Separation von e^- und p^+ mit Hilfe von Neuronalen Netzen im TRD von AMS-02

Philip von Doetinchem

Phys. I B, RWTH-Aachen

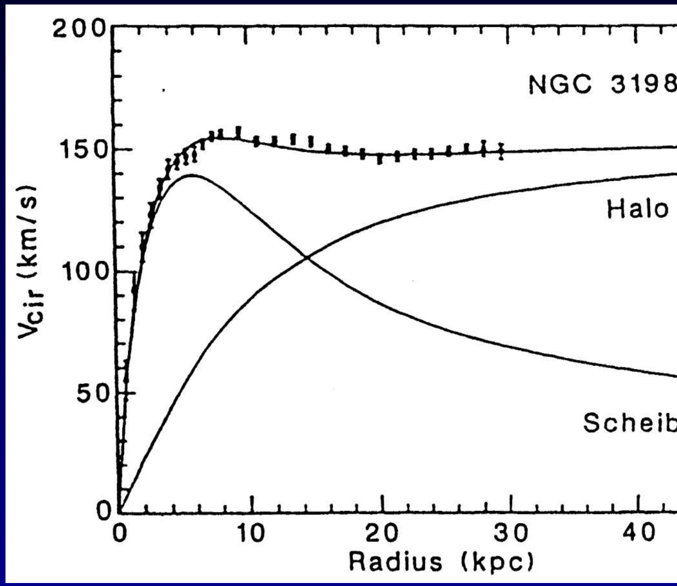
Vortragsstruktur

1. Dunkle Materie
2. AMS-02-Experiment
3. Übergangsstrahlung
4. TRD-Prototyp
5. Klassische Analyse
6. Einführung in Neuronale Netze
7. Anwendung
8. Zusammenfassung und Ausblick

Dunkle Materie

- ⋆ Dunkle Materie: Warum?

$$v \propto \begin{cases} r & r < R \\ r^{-\frac{1}{2}} & r > R \end{cases} \qquad (1)$$



ABER:

$$v(r) \approx \text{const. für } r > R \quad (2)$$

- ⋆ Galaxientstehung: Top-Down-Theorie

Superhaufen ⟶ Galaxiehaufen ⟶ Galaxien
heiße Dunkle Materie

Bottom-Up-Theorie:

Kugelsternhaufen ⟶ Galaxien ⟶ Galaxiehaufen
kalte Dunkle Materie

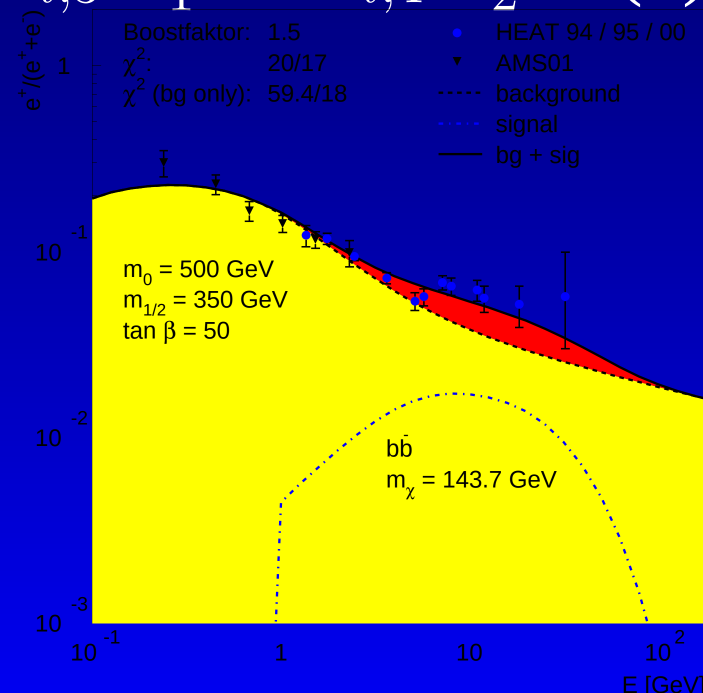
Nicht-Baryonische Dunkle Materie

- verschiedene Kandidaten für Dunkle Materie

  heiß: Neutrinos mit Masse
  kalt: Neutralinos in gebrochener, supersymmetrischer, R-Parität-erhaltender Theorie

$$\chi_i^0 = N_{i,1}\tilde{B} + N_{i,2}\tilde{W}^3 + N_{i,3}\tilde{H}_1^0 + N_{i,4}\tilde{H}_2^0 \qquad (3)$$

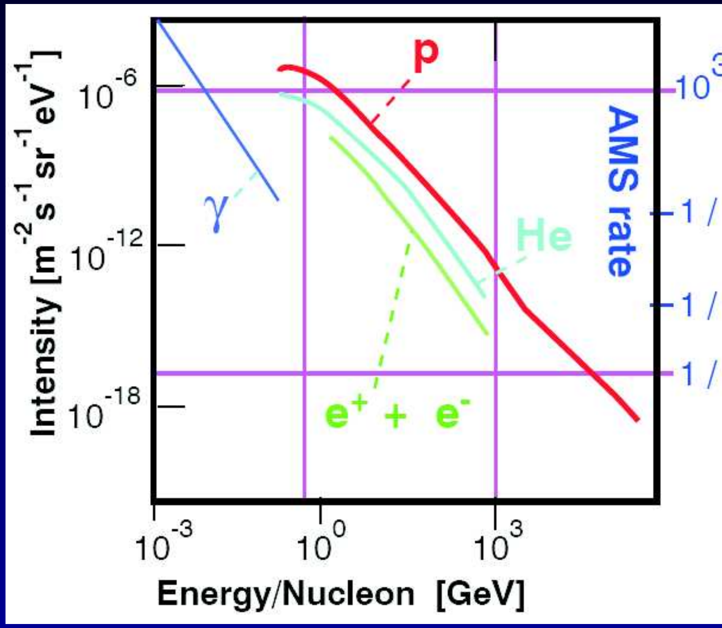
- Messung z.B. über Annihilation von Neutralinos in W^-W^+, Z^0Z^0, $b\bar{b}$, $t\bar{t}$ durch Überhöhung in Antiteilchenspektren



Anforderungen

⋆ Unterschied der Flüsse von Protonen und Positronen im Weltall:

$$\frac{p^+}{e^+} = 10^4 \qquad (4)$$

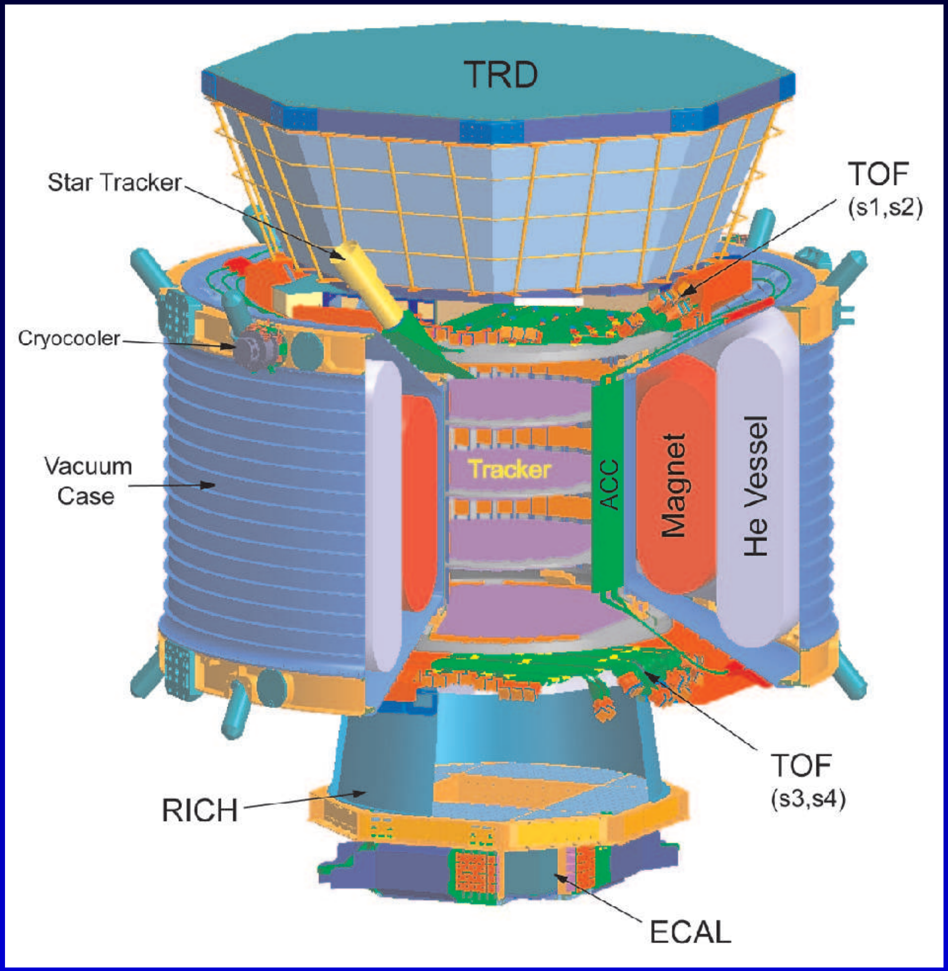


$\Rightarrow$ Trennung von p^+ und e^+ muss ca. 10^6 sein.

⋆ Elektroneffizienz: richtig erkannte Elektronen (Hier: 90 %)

⋆ Der Kehrwert des Anteils der Protonen, die über dieser Schwelle liegen: **Rejection**

AMS-02 Detektor



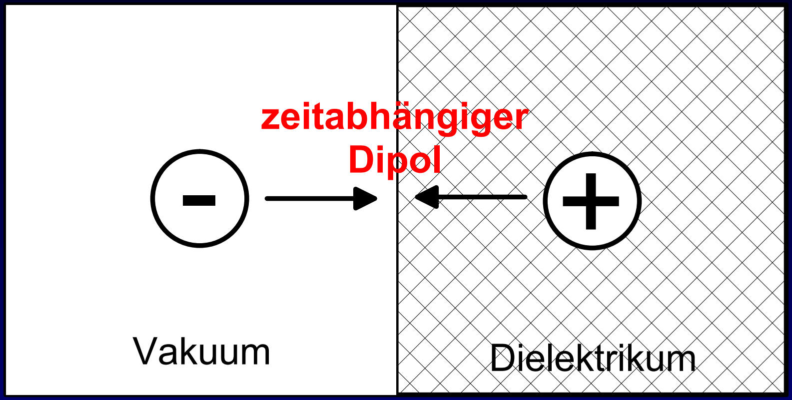
- ⋆ 3 Jahre Betrieb auf der ISS zur Spektroskopie der Höhenstrahlung
- ⋆ TRD:
  Rejection: 10^2 - 10^3
- ⋆ ECAL:
  Rejection: 10^2 - 10^3

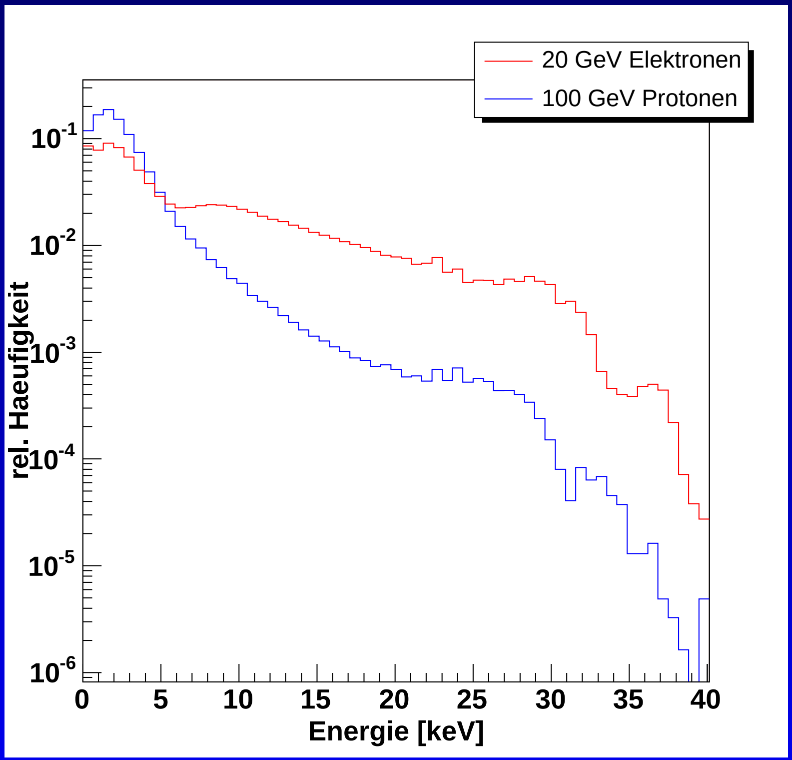
Übergangsstrahlung

- ⋆ Stetigkeit der em-Felder an Grenzflächen sorgt für Strahlung
- ⋆ Wahrscheinlichkeit der Übergangsstrahlung (ladungsunabhängig):

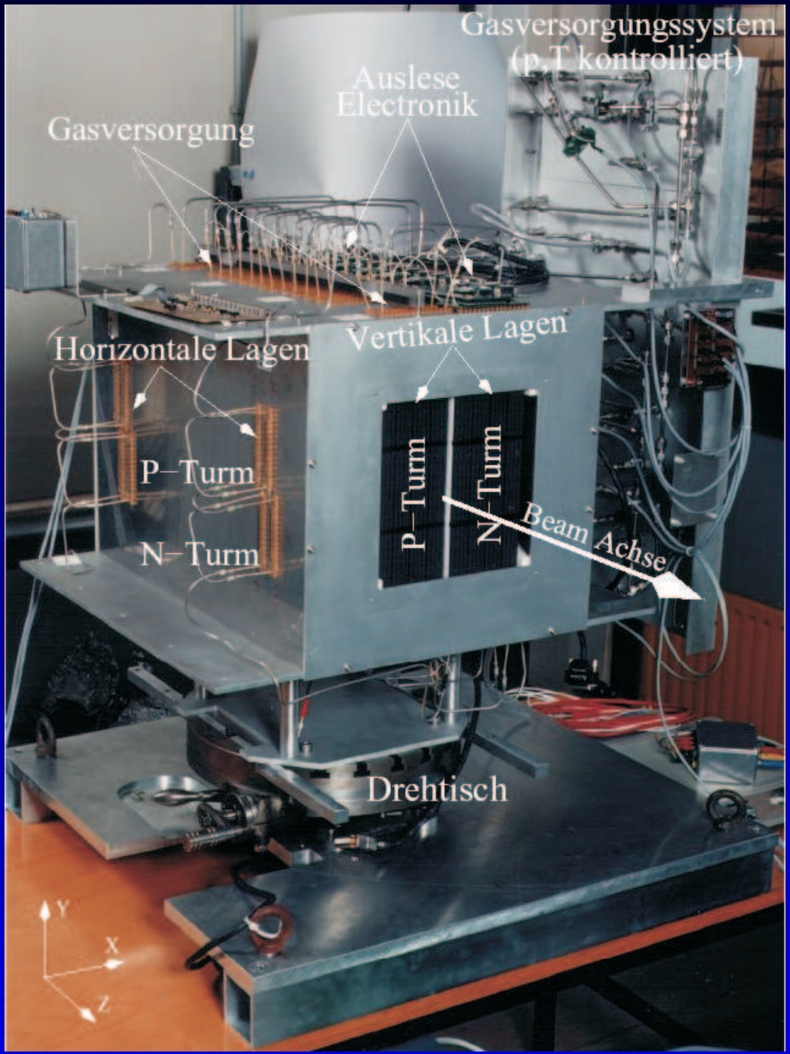
$$W \propto \gamma = \frac{E}{m} \qquad (5)$$

- ⋆ unterschiedlicher Verlauf der Energiedepositionen für Elektronen und Protonen dient zur Unterscheidung





20-lagiger TRD-Prototyp



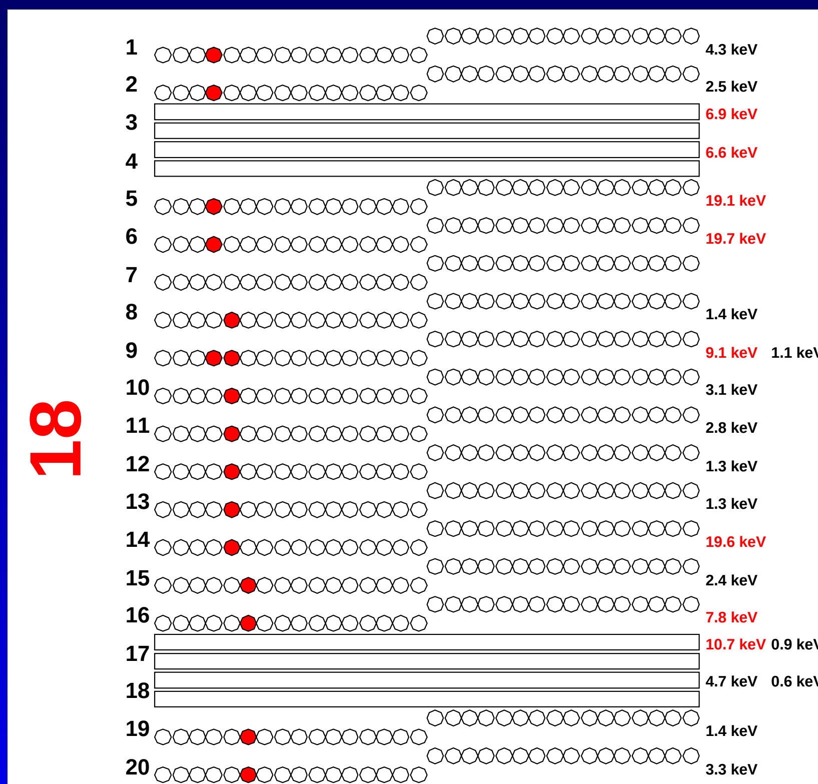
- ⋆ 20 Lagen, damit die Wahrscheinlichkeit der Übergangsstrahlung steigt
- ⋆ Zwischen den Lagen: Radiatormaterial
- ⋆ Auswertung von Beamtest aus dem Jahr 2000

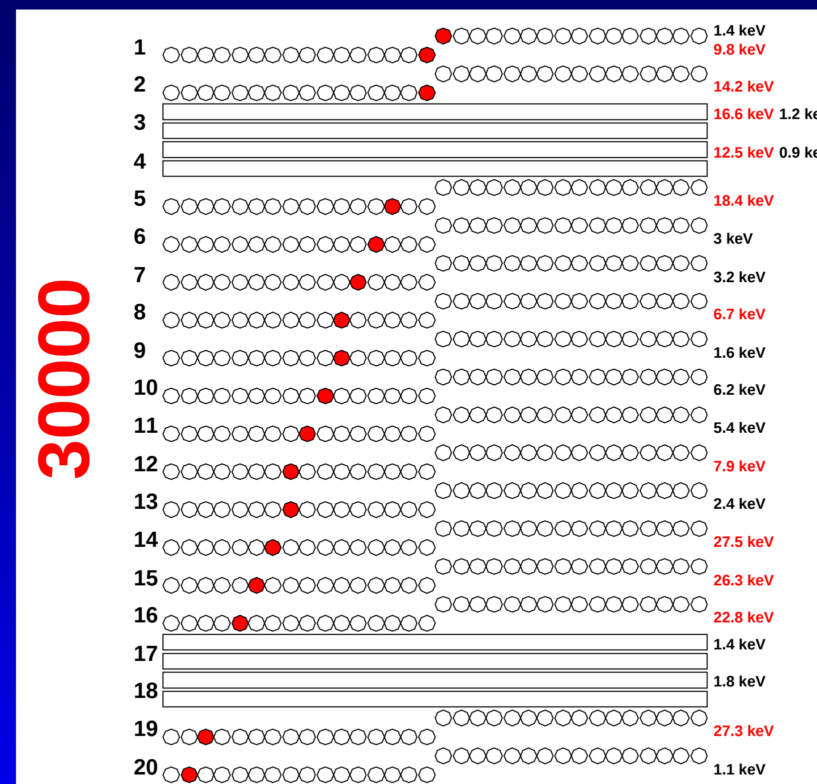
  e^-: 20 GeV

  p^+: 20, 40, 60, 80, 100, 120, 140, 160, 180, 200, 250 GeV

Spurfit

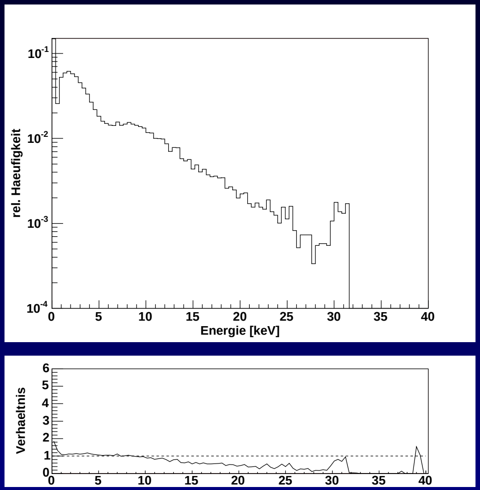
- ⋆ Verwendung des Spurfits aus der Diss. von J. Orboeck
- ⋆ Ermittelung einer Energiedeposition pro Lage für die Analyse:
  1. Nur Hits AUF der Spur
  2. 2 Hits in einem Turm auf einer Lage: Addition
  3. 2 Hits in Grenzröhrchen auf einer Lage: Arithmetisches Mittel



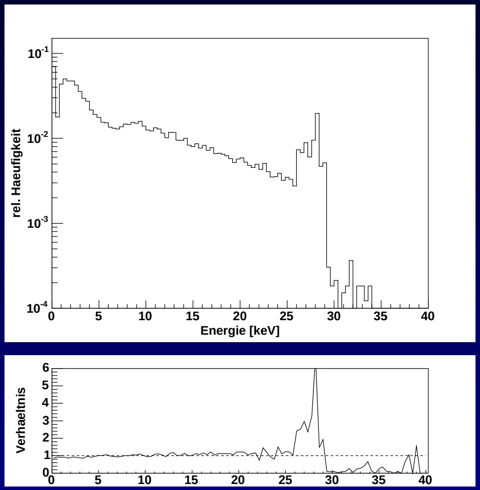


Abweichung von mittlerer Energie

Elektronen:

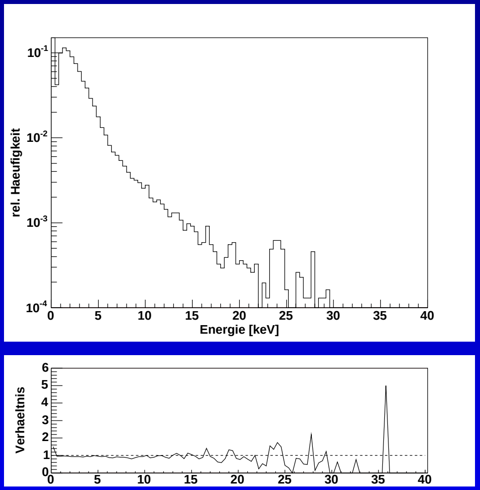


Lage 1

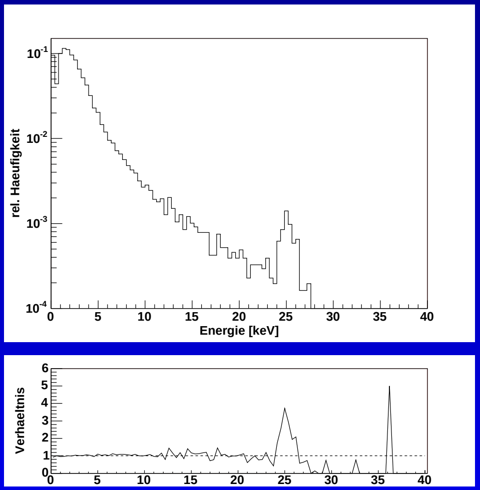


Lage 20

Protonen:

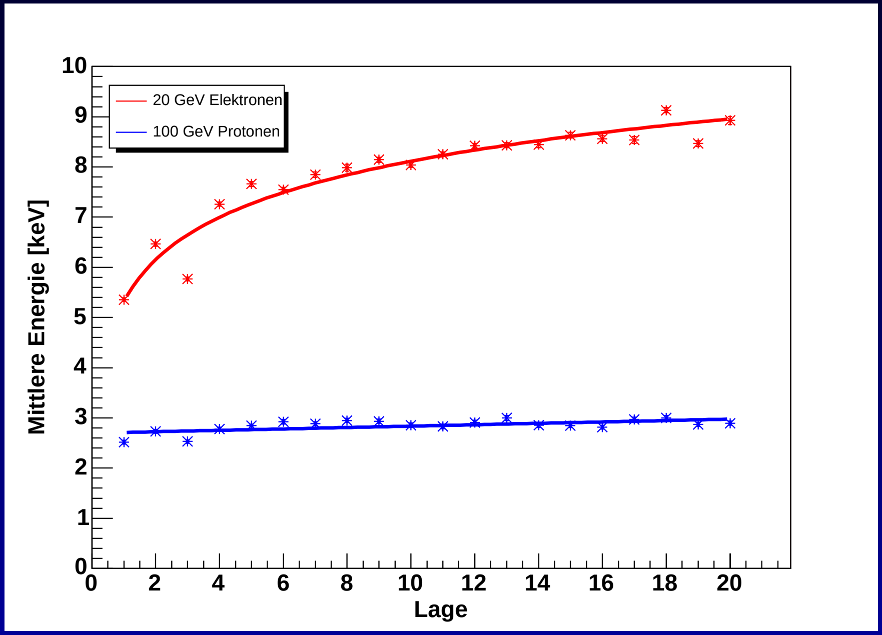


Lage 1

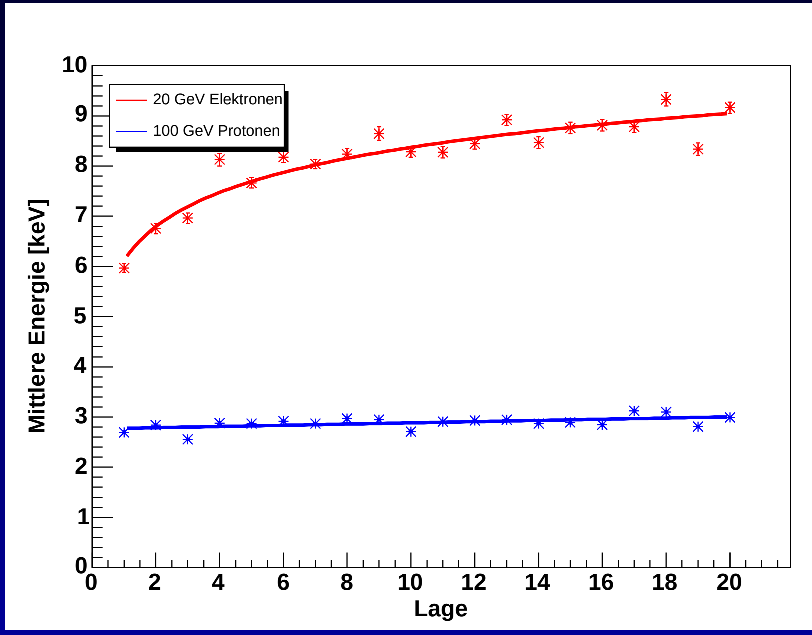


Lage 20

Mittlere Energie pro Lage



Winkel: -1.5°



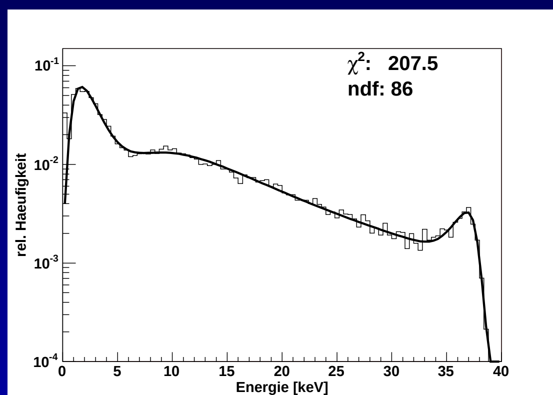
Winkel: 9.3° /8.3°

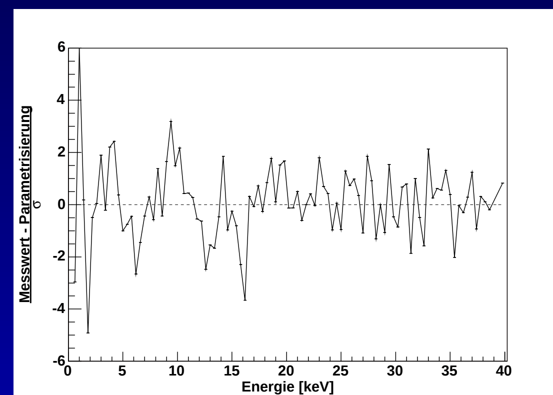
- ⋆ Mittlere Energie wächst, da der WQ klein ist für die Detektion der hochenergetischen TR-Photonen.
- ⋆ Teilchen durchquert je nach Winkel mehr oder weniger Radiatormaterial vor den gedrehten Lagen.

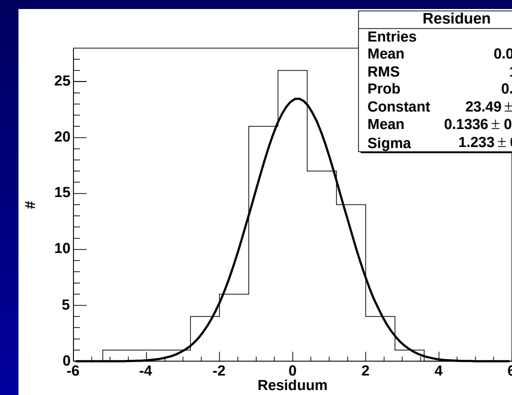
Fit der Energieverteilungen

$$f(x) = [p_0 \cdot \mathrm{Landau}(x, p_1, p_2) + p_3 \cdot \mathrm{Landau}(x, p_4, p_5) + p_6 \cdot \mathrm{Gaus}(x, p_7, p_8)] \cdot \frac{1}{1 + e^{p_9 \cdot (x - p_1}}$$

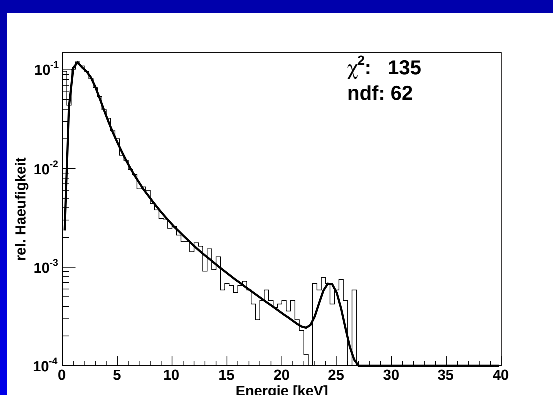
Elektronen

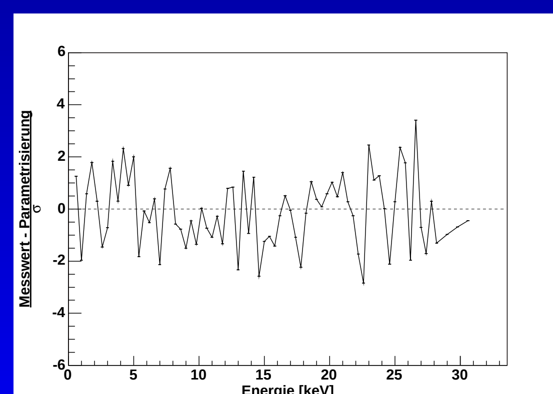


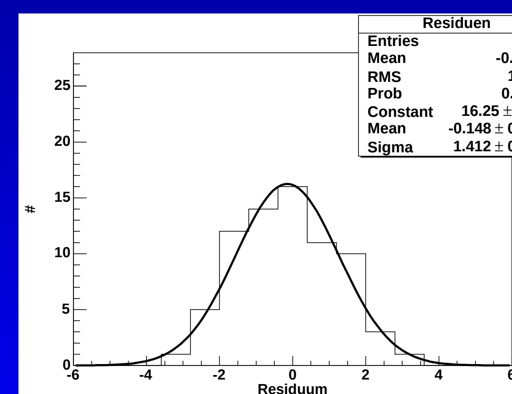




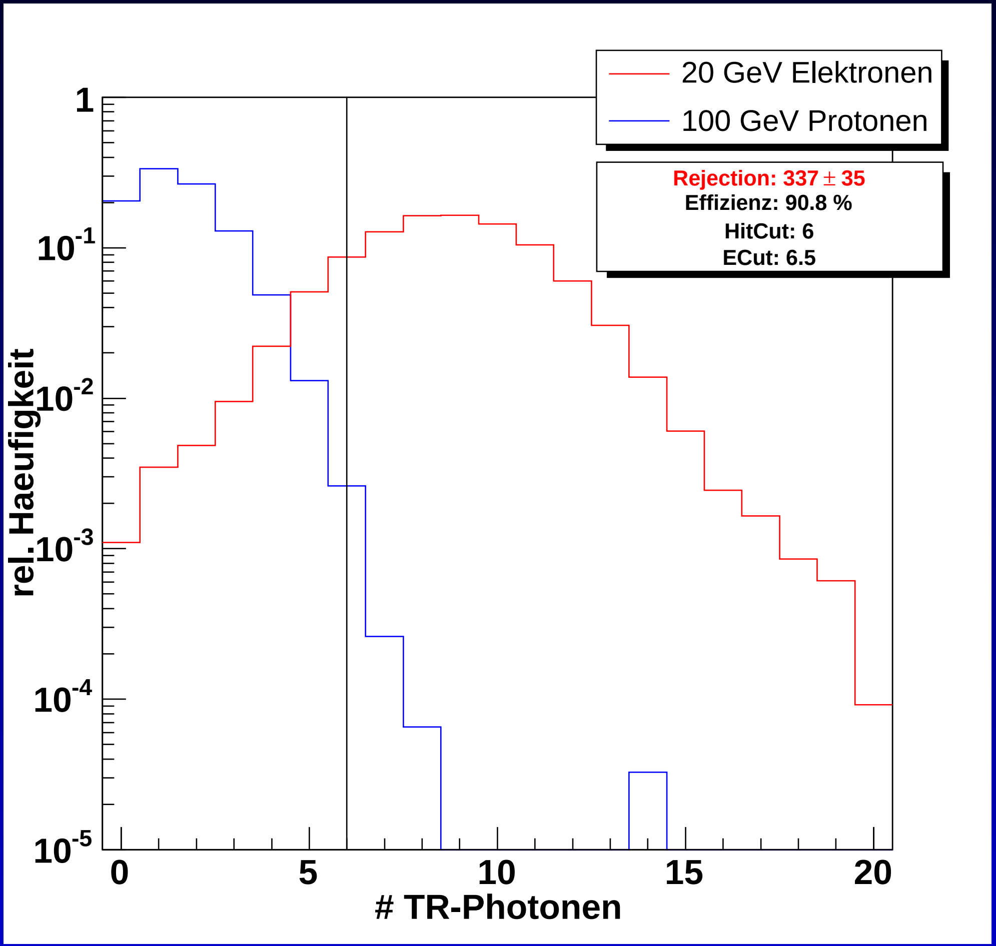
Protonen





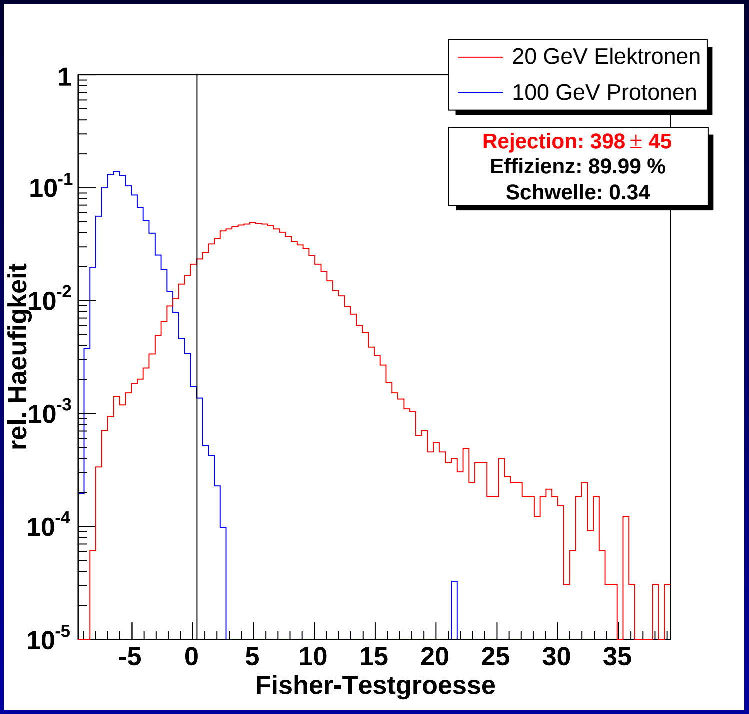


Cluster-Counting



Es wird die Anzahl der hohen Energieeinträge pro Event bestimmt und als Kriterium zur Klassifikation verwendet.

Fisher-Diskriminante



⋆ Schätzung der mittleren Kovarianzmatrix

$$V_{mk}^{(i)} = \frac{1}{N} \sum_{N} (X_m^{(i)} - \bar{X}_m^{(i)})(X_k^{(i)} - \bar{X}_k^{(i)}) \quad ($$

$$V_{mk} = \frac{1}{2}(V_{mk}^{(1)} + V_{mk}^{(2)}) \quad ($$

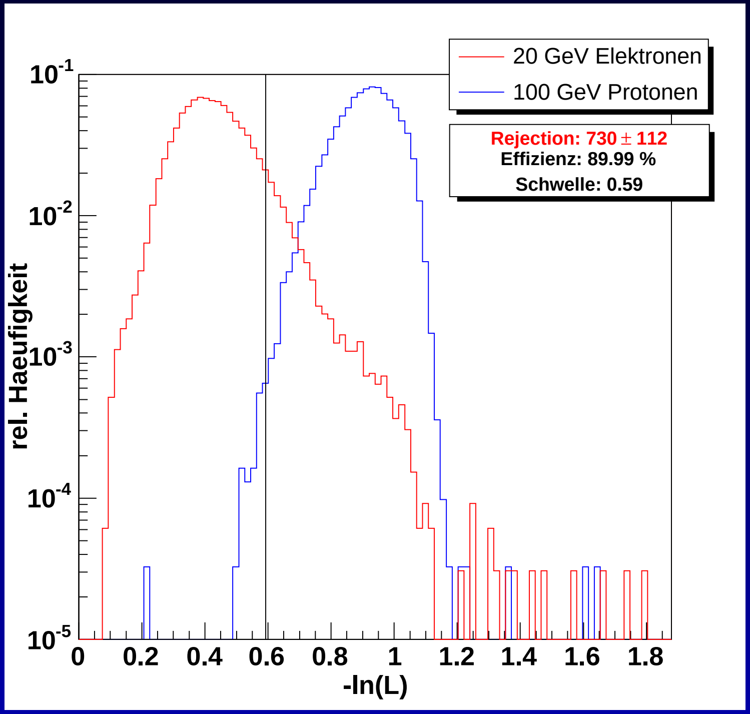
⋆ Fisher Algorithmus um zwei Punktmengen zu trennen:

$$t = \sum_{i=1}^{n} f_i X_i - \frac{1}{2} \sum_{i=1}^{n} f_i (\bar{X}_i^{(1)} + \bar{X}_i^{(2)}) \quad (9)$$

$$f_i = \sum_{k} (V^{-1})_{ik} (\bar{X}_k^{(1)} - \bar{X}_k^{(2)}) \quad (10)$$

⋆ Es kann gezeigt werden, dass der Algorithmus bei Normalverteilung optimal ist.

Likelihood-Methode



- ⋆ Ermittelung des mittleren LH-Wertes pro Event

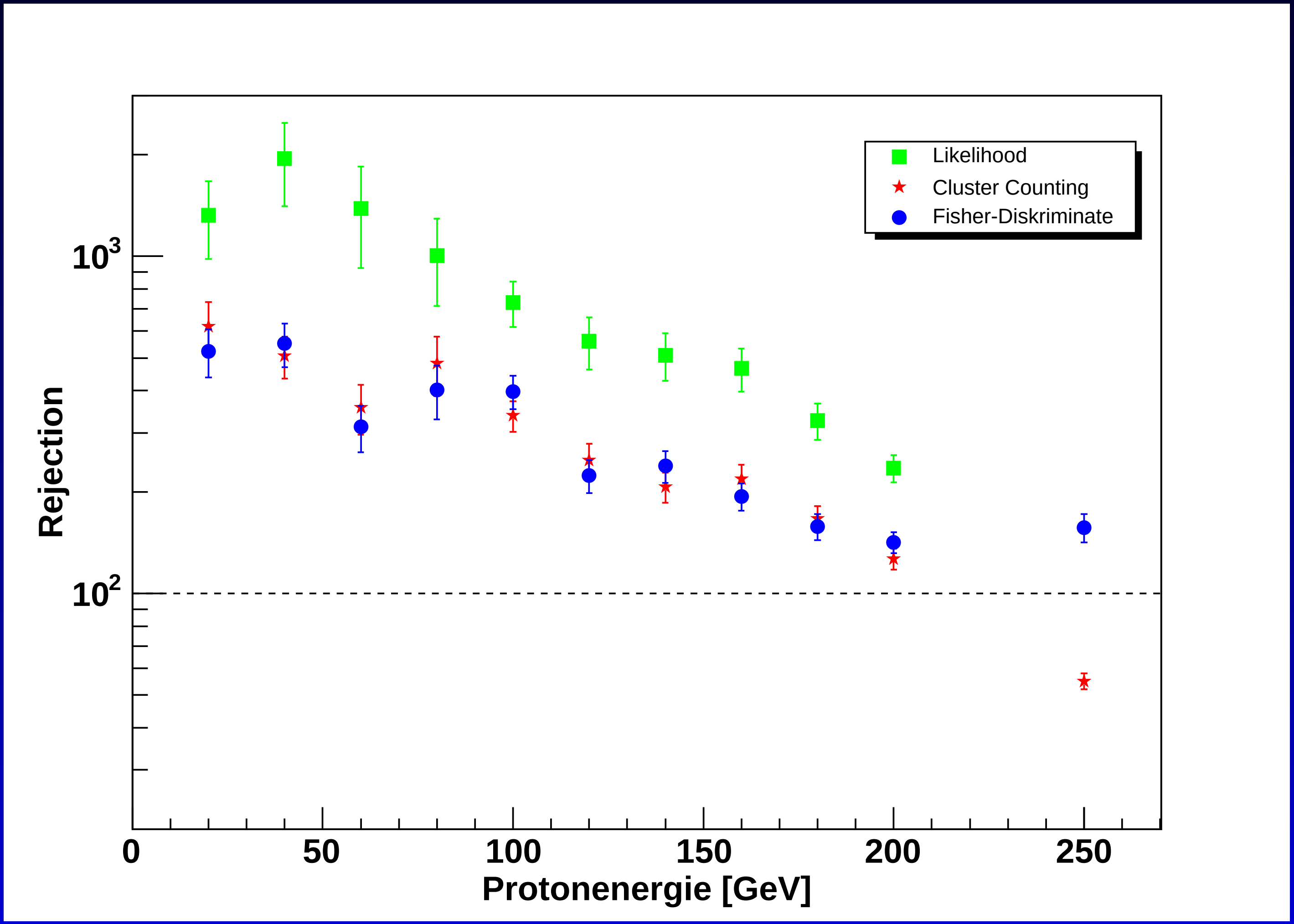
$$\bar{P}_{\mathrm{e/p}} = \sqrt[n]{\prod_{i}^{n} P^{(i)}_{\mathrm{e/p}}(E)} \qquad (11$$

- ⋆ Likelihood-Ratio:

$$L = \frac{\bar{P}_{\mathrm{e}}}{\bar{P}_{\mathrm{p}} + \bar{P}_{\mathrm{e}}} \qquad (12$$

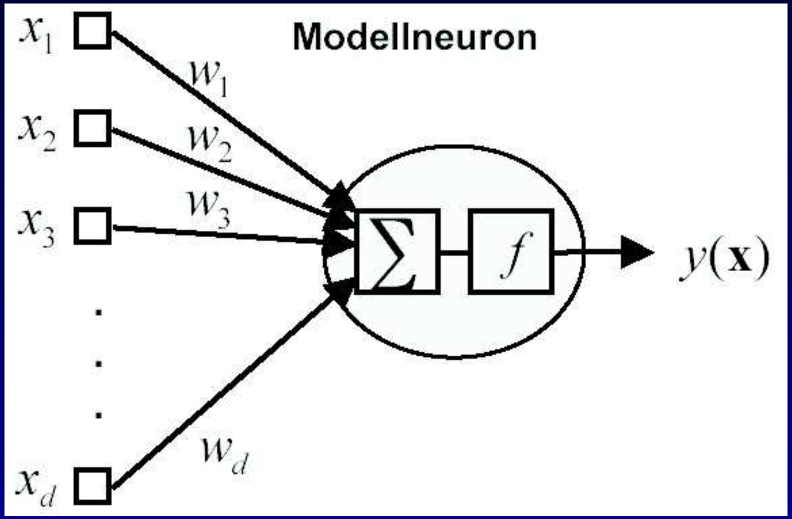
- ⋆ Protonen: Mittelung der Fits für alle Lagen
- ⋆ Elektronen: Lagen 1 - 4 einzelne Fits, Lagen 5 - 20 Mittelung

Vergleich der Methoden



- ⋆ Bei 250 GeV-Protonen noch Probleme mit dem Fit, daher kein LH-Wert!

Neuronale Netze: Konzept



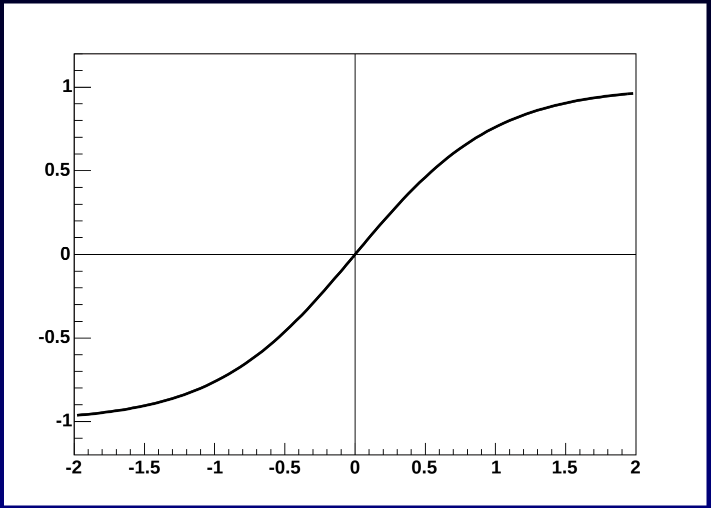
- ⋆ einfaches mathematisches Modell zur biologischen Reizweiterleitung in Nervenzellen (Neuronen)

- ⋆ Eingabe in Neuron löst eine Ausgabe aus, falls die Eingabe über einer gewissen Schwelle liegt.

- ⋆ Eingaben in Neuron addieren sich.

- ⋆ Erstellen eines Netzes aus mehreren Schichten von Neuronen.

NN: Schwelle und Fehler

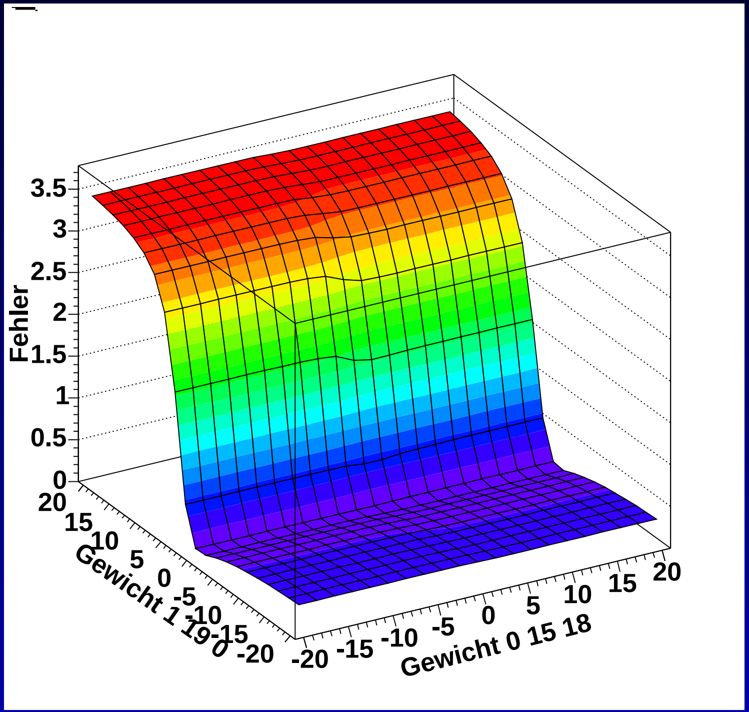


- $\star$ Sigmoide Ausgabefunktion (tanh) statt unstetiger Stufe entspricht mehr dem biologischen Vorbild.
  Vorteil: Differenzierbarkeit

- $\star$ Schwelle der Eingabeneuronen über Bias realisiert, der bei jeder Eingabe konstant 1 ist.
  $\Rightarrow$ Schwellen von Gewicht abhängig.
- $\star$ Ziel: Eingabe soll bestimmte Erwartung bei Ausgabe erfüllen.
- $\star$ Unterschied zwischen Erwartung und tatsächlicher Ausgabe wird durch den mittleren quadratischen Fehler angegeben (MSE):

$$F(\vec{w}) = \frac{1}{N} \sum_{k=1}^{N} \underbrace{\left(E(\vec{x}_k) - o_{\mathrm{Netz}}(\vec{x}_k, \vec{w})\right)^2}_{f_k(\vec{w})} \tag{13}$$

NN: Backpropagation



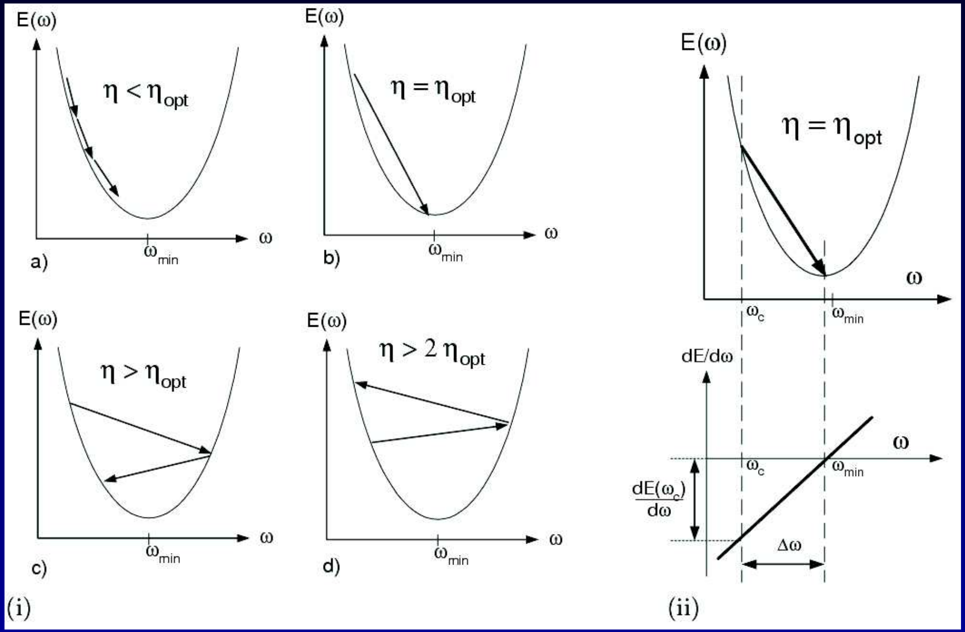
- ⋆ Ziel: möglichst gute Anpassung an Erwartung, der Fehler soll also **MINIMIERT** werden!
- ⋆ Verschieben des Gewichts in Richtung des **NEGATIVEN** Gradienten auf der Fehleroberfläche mit gewisser Schrittweite

$$\vec{w}(t+1) = \vec{w}(t) - \Delta\vec{w}(t+1) \quad (14)$$

$$\Delta w_i(t+1) = \eta \cdot \frac{1}{N}\sum_{k=1}^{N}\frac{\partial f_k(\vec{w})}{\partial w_i} \quad (15)$$

- ⋆ Es gibt einen Algorithmus, der diese Ableitung exakt berechnet, indem er den Fehler der Ausgabe rückwärts durch das Netz propagiert.

NN: Hesse-Matrix



- ⋆ quadratische Näherung im Minimum:

$$F(w) = F(w_{\text{aktuell}}) + (w - w_{\text{aktuell}})\frac{dF(w_{\text{aktuell}})}{dw} + \frac{1}{2}(w - w_{\text{aktuell}})^2 \frac{d^2F(w_{\text{aktuell}})}{dw^2} \quad (16)$$

- ⋆ Minimum des Gewichts wird erreicht, wenn man das Inverse der Hesse-Matrix als Lernrate verwendet:

$$w_{\min} = w_{\text{aktuell}} - \overbrace{\left(\frac{d^2F(w_{\text{aktuell}})}{dw^2}\right)^{-1}}^{\eta_{\text{opt}}} \frac{dF(w_{\text{aktuell}})}{dw} \quad (17)$$

NN: Lernregeln I

- ⋆ Verbesserung des Lernens durch Näherungen zur Hesse-Matrix

- ⋆ Bestimmung der Diagonalelemente der Hesse-Matrix über einen weiteren Backpropagation-Schritt, ergibt Näherung für die **INDIVIDUELLE** Lernrate von jedem Gewicht:

$$\eta_i = \frac{\epsilon}{\langle \frac{d^2 F}{dw_i^2} \rangle + \mu} \tag{18}$$

- ⋆ Bayesian-Interpretation des Neuronalen Netzes. Lernparameter werden so angepasst, dass die Wahrscheinlichkeit der Gewichtsverteilung maximal ist.
  Bei dieser Methode wird auch die Hesse-Matrix verwendet.

NN: Lernregeln II

- $\bar{\delta}$-δ-Lernregel: Anpassung der Lernrate je nach Produkt aufeinanderfolgender Gradienten:

$$\Delta w_i(t+1) = \eta_i(t+1) \cdot \frac{1}{N} \sum_{k=1}^{N} \frac{\partial f_k(\vec{w})}{\partial w_i}(t+1) + \alpha \cdot \Delta w_i(t) \tag{19}$$

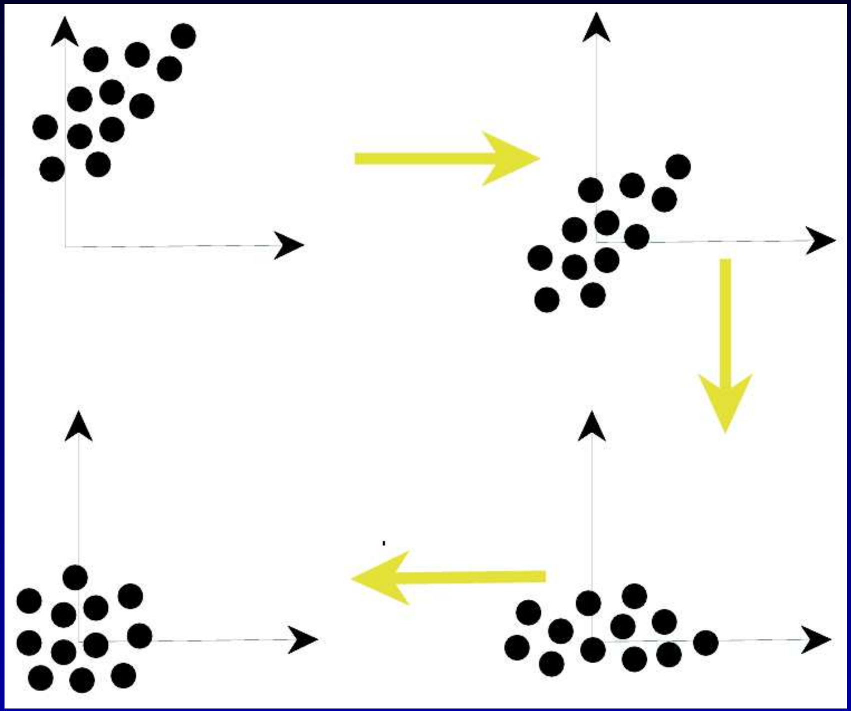
η_i: individuelle Lernrate, α: Moment

Veränderung der Lernrate:

$$\eta_i(t+1) = \begin{cases} \eta_i(t) \cdot \eta^- & \frac{\partial f_k(\vec{w})}{\partial w_i}(t+1) \frac{\partial f_k(\vec{w})}{\partial w_i}(t) < 0 \\ \eta_i(t) + \eta^+ & \frac{\partial f_k(\vec{w})}{\partial w_i}(t+1) \frac{\partial f_k(\vec{w})}{\partial w_i}(t) > 0 \\ \eta_i(t) & \text{sonst} \end{cases} \tag{20}$$

- **RProp**-Lernregel:
  - nur Verwendung des Vorzeichens vom Fehlergradienten
  - Zurücknahme von Schritten, die über ein Minimum hinausgehen.
  - Anpassung der Lernraten wie bei $\bar{\delta}$-δ-Lernregel

NN: Datentransformation



- ⋆ Transformation der Input-Daten, um alle Eingaben gleichwertig zu berücksichtigen.
- ⋆ pro Input-Neuron: Mittelwert 0 mit Breite 1

1. Abzug des Mittelwertes und Normierung auf Breite:

$$x_i^{\text{trafo}} = \frac{x_i - \bar{x}}{\sigma} \tag{21}$$

2. Transformation der Kovarianzmatrix auf die Einheitsmatrix mit Eigenrotation:

$$V = \sum(\vec{x}_n - \bar{\vec{x}})(\vec{x}_n - \bar{\vec{x}})^T \tag{22}$$

Startgewichte und Mini-Batch-Mode

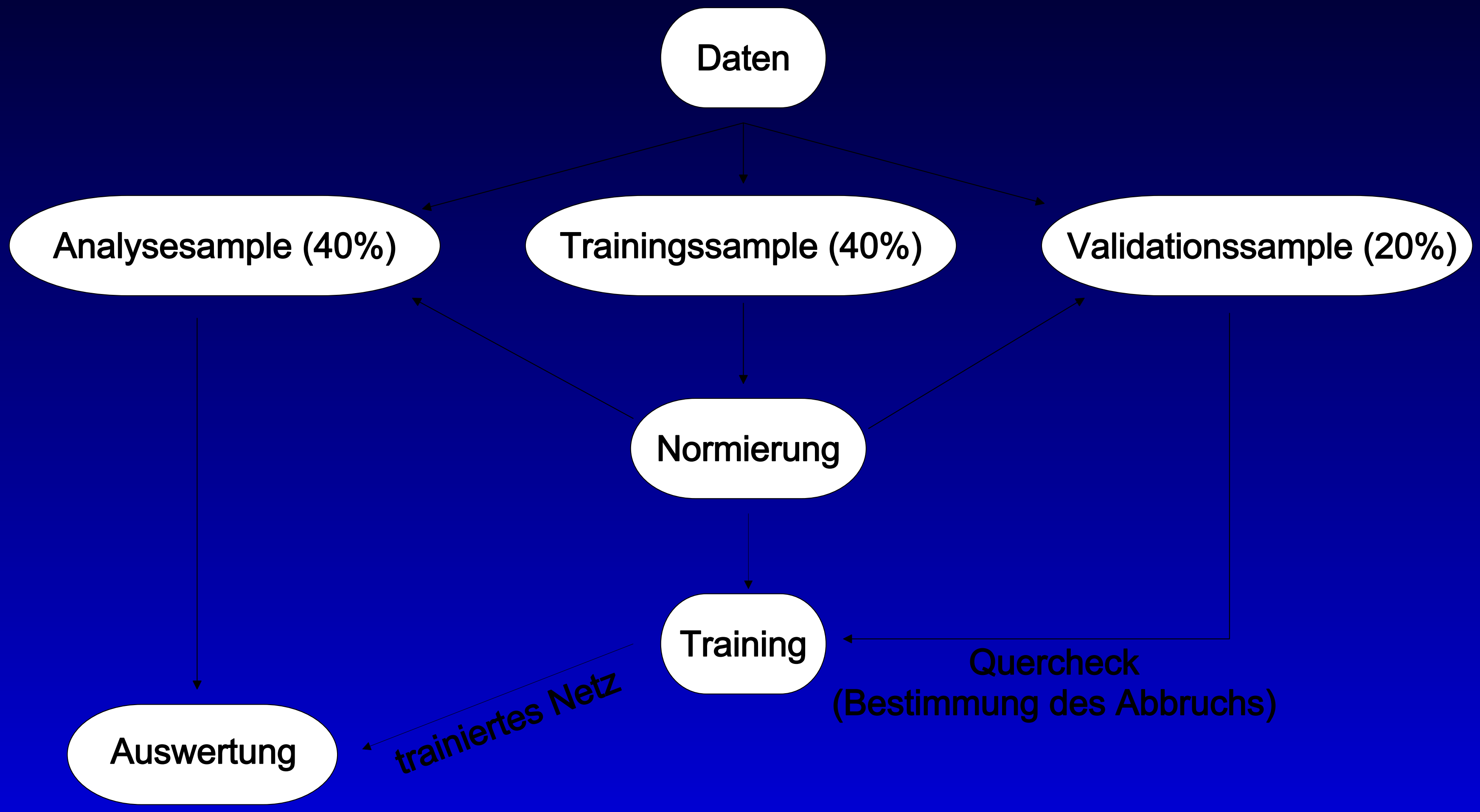
- Nach Transport in die nächste Netzlage, soll die Breite des Signals immer noch 1 sein.
  $\Rightarrow$ vor dem Training: Würfeln der Gewichte um eine Verteilung mit Breite

$$\sigma = m^{-\frac{1}{2}} \tag{23}$$

  m ist die Anzahl der Eingaben in die nächste Schicht.

- Gewichte sollen im Bereich von $3\,\sigma$ liegen, um von Anfang an grobe Ausreisser zu unterbinden!
- Gewichte werden nach 200 bzw. 10 Eventpräsentationen verändert (Mini-Batch-Mode).

Analyseprozess



Aufgaben des Neuronalen Netzes

- ⋆ Neuronales Netz (NN) erhält als Input die Energiedepositionen der Hits auf der Spur pro Lage. Also: 20 Input Neuronen
- ⋆ Finden einer Abbildungsvorschrift:

$$\left.\begin{array}{rcr} \text{Elektronen} & \longrightarrow & 1 \\ \text{Protonen} & \longrightarrow & -1 \end{array}\right\} \longrightarrow \text{Unterscheidungskriteriu}$$

- ⋆ Rejection soll besser als 100 sein.
- ⋆ Auffinden der besten Konfiguration des NN durch Variation der Parameter.
- ⋆ Untersuchung der Energiedepositionen pro Lage auf mögliche Korrelationen.

Auffinden der besten Lernmethode

1. Verwenden eines dreilagigen Netzes mit 20 (21)-Eingaben, variabler Zahl mittlerer Neuronen mit sigmiodem Output und einem sigmoiden Output-Neuron.

2. vorgestellte Lernregeln mit Näherungen zur Hesse-Matrix sind entweder zu ungenau oder zu rechenintensiv!

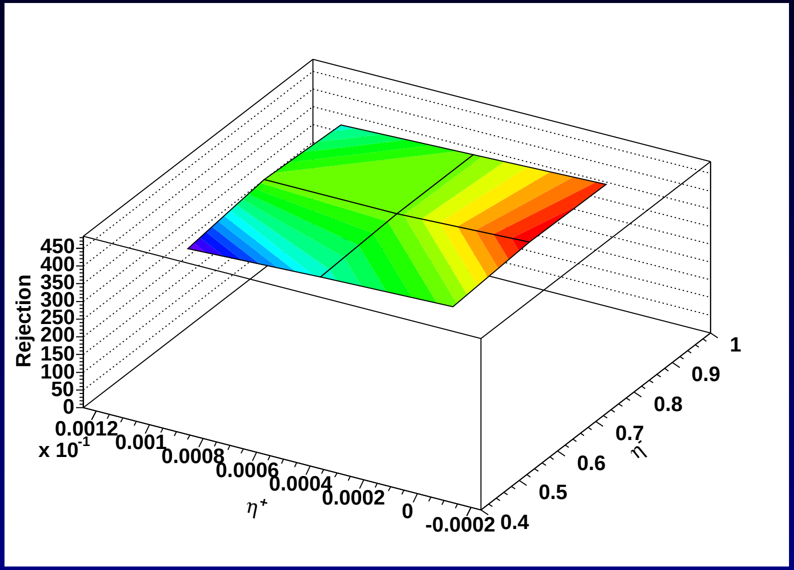
3. Training mit RProp- oder $\bar{\delta}$-δ-Lernregel

4. Eingabe von Energien oder von Likelihood-Ratios

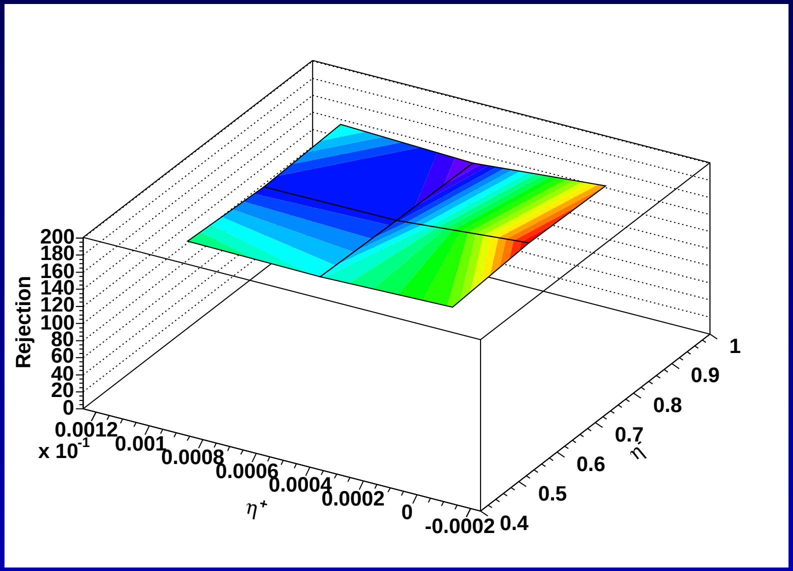
5. Datentransformation:
   (a) Abzug des Mittelwertes und Normierung auf Breite
   (b) Transformation der Kovarianzmatrix auf Einheitsmatrix

RProp-Auswertung

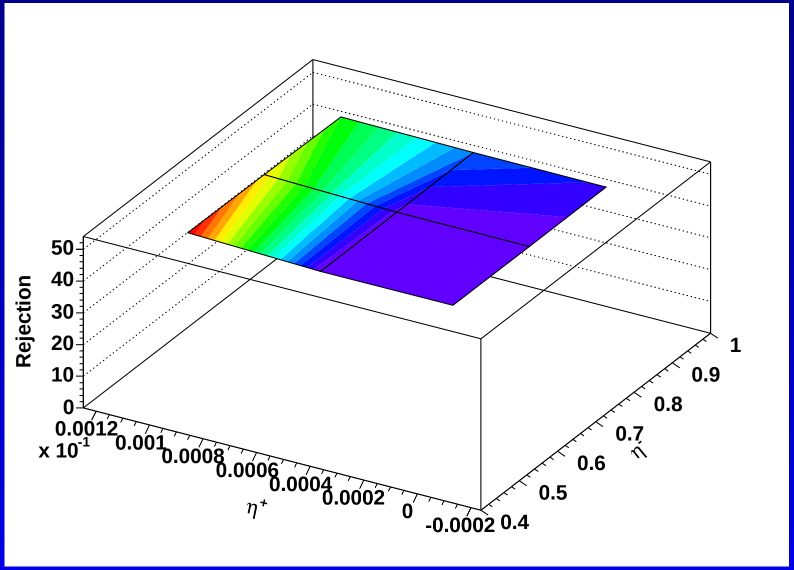
Datentransformation (a), Energieeingabe ⟶



Datentransformation (b), Energieeingabe ↓

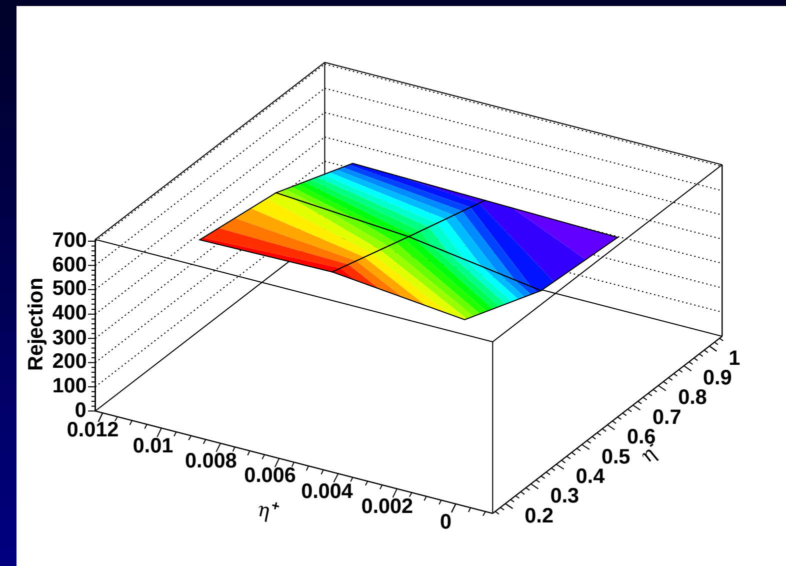


Datentransformation (a), LH-Ratio-Eingabe ⟶

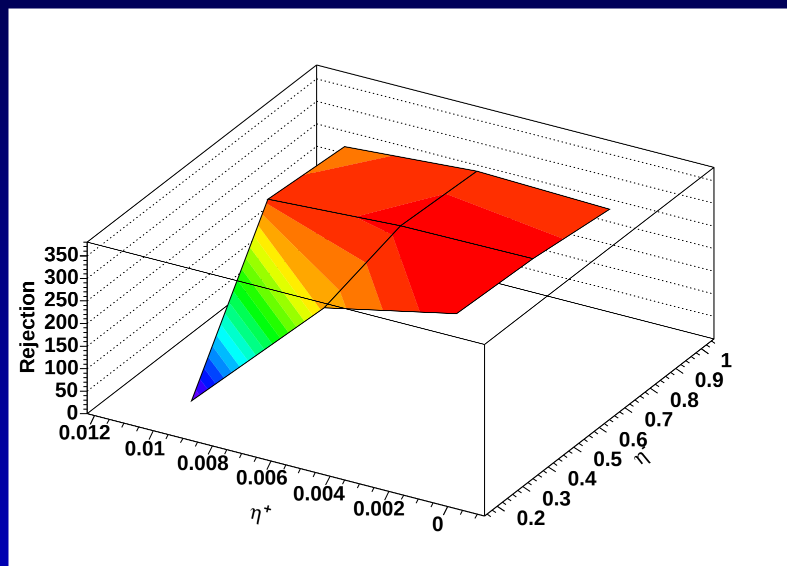


$\bar{\delta}$-δ-Auswertung

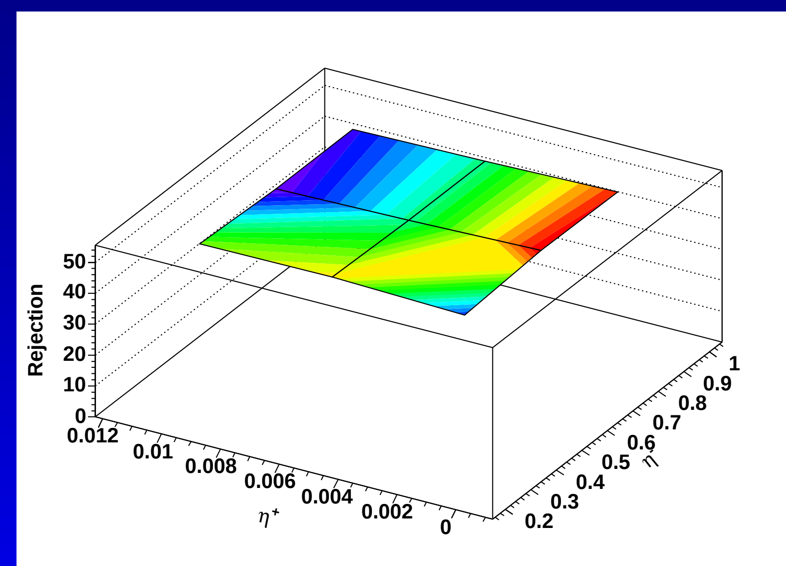
Datentransformation (a), Energieeingabe $\longrightarrow$



Datentransformation (b), Energieeingabe $\downarrow$

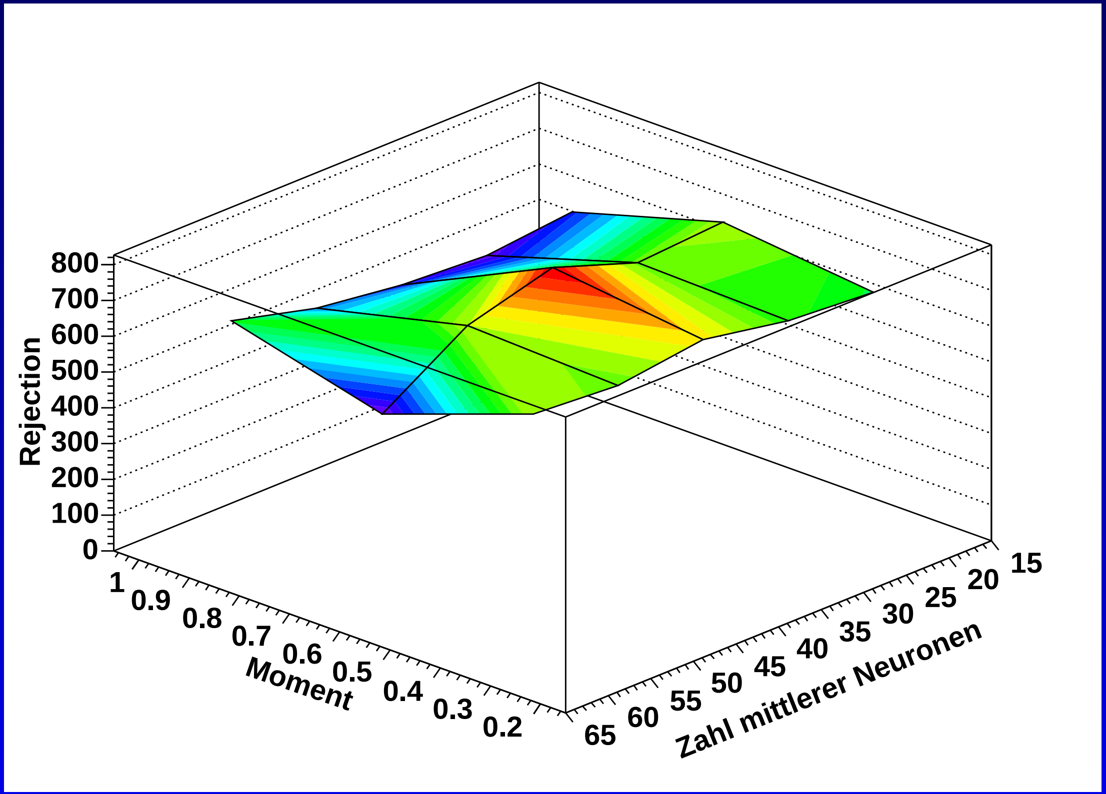


Datentransformation (a), LH-Ratio-Eingabe $\longrightarrow$

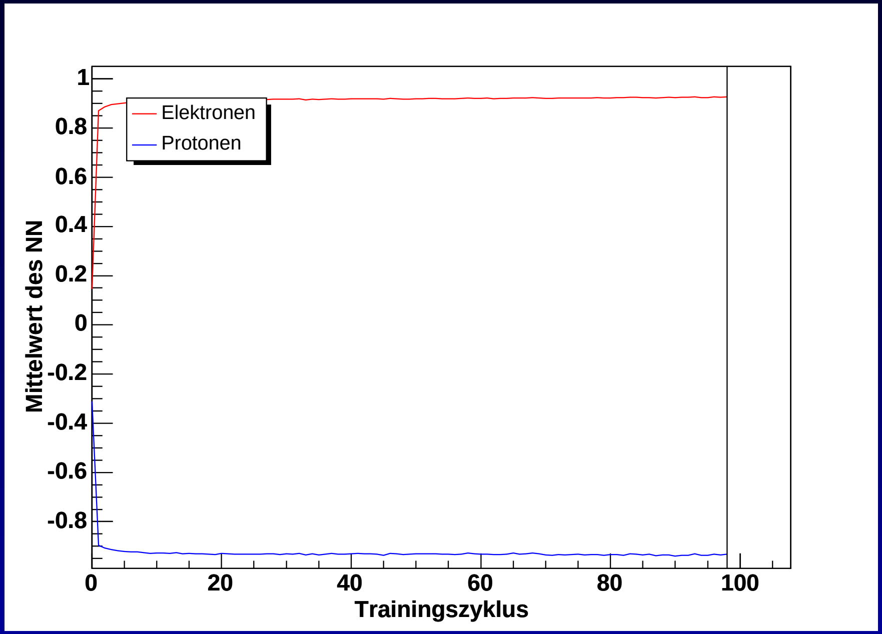


Weitere Parametervariation

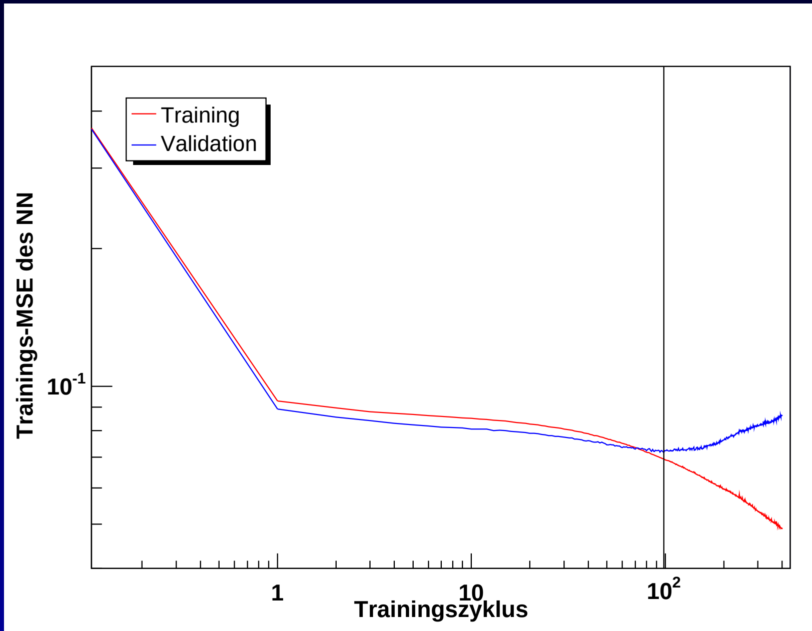
- ⋆ bisher bestes Training mit
  - ⋆ Eingabe der normierten Energien
  - ⋆ $\bar{\delta}$-δ-Lernregel
- ⋆ genauere Variation von mittleren Neuronen und Moment α



Trainingsgrößen des besten Netzes



Mittelwerte

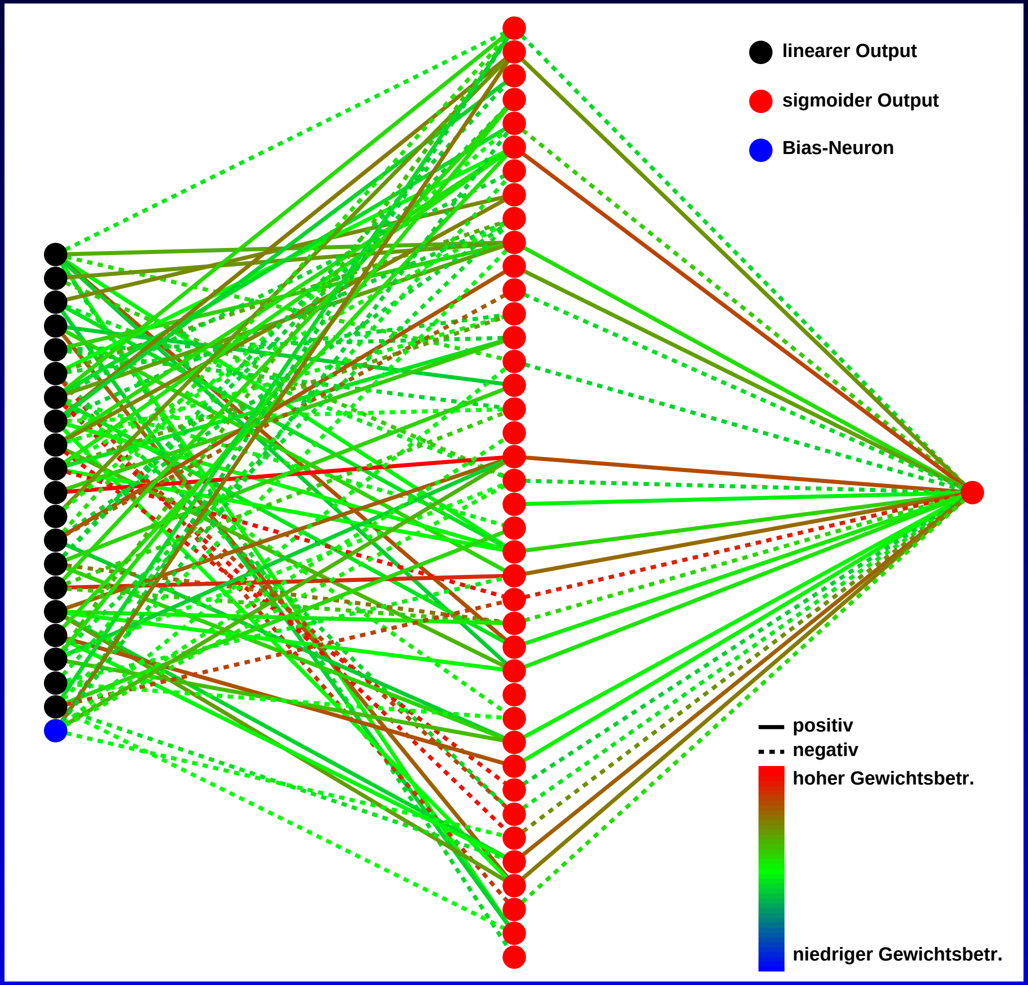


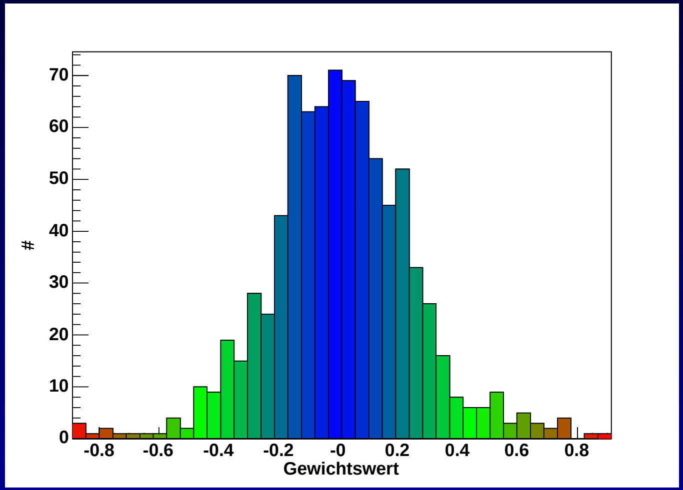
Fehler

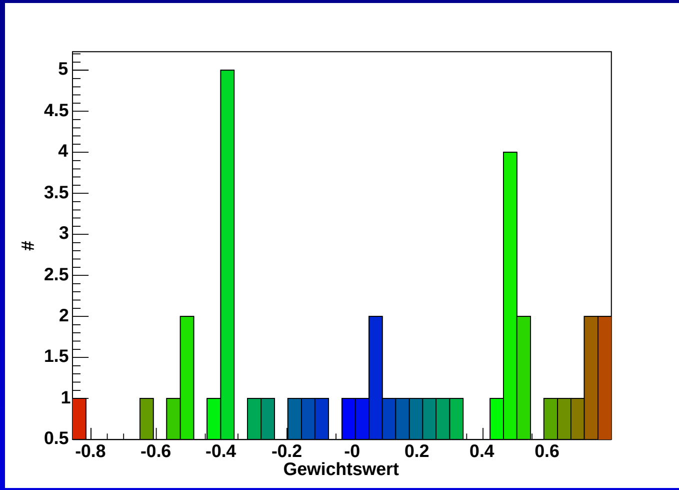
⋆ Training wird an der Stelle mit minimalem Fehler des Validationssamples abgebrochen.

Struktur des Netzes

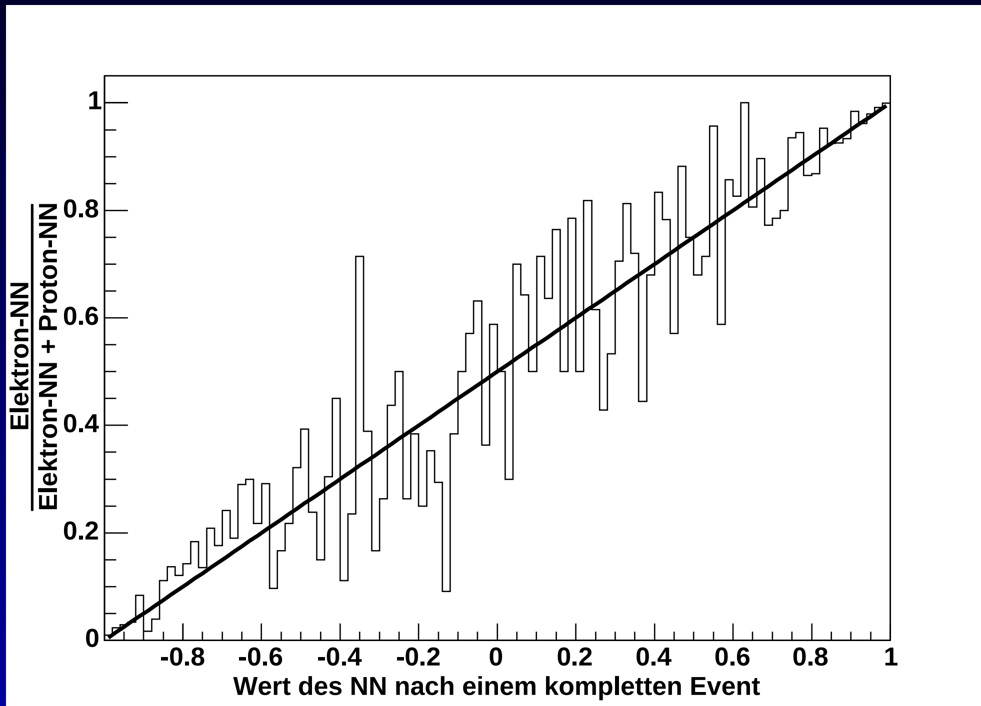
⋆ Gewichte nach dem Training







Test, ob das NN gut trainiert ist



- ⋆ Anteil der Elektronen:

$$P_{\mathrm{e}} = \frac{N_{\mathrm{e}}}{N_{\mathrm{p}} + N_{\mathrm{e}}} \qquad (24$$

- ⋆ theoretischer mittlerer quadratischer Fehler:

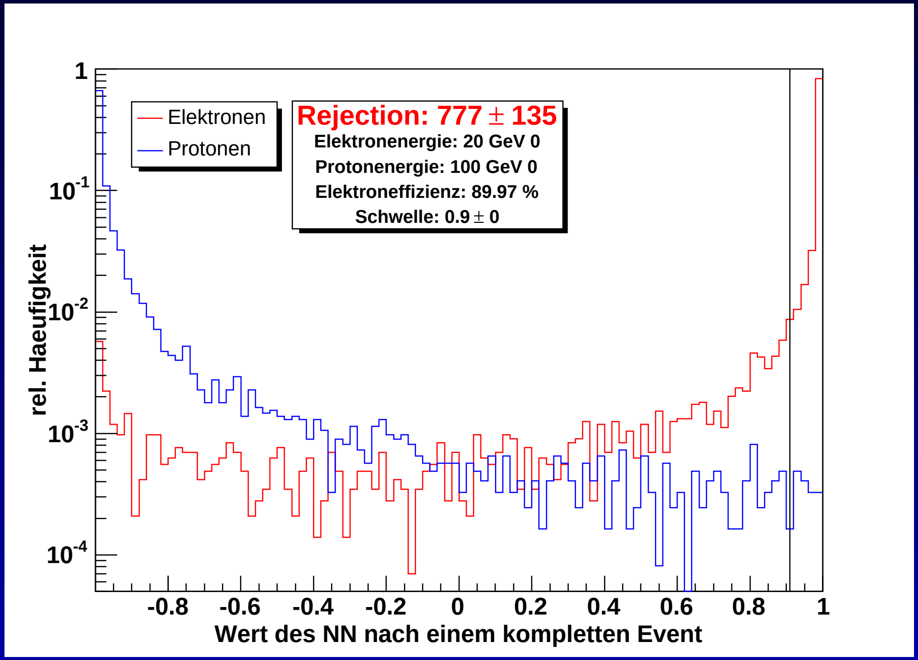
$$F = P_{\mathrm{e}}(1-o_{\mathrm{Netz}})^2+(1-P_{\mathrm{e}})(-1-o_{\mathrm{N}} \qquad (25$$

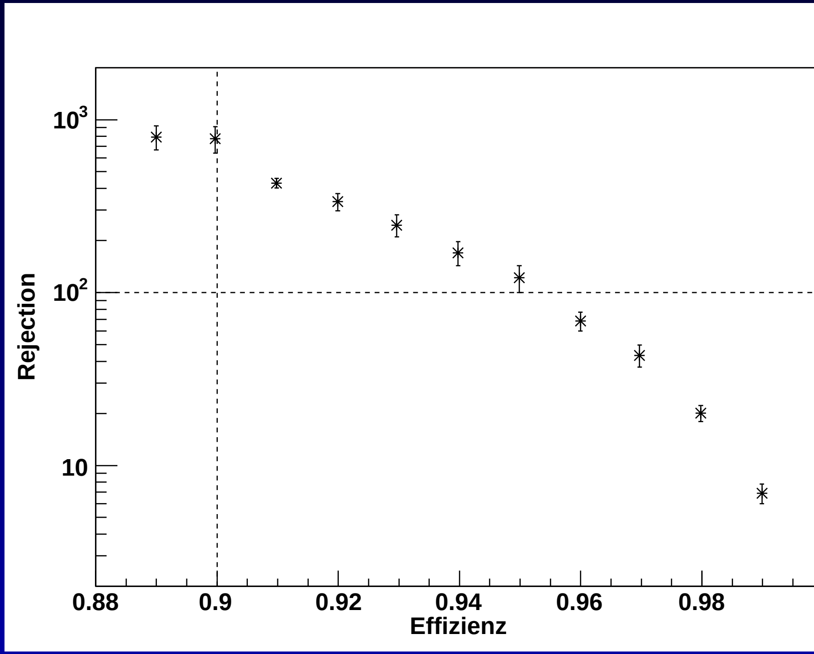
- ⋆ Der Fehler ist minimal, wenn gilt:

$$P_{\mathrm{e}} = \frac{o_{\mathrm{Netz}} + 1}{2} \qquad (26)$$

- ⋆ Das Netz ist also gut trainiert!

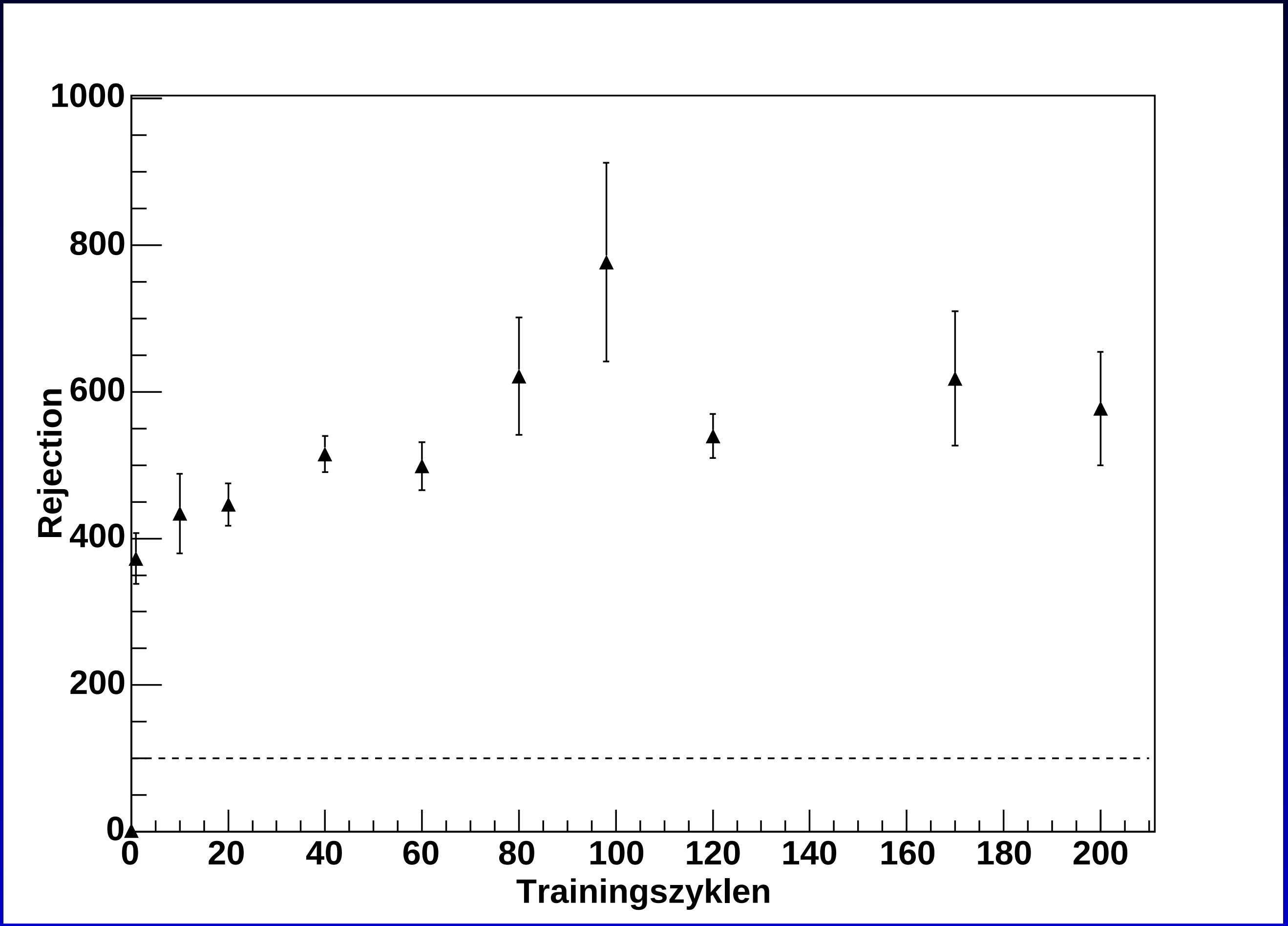
Rejection





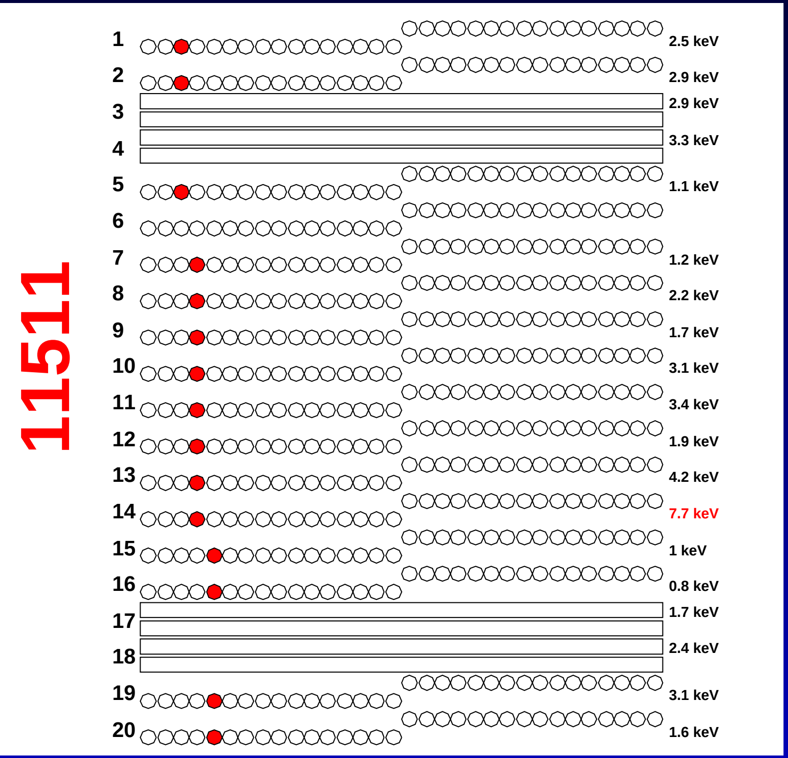
- ⋆ Rejection liegt also weit über den geforderten 100!
- ⋆ Glatter Verlauf bei Änderung der Effizienz!

Rejection während des Trainings

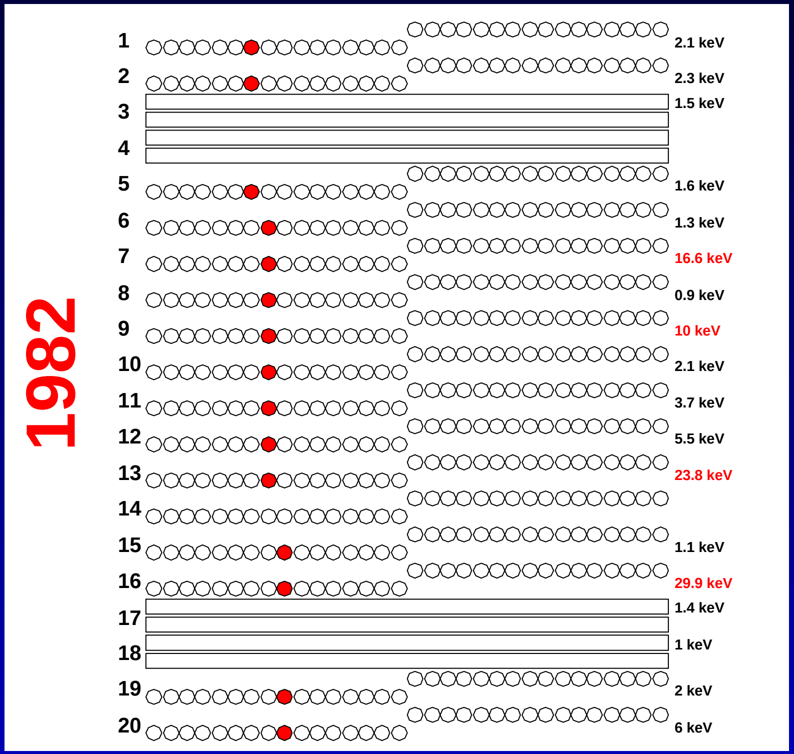


- ⋆ Training wird also wirklich am Maximum der Rejection abgebrochen!

Beispiele für falscherkannte Events

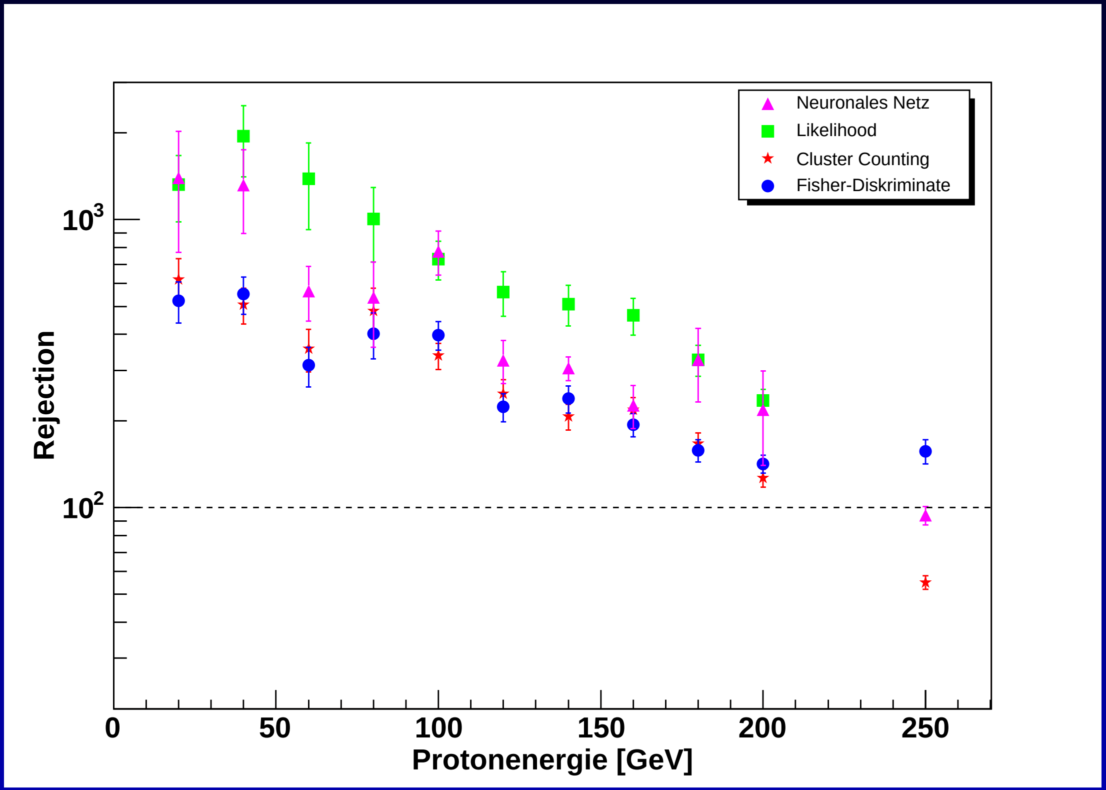


falscherkanntes Elektron:
keine hohen Einträge



falscherkanntes Proton:
wenige sehr hohe Einträge

Training von NNs für alle Energien



- Problem:
  - Zahl der Elektron- und Protonevents nur bei 20/100 GeV ungefähr gleich. Ansonsten immer viel mehr Elektronen.
  - Training aber mit gleich vielen Events beider Klassen besser.

$\Rightarrow$ mögliche Lösung: ein Netz für alle Energien

Änderung der Eingabereihenfolge

- ⋆ Netz wird normal trainiert.
- ⋆ Nachdem Training wird jedes Event durcheinandergewürfelt.
- ⋆ Rejection sinkt um ca. 100!

$\Rightarrow$ Netz ist auf die Eingabereihenfolge sensitiv!

Weiteres NN

- Zusammenfassen aller Analysemethoden mit einem Neuronalen Netz.

- Eingabe von nur 5 Größen:
  - mittlere Energie
  - Anzahl von Hits über bestimmter Energie
  - NN-Wert
  - LH-Wert
  - Fisher-Wert

- Ziel: Ausnutzen von allen individuellen Vorteilen der Methoden.

Zusammenfassung

- ⋆ Klassische Analyse:
  Cluster-Counting, Fisher-Diskriminante und Likelihood-Methode
- ⋆ Fits der Energieverteilungen
- ⋆ Neuronale Netze sind weitere gute Analysemöglichkeit.
- ⋆ NN-Analyse hat bei allen Energien eine Rejection über 100!

Ausblick

- ⋆ Verstehen des Einflusses der Overflows
- ⋆ Ausweitung des Verfahrens auf alle Energien
- ⋆ MC-Tests
- ⋆ NeuroBayes