PROJECT 2. MULTI-CLASS CLASSIFICATION USING NEURAL NETWORKS

Thana Shree Jeevanandam
University of Buffalo
thanashr@buffalo.edu

Abstract

Neural Networks,the major research area of both neuroscience and computer science is modelled loosely on human brain. A neural net consists of thousands or even millions of simple processing nodes that are densely interconnected. Machine Learning,the art of making the computers learn by themselves is growing tremendously in all its aspects. It provides the machines to learn by recognising patterns without human intervention. The topic under study in this project is to recognise an image and identify it as one of ten classes. The Neural Network with different layers and Convolutional Neural Network are used to handle the challenge. The single layer Neural Network is coded from scratch on python. Multilayer Neural Network and Convolutional Neural Network are implemented using open-source neural-network library, Keras. The validation and testing are done to check the prediction accuracy. The different evaluation metrices are used to check the effectiveness of the solution. On the whole, the system is trained on this dataset to understand the relationship between the features and the relevance judgements so that when a new data is given, it could predict with the highest accuracy.

1.INTRODUCTION

Fashion MNIST serves as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits. Class specific treatment can significantly reduce toxicity and increase the efficiency of the therapy. The goal is to increase the prediction accuracies of images. Neural Network is the appropriate analysis to conduct when the dependent variable is among multiple classes. After the task1 with single layer Neural Network, task2 deals with multilayers. Deep-learning networks are distinguished from the more commonplace single-hidden-layer neural networks by their depth; that is, the number of node layers through which data must pass in a multistep process of pattern recognition. Task3 is a regularized version of task2 which uses CNN as the model deals with analyzing visual imagery images.

2. DATASET

The Fashion-MNIST is a dataset of Zalando's article images used for the prediction task consists of 70,000 examples.Each example is a 28x28 grayscale image, associated with a label from 10 classes. Each pixel of the 784 pixels, has a single pixel-value, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. The Dataset is used to populate the Training, Validation and Testing sets. The Dataset contains,

Classes	10
Samples total	70,000
Pixel value	0 - 255
Height of image(in pixel)	28
Width of image(in pixel)	28
Total pixel/image	784

The 10 classes are:

a) T-shirt/top
b)Trouser
c) Pullover
d) Dress
e) Coat
f) Sandal
g) Shirt
h)Sneaker
i) Bag
j)Ankle Boot

The task at hand is to predict the classes for each example using Neural Networks.

3. PREPROCESSING OF DATASET

The project starts with preprocessing of data. As a preprocessing measure, the process initiates reading the contents from the dataset in python. The original CSV data file is processed into a Pandas Dataframe. The steps in preprocessing of data includes,

1. Extracting data.
2. Splitting the dataset.

3.1 EXTRACTING DATA

Fashion MNIST dataset is downloaded and processed into a Numpy array that contains the feature vectors and a Numpy array that contains the labels using fashion mnist reader notebook present inside the scripts folder.

3.2 SPLITTING THE DATASET

The next step of preprocessing is partitioning datasets into Training, Validation and Testing sets. The Training data comprises of 80% of the overall dataset; Validation set comprises of 10% and Testing comprises of the last 10%. The partition should be such that there is no overlapping of data in any of the datasets. The model is then trained on the Training dataset and tested on the Validation and Testing sets.

4.PROCESSING OF DATASET

The preprocessed dataset is all set to get into the process.In task1,the one hidden layered Neural Network uses Gradient Descent to train the model using a group of hyperparameters. In task2, multi-layered Neural Network and in task3, Convolution Neural Network are implemented using Keras using a group of hyperparameters. The steps in the processing include,

4.1 TASK 1

4.1.1 INITIALIZING THE WEIGHTS

The only set of parameters controlling how accurate our model is are the weights and the bias. The changes in the hyperparameter is reflected in the model's accuracy.They include:

1.Learning Rate

2.Epoch (Number of iterations)

3.Bias

4.Batch Size

As the first step, the weights are initialized with respect to the shape of x_train.

4.1.2 SINGLE LAYERED NN

In this project , it takes the 784 sized images and matches them to the respective classes.

The two propagations here are,

1.Forward Propagation

2.Backward Propagation

4.1.3 FORWARD PROPAGATION

The two operations as a part of the forward propagation process are,

1.Applying a linear transformation on the input features

2.Applying a non linear transformation (sigmoid) and getting the hidden layer

3.Applying softmax on the hidden layer to get the output.

```
z1 =  dot( x_train,w1) + b1
p1 = sigmoid(z1)
z2 = dot( p1,w2) + b2
p2 = softmax(z2)
```

4.1.3.1 SIGMOID FUNCTION

The activation function of Logistic Regression is the sigmoid function. It has the range between 0 and 1 which can be interpreted as uncertainties for the outputs of 0 and 1 in the model.

The sigmoid function applies a non linear transformation onto the initial value and the range of the sigmoid function is a set of real values between 0 and 1.

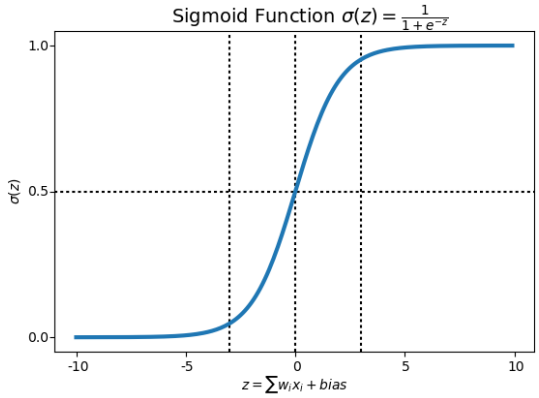

Fig 4.1: SIGMOID FUNCTION

4.1.3.2 SOFTMAX FUNCTION

The softmax function is a more generalized logistic activation function which is used for multiclass classification.The main advantage of using Softmax is the output probabilities range. The range will 0 to 1, and the sum of all the probabilities will be equal to one. It returns the probabilities of each class and the target class will have the high probability.

The formula computes the exponential (e-power) of the given input value and the sum of exponential values of all the values in the inputs. Then the ratio of the exponential of the input value and the sum of exponential values is the output of the softmax function.

4.1.4 BACKWARD PROPAGATION

The weights are initialized randomly, Backward Propagation is done with gradient descent algorithm to stabilize them. We find the extent to which the weights are correct in our assumption.

Dot(Multiply(Dot((a2 - y), w2), Multiply(a1, 1-a1)), x)

The loss is calculated for the training data and the softmax's output. This is used to update the weight.

4.1.4.1 GRADIENT DESCENT

The cost must be decreased in order to increase the accuracy of the model. The Gradient Descent for logistic regression is used to train the model using a group of hyperparameters to bridge the model's output and the true output together with a decrease in the loss/error/cost function.

$$W = W - \alpha \frac{\partial J}{\partial W}$$
$$b = b - \alpha \frac{\partial J}{\partial b}$$

The weights are stabilized by the Gradient Descent Algorithm. It is an optimization algorithm that finds the optimal weights (a,b) that reduces prediction error. It finds the least minimum between y_test (true) and output (predicted). Cost is the loss for the entire dataset.The Gradient Descent Algorithm is run repeatedly with updated weights to get the minimum cost.

4.2 TASK 2

The task2 undergoes the same forward and backward propagations. But it is not limited to a single layer, instead it uses a multilayered NN. The open source library, Keras is used for implementation.

4.3 TASK 3

The first processing step is to take only the important features from the images. Convulutional NN serves this aspect. The later steps are same as the task 1 and task 2 with forward and backward propagation.

5. ARCHITECTURE

5.1 TASK 1

Task1(Fig5.1) uses sigmoid and softmax functions for activation.

5.2 TASK 2

Task2(Fig5.2) uses RELU,tanh and softmax activation functions.

5.3 TASK 3

Task3(Fig5.3) uses two convolutional layers with RELU as activation function.

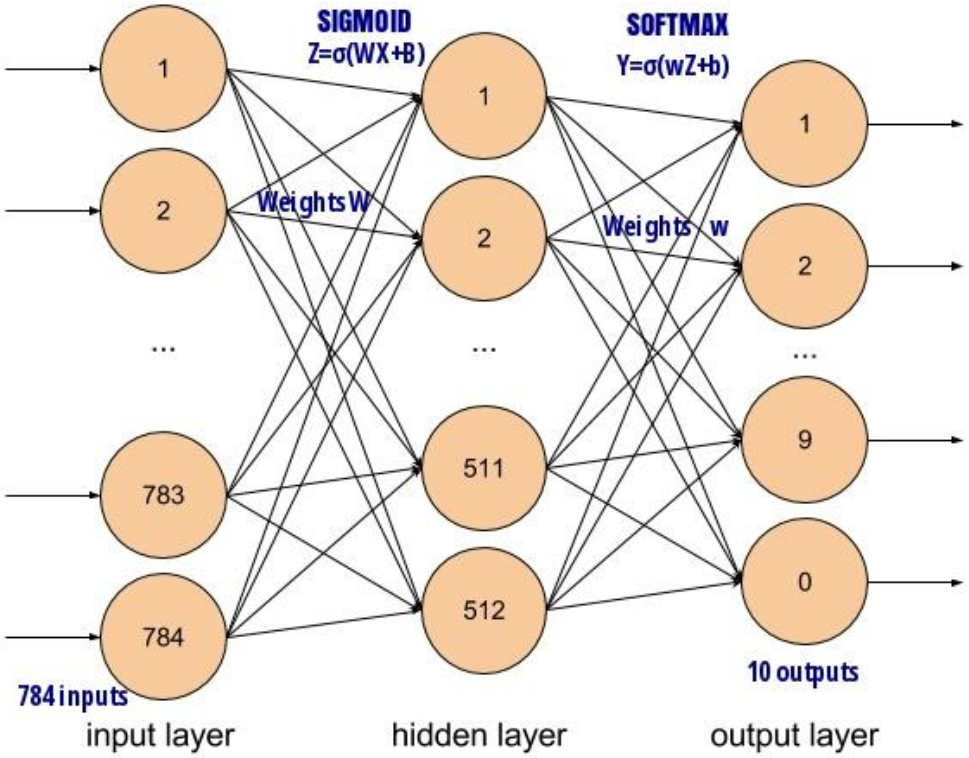

Fig 5.1: Architectural Diagram of Task 1(Single Layered NN)

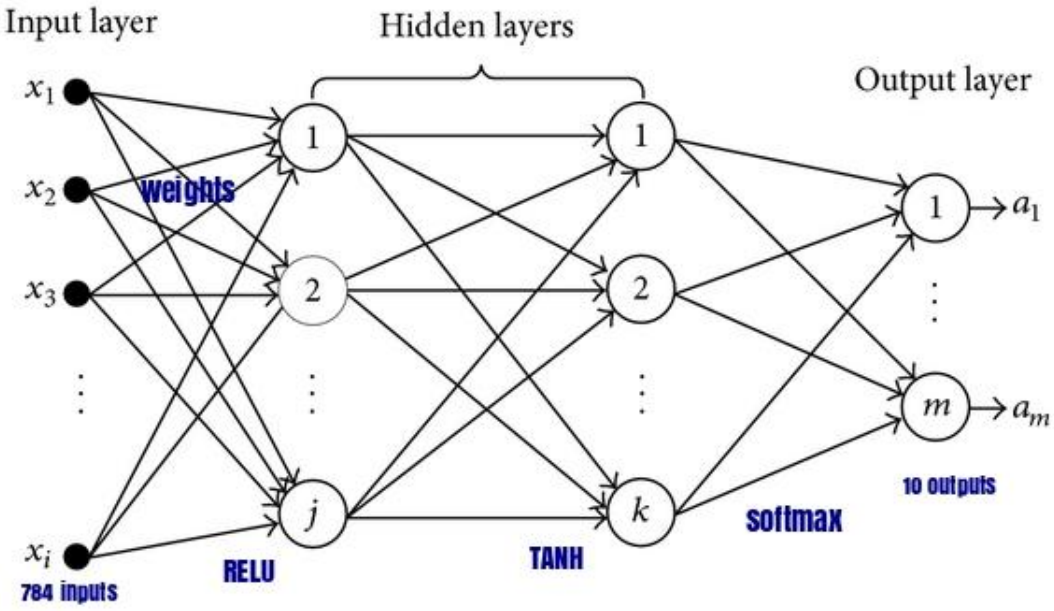

Fig 5.2: Architectural Diagram of Task 2(Multi Layered NN)

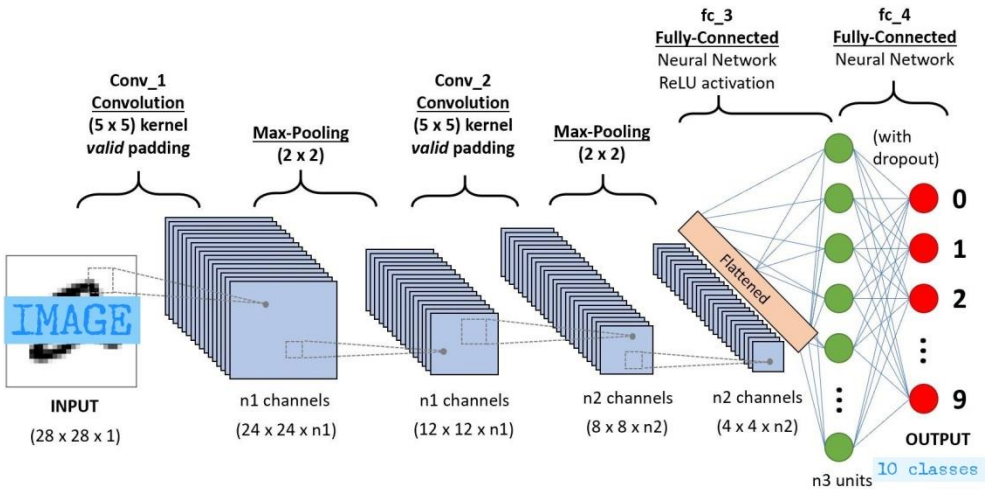

Fig 5.3: Architectural Diagram of Task 3(Convolutional NN)

6.RESULT

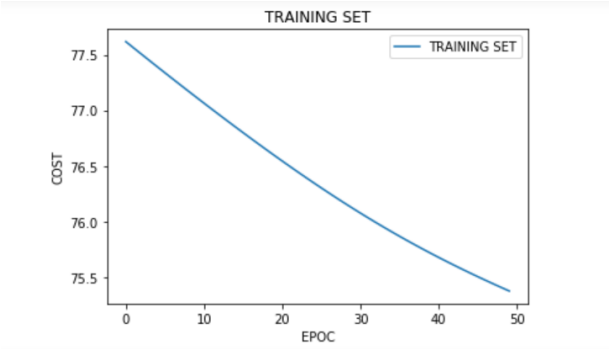

Fig 6.1: Task1: LOSS VS EPOCH

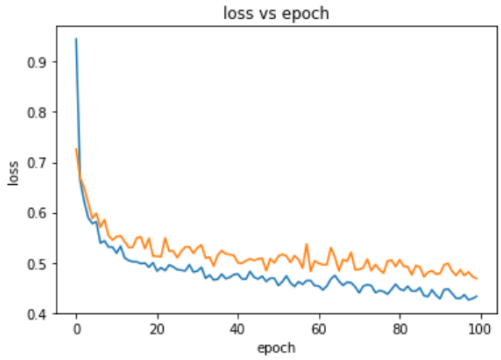

Fig 6.2: Task2: LOSS VS EPOCH (Accuracy : 83.81)

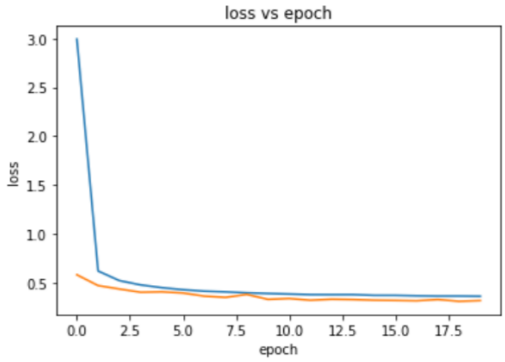

Fig 6.3: Task3: LOSS VS EPOCH (Accuracy : 88.29)

The above graphs represent the change in loss with respect to the change in epoch for the three tasks. The epoc is plotted along the x axis with its corresponding loss along the y axis. We can infer from the above graphs that the cost of validation set is lower than the training set with a smooth curve on convolutional NN. The hyperparameters are changed accordingly to get the highest prediction.Very high values of learning rate causes overfitting. Smaller learning rate skes the graph which is not optimal.

7.CONCLUSION

This project helps to get a clear idea of the different models of Neural Network. It has also helped understand the functioning of the gradient descent. The various values for hyperparameters are used to evaluate each model. The convolutional seems to be the best as it takes only the important features for prediction. This project helped get a deeper insights of the impact of the model selection and hyper parameter selection on the performance of the system. Thus, we performed a Neural Network analysis of different models on Fashion MNIST dataset to predict the correct classes.

REFERENCES

[1] Schlemper, Jo, et al. "A deep cascade of convolutional neural networks for dynamic MR image reconstruction." *IEEE transactions on Medical Imaging* 37.2 (2017): 491-503.

[2] https://sebastianraschka.com/Articles/2015_singlelayer_neurons.html - Single Layered NN

[3]https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f - Multi Layered NN

[4] Python functions - https://docs.scipy.org/