Für die Projektwoche haben wir uns für den Kurs „Low Power Lab für Anwendungen mit Machine Learning“ entschieden. In der ersten Woche haben wir uns einen Überblick übers Machine Learning verschafft und unsere Idee für ein Projekte entwickelt.

Unsere Idee: Ein System entwickeln, das Pakete anhand ihrer Größe erkennen kann.

Die zweite Woche haben wir uns um die Umsetzung grundlegender Funktionalitäten gekümmert. In der letzten Woche wurde dann die Verbesserung unseres Systems und die Dokumentation vorangetrieben.

Wir haben uns intensiv innerhalb der drei Projektwochen mit Python auseinandergesetzt. In dieser Sprache wurde das Training, das Erkennen und das Vorhersagen durchgeführt. Bash nutzen wir, um die Python-Skripte auf dem Pi zu automatisieren. Mit einem PC, auf dem das Betriebssystem Linux lief, wurde der PI über eine SSH-Verbindung gesteuert.

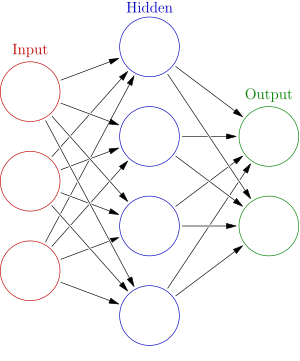
Wir konnten einen Einblick und grundlegendes Verständnis in Machine Learning und deren diversen Methoden gewinnen. Neuronale Netze bestehen aus „Neuronen“, welche vereinfacht dieselben Funktionen eines Neurons im Gehirn haben. Abhängig von verschiedenen Informationen trifft das Neuron unterschiedliche Aussagen. Mehrere verschaltete Neuronen bilden ein Netz welches wie folgt aussieht:[[1]](#footnote-2)

Abbildung 1 Quelle Wikipedia

Durch unseren Zeitrahmen und die Menge an Daten, ist es sinnvoll ein supervised neural Network zu nutzen. Hier werden Daten zusammen mit Labels (hier die Größe der Pakete) zum Trainieren eingesetzt. Da die Labels nicht vom Netz selbst generiert werden, spricht man von Supervised Learning. Das Netzwerk wird also trainiert, um Pakete die richtigen Labels zuzuweisen. Es wurde sich primär mit der Entwicklung von Features beschäftigt. Das Anlernskript wurde so programmiert, dass die Anzahl der Neuronen beliebig wählbar ist, so dass Overfitting vermieden werden kann. Als Overfitting bezeichnet man ein Ereignis, bei dem das neural Network die Trainingsdaten auswendig lernt und keine richtigen Aussagen mehr über neue Daten treffen kann. Der generierte Datensatz wurde zu 90% zum Training und 10% zum Testen benutzt. Eine hohe Bedeutung hat ein guter Datensatz. Durch anfänglich schlechte Kamera-Einstellungen wurden teils schlechte Daten generiert, wodurch das Training und das Testen fehlerhaft war. Mit korrigierten Kamera Einstellungen kommen wir schließlich auf diese Ergebnisse:

Durchschnittstrainingsnote (Genauigkeit)= 0,9777227722772277

Durchschnittstestnote (Genauigkeit)= 0,9849624060150377

Die eingebauten Features sehen wie folgt aus:



Die Anzahl der gelben Pixel und die Position der horizontalen Linie mit den meisten gelben Pixel wird an das neuronale Netz übergeben.

Wenn das neuronale Netz trainiert ist, so erhält man eine Inferenz Datei, die nun die gelernte Paketerkennung enthält.

1. <https://en.wikipedia.org/wiki/Machine_learning> [↑](#footnote-ref-2)