Deep Learning - Mid Semester Project

Itay Etelis (ID. 209041474), Gilad ID. 206947814)

Submitted as mid-semester project report for Deep Learning course, BIU, 2023

1 Introduction

The MNIST dataset has long been used in the machine learning community as a benchmark for evaluating the performance of algorithms on image classification tasks. However, the low complexity of this dataset has led to the development of more challenging variants, such as the Fashion-MNIST dataset.

In this project, we seek to solve the Fashion-MNIST problem using a neural network. This will provide us with an opportunity to explore the capabilities and limitations of deep learning algorithms for image classification tasks. We will use the "language" of deep learning to analyze the data and the problem we are trying to solve, with the goal of developing a neural network that can accurately classify images of clothing items in the Fashion-MNIST dataset.

1.1 Data

Fashion-MNIST is a dataset of Zalando's fashion article images —consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each instance is a 28×28 grayscale image, associated with a label.

1.2 Problem

The objective is to identify (predict) two fashion products from the given images using various implementations of neural networks and compare their results (performance measures/scores) to arrive at the best model.

2 Solution

2.1 General approach

Our approach is rather straightforward — we're reviewing two implementations of neural networks to tackle this problem.
The first implementation is purely based on the 'NumPy' framework.
The second implementation makes use of the Keras framework.

2.2 Design

To estimate classes, a three-layered neural network is used. The three layers have the following units, together with the activations.

Input layer: 784 units (For 784 pixels)
Hidden Layer: 32 units with sigmoid activation
Output Layer: 1 units(For the binary classification) with sigmoid activation

The weight matrices are used to map information from one layer to the other. the matrices:

$W1$, which has dimensions 32 into 784, maps the input data to the hidden layer
$W2$, dimensions 1 to 32, maps the hidden layer to the output layer.

The results are estimated using the weight matrices W1,W2,b0 and b1 and the loss function used in this part was:
The cross-entropy loss penalizes high-confidence classifiers that create wrong estimates of the actual class.
In order to estimate the classes correctly, the neural network has to alter the weight matrices W1,W2,b0 b1 such that the cross entropy loss is minimized.
the way optimizer was selected to achieve this was the stochastic gradient descent.

The neural network was trained for ten Epochs, in approximately nine minutes, and achieved an accuracy of 99.4%

2.3 Base Model

The Base model was built using Keras's Sequential model with a flatten layer in order to convert the 28*28 image matrix to 28**2 1D-array which could be used with Neural Networks,
using dense layers with the following structure:

Input layer: 784 units (For 784 pixels)
Hidden Layer: 40 units with sigmoid activation
Output Layer: 2 units(For the binary classification) with a softmax activation

the first activation is sigmoid and the second is softmax. the inner amount of neurons was chosen so that there is a proportional reduction in size since 784 / 40 = 40 / 2. Our initial optimizer was SGD because it's a basic optimizer that can do the job, and we wanted to so first how good the results would be without using any advance optimizers or activation functions. We also chose 10 epochs because it's a reasonable amount according to different resources that we've checked. The base model achieved 96.85% accuracy which is pretty impressive for a basic model.

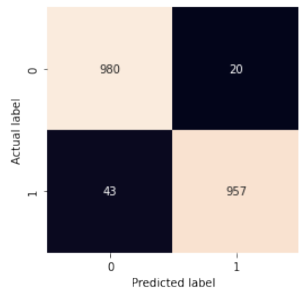

Figure 1: Confusion Matrix of the base model

2.4 First Experiment

In the first experiment we decided to change the activation function from sigmoid to relu and add an additional hidden layer of 150 neurons, thus changing the structure to the following structure:

Input layer: 784 units (For 784 pixels)
Hidden Layer: 150 units with relu activation
Hidden Layer: 40 units with relu activation
Output Layer: 2 units(For the binary classification) with a softmax activation

The reason for the changes are that relu seems to produce better results than sigmoid, and it's a popular activation to choose, and another layer in the network could enhance the network's ability to learn better and thus help to the performance. Indeed when we tested the model, it received a 98.4% accuracy!

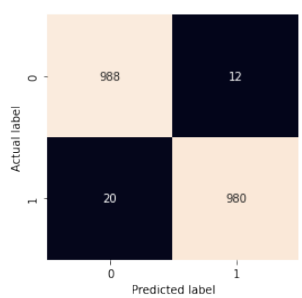

Figure 2: Confusion Matrix of the first experiment

2.5 Second Experiment

In the second experiment, we've decided to increase the number of epochs to 12, in order to give the network more time to learn (keeping checkpoint to best model, so there is no "best model" to lose in case the model will perform worse with time). Moreover, we change the optimizer from SGD to Adam, which is an enhanced version of SGD (which has been proven practically to outperform SGD in many use cases) and so we tried it. After the model has been trained and tested, we saw that the additional epochs did not benefit much, but the change in the optimizer has led to a tremendous increase in accuracy, from 98.4% to 99.4%, this is very close to 100% which will form a perfect model.

2.6 Best model Results and Metrics

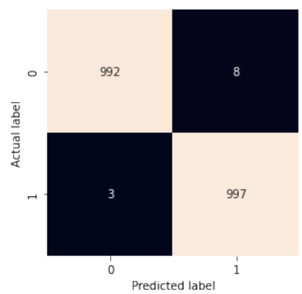

Figure 3: Confusion Matrix of the second experiment

As seen by the confusion matrix, out of 2000 images, the model only missed classified 11, 8 of which are T-shirts and 3 trousers.

3 Discussion

The Fashion MNIST problem is a well-known challenge in the machine learning community that involves classifying images of clothing items into 10 different classes. One approach to solving this problem using the Keras deep learning framework involves building a neural network. In our experiments, we achieved a high accuracy on the test dataset (99%).

This project allowed us to gain a better understanding of the challenges of computer vision and modern machine learning techniques for addressing these challenges. In future work on this dataset, we plan to investigate the use of different neural networks, such as convolutional neural networks.

Our experience with using Keras to implement a neural network brought us closer to the state-of-the-art for this problem. Overall, this was a valuable learning experience, particularly as the importance of computer vision continues to grow.

4 Code

https://colab.research.google.com/drive/1zujcu0BGnRFhveKGn9UlRqT2tzDE4VdW?usp=sharingscrollTo=w
Notebook