Faculty of Science and Engineering

Alexander J. Johnson

Mathematics

Neural Networks with Python and TensorFlow

February 22, 2020

School of Computing, Mathematics and Digital Technology

Abstract

Artificial Intelligence (AI) still remains as one of the greatest challenges in scientific research to this date, but much progress in the field has been made using artificial neural networks. The design of artificial neural networks is loosely inspired by that of biological brains, and serves as an expansion of an earlier concept called the perceptron (Rosenblatt, 1958). By using multiple layers of these artificial neurons, we can form a highly connected system that is referred to as a neural network, these networks can then be trained on a large data set to predict the output with high accuracy.

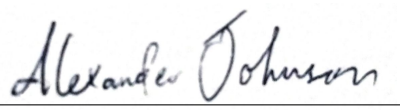
The range applications for neural networks is wide: they can be used to classify data, predict future states of chaotic systems, apply stylisations to images, and control physical/physically-based system in real-time.

TODO: *Abstract.*

Declaration

With the exception of any statement to the contrary, all the material presented in this report is the result of my own efforts. In addition, no parts of this report are copied from other sources. I understand that any evidence of plagiarism and/or the use of unacknowledged third party materials will be dealt with as a serious matter.

Signed



Contents

Chapter 1

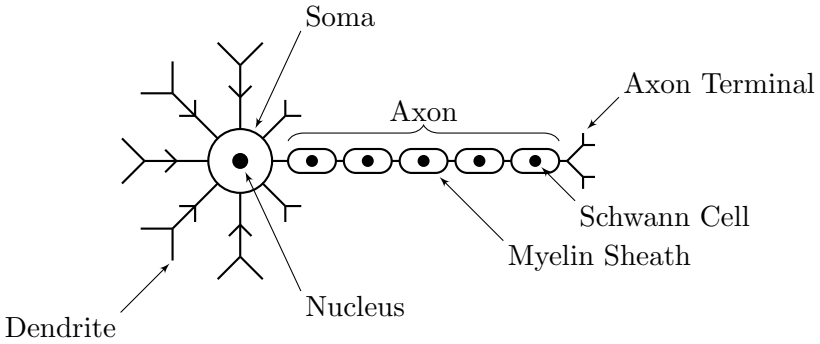
Introduction to Neural Networks

TODO: *Chapter: Introduction*

Biological Neurons

Biological neurons are electrically excitable cells that are found in almost all animals. These neurons can transmit and receive electrical signals to one another via synaptic connections, which maybe either excitatory or inhibitory. Any given neuron will be either active or inactive depending on whether or not its input exceeds a threshold.

Figure 1.1: Diagram of a biological neuron.



Signals are received by the neuron via connections to dendrites and soma. If the threshold is met, electrical signals are sent along the axon to the terminal, where it is connect to more neurons or to a controllable cell such as a neuromuscular junction.

TODO: *Section: Biological Neurons*

Artificial Intelligence

The idea of artificial beings capable of human intelligence can be traced back to mythical stories from ancient Greece. One such story was that of a mythical automaton called Talos, who circled an island's shores to protect it from pirates and other invaders.

By the 19$^{\text{th}}$ century, other notions of artificial intelligence were explored by fiction in stories, such as Mary Shelley's "Frankenstein", and Karel Čapek's "R.U.R.". Some of the fictional writings of the 20$^{\text{th}}$ century further continued to explore the concept in novels such as Isaac Asimov's "I, Robot".

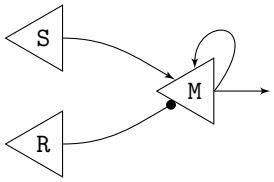
Academic research into artificial intelligence began around the 1940's, primarily due to findings in neurological research at the time. The first explorations of artificial neural networks was done by McCulloch and Pitts, 1943, who investigated how simple logic functions might be performed by idealised networks. The neurons within these networks operated using some basic logic rules, applied to a discrete time system which, can be summarised using the expression

$$N(t) = (E_1(t-1) \vee E_2(t-1) \vee \dots) \wedge \neg(I_1(t-1) \vee I_2(t-1) \vee \dots),$$

where $N(t)$ is the state of a neuron at time t, and $E_i(t-1)$ and $I_i(t-1)$ are the states of the excitatory and inhibitory connections from the previous time step respectively. The result is such that the neuron will only be active if at least one excitatory connection is active and all inhibitory connections are inactive. The versatility of this definition is demonstrated in the following examples.

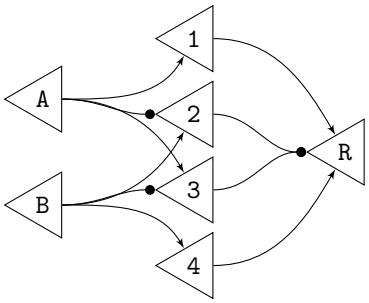
Figure 1.2: Common logic components using McCullochs neurons.

SR Flip-flop



$N_M(t+1) = (N_S(t) \vee N_M(t)) \wedge \neg N_R(t)$

AND Gate



$N_R(t+2) = N_A(t) \vee N_B(t)$

TODO: *Section: Artificial Intelligence*

Perceptrons

The idea of the perceptron was originally conceived by Rosenblatt, 1958, to represent a simplified model of intelligent systems free from particularities of biological organisms, whilst maintaining some of their fundamental properties.

The perceptron was built as a dedicated machine that consisted of a number of photovoltaic, analogous to a retina, that feed into an "association area". This association area contains a number of cells that each calculate a weighted sum of the receptor values and output a signal if it exceeds a threshold. These value weights were implemented using variable resistance wires that the perceptron could adjust automatically. The outputs from the association area are then connected to response cells, which operate in a similar fashion to the association cells. The activation of these response cells are the outputs of the perceptron, and indicated the classification of the input.

This machine was initially trained to reliably identify three different shapes: a square, a circle, and a triangle; and did so with a better than chance probability. When attempting to use the perceptron for more complicated tasks, such as character recognition, it failed to produce better than chance results.

`TODO`: *Subsection: Perceptrons*

Backpropagation

`TODO`: *Subsection: Backpropagation*

Types of Neurons

`TODO`: *Section: Types of Neurons*

CNN

RNN

LSTM

Chapter 2

Neural Networks in Python

Python is a general-purpose programming language designed by Guido van Rossum, with an emphasis on readability and reusability (Rossum, 1996). It comes with an extensive standard library and is one of the most popular programming languages.

There are multiple options for interacting with Python, these include:

- typing commands into an interpreter,
- writing files and running them with an interpreter,
- using an online service such as Google Colab.

`TODO`: *Chapter: Neural Networks in Python*

Single Perceptron Boston Housing Data

`TODO`: *Section: Single Perceptron Boston Housing Data*

Multi Layer Perceptron Boston Housing Data

`TODO`: *Section: Multi Layer Perceptron Boston Housing Data*

XOR Gate

`TODO`: *Section: XOR Gate*

Chapter 3

Introduction To TensorFlow

TensorFlow TODO: *Chapter: Introduction To TensorFlow*

Linear Regression

TODO: *Section: Linear Regression*

XOR

TODO: *Section: XOR*

Boston Housing with Keras

TODO: *Section: Boston Housing with Keras*

Chapter 4

Deep Learning

TODO: *Chapter: Deep Learning*

Recurrent Neural Networks

TODO: *Section: Recurrent Neural Networks*

Convolutional Neural Networks

TODO: *Section: Convolutional Neural Networks*

Image Processing

TODO: *Subsection: Image Processing*

Supervised Learning

TODO: *Section: Supervised Learning*

Unsupervised Learning

TODO: *Section: Unsupervised Learning*

Autoencoding

TODO: *Section: Autoencoding*

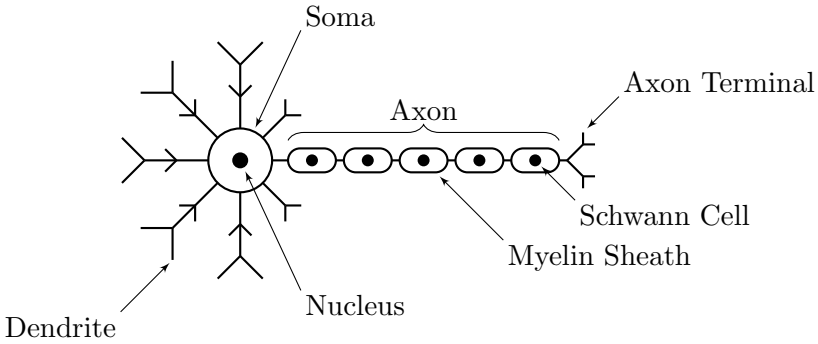
Reinforcement Learning

TODO: *Section: Reinforcement Learning*

Chapter 5

Null

Figure 5.1: Diagram of a biological neuron.



TODO: *Decide parent chapter for CNN, RNN, and LSTM sections*

List of Figures

List of Tables

Bibliography

McCulloch, Warren S and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *The bulletin of mathematical biophysics* 5.4, pp. 115–133.

Rosenblatt, Frank (1958). "The perceptron: a probabilistic model for information storage and organization in the brain." In: *Psychological review* 65.6, p. 386.

Rossum, Guido van (1996). *Foreword for "Programming Python" (1st ed.)* URL: `https://www.python.org/doc/essays/foreword/` (visited on 02/07/2020).