Basic Of Deep Learning - Mid Semester Project Part 1

Ori Malca (ID. 315150599), Maor Sisai (ID. 311304059)

Submitted as mid-semester project report for Basic Of Deep Learning course, Colman, 2023

1 Introduction

Our project solves a binary classification problem on 2 fashion items from the MNIST-Fashion data set.

1.1 Data

The data we received is a labeled data set that contains 70,000 black and white images of fashion items. The size of each individual image is 28x28 pixels. There are 10 types of fashion items in the data set. The data is labeled from 0 to 9 as follows:

Label	Class
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

1.2 Problem

The problem we are dealing with is a binary classification between 2 fashion items from the data set while reaching above 90 percent accuracy on the test set(according to the assignment requirements).

2 Solution

We solved this problem using a simple feed-forward artificial neural network with 3 layers and a non-linear activation function at the layers(so that the classifier produced after the training process will be non-linear) as has been taught in the lectures. First we processed the data in a way that we split the data set into train set and test set including only 2 fashion items instead of 10 (we also normalized the data - scaled each sample/image by 1/255). Then, we trained the model for about 2-3 minutes. After the training phase has been complete, the model predicted above 90 percent accuracy on the test set.

2.1 General approach

Our general approach to solve this problem is using a simple feed-forward ANN (as described at the solution section) with SGD(stochastic gradient descent) as the optimizer of the model, while the loss function we are using is Binary Cross Entropy (since this is a convex function) and a non-linear activation at the layers.

Of course we could use a more complicated architecture to solve this problem and probably get a higher number of accuracy percentage (for example with a CNN we could probably achieve above 95 percent accuracy) but we found this architecture to be ”good enough” in order to solve this kind of problem with an accuracy higher than 90 percent.

2.2 Design

The platform we have been using is Google Colaboratory. We used NumPy library for metric calculations. We used scikit-learn library in order to fetch the MNIST-Fashion data set and calculate metrics on the train set and test set. We used Matplotlib library in order to get a visualization of those metrics.

The architecture of the model is a feed-forward artificial neural network with 3 layers. 784 neurons on the first/input layer(784 neurons as the number of pixels in each image), 64 neurons on the second/hidden layer and 1 neuron on the third/output layer(1 neuron on the output layer because we are dealing with a binary classification problem). The ANN have a total of 50305 learnable parameters(including the biases on each layer - we used a bias on the hidden layer and the output layer). The activation function on the hidden and the output layer is sigmoid. We used BCE(Binary Cross Enropy) as our loss function since BCE provides a convex loss function(thus we are sure the loss function will converge into a global minimum point). We used stochastic gradient descent as the optimizer of the model in order to reduce the lose, although mini-batches could probably be a better choice but we found SGD ”good enough” to deal with this kind of problem. We used a learning rate of 0.01 and 10 epochs as the values for those hyperparameters. It took about 2-3 minutes to train the model for 10 epochs.

This is a pseudo code of Stochastic Gradient Descent algorithm:

Algorithm 1 Stochastic Gradient Descent

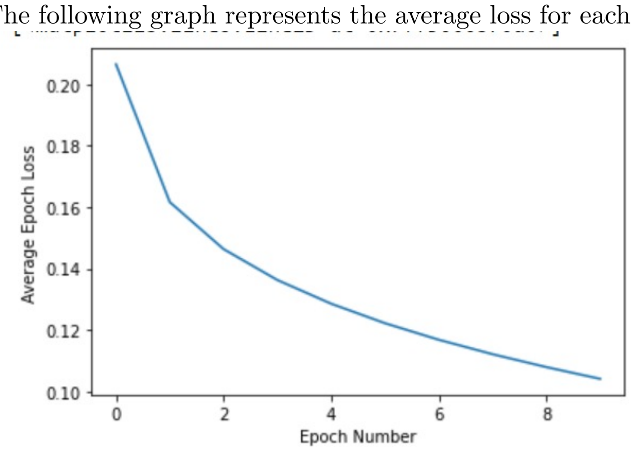
```
    init_params(W1, b1, W2, b2)                ▷ initialize the model parameters
    i=0 to num_of_epochs
        j=0 to num_of_trainingExamples
            # Forward propagation:
            Z1 = $W1 \cdot X[j] + b1$
            A1 = sigmoid(Z1)
            Z2 = $W2 \cdot A1 + b2$
            A2 = sigmoid(Z2)

            # Compute loss:
            loss = binary_cross_entropy(A2, Y[j])

            # Backward propagation:
            dZ2 = $A2 - Y[j]$
            dW2 = $dZ2 \cdot A1^T$
            db2 = $dZ2$
            dA1 = $W2^T \cdot dZ2$
            dZ1 = $(A1 \cdot (1 - A1)) \cdot dA1$
            dW1 = $dZ1 \cdot X[j]^T$
            db1 = $dZ1$

            # Update weights:
            W2 = $W2 - learning\_rate \cdot dW2$
            b2 = $b2 - learning\_rate \cdot db2$
            W1 = $W1 - learning\_rate \cdot dW1$
            b1 = $b1 - learning\_rate \cdot db1$
```

3 Base Model

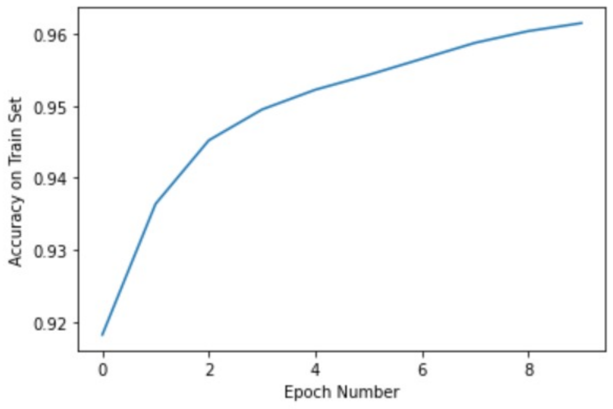
Our base model is a feed-forward artificial neural network with 3 layers(input layer, hidden layer and output layer), sigmoid activation on the layers. On the first layer there are $28 \cdot 28 = 784$ neurons(as the number of pixel in a single image), on the hidden layer there are 64 neurons and on the output layer there is a single neuron(since the problem we are dealing with is a binary classification). The loss function we are using is BCE since BCE provides for us a convex loss function. The total number of learnable parameters in the network is 50305 (including the biases) and we are using SGD as the optimizer of the model.

3.1 Results and Metrics

The following graph represents the average loss for each epoch on the train set:



This graph represents the accuracy for each epoch on the train set:



This is the prediction of the model on 2 samples from the test set:

```
Real= 1 Predicted= [[0.93133163]]
```

```
Real= 0 Predicted= [[0.00026021]]
```

This is the accuracy on the test set: 0.9610111396743788

This is the confusion matrix on the test data:

TN=1154	FP=61
FN=30	TP=1089

4 Discussion

After training the model, we were able to perform the task with an accuracy above 90 percent. On the training process we could observe the average epoch loss decreases while the epoch number increases and the accuracy on train data increases while the epoch number increases. It was a satisfying thing to see.

Overall this project was really enjoyable and satisfying to do. We understand the math of the backpropagation process much better and we learned whole lot from this project.

5 Code

https://colab.research.google.com/drive/1CbNeRMO-EVTe1vh8di8ShKRMUrIt5ugj?usp=sharing