

Übungsblatt: 5

Bearbeitung am 4. Juni

Aufgabe 1: Einrichten von Tensorflow und mehr

Installieren Sie Tensorflow, Keras, Matplotlib und scikit-learn auf Ihrem Rechner. Das Betriebssystem ist egal, auf älteren Rechnern muss eventuell eine ältere Tensorflowversion, z. B. 1.7 oder 1.3, installiert werden (`pip install tensorflow==1.3`) Optional: Installieren Sie alle benötigten Bibliotheken, um Tensorflow auf Ihrer GPU laufen zu lassen.

- <https://www.tensorflow.org/install>
- <https://www.tensorflow.org/install/gpu>

Bei der Installation des Tensorflow-Paketes wird automatisch Keras mitinstalliert. Installieren Sie `matplotlib` und `scikit-learn` analog zu Tensorflow mittels `pip install matplotlib` bzw. `pip install scikit-learn`.

Aufgabe 2: MNIST-Training

- Führen Sie das MNIST-Tutorial für Keras lokal auf ihrem Rechner aus (<https://www.tensorflow.org/tutorials>).
- Am Ende wird im Tutorial auf den Test-Daten evaluiert. Evaluieren Sie zusätzlich auf den Trainingsdaten und vergleichen Sie die beiden Ergebnisse. Was fällt Ihnen auf und wie können Sie das erklären?
- Fügen Sie eine weitere Schicht mit 512 Knoten ein und vergleichen Sie die Ergebnisse mit den Standardeinstellungen.
- Wandeln Sie das MLP in ein einfaches Perzeptron um, d.h. konstruieren Sie das Netz so, dass es nur aus einer Input- und einer Output-Schicht besteht, und vergleichen Sie die Ergebnisse mit den Standardeinstellungen.
- Verwenden Sie anstatt des ReLU eine tanh-Aktivierungsfunktion und vergleichen Sie die Ergebnisse
- Speichern Sie beispielsweise Ihr bestes Ergebnis ab (`model.save_model`)

Aufgabe 3: MNIST-Evaluation

- Erstellen Sie ein weiteres Skript, in welchem Sie das bereits trainierte Modell laden (`model.load_model`) und zum Überprüfen auf den Test-Daten erneut evaluieren (es muss das selbe Ergebnis liefern).
- Werfen Sie einen Blick in die Dokumentation des Keras-Modells (`tf.keras.model` auf <https://www.tensorflow.org> suchen). Sie suchen eine Funktion um die Klasse jedes Testbeispiels vorherzusagen.
- Berechnen Sie nun Manuell (ohne Verwendung der `model.evaluate`-Funktion) die Genauigkeit und vergleichen Sie den Wert mit dem aus Keras berechnet.

- (d) Berechnen Sie für jede Klasse k die individuellen Precision, Recall und F_1 -Scores. Zählen Sie als TP, FP und FN jeweils wie in der Vorlesung besprochen.
- (e) Berechnen Sie die Konfusionsmatrix M_{ij} , die zählt, wie oft Zahlen miteinander verwechselt wurden. D. h. M_{00} zählt alle korrekten Nuller, M_{01} zählt wie oft eine 1 anstatt einer Null vorhergesagt wurde.
- (f) Plotten Sie die Matrix als Bild mittels Matplotlib. Wo treten die häufigsten Verwechslungen auf? (Eventuell müssen Sie die Diagonale auf 0 setzen, damit man mehr erkennt.)
- (g) Nutzen Sie die in scikit-learn.metrics (importieren als import sklearn.metrics) verfügbaren Funktionen, um die Konfusionsmatrix und Metriken zu berechnen und vergleichen Sie mit Ihren Ergebnissen.