

# Übungsblatt 4

Abgabe: 21.07

## Aufgabe 1: CNN beim MNIST Datensatz

Basierend auf der Aufgabe 1 des letzten Übungsblattes sollen Sie den MNIST Datensatz nun noch einmal mit einem CNN testen. Das Laden des Datensatzes ist dabei identisch.

- a) Erstellen Sie die grundlegende Architektur eines CNN. Ein Vorschlag wäre:

```
model = Sequential()
model.add(Conv2D(28, kernel_size=(3,3), input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation=tf.nn.relu))
model.add(Dropout(0.2))
model.add(Dense(10, activation=tf.nn.softmax))
```

Dieser Aufbau entscheidet über die Performance Ihres Neuralen Netzwerkes. Die 10 in "Dense" muss so beibehalten werden, da Sie am Ende entscheiden möchten, welche der Klassen 0-9 nun am besten getroffen wurde.

Trainieren können Sie Ihr Netzwerk durch

```
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
model.fit(x=x_train,y=y_train, epochs=10)
model.evaluate(x_test, y_test)
```

Die Funktion "evaluate" wird Ihnen die Genauigkeit Ihres Modells ausgeben.

- b) Testen Sie folgende Parameter und deren Kombination
- Kernelgrößen 3-5
  - Poolinggrößen 3-5
  - Metriken aus der Vorlesung

Dokumentieren Sie Ihre Beobachtungen bezüglich der Parameter.

- c) Wählen Sie aus Ihren Modellen das Beste und schauen Sie die Bilder, die vom Netzwerk falsch klassifiziert wurden. Was stellen Sie fest?

## Aufgabe 2: Implementieren der Principal Component Analysis

Implementieren sie die PCA aus der Vorlesung. Gehen sie dabei von einem  $n$ -dimensional Input aus. In ihrer Funktion sollen sie dann auswählen können, wie viele Dimensionen sie bei der Reduktion erhalten wollen. Testen und zeigen sie das Ergebnis ihrer Implementierung anhand eines willkürlich erzeugten 2-dimensionalen Datensatzes.