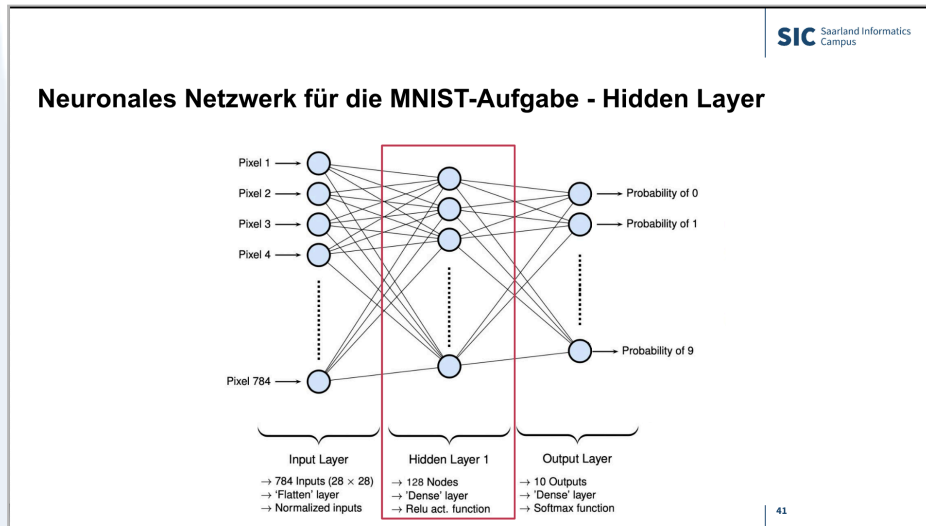




# Neuronale Netze kennenlernen und programmieren

Heute hast Du alle Bestandteile und Parameter kennengelernt, die benötigt werden, um ein Künstliches Neuronales Netz (KNN) für das MNIST-Dataset zu definieren.



Anschließend wurde das Neuronale Netz in Python mit der Bibliothek PyTorch implementiert. Programmiert wurde in einem Jupyter Notebook in Google Colab.

```
Neuronales Netz für das MNIST-Dataset.ipynb
Datei Bearbeiten Anzeige Einfügen Laufzeit Tools Hilfe Alle Änderungen wurden gespeichert
Kommentar Teilen
Verbinden

+ Code + Text

Implementierung eines Künstlichen Neuronales Netzes (KNN) zum MNIST-Dataset

Block 1: Definition der Klasse für das Neuronale Netz und Model instantiieren

[ ] 1 # Version 1.0
2 # Block 1
3 import torch
4 import torch.nn as nn
5
6 # Definiere das neuronale Netzwerk
7 class SimpleNN(nn.Module): # erbt von nn.Module
8     def __init__(self):
9         super(SimpleNN, self).__init__()
10        self.fc1 = nn.Linear(28*28, 128) # fc1 = fully connected layer, 28x28=784 in, 128 out
11        self.relu = nn.ReLU() # Rectified Linear Unit (gerichtetes lineares Element)
12                               # f(x) = max(0, x)
13        self.fc2 = nn.Linear(128, 10) # 128 in, 10 out
14
15    def forward(self, x):
16        x = x.view(-1, 28*28) # Flattening der Eingabe
17        x = self.fc1(x)
18        x = self.relu(x)
```

Alle Links zum Projekt findest Du auf der Webseite des InfoLab: [infolab.cs.uni-saarland.de](http://infolab.cs.uni-saarland.de)

Wir freuen uns, dass Du am Workshop teilgenommen hast!

