

**Vorlesung:** Prof. Dr. Paul Lukowicz

**Übungen:** M.Sc Peter Hevesi, Kunal Oberoi, M.Sc Vitor Fortes, B.Sc Matthias Tschöpe

*Hinweis: Alle Programmieraufgaben sind in Python abzugeben.*

### Aufgabe 4.1

In der Übung haben Sie bereits gelernt wie Sie mit Scikit-learn einen Klassifizierer für handgeschriebene Zahlen implementieren können. Verwenden Sie diesen Code um einen Klassifizierer für das Fashion-MNIST Dataset zu implementieren. Die Daten wurden bereits so konvertiert, dass diese, genau wie in den Übungen, mit `np.load(...)` eingelesen werden können.

Implementieren Sie ein Multilayer Perzeptron, das auf den Testdaten reproduzierbar einen *Score* von  $\geq 0.885$  erreicht. Verwenden Sie als *Score*, die in Scikit-learn bereitgestellte Funktion `score(X, y)`. Diese Funktion kann direkt auf dem Multilayer Perzeptron `mlp` aufgerufen werden (D.h. `score = mlp.score(X_text, y_test)`). Folgender Aufbau kann als Grundgerüst für das Multilayer Perzeptron verwendet werden. Dieses erreicht jedoch nicht die gewünschte Qualität.

- Architektur: (128, 64, 32)
- Aktivierungsfunktion: ReLU
- Optimierer: Adam
- Batch size: 8
- Lernrate: 0.1
- maximale Anzahl an Epochen: 5

Lesen Sie ggf in der Dokumentation<sup>1</sup> nach, wie die Parameter gesetzt werden und welche Möglichkeiten es noch gibt. Erstellen Sie auch eine Confusion Matrix und beschreiben Sie die Ergebnisse mit Ihren eigenen Worten.

Hinweis: Falls Sie sich nicht mehr an den Aufbau eines Klassifizierers mit Scikit-learn erinnern, finden Sie hier<sup>2</sup> nützliche Informationen.

---

<sup>1</sup>[https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html)

<sup>2</sup>[https://scikit-learn.org/stable/modules/neural\\_networks\\_supervised.html](https://scikit-learn.org/stable/modules/neural_networks_supervised.html)

### Aufgabe 4.2

Ein binärer Klassifizierer  $K_1$  liefert auf den Validierungsdaten die nachfolgende Confusion Matrix.

		Ground Truth	
		Klasse A	Klasse B
Predicted	Klasse A	100	8
	Klasse B	2	5

Beantworten Sie die folgenden Fragen:

- (a) Wie viele Daten gehören zur Klasse A und wie viele zur Klasse B?
- (b) Berechnen Sie Precision, Recall, Accuracy und den  $F_1$ -Score
- (c) Ist eine dieser Messungen ausreichend um die Qualität des Klassifizierers zu beschreiben? Wenn ja, welche Messung beschreibt die Qualität des Klassifizierers am besten. Falls nicht was ist das Problem?
- (d) Angenommen wir haben einen zweiten Klassifizierer  $K_2$  der über eine Gleichverteilung zufällig entscheidet ob ein Sample zur Klasse A oder B gehört. Welche Accuracy hat  $K_2$ ?
- (e) Welcher Klassifizierer ist für die korrekte Klassifizierung von Samples aus der Klasse B besser,  $K_1$  oder  $K_2$ ?
- (f) Was würden Sie empfehlen, um  $K_1$  zu verbessern?

Falls Ihnen nicht alle Messungen bekannt sein sollten, dann sehen Sie hier nach<sup>3</sup>.

---

<sup>3</sup>[https://en.wikipedia.org/wiki/Confusion\\_matrix](https://en.wikipedia.org/wiki/Confusion_matrix)