Neural Networks and Deep Learning *

Mohamed Hesham Ibrahim Abdalla[1][0000−1111−2222−3333], Second Author[1][1111−2222−3333−4444], and Third Author[1][2222−−3333−4444−5555]

German University in Cairo, Cairo, Egypt

Abstract. The abstract should briefly summarize the contents of the paper in 15–250 words.

Keywords: Neural Networks · Deep Learning · Machine Learning

1 Introduction

Artificial Neural Networks (ANNs) and preceptrons are intelligent units that has taken inspiration from biology, especially the brain (cite). ANNs work by taking labeled inputs and then trying to find a mathematical rule or function to systematically answer the question of which label belongs to which input, and later identify labels of new inputs that have never been seen before by the network. For example, inputs could be human images and the labels are the gender of the human in that particular image.

The history of ANNs and preceptrons goes back to the 50's and the 60's when the first known preceptron was created. The first preceptron was simulated on an IBM 704 computer at Cornell Aeronautical Laboratory in 1957 (cite). It works by giving(cont)

2 Mathematical Background and Concept

ANNs learn how to accomplish a task by following two main steps: forward propagation and backward propagation. Forward propagation is the process of predicting labels and computing how deviated these labels from the ones provided in the input data. On the other hand, backward propagation tries to correct the predictions by minimizing the diffrence between the input labels and the predicted labels.

2.1 Forward Propagation

ANNs consist of neurons which are connected via links. Each neuron gets an input and produces an output. Each input is given a certain weight which makes that specific input has more or less priority in controlling the output of the neuron. Neurons calculate their output by multiplying their inputs by their weights

* Supported by organization x.

and applying a bias to the multiplication. Eqaution 1 shows a linear mapping of a single input.

$$z^i = w^T * x^i + b \tag{1}$$

where z^i is the linear mapping of the ith example, w^T in <- R^{1xN} is the weight vector of the form $[w_1, w_2, ...w_N]$ and x^i in <- R^{Nx1} is the input vector of the form $[x_1, x_2, ...x_N]$ of the ith example.

This mapping is then forwaded to an activation function (Eqaution 2). Action functions are used to limit the linear transformation output.

$$a^i = g(z^i) \tag{2}$$

where a^i is the output of the activation function (unit activation) on the linear mapping of the ith example.

The choice of the activation function vary depending on the given data (inputs). In general, there are three main categories of the activation functions: binary, linear and non-linear. Binary or threshold functions output a binary value depending on the input. For example, the step function produces a +1 in case z^i is greater than or eqaul to 0 and -1 otherwise (Eqaution 3). Binary functions can not deal with categorical data, therefore they are not widely used.

$$f(x) = \begin{cases} +1 & z^i \leq 0 \\ -1 & \text{otherwise} \end{cases} \tag{3}$$

Another type of activation functions is linear. Linear functions forward the input directly to the output without any transformation. This is useful in problems where the output is continuous, for example, predicting house prices.

Although all of the previous functions are useful for some sitautions, they fail to find a pattern if the data is non-linearly separable since the function will only be able to draw a linear decesion boundary that can devide the data into two groups. Therefore, other functions were used to find non-linear seprations between the data such as sigmoid, ReLU and tanh/ Hyperbolic Tangent.

After calculating the activation unit for the inputs, a loss function L is calculated for each prediction to measure diffrent it is from the real data. the formula of the loss function varies based on the type of the labels. For example, if the labels are discrete valus or categories, Cross-Entropy could be used. Cross-Entropy increases when the predicted probability diverges from the actaul label.

References

1. Author, F.: Article title. Journal **2**(5), 99–110 (2016)
2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.10007/1234567890
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
4. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
5. LNCS Homepage, http://www.springer.com/lncs. Last accessed 4 Oct 2017

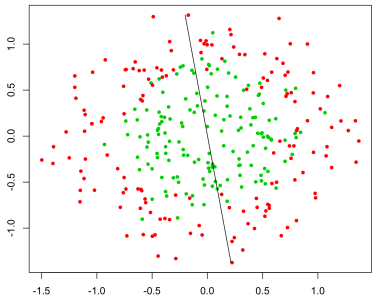
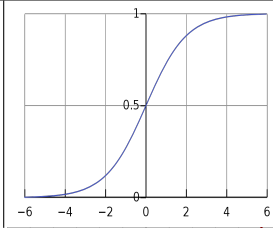


Fig. 1. Example of a linear activation function on non linearly separable data https://www.r-bloggers.com/interactive-visualization-of-non-linear-logistic-regression-decision-boundaries-with-shiny/

Table 1. diffrent Activation functions.

Function name	Formule	Graph
Sigmoid	$h_\theta(x) = \frac{1}{1+e^x}$	
ReLU	$Relu(x) = max(0, x)$	
tanh/ Hyperbolic Tangent	$\tanh(x)$