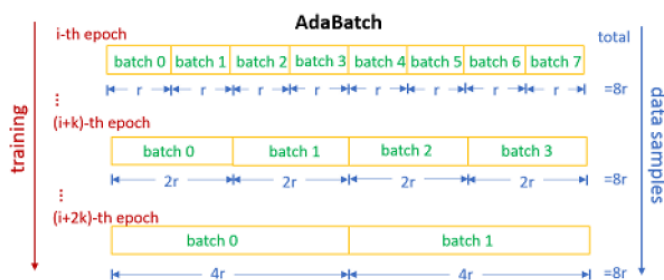
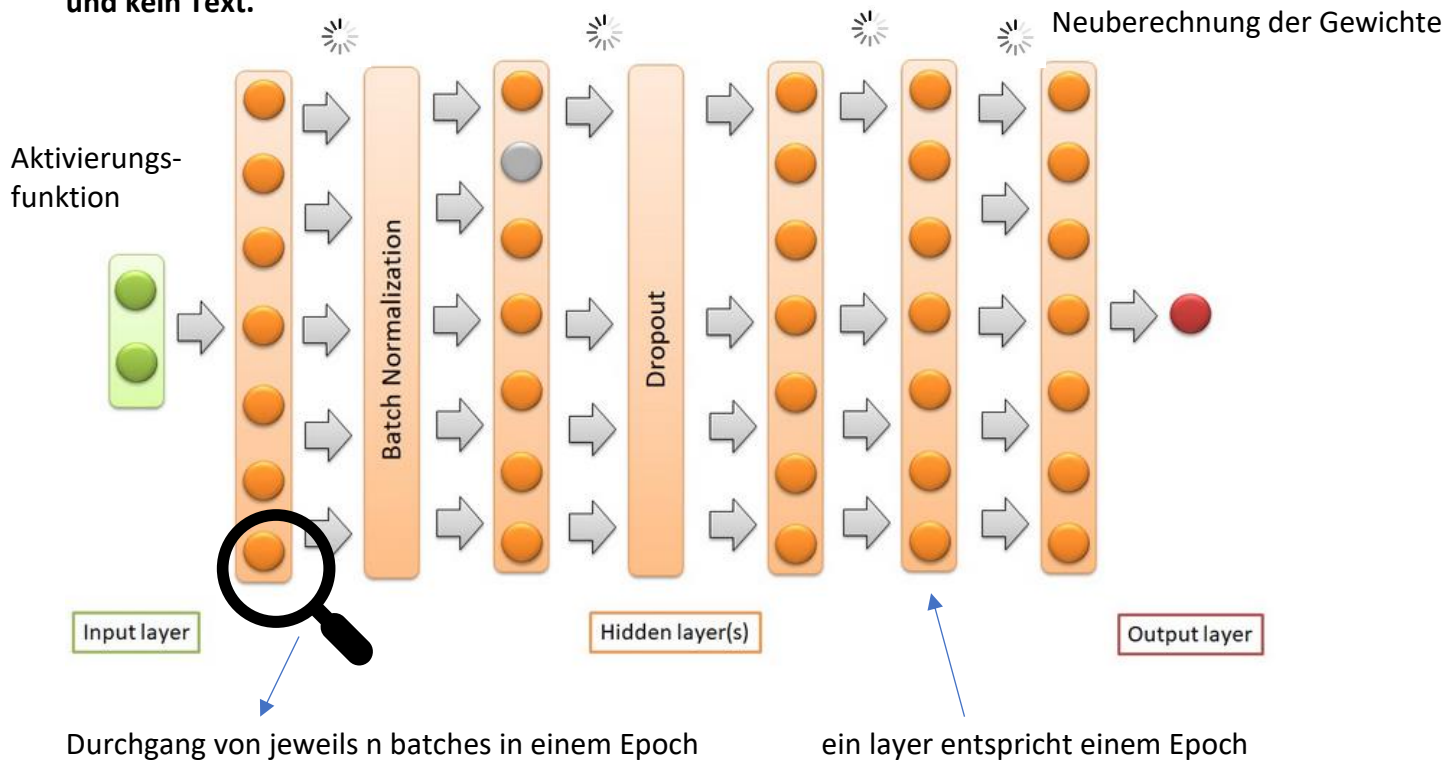


Aufgabe 2

Aufgabe 1: Grundelemente (10 Punkte)

Stellen Sie das Vorgehen und den Aufbau eines Deep Learning Modells grafisch dar. Die Grafik soll alle Optimierungsschritte und Ablaufschritte enthalten. Als Resultat soll eine Grafik abgegeben werden, welche vereinzelt Erklärungen enthalten darf, bitte kein Beispiel und kein Text.



Source: <https://www.analyticsvidhya.com/blog/2021/05/tuning-the-hyperparameters-and-layers-of-neural-network-deep-learning/>

Aufgabe 2: MNIST Modell (10 Punkte)

Erstellen Sie drei Klassifikationsmodelle für den MNIST Datensatz, wobei sie die Auswahl der Parameter variieren. Erklären und beschreiben Sie ihre Auswahl der Parameter und deren Einfluss auf die Resultate. Die Erklärungen sollen den Grund eurer Entscheidungen enthalten.

Die Parameter, welche geändert werden können sind, Batchgrösse, optimizer, Lernrate sowie Anzahl hidden layers.

Modell 1: Batch = 30, Aktivierungsfunktion „ReLU“, optimizer „adam“, lernrate 0.1, hidden layers 2

ReLU ist am beliebtesten für Deep Learning und setzt negative Werte auf 0 und positive auf den Inputwert. Die Batchgrösse um die 30 wird ebenfalls empfohlen als Richtwert.

Die Genauigkeit liegt bei 97.54%, der Verlust liegt bei 10.98%.

Modell 2: Batch = 32, Aktivierungsfunktion „sigmoid“, optimizer „adam“, lernrate 0.1

Sigmoid ist ebenfalls verbreitet, die anderen Parametern werden gleich gelassen.

Die Genauigkeit liegt bei 98.22%, der Verlust liegt bei 10.82%. Das Modell war somit etwas besser als das Modell 1, da die Aktivierungsfunktion effektiver war.

Modell 3: Batch = 32, Aktivierungsfunktion „ReLU“, optimizer „SGD“, lernrate 0.1

Verwendung eines neuen Optimizers, quasi der Basic aller optimizer. Gut und einfach.

Die Genauigkeit liegt bei 97.40%, der Verlust liegt bei 8.28%. Das Modell war weniger genau, dafür ist die Verlustfunktion optimierter. Der SGD ist eigentlich nur der Basis-Optimizer und ist im Detail teils nicht optimal, dafür einfacher.

Modell 4: Batch = 32, Aktivierungsfunktion „ReLU“, optimizer „SGD“, lernrate 0.5

Die Genauigkeit liegt bei 97.53%, der Verlust liegt bei 7.67%. Das Modell 3 wurde so noch leicht optimiert, da das Modell anscheinend schneller gelernt hat. Im Selbsttest habe ich noch für mich verschiedene Raten durchlaufen 😊

Aufgabe 3: Boston Housing (10 Punkte)

Im Keras Paket ist der Datensatz "Boston Housing" enthalten. Erstellen Sie nun für diesen Datensatz ein Regressionsmodell. Erklären und beschreiben Sie die Auswahl der Parameter. Damit meine ich, beschreiben Sie wieso sie die Parameter so ausgewählt haben. Das Resultat soll beschrieben werden. Wie gut schätzen Sie das Resultat ein?

Ich habe die Parameter LSTAT (%lower status der Bevölkerung) und RM (durchschnittliche Anzahl Zimmer pro Wohnung) im Modell aufgenommen. Der Vergleich der Anzahl Zimmer ist einerseits einfach zu bestimmen, da es sich um ganze Zahlen handelt. Zudem ist ein Rückschluss der Anzahl Zimmer auf den Status/Wohlstand sicherlich spannend, ob dies gemäss der Erwartung eintritt und wo allenfalls Abweichungen auftreten.

Die Anwendung in Python hat nicht funktioniert, trotz Doktor Google. Leider keine Unterrichtsbeispiele zur Orientierung vorhanden. Irgendwie klappt es mit dem Befehl LinearRegression() nicht.