NETZWERK TEILCHENWELT

UNIVERSITÄT BONN

Maike Hansen*, Johanna Rätz, Barbara Valeriani-Kaminski

Machine Learning Masterclass - Physik trifft Daten

DPG Frühjahrstagung 2023

21. März 2023

Bundesministerium für Bildung und Forschung

PUNCH 4NFDI

Masterclass - Was ist das?

→ Forscher*innen für einen Tag

Masterclass in der (Teilchen-)physik

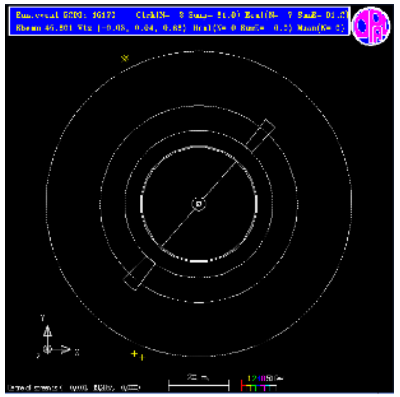
- Ab Klassenstufe 10
- Vorträge + „Hands-On“ Aktivitäten
  - Einführung
  - Datenanalyse am PC (in 2er Gruppen)
  - Diskussion
  - Internationale Video Konferenz

- ATLAS, ALICE, CMS, LHCb, Belle II, Hadronentherapie, Nukleare Astrophysik, IceCube, Auger, **Maschinelles Lernen mit Opal Daten, …**

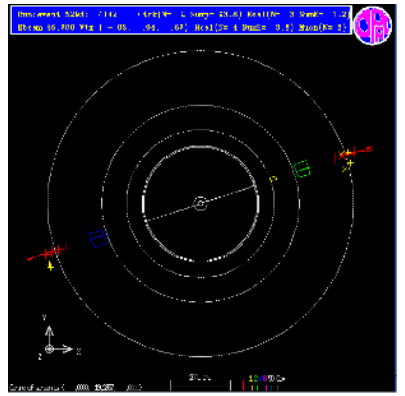
Prinzip der „Machine Learning (ML)“ Masterclass

- Urspr. von Nikolas Tiltmann (Uni Münster)
  - Weiterentwicklung OPAL[1]-MC (Terry Watt)
  - DPG-Vortrag 2021 zur ML-MC
  - GitHub mit Code

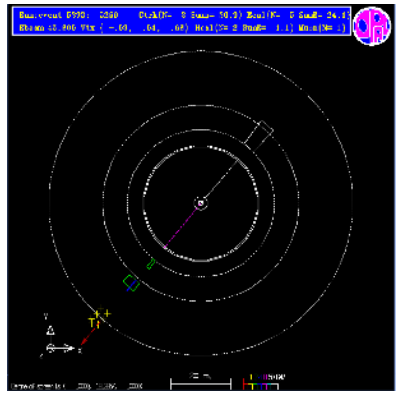
- Ziel: Z-Boson Zerfälle mit maschinellem Lernen klassifizieren
- Schüler*innen programmieren & trainieren neuronales Netz in Jupiter Notebook
- Testläufe urspr. Version mit Schülerpraktikanten (Münster)
- Testläufe der Weiterentwicklung mit Schulklassen (Bonn)



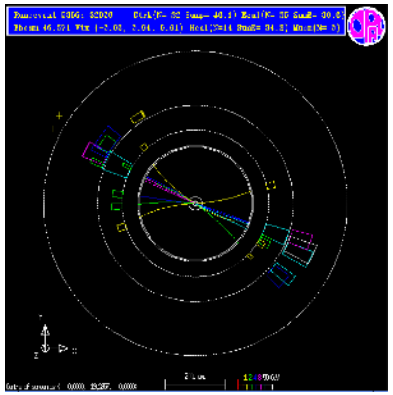
Z→e+e-



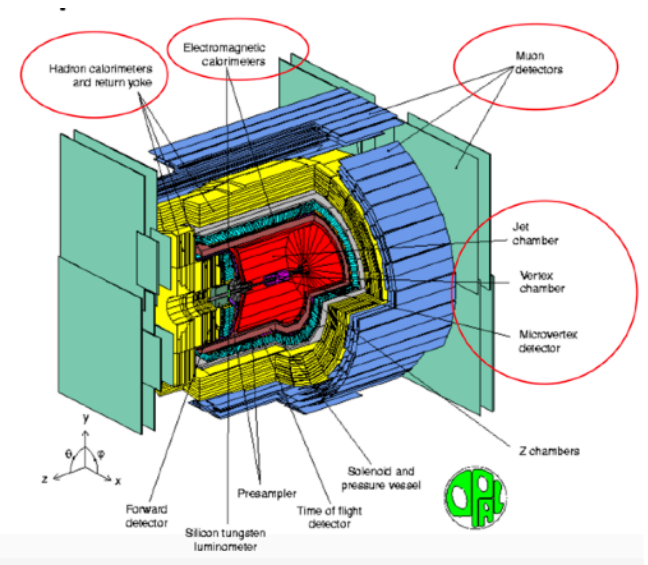
Z→μ+μ-



Z→τ+τ-



Z→quarks



1) **O**mni **P**urpose **A**pparatus at **L**EP, CERN (1989-2000)

Didaktische Weiterentwicklung für Schulklassen

- Johanna Rätz (Uni Bonn, PUNCH4NFDI)
- Anpassung an Tagesveranstaltung mit Schulklassen
- Arbeitsblätter & interaktive Elemente
  - zur theoretischen Vorhersage des Verzweigungsverhältnisses
  - zum Aufbau des Codes
  - zum Training eines Convolutional Neuronalen Netzwerkes
  - Bauernschach: Spiel Mensch gegen Maschine
- Didaktische Aufarbeitung Code/ Jupyter Notebooks
  - Einführungs-Notebook geschrieben
  - Notebook mit Code als Lückentext
    - Möglich auch ohne Programmiererfahrung
    - Bei größeren Gruppen(Schulklassen) besser geeignet

Ablauf der ML-Masterclass

- Dauer ca. 7h (inkl. Pausen)

- Vortrag Standard Modell (AB Verzweigungsverhältnis)
- Vortrag Detektoren & Beschleuniger (Online Spiel)
- Vortrag Machine Learning & Neuronale Netzwerke (Spiel “Mensch, Maschine!”)

- Einführung Programmieren (Einführungs-Notebook) & Jupyter
- Arbeit in 2er Gruppen an Jupyter Notebooks (AB Code & Neural Network Training)
- Diskussion Ergebnisse & Gesellschaftlicher Aspekt von ML

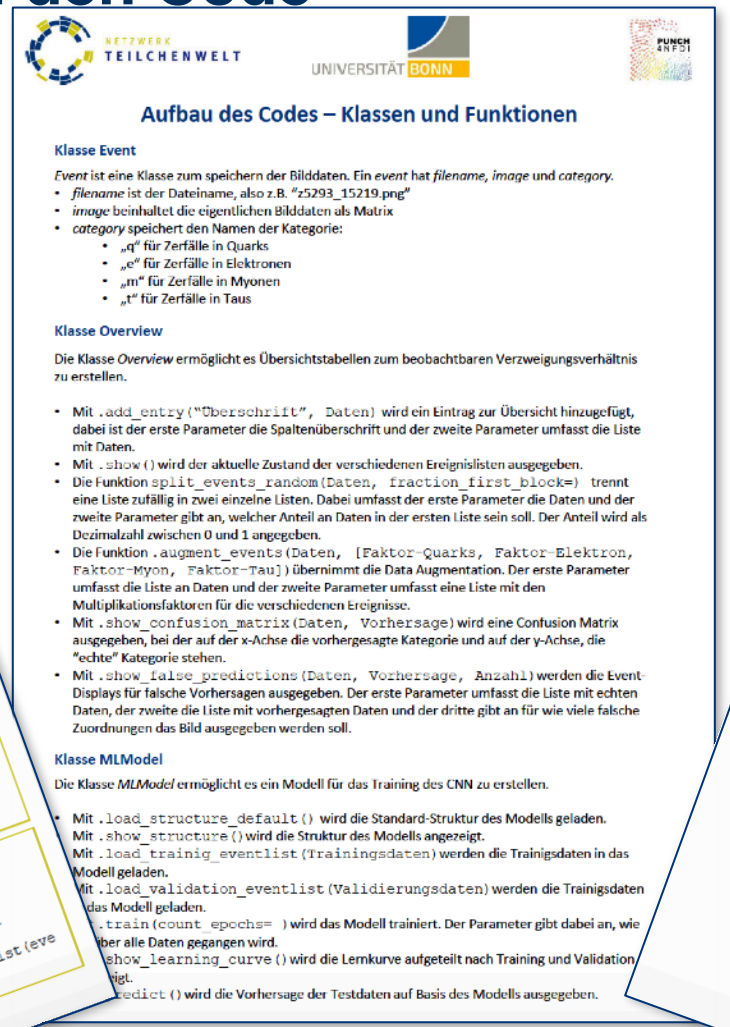
Arbeitsblätter führen durch den Code

Aufbau des Codes

Aufgabe: Auf der linken Seite siehst du den schematischen Aufbau des Codes, der nachher ein *Convolutional Neural Network* zur Auswertung von OPAL-Daten trainiert. Auf der rechten Seite siehst du für einige Elemente die zugehörigen Bausteine des Jupyter Notebooks. **Verbinde** die Code-Bausteine mit den entsprechenden Bestandteilen des Codes. Wenn du fertig bist, **vergleiche** deine Lösung mit der Musterlösung.
Hinweis: Nutze gerne die Übersicht über die verschiedenen Klassen und Funktionen auf der zweiten Seite.
Achtung: Nicht für alles auf der linken Seite ist auf der rechten Seite auch ein Code Block zu finden.

- Pakete und Funktionen laden
- Events laden
- Übersicht zur Analyse des Verzweigungsverhältnisses anlegen und die Rohdaten einfügen
- Bilddateien in zwei Datensätze aufteilen
- Data Augmentation durchführen
- Daten für Training und Validierung aufteilen
- Übersicht erweitern
- Modell für das Training des CNN erstellen
- Training des CNN über mehrere Epochen
- Lernkurve des Modells anzeigen
- Kategorien der Testdaten vorhersagen
- Confusion Matrix anzeigen
- Übersicht über die Verzweigungsverhältnisse um Vorhersage ergänzen
- Ereignisse, die falsch zugeordnet wurden betrachten

```
modell.show_learning_curve()
modell.train(count_epochs= )
eventliste = load_events()
eventliste_vorhersage =
modell.predict(eventliste_test)
show_confusion_matrix(eventliste_
test, eventliste_vorhersage)
overview.add_entry("Vorh",
eventliste_vorhersage)
overview.show()
eventliste_tv, eventliste_test =
split_events_random(eventliste,
fraction_first_block=)
overview = Overview()
overview.add_entry("Koh",eventliste)
overview.show()
faktor =
eventliste_tv_vermehrt =
augment_events(eventliste_tv,
[faktor, faktor, faktor, faktor])
modell = MLModel()
modell.load_structure_default()
modell.show_structure()
modell.load_training_eventlist(
eventliste_training)
modell.load_validation_eventlist(eve
ntliste_validierung)
```



Aufbau des Codes – Klassen und Funktionen

Klasse Event

Event ist eine Klasse zum speichern der Bilddaten. Ein *event* hat *filename*, *image* und *category*.

- *filename* ist der Dateiname, also z.B. "z5293_15219.png"
- *image* beinhaltet die eigentlichen Bilddaten als Matrix
- *category* speichert den Namen der Kategorie:
  - „q" für Zerfälle in Quarks
  - „e" für Zerfälle in Elektronen
  - „m" für Zerfälle in Myonen
  - „t" für Zerfälle in Taus

Klasse Overview

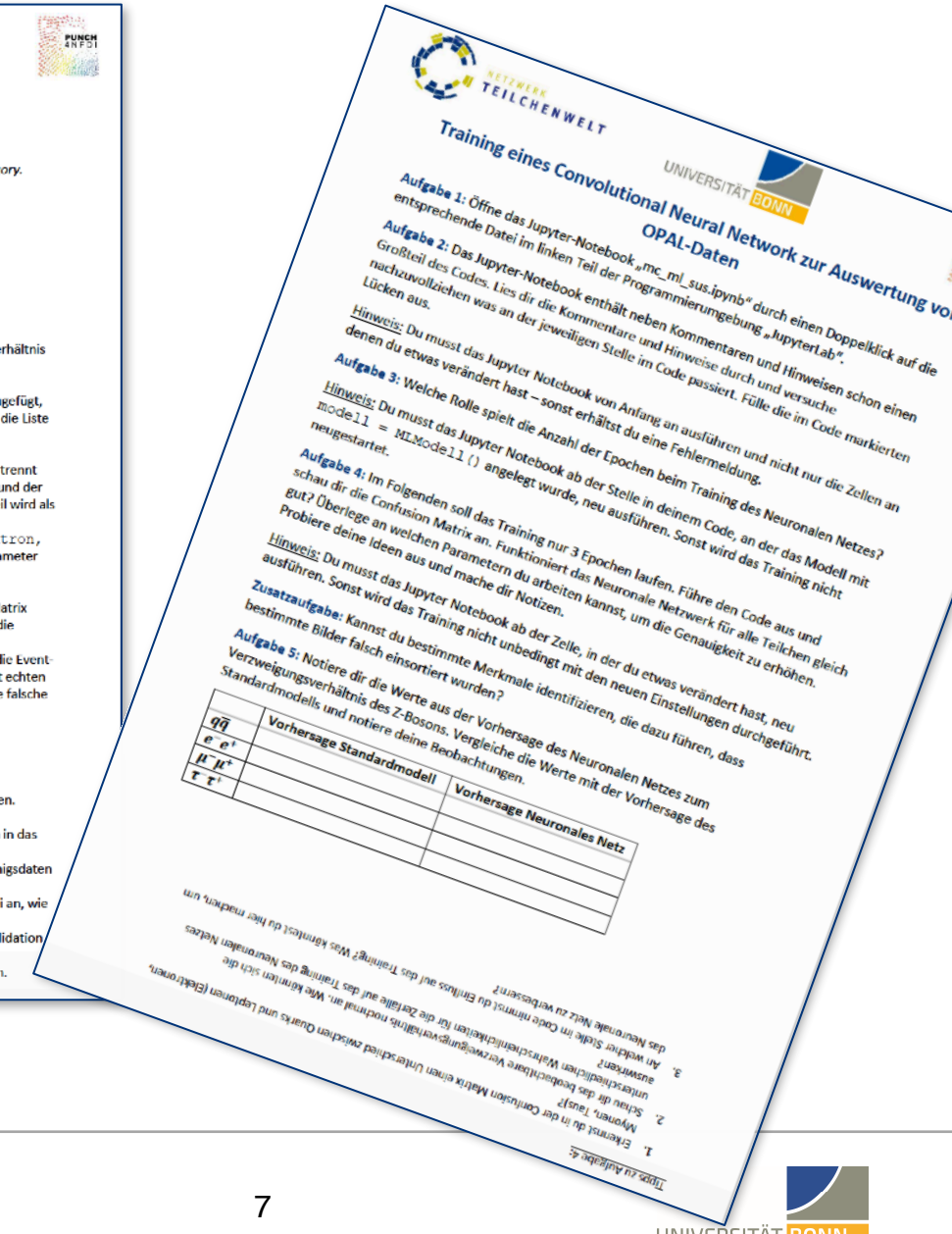
Die Klasse *Overview* ermöglicht es Übersichtstabellen zum beobachtbaren Verzweigungsverhältnis zu erstellen.

- Mit `.add_entry("Überschrift", Daten)` wird ein Eintrag zur Übersicht hinzugefügt, dabei ist der erste Parameter die Spaltenüberschrift und der zweite Parameter umfasst die Liste mit Daten.
- Mit `.show()` wird der aktuelle Zustand der verschiedenen Ereignislisten ausgegeben.
- Die Funktion `split_events_random(Daten, fraction_first_block=)` trennt eine Liste zufällig in zwei einzelne Listen. Dabei umfasst der erste Parameter die Daten und der zweite Parameter gibt an, welcher Anteil an Daten in der ersten Liste sein soll. Der Anteil wird als Dezimalzahl zwischen 0 und 1 angegeben.
- Die Funktion `.augment_events(Daten, [Faktor-Quarks, Faktor-Elektron, Faktor-Myon, Faktor-Tau])` übernimmt die Data Augmentation. Der erste Parameter umfasst die Liste an Daten und der zweite Parameter umfasst eine Liste mit den Multiplikationsfaktoren für die verschiedenen Ereignisse.
- Mit `.show_confusion_matrix(Daten, Vorhersage)` wird eine Confusion Matrix ausgegeben, bei der auf der x-Achse die vorhergesagte Kategorie und auf der y-Achse, die "echte" Kategorie stehen.
- Mit `.show_false_predictions(Daten, Vorhersage, Anzahl)` werden die Event-Displays für falsche Vorhersagen ausgegeben. Der erste Parameter umfasst die Liste mit echten Daten, der zweite die Liste mit vorhergesagten Daten und der dritte gibt an für wie viele falsche Zuordnungen das Bild ausgegeben werden soll.

Klasse MLModel

Die Klasse *MLModel* ermöglicht es ein Modell für das Training des CNN zu erstellen.

- Mit `.load_structure_default()` wird die Standard-Struktur des Modells geladen.
- Mit `.show_structure()` wird die Struktur des Modells angezeigt.
- Mit `.load_trainig_eventlist(Trainingsdaten)` werden die Trainigsdaten in das Modell geladen.
- Mit `.load_validation_eventlist(Validierungsdaten)` werden die Trainigsdaten in das Modell geladen.
- `.train(count_epochs= )` wird das Modell trainiert. Der Parameter gibt dabei an, wie oft über alle Daten gegangen wird.
- `.show_learning_curve()` wird die Lernkurve aufgeteilt nach Training und Validation angezeigt.
- `.predict()` wird die Vorhersage der Testdaten auf Basis des Modells ausgegeben.



Training eines Convolutional Neural Network zur Auswertung von OPAL-Daten

Aufgabe 1: Öffne das Jupyter-Notebook „mc_ml_sus.ipynb" durch einen Doppelklick auf die entsprechende Datei im linken Teil der Programmierumgebung „JupyterLab".

Aufgabe 2: Das Jupyter-Notebook enthält neben Kommentaren und Hinweisen schon einen Großteil des Codes. Lies dir die Kommentare und Hinweise durch und versuche nachzuvollziehen was an der jeweiligen Stelle im Code passiert. Fülle die im Code markierten Lücken aus.

Hinweis: Du musst das Jupyter Notebook von Anfang an ausführen und nicht nur die Zellen an denen du etwas verändert hast – sonst erhältst du eine Fehlermeldung.

Aufgabe 3: Welche Rolle spielt die Anzahl der Epochen beim Training des Neuronalen Netzes?

Hinweis: Du musst das Jupyter Notebook ab der Stelle in deinem Code, an der das Modell mit `modell = MLModell()` angelegt wurde, neu ausführen. Sonst wird das Training nicht neugestartet.

Aufgabe 4: Im Folgenden soll das Training nur 3 Epochen laufen. Führe den Code aus und schau dir die Confusion Matrix an. Funktioniert das Neuronale Netzwerk für alle Teilchen gleich gut? Überlege an welchen Parametern du arbeiten kannst, um die Genauigkeit zu erhöhen. Probiere deine Ideen aus und mache dir Notizen.

Hinweis: Du musst das Jupyter Notebook ab der Zelle, in der du etwas verändert hast, neu ausführen. Sonst wird das Training nicht unbedingt mit den neuen Einstellungen durchgeführt.

Zusatzaufgabe: Kannst du bestimmte Merkmale identifizieren, die dazu führen, dass bestimmte Bilder falsch einsortiert wurden?

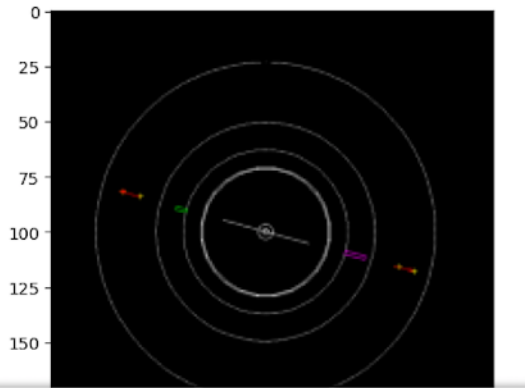
Aufgabe 5: Notiere dir die Werte aus der Vorhersage des Neuronalen Netzes zum Verzweigungsverhältnis des Z-Bosons. Vergleiche die Werte mit der Vorhersage des Standardmodells und notiere deine Beobachtungen.

	Vorhersage Standardmodell	Vorhersage Neuronales Netz
$q\bar{q}$		
e^-e^+		
$\mu^-\mu^+$		
$\tau^-\tau^+$		

Tipps zu Aufgabe 4:

1. Erkennst du in der Confusion Matrix einen Unterschied zwischen Quarks und Leptonen (Elektronen, Myonen, Tau)?
2. Schau dir das beobachtbare Verzweigungsverhältnis nochmal an. Wie könnten sich die unterschiedlichen Wahrscheinlichkeiten für die Zerfälle auf das Training des Neuronalen Netzes auswirken?
3. An welcher Stelle im Code könnte du Einfluss auf das Training? Was könntest du hier machen, um das Neuronale Netz zu verbessern?

Jupyter Notebook als "Lückentext"



In [ ]:
```
eventliste = load_events()
```

Mit `eventliste[x]` kann auf das `x-1`-ste Ereignis zugegriffen werden. Die Methode `.show_image()` gibt das Ereignis grafisch aus. Dieser Schritt hat keine Auswirkungen auf das Neuronale Netz o.ä sondern dient nur der Visualisierung.

In [ ]:
```
eventliste[53].show_image(show_category=True)
```

3. Training & Validierung

Als nächstes werden die Bilddateien in zwei Datensätze aufgeteilt. Einen für Training und Validierung (Abkürzung `tv`) und einen für den anschließenden Test des KNN. Wähle an dieser Stelle das Verhältnis selber. Trage dafür eine Zahl zwischen `0` und `1` ein. Diese Zahl gibt an wie viel Prozent der Bilddateien in den Datensatz für Training und Validierung hinzugefügt werden.

Beispiel: eventliste_tv, eventliste_test = split_events_random(eventliste, fraction_first_block=0.1) bedeutet, dass 10% der Bilddateien zufällig in den Datensatz für Training und Validierung (`eventliste_tv`) hinzugefügt werden und 90% für den anschließenden Test (`eventliste_test`) genutzt werden.

Hinweis: Bedenke, dass eine gute Datenbasis für Training und Validierung gebraucht wird, aber die für Training und Validierung verwendeten Daten bekannt sein müssen. Werden sehr viele Daten für Training und Validierung genutzt, bleiben nicht mehr ausreichend Daten für die eigentliche Datenauswertung mit dem KNN.

In [ ]:
```
eventliste_tv, eventliste_test = split_events_random(eventliste, fraction_first_block=0.3)
```

Im nächsten Schritt folgt die Data Augementation, das bedeutet, dass die `tv`-Daten vervielfältigt werden, damit das Training einfacher wird. Wähle zunächst für alle vier Faktoren (`faktor_q` , `faktor_e` , `faktor_m` , `faktor_t`) einen ganzzahligen Wert zwischen `2` und `5` mit dem die Daten vervielfältigt werden. Später kann es sinnvoll sein, diese Werte teilweise nochmal zu verändern (grundsätzlich sind dann auch größere Werte als `5` möglich). Beim Vervielfältigen werden jeweils Kopien des Bildes mit zufälligen Rotationen erstellt. Die Hälfte der Bilder wird zusätzlich zufällig gespiegelt.

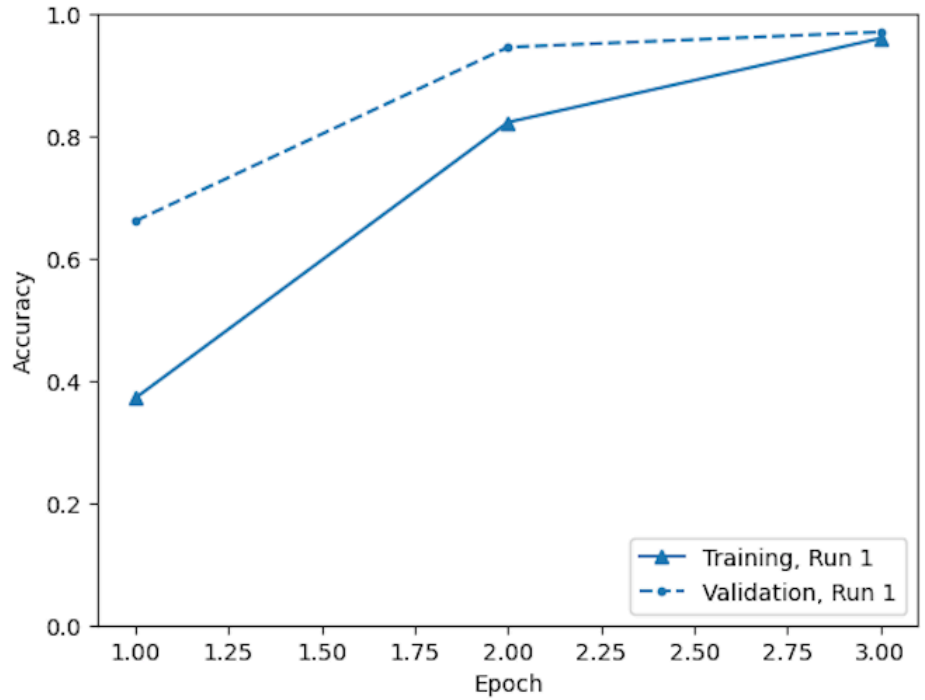
In [ ]:
```
faktor_q =          # Quarks
faktor_e =          # Elektronen
faktor_m =          # Myonen
faktor_t =          # Tauonen
eventliste_tv_vermehrt = augment_events(eventliste_tv, [faktor_q, faktor_e, faktor_m, faktor_t])
```

In [ ]:
```
modell.train(count_epochs=3)
```

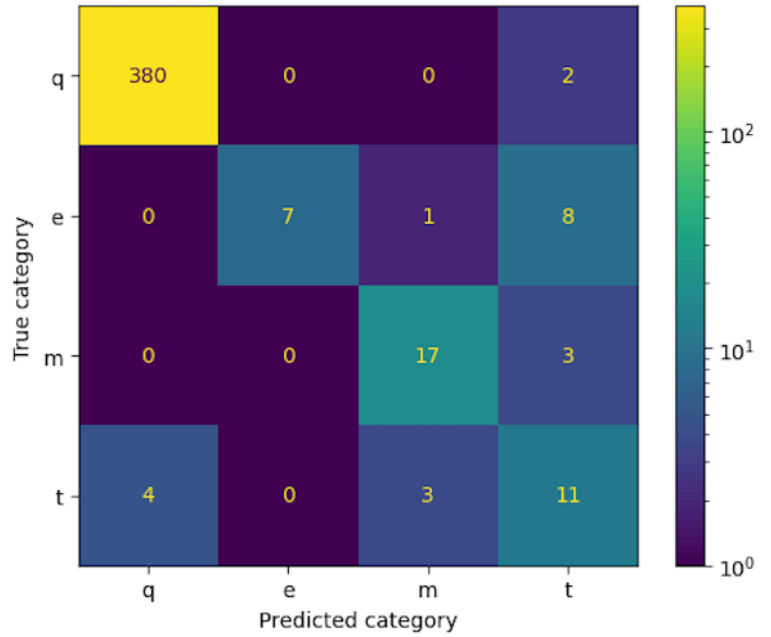
Machinelles Lernen mit OPAL Daten

- Ausgabe des CNN = Wahrscheinlichkeiten der Z-Boson Zerfallskanäle
- Vergleich mit Vorhersage auf AB & Rückbezug OPAL

```
[10]: modell.show_learning_curve()
```




```
[11]: eventliste_vorhersage = modell.predict(eventliste_test)
show_confusion_matrix(eventliste_test, eventliste_vorhersage)
```




```
Total prediction accuracy: 0.9518
```

Lernziele der ML Masterclass

- Teilchenphysik erlebbar machen
  - Standardmodell & Teilchenzerfälle am Bsp. des Z-Bosons verstehen
  - Aufbau Detektoren
  - Datenanalyse

- Was steckt hinter Neuronalen Netzwerken & Künstlicher Intelligenz
  - Anwendung & Grenzen von maschinellem Lernen
  - Knüpft über KI an Erlebnishorizont der Schüler an (ChatGPT etc.)

- **Interdisziplinär: verbindet Informatik & Physik**
  - Bedeutung von Programmieren als Methodik in der modernen Physik

NETZWERK TEILCHENWELT

UNIVERSITÄT BONN

Verzweigungsverhältnis der Zerfallskanäle des Z-Bosons

Das Standardmodell liefert Vorhersagen zum Verzweigungsverhältnis der Zerfallskanäle des Z-Bosons. Mit Hilfe von Messungen am OPAL-Detektor kann untersucht werden, ob die Beobachtungen im Experiment mit der Vorhersage übereinstimmen. Dies geschieht in drei Schritten:

1. Verzweigungsverhältnis der Zerfallskanäle des Z-Bosons theoretisch bestimmen
2. Daten der Messung auswerten
3. Vergleich der beiden Ergebnisse

Für die **Zerfallskanäle** des Z-Bosons gilt nach der Vorhersage des Standardmodells:

- 10 % : Zerfall in Elektron-Positron (e^+e^-), Myon-Antimyon ($\mu^-\mu^+$), Tau-Antitau ($\tau^-\tau^+$) (jeweils gleich wahrscheinlich)
- 20 % : Zerfall in Neutrino-Antineutrino ($\nu_e\overline{\nu_e}, \nu_\mu\overline{\nu_\mu}, \nu_\tau\overline{\nu_\tau}$) (jeweils gleich wahrscheinlich)
- 70 % : Zerfall in Quark-Antiquark ($q\bar{q}$) (15 Zerfallsmöglichkeiten, 5 Quarks à 3 Farben)

Neutrinos wechselwirken nicht mit dem OPAL-Detektor und können daher nur indirekt nachgewiesen werden. Im Rahmen der Messung kann daher nur der Zerfall des Z-Bosons in $e^+e^-, \mu^-\mu^+, \tau^-\tau^+$ und $q\bar{q}$ beobachtet werden. Um diese Beobachtungen mit der Vorhersage des Standardmodells zu vergleichen, muss daher aus den oben angegebenen Zerfallskanälen das beobachtete Verzweigungsverhältnis berechnet werden.

Aufgabe: Berechne das beobachtbare Verzweigungsverhältnis, also jeweils die Wahrscheinlichkeit mit der der Zerfall des Z-Bosons in $e^+e^-, \mu^-\mu^+, \tau^-\tau^+$ oder $q\bar{q}$ mit dem OPAL-Detektor beobachtet werden kann.

Tipp 1: Überlege, wie sich die Grundgesamtheit jetzt verändert. Was soll zusammen 100% ergeben?

Tipp 2: Stell dir vor du hast 1000 Z-Bosonen, die in $e^+e^-, \mu^-\mu^+, \tau^-\tau^+$ oder $q\bar{q}$ zerfallen. Wie oft erhältst du dann welche Paare?

Evaluation Testläufe an Schulen

Highlights

- "letztlich hat mir der detaillierte Einblick in die Teilchenphysik richtig gut gefallen“
- "auch fand ich cool, dass mir trotz keiner Kenntnisse in Informatik das Jupyter-Programm total Spaß gemacht hat“
- "Ich hab vieles verständlich erklärt bekommen, von dem ich mich immer ferngehalten habe, weil ich nicht wusste wie es funktioniert (Informatik)“
- "vielseitig, Bezug zur Informatik, tolle Ergänzung zum Physikunterricht, Erweiterung des Horizonts“

Lowlights

- „Fehlen von Zusammenhängen mit aktueller Forschung“

Zusammenfassung & Ausblick

- ML-Masterclass didaktisch weiterentwickelt (Arbeitsblätter, Spiele, detaillierter Ablaufplan etc.)
- Erfolgreiche Anwendung mit Schulklassen
  - Sehr positives Feedback
  - Interdisziplinär: Verbindung zur Informatik
- Was nun?
  - Zentralen Server/Jupyter Hub? → von jeder Schule über Browser erreichbar
  - Webseite anlegen mit allen Materialien, die für MC genutzt werden kann
  - Durchführung in KI-Ausstellung im Deutschen Museum Bonn

- Für die Zukunft: MC mit aktuelleren Event Displays denkbar (z.B. Belle II, …)?

- Danke auch an Johanna Rätz (Uni Bonn), Barbara Valeriani-Kaminski (Uni Bonn), David Borgelt (Uni Münster), Nicolas Tiltmann (Uni Münster), Oliver Freyermuth (Uni Bonn)

Zusammenfassung & Ausblick

- ML-Masterclass didaktisch weiterentwickelt (Arbeitsblätter, Spiele, detaillierter Ablaufplan etc.)
- Erfolgreiche Anwendung mit Schulklassen
  - Sehr positives Feedback
  - Interdisziplinär: Verbindung zur Informatik
- Was nun?
  - Zentralen Server/Jupyter Hub? → von jeder Schule über Browser erreichbar
  - Webseite anlegen mit allen Materialien, die für MC genutzt werden kann
  - Durchführung in KI-Ausstellung im Deutschen Museum Bonn

- Für die Zukunft: MC mit aktuelleren Event Displays denkbar (z.B. Belle II, …)?

- Danke auch an Johanna Rätz (Uni Bonn), Barbara Valeriani-Kaminski (Uni Bonn), David Borgelt (Uni Münster), Nicolas Tiltmann (Uni Münster), Oliver Freyermuth (Uni Bonn)

Danke für die Aufmerksamkeit!

NETZWERK TEILCHENWELT

UNIVERSITÄT BONN

Backup

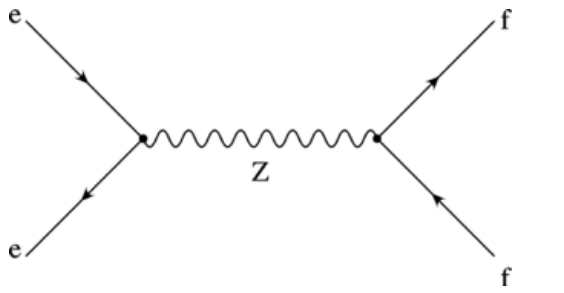
Detaillierter Ablauf der ML-Masterclass / Zeitplan

Zeit	Inhalte
15 Min	Begrüßung
60 Min	**Einführungsvortrag Standard Modell** (AB Verzweigungsverhältnis)
15 Min	Pause
45 Min	**Vortrag Detektoren & Beschleuniger** (Aufgabe zum Detektor)
15 Min	Pause
45 Min	**Machine Learning & Neuronale Netzwerke** (Spiel “Mensch, Maschine!”)
15 Min	**Einführung in den Code**
30 Min	Einführung **Jupyter**
45 Min	Mittagspause
75 Min	**Arbeiten mit dem Code** in 2er Gruppen
10 Min	Besprechung der AB, **Diskussion Hyperparameter und Ergebnisse des KNN**
10 Min	Rückbezug zum OPAL Experiment & dem Verzweigungsverhältnis
10 Min	**Gesellschaftlicher Aspekt von ML**
10 Min	Evaluation

Dauer ca. 7h - z.B. 8.30-15.30

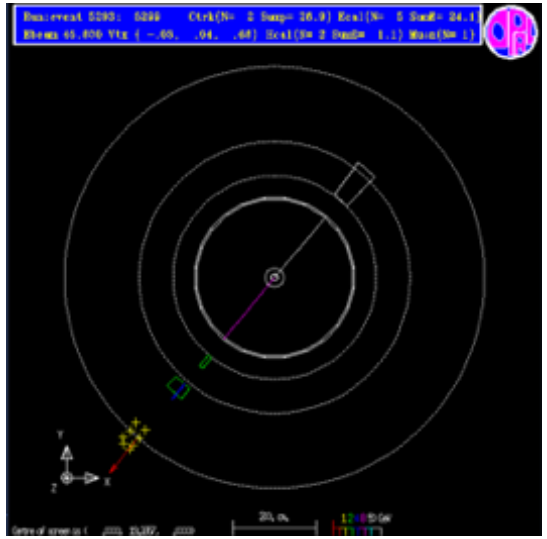
Zerfall des Z-Bosons

- 4 Möglichkeiten:



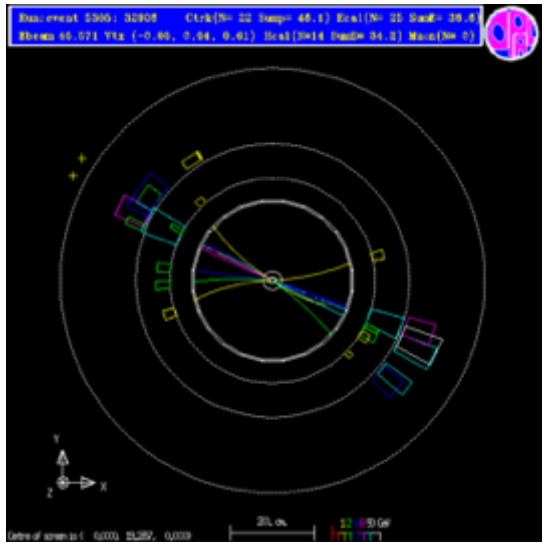
$f = \mathrm{e},\ \mu,\ \tau$ oder Quarks

- Tau sind sehr kurzlebig
  - $\tau = 2{,}9 * 10^{-13}\mathrm{s} = 0.000000000000029\mathrm{s}$
  - Zerfallen direkt am Entstehungspunkt
- Zerfallen in e, μ oder Quarks



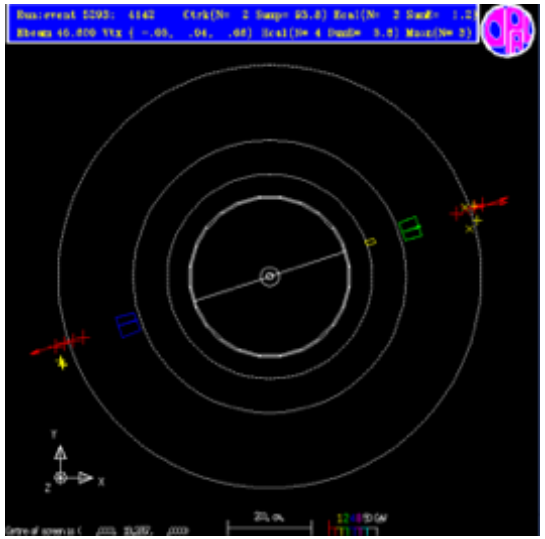
$\tau^+\tau^-$
Ca. 4%

e^+e^-
Ca. 4%



Quarks
Ca. 88%

$\mu^+\mu^-$
Ca. 4%



Das Neuronale Netz

- Convolutional neural network
- Zufälliges Aufteilen der Daten in Trainings- und Test-Daten
- Vervielfältige Events für das Training durch zufälliges Drehen und Spiegeln

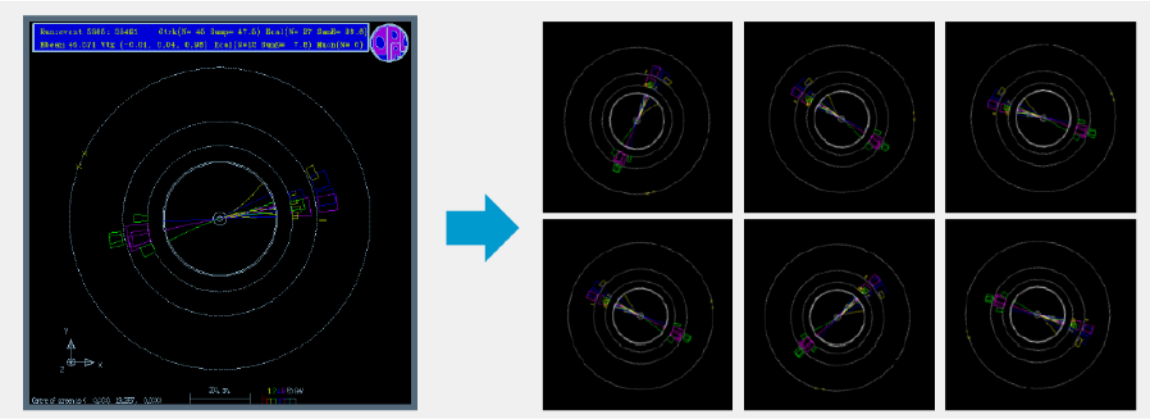
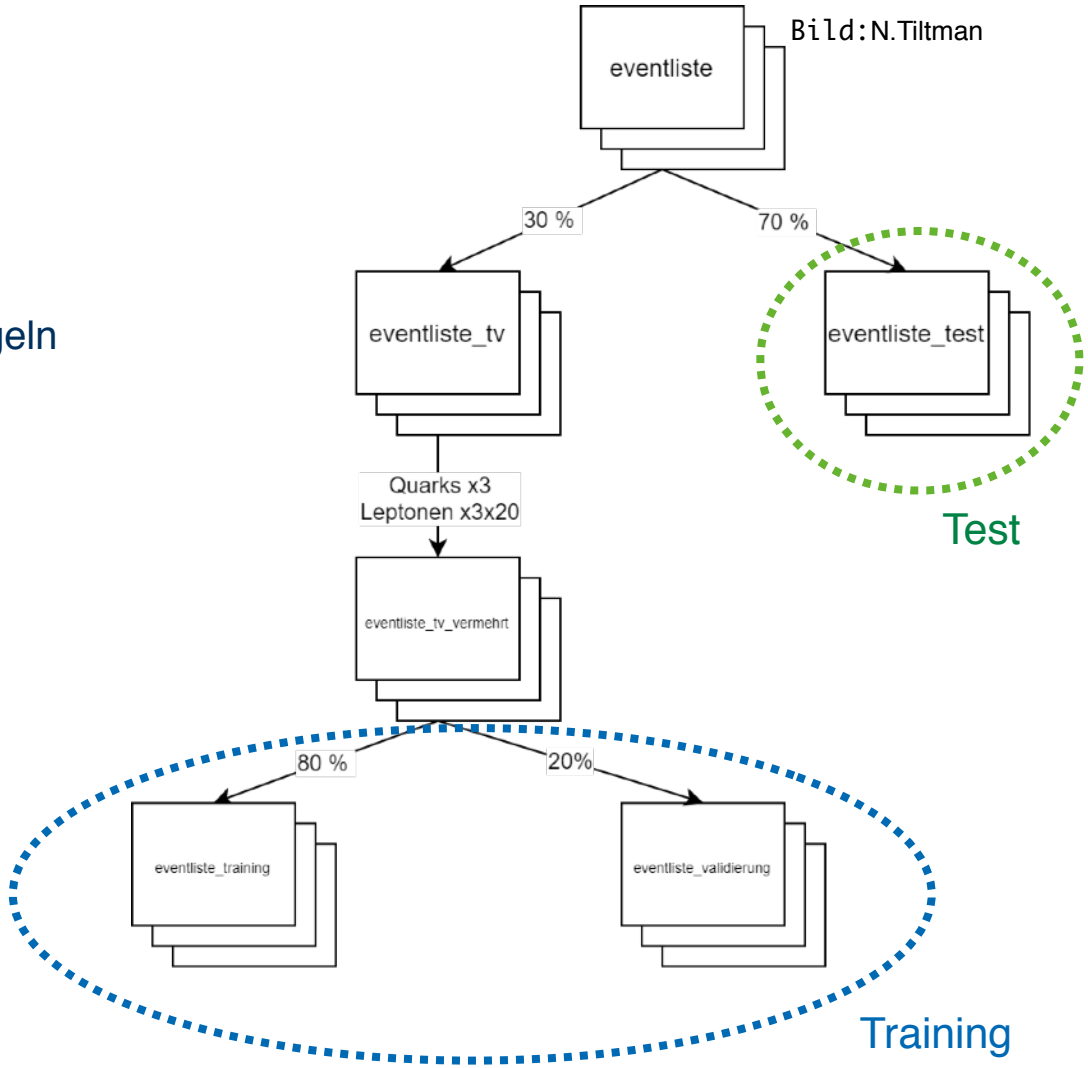


Bild:N.Tiltman

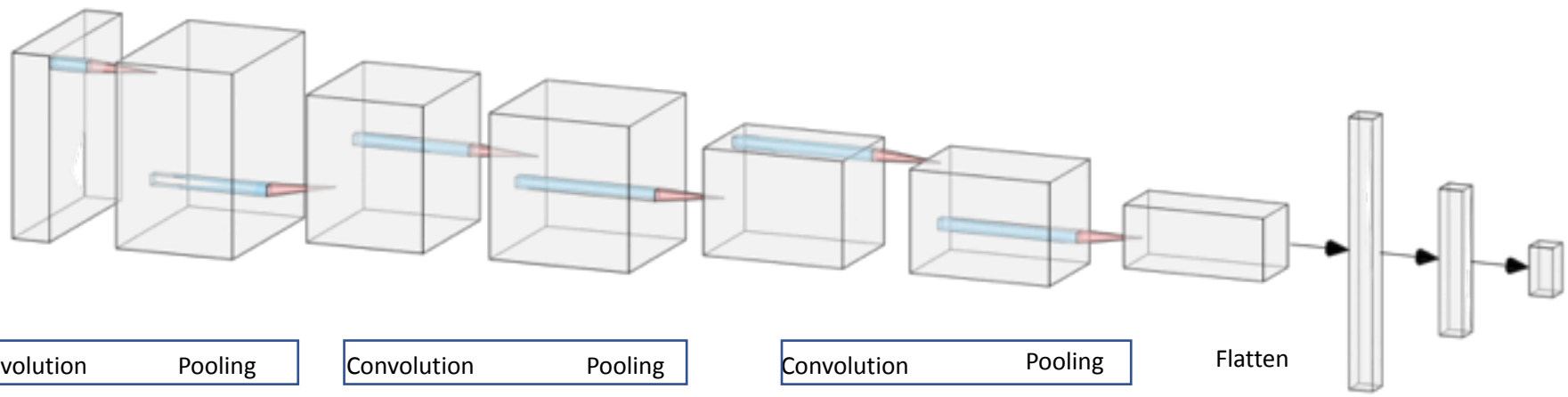
- Für Training braucht man getrennte Trainings- und Validierungs-Daten:
  - Großteil für Training
  - Überprüfe Erfolg mit Validierungs-Daten



Convolutional Neural Network

- 3 Ebenen (jew. convolution & pooling layer)
- 2 vollverbundene Ebenen

Eingabe:
200x200px x 3 Farben



Ausgabe:
4 Neuronen
entsprechend
den 4 Klassen

Voll verbundenes Netzwerk
(wie im Film)