die Änderungen in infinitesimalen Schritten erfolgen, ein finites $\eta$ ist jedoch für die praktische Durchführung notwendig. Es wird gerade so groß gewählt, daß keine Oszillationen auftreten. Durch die Impulsterme kann $\eta$ vergrößert werden, ohne daß Oszillationen auftreten. Ein großes $\eta$ führt demnach zu großen Änderungen der $\omega$, und zu schnellem Lernen. Bei zu groß gewähltem $\eta$ besteht die Gefahr, daß nur ein lokales Minimum gefunden wird.

### Reihenfolge der Lernmuster

Ein weiterer wichtiger Punkt in bezug auf lokale Minima ist die Art, wie die Lernmuster dem Netzwerk angeboten werden. Grundsätzlich sollten die Lernmuster in *zufälliger Reihenfolge* gelernt werden, insbesondere wenn nur relativ wenige Lernmuster zur Verfügung stehen. Muster verschiedener Klassen sollten gleichverteilt gelernt werden. Existieren beispielsweise zwei Klassen von Mustern mit unterschiedlicher Anzahl, so sollte die zufällige Auswahl eines Musters abwechselnd aus den beiden Klassen erfolgen. Die Notwendigkeit dieser Vorgehensweise zeigt sich dabei unter Umständen erst, wenn das trainierte Netzwerk nicht gelernte Muster

[6]Das Netzwerk soll die Aktivität des Eingangsneurons unverändert auf das Ausgangsneuron abbilden: x = y. Die Schwellen sind in dem Beispiel auf Null gesetzt

klassifiziert. Das bedeutet, daß allein der Fehler $E$ nach dem Lernen nur bedingt eine Aussage darüber zuläßt, wie gut das trainierte Netzwerk auf unbekannte Muster reagiert.

#### Kumulative Adaption

Neben der durch die Gleichungen 5.14 und 5.16 beschriebenen Adaption der Gewichte $\omega$, bei der die Minimierung der Fehlerfunktion in Richtung des steilsten Gradienten bezüglich der summierten Fehlerquadrate *eines Musters* nach jedem Lernschritt erfolgt, können die Gewichte auch erst nach einer bestimmten Zahl von Lernschritten angepaßt werden. Bei der *kumulativen Veränderung* der Gewichte werden die berechneten Gewichtsänderungen *aller* zu lernenden Muster zunächst gespeichert und anschließend in Richtung des steilsten Gradienten der Gesamtfehlerfunktion minimiert. Die Reihenfolge der Muster spielt dann keine Rolle mehr, führt jedoch zu einer Erhöhung des Rechenaufwand [NNW90].

### 5.4.3 Lineare Teilbarkeit

Die $n$ Eingänge eines Neurons $i$ lassen sich in einem Vektor $\vec{x}$ zusammenfassen. Ein bestimmter Eingangsvektor beschreibt einen Punkt im $n$-dimensionalen Raum. Die Ausgangsaktivität des Neurons ist durch die Gleichung 5.1 beschrieben. Alle Punkte $\vec{x}$, die der Gleichung

$$\theta_i + \sum_{j=1}^{n} \omega_{ij} \cdot x_j = 0 \tag{5.18}$$

genügen, liegen auf einer Hyperebene. Die Aktivierungsfunktion $g(a)$ trennt den durch alle möglichen Vektoren $\vec{x}$ aufgespannten Raum in zwei Teilräume, indem sie die beiden Teilräume mit einer Aktivität null oder eins indiziert. Durch die Schwelle $\theta$ wird die Ebene im Raum verschoben. Mit einer unstetigen Aktivierungsfunktion der Form

$$g(a) = \begin{cases} 0 & \text{für } a \leq 0 \\ +1 & \text{für } a > 0 \end{cases} \tag{5.19}$$

ist die Trennung scharf. Mit einer stetigen Aktivierungsfunktion 5.8 ist die Trennung dagegen kontinuierlich, was bedeutet, daß Punkte $\vec{x}$, je nachdem wie nahe sie an der durch Gleichung 5.18 beschriebenen Ebene liegen, und welche Steilheit die Aktivierungsfunktion $g$ aufweist, mit einem Wert $\approx 0,5$ klassifiziert werden. Neben der hochparallelen Informationsverarbeitung des Netzwerks ist diese kontinuierliche Trennung ein Grund für die Toleranz gegenüber von gelernten Mustern abweichenden Mustern bei der Klassifizierung.

#### Aufgabe des Neurons

Ein Neuron mit $n$ Eingängen ist also in der Lage, einen $n$-dimensionalen Raum *linear*, durch eine Hyperebene, in zwei Teilräume zu trennen, und so durch Vektoren repräsentierte Muster zu klassifizieren. Diese Muster müssen jedoch *linear teilbar* sein. Das bedeutet, daß alle Muster einer Klasse sich in einem Teilraum befinden müssen. Generell sind neuronale Netzwerke ohne verdeckte Schichten nur in der Lage, linear teilbare Muster zu klassifizieren. Dieses haben Minskey und Papert im Jahr 1969 gezeigt [MinPa69], mit dem Hinweis, daß sich dieses Problem durch Hinzufügen einer zusätzlichen, verdeckten Neuronenschicht (Hidden-Layer) lösen läßt.

Bis der BPG-Algorithmus im Jahr 1986 von McCelland und Rumelhart vorgestellt wurde [McCeRum86a], gab es keinen allgemeingültigen Algorithmus zum Trainieren mehrschichtiger Feedforward-Netzwerke.

### Das XOR-Problem

Das einfachste Beispiel für ein *nicht linear teilbares* Muster ist das Exklusiv-Oder-Problem, oder XOR-Problem. Das XOR stellt eine logische Funktion in der Boolschen Algebra dar. Es ist eine skalare Funktion von zwei Variablen. Sie genügt der Wahrheitstabelle 5.1

| $x_1$ | $x_2$ | y |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 0 |

Tabelle 5.1:
**XOR**

Ein neuronales Netz, ohne verdeckte Schicht, das diese Funktionsvorschrift ausführen soll, hat zwei Eingangsneuronen $x_1$, $x_2$ und ein Ausgangsneuron $y$. Da es sich um ein zweidimensionales Problem handelt, liegen die möglichen Eingangsmuster als Punkte in einer Ebene. Die Trennung der Teilräume, also hier Halbebenen, erfolgt durch eine Gerade. Diese wird durch die Gleichung

$$\theta = \omega_1 \cdot x_1 + \omega_2 \cdot x_2 \tag{5.20}$$

beschrieben. Um der XOR-Funktion zu genügen, müssen $\omega_1$ und $\omega_2$ so bestimmt werden, daß folgende vier Ungleichungen erfüllt sind

$$\begin{array}{rcrcrcl} 0\cdot\omega_1 & + & 0\cdot\omega_2 & < & \theta & \Rightarrow & 0 < \theta \\ 1\cdot\omega_1 & + & 0\cdot\omega_2 & > & \theta & \Rightarrow & \omega_2 > \theta \\ 0\cdot\omega_1 & + & 1\cdot\omega_2 & > & \theta & \Rightarrow & \omega_1 > \theta \\ 1\cdot\omega_1 & + & 1\cdot\omega_2 & < & \theta & \Rightarrow & \omega_1 + \omega_2 < \theta. \end{array} \tag{5.21}$$

Die letzten drei Ungleichungen von 5.21 stehen im Widerspruch: Es gibt keine zwei reellen Zahlen $\omega_1$ und $\omega_2$, die beide größer als eine reelle Zahl $\theta$ sind, deren Summe aber kleiner als $\theta$ ist

$$\omega_1 + \omega_2 < \theta \wedge \omega_1 > \theta \wedge \omega_2 > \theta. \tag{5.22}$$

Dieser Widerspruch ist der Beweis für die Unlösbarkeit des XOR-Problems mit einem neuronalen Netz ohne verdeckte Schicht. Die graphische Interpretation dieses Widerspruchs ist in Abbildung 5.9 dargestellt. Durch Hinzufügen eines einzigen Hidden-Neurons wird das XOR-Problem lösbar, weil dadurch das in zwei Dimensionen nicht lösbare XOR-Problem in ein in drei Dimensionen lösbares, d.h. linear teilbares, Problem übergeht [McCeRum88].

Im mehrdimensionalen Fall - bei Neuronen mit vielen Eingängen - ist eine so einfache graphische Interpretation dann nicht mehr möglich. Die Frage nach der linearen Teilbarkeit läßt sich dann ebenfalls nicht mehr in so einfacher Weise beantworten.

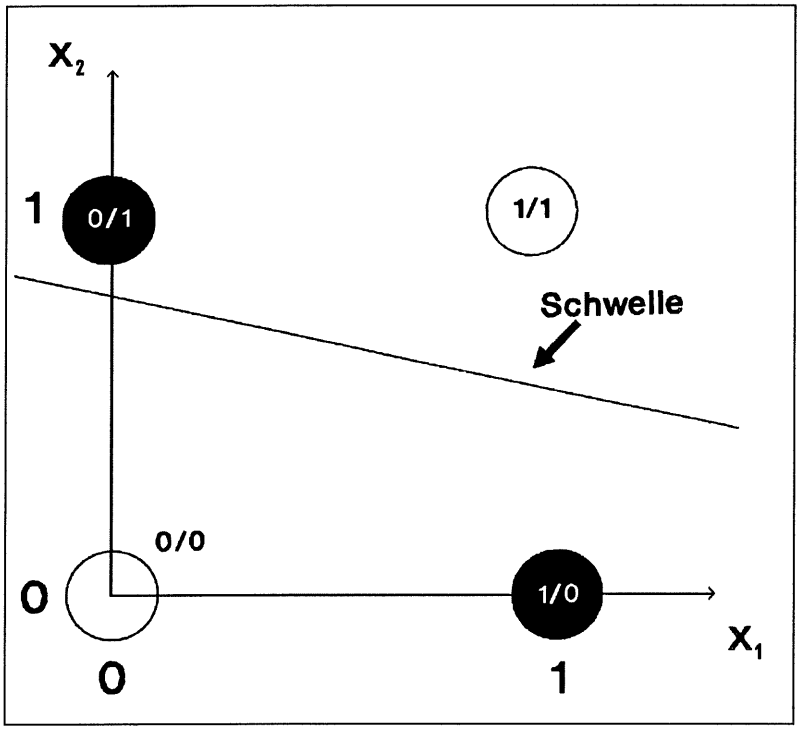


Abbildung 5.9: **Das XOR-Problem** *[NNW90]. Graphische Interpretation: Es ist nicht möglich, eine einzige Gerade so zu legen, daß eine Klassifizierung im Sinne der XOR-Funktion erfolgt.*

### 5.4.4 Bedeutung der Schichten

Um nicht linear teilbare Muster klassifizieren zu können, bedarf es verdeckter Neuronenschichten. Jeder Schicht eines neuronalen Netzwerks kommt eine besondere Bedeutung zu. Wieviele verdeckte Neuronenschichten für eine Aufgabe benötigt werden, hängt prinzipiell von der Topologie der verschiedenen Musterklassen ab. Eine Abbildung $\vec{y} = F(\vec{x})$, die ein neuronales Netzwerk mit *zwei* verdeckten Schichten durchführt, ist zusammen mit der nichtlinearen Aktivierungsfunktionen der Neuronen vergleichbar mit der Entwicklung einer Funktion $F$ in Gaußfunktionen. Fast alle Aufgaben lassen sich daher mit zwei verdeckten Neuronenschichten lösen - meist reicht jedoch eine verdeckte Schicht aus.

#### Bedeutung der Input-, Hidden- und Output-Layer

Ein Neuron ist in der Lage seinen Parameterraum *linear* zu teilen. Durch logische UND-Verknüpfung[7] beliebig vieler Neuronen werden *konvexe* Volumina mit beliebig vielen Hyperflächen beschrieben. Die Beschreibung eines *konkaven* Volumens ist damit jedoch noch nicht möglich.

Erst die logische ODER-Verknüpfung[8], durch die Ausgangsneuronen, der die konvexen Volumina beschreibenden Neuronen der verdeckten Schicht, erlaubt die Beschreibung beliebiger

[7]Die Aktivität des Ausgangs ist dann *eins*, wenn die Aktivitäten *jedes* Eingangs *eins* ist
[8]Die Aktivität des Ausgangs ist dann *eins*, wenn die Aktivität *eines* Eingangs *eins* ist

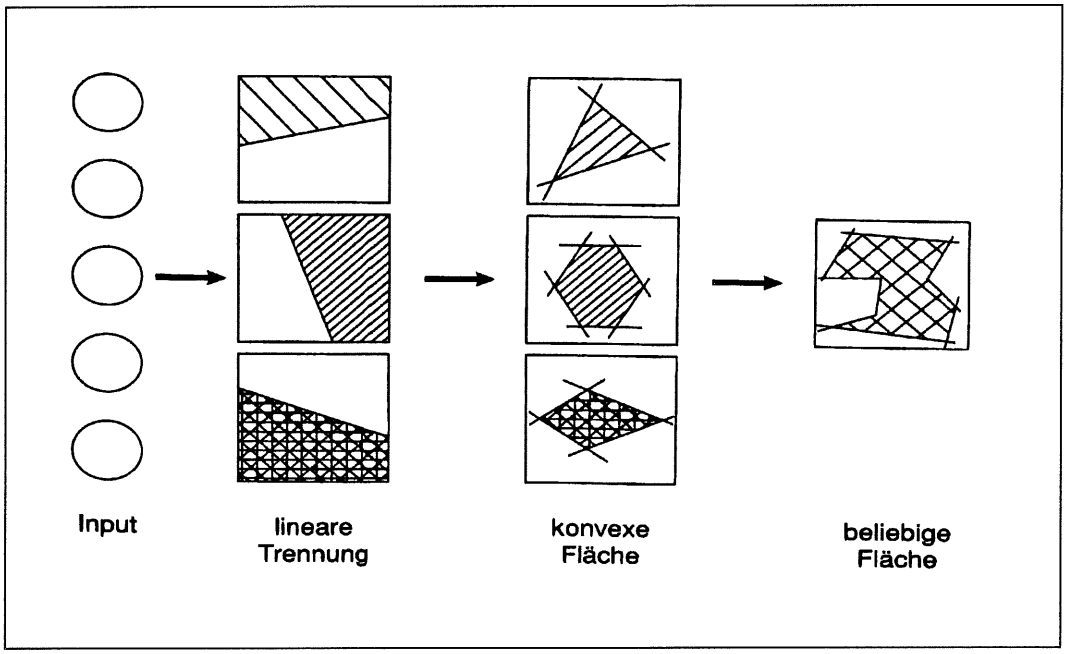


Abbildung 5.10: **Bedeutung der Schichten** *[NNW90]. Jeder Neuronen-Schicht im BPG-Netz kommt, in Bezug auf die Musterklassifizierung, eine bestimmte Bedeutung zu. Erst durch die verdeckte(n) Neuronen-Schicht(en) können komplexe Muster klassifiziert werden.*

Volumina, und somit Musterklassen. Das ist für den zweidimensionalen Fall in Abbildung 5.10 dargestellt, in der Flächen durch Geraden in der beschriebenen Weise definiert werden.

### Zahl der Hidden-Layer

Ein BPG-Netz mit nur einer verdeckten Schicht kann Muster verschiedener Klassen nicht klassifizieren, wenn diese schalenartig einander umschließen, wie z.B. konzentrische Kugeln. Solche Mustertopologien erfordern zusätzliche Neuronenschichten.

### Zahl der Hidden-Neuronen

Für ein BPG-Netzwerk mit einer verdeckten Neuronenschicht ist die Form der zu separierenden Musterverteilungen für die Zahl der benötigten Hidden-Neuronen von Bedeutung, da eine gegebene Verteilung von Eigenschaftsvektoren einer Musterklasse von einem n-dimensionalen Polyeder umschlossen wird. Dieser kann optimal beschrieben werden, wenn die Zahl der Hidden-Neuronen gleich der Zahl der Hyper-Flächen des Polyeders ist. Ob diese Zahl tatsächlich erforderlich ist, hängt von der Orientierung der verschiedenen Musterklassen zueinander ab. Für eine elektronische Implementation ist eine große Zahl von Hidden-Neuronen vorteilhaft, weil dadurch die Redundanz des Netzwerks gegenüber Fehlern der synaptischen Gewichte erhöht wird [Intel90c]. Eine große Zahl von Hidden-Neuronen erfordert jedoch einen hohen Rechenaufwand in der Lernphase. Die Zahl der benötigten Hidden-Neuronen muß in der Regel empirisch bestimmt werden, es sei denn, man kann die Form einer Verteilung analytisch bestimmen, oder bestimmte Symmetrien in den Mustern ausmachen, wie etwa beim Enkoderproblem [McCeRum86a].

### 5.4.5 Klassifizierungsgrenze: Das Bayesian Limit

Die Klassifizierung eines BPG-Netzwerks erfolgt nur bis zu einer theoretisch möglichen Grenze, dem *Bayesian Limit* [DudHa73]. Diese Grenze ergibt sich aus der Überlappung von Mustern verschiedener Klassen, wenn zu klassifizierende Eingangsmuster stark ähnlich sind, aber zu unterschiedlichen Klassen gehören. Zwei durch Vektoren beschriebene Muster werden als ähnlich bezeichnet, wenn sie sich nur in wenigen Komponenten geringfügig unterscheiden, also nahe beieinander liegende Raumpunkte beschreiben. Formal wird die Klassifizierungsgrenze durch die *Bayes'sche Entscheidungstheorie* beschreiben. Die *Bayesische Regel* hat im Fall eindimensionaler Verteilungen mit $n$ Musterklassen die folgende Form

$$P(\kappa_j|x) = \frac{p(x|\kappa_j)P(\kappa_j)}{p(x)} \tag{5.23}$$

wobei

$$p(x) = \sum_{j=1}^{n} p(x|\kappa_j)P(\kappa_j). \tag{5.24}$$

Dabei beschreibt die Gleichung 5.23 die Wahrscheinlichkeit, daß bei zufälliger Vorgabe eines Wertes $x$ ein Muster aus der Klasse $\kappa_j$, von $n$ Klassen, ausgewählt wird. Dabei ist $p(x|\kappa_j)$ die Verteilung (Wahrscheinlichkeitsdichte) der Muster der Klasse $\kappa_j$ als Funktion von $x$ und $P(\kappa_j)$ ist die Wahrscheinlichkeit bei zufälliger Auswahl eines Musters, aus der Gesamtheit aller Muster, eines der Klasse $\kappa_j$ auszuwählen.

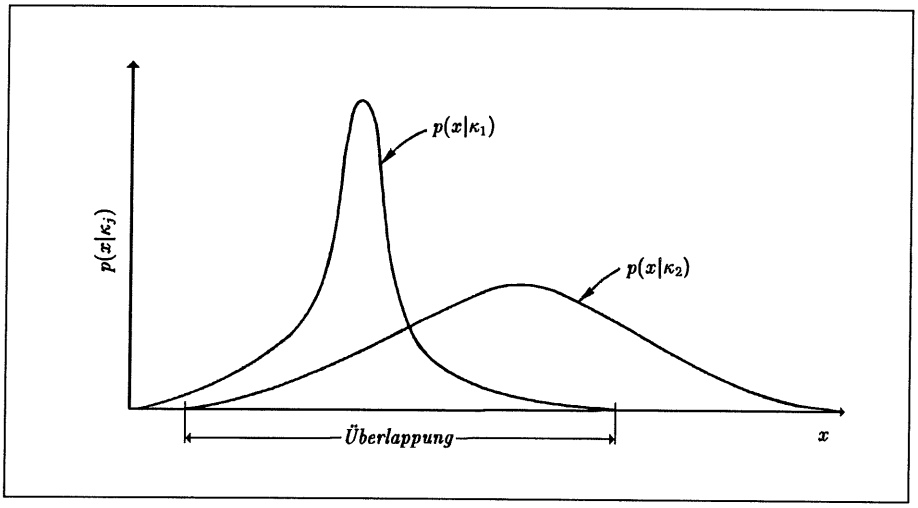


Abbildung 5.11: **Wahrscheinlichkeitsdichten zweier Klassen** *[DudHa73]. Bei der Überlappung von Eigenschaften können Muster x, die in diesen Bereich fallen, nicht eindeutig einer Klasse zugeordnet werden.*

In Abbildung 5.11 sind die Wahrscheinlichkeitsdichten von zwei Klassen von Mustern dargestellt. Die Klassifizierung von Mustern $x$, die in dem überlappenden Bereich liegen, kann nicht fehlerfrei erfolgen. Die Wahrscheinlichkeit, bei der Klassifizierung eines Musters dieses falsch zu klassifizieren, ist in diesem Fall beschrieben durch

$$P(error|x) = \begin{cases} P(\kappa_1|x) & \text{wenn als } \kappa_2 \text{ klassifiziert} \\ P(\kappa_2|x) & \text{wenn als } \kappa_1 \text{ klassifiziert.} \end{cases} \tag{5.25}$$

Die mittlere Fehlerwahrscheinlichkeit ist gegeben durch

$$\begin{aligned} P(error) &= \int_{-\infty}^{+\infty} P(error, x)dx \\ &= \int_{-\infty}^{+\infty} P(error|x)p(x)dx. \end{aligned} \tag{5.26}$$

Diese theoretische Grenze gilt nicht nur für die Klassifizierung von Mustern mit einem neuronalen Netzwerk, sondern auch im Fall analytisch formulierter Regeln. Denn auch dabei erfolgt eine Unterteilung des gesamten Eigenschaftsraumes in Raumgebiete, die den verschiedenen Musterklassen zugeordnet werden.

## 5.5 Klassische Modelle neuronaler Netzwerke

Im Laufe der Jahre wurden verschiedene neuronale Netzwerkmodelle entwickelt, die sich in ihren Eigenschaften und der Anwendbarkeit für bestimmte Aufgabenstellungen unterscheiden. Diese Modelle lassen sich über verschiedene Netzwerkparameter in Klassen einteilen, als da sind: Anzahl der Schichten, Rückkopplung, Art des Lernens.

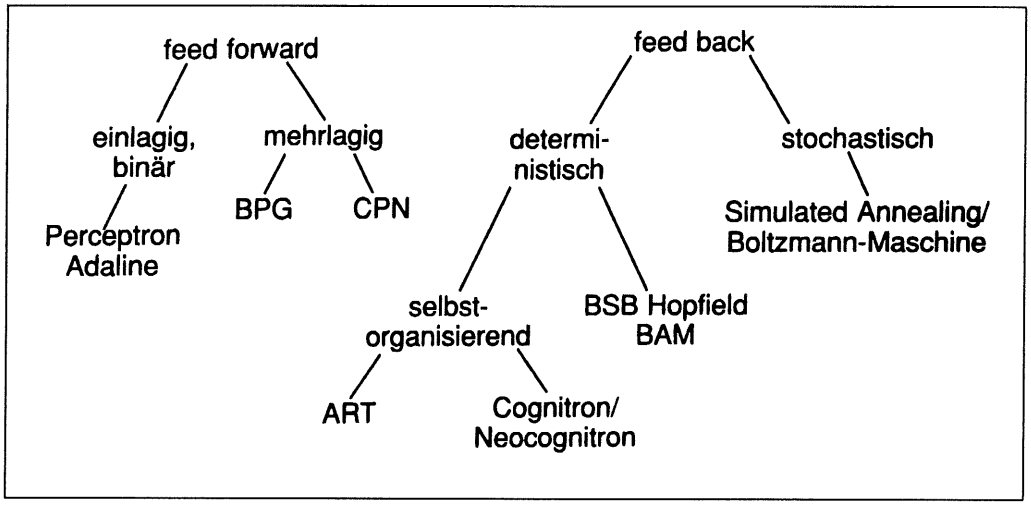


Abbildung 5.12: **Klassen neuronaler Netzwerke** *[NNW90].*

In der Abbildung 5.12 ist eine Einteilung verschiedener Netzwerkmodelle dargestellt, sie ist nur eine Möglichkeit von vielen. Die verschiedenen Netzwerke sind mit unterschiedlichen Motivationen entwickelt worden. Sie sind für bestimmte Aufgaben unterschiedlich gut geeignet. Welcher Netzwerktyp für eine bestimmte Aufgabe am besten geeignet ist, ergibt sich zum einen aus dem Vergleich der gestellten Aufgabe mit bestehenden Lösungen vergleichbarer Aufgaben, zum anderen aber insbesondere aus (Simulations-)Ergebnissen selbst.

## 5.6 Netzwerkmodelle und Anwendungsgebiete

Entscheidend für das Anwendungsfeld eines neuronalen Netzwerks ist die Art des Lernens. Daher unterscheidet man zwischen verschiedenen Arten des Lernens. Psychologisch wird das Lernen als die Modifikation von Verhalten aufgrund von Erfahrung bezeichnet. Das einfachste Modell ist das Reiz-Reaktionsmodell. Gelernt wird dabei, auf einen bestimmten Reiz mit einer bestimmten Reaktion zu antworten. Neuronale Netzwerke finden in diesem Sinne Anwendung auf bestimmte Probleme. So können unbekannte Muster anhand von Mustern, die in der Lernphase gelernt wurden, klassifiziert werden, wie Schriftzeichen, ja sogar Handschrift [Fuku83], veränderte oder verrauschte Muster gelernten Mustern zugeordnet werden, oder gelernte Reaktionen für Regelungsaufgaben eingesetzt werden, wobei das betreffende Netzwerk in der Lage ist, bei abweichender Regelungsituation hinzuzulernen und sich dabei angemessen zu verhalten. Beispiele dafür sind etwa das Balancieren eines Stabes [Bart83], [NNW90] oder die visomotorische Steuerung eines Roboters [Schul91]. Ein anderes Anwendungsgebiet sind Optimierungsaufgaben, die mit dem klassischen Problem des Handlungsreisenden[9] verwandt sind. Da die Spurerkennung anhand von Spursegmenten diesem Problem verwandt ist, kann sie mit geeigneten neuronalen Netzwerken erfolgen [Den89], [DenLinn90], [LUTP90-6].

### 5.6.1 Informationsfluß

Das Hauptkriterium zur Einteilung der verfügbaren Modelle Neuronaler Netzwerke ist der Informationsfluß im Netzwerk. Im biologischen System ist die klare Trennung zwischen *feed forward* und *feed back* nicht möglich, da die Topologie des biologischen Neuronalen Netzwerks beide Aspekte vereint. Im Fall künstlicher neuronaler Netzwerke besteht diese Trennung aufgrund der stark unterschiedlichen mathematischen Beschreibungsweisen, insbesondere in Bezug auf die Lernalgorithmen.

#### Feedforward

Rückkopplungfreie Netzwerke eignen sich für Aufgaben aus den Bereichen der Mustererkennung, Klassifizierung und Assoziativspeicher. Sie werden in der Lernphase für eine bestimmte Aufgabe mit korrelierten Ein- und Ausgabe-Lernmustern trainiert.

#### Feedback

Rückgekoppelte Netzwerke eignen sich zur Rekonstruktion verrauschter Muster als auch für Optimierungsaufgaben. Es wird eine Energiefunktion definiert, die es erlaubt, die Güte einer gefundenen Lösung zu bewerten. Das betreffende Netzwerk minimiert diese Energiefunktion. Dabei erfolgt die Minimierung im Sinne eines thermodynamischen Systems, das bestrebt ist, den Zustand geringster Energie anzunehmen.

### 5.6.2 Modelle

Es gibt neben dem im Abschnitt 5.4 ausführlich vorgestellten Backpropagation-Netzwerk noch zahlreiche andere Netzwerkmodelle, die jeweils für bestimmte Aufgabenstellungen prädestiniert sind. Für Probleme der Musterklassifizierung ist das BPG-Netzwerk bestens geeignet. Zudem sind integrierte Schaltungen verfügbar, welche das BPG-Netzwerk nachbilden. Außerdem ist

[9] *engl. travelling salesman problem (TSP)*

das BPG-Netzwerk besonders für zeitkritische Anwendungen gut geeignet, da es sich um ein nichtrückgekoppeltes Netzwerk handelt. Dieser Umstand stellt eine wichtige Randbedingung bei der Auswahl des Netzwerktyps dar.

Im folgenden werden die Modelle neuronaler Netzwerke kurz vorgestellt, die für die Gesamtheit aller neuronalen Netzwerke einen repräsentativen Charakter aufweisen.

### Perceptron

Im Jahre 1958 wurde das Perceptron-Netzwerk von F. Rosenblatt vorgestellt [Ros58]. Das Perceptron stellt ein *feed forward* Netz mit binären Schwellwertelementen ohne verdeckte Neuronen-Schicht(en) dar, und kann somit nur linear teilbare Muster klassifizieren, was eine gravierende Einschränkung darstellt.

### Adaline und Madaline

Das Adaline (**ada**ptive **l**inear **ne**uron) stellt ebenfalls ein einschichtiges *feed forward* Netzwerk dar. Die Neuronen können in Abhängigkeit von der gewichteten Summe der Eingänge die Werte $-1$ oder $+1$ annehmen. Dieses Netzwerk wurde im Jahr 1960 von B. Widrow und E. Hoff vorgestellt [NNW90]. Sie führten anstatt variabler Schwellen den *Bias* ein, siehe Abschnitt 5.2.1. Zudem stellten sie die Delta-Lernregel vor, die gegenüber der beim Perceptron verwendeten Lernregel eine höhere Lerngeschwindigkeit erzielt. Diese Lernregel verwendet zur Korrektur der synaptischen Gewichte ein Fehlersignal. Der Backpropagation-Algorithmus stellt eine verfeinerte Form dieser Regel dar.

Madaline (**m**ultiple **ada**ptive **l**inear *n*euron) stellt ein um eine verdeckte Schicht erweitertes Adaline-Netz dar, kann aber auch nur linear teilbare Muster klassifizieren. Im Gegensatz zum Adaline-Netzwerk kann es mehr Muster klassifizieren, als es Eingangsneuronen hat, jedoch nicht beliebig viele.

### Das Hopfield-Modell

Das im Jahr 1982 von Hopfield vorgestellt Netzwerk ist das erste rückgekoppelte Netzwerk [Hop82]. Aufgrund der von Hopfield gemachten Konstruktion genügt es derselben Dynamik wie Spingläser und erschließt so die damit verbundenen mathematisch-physikalischen Hilfsmittel. Das Hopfield-Netz besteht aus einer einzigen Neuronenschicht. Die Neuronen dieser stellen gleichsam Atome dar, die über ihre Spins untereinander wechselwirken; sie sind vollständig und symmetrisch miteinander vernetzt. Hopfield-Netze lassen sich für Optimierungsaufgaben und zur Rekonstruktion verrauschter Muster einsetzen.

### Boltzmann-Maschine

Im Jahr 1985 wurde das als Boltzmann-Maschine bezeichnete Netz von Ackley, Hinton und Sejnowski vorgestellt [Ack85]. Die Boltzmann-Maschine stellt eine Weiterentwicklung des Hopfield - Netzes dar. Ein globales Minimum der Energiefunktion wird mit Hilfe des sogenannten *simulated annealing* [10] gefunden, was eine Analogie zu thermischem Rauschen darstellt, wodurch lokale Minima der Energiefunktion überwunden werden können.

[10] *engl. simulated annealing* $\approx$ simulierte Kristallisation

## 5.7 Elektronische Realisierung eines neuronalen Netzes

Es gibt integrierte Schaltkreise (Neuro-Chips), welche die für den Aufbau eines neuronalen Netzwerks notwendigen Funktionen zur Verfügung stellen. Dabei beschränkt sich die Nachbildung neuronaler Netzwerke meist auf den Ausführungsmodus. Es werden aber auch schon Überlegungen angestellt, den Lernmodus mit einzubeziehen [MfNN90]. Die Signalverarbeitung erfolgt digital, analog oder optisch.

### 5.7.1 Anforderungen an einen Neuro-Chip

Zur Implementation eines BPG-Netzwerks mit elektronischen Mitteln sind verschiedene Voraussetzungen zu erfüllen. Diese ergeben sich aus der in Abschnitt 5.4 dargestellten mathematischen Beschreibung des neuronalen Netzwerks.

#### Matrixmultiplikation

Wesentlich für die Abbildung eines Vektors auf den Ausgangsvektor der Folgeschicht ist die Matrixmultiplikation dieses Vektors mit den synaptischen Gewichtsfaktoren $\omega$, Gleichung 5.11, Abschnitt 5.4. Die Multiplikationen müssen mit hinreichender Genauigkeit durchgeführt werden.

#### Aktivierungsfunktion

Neben der Matrixmultiplikation muß eine Aktivierungsfunktion entsprechend Gleichung 5.2, Abschnitt 5.2.1 nachgebildet werden, wie bei der Simulation des neuronalen Netzwerks verwendet.

#### Speicherung der synaptischen Gewichte

Die während der Lernphase bestimmten synaptischen Gewichte $\omega$ müssen mit hinreichender Genauigkeit gespeichert werden. Bei zu geringer Genauigkeit weicht das Verhalten des elektronischen neuronalen Netzwerks zu stark von dem simulierten ab.

#### Kaskadierbarkeit

Bei der Implementation eines neuronalen Netzwerks mit sehr vielen Neuronen muß auf industriell gefertigte hochintegrierte Schaltkreise zurückgegriffen werden, da nur so der getriebene Aufwand vertretbar bleibt. Ist die Anzahl der in einem Neuro-Chip verfügbaren Neuronen für eine angestrebte Anwendung zu gering, so muß die Möglichkeit der Kaskadierung mehrerer Chips bestehen.

### 5.7.2 Einige verfügbare Neuro-Chips

Bei den meisten Neuro-Chips handelt es sich um Labormuster, die somit nicht ohne weiteres für praktische Anwendungen zur Verfügung stehen. So ist es auch schwierig, Datenblätter zu Neuro-Chips zu bekommen. Zu den im folgenden aufgeführten Chips liegen Datenblätter [Intel90a] und [Intel90b], [Oxf90] vor oder zumindest Artikel aus Fachzeitschriften [ct12/89], [ct5/91].

| Chip | *cps* | Firma | $\omega$ [bits] | Arbeitsweise |
|---|---|---|---|---|
| IPRMM | $25M$ | Oxford Computer Inc. | 8 | digital |
| NBS | $55M$ | Micro Devices | 16 | digital |
| ETANN | $2G$ | Intel | 6-7 | analog |
| (GaAs) | $50G$ | Mitsubishi | 8 | optisch |

Tabelle 5.2: **Verfügbare Neuro-Chips**

Die Verarbeitungsgeschwindigkeit von Neuro-Chips wird in Einheiten *cps*[11] angegeben. Das ist die Anzahl der für die Matrixmultiplikation notwendigen skalaren Multiplikationen zusammen mit der anschließenden Aufsummierung der skalaren Produkte. Für die Beurteilung, ob die Informationsverarbeitung eines vorliegenden Neuro-Chips für eine bestimmte Aufgabe schnell genug erfolgt, ist jedoch nur die Zeit, die benötigt wird, um auf eine Eingabe mit einer Ausgabe zu reagieren, von Bedeutung. Bei seriell arbeitenden Neuro-Chips wie dem NBS oder dem IPRMM ist diese Zeit mit der *cps*-Rate direkt korreliert. Bei parallel arbeitenden Neuro-Chips hängt sie jedoch davon ab, ob alle Neuronen an der Informationsverarbeitung beteiligt sind.

### Der ETANN-Chip

Die Anforderungen, die sich im Zusammenhang mit dem Z-Kammer-Trigger am H1-Detektor ergeben, werden von dem ETANN-Chip[12] [Intel90a], [Intel90b] erfüllt. Insbesondere, da die Informationsverarbeitung analog erfolgt, was von Vorteil ist, weil die Z-Vertex-Histogrammdaten des Z-Kammer-Triggers ebenfalls in analoger Form vorliegen.

Die Eingangsaktivitäten müssen beim ETANN-Chip in Form von (differentiellen) elektrischen Spannungen vorliegen. Sie werden mit gespeicherten Spannungen, den synaptischen Gewichten, multipliziert. Das Ergebnis sind (differentielle) Ströme, die entlang von, als Dendriten bezeichneten, Leiterbahnen aufsummiert werden. Diese werden nichtlinearen Verstärkern zugeführt, die eine sigmoide Charakteristik aufweisen und eine (differentielle) Ausgangsspannung liefern. Wesentliches Element der Synapsen des ETANN-Chips ist der *Gilbertmultiplizierer*. Dieser ist in der Lage, eine vorzeichenbehaftete Eingangsspannung $\Delta V_{in}$ mit einem (differentiellen) synaptischen Gewicht $\Delta V_{weight}$ gemäß der Gleichung

$$\Delta I_{out} = \Delta V_{in} \cdot \Delta V_{weight} \tag{5.27}$$

zu multiplizieren. Die Speicherung der synaptischen Gewichte erfolgt analog in EEPROM-Zellen[13]. Das ist das besondere Element der Synapsen des ETANN-Chips, welches die nichtflüchtige Speicherung der synaptischen Gewichte in komfortabler Weise erlaubt. Zwar finden die diesen sehr ähnlichen EPROM-Zellen schon seit geraumer Zeit in digitalen Speicherbausteinen Anwendung [Intel86], doch die Verwendung als analoger Speicher ist neu. Im Gegensatz zu EPROM-Speichern [14] können die synaptischen Gewichte des ETANN-Chips nach erfolgter Programmierung wieder *elektrisch* gelöscht werden.

Die Genauigkeit, mit der die synaptischen Gewichte gespeichert werden, wird im Datenblatt mit 6...7 Bit angegeben. Es zeigt sich in der Simulation, daß sich das Verhalten des

[11] *cps = engl. connections per second* $\approx$ Verbindungen pro Sekunde

[12] ETANN = *Electrically Trainable Analog Neural Network* $\approx$ elektrisch trainierbares analoges neuronales Netzwerk

[13] *EEPROM = Electrically Erasable Programmable Read Only Memory*

[14] *EPROM = Erasable Programmable Read Only Memory*

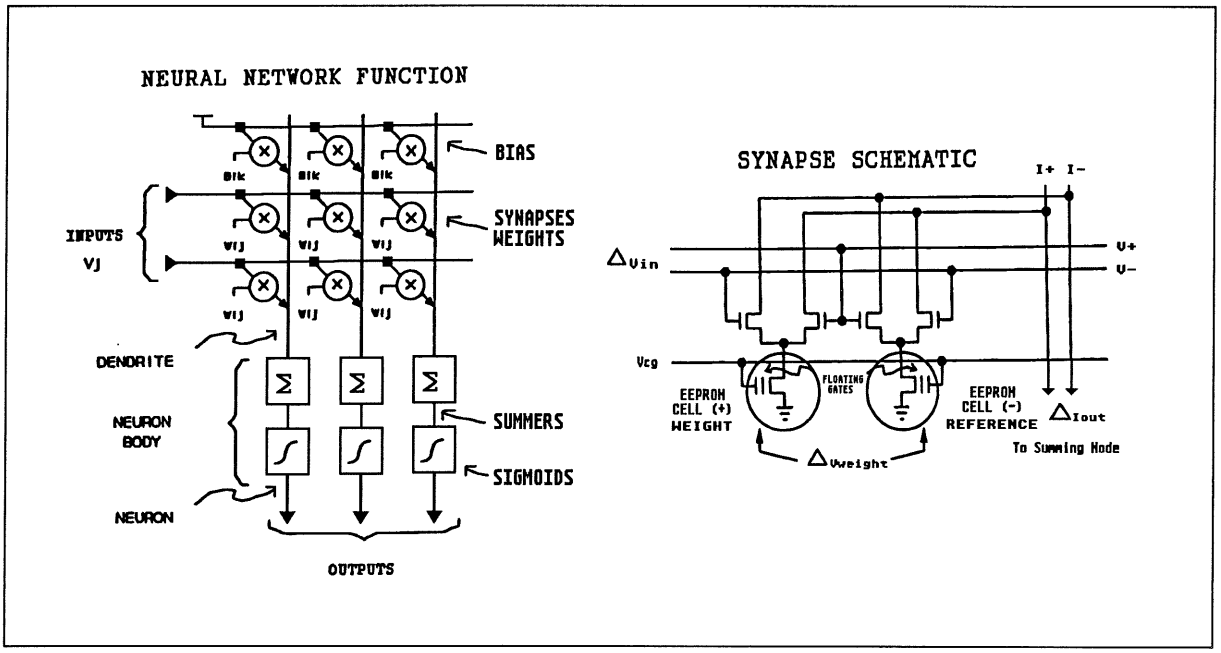


Abbildung 5.13: **ETANN-Chip** *[Intel90a]. Links ist das Funktionsschema des Chips als Ganzes dargestellt. Deutlich sichtbar ist dabei die Anordnung der Synapsen in einer Matrix, welche zur Matrixmultiplikation der Eingangsspannungen mit den in den Synapsen gespeicherten Gewichten dient. Rechts ist das Schema für eine Synapse dargestellt. Die Speicherung der synaptischen Gewichte erfolgt in analoger Form als elektrische Ladung. Nichtlineare Verstärker bilden die neuronale Aktivierungsfunktion nach.*

neuronalen Netzwerks für den Z-Kammer-Trigger erst unterhalb einer Auflösung 5...6 Bit signifikant bezüglich der Trigger-Eigenschaften verändert. Der ETANN-Chip ist kaskadierbar, was aufgrund der vielen Signale notwendig ist. Die Antwortzeit wird im Datenblatt mit $5\mu s$ für den 80170NW [Intel90a] und mit $0,5\mu s$ für den Nachfolgetyp 80170NX [Intel90b] angegeben, so daß ein Einsatz als $2^{nd}$-Level-Trigger möglich ist.

## 5.8 Neuronale Netze kontra von Neumann-Architektur

Neuronale Netze unterscheiden sich sowohl in der Architektur, als auch in ihren Eigenschaften und Fähigkeiten deutlich gegenüber der von-Neumann-Architektur konventioneller Computer. Daraus resultieren unterschiedliche Eignungen für bestimmte Aufgaben. Diese stark unterschiedlichen Eignungsgrade werden beim Vergleich zwischen Gehirn und dem konventionellen Computer, bzw. konventioneller digitaler oder analoger Schaltungstechnik, besonders deutlich, unabhängig davon, daß das Gehirn eine wesentlich höhere Komplexität aufweist. Die Frage, was die Unterschiede von Gehirn und Computer ausmacht, hat unter anderem von Neumann selbst beschäftigt [Neu58].

### 5.8.1 Die von-Neumann-Architektur

Die von-Neumann-Architektur liegt prinzipiell jedem konventionellen Computer zugrunde. Das Attribut konventionell trägt dabei dem Umstand Rechnung, daß erweiterte oder andere Archi-

tekturen entwickelt wurden, die von der konventionellen Struktur abweichen. Als Beispiele sind in diesem Zusammenhang Vektorrechner, Parallelrechner genannt, die unter Umständen aus vielen tausend, parallel an einer Aufgabe arbeitender, (von-Neumann-)Prozessoren aufgebaut sind, sogenannte Pinboard-Maschinen[15], die eine besondere Ausführung von Parallelrechnern darstellen, sowie auch Neuro-Computer, die auf neuronalen Netzen aufbauen.

### Prozessor

Die von-Neumann-Architektur weist die in Abbildung 5.14 dargestellte Blockstruktur auf. Es gibt einen Prozessor, der sich in Steuerwerk, bzw. Leitwerk, und Rechenwerk aufgeteilt. Er kommuniziert über Leitungsbündel, die funktional in Adressbus, Datenbus und Steuerbus aufgeteilt werden, sowohl mit dem Hauptspeicher, den sich Programme und Operanden teilen, sowie mit Eingabe- und Ausgabeeinheiten als Schnittstellen zur Außenwelt. Daneben existieren noch Peripheriegeräte, wie etwa externe Speichermedien, die für die prinzipielle Funktionsweise jedoch nicht von Bedeutung sind.

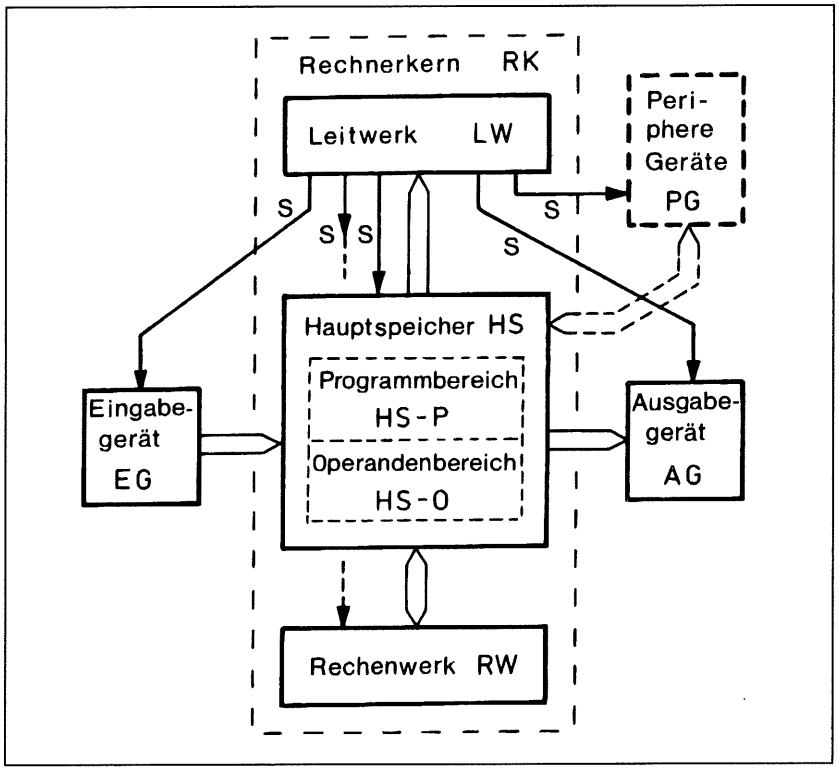


Abbildung 5.14: **Von-Neumann-Architektur** *[Lag87]. Die dargestellte Blockstruktur liegt prinzipiell jedem konventionellen Computer zugrunde, sie unterscheidet sich deutlich von der des neuronalen Netzwerks.*

[15] *engl. pinboard* ≈ *schwarzes Brett* (mit Aufgaben, die zu erledigen sind)

### Busse

Über den Adressbus adressiert der Prozessor eine bestimmte Speicherstelle, um über den Datenbus einen Befehl oder Operanden zu lesen oder zu schreiben. Die Wahl zwischen Lesen und Schreiben, sowie die Steuerung des zeitlichen Ablaufes, erfolgt über den Steuerbus. Das Steuerwerk übernimmt die Ablaufsteuerung bezüglich der Kommunikation mit dem Hauptspeicher sowie die Interpretation von Befehlen und der sich daraus ergebenden Steuerung des Rechenwerks, das für die Veränderung der Operanden zuständig ist. Die zeitliche Signalabfolge zur Ansteuerung der einzelnen Prozessorelemente erfolgt synchron und wird aus einem Taktsignal gewonnen.

### Erweiterte Architektur

Die Busse stellen gewissermaßen ein Nadelöhr für die rasche Informationsverarbeitung dar. In der Regel kann nur ein Befehl pro Zeiteinheit über den Bus übertragen werden. Die Abarbeitung eines Befehls benötigt meist mehrere Taktzyklen. Zur Beschleunigung der Befehlsabarbeitung sind im Steuerwerk mancher Prozessoren sogenannte Pipelines[16] implementiert, die bewirken, daß ein neuer Befehl schon bearbeitet werden kann, bevor ein sich gerade in der Ausführung befindlicher Befehl abgeschlossen ist. Vektorrechner verfügen zudem über mehrere Rechenwerke, welche die gleichzeitige Bearbeitung mehrerer Operanden erlauben, was sich besonders bei Vektoroperationen beschleunigend auswirkt.

## 5.8.2 Das neuronale Netz

Das neuronale Netz baut sich aus vielen, jedoch sehr einfachen, informationsverarbeitenden Elementen - den Neuronen - auf, die auch als die Prozessorelemente des neuronalen Netzwerks bezeichnet werden. Die Neuronen verfügen nicht über einen Befehlssatz, der die Informationsverarbeitung durch eine Folge von Befehlen, d.h. ein Programm, explizit vorgibt. Vielmehr erfolgt die Infomationsverarbeitung durch Abbildung eines Eingangsvektors über die Neuronenschichten auf einen Ausgangsvektor, wobei die Verarbeitung der Neuronen durch deren Propagierungs- und Aktivierungsfunktion bestimmt ist. Die Vorschrift, wie ein Eingangsvektor auf einen Ausgangsvektor abzubilden ist, wird in der Lernphase iterativ bestimmt.

### Neuronen

Die Neuronen können vollständig verbunden sein, d.h. jedes Neuron einer Neuronenschicht ist mit jedem Neuron der folgenden Schicht verbunden. Die Verbindungen zwischen den Neuronen können sowohl unidirektional als auch bidirektional sein. Zudem können Rückkopplungswege zwischen Teilnetzen bestehen.

Die einfache Art der Informationsverarbeitung der einzelnen Neuronen läßt sich gut sowohl mit digitalen, als auch mit analogen Schaltelementen nachbilden. Problematisch ist dabei jedoch die hochgradige Vernetzung der Neuronen, so daß größere Netze die Integration auf einem Chip erfordern. Diesbezüglich werden schon zahlreiche Anstrengungen unternommen, wie man [MfNN90] entnehmen kann.

[16] *engl. pipeline* ≈ Rohrleitung

### 5.8.3 Der Vergleich

Der wesentliche Unterschied im Vergleich mit dem neuronalen Netz besteht darin, daß ein von-Neumann-Computer einen Prozessor mit einem mehr oder weniger umfangreichen Befehlssatz hat, und daß die Abarbeitung der Befehle grundsätzlich sequentiell erfolgt. Die Interpretation und Ausführung des komplexen Befehlssatzes erfordert eine aufwendige (digitale) Logik. So benötigt man zur Implementation eines derartigen Prozessors einige Tausend bis zu einigen Millionen elementare Logikelemente.

| Eigenschaft | neuronales Netz | von Neumann |
|---|---|---|
| Informationsspeicherung | verteilt | lokalisiert |
| Informationsverarbeitung | verteilt | lokalisiert |
| Parallelisierbarkeit | vollständig | möglich |
| Systemredundanz | hoch | gering |
| Datentoleranz | hoch | gering |
| Abstraktionsfähigkeit | hoch | gering |
| Programmierung | implizit | explizit |
| Verarbeitungstransparenz | gering | hoch |
| Rechengenauigkeit | gering | hoch |
| Verarbeitung | asynchron | synchron |

Tabelle 5.3: **Vergleich der wesentlichen Eigenschaften**

Demgegenüber ist bei neuronalen Netzen die Speicherung von Information auf das gesamte Netzwerk verteilt, in Form der synaptischen Gewichte. Auch die eine Informationsverarbeitung bestimmende Abbildungsvorschrift liegt nicht lokalisierbar, d.h. in Form von Programmbefehlen in bestimmten Speicherstellen, sondern verteilt über das gesamte Netzwerk, repräsentiert durch die synaptischen Gewichte, vor. Die unterschiedlichen Eigenschaften neuronaler Netze und der von-Neumann-Architektur sind in Tabelle 5.3 gegenübergestellt.

#### Eigenschaften

Das neuronale Netzwerk verarbeitet Information, wie auch der von-Neumann-Computer, indem es in Abhängigkeit von der Eingabeinformation eine Ausgabeinformation liefert; nur ist die Informationsverarbeitung nicht durch explizit formulierte Algorithmen vorgegeben, sondern wird implizit, iterativ während der Lernphase bestimmt. Das ist eine der Stärken neuronaler Netzwerke, wenn es nicht oder nur schwer möglich ist, eine Problembeschreibung in analytischer Weise zu formulieren.

#### Eignung

Aufgrund der unterschiedlichen Eigenschaften sind neuronale Netzwerke und der konventionelle Computer für verschiedene Aufgaben unterschiedlich gut geeignet. In Tabelle 5.4 sind die unterschiedlichen Eignungen gegenübergestellt.

Neuronale Netzwerke und von-Neumann-Rechner verhalten sich bezüglich der Informationsverarbeitung und Eignung für bestimmte Aufgaben komplementär. Zudem unterscheiden sich die verschiedenen Netzwerkmodelle selbst auch noch untereinander in Bezug auf die Eignung für bestimmte Aufgaben. Aufgrund der einfachen Operation des einzelnen Neurons lassen sich neuronale Netzwerke mit vielen Neuronen, wie sie für komplexe Aufgaben erforderlich sind, gut mit

| Aufgabe | neuronales Netz | von Neumann |
|---|---|---|
| Musterklassifizierung | gut | schwierig |
| Musterrekonstruktion | gut | schwierig |
| Assoziativspeicher | gut | wenig |
| Entscheidung | gut | wenig |
| Optimierung | gut | zeitkritisch |
| Regelung | problemabhängig | |
| Exakte Datenverarbeitung | schlecht | gut |
| Massendatenspeicherung | schlecht | gut |

Tabelle 5.4: **Gegenüberstellung der Aufgabeneignung**

elektronischen Mitteln realisieren. Dagegen lassen sich komplizierte Algorithmen, die auf einem Computer entwickelt werden, nur mit großem Aufwand in Form von elektronischen Schaltungen realisieren. Zudem ist es im Fall eines neuronalen Netzwerks leicht möglich, Veränderungen vorzunehmen, wenn sich durch erneutes Lernen mit anderen Mustern Verbesserungen erzielen lassen. Ein verbesserter Algorithmus läßt sich unter Umständen nur schwer in einer bestehenden Elektronik implementieren. Im ungünstigsten Fall erfordert er die Neukonstruktion der Elektronik.

# Kapitel 6

# Simulation

Der Signalfluß, der sich bei Anwendung eines neuronalen Netzes zur Analyse der verschiedenen Signale des Z-Triggers ergibt, wurde unter Berücksichtigung spezifischer Eigenschaften des neuronalen Netzes in der Simulation durch entsprechenden Datenfluß nachgebildet. Anhand von geeigneten Daten wurde das Verhalten des Gesamtsystems untersucht.

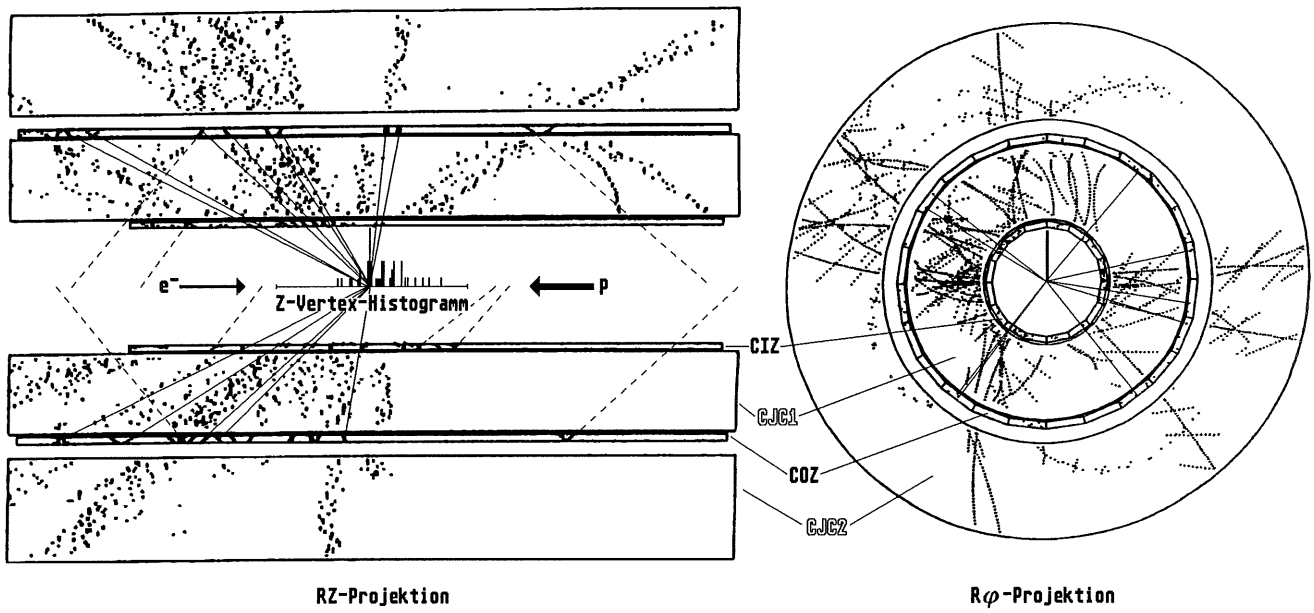


Abbildung 6.1: **Z-Vertex-Histogramm** *Es ist ein simuliertes $b\bar{b}$-Ereignis bei z=0 zusammen mit dem vom konventionellen Teil des Triggers erzeugten Z-Vertex-Histogramm dargestellt (Datensatz: HERA02.I0010004.PART01, RUN: 1, EVENT: 2). Nachträglich wurden einige Spuren und Spursegmente skizziert. Durchgezogene Linien stehen dabei für Spuren, die durch Koinzidenz zwischen CIZ und COZ der Vertex-Region zuzuordnen sind, unterbrochene Linien für Spursegmente in CIZ oder COZ, die einem Bereich außerhalb der Vertex-Region zuzuordnen sind. Um Fehler durch zufällige Koinzidenzen zu reduzieren, werden alle in den Driftzellen von CIZ und COZ gefundenen Primärspuren in Vertex- und Untergrundspuren unterteilt und gezählt. Diese Information steht in Form der so genannten Zell-Histogramme zur Verfügung (s. Abb. 6.2). Einige Gesamthistogramme sind in den Abbildungen 6.3, 6.4, 6.5, 6.6 dargestellt.*

## 6.1 Monte-Carlo-Daten

Für den Z-Trigger ist die Physik der zu klassifizierenden Ereignisse nur insofern von Bedeutung, als daß die Reaktion in der Vertexregion stattfinden muß, und daß dabei entstehende (geladenen) Zerfallsprodukte in den sensitiven Bereich der Z-Kammer fallen müssen. Bei allen physikalisch interessante Reaktionen werden viele Teilchen in den Bereich der Z-Kammern emittiert. Bei den Simulationen wurden $b\bar{b}$-Ereignisse stellvertretend für physikalisch relevante Ereignisse verwendet, da diese beim Triggern wegen der kleinen im Kalorimeter deponierten Energie besonders auf einen Spur-Trigger angewiesen sind.

Untergrundereignisse können vom Z-Trigger - anhand des Z-Vertex-Histogramms und der Zell-Histogramme - nur dann von physikalisch interessanten Ereignissen unterschieden werden, wenn der überwiegende Teil der Vertices Bereichen außerhalb der Wechselwirkungszone zuzuordnen ist. So können Untergrundereignisse die von Strahl-Restgas-Wechselwirkungen herrühren, und vorzugsweise aus der Wechselwirkungszone kommen, nicht von physikalisch interessanten Ereignissen unterscheiden werden. Daher ist deren Unterdrückung eher Aufgabe von Energie-Triggern oder Spur-Triggern mit einer Impulsschwelle. Um der Güte des Z-Triggers bezüglich seiner topologischen Qualitäten zu untersuchen, wurden daher stellvertretend für Untergrundereignisse solche aus Strahl-Wand-Wechselwirkungen verwendet.

### 6.1.1 Primärdaten

Die zur Verfügung stehenden Monte-Carlo-Daten (Primärdaten) - Spurkoordinaten in Driftkammern - wurden dem neuronalen Netzwerk bei der Simulation *nicht direkt* zugeführt. Denn bei einer möglichen zukünftigen Implementation eines neuronalen Netzwerks zur Separation von Untergrund und Physik wird die Klassifizierung der Ereignisse anhand der Histogramme (Sekundärdaten) des Z-Triggers erfolgen. Die Bestimmung der Spursegmente und Vertices (Tracking) aus den Signalen der Driftkammern wird ausschließlich Aufgabe der konventionellen Trigger-Elektronik sein. Das Tracking ist auch mit neuronalen Netzwerken möglich, wobei dazu neben BPG-Netzen auch Hopfield-Netze zum Einsatz kommen [DenLess90], [Stim90].

Doch auch bei der Anwendung eines neuronalen Netzwerks auf die komplexen Histogramme des Z-Triggers wird die Abstraktionsfähigkeit neuronaler Netzwerke sowie die vergleichsweise einfache Realisierung deutlich: Es bedarf keiner ausgeklügelten Algorithmen zur Signalananlyse - allein die Verfügbarkeit repräsentativer Lerndaten genügt, um das neuronale Netzwerk für die vorgesehene Aufgabe zu "programmieren".

#### Datenformat

Die Primärdaten lagen in Form von BOS-Bänken [BOS87] vor, die ein flexibles Datenformat zur Speicherung von relevanten Informationen eines Ereignisses aufweisen, wie Energie, Impuls, Teilchen-Code [Phy0488] und Spurkoordinaten, aber auch Daten zur Geometrie einzelner Detektorkomponenten. Die Spurkoordinaten erlauben zusammen mit den geometrischen Daten die Berechnung der Reaktion einzelner Detektorkomponenten, im Fall des Z-Triggers das Verhalten der Signaldrähte der Z-Kammer.

### Verarbeitung

Zur Simulation des konventionellen Teils der Z-Trigger-Logik konnte auf bestehende Programme zurückgegriffen werden[1]. Die mit diesen Programmen erzeugten Histogramme standen so für die Simulationen des neuronalen Netzwerks zur Verfügung. Die Aufgabe bestand in der Bildung der eigentlichen Trigger-Bedingung anhand dieser Histogramme.

Da zum Zeitpunkt der Arbeit auch schon analytisch formulierte Lösungsansätze zur Auswertung der Z-Trigger-Histogramme vorlagen (Abschnitt 4.2.3), konnten diese mit den Simulationen des neuronalen Netzwerks verglichen werden.

## 6.2 Z-Trigger-Signale

Der konventionelle Teil des Z-Triggers erzeugt aus den Primärsignalen der Z-Kammer des H1-Detektors verschiedene Signale, die in Gruppen zusammengefaßt als Histogramme interpretiert werden können. Jedem Ereignis, das in der Z-Kammer seine Spuren hinterläßt, wird so eine Gruppe von Histogrammen zugeordnet. Erst die Auswertung dieser Histogramme durch Teile der Z-Trigger-Logik liefert das eigentliche Trigger-Signal, das dem H1-Zentral-Trigger zur Verfügung gestellt wird, um zusammen mit den Trigger-Signalen der anderen Trigger-Systeme des H1-Detektors gegebenenfalls die Auslese eines Ereignisses zur dauerhaften Speicherung durch den IBM-Großrechner des DESY zu veranlassen.

Zur Auswertung der vom Z-Trigger gelieferten Histogramme wurde in der Simulation ein BPG-Netz eingesetzt, mit einem Ausgangsneuron, einer verdeckten Schicht sowie einer Anzahl von Eingangs-Neuronen, die der Gesamtzahl der verwendeten Histogramm-Bins entsprach. Die Bedeutung der Histogramme sowie ihr Zustandekommen, wird in den folgenden Abschnitten dargestellt.

### 6.2.1 Signalübersicht

Die vom Z-Trigger erzeugten Histogramme weisen unterschiedliche Zahlen von Bins auf, wie in Tabelle 6.1 dargestellt. Bei den durchgeführten Simulationen mit dem BPG-Netzwerk wurde wahlweise ein BPG-Netz mit 96 (nur Z-Vertex-Histogramm) oder 174 Eingangsneuronen (Z-Vertex- und Zell-Histogramme) verwendet. Die Abbildung 6.2 verdeutlicht die Erzeugung der verschiedenen Histogramme aus den in CIZ und COZ gefundenen Spursegmenten.

| Histogramm | Bins |
|---|---|
| Z-Vertex-Histogramm (bei $\approx 20\ MHz$ ) | 96 |
| CIZ-Zell-Histogramm *vertex* (VtxCIZ) | 15 |
| CIZ-Zell-Histogramm *background* (BgrCIZ) | 15 |
| COZ-Zell-Histogramm *vertex* (VtxCOZ) | 24 |
| COZ-Zell-Histogramm *background* (BgrCOZ) | 24 |
| $\mathrm{Bin}_{gesamt} : 96_{ZVtx} + 78_{Zell} =$ | 174 |

Tabelle 6.1: **Übersicht: Z-Trigger-Histogramme**

[1]Autor: Behrend, H.-J., DESY 1991

### Z-Vertex-Histogramm

Koinzidenzen zwischen Spursegmenten von CIZ und COZ erzeugen Einträge im *Z-Vertex-Histogramm.* Jeder Eintrag steht für eine gefundene Spur. Die Auflösung der Spursegmente wird durch die Taktfrequenz der den Signaldrähten der Driftkammern zugeordneten Schieberegister bestimmt. Bei dem zur Simulation des Z-Triggers verwendeten Programm war eine Taktfrequenz von $\approx 20\ MHz$ vorgegeben, womit die Auflösung $\approx 2,4\ mm$ betrug. Diese bestimmt ihrerseits die Auflösung der Z-Vertex-Histogramm-Bins mit $\Delta z_{Bin} \approx 5mm$. Für das Z-Vertex-Histogramm sind daher 96 Bins definiert.

### Zell-Histogramme

Eine Koinzidenz zwischen einem in CIZ und COZ gefundenen Spursegment führt zusätzlich zum Eintrag im Z-Vertex-Histogramm zu je einem Eintrag im einer Driftkammer zugeordneten *Vertex-Zell-Histogramm.* Zeigt ein gefundenes Spursegment auf einen Ort außerhalb der Vertex-Region, so erfolgt ein Eintrag im *Untergrund-Zell-Histogramm* der betreffenden Driftkammer (s. Abb. 6.2). Die Zahl der Zell-Histogramm-Bins einer Driftkammer ist gleich der Zahl der Driftzellen von CIZ (15) und COZ (24), da jedes Bin genau einer Driftzelle zugeordnet ist. Manche Ereignisse niedriger Multiplizität lassen sich nur bei Bewertung des Z-Vertex- zusammen mit den Zell-Histogrammen richtig klassifizieren. Bei Ereignissen hoher Multiplizität wirkt die ergänzende Auswertung der Zell-Histogramme zufälligen Koinzidenzen entgegen.

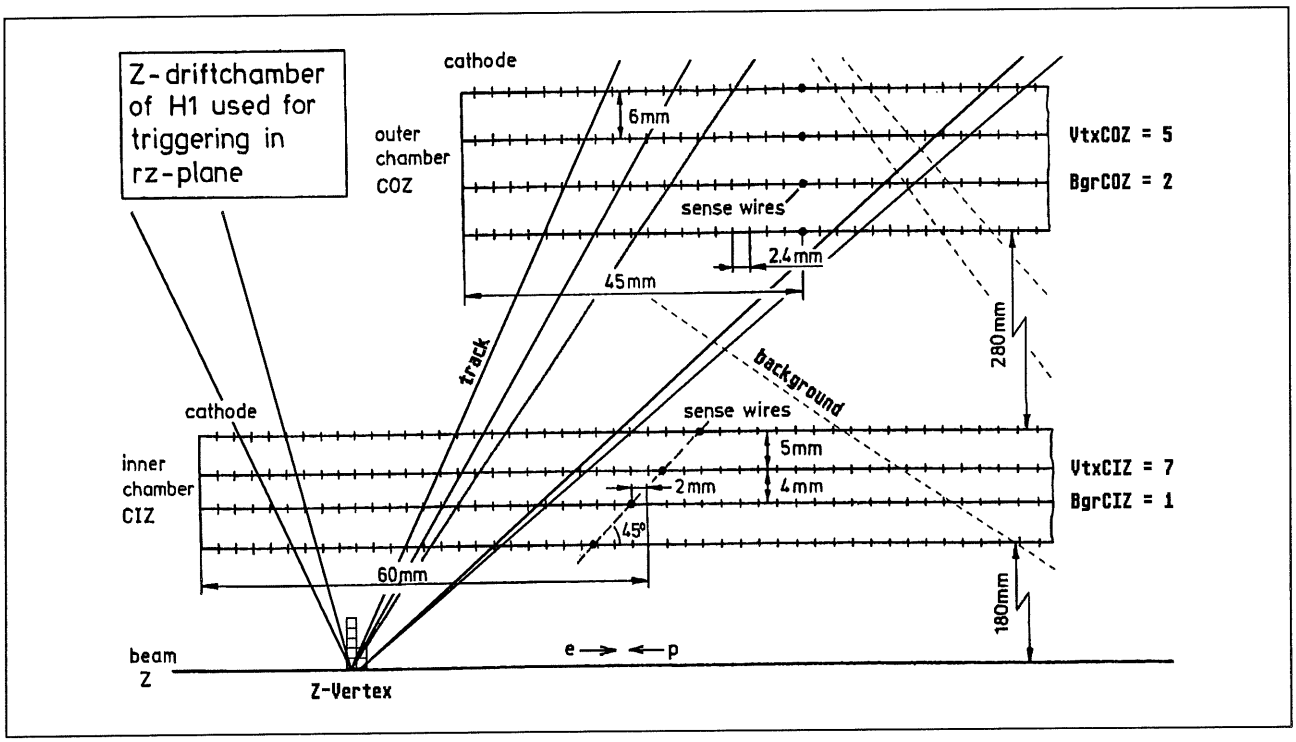


Abbildung 6.2: **Histogramme des Z-Triggers** *[Beh91]. Koinzidenzen zwischen Spursegmenten in CIZ und COZ, die der Vertex-Region zuzuordnen sind (track), führen zu Einträgen im Z-Vertex-Histogramm (Z-Vertex) und in den Vertex-Zell-Histogrammen (VtxCIZ und VtxCOZ) der Driftkammern. Spursegmente, die nicht der Vertex-Region zuzuordnen sind (background) bewirken Einträge in den Untergrund-Zell-Histogrammen (BgrCIZ und BgrCOZ).*

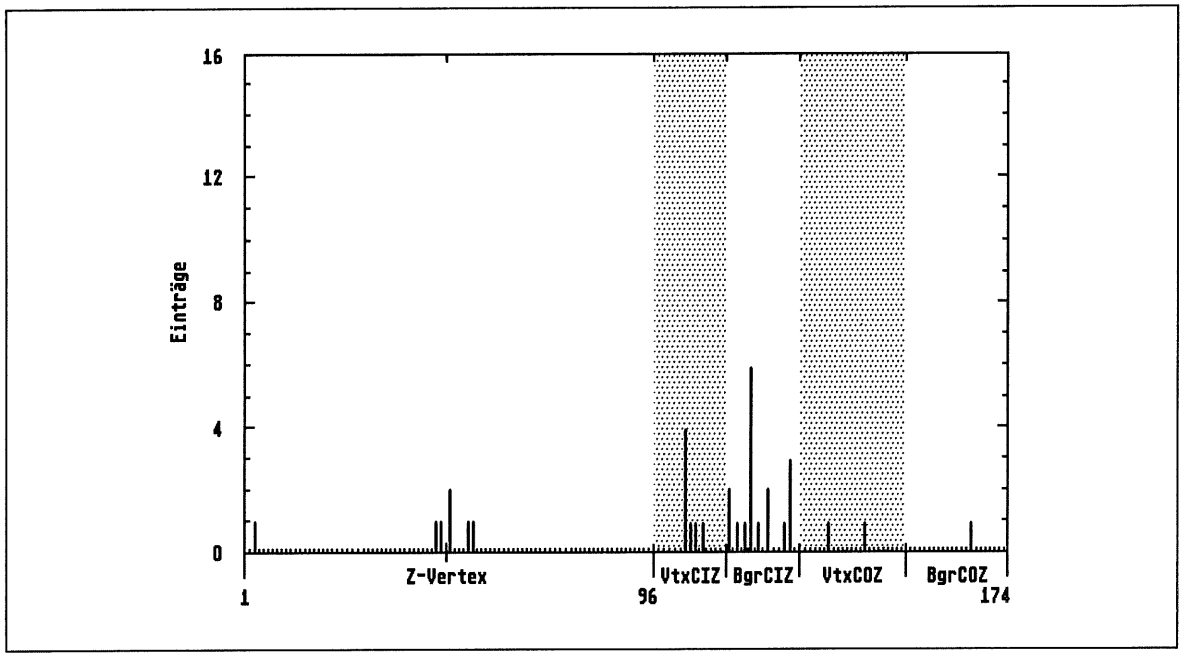


Abbildung 6.3: **Histogramme eines $b\bar{b}$-Ereignisses niedriger Multiplizität.** *Es wurde vom neuronalen Netz richtig mit $y \approx 0,7$ klassifiziert. Die richtige Zuordnung durch ein algorithmisches Verfahren erfordert die genaue (und aufwendige) Analyse des Z-Vertex-Histogramms zusammen mit den Zell-Histogrammen, um nicht mit Untergrund niedriger Multiplizität verwechselt zu werden (z.B. Abb. 6.5).*

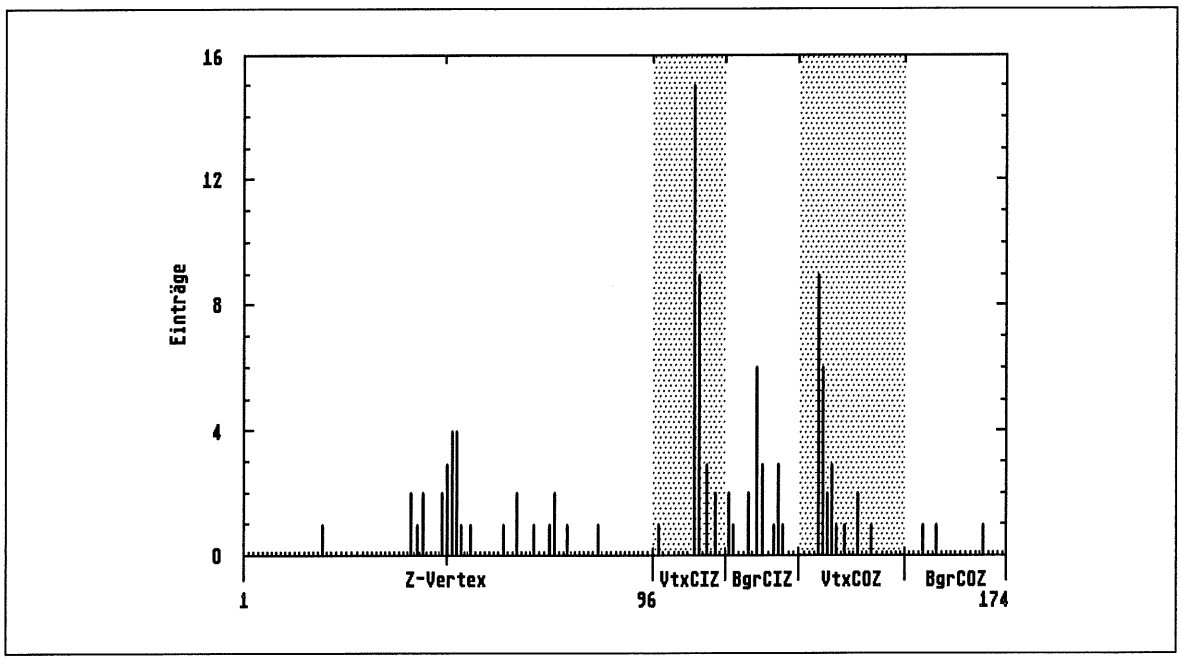


Abbildung 6.4: **Histogramme eines $b\bar{b}$-Ereignisses hoher Multiplizität.** *Dieses wurde vom neuronalen Netzwerk ebenfalls richtig mit $y \approx 0,7$ klassifiziert, also nicht mit einem ähnlichen Untergrundereignis (z.B. Abb. 6.6) verwechselt.*

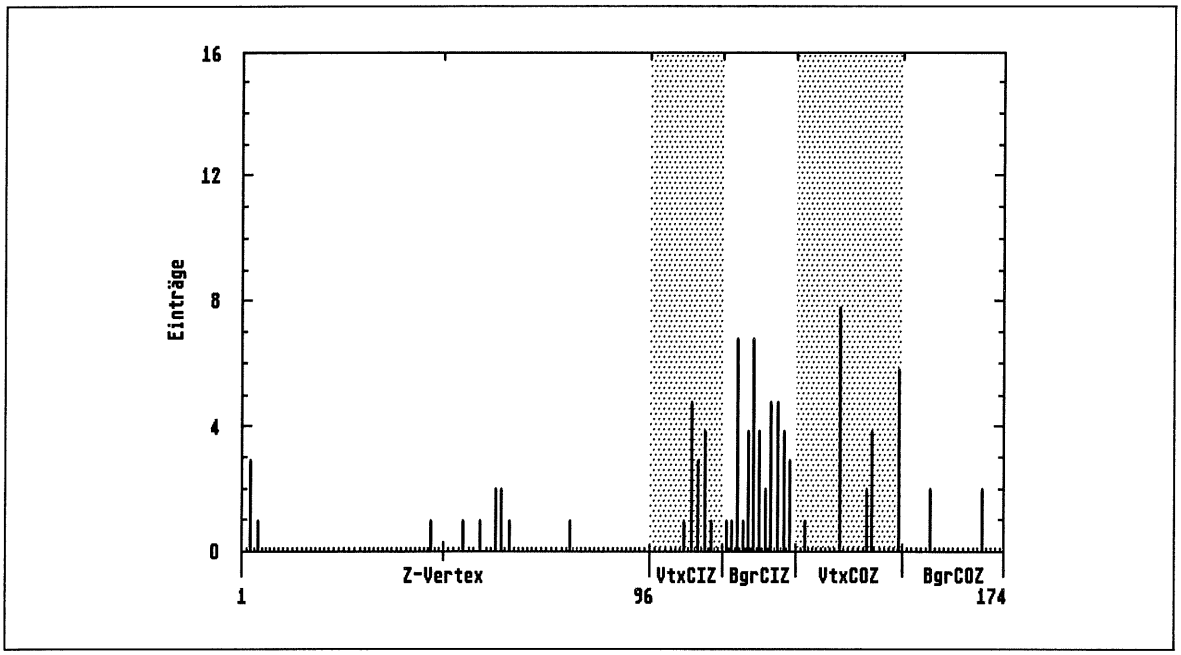


Abbildung 6.5: **Histogramme eines Strahl-Wand-Ereignisses niedriger Multiplizität.** *Es wurde mit $y \approx 0,3$ richtig klassifiziert. Ein neuronales Netzwerk ist in der Lage, komplexe Klassifizierungsregeln zu repräsentieren.*

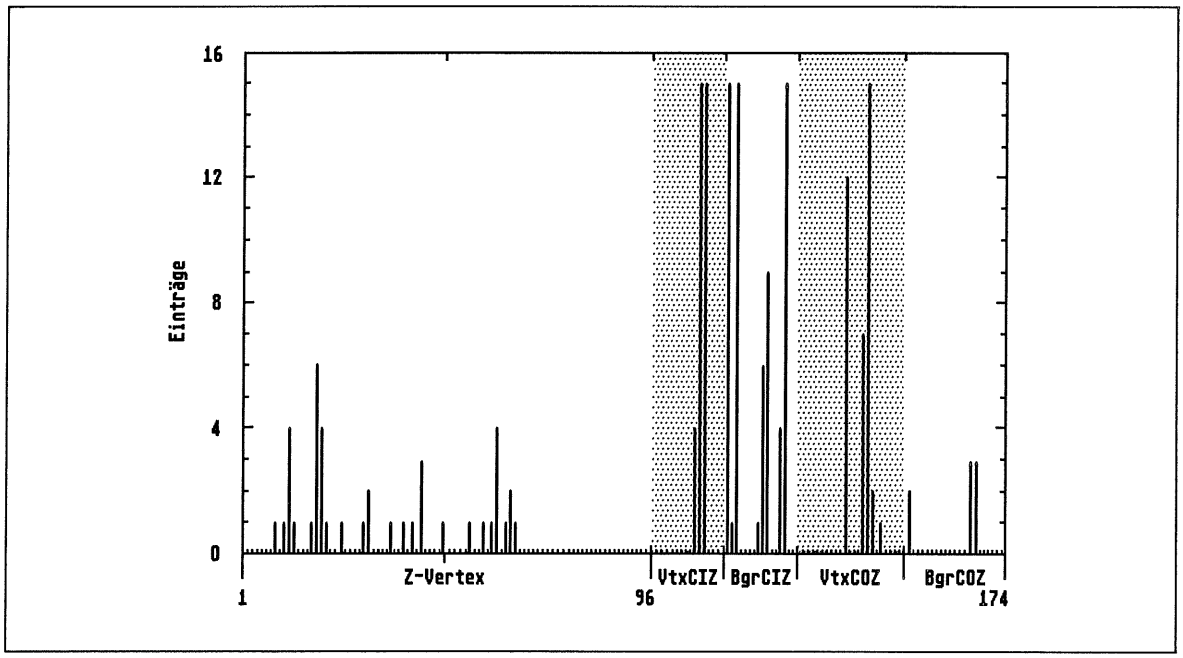


Abbildung 6.6: **Histogramme eines Strahl-Wand-Ereignisses hoher Multiplizität.** *Auch dieses Ereignis wurde richtig mit $y \approx 0,3$ klassifiziert. Abgesehen von der Schwierigkeit, eine analytische Regel für die Klassifizierung zu formulieren, ist es unter Umständen sehr aufwendig eine solche Klassifizierungsregel durch geeignete (schnelle) Elektronik nachzubilden.*

### 6.2.2 Sekundärdaten

Die vom Simulationsprogramm des Z-Triggers erzeugten Z-Vertex- und Zell-Histogramme eines Ereignisses stellen zusammengefaßt einen Spaltenvektor $\vec{x}^j$ (= transponierter Zeilenvektor ${}^t\vec{x}_j$) dar, der folgende Form aufweist

$$\vec{x}^j = {}^t( \underbrace{\underbrace{x_1,\ldots,x_{96}}_{\text{ZVertex}}}_{\text{Z-Vertex-Histogramm}} , \underbrace{\underbrace{x_{97},\ldots,x_{111}}_{\text{VtxCIZ}}, \underbrace{x_{112},\ldots,x_{126}}_{\text{BgrCIZ}}, \underbrace{x_{127},\ldots,x_{150}}_{\text{VtxCOZ}}, \underbrace{x_{151},\ldots,x_{174}}_{\text{BgrCOZ}}}_{\text{Zell-Histogramme}} ). \quad (6.1)$$

Diese Spaltenvektoren[2] dienten als Eingangsvektoren für das neuronale (BPG)-Netz. Bei den Simulationen wurde entweder nur das Z-Vertex-Histogramm als Eingangsvektor verwendet

$$\vec{x}^{96} = {}^t(x_1,\ldots,x_{96}), \quad (6.2)$$

oder zusammen mit den Zell-Histogrammen

$$\vec{x}^{174} = {}^t(x_1,\ldots,x_{174}). \quad (6.3)$$

Dem entsprechend wurde das neuronale Netzwerk mit 96, bzw. 174 Eingangs-Neuronen definiert. Beim Vergleich der Simulationsergebnisse wurde die Bedeutung der Zell-Histogramme für die Untergrundunterdrückung, als auch für die Separation von Ereignissen niedriger Multiplizität deutlich.

#### Bedeutung für neuronales Netzwerk

Für das neuronale Netzwerk bedeutet die Hinzunahme der Zell-Histogramm-Information zum Z-Vertex-Histogramm gemäß der Bayes'schen Entscheidungstheorie, daß durch diese Maßnahme die Verteilungen der unterschiedlichen Klassen weniger überlappen, so daß dadurch die Fehlerwahrscheinlichkeit bei der Klassifizierung reduziert wird (Abschnitt 5.4.5). Das wird besonders bei der Klassifizierung von Zwei-Spur-Ereignissen deutlich. Die Reduzierung der Fehlerwahrscheinlichkeit erfolgt jedoch nur dann, wenn die zusätzliche Information nicht redundant ist. Das in der Lernphase des neuronalen Netzwerks erzielte Klassifizierungsergebnis kann als untere Schranke für die Überlappung der verschiedenen Musterklassen angesehen werden.
Auch ein konventioneller Ansatz unterliegt dieser theoretischen Klassifizierungsgrenze, denn eine analytisch formulierte Trigger-Bedingung bewirkt auch nur die Separation des gesamten n-dimensionalen Eigenschaftsraumes in, den unterschiedlichen Klassen zugeordnete, Teilräume.

## 6.3 Datenfluß

Die Simulationen des neuronalen Netzwerks, wie auch die Simulationen des konventionellen Teils des Z-Triggers, wurden auf der IBM-Großrechenanlage des DESY durchgeführt. Der dem zugrunde liegende Datenfluß ist in Abbildung 6.7 skizziert. Die Simulationen lieferten als Ergebnis neben den durch das neuronale Netzwerk erreichten Klassifizierungen auch die in der Lernphase bestimmten synaptischen Gewichte. Im Falle einer Implementation werden diese synaptischen Gewichte in die betreffende neuronale Elektronik geladen.

[2] In Abschnitt 5.4 wurden (willkürlich) Spaltenvektoren zur formalen Beschreibung des BPG-Netzwerks anstelle von Zeilenvektoren gewählt

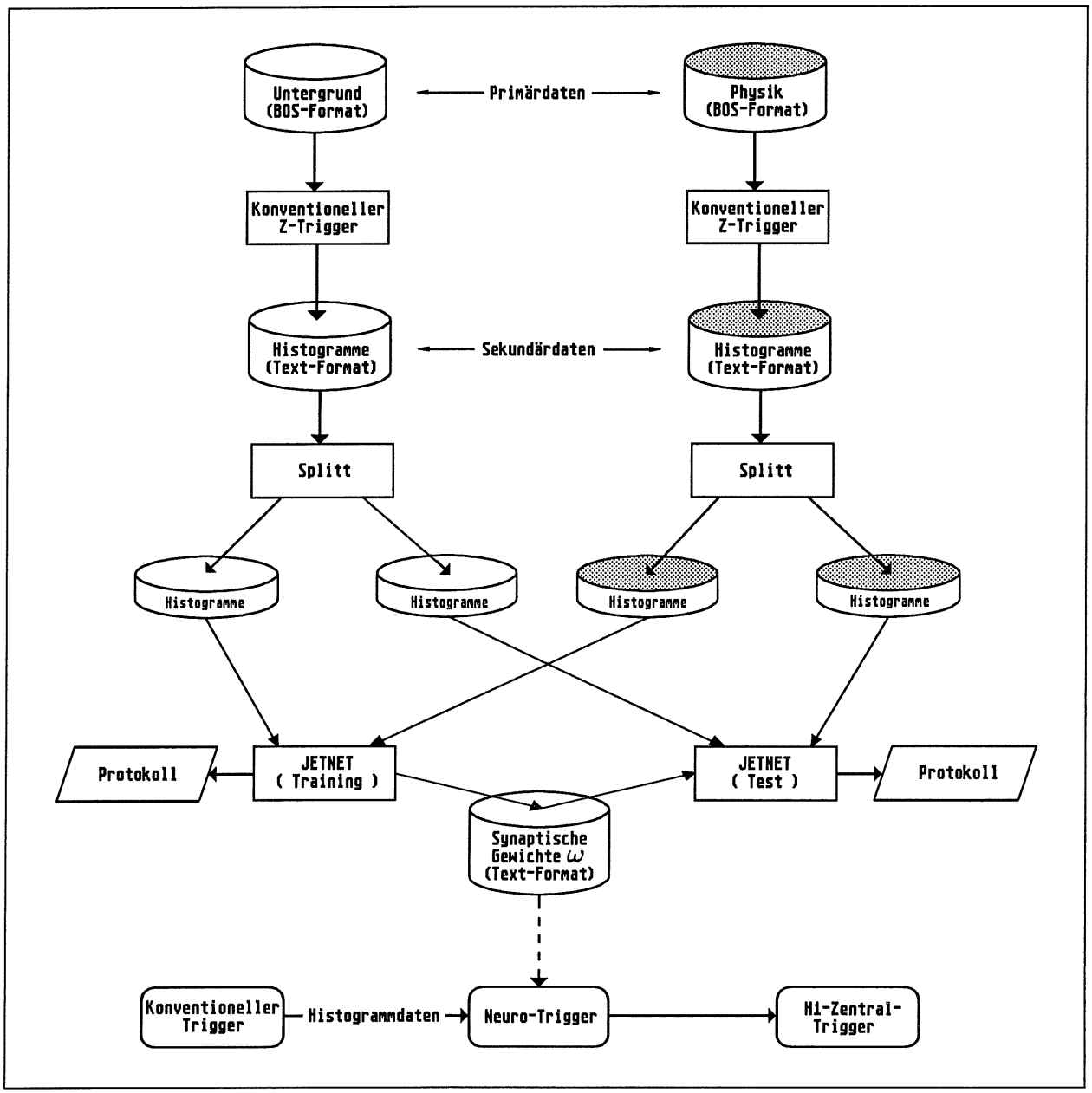


Abbildung 6.7: **Datenfluß** *Für Simulationen standen bei DESY auf der IBM-Großrechenanlage Monte-Carlo-Daten für Untergrund und Physik in Form von BOS-Bänken zur Verfügung. Diese Daten wurden jedoch nicht direkt verwendet, sondern es wurden daraus mit einem bestehenden Programm, welches den konventionellen Teil des Triggers nachbildet, Histogramme generiert. Diese wurden in Lern- und Test-Daten aufgeteilt, und für die Simulationen mit dem neuronalen Netzwerks mit dem Programmpaket* JETNET *[LUTP90-6] verwendet. Die in der Lernphase (Training) berechneten synaptischen Gewichte $\omega$ wurden in einem zweiten Schritt bei der Klassifizierung nicht gelernter Ereignisse (Test) verwendet. Zwecks Portabilität wurden die $\omega$ zusammen mit den topologischen Netzwerkparametern in Text-Dateien geschrieben, um so auf anderen Rechnern (Macintosh, IBM-PC, Atari ST, ...) in einfacher Weise verfügbar zu sein, wenn solche zur Steuerung eingesetzt werden.*

### 6.3.1 Primärdatenverarbeitung

Die Primärdaten (BOS-Format) für Untergrund und Physik wurden nacheinander dem Programm, welches den konventionellen Teil des Z-Triggers simuliert, als Eingabe zugeführt. Dieses liefert zu jedem Ereignis, sofern es Spuren in CIZ und COZ hinterläßt, die verschiedenen Histogramme. Das Programm wurde so modifiziert, daß die Histogramme zusammen mit den Identifikationsnummern der Ereignisse (Run #, Event #) auf Text-Dateien[3], die Sekundärdaten, geschrieben wurden.

Das Text-Format wurde gewählt, um diese Daten in einfacher Weise für $\mu$-Computer verfügbar zu machen; denn erste Versuche zur Separation von Untergrund und Physik durch neuronale Netzwerke wurden mit dem Programm NNSIMU, welches dem Buch [NNW90] beiliegt, durchgeführt. Da dieses Programm nicht im Quelltext vorlag und nur auf einem IBM-PC läuft, wurde auf einem Macintosh-Computer[4] ein kleines C-Programm geschrieben, mit dem die Histogramm-Daten (Text-Datei) vom IBM-Großrechner in das für NNSIMU erforderliche Format (Text-Datei) konvertiert werden konnten.

#### Lern- und Test-Datensätze

Die Klassifizierungsergebnisse während der Lernphase läßt nur eine Aussage darüber zu, bis zu welchem Grad die vorliegenden Z-Trigger-Signale prinzipiell die Separation von Untergrund und Physik durch das neuronale Netzwerk erlauben.

Erst die Klassifizierung von nicht gelernten Mustern erlaubt die Beurteilung der praktischen Einsetzbarkeit des neuronalen Netzes. Daher wurden die Sekundärdatensätze von Untergrund und Physik (Histogramm-Datensätze) jeweils in etwa zwei gleich große Teildatensätze aufgeteilt (Splitt). Da die Daten in Form von Text-Dateien vorlagen, konnte dies in einfacher Weise mit dem auf der IBM-Großrechenanlage verfügbaren Editor[5] erfolgen.

#### Lernphase

Die Lerndaten wurden JETNET [LUTP90-6] in der Lernphase zugeführt, das als Ergebnis die synaptischen Gewichte $\omega$ lieferte. Parallel zur Lernphase wurden Protokolle erzeugt, um das Verhalten des neuronalen Netzwerks in der Lernphase beurteilen zu können. Neben dem numerischen Endergebnis wurden das Klassifizierungsverhalten in Abhängigkeit vom durchgeführten Lernzyklus graphisch mit dem Programm Paket GEP [GepV4.7] erfaßt.

#### Testphase

Die nicht zum Lernen verwendeten Ereignisse wurden, auf Grundlage der in der in der Lernphase bestimmten synaptischen Gewichte $\omega$, in der Testphase dem neuronalen Netzwerk zur Klassifizierung angeboten. Das Ergebnis wurde dabei numerisch und graphisch protokolliert. Die graphische Darstellung der Klassifizierung, in Form eines Histogramms, erlaubt die qualitative Beurteilung der Ergebnisse.

---

[3] $\approx$ DISP=(NEW,CATLG,CATLG), DCB=R01DCB.CARD, SPACE=(TRK,(10,5),RLSE), UNIT=FAST

[4] Das Z-Trigger-Test-Programm ICCP wurde auch auf einem Macintosh-Computer in C geschrieben

[5] IBM ISPF/PDF editor

### 6.3.2 Datensätze

Als Primärdaten wurden, stellvertretend für Untergrund und Physik, Monte-Carlo-Ereignisse aus verschiedenen auf der IBM-Großrechenanlage bei DESY verfügbaren Datensätzen ausgewählt, die in Tabelle 6.2 zusammengefaßt sind.

| **Primärdaten (Spuren, BOS-Format)** | | |
|---|---|---|
| Datensatzname | Inhalt | Ereignisse |
| HERA02.H1SIM.I0000003.PART01<br>HERA02.H1SIM.I0000003.PART02 | 25 $Hz/bw$ | 3795 |
| HERA02.H1SIM.I0010004.PART01 | $b\bar{b}$, $z = 0cm$ | 1000 |
| F36HJB.SIM160.VTEX.SMEAR.BBB.RUN01 | $b\bar{b}$, $z = \pm\ 25cm$ | |
| F36HJB.SIM160.VTEX.SMEAR.PAIRS.RUN01 | $l\bar{l}$, $z = \pm\ 25cm$ | |

| **Sekundärdaten (Histogramme, Text-Format)** | | |
|---|---|---|
| H1KLAR.BW.IDNR.ALL | 25 $Hz/bw$ | 361 |
| H1KLAR.BBBAR.IDNR.ALL | $b\bar{b}$, $z = 0\ cm$ | 882 |
| H1KLAR.SMEAR.BBBAR.XHISTO.ALL | $b\bar{b}$, $z = \pm\ 25cm$ | 498 |
| H1KLAR.SMEAR.PAIRS.XHISTO.ALL | $l\bar{l}$, $z = \pm\ 25cm$ | 617 |

Tabelle 6.2: **Datensatzübersicht**

Ereignisse, die Strahl-Wand-Wechselwirkungen zugeordnet werden, sind mit $bw$[6] bezeichnet. Die Zahl der falsch klassifizierten Untergrund-Ereignisse ist einer datensatzspezifischen Konstante proportional, und erlaubt so die quantitative Beurteilung der Untergrundunterdrückung. Die Abkürzung $b\bar{b}$ steht für Ereignisse aus $b\bar{b}$-Reaktionen, die Abkürzung $l\bar{l}$ für Lepton-Antilepton-Paare. Datensätze, die mit $z = 0$ spezifiziert sind, beinhalten Ereignisse aus bei $z = 0\ cm$ im Zentrum des H1-Detektors lokalisierten $eP$-Kollisionen, im Gegensatz zu $z = \pm 25\ cm$, bei denen sich die $eP$-Kollisionen über die gesamte Wechselwirkungszone erstrecken.

#### Wertebereich der Histogramm-Bins

Die Bins der verschiedenen Histogramme des Z-Triggers, die als Eingangsaktivitäten für das neuronale Netzwerk dienen, erstrecken sich über einen bestimmten Wertebereich. Bei den Simulationen wurde deutlich, daß die Beschränkung des Wertebereichs zu einer besseren Klassifizierung von nicht gelernten Ereignissen führt.

Die Begrenzung des Wertebereichs steht auch mit einer möglichen elektronischen Implementation im Einklang, da Neuro-Chips nur einen begrenzten Wertebereich für die Eingangsaktivitäten zulassen. Die Bins der Z-Vertex-Histogramme brauchen nicht begrenzt zu werden (s. Abb. 6.8), die Notwendigkeit ergibt sich erst für die Zell-Histogramme (s. Abb. 6.9). Der relativ geringe Anteil der Bins, die erheblich größer als der Mittelwert sind, ist bezüglich der Überlappung der verschiedenen Musterklassen nicht von Bedeutung, so daß das Abschneiden dieser keinen negativen Einfluß auf die Separabilität hat. In der Lernphase bewirken diese hohen Bin-Werte jedoch eine Verschiebung der Fehlerfunktion $E$ gegenüber den nicht begrenzten, und somit auch eine Verfälschung der internen Repräsentation der unterschiedlichen Musterklassen, was in der Testphase einen negativen Einfluß auf die Klassifizierung nicht gelernter Ereignisse zur Folge hat.

[6] *engl. beamwall* ≈ Strahl-Wand

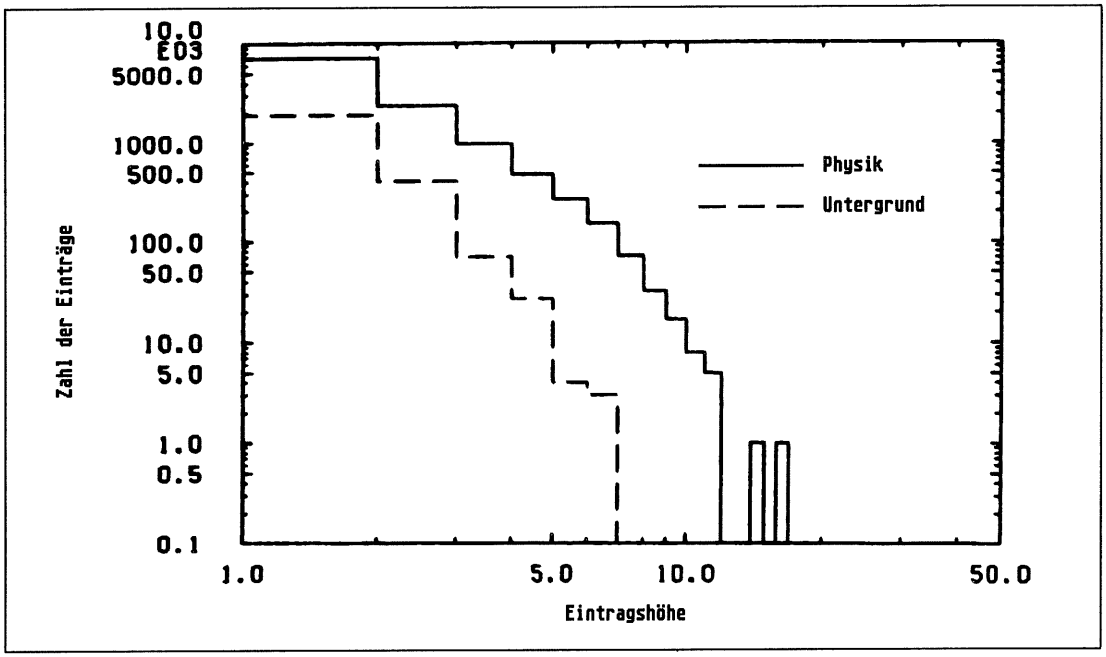


Abbildung 6.8: **Verteilung der Z-Vertex-Histogramm-Bins** *Der Wertebereich ist im wesentlichen auf den Bereich* $\approx (0), 1, \ldots, 15$ *beschränkt, mit einem Mittelwert von* $\approx 3$.

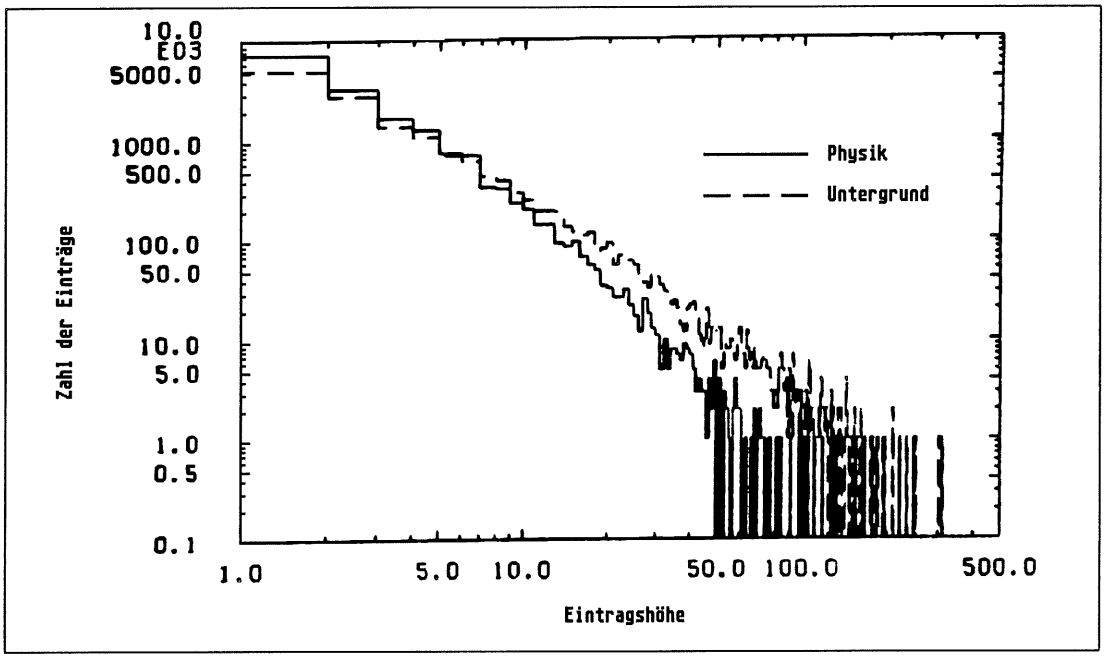


Abbildung 6.9: **Verteilung der Zell-Histogramm-Bins** *Der Anteil von Ereignisse hoher Multiplizität führt zu großen Einträgen in den Zell-Histogrammen.*

## 6.4 Effizienz und Untergrundraten

Für die Beurteilung der durch das neuronale Netzwerk erreichten Klassifizierung sind die Anteile der richtig klassifizierten Muster aller Klassen zu betrachten; denn die Verbesserung der Klassifizierung einer Klasse kann eine Verschlechterung für die andere(n) Klasse(n) zur Folge haben. Im Trivialfall werden alle Muster mit dem gleichen Wert klassifiziert.

Für die Beurteilung der Effizienz des Trigger-Systems, in Bezug auf physikalisch relevante Ereignisse, ist der prozentuale Anteil richtig klassifizierter physikalisch relevanter Ereignisse entscheidend. Außerdem ist - insbesondere für den Z-Trigger - von Bedeutung, zu welchem Anteil physikalisch relevante Ereignisse niedriger Multiplizität verloren gehen. Die Angabe einer Rate macht in diesem Zusammenhang keinen Sinn, da sie bei vorgegebener Luminosität vom Wirkungsquerschnitt, der im Einzelfall betrachteten Reaktion abhängt.

Für die Beurteilung der Untergrundunterdrückung sind die mit dem prozentualen Anteil richtig klassifizierter Untergrundereignisse verknüpften absoluten Untergrundraten von Bedeutung. Die resultierende Untergrundrate ist der Zahl falsch klassifizierter Untergrundereignisse proportional. Die datensatzspezifischen Werte *Untergrundrate/Ereignis* wurden der Datensatzbeschreibung HERA02.H1FILES(+FILES) entnommen.

Für die quantitative Beurteilung der Simulationsergebnisse, in Bezug auf die Einsetzbarkeit eines neuronalen Netzwerks zur Untergrundunterdrückung im Zusammenhang mit dem Z-Trigger, sind die mit dem neuronalen Netzwerk erreichte Effizienz bezüglich der Physik (prozentual) sowie die erreichte Untergrundrate (absolut) gemeinsam zu betrachten.

## 6.5 Das Programmpaket JETNET

Zur Simulation des BPG-Netzwerks kann auf bestehende Programme zurückgegriffen werden. Für die vorliegende Arbeit wurde das in FORTRAN 77 geschriebene Programmpaket JETNET (Version 1.1) verwendet [LUTP90-6]. JETNET stellt eine Sammlung von Routinen zur Simulation eines BPG-Netzwerks gemäß der in Kapitel 5 dargestellten Algorithmen dar. Eine vergleichbare Sammlung von Funktionen zur Simulation eines BPG-Netzwerks (sowie anderer Netzwerkmodelle) liegt auch dem Buch [McCeRum88] als C-Quell-Code bei.

Die Routinen von JETNET, als auch die für die Kommunikation dieser untereinander verwendeten common-Blöcke, sind in [LUTP90-6] und der zum Quell-Code gehörenden Dokumentation[7] beschrieben.

Um JETNET für die Simulationen auf der IBM-Großrechenanlage verwenden zu können, mußten die Eingabe- und Ausgabe-Routinen etwas modifiziert werden, sowie eine Routine zum Lesen der Histogramm-Daten geschrieben werden. Für die eigentliche Simulation wurde ein Rahmenprogramm mit Hilfe von Job-Karten in der JCL[8] erstellt, durch die auch die Speicherplatzanforderung die Datensätze erfolgte. Die Routinen zum Lernen und Testen wurden gemäß der Beispiele aus der Dokumentation von JETNET erstellt, und durch Routinen zur numerischen und graphischen Protokollierung (GEP) erweitert.

[7] Beides ist auf Anfrage via BITNET von den Autoren erhältlich, s. [LUTP90-6]

[8] JCL = **J**ob **C**ontrol **L**anguage ≈ Aufgaben Steuersprache (IBM)

### 6.5.1 Simulationsparameter

Allen durchgeführten Simulationen lag ein dreischichtiges BPG-Netzwerk der in Abbildung 5.6 (S. 37) dargestellten Struktur mit grundsätzlich den in Tabelle 6.3 (S. 70) zusammengefaßten Parametern zugrunde.

| | |
|---|---|
| Ausgangs-Neuronen $y_i$ = | 1 |
| Hidden-Neuronen $h_j$ = | 16 |
| Eingangs-Neuronen $x_k$ = | 96, 174 |
| $g(x,T)$ = | $1/(1+exp(-x/T))$, mit $T = 0,5$ |
| $\pm\Delta\omega_{init}$ = | 0,1 |
| $\eta$ = | 0,001 |
| $\alpha$ = | 0,5 |
| $\omega$-Aktualisierung : | Nach jedem 10ten Lernmuster |
| Lernzyklen = | 200 ($\cdot$*Lernmusterzahl*) |
| $y_1$ = | 0 $\Leftrightarrow$ Untergrund ($y_1 < 0,5$) |
| | 1 $\Leftrightarrow$ Physik ($y_1 \geq 0,5$) |
| $x_i \in$ | $[0,\ldots,15]$ (Bin-Begrenzung) |
| Zufallsgenerator : | H1RND |

Tabelle 6.3: **Standardparameter**

Der in der Tabelle 6.3 mit $\omega$-Aktualisierung bezeichnete Parameter bedarf einer kurzen Erklärung: Wie im Abschnitt 5.4 (S. 37) beschrieben, kann die Korrektur der synaptischen Gewichte $\omega$ nach jedem Anlegen eines Lernmusters, und durchgeführter Fehlerberechnung erfolgen, es ist aber auch eine kumulativer Korrektur der synaptischen Gewichte $\omega$ möglich. Der Parameter $\omega$-Aktualisierung gibt an, wie oft ein neues Lernmuster an das BPG-Netzwerk angelegt wird, bis tatsächlich die Korrektur der synaptischen Gewichte durchgeführt wird.

Bei den durchgeführten Simulationen diente die Zahl der Lernzyklen als Abbruchkriterium, da dies in einfacher Weise die Festlegung der für die Stapel-Verarbeitung erforderliche Zeitbegrenzung erlaubt [New88] und außerdem besser vergleichbare Ergebnisse liefert. Prinzipiell kann auch der quadratische Fehler $E$ oder die Zahl der richtig klassifizierten Muster als Abbruchkriterium dienen.

Um ein optimales Konvergenzverhalten des BPG-Netzwerks in der Lernphase zu gewährleisten, wurden die Lernmuster zufällig und abwechselnd aus den beiden Musterklassen ausgewählt (s. Abschnitt 5.4.2, S. 41).

# Kapitel 7

# Ergebnisse

Die Simulationen des BPG-Netzwerks wurden in Hinblick auf eine elektronische Implementation durchgeführt. Dazu wurde untersucht, wie gut sich die Trennung von Untergrund und Physik mit dem neuronalen Netz realisieren läßt. Dabei wurden die besonderen Eigenschaften des BPG-Netzwerks deutlich.

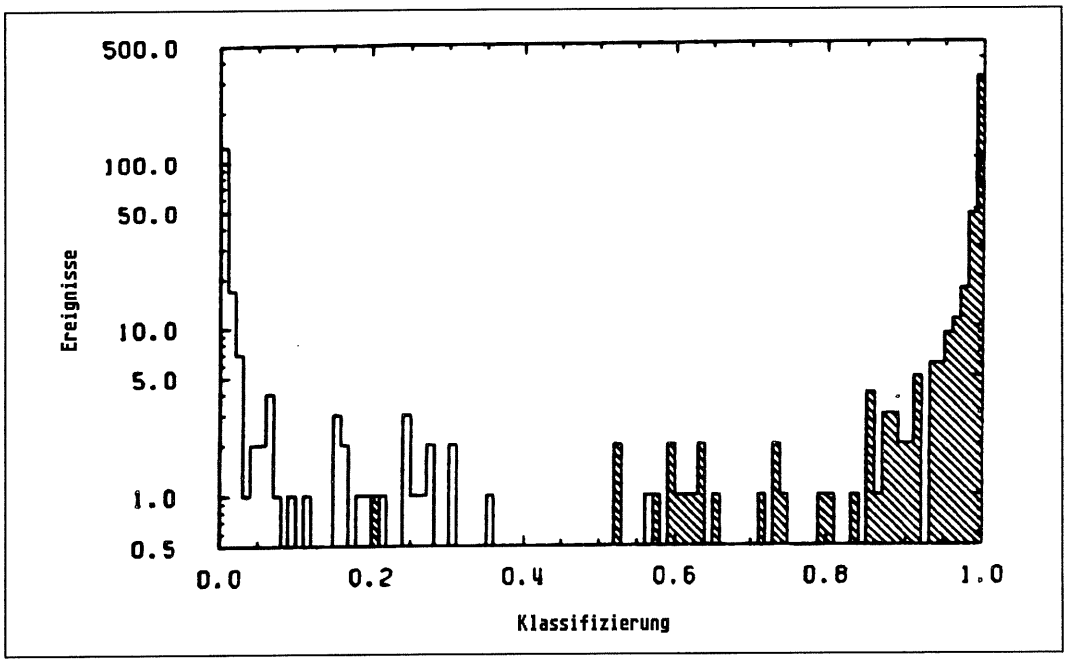


Abbildung 7.1: **Lokalisierte $b\bar{b}$-Ereignisse** . *Das BPG-Netz ordnet jedem Ereignis eine reelle Zahl $y = 0 \ldots 1$ zu (Abszisse). Die (semilogarithmische) Histogrammdarstellung erlaubt die qualitative Beurteilung. Bei einer Schwelle von $y_s = 0,5$ wurden im vorliegenden Fall Untergrundereignisse zu $99,4\%$ mit $y < y_s$, und bei z=0 cm lokalisierte $b\bar{b}$-Ereignisse zu $99,8\%$ mit $y \geq y_s$ klassifiziert. Durch Variieren der Schwelle $y_s$ läßt sich eine Anpassung der resultierenden Untergrundrate an experimentelle Erfordernisse erreichen.*

Die erzielten Ergebnisse stimmen zuversichtlich, daß die Erweiterung des Z-Kammer-Triggers durch ein neuronales Netzwerk ein leistungsfähiges Gesamtsystem ergibt.

## 7.1 Verhalten des neuronalen Netzwerks

Das Verhalten des neuronalen Netzwerks in der Lern- und Testphase hat für die Interpretation der Ergebnisse jeweils eine andere Bedeutung. In der Lernphase wird der Grad der Überlappung von Mustern *verschiedener Klassen* deutlich[1], also bis zu welchem Grade die verschiedenen Musterklassen prinzipiell anhand der verfügbaren Information (Z-Vertex- und Zell-Histogramme) separabel sind. Das Konvergenzverhalten gibt Aufschluß darüber, ob die den Lernvorgang bestimmenden Parameter in geeigneter Weise gewählt sind. In der Testphase gibt die Klassifizierung von *nicht gelernten Mustern* Aufschluß darüber, in wieweit die zum Lernen verwendeten Muster für die Gesamtheit der verfügbaren Muster repräsentativ sind.

#### Repräsentativität von Lernmustern

Die zum Lernen verwendeten Muster sind dann für ihre Klasse repräsentativ, wenn ihre Verteilung alle möglichen Muster beschreibt. Dabei ist nicht ausschließlich die Zahl der Lernmuster ausschlaggebend. Dieser Punkt ist für die praktische Anwendung des neuronalen Netzwerks von großer Bedeutung. Die Klassifizierung von realen Daten wird nur dann ein gutes Ergebnis liefern, wenn die zum Lernen verwendeten (idealisierten) Monte-Carlo-Daten für die Gesamtheit aller möglichen (realen) Ereignisse repräsentativ sind, wenn also die Lern- und Testmuster *derselben* Klassen *maximal* überlappen.

### 7.1.1 Lernphase

Die graphische Darstellung der zeitlichen Entwicklung des mittleren quadratischen Fehlers $E$ (s. Abb. 7.2) sowie der Anteile richtig klassifizierter Ereignisse (s. Abb. 7.3), erlaubt die qualitative Beurteilung des Konvergenzverhaltens in der Lernphase. Dabei wird ein Untergrundereignis zunächst als richtig klassifiziert gewertet, wenn das Netzwerk diesem einen Wert $<0,5$ zuordnet. Ein Physikereignis wird bei einem Wert $\geq 0,5$ als richtig klassifiziert gewertet. Die gewünschte Zuordnung wird in der Lernphase festgelegt.

#### Normale Konvergenz

Bei Vorgabe geeigneter Parameter konvergiert der BPG-Algorithmus schnell. Der quadratische Fehler $E$ fällt schnell gegen Null, und damit einher geht der Anteil der zum Lernen verwendeten Ereignisse, die richtig klassifiziert werden, gegen den theoretisch möglichen Grenzwert, der durch die Überlappung der verschiedenen Musterklassen bestimmt ist.

#### Gestörte Konvergenz

Bei ungünstig gewählten Parametern ist das Konvergenzverhalten des neuronalen Netzwerks in der Lernphase gestört. Die Ursache der dabei auftretenden Oszillationen liegt in lokalen Minima der Fehlerfunktion, die bei ungünstig gewählten Parametern nur unzureichend bedämpft werden. Als Beispiel dafür sind der Fehler $E$ und die Zahl der richtig klassifizierten Ereignisse für einen (zu großen) Lernparameter $\eta = 0,5$ in den Abbildungen 7.4 und 7.5 über der Zahl der durchgeführten Lernzyklen aufgetragen.

[1] s. Bayesian Limit, Abschnitt 5.4.5

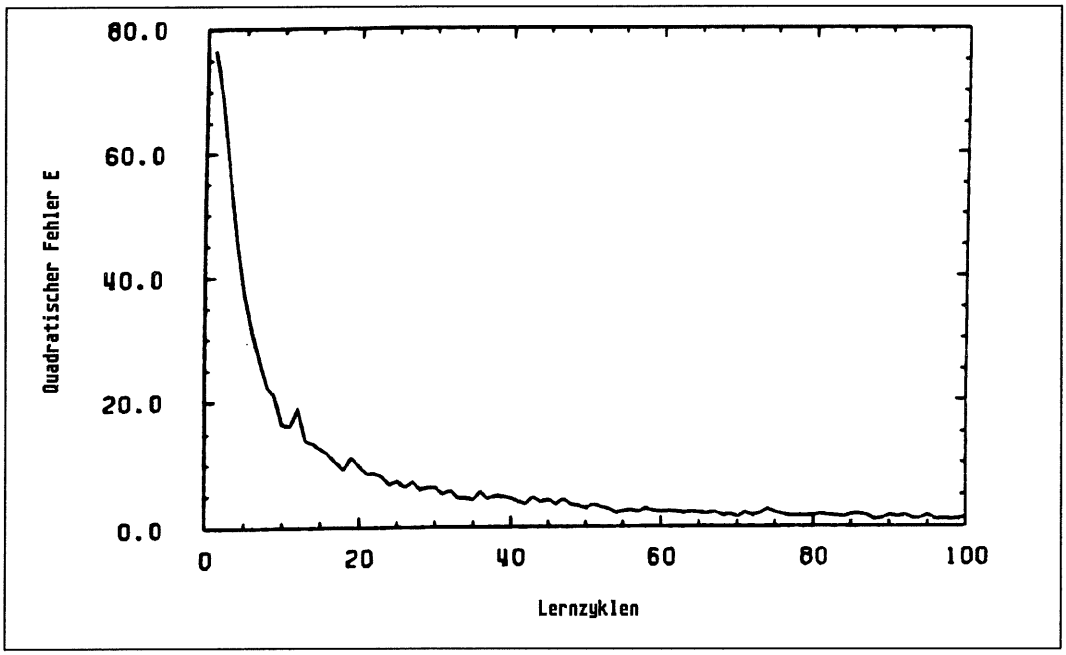


Abbildung 7.2: **Quadratischer Fehler E** , *aufgetragen als Funktion der durchgeführten Lernzyklen, im Fall von normaler Konvergenz.*

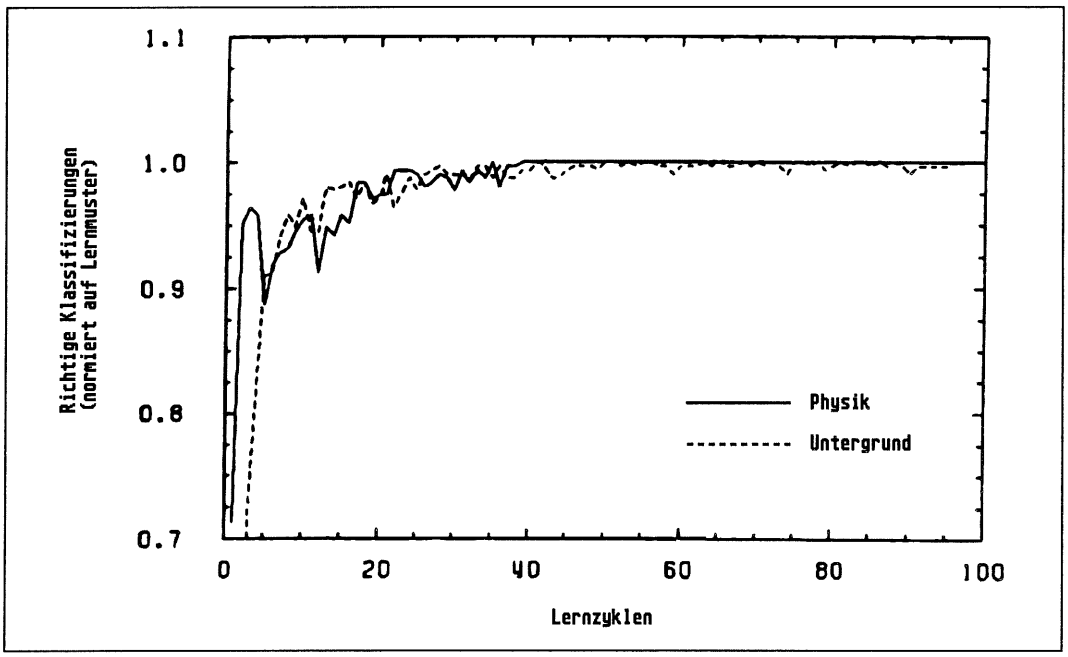


Abbildung 7.3: **Klassifizierung in der Lernphase** . *Aufgetragen ist der Anteil der richtig klassifizierten Ereignisse als Funktion des Lernzyklus, zu dem in Abbildung 7.2 dargestellten quadratischen Fehler E.*

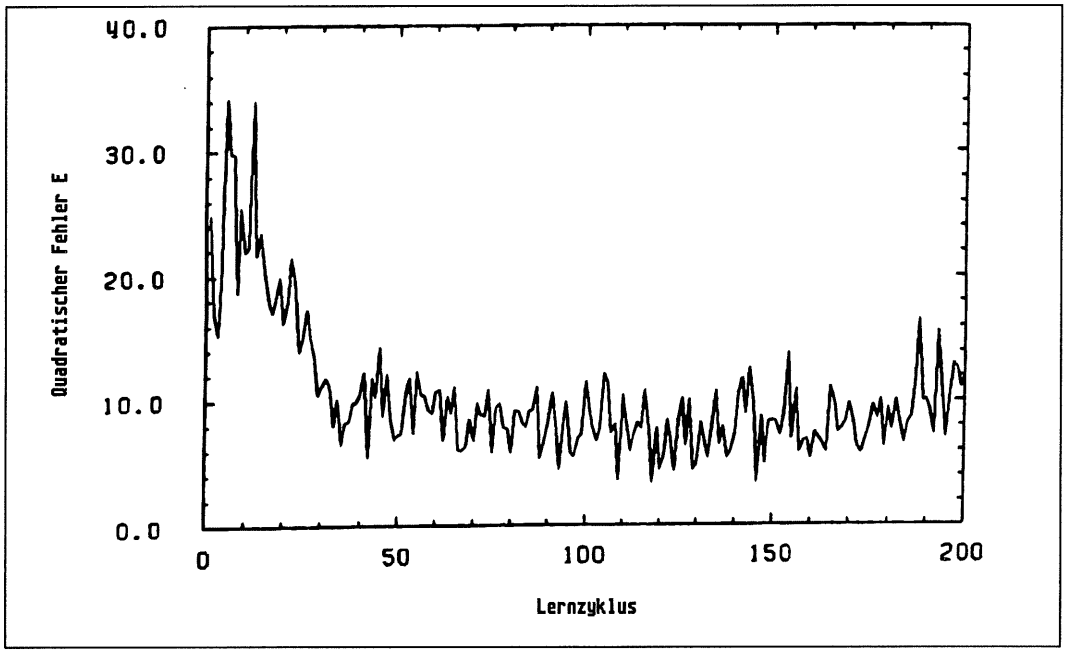


Abbildung 7.4: **Divergenter Fehler E** . *Bei einem zu groß gewählten Lernparameter $\eta = 0,5$ zeigt der quadratische Fehler E einen sehr unregelmäßigen Verlauf.*

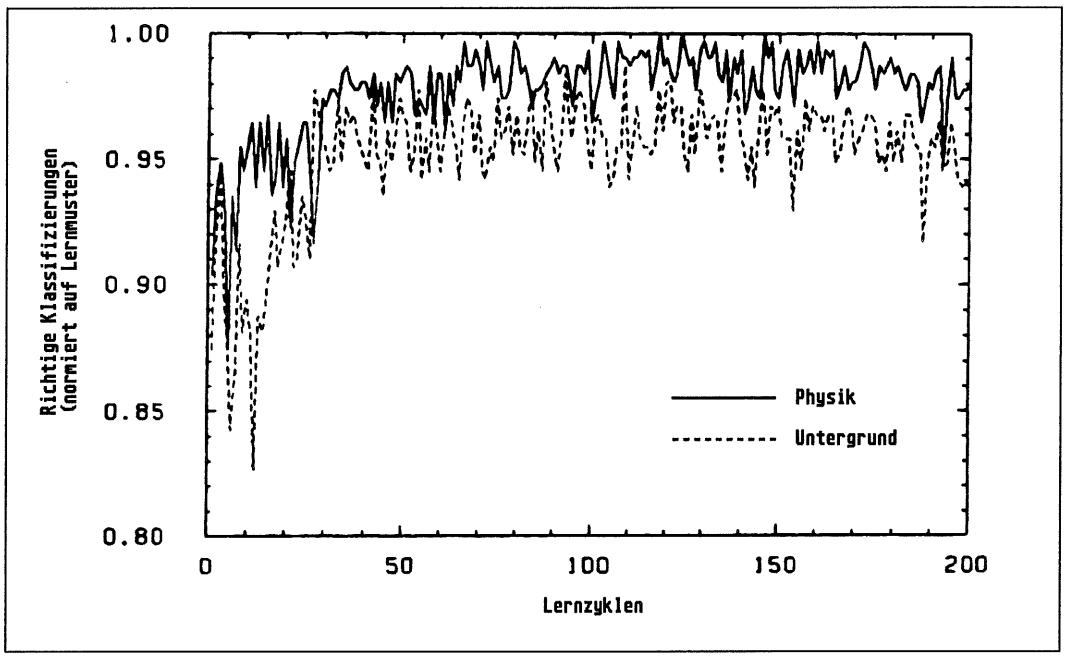


Abbildung 7.5: **Divergente Klassifizierung** *für ein zu großes $\eta = 0,5$. Neben dem in Abbildung 7.4 aufgetragenen quadratischen Fehler E schwankt auch die Zahl der in der Lernphase richtig klassifizierten Ereignisse in sehr unregelmäßiger Weise.*

## 7.2 Simulationsergebnisse

Anhand der in Tabelle 7.1 und Tabelle 7.2 zusammengefaßten Ergebnisse wird deutlich, daß das neuronale Netzwerk eine wirkungsvolle Untergrundunterdrückung (s. Spalte: *Untergrund*) zusammen mit einer hohen Effizienz (s. Spalte: $y_1 \geq y_s$) in Bezug auf physikalisch relevante Ereignisse erlaubt. In die jeweils mit $x_i$ bezeichnete Spalte ist die Zahl der Eingangs-Neuronen des bei der betreffenden Simulation verwendeten Netzwerks eingetragen (96 $\Leftrightarrow$ Z-Vertex-Histogramm, 174 $\Leftrightarrow$ Z-Vertex-Histogramm & Zell-Histogramme). Die mit *train/test* bezeichnete Spalte gibt jeweils die für die Lernphase bzw. Testphase des neuronalen Netzwerks verwendeten Datensatzteile an (s. Abb. 6.7). Die mit (Abb.) gekennzeichneten Ergebnisse sind zur qualitativen Beurteilung als Klassifizierungs-Histogramme dargestellt.

| **Parameter** | | | **Klassifizierung** für Schwelle $y_s = 0,5$ | | |
|---|---|---|---|---|---|
| Physik, $z[cm]$ | $x_i$ | train / test | $y_0 < y_s$ | *Physik* : $y_1 \geq y_s$ | *Untergrund* |
| $b\bar{b}$, 0 | 96 | 1(2) | 2(2) | 95,6% (172/180) | 97,7% (431/441) | 225 *Hz* |
| $b\bar{b}$, 0 | 174 | 1(2) | 2(2) | 93,9% (169/180) | **99,1%** (437/441) | **275** *Hz* |
| $b\bar{b}$, 0 (Abb.7.1) | 174 | 2(2) | 1(2) | 99,4% (179/181) | **99,8%** (440/441) | **75** *Hz* |
| $b\bar{b}$, $\pm 25$ | 96 | 1(2) | 2(2) | 74,4% (134/180) | 78,7% (196/249) | 1150 *Hz* |
| $b\bar{b}$, $\pm 25$ (Abb.7.6) | 174 | 1(2) | 2(2) | 87,2% (157/180) | **98,0%** (244/249) | **575** *Hz* |
| $l\bar{l}$, $\pm 25$ | 96 | 1(2) | 2(2) | 59,4% (107/180) | 85,1% (263/309) | 1825 *Hz* |
| $l\bar{l}$, $\pm 25$ (Abb.7.7) | 174 | 1(2) | 2(2) | 95,0% (171/180) | **99,4%** (307/309) | **225** *Hz* |

Tabelle 7.1: **Simulationsergebnisse** *Durch den Vergleich der Ergebnisse mit Zell-Histogrammen (174) und ohne (96) wird die Bedeutung dieser, insbesondere bei der Klassifizierung von Zwei-Spur-Ereignissen $l\bar{l}$, deutlich. Problematisch ist die Repräsentativität der Lernmuster, die auch Ursache für das unbefriedigende Ergebnis bei den über die gesamte Vertex-Region verteilten $b\bar{b}$-Ereignissen ist, was auch bei den lokalisierten $b\bar{b}$-Ereignissen deutlich wird, wo die Vertauschung der Lern- und Test-Daten (train/test: 1(2)/2(2) $\leftrightarrow$ 2(2)/1(2)) zu sehr unterschiedlichen Ergebnissen führte. Die* **fett** *dargestellten Ergebnisse werden im folgenden mit Simulationen zu experimentellen und technischen Randbedingungen verglichen (s. Tab 7.3 und 7.4).*

| **Parameter** | | | **Klassifizierung** für Schwelle $y_s = 0,5$ | | |
|---|---|---|---|---|---|
| Physik, $z[cm]$ | $x_i$ | train / test | $y_0 < y_s$ | *Physik* : $y_1 \geq y_s$ | *Untergrund* |
| $b\bar{b}$, 0 | 96 | 1+2 | 1+2 | 98,9% (357/361) | 97,6% (861/882) | 100 *Hz* |
| $b\bar{b}$, 0 | 174 | 1+2 | 1+2 | 100% (361/361) | 100% (882/882) | 0 *Hz* |
| $b\bar{b}$, $\pm 25$ | 96 | 1+2 | 1+2 | 83,7% (302/489) | 73,1% (364/498) | 1475 *Hz* |
| $b\bar{b}$, $\pm 25$ | 174 | 1+2 | 1+2 | 99,4% (359/361) | 100% (499/499) | 50 *Hz* |
| $l\bar{l}$, $\pm 25$ | 96 | 1+2 | 1+2 | 71,5% (258/361) | 86,4% (533/617) | 2575 *Hz* |
| $l\bar{l}$, $\pm 25$ | 174 | 1+2 | 1+2 | 99,4% (359/361) | 100% (617/617) | 50 *Hz* |

Tabelle 7.2: **Überlappungsgrad** *Die Klassifizierung von Gesamtdaten (1+2) liefert eine untere Schranke für den Überlappungsgrad der unterschiedlichen Musterklassen, und zeigt damit, bis zu welchem Grad die Separation von Untergrund und Physik durch das neuronale Netz möglich ist.*

Die Anteile richtig klassifizierter Ereignisse beziehen sich auf die *Sekundärdaten* (Histogramme des Z-Triggers) und erlauben daher die Beurteilung der vom neuronalen Netzwerk durchgeführten Separation von Untergrund und Physik. Dem vorangegangen ist die Verarbeitung der *Primärdaten* (Signale der Driftkammern) durch den (simulierten) Z-Trigger, wobei eine Untergrundunterdrückung von $\approx 90\%$ und eine Effizienz von ebenfalls $\approx 90\%$ erreicht wurde (s. Tab 6.2). Diese Effizienz hängt von der Winkelverteilung physikalisch interessanter Ereignisse ab.

#### Klassifizierungs-Histogramm

Die semilogarithmische Histogrammdarstellung der vom neuronalen Netzwerk erreichten Klassifizierung stellt besonders deutlich den Anteil der falsch sowie nicht eindeutig zugeordneten Ereignisse heraus (s. Abb. 7.1, 7.6, 7.7), und erlaubt so die qualitative Beurteilung. Inwieweit eine Schwelle $\neq 0{,}5$) zu einem Verlust von physikalisch relevanten Ereignissen führt, auf die zu Gunsten einer niedrigeren Untergrundrate verzichtet werden kann, kann anhand des Histogramms abgeschätzt werden.

#### Kontinuierliche Klassifizierung

Die Eigenschaft des neuronalen Netzwerks, ein Ereignis mit einer reellen Zahl $0 \leq y \leq 1$ zu klassifizieren, liefert die Möglichkeit Systemen höherer Trigger-Stufen im Fall nicht eindeutig klassifizierter Ereignisse dieses als Zusatzinformation zur Verfügung zu stellen, wenn die Klassifizierung eines Ereignisses etwa in den Bereich $0{,}2 < y < 0{,}8$ fällt. Der vom neuronalen Netzwerk gelieferte Wert $y$ ist ein Maß für Ähnlichkeit oder den Bekanntheitsgrad eines klassifizierten Ereignisses zu einer bestimmten Ereignisklasse.

### 7.2.1 Klassifizierungs-Schwelle

Die Verschiebung der Klassifizierungs-Schwelle $y_s$ erlaubt auch während des Experiments die Untergrundunterdrückung kontinuierlich zu beeinflussen und so experimentellen Gegebenheiten anzupassen. Der in der Regel vom neuronalen Netzwerk erreichte hohe Anteil richtig klassifizierter physikalisch relevanter Ereignisse läßt die Verschiebung der in der Tabelle 7.1 zunächst vorgegebenen Schwelle von $y_s = 0{,}5$ hin zu größeren Werten zu, was eine stärkere Reduzierung der Untergrundraten ohne deutliche Verschlechterung der Effizienz bewirkt.

| $y_s \rightarrow$ | 0,1 | 0,2 | 0,3 | 0,4 | 0,5 | 0,6 | 0,7 | 0,8 | 0,9 |
|---|---|---|---|---|---|---|---|---|---|
| $b\bar{b}$, 0, 2(2) | 100%<br>550 Hz | 99,5%<br>500 Hz | 99,3%<br>325 Hz | 99,3%<br>300 Hz | **99,1%**<br>**275 Hz** | 98,9%<br>250 Hz | 98,4%<br>150 Hz | 97,7%<br>125 Hz | 94,6%<br>50 Hz |
| $b\bar{b}$, 0, 1(2) | 100%<br>1,4 kHz | 100%<br>1 kHz | 100%<br>825 Hz | 99,8%<br>75 Hz | **99,8%**<br>**75 Hz** | 99,8%<br>25 Hz | 98,6%<br>0 Hz | 96,1%<br>0 Hz | 92,7%<br>0 Hz |
| $b\bar{b}$, $\pm 25$ | 100%<br>800 Hz | 99,6%<br>725 Hz | 99,2%<br>700 Hz | 98,8%<br>600 Hz | **98,0%**<br>**575 Hz** | 94,0%<br>525 Hz | 88,8%<br>400 Hz | 81,5%<br>300 Hz | 68,3%<br>175 Hz |
| $l\bar{l}$, $\pm 25$ | 100%<br>600 Hz | 100%<br>275 Hz | 100%<br>250 Hz | 100 %<br>225 Hz | **99,4%**<br>**225 Hz** | 99,0%<br>225 Hz | 98,4%<br>225 Hz | 97,1%<br>175 Hz | 92,9%<br>100 Hz |

Tabelle 7.3: **Effizienz und Untergrundraten** *in Abhängigkeit von der Schwelle $y_s$. Die Ergebnisse dieser Tabelle sind in den Abbildungen 7.9, 7.8, 7.10 und 7.11 graphisch dargestellt.*

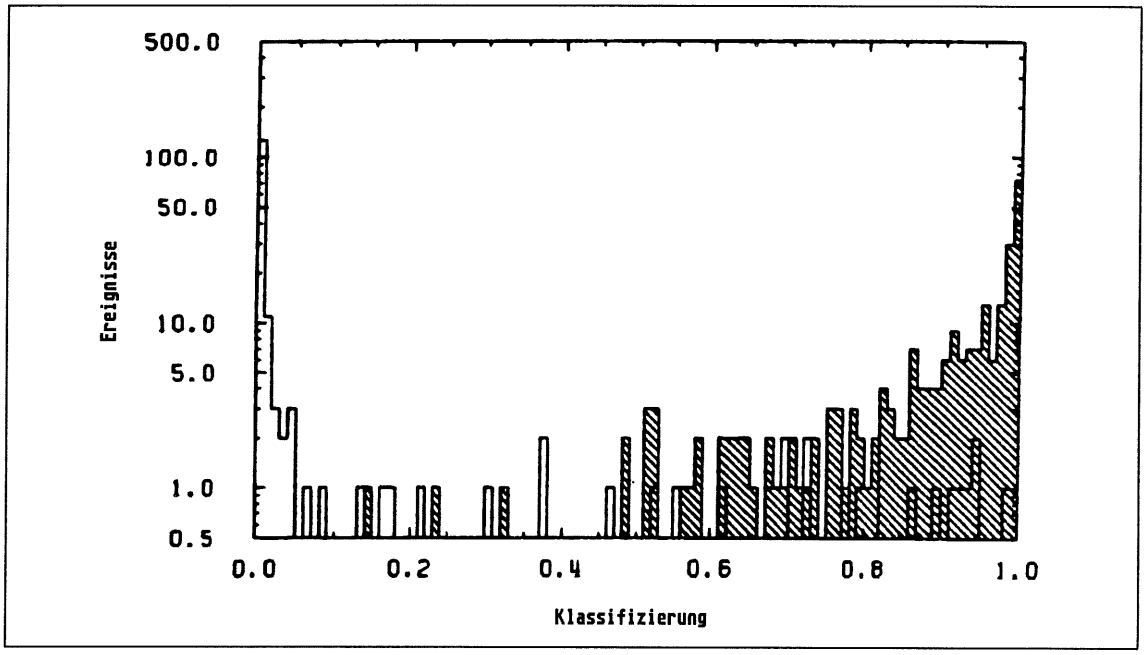


Abbildung 7.6: **Verschmierte $b\bar{b}$-Ereignisse** *wurden zu* $98,0\%$ *mit einem Wert* $y \geq 0,5$, *die Untergrundereignisse zu* $87,2\%$ *mit* $y_0 < 0,5$ *klassifiziert.*

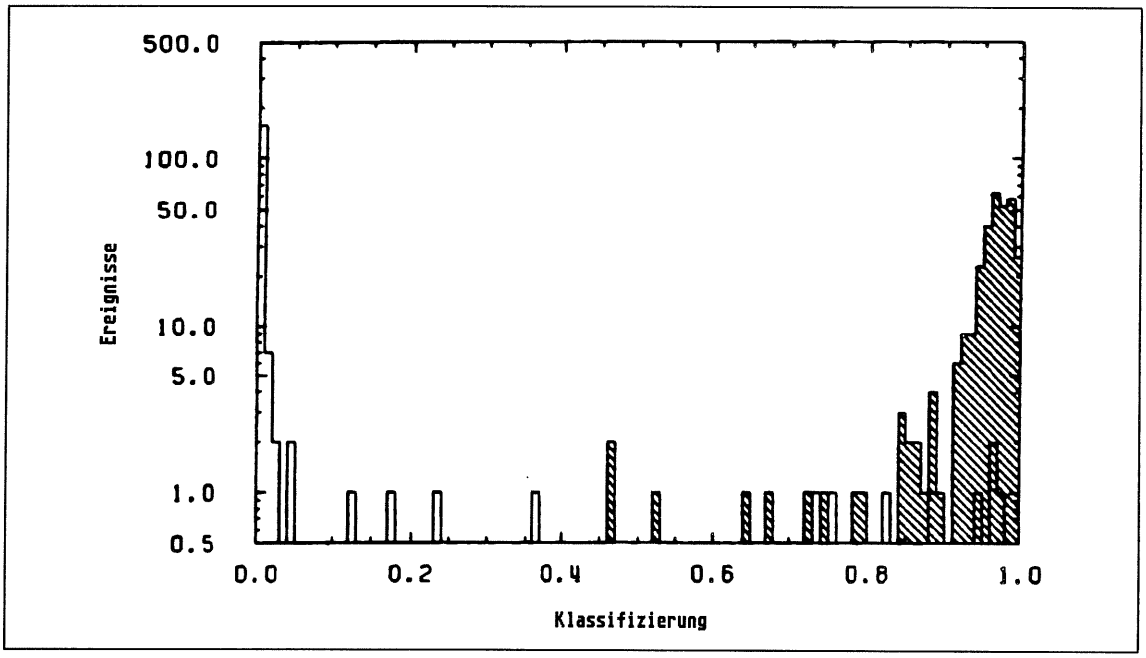


Abbildung 7.7: **Verschmierte Zwei-Spur-Ereignisse** *wurden zu* $99,4\%$ *mit* $y \geq 0,5$ *und die Untergrundereignisse zu* $95,0\%$ *mit* $y>0,5$ *klassifiziert, was auf Zell-Histogramme zurückzuführen ist.*

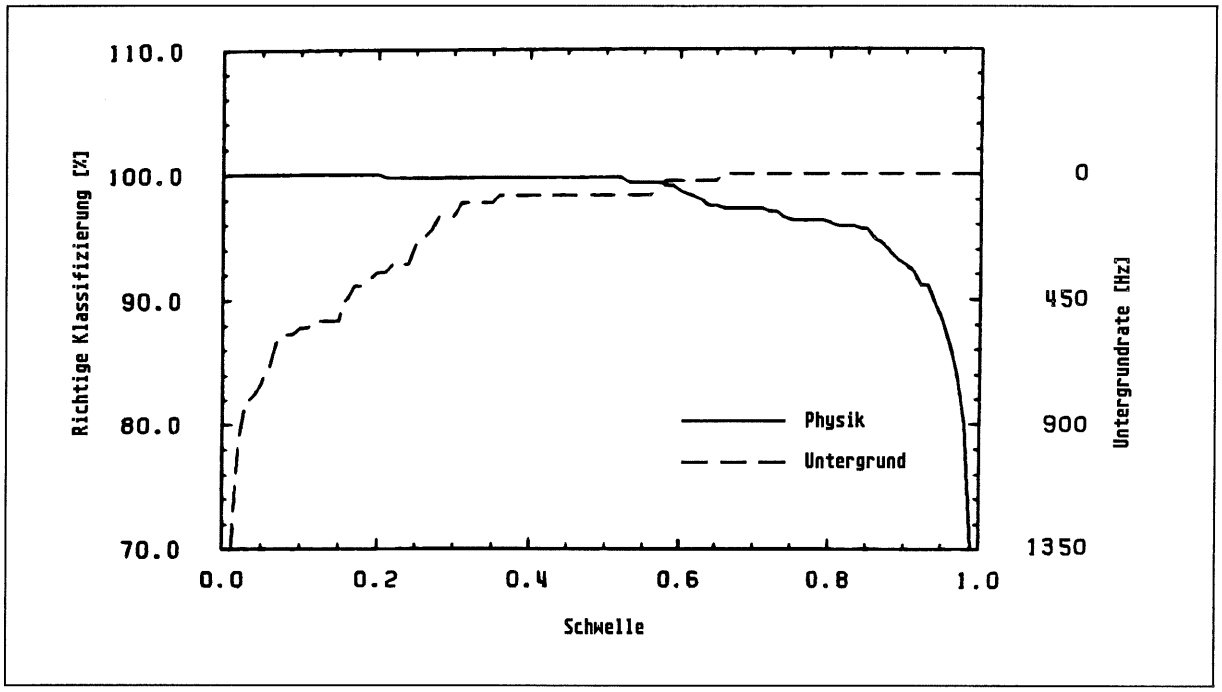


Abbildung 7.8: **Effizienz lokalisierter $b\bar{b}$-Ereignisse 1(2)** *(z=0 cm) und Untergrundrate in Abhängigkeit von der Schwelle $y_s$.*

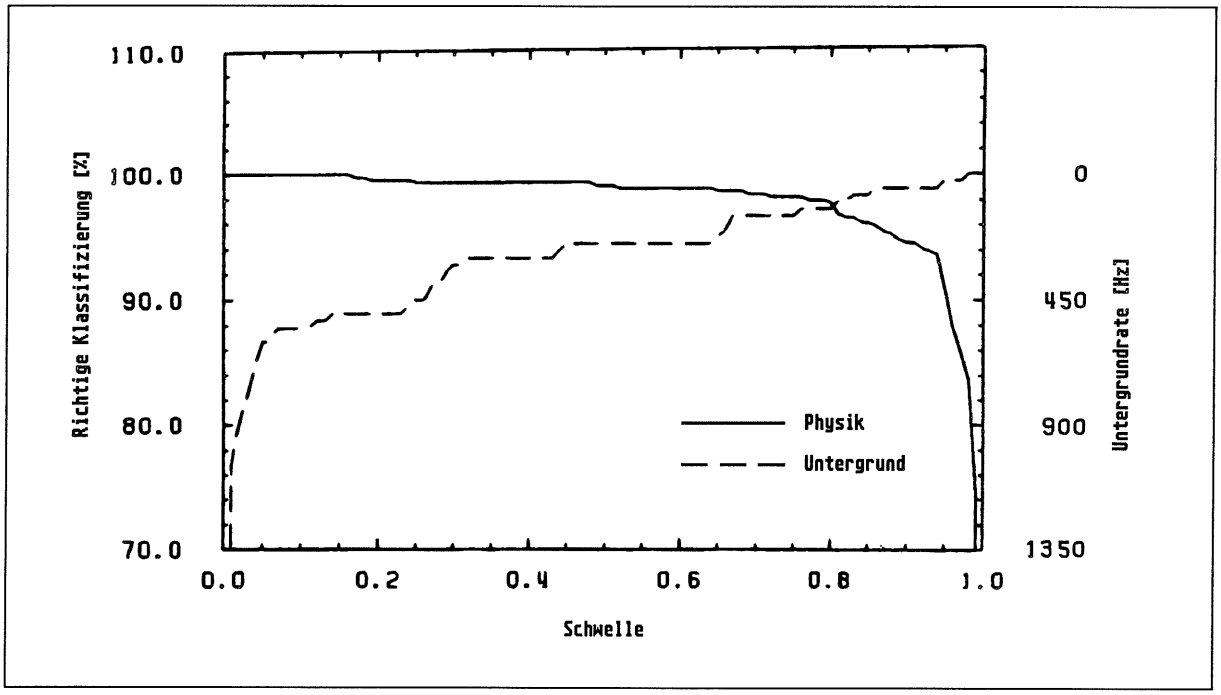


Abbildung 7.9: **Effizienz lokalisierter $b\bar{b}$-Ereignisse 2(2)** *(z=0 cm) und Untergrundrate in Abhängigkeit von der Schwelle $y_s$.*

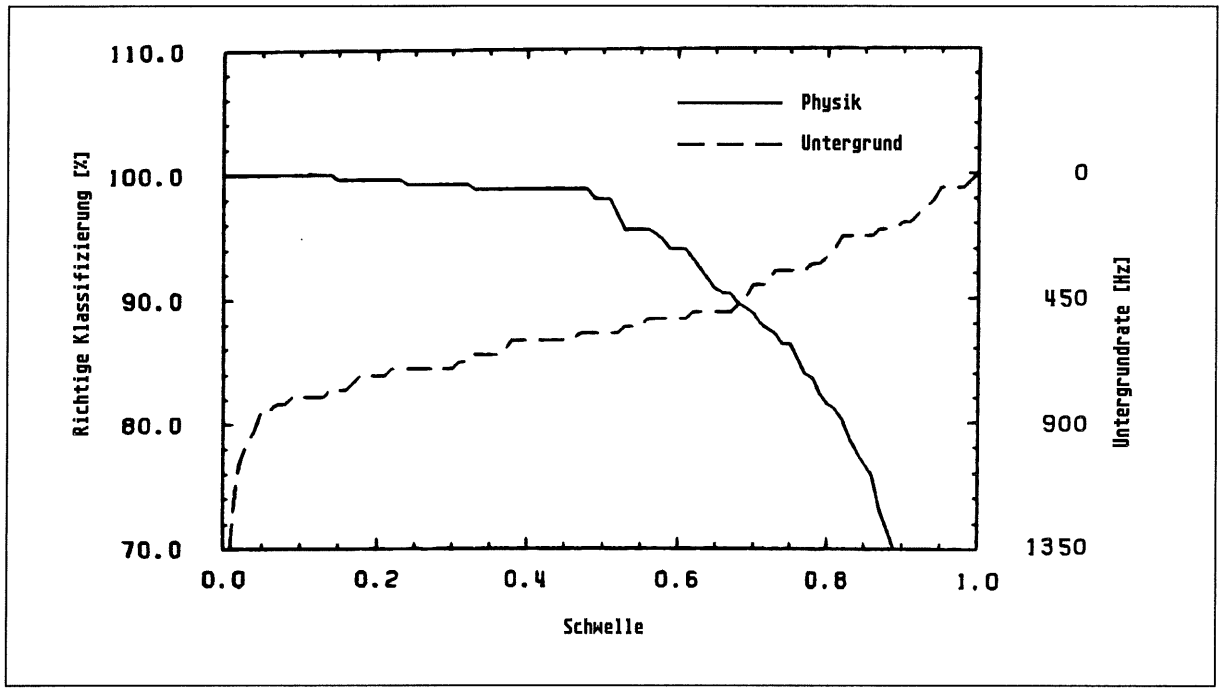


Abbildung 7.10: **Effizienz verschmierter $b\bar{b}$-Ereignisse 2(2)** *($z$=±25 cm) und Untergrundrate in Abhängigkeit von der Schwelle $y_s$.*

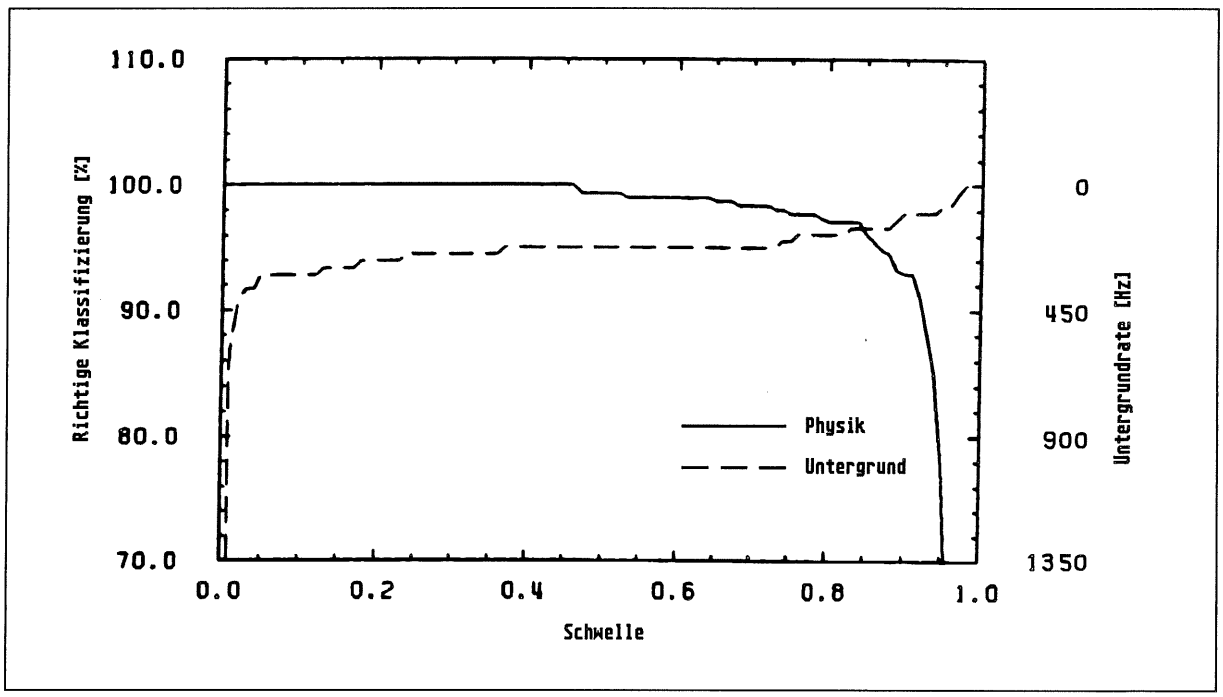


Abbildung 7.11: **Effizienz verschmierter Zwei-Spur-Ereignisse 2(2)** *($z$=±25 cm) und Untergrundrate in Abhängigkeit von der Schwelle $y_s$.*

### 7.2.2 Technische Randbedingung durch synaptischen Gewichte

Alle im vorangegangenen beschriebenen Simulationsergebnisse wurden auf Grundlage von quasi reell[2] dargestellten synaptischen Gewichten $\omega$ durchgeführt. Neuro-Chips können die synaptischen Gewichte jedoch nur mit einer endlichen Genauigkeit darstellen. Bei digital arbeitenden Neuro-Chips ist die Darstellung zudem noch diskret und die Informationsverarbeitung erfolgt auf Grundlage von ganzzahliger Arithmetik, da ein Rechenwerk für die Arithmetik mit reellen Zahlen einen höheren Schaltungsaufwand erfordert. Sowohl bei digital als auch bei analog arbeitenden Neuro-Chips wird die Darstellungsgenauigkeit der synaptischen Gewichte in *Bit* angegeben. Mit $m$ Bits lassen sich natürliche Zahlen $n$ im Bereich

$$n \in \{0, \ldots, 2^{m-1}\} \qquad \text{mit} \qquad n \in \mathbb{N} \tag{7.1}$$

darstellen, ganze Zahlen (2er-Komplement) im Bereich von

$$z \in \{-2^{m-1}, \ldots, 0, \ldots, 2^{m-1} - 1\} \qquad \text{mit} \qquad z \in \mathbb{Z}. \tag{7.2}$$

Steht ein Wort mit $m$ Bits für ein reelles Zahlenintervall, so wird dieses dadurch in $2^m$ Teilintervalle diskretisiert. Die bisherigen Simulationsergebnisse werden nur dann auf geeignete Neuro-Chips übertragen sein, wenn auch bei diskreter Darstellung der synaptischen Gewichte $\omega \in \{-1, \ldots, +1\}$ gleiche Ergebnisse zu erzielen sind. Um dieses zu untersuchen, wurde eine Diskretisierung der synaptischen Gewichte $\omega_{ij}, \omega_{jk}$ und Schwellen $\omega_{i0}, \omega_{j0}$ (Bias) durch folgenden Ausdruck vorgenommen

$$\omega_{diskr.}(m) = \frac{1}{2^{(m-1)}} \cdot \mathrm{int}(\omega_{kont.} \cdot (\frac{1}{2} + 2^{(m-1)})), \tag{7.3}$$

wobei die Funktion $\mathrm{int}(x)$ Nachkommastellen abschneidet. Gemäß der Gleichung 7.3 wurden Simulationen bei einer Schwelle von $y_s = 0,5$ mit Genauigkeiten der synaptischen Gewichte $\omega$ im Bereich von $m = 1, \ldots, 16$ mit den Tabelle 7.1 und 7.3 zugrundeliegenden Ereignissen durchgeführt. Die dabei erzielten Ergebnisse sind in Tabelle 7.4 zusammengefaßt.

| | $m(\omega)[bit]$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9-16 |
| $b\bar{b}$, 0, 2(2) | 0% *max.* | 0% *max.* | 100% *max.* | 98,9% 225 Hz | 99,5% 300 Hz | 99,1% 300 Hz | 98,9% 300 Hz | 99,1% 275 Hz | **99,1% 275 Hz** |
| $b\bar{b}$, 0, 1(2) | 0% *max.* | 100% *max.* | 100% *max.* | 99,5% 1275 Hz | 99,5% 100 Hz | 99,8% 75 Hz | 99,8% 75 Hz | 99,8% 75 Hz | **99,8% 75 Hz** |
| $b\bar{b}$, $\pm 25$ | 100% *max.* | 100% *max.* | 100% *max.* | 96,4% 950 Hz | 98,4% 625 Hz | 97,2% 550 Hz | 97,6% 575 Hz | 98,4% 575 Hz | **98,0% 575 Hz** |
| $l\bar{l}$, $\pm 25$ | 100% *max.* | 100% *max.* | 100% *max.* | 100% 275 Hz | 100% 250 Hz | 100% 225 Hz | 99,7% 225 Hz | 99,4% 225 Hz | **99,4% 225 Hz** |

Tabelle 7.4: **Diskretisierte synaptische Gewichte** *und deren Einfluß auf Effizienz [%] und Untergrundraten [Hz] bei einer Schwelle $y_s = 0,5$. Für 9-16 Bit sind die erzielten Ergebnisse mit denen bei kontinuierlicher $\omega$-Darstellung identisch (s. Tab 7.1 und Tab. 7.3). Das neuronale Netzwerk zeigt jedoch schon bei einer Genauigkeit der synaptischen Gewichte $\omega$ von $\approx$ 5-6 Bit stabiles Verhalten.*

[2]Variablen vom Typ `REAL`

### 7.2.3 Interpretation der Ergebnisse

Der große Vorteil neuronaler Netzwerke, bei der Anwendung auf komplexe Aufgaben der Musterklassifizierung diese nicht durch die explizite Formulierung von Algorithmen zu lösen, die sich unter Umständen nur äußerst schwer mit elektronischen Mitteln implementieren lassen, sondern durch Lernen mit für die Aufgabe repräsentativen Mustern, muß mit den Eigenheiten der Lernphase in Relation gesetzt werden, was anhand der stark unterschiedlichen Ergebnisse deutlich wird.

#### Klassifizierungsgrenze

Die vom Netzwerk in der Lernphase erreichte Klassifizierung kann als untere Schranke für die durch das Bayesian Limit gegebene theoretische Klassifizierungsgrenze interpretiert werden. So kann das neuronale Netzwerk verwendet werden, um diese Grenze abzuschätzen. Diesem Zweck dienen die Simulationen mit ungeteilten Sekundärdatensätzen (1+2). Dabei wurde die Bedeutung der Zell-Histogramme besonders deutlich.

#### Z-Vertex- und Zell-Histogramme

Bei den Simulationen mit bei $z = 0$ *cm* lokalisierten $b\bar{b}$-Ereignissen hat die Auswertung der Zell-Histogramme zusätzlich zum Z-Vertex-Histogramm scheinbar keinen großen Einfluß. Die Analyse der falsch klassifizierten Ereignisse macht jedoch die Notwendigkeit der Auswertung der gesamten verfügbaren Information deutlich: Wird nur das Z-Vertex-Histogramm ausgewertet, so führt es zu einem großen Verlust von Zwei-Spur-Ereignissen, was besonders bei den Simulationen deutlich wurde, für die Datensätze verwendet wurden, die ausschließlich solche Ereignisse enthielten.

#### Wertebereichsbegrenzung

Die Begrenzung der dem Netzwerk als Eingangsaktivitäten zugeführten Histogramm-Bins auf Werte $\leq 15$ erwies sich als vorteilhaft, um die erreichbare Untergrundrate weiter zu reduzieren. Die Erklärung dafür wurde bereits im Abschnitt 6.3.2 diskutiert.

#### Repräsentativität

Sind nur relativ wenige Ereignisse zum Lernen verfügbar, so kann das neuronale Netzwerk für diese zwar in der Lernphase eine interne Repräsentation finden und, wenn die Ereignisse nicht überlappen, diese auch mit 100% richtig klassifizieren. Doch wenn nicht zu jedem möglichen Punkt der Vertex-Region alle möglichen Klassen gelernt werden, so wird die Testsphase nur ein mäßiges Ergebnis liefern. Es ist also entscheidend, wieviele mögliche, nicht ähnliche Histogramme es zu jedem Z-Bin gibt, und wieviele davon im Lerndatensatz enthalten sind. Diese Problematik wurde bei über $z = \pm 25$ *cm* verschmierten Ereignissen - also mit realer Vertex-Verteilung - deutlich. Bei $z = 0$ *cm* lokalisierten $b\bar{b}$-Ereignissen führte die Vertauschung der Lern- und Testdatensatzteile zu unterschiedlichen Ergebnissen, weil der eine Teil der Datensätze weniger repräsentativ für den anderen Teil war.

### Lokalisierte $b\bar{b}$-Ereignisse

Die besten Ergebnisse wurden mit lokalisierten Ereignissen erzielt. Der Grund dafür ist, daß für das Training des neuronalen Netzwerks so viele Daten in der Lernphase zur Verfügung stehen müssen, die weitgehend alle möglichen Ereignisklassen in ausreichender Zahl enthalten müssen, wenn die Klassifizierung nicht gelernter Ereignisse erfolgreich verlaufen soll, da das neuronale Netz mit allen möglichen Klassen von Ereignisse trainiert werden muß.

### Verschmierte $b\bar{b}$-Ereignisse

Die Simulationen mit sich über die gesamte Wechselwirkungszone erstreckenden $b\bar{b}$-Ereignissen, sind verglichen mit den Ergebnissen der $b\bar{b}$-Ereignisse für $z = 0cm$, auf den ersten Blick unbefriedigend, besonders, wenn man die dabei erzielten Untergrundraten betrachtet und bedenkt, daß gerade derartige Simulationen für die praktische Anwendung des neuronalen Netzwerks von besonderer Bedeutung sind. Der Umstand, daß dabei jedoch etwa 100% der Lernmuster richtig klassifiziert wurden, ist als Zeichen dafür zu werten, daß das neuronale Netzwerk grundsätzlich in der Lage ist anhand der verfügbaren Information, Ereignisse fast vollständig zu separieren, also daß die Eigenschaften der Ereignisse der unterschiedlichen Klassen allenfalls geringfügig überlappen. Die Ursache hierfür ist, in etwas anderer Form, in der Repräsentativität der Lernmuster zu suchen. Zu jedem Ort auf der Strahlachse, innerhalb der Wechselwirkungszone, gibt es eine bestimmte Anzahl repräsentativer Ereignisse. Da für die Simulation der verschmierten Ereignisse etwa genausoviele wie für die Simulationen bei $z = 0\ cm$ verwendet wurden, verteilen sich dabei die Ereignisse über die Z-Vertex-Region, so daß zu jedem Z-Vertex-Bin nur ein relativ geringer Anteil von der Gesamtzahl der Ereignisse zur Verfügung steht ($\approx 1/96$), im Gegensatz zu den Simulationen bei $z = 0\ cm$, wo alle Ereignisse für ein Z-Vertex-Bin zur Verfügung stehen.

### Verschmierte Zwei-Spur-Ereignisse

Bei den Simulationen mit verschmierten Zwei-Spur-Ereignissen wird, verglichen mit dem Ergebnis bei den verschmierten $b\bar{b}$-Ereignissen, deutlich, daß aufgrund der einfachen Struktur der Zwei-Spur-Ereignisse - es gibt nur sehr wenig signifikant unterschiedliche Histogramme - auch bei relativ wenigen Lernmustern ein besseres Ereignis erzielt werden konnte. Daß dabei jedoch noch relativ viele Untergrund-Ereignisse falsch klassifiziert werden, liegt in der Ähnlichkeit mancher Untergrund-Ereignisse zu Zwei-Spur-Ereignissen begründet, wie anhand der betreffenden Klassifizierungs-Histogramme zu sehen ist.

### Datensatzvertauschung

In der Regel wurde jeweils der erste Teil der Sekundärdaten (1(2)) zum Training verwendet, während der zweite Teil (2(2)) für die Testphase verwendet wurde. Beim Vertauschen der Teile in Lern- und Testphase wurden Unterschiede insbesondere in Bezug auf die Untergrundraten deutlich (s. Tab. 7.1). Die Ursache für diese Unterschiede ist in den unterschiedlichen Graden der Repräsentativität der Datensatzanteile für die Gesamtheit aller möglichen Muster zu suchen. Dieser schwache Effekt konnte nur im Zusammenhang mit den lokalisierten $b\bar{b}$-Ereignissen beobachtet werden, da für diese Klasse von Ereignissen die besten Ergebnisse erzielt wurden.

## 7.3 Diskussion der Ergebnisse

Der Z-Trigger reduziert eine Untergrundrate von $\approx 100\ kHz$ auf zunächst $\approx 10\ kHz$ (Tab. 6.2, S. 67), und ordnet dabei jedem Ereignis ein Z-Vertex-Histogramm und die Zell-Histogramme zu (s. Abb. 6.1, S. 58 und Abb. 6.2, S. 61). Die analytische Bewertung dieser Histogramme liefert eine Untergrundrate von $\approx 200\ Hz$, bei einer Effizienz von $\approx 86\%$ für den getesteten Zerfallskanal der Ereignisse $\gamma g \rightarrow b\bar{b}$, die stellvertretend für physikalisch relevante Ereignisse standen. Die Separation von Untergrund und Physik anhand dieser Histogramme mit einem neuronalen Netzwerk - anstelle einer analytisch formulierten Regel - war Gegenstand der vorliegenden Arbeit.

Nach dem Training mit einem Teil der Histogrammdaten wurden bei der Klassifizierung des anderen Teils der nicht gelernten Histogrammdaten (s. Abb. 6.7, S. 65) Untergrundraten im Bereich von $\approx 100 - 300\ Hz$ bei einer Effizienz von $\approx 99\%$ erreicht (s. Tab. 7.1, S. 75 und Tab. 7.3, S. 76). Beim Lernen mit gesamten Datensätzen (1+2) wurden Untergrundraten $\approx 50\ Hz$ und weniger bei einer Effizienz bis zu 100% erreicht (s. Tab. 7.2, S. 75).

Die mit dem neuronalen Netzwerk erreichten Untergrundraten und Effizienzen liegen in einer Größenordnung, wie sie für einen $2^{nd} - 3^{rd}$ Level Trigger erforderlich sind (s. Tab. 4.1, S. 22). Dabei kann mit hochgradig parallel arbeitenden Neuro-Chips eine Verarbeitungsgeschwindigkeit ($\approx 1\ \mu s$) erreicht werden (s. Tab. 5.2, S. 52), die für festverdrahtete Logik von totzeitfreien $1^{st}$ Level Triggern (s. Tab. 4.1, S. 22) erforderlich ist, verbunden mit einer Untergrundunterdrückung und Effizienz, die konventionell mit $3^{rd}$ Level Triggern nur auf Grundlage von Mikroprozessoren erreicht wird.

### Klassifizierungsgrenze

Der entscheidende Punkt für die mit dem neuronalen Netzwerk prinzipiell erreichbare Separation von Untergrund und Physik ist die durch die Überlappung der verschiedenen Musterklassen bestimmte theoretische Grenze: Das Bayesian Limit (Abschnitt 5.4.5, S. 47, ff.). Auch ein konventioneller Lösungsansatz unterliegt dieser Grenze. Die Verarbeitung der primären Driftkammersignale durch den konventionellen Teil des Z-Triggers führt zu einer geringen Überlappung von Untergrund und Physik, so daß die Separation zu einem hohen Grade möglich ist, wie durch die Simulationen deutlich wurde, bei denen gelernte Ereignissen klassifiziert wurden (s. Tab. 7.2, S. 75). Der Einfluß auf die Überlappung der Musterklassen wird beim Vergleich der Simulationsergebnisse zwischen ausschließlicher Verwendung des Z-Vertex-Histogramms mit den bei der zusätzlichen Verwendung der Zell-Histogramme (Tab. 7.1, S. 75 und Tab. 7.2, S. 75) erzielten Ergebnisse deutlich.

### Lerndaten

Als problematisch stellte sich die Repräsentativität der Lernmuster heraus (s. Tab. 7.1, S. 75), was für die Implementation eines neuronalen Netzwerks im Zusammenhang mit dem Z-Trigger den entscheidenden Faktor darstellen wird. Für eine Implementation werden Monte-Carlo-Daten zur Verfügung stehen müssen, die für reale Ereignisse repräsentativ sind, damit im Experiment mit zu den Simulationsergebnissen vergleichbare Ergebnisse erzielt werden können. Denn Monte-Carlo-Daten, die auf idealisierten Modellen beruhen, die nicht alle Aspekte realer Verhältnisse vollständig berücksichtigen können, weisen sie auch zu einem gewissen Grade einen idealisierten Charakter auf.

### Abstraktionsfähigkeit

Die besondere Eigenschaft neuronaler Netzwerke auf nicht gelernte Muster in angemessener Weise zu reagieren (Generalisation, Abschnitt 5.3.2, S. 37) wurde bei den erreichten Klassifizierungen deutlich, da der überwiegende Teil dieser auf Grundlage von nicht gelernten Ereignissen erfolgte. Die Klassifizierung von gelernten Gesamtdatensätzen diente lediglich der Bestimmung einer oberen Klassifizierungsschranke. Die Abstraktionsfähigkeit des neuronalen Netzwerks wird bei einer technischen Realisierung von großer Bedeutung sein, insbesondere in Bezug auf den idealisierten Charakter von zum Lernen verwendeten Monte-Carlo-Daten.

### Experimentelle Vorteile

Die kontinuierliche Klassifizierung von Ereignissen durch ein neuronales Netzwerk, die ein Maß für die Ähnlichkeit eines vorliegenden Ereignisses zu einer bestimmten Klasse darstellt, erlaubt die einfache Anpassung an experimentelle Gegebenheiten (s. Tab. 7.3, S. 76), indem das endgültige (digitale) Trigger-Signal durch Diskriminierung des Signals des Ausgangsneurons gewonnen wird. Die Variation der Diskriminatorschwelle erlaubt in einem gewissen Bereich die Veränderung der erreichten Untergrundrate (s. Tab. 7.3, S. 76). Zudem kann höheren Trigger-Stufen aufgrund der kontinuierlichen Klassifizierung durch das neuronale Netzwerk bei nicht eindeutig klassifizierten Ereignisse diese Information ergänzend zur Verfügung gestellt werden.

### Technische Realisierbarkeit

Da die Informationsverarbeitung des neuronalen Netzwerks ausschließlich durch die synaptischen Gewichte bestimmt wird (s. Abschnitt 5.4, S. 37, ff.), ist die direkte Übertragung von auf dem Computer bestimmten synaptischen Gewichten durch das Training mit Lernmustern auf ein entsprechendes elektronisch realisiertes neuronales Netzwerk möglich. Durch eine Simulation, die der Auflösung der synaptischen Gewichte verfügbarer Neuro-Chips Rechnung trug konnte gezeigt werden, daß auch die dabei erzielten Ergebnisse mit den vorangegangenen Simulationen im Einklang stehen (s. Tab. 7.4, S. 80).

### Flexibilität

Mögliche Verbesserungen, die bei einer Simulation erzielt werden - sei es durch verbesserte Monte-Carlo-Daten oder die Verfügbarkeit von realen Daten - können daher direkt auf ein bestehendes, elektronisch realisiertes, neuronales Netzwerk übertragen werden. Dazu ist nur die Modifikation der (gespeicherten) synaptischen Gewichte notwendig, ohne irgendwelche Veränderungen an der Schaltung selbst vornehmen zu müssen.

### Ausblick

Der Schaltungsaufwand um ein den durchgeführten Simulationen äquivalentes neuronales Netzwerk zu realisieren ist relativ gering. So würden vier ETANN-Chips genügen um ein entsprechendes neuronales Netzwerk zu implementieren. Die vorliegenden Ergebnisse genügen sicher nicht in vollem Umfang den Anforderungen, die an ein einsatzfähiges System zu stellen sind. Die einfache Übertragbarkeit von verbesserten Simulationsergebnissen - auch während des laufenden Experiments - rechtfertigt jedoch eine elektronische Realisierung eines neuronalen Netzwerks im Zusammenhang mit dem Z-Trigger.

# Kapitel 8

# Zusammenfassung

Im Rahmen der vorliegenden Arbeit wurde durch Computersimulationen die Möglichkeit untersucht, die vom konventionellen Teil des Z-Triggers am H1-Detektor zur Verfügung gestellten Signale mit Hilfe eines neuronalen Netzwerks zu bewerten, um so eine Separation von Untergrundereignissen und physikalisch relevanten Ereignissen zu erreichen.

Dazu wurde ein neuronales Netzwerk mit einem Teil der verfügbaren Daten trainiert, und war anschließend in der Lage, unbekannte - das heißt nicht zum Training verwendete - Ereignisse optimal zu klassifizieren, sofern zum Training eine ausreichende Zahl von repräsentativen Lerndaten zur Verfügung stand. Dabei wurde deutlich, daß die Klassifizierung der Ereignisse anhand der Histogramme des Z-Triggers zu den mit relativ komplizierten Algorithmen durchgeführten Klassifizierungen vergleichbare Ergebnisse lieferte. Die Histogramme der falsch klassifizierten Ereignisse waren in beiden Fällen sehr ähnlich, so daß eine richtige Klassifizierung aufgrund der Überlappung nicht möglich war. Teilweise wurden dieselben Ereignisse sowohl vom neuronalen Netzwerk als auch vom algorithmischen Lösungsansatz falsch klassifiziert, was anhand der Identifikationsnummern der Ereignisse zu sehen war - obwohl beide Lösungsansätze unabhängig voneinander gemacht wurden.

Der mit einem neuronalen Netzwerk durchgeführte Lösungsansatz unterscheidet sich deutlich von dem zunächst vorliegenden konventionellen Lösungsansatz für derartige Aufgaben der Mustererkennung, bei denen explizite Regeln formuliert werden, deren Umsetzung in geeignete Elektronik unter Umständen einen sehr hohen Aufwand bedeutet.

Durch Computersimulation eines neuronalen Netzwerks gewonnene Ergebnisse sind prinzipiell direkt auf ein elektronisch realisiertes neuronales Netzwerk übertragbar, da die Informationsverarbeitung eines neuronalen Netzwerks ausschließlich von in der Lernphase bestimmten Parametern abhängt. Dazu müssen jedoch technische Randbedingungen, die seitens verfügbarer Neuro-Chips vorgegeben sind, berücksichtigt werden. Es konnte jedoch gezeigt werden, daß auch bei Berücksichtigung dieser technischen Randbedingungen mit den vorangegangenen Simulationen vergleichbare Ergebnisse zu erzielen sind.

# Anhang A

# Z-Trigger-Testprogramm ICCP

Mit dem Programm ICCP[1] können Grundfunktionen der Eingangs- und Kombinationslogik getestet werden. Zudem dient es dem Laden der LCA mit den Konfigurationsdaten. Die Ladefunktion ist nicht auf einzelne LCA beschränkt, sondern ermöglicht das automatische Laden aller LCA ($\approx$ 1500) des Z-Triggers mit Hilfe einer Kommandodatei, welche die Ladekommandos - aber auch beliebige andere - als Textzeilen enthält, so daß sie mit jedem beliebigen Text-Editor erstellt werden können oder in einfacher Weise durch ein Programm.

Einige Funktionen dienen der Überprüfung von ICCP selbst, da sich die zu testende Z-Trigger-Elektronik zum Zeitpunkt der Programmerstellung noch in der Entwicklungsphase befand. Zunächst standen nur eine VME-Speicherkarte und eine Bus-Monitor-Karte [VMDIS8003] zur Verfügung, um die Zugriffe auf das VMEbus-System praktisch erproben zu können. Um das Laden der LCA zu testen wurde eine VME-Experimentier-Karte[2] von einem Praktikanten[3] mit den notwendigen Grundbausteinen sowie zwei LCA bestückt. Für einen ersten Test wurde mit dem LCA-Entwicklungssystem ein UND-Gatter mit zwei Eingängen definiert und eine LCA-Konfigurationsdatei generiert, so daß eine einfache Verifikation möglich war.

Erste Versuche, diese Konfigurationsdatei einem LCA zuzuführen scheiterten an der Unzulänglichkeit des zunächst verwendeten RTF-Fortran77 Compilers [RTF68K], Objekt-Code nur für Programme $<$ 32 KByte ausführbarem Code ordnungsgemäß erzeugen zu können; andernfalls liefert der MPW-Linker [RTF77MPW] dahingehende Fehlermeldungen. Um in der Sache voranzukommen wurde auf dem heimischen Atari-ST-Computer ein Programm in Pascal [STPas86] geschrieben, welches nach dem Einlesen der LCA-Konfigurationsdatei die LCA-Ladesequenz generierte und diese Folge von Integer-Zahlen in eine Text-Datei schrieb. Das RTF-Programm, um diese Integer-Zahlen auf den VME-Bus zu übertragen, war dann auch kurz genug um den LCA ordnungsgemäß zu konfigurieren. Da MPW-C [MPWC] in der H1-Gruppe die Standardsprache für Macintosh-Programme ist - und aus Interesse eine neue Programmiersprache zu lernen - wurde nach Rücksprache mit dem geistigen Vater[4] des Z-Triggers die Entscheidung getroffen, ICCP in der Sprache C zu kodieren.

Das Programm ICCP ist in MPW-C geschrieben und läuft unter der MPW-Oberfläche als *Tool* auf Rechnern ab Macintosh II. Die Kommunikation zwischen Programm und VMEbus-System läuft über ein Micron-MACVee-Interface [Micron90], [MacVEE90]. Die Micron-Karte steckt dabei in einem NUBUS-Slot des Macintosh und sind über Flachbahnkabel mit der im VME-Crate steckenden MacVEE-Karte verbunden.

[1] **I**nput **C**ombination **C**ard **P**ulser $\approx$ Impulsgeber für Eingangs- und Kombinationslogik
[2] VME-Experiment-Board, GOEB, W50/5009-00/Vers. 2, 29.9.1989
[3] Röver, J.
[4] Behrend, H.-J.

# A.1 Programmoberfläche

Die Programmoberfläche ist kommandoorientiert, wobei die für die Kommandos verwendeten Kürzel an die in der Beschreibung des Z-Trigger-Elektronik benutzten Bezeichnungen anlehnen [Zim90b], [Zim90c] um Schwierigkeiten zu vermeiden. Zudem ist eine HELP-Funktion implementiert, die nach Eingabe von `HELP` eine Übersicht weiterer SUB-HELP-Funktionen liefert.

## A.1.1 Kommandosyntax

Grundsätzlich beginnt jeder Befehl mit einem Namen, teilweise gefolgt von einem Spezifikationsparameter (#). Eine Wertzuweisung, d.h. eine Schreiboperation, wird durch ein Gleichheitszeichen (=) hinter dem Kommando und darauf folgenden Wert (x) bewirkt. Die zuzuweisenden Werte sind dabei in *hexadezimaler* Form einzugeben. Einige Kommandos dienen sowohl dem Schreiben als auch dem Lesen von Werten. Gelesen wird bei solchen Kommandos einfach durch Eingabe des Befehls, gegebenenfalls mit Spezifikationsparameter ohne nachfolgendes Gleichheitszeichen und Wert. Die Kommandos weisen folgende Form auf

```
kommando # = x * n
```

Für Fehlersuche mit dem Oszilloskop und Tests kann den meisten Kommandos ein optionaler Wiederholungsparameter (*n) angehängt werden. Dieser wird durch ein Multiplikationszeichen spezifiziert und in *dezimaler* Form angegeben. Wird als Wiederholungsparameter null eingegeben, so wird die betreffende Funktion bis zum Betätigen der Esc-Taste ausgeführt. Groß- und Kleinschreibung wird nicht unterschieden, Leerzeichen werden ignoriert. In Tabelle A.1 sind einige Kommandobeispiele aufgeführt.

| | | | | |
|---|---|---|---|---|
| `cr` | : | card reset | : | Die Karte wird initialisiert. |
| `il 3.2` | : | init LCA 3.2 | : | LCA 3.2 wird initialisiert. |
| `src 1.1 *1000` | : | shift register clock 1.1 | : | Signal wird 1000 mal wiederholt. |
| `cc *0` | : | card clock | : | Das Signal wird bis zum Drücken der Esc-Taste wiederholt. |
| `mc =3f` | : | memory counter | : | Adresszähler wird auf den hexadezimalen Wert 3f gesetzt. |
| `mc` | : | memory counter | : | Ausgabe des Adresszählers |
| `loadlca 2.2 = test.mcs` | : | load lca 2.2 | : | LCA 2.2 mit den Konfigurationsdaten der Datei test.mcs geladen. |
| `chkrdm` | : | check raw data memory | : | Testet den Rohdatenspeicher auf Lese- und Schreiboperationen, als auch auf kurzgeschlossene Daten- oder Adressleitungen. |

Tabelle A.1: **Beispiele einiger Kommandos** *für eine Eingangskarte. Durch Eingabe von HELP erhält man eine Liste weiterer HELP-Funktionen für alle implementierten Kommandos.*

Die Kommandos beziehen sich immer auf eine Karte. Jede Karte wird durch eine Modul-Nummer sowie eine Crate-Nummer spezifiziert. Nach dem Programmstart sind diese beiden Nummern zunächst auf den Wert Null voreingestellt. Andere Einstellungen können durch Eingabe der betreffenden Systemkommandos gesetzt werden.

### Kommandoausführung

Auf eine graphische Benutzeroberfläche wurde verzichtet, da ein lauffähiges Programm schnell für die Tests der ersten Prototypen verfügbar sein sollte. Außerdem erfordert die Möglichkeit Kommandodateien ausführen zu können einen Kommandointerpreter. Da die Ausführung der Kommandos innerhalb von ICCP durch den Aufruf der Routine

```
execom(char command[]);
```

erfolgt, ist der nachträgliche Aufsatz einer graphischen Oberfläche möglich, ohne daß dabei die gesamte Struktur von ICCP verändert werden muß. Die Variable `command` steht für die eine ein bestimmtes Kommando spezifizierende Zeichenkette, so wie sie manuell oder über eine Kommandodatei eingegeben wird. Die Ausgaben von ICCP, die über `printf();` erfolgen, müßten jedoch auf eine graphische Oberfläche umgelenkt werden.

### Kommandodatei

Eine Folge von Befehlen kann in einer Kommandodatei zusammengefaßt werden. Die Kommandos werden wie im interaktiven Betrieb eingegeben. Aus einer Kommandodatei können keine weiteren Kommandodateien aufgerufen werden. Aufgerufen wird eine Kommandodatei durch Eingabe von

```
@dateiname.
```

Kommandodateien können mit jedem ASCII-Text-Editor - auch unter MPW - erstellt werden. So kann die Kommandosequenz zum Laden der LCA einer Karte sowie der gesamten Trigger-Elektronik bequem durch eine Kommandodatei erfolgen. Eine Kommandodatei namens `STARTUP` im gleichen Verzeichnis wie ICCP wird beim Programmstart automatisch ausgeführt.

### Help-Funktionen

Die Eingabe des Kommandos `HELP` liefert eine einleitende Übersicht über die verschiedenen HELP-Funktionen (s. Tab A.2). Da eine HELP-Funktion nur den Inhalt einer ihr zugeordneten Datei ausgibt, können Modifikationen an ICCP in den HELP-Dateien leicht angepaßt werden. Die wichtigen Systemfunktionen werden durch Eingabe von `HELP SYS` auf dem Bildschirm ausgegeben (s. Tab A.3).

| | | |
|---|---|---|
| `type ic` | : | Selektiert die Input Card |
| `type cc` | : | Selektiert die Combination Card |
| `help sys` | : | Parameter - Zuweisungen |
| `help ic` | : | Grundkommandos der Input Card (IC) |
| `help cc` | : | Grundkommandos der Combination Card (CC) |
| `help ic test` | : | Kommandos der IC für automatische Tests |
| `help cc test` | : | Kommandos der CC für automatische Tests |

Tabelle A.2: **Sub-Help-Funktion** *nach Eingabe von HELP.*

| | | |
|---|---|---|
| `micron = x` | : | Slot-Adresse der Micron-Karte setzen |
| `crate = x` | : | Crate-Nummer setzen |
| `modul = x` | : | Modul-Adresse setzen |
| | | |
| `type ic` | : | Typ Eingangskarte (prompt: `ic >`) |
| `type cc` | : | Typ Kombinationskarte (prompt: `cc >`) |
| | | |
| `LoadLCA n = name.MCS` | : | Konfiguriert LCA n mit datei = name.MCS |
| `LoadLCA n = name.BIN` | : | Konfiguriert LCA n mit datei = name.BIN |
| `conv2bin = datei.MCS` | : | Konvertiert datei.MCS in die Datei.BIN |
| `dumpfile = name.*` | : | Gibt Inhalt von datei.* aus |
| | | |
| `@datei` | : | Führt Kommandosequenz in datei aus |
| | | |
| `test off` | : | Realer Zugriff auf VMEbus |
| `test on` | : | Simulierter Zugriff auf VMEbus |
| `rdword = x` | : | Setzt Wert x für simulierten Zugriff |
| `rdword` | : | Liest Wert (x) bei simuliertem Zugriff |
| `offset = x` | : | Verschiebt VME-Basisadresse, für Tests |
| `offset` | : | Zeigt VME-Basisadresse an |
| | | |
| `exit` | : | Programm ICCP verlassen. |

Tabelle A.3: **Systemfunktionen** *Um mit einer Karte zu kommunizieren müssen Micron-, Modul- und Crate-Adresse spezifiziert werden. Im Testmodus werden die betreffenden VME-Zugriffe nicht ausgeführt, es werden dann nur die Aktionen auf dem Bildschirm ausgegeben.*

### Automatische Testfunktionen

Neben den Kommandos zur Überprüfung elementarer Funktionen und Fehlersuche - VME-Zugriff, Kurzschlüsse zwischen Daten- oder Adressleitungen - sind einige Kommandos implementiert, die automatisch Tests mit LCA durchführen, sofern diese konfiguriert sind. So können etwa die Primärschieberegister der Eingangskarten auf ihre Funktionstüchtigkeit hin überprüft werden. Derartige Tests sind jedoch immer nur in Zusammenhang mit geeigneten LCA-Konfigurationsdaten möglich und erfordern unter Umständen die Erweiterung von ICCP. Wie für die Grundfunktionen von ICCP sind auch für die automatischen Testfunktionen HELP-Funktionen implementiert.

### Beenden des Programms

Das Testprogramm ICCP sollte grundsätzlich nur durch Eingabe von `EXIT` beendet werden, da beim Start einige Systemfehlerbehandlungsvektoren (680x0-Traps [Mot86]) verändert werden und der alte Zustand nur durch reguläres Verlassen des Programms wiederhergestellt wird. Durch die Verwaltung dieser Vektoren durch ICCP wird ein Programmabbruch aufgrund von defekten Modulen verhindert; es wird gegebenenfalls nur eine entsprechende Fehlermeldung ausgegeben.

## A.2 Logic Cell Array

Die LCA sind vergleichbar mit sogenannten Gate-Arrays, die vorfabrizierte anwenderspezifische Schaltkreise darstellen. Bei Gate-Arrays erfolgt der letzte Herstellungsschritt, die Verdrahtung der Logik-Gatter, beim Hersteller. Beim LCA erfolgt die "Verdrahtung" durch Laden mit Konfigurationsdaten, so daß diese auch nachträglich modifizierbar ist. Die zur Konfiguration der LCA notwendigen Daten werden mit Hilfe des zu diesen gehörenden Entwicklungssystems der Firma *Xilinx* generiert [Xilinx89], [Xilinx90].

### A.2.1 Das LCA-Entwicklungssystem

Das LCA-Entwicklungssystem läuft unter MS-DOS auf PC und auf Unix-Workstations. Ist der Entwurf eines LCA abgeschlossen, so muß der die LCA-Konfiguration bestimmende Datenstrom mit Programmen des LCA-Entwicklungssystems erzeugt werden (s. Abb. A.1), da die LCA-Konfigurationsdaten zunächst in einem internen Format (*.LCA) vorliegen. Das System unterstützt verschiedene Dateiformate (MCS, EXO, TEK). ICCP verarbeitet zur Zeit jedoch nur das MCS-Format und ein binäres Datenformat[5].

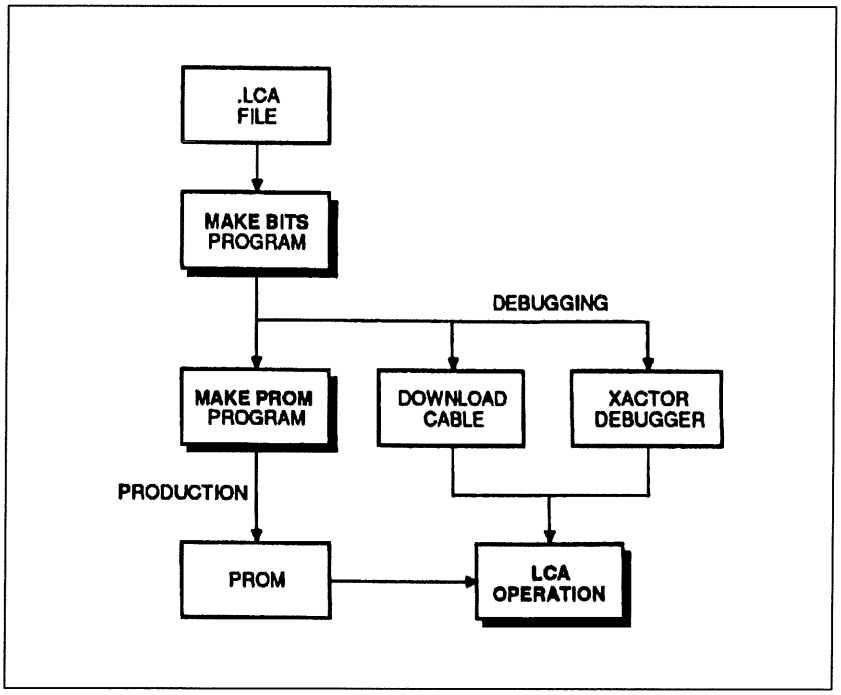


Abbildung A.1: **Generierung der LCA-Daten** *[Xilinx89]. Aus der Datei, welche die Verschaltung eines LCA beschreibt (*.LCA), wird mit dem Programm MAKEBITS ein "Bitstrom" generiert, der als Eingabe für MAKEPROM dient, das daraus eine PROM-Datei generiert.*

[5]Handelsübliche Programmiergeräte für PROM beherrschen in der Regel ebenfalls dieses binäre Format

### MCS-PROM-Datei

Für das Testprogramm ICCP wurde das MCS-Format gewählt. Die Abbildung A.2 einen Ausschnitt aus einer PROM-Datei im MCS-Format. Diese **muß** mit der UP-Option generiert werden, denn die ersten vier Daten-Bytes - diese werden auch als erste an den betreffenden LCA übertragen - geben die Anzahl der gesamten Konfigurations-Bytes eines oder einer Kette (Master-Slaves) von LCA an und werden vom (Master-)LCA gezählt um die Konfiguration zu koordinieren.

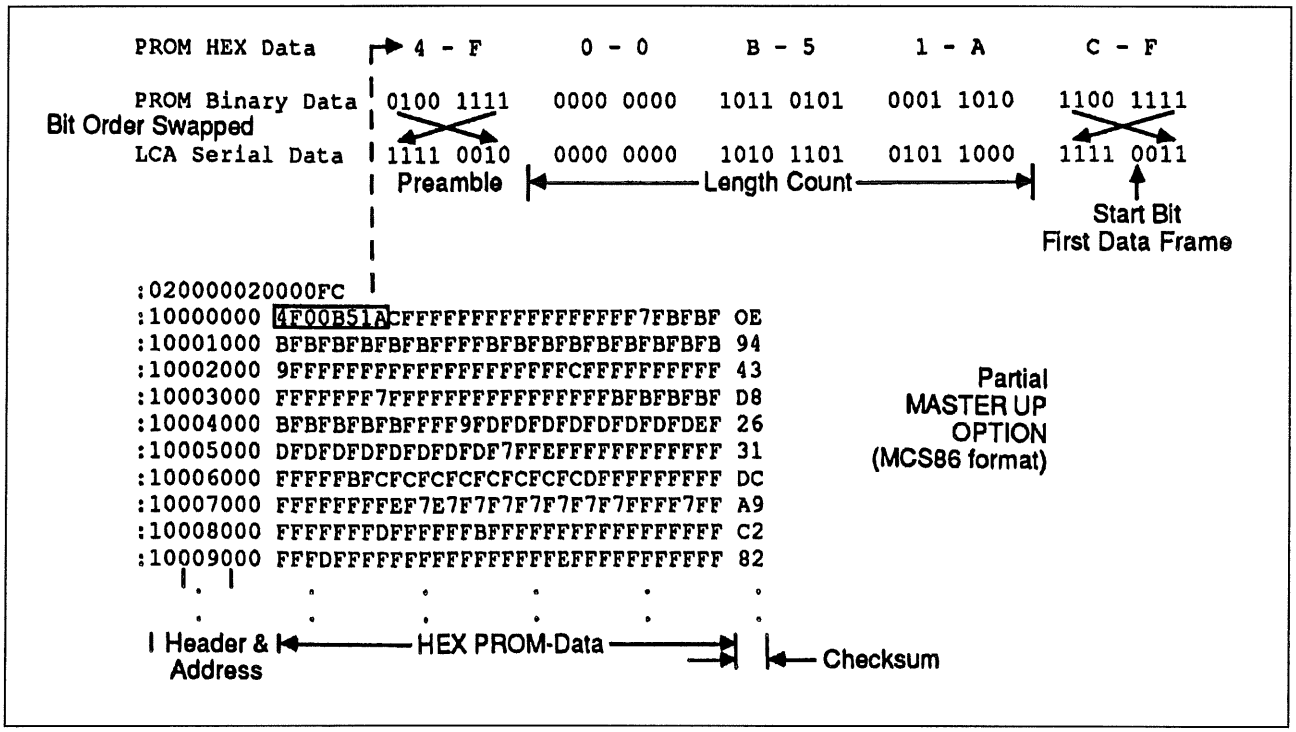


Abbildung A.2: **LCA-Konfigurationsdatei** *im MCS-Format [Xilinx90]. Die ersten vier Daten-Bytes werden zur Koordination der Konfiguration ausgewertet.*

### BIN-Datei

Da das Interpretieren der (Text-)PROM-Datei eine relativ lange Zeit in Anspruch nimmt, wurde in ICCP eine Programmfunktion implementiert, welche die Konvertierung von PROM-Dateien im MCS-Format in ein äquivalentes Binär-Format (*.BIN) durchführt. Beim Laden der LCA auf Grundlage der Binär-Dateien, wird damit die Ladegeschwindigkeit um einen Faktor $\approx 15$ gesteigert. Beim Laden aller LCA des Z-Triggers wird die Ladezeit dann von *ca.* 60 *Min.* auf *ca.* 4 *Min.* verkürzt. Da die MCS-Datei neben den eigentlichen LCA-Konfigurationsdaten zusätzliche Informationen - für die Programmierung von PROM - enthält, die für das Laden der LCA nicht von Bedeutung sind, ist die Datenmenge bei gleichem Informationsgehalt für die Binär-Dateien um einen Faktor $\approx$ 3 geringer. Das ist von Bedeutung, wenn man bedenkt, daß eine LCA-PROM-Datei eine Länge von $\approx$ 100 *KByte* (für eine von vielen Ketten) aufweisen kann[6].

[6]Das MCS-Format läßt eine maximale PROM-Größe von nur 64 *KByte* zu

### Konversionsprogramm für LCA-Konfigurationsdaten

Dieser Programmteil wurde als eigenständiges Programm aus ICCP extrahiert und so modifiziert, daß es durch Änderung von nur einer Programmzeile an praktisch jeden Computer angepaßt werden kann[7]. Das Konversionsprogramm CONV2BIN wurde auf einem Atari-ST unter Turbo-C [TCST90] modifiziert und getestet. Nach Änderung einer Definitionszeile lief es dann auch ordnungsgemäß auf einem Macintosh unter MPW.

Da CONV2BIN leicht auf jeden Rechner portierbar ist, kann zukünftig auf den Rechnern, auf denen die LCA "verdrahtet" werden, gleich anschließend die Konversion in eine Binär-Datei erfolgen, wodurch sich der benötigte Speicherbedarf reduzieren läßt.

## A.3 LCA-Konfiguration

Für das Laden der Konfigurationsdaten bietet der LCA mehrere Möglichkeiten an. Bei den Karten des Z-Triggers wird jedoch nur von dem *peripheral master mode* Gebrauch gemacht. Bei diesem Modus werden die Konfigurationsdaten, nachdem auf den LCA ein Initialisierungsimpuls gegeben wurde, einfach byte-weise auf den betreffenden LCA geschrieben.

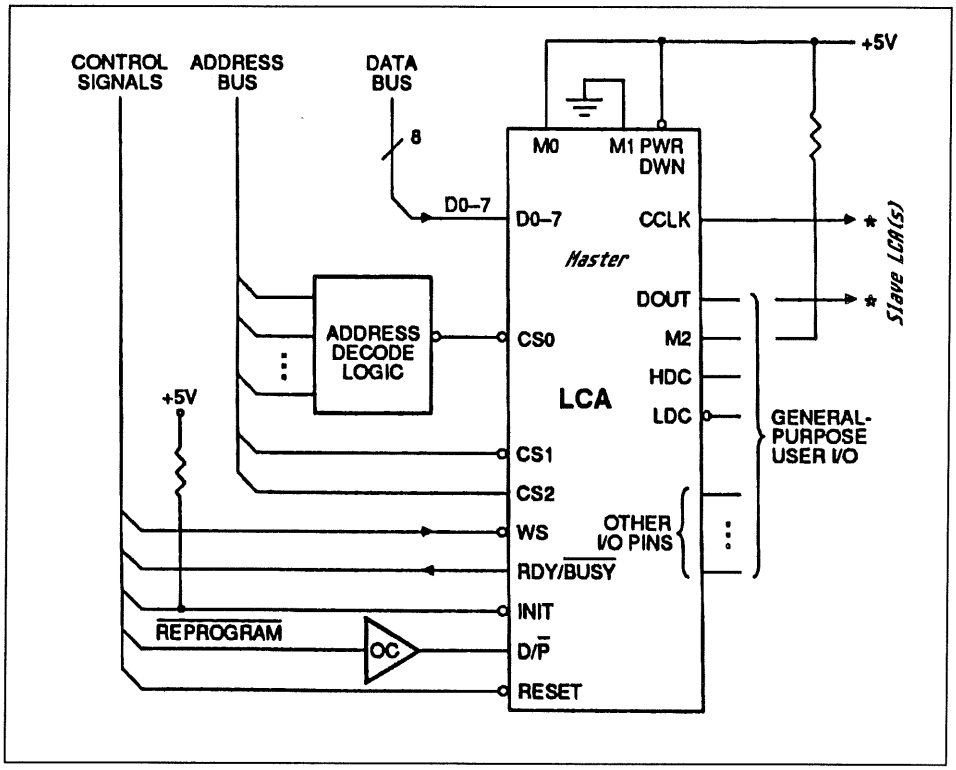


Abbildung A.3: **Peripheral Mode** *[Xilinx90]. Der erste (Master-)LCA einer Kette reicht die Konfigurationsdaten bitseriell an die nachfolgenden LCA (Slaves) weiter.*

[7]Die Sprache C erlaubt die bedingte Compilierung [KeRi83] in Abhängigkeit von Definitionen (`# define`)

Sind LCA in Ketten zusammengefaßt, so empfängt der erste LCA der Kette die Konfigurationsdaten - für die gesamte Kette - und reicht die Daten *bitseriell* an die nachfolgenden LCA weiter (Abb. A.3). Damit dieses ordnungsgemäß funktioniert, muß der Anordnung der Kette bei der Generierung der PROM-Datei mit MAKEPROM Rechnung getragen werden. Liegen die Konfigurationsdaten im richtigen Format vor, so ist die Durchführung der Konfiguration vergleichsweise einfach: Die Daten-Bytes der betreffenden Konfigurationsdatei werden einfach auf die spezielle LCA-Adresse geschrieben.

Die Tests der Prototypen wurden von einem Ingenieur[8] durchgeführt. Nur durch intensive Gespräche war die Pflege von ICCP möglich und konnten Fehler in ICCP und in Prototypen lokalisiert werden. Die Komplexität der Z-Trigger-Elektronik erfordert ein hohes Maß an Team-Arbeit. Die Erweiterung von ICCP wurde von einem anderen Ingenieur[9] inzwischen übernommen. Die Entwicklung einer graphischen Oberfläche für ICCP wurde ebenfalls durch einen DESY-Mitarbeiter übernommen[10].

[8] Klär, H.
[9] Reumann, M.
[10] Franke, G.

# Literaturverzeichnis


[Ack85] [Ackley, Hinton, Sejnowski] *A learning algorithm for Boltzmann machines*, (Cognitive Science 9:147-169 (1985)), in [AndRo88], S. 635, ff.

[AndJ88] [Anderson, J. R.] *Kognitive Psychologie: Eine Einführung*, (Spectrum der Wissenschaft-Verlagsgesellschaft 1988, ISBN 3-922508-19-7)

[AndRo88] [Herausgeber: Anderson, J. A., Rosenfeld, E.] *Neurocomputing: Foundations of Research*, (MIT Press 1988, ISBN 0-262-01097-6)

[Barb87] [Barbagli, G., et al.] *Tagging of* $\gamma \rightarrow q\bar{q}$ *Events and Direct Measurement of the Gluon Structur Function at HERA* (In [HERA87a], S. 135, ff.)

[Bart83] [Barto, Sutton, Anderson] *Neurolike adaptive elements that can solve difficult learning control problems*, (IEEE Transactions on Systems, Man and Cybernetics SMC-13:834-846 (1983)), in [AndRo88], S. 535, ff.

[Beh90] [Behrend, H.-J., Schröder, V., Stephens, R.] *Z-Chamber Trigger*, (Internal Report DESY, H1-TR310-Feb. 90, February 2, 1990)

[Beh91] [Behrend, H.-J., Zimmermann, W.] *A Hardwired Trigger Processor using "Logic Cell Arrays" (XILINX)*, (Internal Report DESY, 6 March 1991)

[BOS87] [Blobel, V.] *The BOS-System, Dynamic memory management, FORTRAN 77 Version*, (Internal Report DESY, December 10, 1987)

[Bron81] [Bronstein, Semendjajew] *Taschenbuch der Mathematik & Ergänzungsband*, (Verlag Harri Deutsch 1981, ISBN 3-87144-492-8 & 3-87144-493-6)

[Coo87] [Cooper-Sarkar, Ingelman, Long, Roberts, Saxon] *Measurement of the Longitudinal Structur Function and the small x Gluon Density of the Proton*, (In [HERA87a], S. 231, ff.)

[ct12/89] [Herausgeber: Ch. Heise, Autor: Janosch, B.] *Optische Neurocomputer*, (Verlag Heinz Heise GmbH & Co KG 1989, *c't* - magazin für computer technik), Heft 12, S. 38 ff.

[ct5/91] [Herausgeber: Ch. Heise, Autor: Zerbe, K.] *Die Neuro-Chips kommen*, (Verlag Heinz Heise GmbH & Co KG 1991, *c't* - magazin für computer technik), Heft 5, S. 182 ff.

[Dal91] [Herausgeber: Master International] *Ausstellungskatalog zur Dalí-Ausstellung in Venedig, April 1991*, (Master International s.r.l. Milano, Italia 1991)

[Den89] [Denby, B.] *Neural Networks and Cellular Automata Algorithms*, (FSU-SCRI-88-141, June 1989, Florida State University)

[DenLinn90] [Denby, B., Linn, S.] *Status of HEP Neural NET Research in the U.S.A.*, (FERMILAB-Conf-90/21, Fermi National Accelerator Laboratory, January 1990)

[DenLess90] [Denby, B., Lessner, E.] *Tests of Track Segment and Vertex Finding with Neural Networks*, (FERMILAB-Conf-90/68, Fermi National Accelerator Laboratory, April 1990)

[Desy81] [Herausgeber: Deutsches Elektronen Synchroton DESY] *Wissenschaftlicher Jahresbericht 1981*, (DESY, 1981)

[DudHa73] [Duda, R.O. and Hart, P.E.] *Pattern Classification and Scene Analysis*, (John Wiley & Sons, Inc. 1973, ISBN 0-471-22361-1)

[Fis84] [Fischer, G.] *Lineare Algebra*, (Vieweg 1984)

[Ftp91] *Dateitransfer mit FTP-Protokoll*, (DESY 1991, IBM-Großrechenanlage (HELP))

[Fuku83] [Fukushima, Miyake, Ito] *Neocognitation: a neural network model for a mechanism, of visual pattern recognition*, (IEEE Transactions on Systems, Man, and Cybernetics SMC-13:826-834 (1983)), in [AndRo88], S. 526, ff.

[FVS77] [Herausgeber: DESY] *VS FORTRAN 77 Language Reference, IBM Fortran 77*, (Internal Report DESY, Februar 1988)

[GepV4.7] [Bassler, E.] *GEP - Graphical Editor Programm for User Data / Version 4.7* (Internal Report DESY R02-81/02, September 1988)

[Glü87] [Glück, M.] *Heavy Flavor Contributions to Structur Functions and Scaling Violation*, (In [HERA87a], S. 119, ff.)

[GuN80] [Herausgeber: Spektrum der Wissenschaft Verlagsgesellschaft] *Gehirn und Nervensystem*, (Spectrum der Wissenschaft-Verlagsgesellschaft 1980, ISBN 3-922508-21-9)

[Hebb49] [Hebb, Donald, O.] *The Organisation of Behavior*,
(New York: Wiley, Introduction and Chapter 4, The first stage of perception: growth of the assembly (1949), pp. xi-xix, 60-78), in [AndRo88], S. 43, ff.

[Hei90] [Heidrich, K., Kießling, R., Maluschka, R.] *TEX-Shell, Version 3.63*,
(Heidrich, Kießling, Maluschka, 15. Dezember 1990)

[HERA87a] [Herausgeber: DESY] *Proceedings of the HERA Workshop, Vol 1*,
(DESY, October 12-14, 1987)

[HERA87b] [Herausgeber: DESY] *Proceedings of the HERA Workshop, Vol 2*,
(DESY, October 12-14, 1987)

[Hop82] [Hopfield, J. J.] *Neural networks and physical systems with emergent collective computational abilities*,
(Proceedings of the National Academy of Sciences 79:2554-2558 (1982))
in [AndRo88], S. 460, ff.

[Ing87a] [Ingelman, G., et al] *Deep Inelastic Physics and Simulation*,
(In [HERA87a], S. 3, ff.)

[Ing87b] [Ingelman, G., et al] *Separation of Deep Inelastic Charged and Neutral Current Events*,
(In [HERA87a], S. 19, ff.)

[Intel86] *Memory Components Handbook*,
(Intel Corporation, 1986)

[Intel90a] *80170NW, Electrically Trainable Analog Neural Network*,
(Intel Corporation, May 1990)

[Intel90b] *80170NX changes from the 80170NW, Advance Notice*,
(Intel Corporation, November 1990)

[Intel90c] [Roy, M., Holler, M., Tam, S., Castro, H.] *Testing an Analog VLSI Neural Network with 10240 "Floating Gate Synapses*,
(1990 IEEE VLSI TEST Symposium, Intel Corporation, April 1990)

[Jor86] [Jordan, M. I.] *Proceedings of the Eighth Conference of the Cognitive Science Society*,
(Hillsdale, NJ: Lawrence Erlbaum Associates, ©1986 by M. I. Jordan),
in [McCeRum88], S. 156, ff.

[KeRi83] [Kernigan, Ritchie] *Programmieren in C,*
*mit dem Reference Manual in deutscher Sprache*,
(Carl Hansa Verlag München Wien 1983, ISBN 3-446-13878-1)

[Ker91] *Dateitransfer mit Kermit-Protokoll*,
(DESY 1991, IBM-Großrechenanlage (HELP))

[Kop91a] [Kopka, H.] LATEX, *Eine Einführung*,
(Addison-Wesley 1991, ISBN 3-89319-338-3)

[Kop91b] [Kopka, H.] LaTeX, *Erweiterungsmöglichkeiten, mit einer Einführung in Metafont*, (Addison-Wesley 1991, ISBN 3-89319-356-1)

[Kre87] [Krehbiel, H.] *Ein Fußweg zu* LaTeX, (Interner Bericht DESY, 13. September 1987)

[Lag87] [Lagemann, K.] *Rechnerstrukturen* , (Springer Verlag Berlin Heidelberg 1987, ISBN 3-540-17618-7)

[Lar91] [Larsson, L.] *ICCP-Programmbeschreibung / Version 1.02*, (DESY, 31.1.1991, unveröffentlicht)

[Leo87] [Leo, W. R.] *Techniques for Nuclear and Particle Physics Experiments*, (Springer-Verlag Berlin Heidelberg 1987, ISBN 0-387-17386-2)

[Lin91] [Lindner, S.] *TeXauf dem Atari ST/TT, eine kleine Einführung zur Version 2.0*, (Stefan Lindner, Januar 1991)

[Loh83] [Lohrmann, E.] *Einführung in die Elementarteilchenphysik*, (Teubner-Studienbücher: Physik 1983, ISBN 3-519-03055-1)

[Loh86] [Lohrmann,E.] *Hochenergiephysik*, (Teubner-Studienbücher: Physik 1986, ISBN 3-519-23043-7)

[Look109] [Levonian, S.] *EVLOOK - General Purpose H1 Event Display, V. 1.09*, (DESY 22/04/91, IBM-Großrechenanlage)

[LUTP90-6] [Peterson, C.] *Neural Networks and High Energy Physics*, (LU TP 90-6, May 1990, University Lund, Sweden)

[LUTP90-8] [Lönnblad, L., Peterson, C., Rögnvaldsson, T.] *Using Neural Networks to Identify Jets*, (LU TP 90-8, May 1990, University Lund, Sweden)

[MacVEE90] [Taylor, B. G.] *The MacVEE Hardware User Manual, including MacVEEPlus, MacVEE SE and MAC-CC*, (ECP Division, CERN European Laboratory for Particle Physics, CH-1211 Geneva 23, Switzerland, 1990)

[McCeRum86a] [Rumelhart, McCelland] *Parallel Distributed Processing, Volume 1: Foundations*, (MIT Press 1986, ISBN 0-262-18120)

[McCeRum86b] [McCelland, Rumelhart] *Parallel Distributed Processing, Volume 2: Psychological and Biological Modells*, (MIT Press 1986, ISBN 0-262-13218)

[McCeRum88] [McCelland, Rumelhart] *Explorations in Parallel Distributed Processing*, (Including source code of discussed neural networks) (MIT Press 1988, ISBN 0-262-63113-X)

[MfNN90] [Goser, K., Ramacher, U., Rückert, U.] *Proceedings of the 1st International Workshop on MICROELECTRONICS FOR NEURAL NETWORKS*, (University of Dortmund 1990, ISBN 3-89227-021-X)

[Micron90] [Taylor, B. G.] *The MICRON User Manual, MacVEE Interface for the Macintosh II Family*, (ECP Division, CERN European Laboratory for Particle Physics, CH-1211 Geneva 23, Switzerland, 1990)

[MinPa69] [Minsky, Papert] *Perceptrons*, (MIT Press 1969, Third printing 1988, ISBN 0-262-6311-3)

[Mot86] [Herausgeber: MOTOROLA Semiconductors] *M68000 16/32-Bit Microprocessor, Programmer's Reference Manual*, (Motorola Inc. 1986)

[MPWC] [Herausgeber: Apple Computers] *Micro Programmers Workshop C*, (Apple Computers Inc., 1988)

[Net87] [Netter, F. H.] *Nervensystem I, Neuroanatomie und Physiologie*, (Georg Thieme Verlag 1987, ISBN 3-13-524401-6)

[Neu58] [von Neumann, J.] *The Computer and the Brain*, (New Haven: Yale University Press, pp. 66-82 (1958)), in [AndRo88], S. 83, ff.

[New88] [Butenschön, H., Sommer, H.-P.] *Newlib Reference Manual*, (Internal Report DESY R-88/01)

[NNW90] [Schöneburg, Hansen, Gawelczyk] *Neuronale Netzwerke (mit DOS-Diskette)*, (Markt & Technik 1990, ISBN 3-89090-329-0)

[Oxf90] *Intelligent Pattern Recognition Memory Module*, (Oxford Computer Inc., August 1990)

[Per87] [Perkins, D. H.] *Introduction to High Energy Physics*, (Addison-Wesley Publishing Company 1987, ISBN 0-201-12105-0)

[Phy0488] [Herausgeber: Particle Data Group] *PHYSICAL LETTERS B - Vol. 204 (1988) pp. 1-486, Review of Particle Properties*, (Particle Data Group, North-Holland Amsterdam, 14 April 1988)

[RepST91] [TradeIT] *REPRO STUDIO junior 2.0 ST Handbuch, Digitale Reprographie und ScannSoftware*, (TradeIT, 1991)

[Ros58] [Rosenblatt, F.] *The perceptron: a probabilistic model for information storage and organisation in the brain*, (Psychological Review 65:386-408 (1958)), in [AndRo88], S. 89, ff.

[RTF68K] [H. von der Schmidt] *Real-Time Fortran 77 for 68K Processors, Manual of Compiler and Run-Time Library with Appendix for OS-9*, (Physikalisches Institut Heidelberg, May 25, 1988)

[RTF77MPW] [Campell, A.] *RTF-Fortran-77 for Macintosh Programmer's Workshop*, (Internal Report DESY, H1-06/88-8P, Rev. October 1988, Version 1988)

[Schm88] [Schmüser, P.] *Feynman-Graphen und Eichtheorien für Experimentalphysiker*, (Springer-Verlag Berlin Heidelberg 1988, ISBN 3-540-18797-9)

[Schul91] [Schulten, Martinez, Thomas] *Neuronale Netzwerke, Eine Einführung in die Neuroinformatik selbstorganisierender Netzwerke*, (Addison-Wesley 1991, ISBN 3-89319-131-3)

[Stim90] [Stimpfl-Abele[(a)]*, G., Garrido[(b)]*, L.] *Fast Track Finding with Neural Nets*, ((a) Department of Physics Florida State University, USA, (b) Laboratori de Fisica d'Altes Energies Universitat Autonoma de Barcelona Spain, * Now at CERN, August 1990)

[Stir87] [Stirling, W. J.] *Photoproduction of Large Momentum Jets at HERA*, (In [HERA87a], S. 331, ff.)

[STPas86] [Herausgeber: CCD] *Handbuch zum ST Pascal plus 2.0, für alle Atari ST-Computer, entwickelt von J. Lohse*, (Creative Computer Design, D-6228 Etville, ©1986/87)

[Stry90] [Stryer, L.] *Biochemie*, (Heidelberg: Spektrum-der-Wissenschaft-Verlagsgesellschaft 1990, ISBN 3-89330-690-0)

[TCST90] [Herausgeber: Borland] *Handbücher zum Borland Turbo C 2.0 für den Atari ST*, (Borland GmbH, München, Juni 1990)

[TeXSpl91] [Larsson, L.] *TeXSpell Version 0.9, Ein Spell-Check-Programm für TEX- und ASCII-Quelltexte*, (Hamburg, 18.9.1991, unveröffentlicht)

[Tka87] [Tkaczyk, S. M., Stirling, W. J., Saxon, D. H.] *Inclusive J/Ψ Production and Measurement of the low-x Gluon Distribution of the Proton*, (In [HERA87a], S. 265, ff.)

[TPH86] *Technical Proposal for the H1 Detector*, (DESY, March 25, 1986)

[TPR89] *Technical Progress Report 1989, H1 Collaboration*, (DESY, August 24, 1989)

[TPR90] *Technical Progress Report 1990, H1 Collaboration*, (DESY, August 27, 1990)

[Val83] [Herausgeber: Valvo Unternehmensbereich Bauelemente der Philips GmbH] *Der 16bit-Mikroprozessor SC 68000, Eigenschaften*

[VMDIS8003] [Herausgeber: Creative Electronic Systems S. A.] *VME Display, VMEDIS 8002/3/3A, User's Manual*, (Creative Electronics Systems S. A., CH-1213 Petit-Lancy/Switzerland)

[VME85] *VMEbus Specification Manual, Revision C.1,*
(Also known as: ICE 821 BUS and IEEE P1014/D1.2, October 1985)

[VME86] *ELEKTRONIK Sonderheft Nr. 229, Der VMEbus,*
*Ein Bussystem für 16/32-Bit-Mikroprozessoren,*
(Franzis-Verlag GmbH, München, 1986)

[Weh85] [Wehnes, H.] *FORTRAN 77,*
*Strukturierte Programmierung mit FORTRAN 77,*
(Carl Hansa Verlag München Wien 1985, ISBN 3-446-14259-2)

[WidHof60] [Widrow, B. and M. E. Hoff] *Adaptive switching circuits,*
(1960 IRE WESCON Convention Record, New York: IRE, pp. 96-104),
in [AndRo88], S. 123, ff.

[Won88] [Wonneberger, R.] *Kompaktführer* LaTeX,
(Addison-Wesley 1988, ISBN 3-89319-152-6)

[Xilinx89] *The Programmable Gate Array Data Book,*
(Xilinx 1989)

[Xilinx90] *LCA Programmable Gate Array User's Guide,*
*Design Implementation Application Note,*
*Configuring Xilinx Logic Cell*$^{tm}$ *Arrays,*
(Xilinx 1990)

[Zim90a] [Zimmermann, W.] *Technical Proposal, Central Z-Trigger,*
*COZ + CIZ, H1,*
(DESY, 22.6.90, unpublished)

[Zim90b] [Zimmermann, W.] *Input Card, Central Z-Trigger,*
*H1, Layout-Nr. 5253, 1. Prototype,*
(DESY FH1T, Status 4.9.90, unpublished)

[Zim90c] [Zimmermann, W.] *Combination Card, Z-Trigger, H1*
*Layout-Nr. 5266-00, 1. Prototype,*
(DESY FH1T, Status 22.11.90, unpublished)

# Danksagung

Ich danke Herrn Prof. Blobel, der diese Arbeit ermöglicht hat.

Herrn Dr. Behrend danke ich für die ausgezeichnete Betreuung, hilfreiche Vorschläge und die ständige Bereitschaft auch in längeren Gesprächen wichtige Fragen zu klären. Außerdem danke ich ihm für das Einverständnis zum heimischen Zusammenschreiben, wodurch ich viel Fahrzeit gespart habe.

Herrn Dr. Struck danke ich für den Hinweis auf die Diplomarbeit in der H1-Gruppe. Herrn Dr. Kiesling möchte ich besonders für die Empfehlung eines in das Themengebiet der neuronalen Netzwerke einleitenden Buches danken, welches mir die Einarbeitung in dieses Gebiet sehr erleichtert hat. Herrn Dr. Grindhammer und Herrn Dr. de Roeck danke ich für die Unterstützung bei den ersten Versuchen mit dem Programmpaket Jetnet der Universität Lund. Dem DESY-Rechenzentrum und der DESY-Bibliothek danke ich für die engagierte Unterstützung. Frau Härtel und Frau Kleinebenne danke ich für die prompte Unterstützung in allen organisatorischen Fragen und bei der Bereitstellung von Arbeitsmitteln.

Ich danke Martha-Louise Lessing und Jens Scholz für die angenehme, freundschaftliche Atmosphäre während der gemeinsam Laborzeit. Jens Röver danke ich für die Hilfe bei ersten Gehversuchen mit C und für die akkurate Bestückung einer LCA-Testplatine. Dirk Düllmann danke ich für die Beantwortung jeder Frage im Zusammenhang mit dem Macintosh. Herrn Dr. Schröder und Herrn Zimmermann danke ich für hilfreiche Diskussionen. Guy Cases und Dirk Michelsen danke ich für technische Hilfestellungen bei Problemen mit dem IBM-Großrechner. Christian Markus danke ich für die rasche Unterstützung beim Problem mit dem DVI-Treiber nach einer beinahe schlaflosen Nacht. Kristina Rieger danke ich für ihre fröhlich sonnige Ausstrahlung, die dadurch zu der sehr angenehmen Arbeitsatmosphäre während des Zusammenschreibens beitrug.

Herrn Dr. Schumacher danke ich für seine Mühe und konstruktive Kritik an ersten Entwürfen. Herrn Trinks danke ich für das schnelle und genaue Korrekturlesen zum Ende der Arbeit. Doris Schröder, Lothar Albaum und Michael Reumann danke ich ebenfalls für das Korrekturlesen der Arbeit.

Allen Freundinnen und Freunden danke ich für das Verständnis, daß ich insbesondere zum Ende der Arbeit nur sehr wenig Zeit für private Aktivitäten hatte.

Ganz besonders danke ich meinen Eltern, die mir das Physikstudium ermöglicht haben.

# Erklärung

Hiermit erkläre ich, daß ich diese Diplomarbeit selbst verfaßt habe und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Hamburg, den

Lars Larsson