# Implenting Backpropagation Algorithm - Project 1*


1st Abhishek Pareek
*ECE 542*
*North Carolina State University*
Raleigh, US
apareek2@ncsu.edu

2nd Vishal Shitole
*ECE 542*
*North Carolina State University*
Raleigh, US
vgshitol@ncsu.edu

3rd Christian Perez
*ECE 542*
*North Carolina State University*
Raleigh, US
cperezo@ncsu.edu


## I. Derivations

$$\frac{\partial L(\sigma(a))}{\partial a_k} = -\sum_{j}^{10}[\frac{\partial((1-y_j)\ln(1-\sigma(a_j)))}{\partial a_k} + \frac{\partial(y_j \ln(\sigma(a_j)))}{\partial a_k}] \tag{1}$$

For $j \neq k$ the function is treated as a constant:

$$j \neq k : \frac{\partial L(\sigma(a))}{\partial a_k} = 0 \tag{2}$$

For $j = k$ the derivation as follows:

$$j = k : \frac{\partial L(\sigma(a))}{\partial a_k} \tag{3}$$

$$= -\frac{\partial((1-y_k)\ln(1-\sigma(a_k)))}{\partial a_k} - \frac{\partial(y_k \ln(\sigma(a_k)))}{\partial a_k} \tag{4}$$

$$= -(1-y_k)\cdot\frac{1}{1-\sigma(a_k)}\cdot[-\sigma(a_k)\cdot(1-\sigma(a_k))] - y_k \cdot \frac{1}{\sigma(a_k)}\cdot[\sigma(a_k)\cdot(1-\sigma(a_k))] \tag{5}$$

$$= (1-y_k)\cdot\sigma(a_k) - y_k\cdot(1-\sigma(a_k)) \tag{6}$$

So for $j = k$:

$$j = k : \frac{\partial L(\sigma(a))}{\partial a_k} = \hat{y}_k - y_k \tag{7}$$

For every partial:

$$\nabla_a L(\sigma(a)) = [\hat{y}_1 - y_1, ..., \hat{y}_{10} - y_{10}] \tag{8}$$

Finally:

$$\nabla_a L(\sigma(a)) = \hat{y} - y \tag{9}$$

## II. Final Structure and hyperparameters

The final structure of the model was set to:

- mini batch size of 128 observations
- learning rate of $7 \cdot 10^{-3}$
- lambda coefficient of 0
- Number of epochs 20
- network structured was kept as 3 layers (images below)

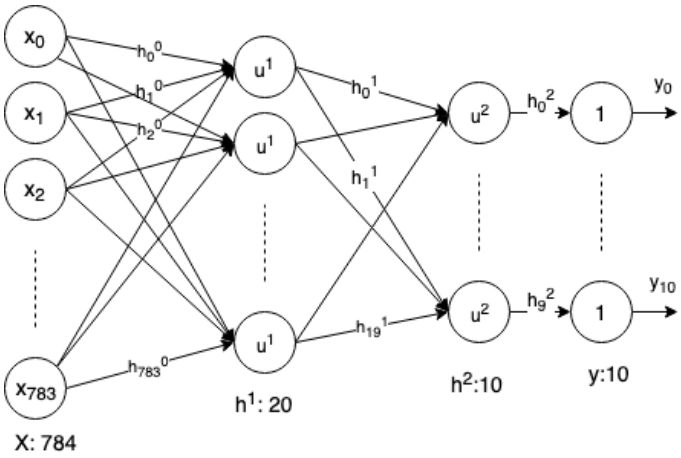


Fig. 1. Neural network structure.

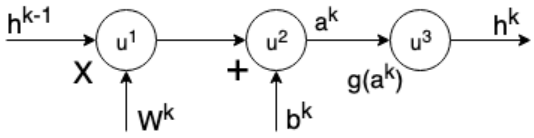


Fig. 2. Single neuron.

$$h^{(k)} = g(a^{(k)}) \tag{10}$$

$$g(a^{(k)}) = \sigma(a^{(k)}) = 1/1 + exp^{-a} \tag{11}$$

$$a^{(k)} = W^{\top} h^{(k-1)} + b^{(k)} \tag{12}$$

## III. Learning curve

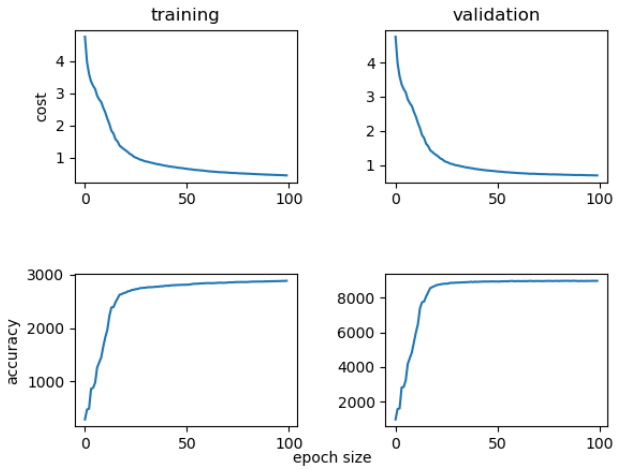


Fig. 3. Learning curve.

Based on the learning curve, we decided to choose a size of 20 for the epochs given that, starting at that point, the benefits on the reduced cost start to decrease significantly.

## IV. Accuracy

We tried to train our model on training data set and received a test accuracy of **90.81%**.

In order to increase the model performance we tried to train the model on training and validation data set together, we test it over the test data set obtaining an accuracy of 87.89%.

Instead of increasing, our testing accuracy decreased when we trained on training and validation data set together. Training the model on just training data set containing 3000 inputs is giving better results, thus we decide to proceed with just training data set only.