

IAML DL - Study Guide - Week 09 10

Sambit Paul, Pavlos Andreadis

March 2020

1 Introduction

Week 9 provides a flavour of how state-of-the-art machine learning techniques work. We introduce two concepts, deep learning with neural networks and the basic building blocks of neural networks - perceptrons. Perceptrons can be considered as single layer neural networks with a single computation node and one activation function.

Artificial Neural Networks are a connected set of perceptrons which introduces non-linearity through the use of activation functions. This results in a series of transformations over the input data which is often referred to as "learning features". It is important to note that this does not erase the need for feature engineering, as the input data still needs to be designed.

For a neural network model to perform well, large amounts of training data are usually required. That being said, it is possible to start training from a "pre-trained" network, i.e. a network that has already been trained for a different, in some sense, compatible task. This can significantly reduce the need for training data and training time. See "transfer learning" for this latter, more advanced topic [Tan et al., 2018].

Training neural networks is typically referred to as "Deep Learning".

2 The Perceptron

- The 'Perceptron' was first presented in Rosenblatt [1957] where it was described as "a class of artificial nerve nets, embodying aspects of the brain and receptors of biological system" Kanal [2003].
- Before the emergence of the perceptron, the first step to developing an artificial neuron mimicking the functionality of a biological neuron was taken in McCulloch and Pitts [1943], which introduced the McCulloch-Pitts (MCP) Neuron. This article provides a gentle introduction to the topic.
- The paper Rosenblatt [1957] can be found [here](#), and can be interesting if you would like to understand how the idea of 'perceptrons' came to be.

- The fundamental difference between the perceptron and logistic regression is the activation function that is used:
 - Logistic regression uses the sigmoid function to squish the values.
 - Perceptrons use a step function to squish values.

A comparison is provided in Figure 1.

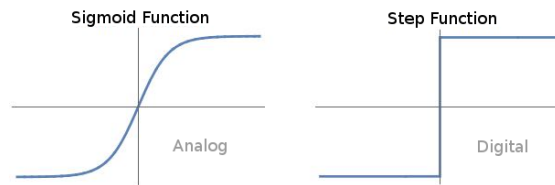


Figure 1: Step Function and Sigmoid Function

- The major achievements of Rosenblatt with his Perceptron model was not just an improvement over the *McCulloch-Pitts Neuron* for binary classification, but also building a simple yet reliable and efficient algorithm to train the perceptron (learn the weights).

Algorithm 1: Perceptron Learning Algorithm

Input: Training Examples: $[x_i, y_i]_{i=1}^n$
 Initialise w randomly
while *!convergence* **do**
 // loop through n examples
 for $i=1, n$ **do**
 // wrong prediction updates the weights
 if x_i is in class 1 & $w \cdot x < 0$ **then**
 $w = w + x$
 else if x_i is in class 0 & $w \cdot x > 1$ **then**
 $w = w - x$
 end
 end
end

This [gist](#) provides an implementation of the learning algorithm in Python.

3 Artificial Neural Networks

- The most convenient book to refer to for this section is [Goodfellow et al. \[2016\]](#).

- Artificial Neural networks are models connecting multiple perceptrons (known in the area as "neurons"), which all are trained together. That being said, perceptron might no longer be the correct term to use, as the transformation induced is not in general a step function. The different transformations that can be applied to the input are referred to as activation functions (which could be a step function).

By setting the appropriate loss function, we can train neural networks for classification or regression, while neural networks also see use for clustering [Min et al., 2018].

- Goodfellow et al. [2016] Section 6.2 provides a good description of the gradient descent technique, as employed to train neural networks.
- 2 key elements which determine the function of a neural network are:
 - the *Loss Function*: A concept you should, in general, already be familiar with but for which you can read more in this [article](#). It is worth noting that when introducing a new problem, or trying to improve the formulation of a preexisting one, we will be looking to define a loss function for the neural network to be trained with. Though this is not technically different than what you have seen so far with other machine learning techniques, you might be accustomed to by this point with the idea that specific techniques have specific loss functions. Though that is not generally true, it is most definitely not the case for neural nets.
 - *Activation Functions*: Typically non-linear functions applied to the output of each layer of the neural network. You can read about the different activation functions in this [article](#). Nwankpa et al. [2018] also provides a useful summary of the most frequently used activation functions.
- You can find a good description of tackling a classification problem with neural networks in Zhu [2018]. This paper is on the MNIST hand-written digit database, which is a staple introductory problem in the area.

The below represent more advanced topics in the area that go well beyond the content of this course:

- **Convolutional Neural Networks:**
This [video](#) provides a succinct and clear explanation of how CNNs work.
- **Recurrent Neural Networks:**
This [video](#) provides a great introduction to RNNs.
- **Machine Learning Practical course:**
This on-campus course on Applied Deep Learning has most of its resources (including some very useful training labs) available at this [page](#). *Note that this is a year long course for which we cannot provide you with support through IAML.*

References

- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- Laveen N Kanal. Perceptron. *Encyclopedia of Computer Science*, 2003.
- Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- Erxue Min, Xifeng Guo, Qiang Liu, Gen Zhang, Jianjing Cui, and Jun Long. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, 6:39501–39514, 2018.
- Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018.
- Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.
- Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018.
- Wan Zhu. Classification of mnist handwritten digit database using neural network. *Research School of Computer Science, Australian National University, Acton, ACT*, 2601, 2018.