Datensätze:

* BlackandWhite:

Ein Datensatz der nur aus 3 Bildern besteht. Diese wurden mit draw.io erstellt und besteht aus einfachen rechteckigen Schwarzen Symbolischem Besteck auf weißem Hintergrund. Er wurde während des Programmierens verwendet, um die Funktion zu überprüfen.

* Real\_Simple:

Selbst erstellte echte Bilder von Besteck auf weißem Hintergrund aber alle in der gleichen Position.

* Simple:

Selbst erstellte echte Bilder von Besteck auf weißem Hintergrund in verschiedenen Orientierungen.

Künstliches Neuronales Netz:

Das Netz wurde wie in der Vorlesung besprochen aufgebaut. Es wurde nur die Numpy Bibliothek (https://numpy.org/) verwendet, eine Bibliothek für die Verarbeitung von Arrays. Um die Initialisierung der Gewichtige sowie Punktoperationen durch führen zu können. Zudem ist das Netz als Klasse aufgebaut, so dass die Operationen parallel durchgeführt werden können und die Gewichte nicht einzeln initialisiert werden müssen.

Generell ist das Netz in drei Teile aufgeteilt:

1. Die Initialisierung von Neuronen, Lernrate, Gewichten und Aktivierungsfunktion
2. Das Training welches wiederum aus:
   1. dem Vorwärtsdurchgang
   2. dem Berechnen des Fehlers
   3. und dem Anpassen der Gewichte über diesen Fehler aufgebaut ist
3. Und dem Testen

Abschließend kann noch ein „test.jpg“ getestet werden. Der Code wurde ausführlich kommentiert und ist durch den gleichen Aufbau wie in den Vorlesungen gut nach voll Ziebart.

Ergebnisse:

Abschließend muss gesagt werden. Das Netz funktionierte nur sehr bedingt. Die Schwarzweißenbilder konnten gut erkannt werden und auch nach wenigen Epochen.

Bei den realen Daten war dies wesentlich schwerer.

Mit 40000 Eingangsneuronen und 4000 Versteckten Neuronen konnte auf dem real\_simple Datensatz eine gute Genauigkeit von 66,6% nach 25 Epochen erreicht werden. Es kann also 2/3 Klassen gut erkennen und Klassifizieren.

Mit 40000 Eingangsneuronen und 8000 Versteckten Neuronen konnte auf dem real\_simple Datensatz eine gute Genauigkeit von 83% erst nach 50 Epochen erreicht werden. Heißt das Netz hat diese Daten eigentlich schon auswendig gelernt und kann nicht gut verallgemeinern.

Eine Abhilfe wäre die Neuronen zahl oder die Schichten zu erhöhen, dies führt aber zu einer wesentlich längeren Berechnungszeit. Dennoch muss man beachten das jeder Trainingsvorgang andere Ergebnisse liefern kann. Da die Gewichte zufällige initialisiert werden.

Dennoch für ein einfaches Fully-Connected Netzwerk ohne die Verwendung von High-Level Bibliotheken sind wir mit dem Ergebnis durch aus zufrieden.