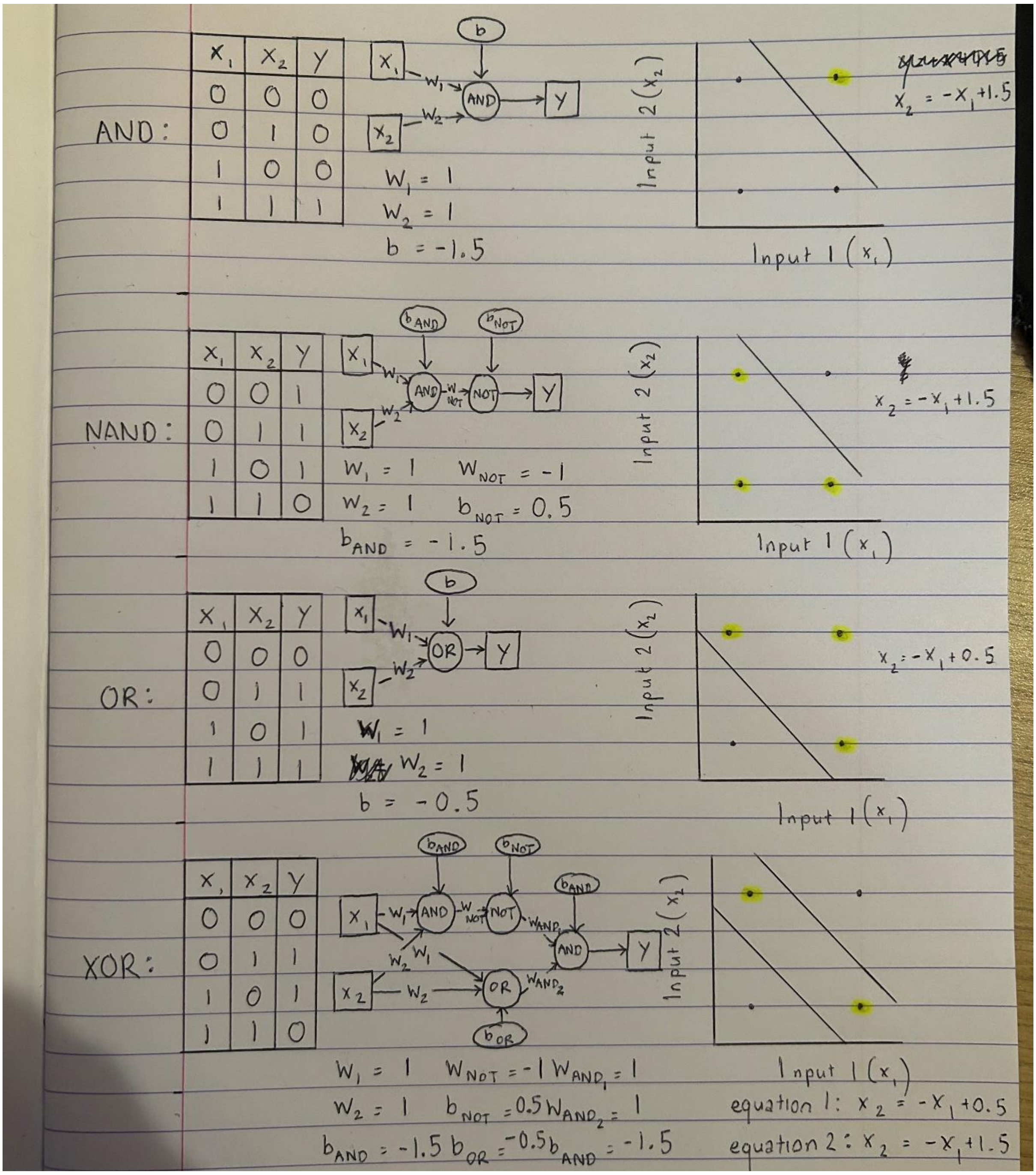
EE40098 Coursework 1: Neural Networks [199084087]

Exercise 1:



Where ‘b’ stands for biases and ‘w’ stands for weights. The highlighted areas on the state space graph indicate where the perceptron fired (output = 1). The graph equations are the linear separators for each logic gate. The YouTube link at the end of the report presents the truth tables and state space graphs in more detail and shows test examples for the other exercises.

Exercise 2:

a) The initial parameters consisted of a learning rate of 0.1 and 100 hidden nodes, which produced a score of 96.4%. Decreasing the learning rate to 0.05 and doubling the number of hidden nodes increased the accuracy however, expanding the number of hidden nodes past 200 resulted in a drop in performance. The best score achieved was 97.81% as shown in figure 1. This was done utilizing 50 epochs, 10 nodes, 200 hidden nodes, and a learning rate of 0.01. After 10 runs, the average accuracy was 97.72%.

```
epoch:  48
Accuracy Train:  0.99435
Accuracy Test:  0.9781
epoch:  49
Accuracy Train:  0.9946333333333334
Accuracy Test:  0.978
```

Figure 1: The highest accuracy test results for the handwritten dataset.

b) Replacing the MNIST handwritten dataset (MHWD) with the MNIST fashion dataset (MFD), the network obtained a test accuracy of 80.06%. Changes were then made to the network structure; increasing the number of hidden nodes to 240 and using a learning rate of 0.04 resulted in an accuracy of 86.24% after 12 epochs. The best result produced was 88**.**53**%** after 43 epochs with 228 hidden nodes and a learning rate of 0.01 as shown in figure 2.

```
Accuracy Train:  0.9270833333333334
Accuracy Test:  0.884
epoch:  41
Accuracy Train:  0.9280833333333334
Accuracy Test:  0.8847
epoch:  42
Accuracy Train:  0.9287666666666666
Accuracy Test:  0.8853
```

Figure 2: The highest accuracy test results for the fashion dataset.

The accuracy of the MFD was considerably lower than the accuracy of the MHWD. A major factor for this is due to the increased complexity of the images. Compared to MHWD, the images in the MFD are highly dimensional; they contain more information, as shown in the video, making it harder to differentiate them. This is further proven when comparing the pixel densities of the images. Figure 3 highlights this, conveying that the more intense the colours, the easier it will be for the network to distinguish between each class. It can be seen that there a more milder areas in the MFD compared to the MHWD.

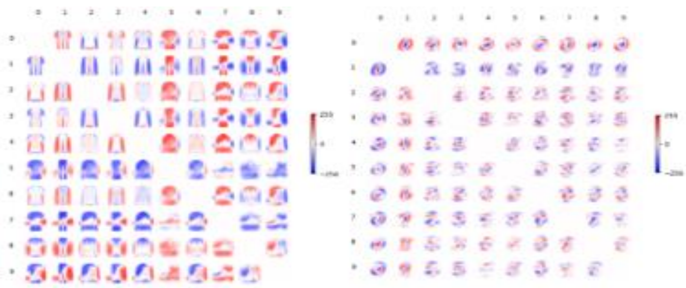


Figure 3: Pixel densities, the left being the MINST fashion dataset and the right being the handwritten dataset.

This implies that for the MHWD, it is possible to correctly distinguish between several digits by looking at a few pixels, whereas with the more diverse MFD, the algorithm has to learn more advanced features to consistently differentiate the individual classes reliably. Thus, it can be concluded that the MFD is more difficult to achieve high accuracy on, with a higher level of programming required to increase the accuracy.

c) To improve the overall score on both datasets, the problems with the current code need to be addressed. The first issue is overfitting. Overfitting occurs when a model becomes overly sensitive to the noise and detail in the training set, which has a detrimental effect on the model's ability to perform well on subsequent sets of data. Preventing overfitting is key to increasing the scores of both datasets. This can be done through regularization and hyperparameter tuning.

Hyperparameter tuning is the process of fine-tuning the parameters to perfectly fit our model. Although this was constantly done throughout the exercise, trial and error meant that the values were not perfect, as testing all possible parameter combinations within the limited period was far-fetched. This was evident when testing as a learning rate value too small resulted in long training times, whereas a value too high led to sub-optimal weights, which caused an unstable training process with lower accuracy. To combat this, switching from a fixed learning rate such as the 0.01 used in part b, to a variable learning rate would not only improve the speed of the training process but also increase the accuracy as dynamically reducing the learning rate helps the algorithm to converge faster and reach closer to the global minima, lowering the value of the cost function, which increases the predictive capability of the model. Additionally, incorporating multiple hidden layers into the network, rather than just 1 as we have done, creates a deeper and denser network (deep learning), which allows the model to learn more complex features. This would greatly improve the accuracy of both models, specifically the MFD, however it must be noted that too many hidden layers have an adverse effect and cause overfitting, which would decrease the models' accuracy.

Regularization refers to techniques that are used to calibrate machine learning models to minimize the adjusted loss function and prevent overfitting or underfitting. This is not present within the current model. After adding multiple hidden layers, the two techniques employed would be dropout regularization and L2 regularization. Dropout Regularization is the process of randomly dropping neurons while training. This prevents co-adapting, the process of units changing to fix mistakes of other units, which increases the models' accuracy by allowing for better generalisation, as the model relies less on specific neurons to produce an output, with a second benefit of reducing overfitting. Ideally, with 3 hidden layers, dropping 20-25% of the neurons after each pooling layer would lead to better scores. L2 Regularization helps to penalise the loss function more by adding another term to the loss function, the squared magnitude, as shown in figure 4. The higher loss function causes weights to be smaller, which reduces overfitting.

$$Loss = Error(y, \widehat{y}) + \lambda \sum_{i=1}^{N} w_i^2$$

Figure 4: New L2 Regularization equation.

The second issue is the dataset; more samples would produce more accurate results. This can be achieved through data augmentation, the process of applying random yet realistic changes to boost the diversity of the training set. These changes can include rotating some images by, for example, 8 degrees or zooming in/shifting the images in both directions. This prevents data scarcity, which is desired, as there would be less variance within the model. This would especially help the hand-written dataset, which contains some irregularities within the hand-written numbers.

A complete overhaul of the current model and using a different, higher-level model would also increase the accuracy. A convolutional neural network (CNN), such as LeNet-5, can be implemented. CNNs typically score upwards of 99% for the MHWD and 90% for the MFD. They are comprised of several node layers, each containing an input layer, hidden layers, and an output layer. Each node is interconnected and assigned an individual weight and threshold value, which are designed to learn the spatial properties of features, assisted by a process called backpropagation. A key feature of this is the kernel, which is a tiny array of numbers that aids feature extraction by allowing for the generation of a large number of feature maps that each represent a distinct feature of the input tensor. More complex models, such as Fine-Tuning DARTS or Shake-Shake produce accuracy ratings above 96% for the MFD.

YouTube Link to Code: https://www.youtube.com/watch?v=6gsn4cUTYN4&ab_channel=Zechi