# Neural Network with Hidden Layers

By Benedikt Willecke IES19372

## 1) Original training & validation accuracy (15 epochs; 0,0001 learning rate)

```
epoch : 15
training loss: 0.2168, acc 93.9333
validation loss: 0.2143, validation acc 94.0300
```

## 2) Best activation function for the first two hidden layers

```
[149] class Classifier(nn.Module):

          def __init__(self, D_in, H1, H2, D_out):
              super().__init__()
              self.linear1 = nn.Linear(D_in, H1)
              self.linear2 = nn.Linear(H1, H2)
              self.linear3 = nn.Linear(H2, D_out)
          def forward(self, x):
              x = F.tanh(self.linear1(x))
              x = F.tanh(self.linear2(x))
              x = self.linear3(x)
              return x
```

```
epoch : 15
training loss: 0.0818, acc 97.7067
validation loss: 0.0938, validation acc 97.2700
```

## 3) Best number of hidden lay

```
class Classifier(nn.Module):

    def __init__(self, D_in, H1, H2, H3, D_out):
        super().__init__()
        self.linear1 = nn.Linear(D_in, H1)
        self.linear1_2 = nn.Linear(H1, H2)
        self.linear2 = nn.Linear(H2, H3)
        self.linear3 = nn.Linear(H3, D_out)
    def forward(self, x):
        x = F.relu(self.linear1(x))
        x = F.relu(self.linear1_2(x))
        x = F.relu(self.linear2(x))
        x = self.linear3(x)
        return x
```

```
model = Classifier(784, 1024, 512, 128, 10)
```

epoch : 15

training loss: 0.0256, acc 99.2417

validation loss: 0.0652, validation acc 98.0200

## 4) Reason for having training, validation and test dataset

**Training set:**
A set of data used to make the NN learn and fit the parameters to the dataset

**Validation set:**
A set of data used to prevent overfitting of the model to the training dataset in order to get a NN that works for in a general way.

**Test set:**
A set of data to assess the performance of a fully trained NN. After this assessment no further tuning of the model is necessary.

**Why separate the datasets?**
In order to get a higher accuracy and to prevent bias and overfitting