

# Variability in Feed Forward Neural Network

NIVHSA

India

January 7, 2021

## Abstract

Weights of a simple Feed Forward Neural Network (FFNN) attain finite values upon first order optimisation algorithms. We propose an alternate method in which the weights are modelled as a composite function; 1) mapping between input and output spaces, optimised via back propagation algorithm 2) Information captured inherently present in the input space via clustering.

## 1 Introduction

Fig.1 illustrates a basic FFNN [1],  $\mathbf{x}$  is a 2-dimensional vector( $x_1, x_2$ ) provided as two inputs to the network.

The forward propagation eqs. as follows:

$$z_1^{[0]} = w_{11}^{[0]}.x_1 + w_{21}^{[0]}.x_2 + b_1^{[0]}; \quad a_1^{[0]} = \sigma(z_1^{[0]}) \quad (1)$$

$$z_2^{[0]} = w_{12}^{[0]}.x_1 + w_{22}^{[0]}.x_2 + b_2^{[0]}; \quad a_2^{[0]} = \sigma(z_2^{[0]}) \quad (2)$$

$$z_1^{[1]} = w_{11}^{[1]}.a_1^{[0]} + w_{21}^{[1]}.a_2^{[0]} + b_1^{[1]}; \quad a_1^{[1]} = \sigma(z_1^{[1]}) \quad (3)$$

where  $\sigma$  is the sigmoid function.

The Backward Propagation eqs. as follows:

$$l = \hat{y}_1 - y_1 \quad (4)$$

We consider stochastic gradient descent (SGD) algorithm [2].

$$w_{ij} := w_{ij} - \alpha \cdot \frac{\partial l}{\partial w_{ij}}, \quad (5)$$

where  $\alpha$  is the learning rate. For e.g., from Fig.1

$$w_{21}^{[0]} := w_{21}^{[0]} - \alpha \cdot \frac{\partial l}{\partial w_{21}^{[0]}}$$

Where

$$\frac{\partial l}{\partial w_{21}^{[0]}} = \frac{\partial l}{\partial a_1^{[1]}} \cdot \frac{\partial a_1^{[1]}}{\partial z_1^{[1]}} \cdot \frac{\partial z_1^{[1]}}{\partial a_1^{[0]}} \cdot \frac{\partial a_1^{[0]}}{\partial z_1^{[0]}} \cdot \frac{\partial z_1^{[0]}}{\partial w_{21}^{[0]}}$$

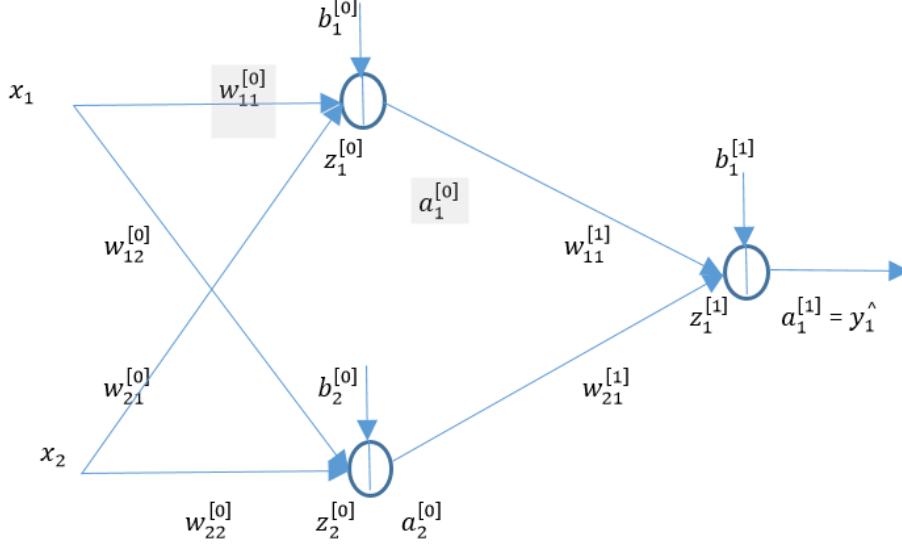


Figure 1: Feed Forward Neural network with one hidden layer

## 2 Methods

The weight

$$w_{ij} = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_{ij}}} \cdot e^{-\mu_{ij}^2 / (2 \cdot \sigma_{ij}^2)}$$

and the weight update equation upon modification from eq.5, we get

$$k \cdot e^{-\mu_{ij}^2 / (2 \cdot \sigma_{ij}^2)} := k \cdot e^{-\mu_{ij}^2 / (2 \cdot \sigma_{ij}^2)} - \alpha \cdot \frac{\partial l}{\partial \mu_{ij}} \cdot \frac{d\mu_{ij}}{dw_{ij}}; \quad k = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_{ij}}}$$

Now,  $d\mu_{ij}/dw_{ij} = [dw_{ij}/d\mu_{ij}]^{-1} = -\frac{\sigma_{ij}}{k \cdot \mu_{ij}} \cdot e^{-\mu_{ij}^2 / (2 \cdot \sigma_{ij}^2)}$

$$k \cdot e^{-\mu_{ij}^2 / (2 \cdot \sigma_{ij}^2)} := k \cdot e^{-\mu_{ij}^2 / (2 \cdot \sigma_{ij}^2)} + \alpha \cdot \frac{\sigma_{ij}^2}{k \cdot \mu_{ij}} \cdot e^{\mu_{ij}^2 / (2 \cdot \sigma_{ij}^2)} \cdot \frac{\partial l}{\partial \mu_{ij}}$$

The final new weight update equation is given by:

$$\mu_{ij} := \mu_{ij} + \alpha \cdot \frac{\sigma_{ij}^2}{k^2} \cdot e^{\mu_{ij}^2 / (\sigma_{ij}^2)} \cdot \frac{\partial l}{\partial \mu_{ij}} \quad (6)$$

Deviation in eq. 6,

$$\sigma_{ij} = \|\mathbf{x} - \mathbf{x}_c\|_2$$

, where  $\mathbf{x}_c$  is the centroid obtained by cluster centre computed via any cluster classification algorithm such as kmeans, Gaussian Mixture Model(GMM). In the case of classification, the number of clusters equals the number of class labels while for regression based, number of clusters equals the optimal clusters computed via elbow curve or silhouette score.

### 3 Discussion

In this paper we aimed at providing an alternative to the current back propagation algorithm in that we hope to address two factors. First, the new weight update equation is a composite function partly capturing the mapping information between the input space and output space along with information captured inherently present in the input space via clustering. Secondly, We introduced the deviation factor as a measure of variability for the weights. Intuitively, the weights are no longer finite irrespective of the inputs instead depends on the input itself, this could provide an extra degree of freedom for the weights and enhance learning. It remains to be seen via simulations as future work.

### References

- [1] Andina, Diego, Pham, D., Vega-Corona, Antonio, Seijas, Juan , Torres-Garcia, J. (2007). Neural Networks Historical Review.
- [2] Rumelhart, D., Hinton, G., Williams, R. Learning representations by back-propagating errors. Nature 323, 533?536 (1986). <https://doi.org/10.1038/323533a0>