

Arbeitsblatt: Einführung in Neural Networks mit WebNet

Herzlich willkommen zu diesem Arbeitsblatt, das sich mit Neural Networks und Supervised Learning befasst! In den folgenden Aufgaben (A und B) werden wir einen Blick auf die Grundlagen dieser faszinierenden Technologien werfen und dabei insbesondere die von mir im Rahmen meiner Masterarbeit entwickelte Plattform, **WebNet**, kennenlernen.

URL: <https://webnet-builder-main.infra.wogra.com/>

Aufgabe A – Titanic (Klassifikation):

Am 15. April 1912 sank die als "unsinkbar" geltende Titanic während ihrer Jungfernfahrt, nachdem sie mit einem Eisberg kollidiert war. Leider gab es nicht genügend Rettungsboote für alle an Bord, was zum Tod von 1502 der insgesamt 2224 Passagiere und Crewmitglieder führte.

Deine Aufgabe ist es ein Vorhersagemodell zu erstellen, das die Frage beantwortet: "**Welche Gruppe von Personen hatte eine höhere Überlebenschance?**" Dafür kannst du die folgenden Passagierdaten verwenden:

Variable	Definition	Information
survival	Überleben	0 = Nein, 1 = Ja
pclass	Ticket class	1 = 1. Klasse, 2 = 2. Klasse, ...
sex	Geschlecht	Dezimalzahl wenn < 1 oder geschätzt.
age	Alter	
sibsp	Geschwister/Ehepaare	Anzahl
parch	Eltern/Kinder	Anzahl
ticket	Ticketnummer	
fare	Fahrpreis	
cabin	Kabinennummer	
embarked	Einsteigehafen	C = Cherbourg, Q = Queenstown, ...

Bevor du die folgenden Schritte durchführen kannst, musst du zunächst ein neues Projekt mit dem Namen deiner Wahl erstellen!

1. Datensatz

- Importiere den Titanic-Datensatz (**titanic-train.csv**).
- Untersuche die ersten Zeilen des Datensatzes, um seine Struktur zu verstehen. Nicht alle Features sind gleich wichtig für die Vorhersage!
- Entscheide ob, und wenn ja welche Werte du **skalieren** / **enkodieren** möchtest. (Select-Element im Header der Tabelle)

- Entscheide, welche Spalten du als **Eingabe** für das Modell verwenden möchtest und welcher als **Vorhersagewert** dienen soll.
- Bestimme das Verhältnis zwischen **Trainings-** und **Validierungsdaten**.

2. Modellierung

- Konstruiere das folgende Modell
 - **Inputlayer**: (Shape = Anzahl deiner ausgewählten Merkmale, beachte dass One-Hot Encoder die Anzahl der Features erhöhen.)
 - Hidden Layer: 2 **Dense-Layer** (128 und 64 Neuronen und Relu-Aktivierungsfunktion)
 - **Outputlayer** (Sigmoid-Aktivierungsfunktion)

3. Training

- Stelle die Trainingsparameter wie folgt ein:
 - **Epochen**: 100, **Batch-Größe**: 32, **Lernrate**: 0,01, **Optimierer**: Adam, **Verlustfunktion**: MSE, **Shuffle** aktivieren.
- Trainiere dein Modell und überwache dabei die Genauigkeit (**Accuracy**) für das Training sowie die Validierung in der Visualisierung.
- Ändere die Einstellungen deines Modells, wie zum Beispiel die Lernrate, Batch-Größe und die Anzahl der Neuronen in den versteckten Schichten, und führe anschließend weitere Trainingseinheiten durch.

4. Evaluation

- Falls du mehrere Trainingsläufe durchlaufen hast, kannst du sie hier miteinander vergleichen, indem du dir die Visualisierungen ansiehst und die Accuracy & Val_Accuracy miteinander vergleichst.
- **Vorhersage**: Lade dein bestes Modell und lasse dein Modell Samples aus deinem Datensatz vorhersagen und vergleiche die Vorhersage mit dem tatsächlichen Wert.
 - Experimentiere jetzt mit unterschiedlichen Eingabedaten, um zu ermitteln, ob eine von dir konzipierte fiktive Person das Unglück der Titanic überlebt hätte!

Aufgabe B – Boston House Prices (Regression):

Wir werden nun ein neues Szenario betrachten und uns dem Boston Housing Price Problem zuwenden. Hierbei geht es darum, ein Modell zu entwickeln, das den Preis von Immobilien in Boston vorhersagt, basierend auf verschiedenen Merkmalen. Im Gegensatz zur vorherigen Aufgabe konzentrieren wir uns hier auf die Analyse von Hauspreisen. Folgende Features können dafür genutzt werden:

Variable	Definition
CRIM	Pro-Kopf-Kriminalitätsrate in der Stadt
ZN	Anteil an Wohnland für Grundstücke > 25.000 sq.ft.
INDUS	Anteil nicht-einzelhandels Gewerbeflächen
CHAS	Charles River Variable (1 für Flussufer, sonst 0)
NOX	Stickoxid-Konzentration (Teile pro 10 Millionen)
RM	Durchschnittliche Zimmeranzahl pro Wohnung
AGE	Anteil vor 1940 erbaute Eigentumswohnungen
DIS	Gewichtete Entfernungen zu fünf Beschäftigungszentren
RAD	Index der Zugänglichkeit zu radialen Autobahnen
TAX	Grundsteuersatz pro 1000 Dollar
PTRATIO	Schüler-Lehrer-Verhältnis nach Region
B	Anteil afroamerikanischer Bevölkerung
LSTAT	Prozentsatz sozial schwächerer Bevölkerung
MEDV	Preis von Häusern in Tausend Dollar

Erstelle ein neues Projekt mit dem Namen deiner Wahl!

1. Datensatz

- Importiere den Boston-Datensatz (**boston-train.csv**).
- Untersuche die ersten Zeilen des Datensatzes, um seine Struktur zu verstehen.
- Entscheide ob, und wenn ja welche Werte du **skalieren** / **enkodieren** möchtest.
- Entscheide, welche Spalten du als **Eingabe** für das Modell verwenden möchtest und welcher als **Vorhersagewert** dienen soll.
- Bestimme das Verhältnis zwischen **Trainings-** und **Validierungsdaten**.

2. Modellierung

- Überlege dir eine Architektur für dein Modell. Verwende dafür ausschließlich **Basic-Layer**.
- Überlege dir, ob und wenn ja, wie du die **Einstellungen** der einzelne Layer verändern möchtest.

3. Training

- Du hast die Freiheit alle **Trainingsparameter** so einzustellen, wie du es für richtig erachtest.
- Trainiere dein Modell und überwache dabei den Verlust (**Loss**) für das Training sowie die Validierung in der Visualisierung.

4. Evaluation

- Hier kannst du nun unterschiedliche Trainingsläufe analysieren und vergleichen. Verändere Hyperparameter und sehe dir hier die Auswirkungen davon an. Versuche ein möglichst **optimales Modell** zu konstruieren. Du hast auch die Möglichkeit vergangene Trainings zu laden und mit deren trainierten Gewichte weiter zu trainieren.
- **Vorhersage (Optional):** Experimentiere mit der Anpassung einzelner Merkmale, um ein tieferes Verständnis dafür zu entwickeln, wie signifikant **der Einfluss jedes einzelnen Merkmals** auf das Ergebnis ist.