**Entwicklung neuronales Netz zur Wettervorhersage**

**Ziel**: Vorhersage der Temperatur in Celsius und der Menge an Niederschlag in Millimeter für den nächsten Tag anhand diverser Daten der vorherigen Tage

**Entwicklungsumgebung:** Das neuronale Netz ist in der Programmiersprache Python 3.8 geschrieben. Für die Entwicklung des neuronalen Netzes habe ich als Entwicklungsumgebung Visual Studio Code mit einem Python-Plugin benutzt.

**Datenakquise:** Bevor mit der Implementierung des neuronalen Netzes begonnen werden konnte, mussten erst genügend Daten gesammelt werden. Dafür habe ich die World Weather Online API genutzt (<https://www.worldweatheronline.com/developer/>, Stand: 21.12.2019). Mittels Get-Requests habe ich die benötigten Daten in XML-Form erhalten. Jedoch konnte ich pro Request nur die Daten für einen Monat erhalten, deshalb habe ich 10 einzelne Requests verschickt um die kompletten Daten für die Monate Januar bis Oktober 2019 zu erhalten.

**Datenverarbeitung:** Die 10 XML-Dateien habe ich zu einer einzigen zusammengefügt. Aus dieser XML-Datei wurden die benötigten Daten anschließend in eine CSV-Datei geparst und gespeichert. Diese CSV-Datei konnte ich nun zur Entwicklung des neuronalen Netzes zur Wettervorhersage nutzen.

**Erstellung eines Datenframes:** Mittels der Bibliothek ‚pandas‘ habe ich aus der CSV-Datei ein Datenframe erstellt. In diesem Datenframe haben sich nach einem weiteren Aufräumprozess nur noch Spalten, welche entweder zur Kategorie ‚Feature‘, ‚Label‘ oder ‚Index‘ zugewiesen wurden, befunden (Index ist in meinem Fall die Spalte ‚date‘). Zuletzt habe ich dem Datenframe noch die Spalte ‚Label‘ hinzugefügt, diese Spalte mit den Werten des zu bestimmenden Attributes gleich gesetzt, wie z.B der Temperatur und die Werte dieser Spalten wurden um die vorherzusagende Reichweite verschoben. Diese Reichweite beträgt in meinem Fall 8, da ein Tag aus 8 Einträgen besteht. Die leeren Spalten wurden anschließend entfernt.

**Trennung der Daten in Feature und Label + Generierung von Test- und Trainings-Sets:** Ein Model zur Vorhersage von Daten braucht Features und Labels um zu funktionieren. Bei dem aufgeräumten Datenframe ist jede Spalte ein Feature, welches weder Label noch Index ist. Ich habe ein Array bestehend aus den Werten der Features und ein Array bestehend aus den Werten der Labels gemacht. In diesem Fall wurde immer jeweils nur ein Label benutzt. Sowohl das Feature-Array, als auch das Label-Array wurde mittels der Methode ‚train\_test\_split()‘ aus der Bibliothek ‚sklearn‘ in ein Test- und ein Trainings-Set geteilt. Das Testset macht 20% aus. Die restlichen 80% wurden dem Trainingsset zugewiesen.

**Erstellung eines passenden Models zur Vorhersage:** Zur Vorhersage eignet sich ein Model der linearen Regression am ehesten. Es wurde ein Model mittels der Funktion ‚LinearRegression()‘ aus der Bibliothek ‚sklearn‘ erstellt. Dieses Model wurde mit dem Trainingsset für Feature und Label trainiert. Anschließend lasse ich mir den Score für das Testset ausgeben. Dieser Score liegt meistens immer über 0.88. Dies ist ein relativ hoher Score und besagt, dass das Model gut funktioniert.

**Nutzung des Models zur Vorhersage:** Diesem Model übergebe ich nun die Features, welche vorher raus verschoben wurden um die vorherzusagende Reichweite. Das Model liefert mir dann die zu erwartenden Werte des Labels

**Grafische Darstellung des Ergebnisses:** Zuletzt lasse ich mir das Ergebnis noch grafisch darstellen mittels der Bibliothek ‚matplotlib‘. Bei der grafischen Repräsentation handelt es sich um einen 2d Graphen, welcher auf der x-Achse das Datum und auf der y-Achse das Label anzeigt. Die bekannten Daten werden in rot dargestellt und die erwartenden Werte in blau.