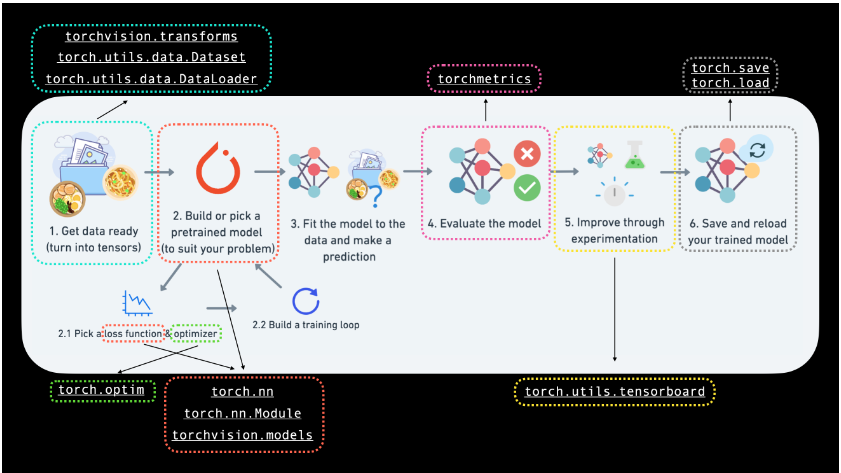
**Computer Vision**

**Notes:**

Computer Vision ist die Tatsache, dass man einem Computer lehrt, zu sehen.

Das besteht ( binary classification, multi-class classification, objet detection, panoptic segmentation)



Als Datensatz habe ich FashionMNIST verwendet.

MNIST steht für Modified National Institute of Standards and Technology. Der ursprüngliche MNIST-Datensatz enthält Tausende von Beispielen für handgeschriebene Zahlen (von 0 bis 9) und wurde zur Erstellung von Computer-Vision-Modellen verwendet, um Nummern von Postdiensten zu identifizieren. FashionMNIST, hergestellt von Zalando Research, ist eine ähnliche Konfiguration.

In unserem Dataset haben wir es mit 10 verschiedenen Arten von Kleidung zu tun. Das bedeutet, dass unser Problem eine Mehrklassenklassifikation ist.

In diese Aufgabe 3 unterschiedlichen Modellen wurden trainiert, damit zu sehen, welchen Modellen performs am besten:

model\_0 - unser Basismodell mit zwei nn.Linear()-Schichten.

model\_1 - der gleiche Aufbau wie unser Basismodell, jedoch mit nn.ReLU()-Schichten zwischen den nn.Linear()-Schichten.

model\_2 - unser erstes CNN-Modell, das die TinyVGG-Architektur auf der CNN Explainer Website nachahmt

Ein Bild, das Text enthält.

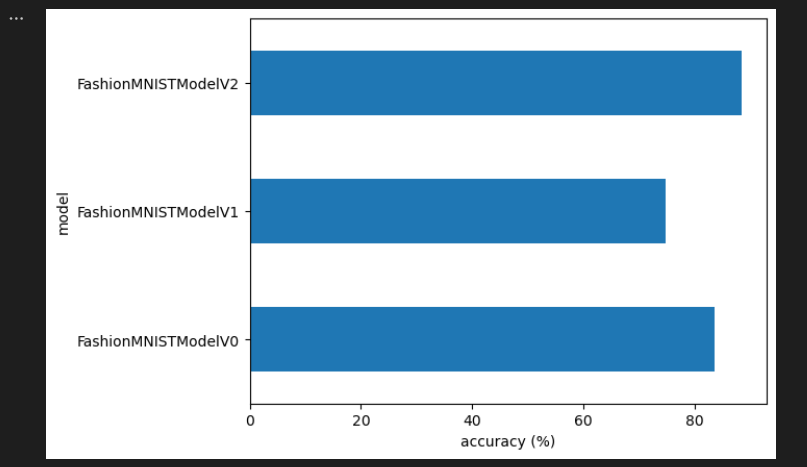
Automatisch generierte Beschreibung

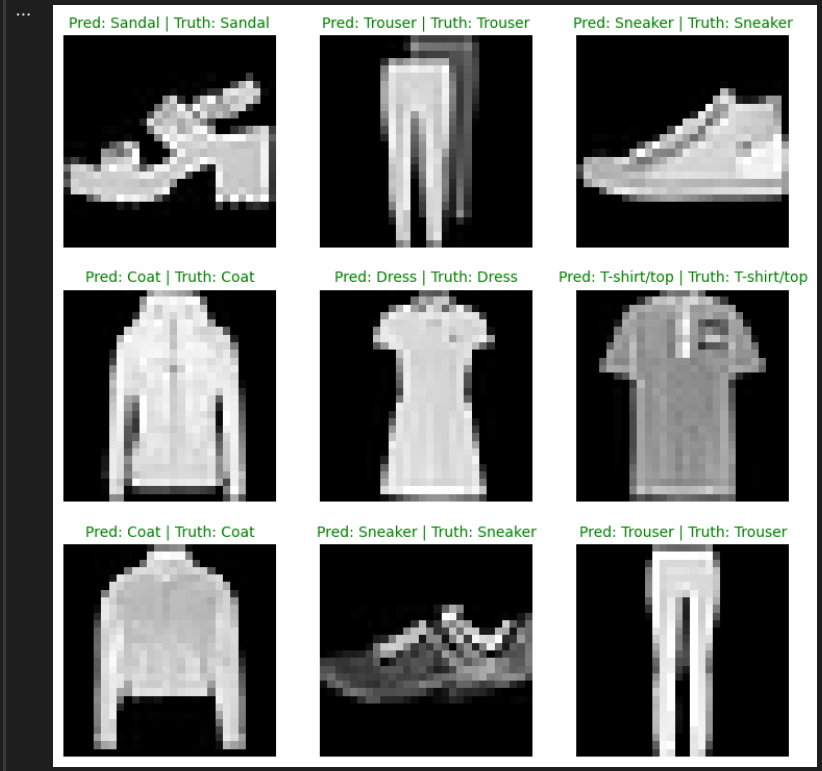
Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Der CNN (FashionMNISTModelV2) modelle performed the best (lowest loss, highest accuracy) but had the longest training time.

And der baseline model (FashionMNISTModelV0) performed better than model\_1 (FashionMNISTModelV1).





Während ich an dieser Arbeit arbeitete, wurde ich mit dem Problem des Overfitting konfrontiert, da es schien, dass mein Modell die Trainingsdaten überpasste.

Overfitting bedeutet, dass unser Modell die Trainingsdaten gut lernt, aber diese Modelle lassen sich nicht auf Testdaten verallgemeinern.

* Building a Convolutional Neural Network (CNN)

CNN sind für ihre Fähigkeit bekannt, Modelle in visuellen Daten zu finden.

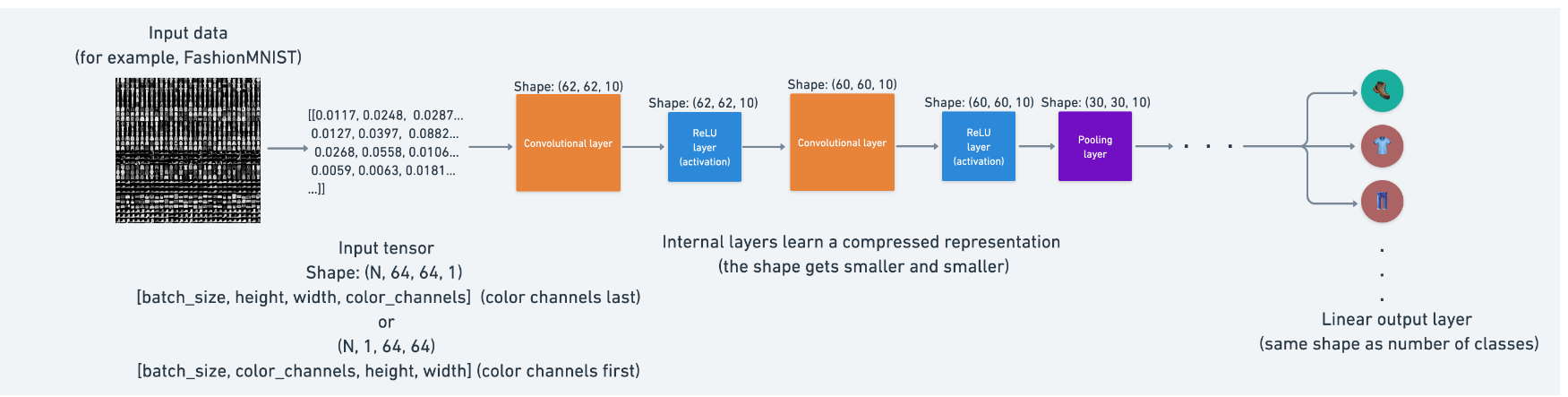
Es folgt der typischen Struktur eines faltenden neuronalen Netzes

Input layer -> [Convolutional layer -> activation layer -> pooling layer] -> Output layer

Der Inhalt von [Convolutional layer -> activation layer -> pooling layer] kann je nach Bedarf erhöht und mehrfach wiederholt werden.

*Frage: Kann man CNN auch für strukturierte Daten verwenden, z. B. row and column date?*

*Jede Schicht eines neuronalen Netzes versucht, Daten aus einem höherdimensionalen Raum in einen niedrigerdimensionalen Raum zu komprimieren.*

**