

# Computer aided transcription: Can Neural Networks be used to analyse audio for generating drum-kit notation?

Daniel Topping 100369076

October 22, 2018

## Contents

<b>1</b>	<b>Literature Review</b>	<b>1</b>
1.1	Neural networks . . . . .	1
1.1.1	Biological neurons . . . . .	1
1.1.2	Perceptrons (early neuron models) . . . . .	1
1.1.3	Neural networks . . . . .	2
1.1.4	Convolutional neural networks . . . . .	2
1.2	Obstacles . . . . .	3
1.2.1	Creating training data . . . . .	3
1.2.2	Multi-label data . . . . .	3
1.2.3	Optimising to avoid local minima . . . . .	3
1.2.4	Overfitting . . . . .	3
<b>2</b>	<b>Abstract</b>	<b>3</b>
<b>3</b>	<b>Bibliography</b>	<b>3</b>

# 1 Literature Review

## 1.1 Neural networks

Artificial neural networks (NN's) are models loosely based upon the biological processes which happen in the central nervous system of complex organisms. They are models designed to approximate a function which addresses a sort of pattern in some data [7]; this is useful because once this approximation is in place the NN can recognise certain classes it has been trained to recognise in new data.

The behaviour of neurons has been known to have useful applications in computing since 1943, when McCullock and Pitts explained that the characteristics of neurons made them useful in 'propositional logic' [1]. This is useful in computing, because neurons can be described as a 'computational unit' [2]. During this time, simplistic neuron models which resembled basic logic gates were used to solve problems, however these models were bespoke for the problems they addressed and so couldn't be reused for slightly different problems. [2]

### 1.1.1 Biological neurons

Neurons are useful, as inspiration for these models, because each neuron performs a decision. Neurons get excited by input from other neurons which travel across many synapses to the neuron. If this combined excitation reaches a threshold point; then the neuron fires a signal which propagates down its axon to other synapses of other neurons. But if it doesn't reach the threshold it doesn't fire. [2]

### 1.1.2 Perceptrons (early neuron models)

The 'Perceptron' was introduced in 1958 and perfected in 1960 [2] which improved upon the prior model by adding weights to each input connection. This meant that learning algorithms could be used on a single perceptron, adjusting the connection weights incrementally to produce desirable output, without having to change the overall structure, as you would with a network of McCullock-Pitts neurons. However, perceptrons still had the downside of not being accurate when approximating when the decision boundary isn't linear [13].

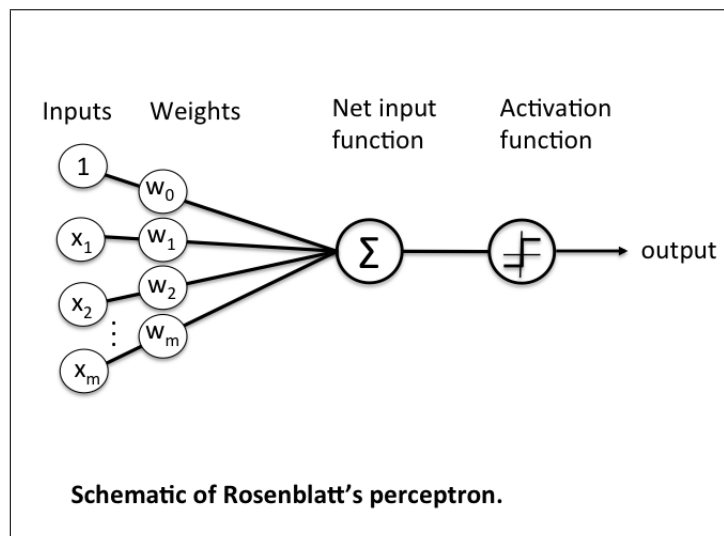


Figure 1: [13]

1.1.3 Neural networks

Neural networks are when multiple neurons are layered to create more complexity, capable of understanding more sophisticated patterns. However, although these networks can be largely successful on their own they can be very resource intensive if they contain thousands of nodes so NN's aren't very well optimised if you just increase the amount of neurons without considering any processing which can simplify the data.

1.1.4 Convolutional neural networks

Convolutional neural networks (CNN's) are neural networks which came about after D. H. Hubel and T. N Wiesel's work on the visual cortex, where in their 1961 paper they describe a cat's visual system. This work inspired a structural design of neural networks which neurons only took inputs from small regions of data. [10]

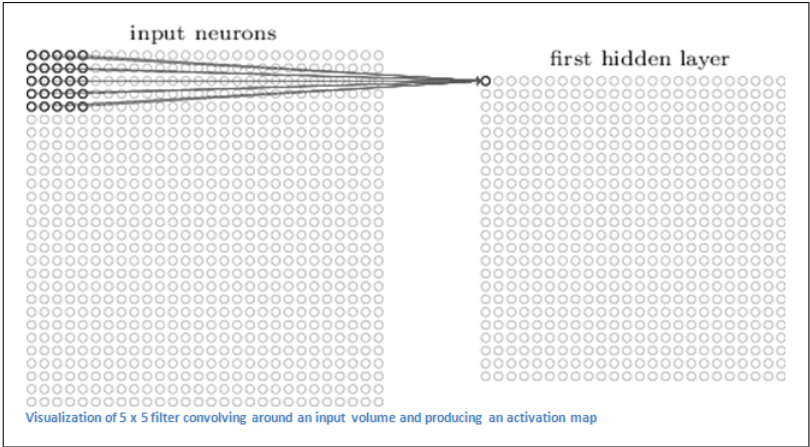


Figure 2: [12]

Modern CNN's perform some kind of filtering by having specially selected weights on the neurons. For example, in images, the Sobel operator can be used to do image smoothing and edge detection [4]. Also after convolution layers a technique called pooling is often used to reduce the data into a smaller size [12].

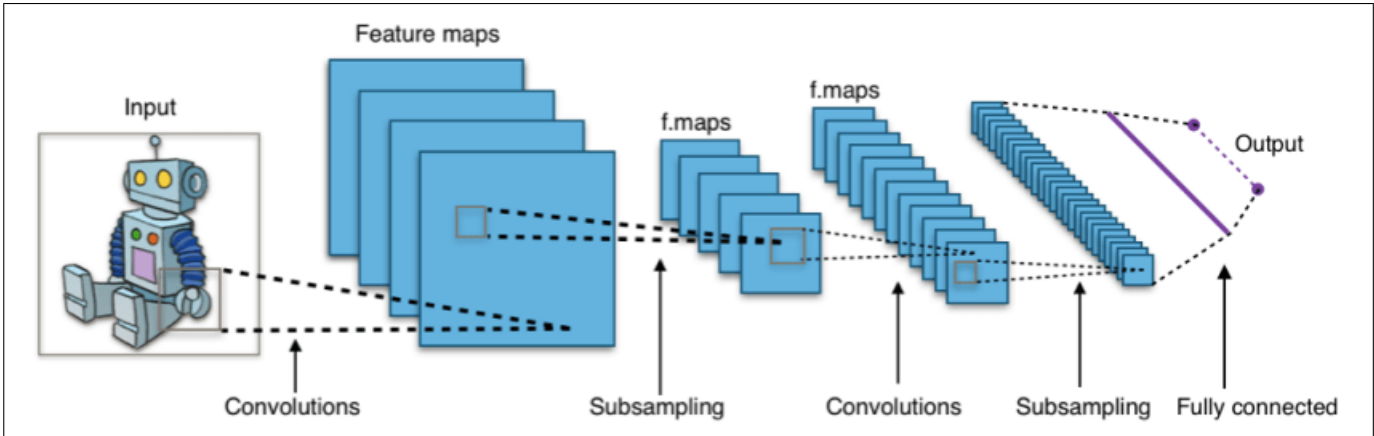


Figure 3: The standard CNN structure, with convolution and pooling layers [Yung].

## 1.2 Obstacles

### 1.2.1 Creating training data

While it is possible to create a dataset manually, this can take time and memory; especially if you want a high accuracy NN, because more is typically better.

### 1.2.2 Multi-label data

Many neural networks are designed to work using labels (what the data represents), that are mutually exclusive. So the MNIST dataset [MNIST] for example has pictures of handwritten numbers with labels 0-9 and only one of these labels is true per image. However, in the case that I am going to investigate the labels are not mutually exclusive, because at one point in time multiple drum and cymbal combinations could be hit at once.

A cost function in the output layer needs to be used which will not be affected by the other output nodes, for this reason Softmax cannot be used since all output nodes sum to 1.0 with softmax. It was found that the sigmoid function performed better than ReLU and hyperbolic functions by [ghlmowyyzz2017].

### 1.2.3 Optimising to avoid local minima

[11]

### 1.2.4 Overfitting

Overfitting is when the machine learning technique used to approximate some unknown model, creates an overcomplicated model; which is very accurate to the training data, but not accurate (because of this overcomplexity) to new test data [[2]]. [9] found that it is best to use data augmentation to create transforms in the data space (before getting processed by a NN) than mid-way through in the feature space (after convolution layers, but before feature analysis layers of a NN).

## 2 Abstract

## 3 Bibliography

### References

- [1] Warren McCullock and Walter Pitts. *A logical calculus of the ideas immanent in nervous activity*. 1943.
- [2] R. Rojas. *Neural Networks - A Systematic Introduction*. Berlin, Heidelberg: Springer-Verlag, 1996. ISBN: 978-3-540-60505-8.
- [3] E. Selfridge-Field. *Beyond MIDI*. Cambridge, Massachusetts: MIT Press, 1997. ISBN: 978-0-262-19394-8.
- [4] A. Cawsey. *The essence of artificial intelligence*. Hemel Hempstead, Hertfordshire: Prentice Hall Europe, 1998. ISBN: 978-0-13-571779-0.
- [5] Yann LeCun et al. *Gradient-Based Learning Applied to Document Recognition*. 1998.
- [6] M. Müller. *Fundamentals of music processing*. Berlin, Heidelberg: Springer-Verlag, 2015. ISBN: 978-3-319-21944-8.
- [7] MIT OpenCourseWare. *12a: Neural Nets*. Online Video. Accessed on 10-10-2018. 2016. URL: <https://www.youtube.com/watch?v=uXt8qF2Zzfo>.
- [8] MIT OpenCourseWare. *12b: Deep Neural Nets*. Online Video. Accessed on 10-10-2018. 2016. URL: [https://www.youtube.com/watch?v=VrMHA3yX\\_QI](https://www.youtube.com/watch?v=VrMHA3yX_QI).
- [9] Sebastien C. Wong et al. *Understanding Data Augmentation for Classification: When to Warp?* IEEE, 2016.

- [10] Stanford University School of Engineering. *Lecture 5 — Convolutional Neural Networks*. Online Video. Accessed on 22-10-2018, slides at: [http://cs231n.stanford.edu/slides/2017/cs231n\\_2017\\_lecture5.pdf](http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture5.pdf). 2017. URL: <https://www.youtube.com/watch?v=bNb2fEVKeEo>.
- [11] *Global vs. Local Minima Using ga*. Date last accessed 11/10/2018. URL: <http://uk.mathworks.com/help/gads/example-global-vs-local-minima-with-ga.html>.
- [12] Michael A. Nielsen. *Neural Networks and Deep Learning*. Accessed on 22-10-2018. URL: <http://neuralnetworksanddeeplearning.com/chap6.html>.
- [13] Sebastian Raschka. *Single-Layer Neural Networks and Gradient Descent*. Accessed on 17-10-2018. URL: [https://sebastianraschka.com/Articles/2015\\_singlelayer\\_neurons.html](https://sebastianraschka.com/Articles/2015_singlelayer_neurons.html).