Prediction of Fashion Apparels using Neural Networks

Vineeth Thayanithi

Department of Computer Science and Engineering

University at Buffalo

1. Abstract

The Fashion Industry is one of the most trending industry in the current day. It is also evident from the amount of fashion apparels and accessories that are being introduced on a daily basis. Although right figures have not been established, it is estimated that around 80 to 100 billions of clothing pieces are made each year. The present trend has become so unexpected that even we humans as a species with prior knowledge about clothing apparels are sometimes unable to categorize a new clothing item. This being said, we try to mimic the recognition of human and the signaling pattern used by our information processing system to make a machine try to categorize an apparel to its respective class.

2. Introduction

We would be using a machine learning model called as neural networks for implementing this system. The basic intuition of the neural network is borrowed from the internal organization of the human body. The human body consists of thousands of nerves called neurons. The structure of the neuron is as below

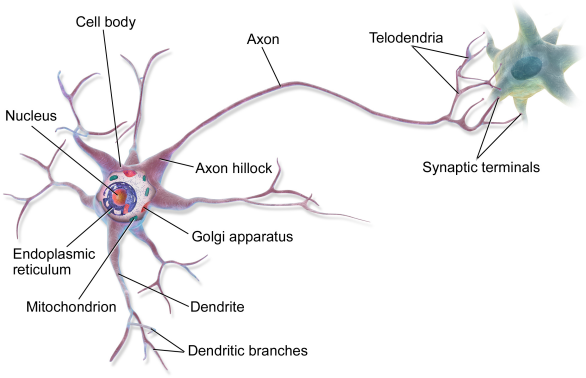


Figure 1: Neuron

Although the structure of a neuron contains many regions, we mainly concentrate on 2 of these. One is the sypnatic terminal and the other are the dentrites. Our body communicates information in the form of electrical signals through neurons. When the signal has reached some minimum threshold value, it activates the sypnatic terminal. These signals then pass through the axon reaching the dentrite branches and forwards the siganl to the neuron that is connected next to it.

A neural network follows a similar working pattern as described above. The neural network consists of collection of inputs called as a neuron. Upon reception of information, it can perform certain functions on the input and pass values to the neurons that are connected to it. The neural network can be visualized as below

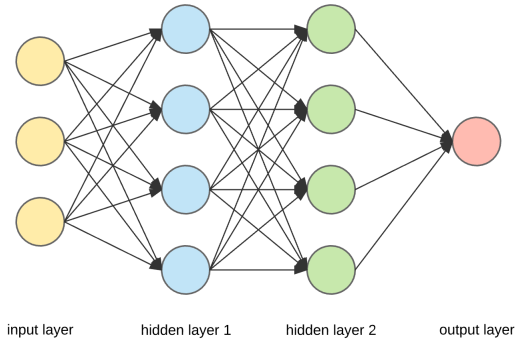


Figure 2: Neural Network

The neural network usually consists of an input layer, an output layer and one or more hidden layers. The input layer obtains the inputs from the data provided and the output is obtained after multiple operations within the hidden layer.

3. Dataset

We would be using the Fashion-MNIST data set for implementing the neural net from scratch, dense neural net using keras and convolutional neural net using keras. The Fashion-MNIST is a replacement dataset to the conventional MNIST.The MNIST consists of pictures of size 28x28 which contains handwritten numerical values from 0-9. Although, the MNIST is the most popular dataset for starting with neural networks, there also exists several issues with the MNIST. The MNIST dataset is claimed to be easy in a way that it can convultional networks can reach an accuracy of 99.7% and classic machine learning strategies producing an accuracy of 97%. The MNISt dataset is also overused and is not efficient in solving computer vision problems. Hence we move on to the Fashion-MNIST. The Fashion-MNIST has a similar collection with 60000

examples in the training set and 10000 in the test set.This dataset contains images of size 28x28 in grayscale of c The Fashion - MNIST has 10 classes in total

4. Pre-Processing

4.1. Neural Network from scratch

We use the load_mnist method from the mnist_reader.py for loading the train and the test sets. The parameter kind in the load_mnist funtions specifies whether the data is a train or a test set, train and t10k specifies the train and the test sets in the kind parameter. Once the dataset is imported, we would have to normalize the dataset. We could achieve normalization by dividing the datasets by 255(Maximum RGB value - Minimum RGB value). Normalization is required for neural networks to have a good performance. Normalization helps reduce the differences in the inputs given to the neural networks.

The labels in y_test and y_train as in the dimensions of 10000,1 and 60000,1 respectively with the label values lying within the range 0-9 denoting the class of clothing it belongs to . However, we need a 10000,10 and 60000,10 dimension in binary form with 1 denoting the class being true and 0 being the class being false. Hence, we do what is called as the one hot encoding. We import OneHotEncoder from sklearn.preprocessing to achieve this. The fit_transform() method is used for one hot encoding.It calls fit() and transform() together. The fit method computers mean and std. dev to center the data.

4.2. Multi Layer Neural Network with keras

Importing and normalization of the dataset is the same as it was for task1. For one hot encoding we use keras.utils.to_categorical() method. This performs the same task as the OneHotEnconder() from sklearn.preprocessing.

4.3. Convolutional Multi-Layer Neural Network with keras

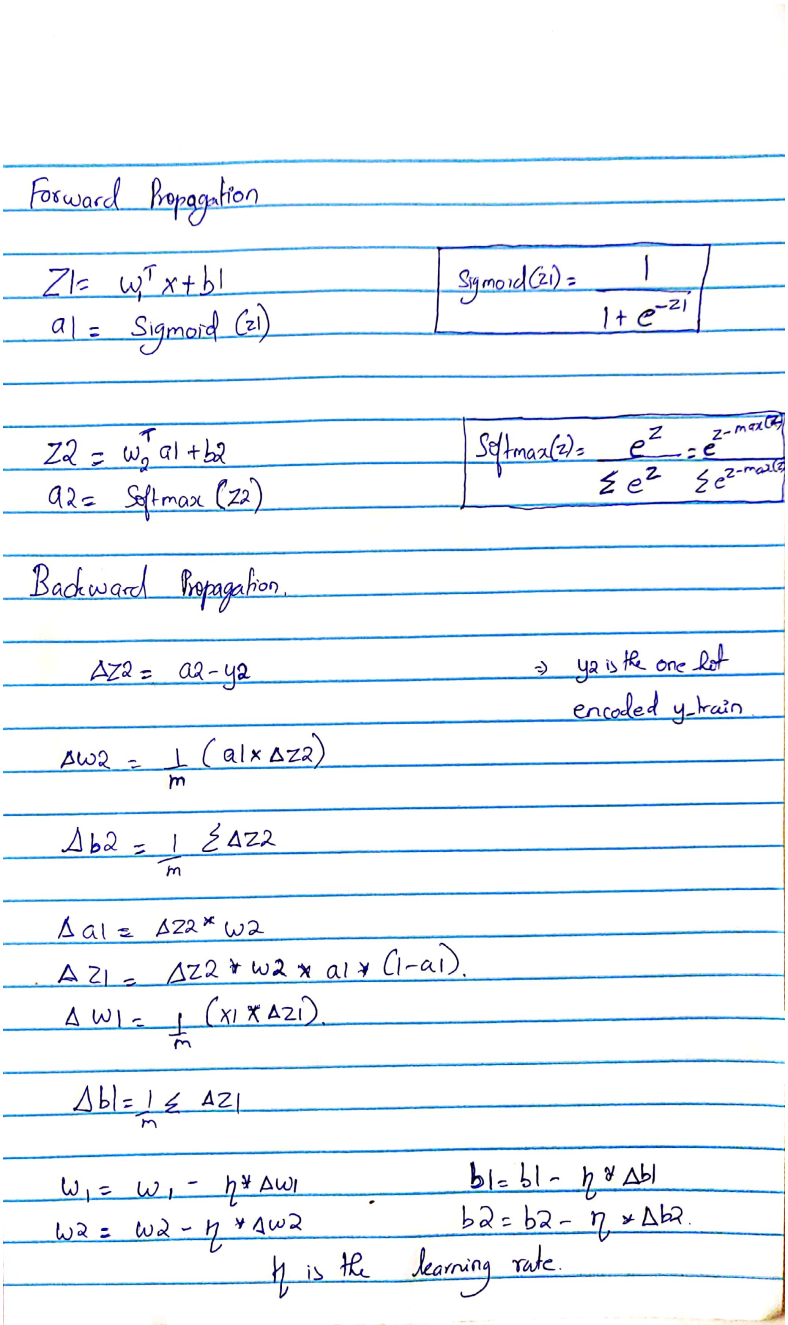
All the preprocessing tasks are the same as task 2 with the only difference being x_train and x_test are reshaped to (60000,28,28,1) and (10000,28,28,1) to operate with the convolutional kernel.

5. Architecture

5.1. Neural Network from scratch

The neural net can be seen as an extended version of logistic regression. For this task we use 3 layers. One layer is used as the input layer, the second layer is the hidden layer and the third layer is the output layer. We need 2 weights for this architecture and the shape of the weights would be (Number of Input feature, Number of nodes in the hidden layer1) and (Number of nodes in the hidden layer1, Number of output classes), i.e. (784,64) and (64,10). The biases

would be zeros of size (64,10) and (10). We use gradient descent for updating the weights during each epoch. The equations for updating weights is given below.



5.2. Multi-Layer Neural Networks using Keras

Keras is a high level deep learning library that uses either tensorflow or theano as its backend. For implementing the multilayer network, we construct a Sequential model with 3 Dense layers. Dense layers has a structure in which each node is connected to every other node in the next layer. The hidden layer has a Sigmoid activation and the output layer has a softmax activation. The model is compiled with Stochastic Gradient Descent optimizer with categorical cross entropy as its loss function.

5.3. Convolutional Neural Networks using Keras

Convolution is a mathematical operation between 2 inputs and is defined as how one input changes due to the other input. For implementing the Convolutional neural network, we construct a Sequential model with 2 Convolutional layers with a kernel size of (3,3) . Max pooling is applied on the to the output of the convolutional layer.MaxPooling is the process of obtaining the maximum value when a window slides across the given input. The max pooling helps to reduce the dimentionality of the Conv2D layer. To prevent over-fitting we introduce a drop after each max pooling. The output is then flatten and is sent to 2 other dense layers. These layers use sigmoid and softmax as their activation function. The model is then compiled with Stochastic Gradient Descent optimizer and Categorical_crossentropy loss

6. Result

6.1. Neural Networks from scratch

After finding the weights from gradient descent, we run the model against the test data set. The model produces an accuracy of 76% against the test set. The figure below shows the loss vs epochs graph of the model

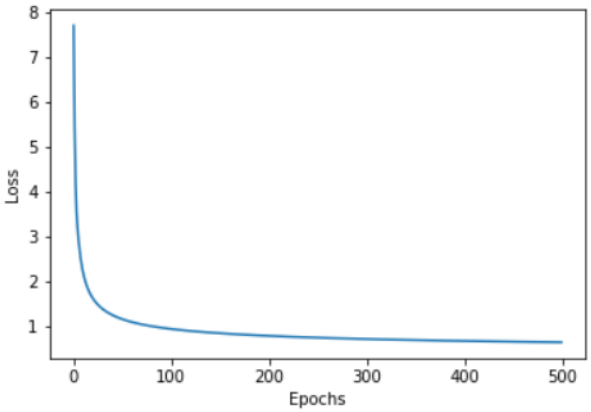


Figure 3: Loss vs Epochs

6.2. Multi-Layer Neural Networks using Keras

This model produces an accuracy of 86.58 % when it is run against the test data with a validation accuracy of 87%. The model is run for 40 epochs and we observe a loss of 0.3225.The loss vs epochs graph is as below

```
Accuracy: 86.58
```

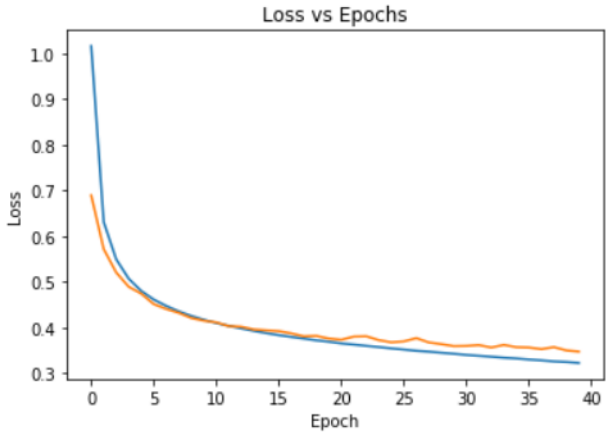


Figure 4: Loss vs Epochs

6.3. Convolutional Neural Networks using Keras

The CNN performs the best amongst all the models that we have built, the CNN has a 90.02 % accuracy on the test data with 20 epochs.

```
Accuracy: 90.92
```

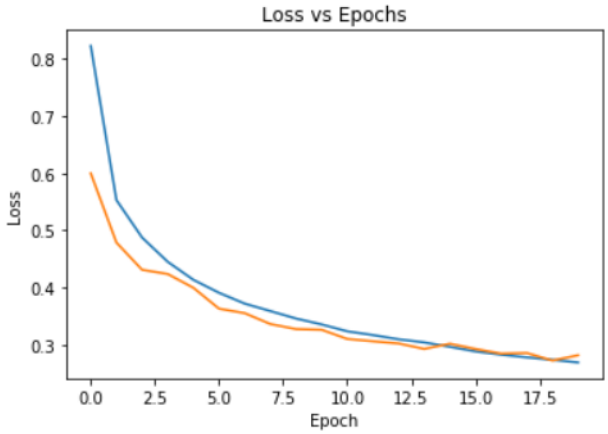


Figure 5: Loss vs Epochs

7. Conclusion

We have produced 3 neural network models that help us to classify clothing classes. The first model has an accuracy of 76%, the second model that is a multilayer neural network that is built using the high level library keras has an accuracy of 86.58%. However, we observe that the convolutional neural network built using keras is the best method for solving computer vision problems which classifies the Fashion - MNIST dataset with an accuracy of 90.92 in just around 20 epochs.

8. Reference

1. Project-2 Description
2. Recitation Sheets
3. Softmax Implementation
-https://stackoverflow.com/questions/34968722/how-to-implement-the-softmax-function-in-python
4. Keras Documentation - https://keras.io/
5. Fashion - MNIST - https://github.com/zalandoresearch/fashion-mnist
6 Lecture Slides - Prof. Sagur Srihari