CSI 5340 Assignment 2

Name: Alim Manjiyani

Student Number: 300229095

5th October 2021

This assignment deals with exploring and understanding different intricate concepts such as Backpropagation, Softmax regression, Multi-Layer Perceptron and Convolutional Neural Network.

1 Question 1: Backpropagation

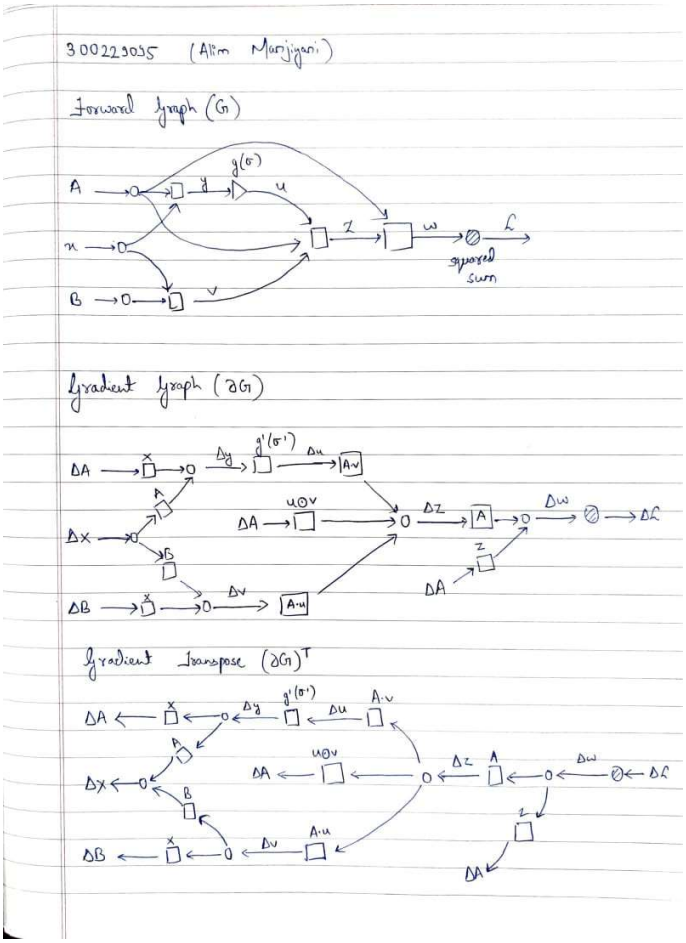

Fig. 1 Forward Graph, Gradient Graph and its transpose

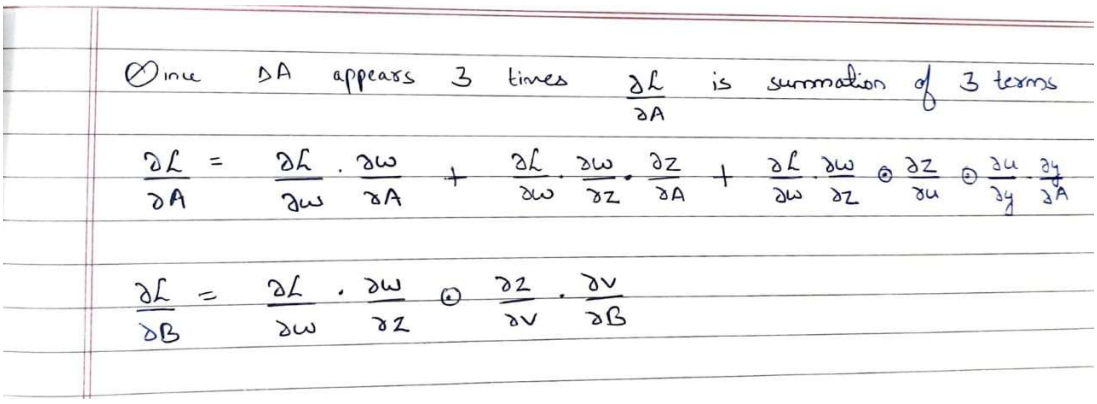

Fig. 2 Formula Used

The above-mentioned equation helps us to implement the backpropagation for the given network.

Parameter Settings:

Number of inputs (N): 100

Input Dimension (K): 5

Iterations: 1000

Optimizer: Gradient Descent

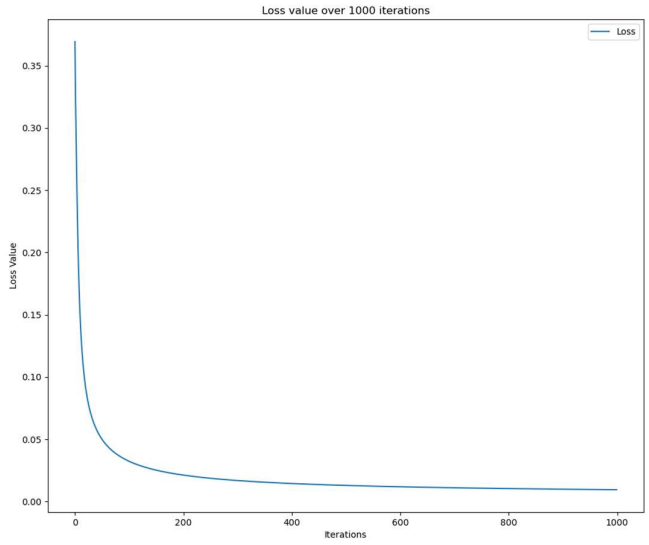

Fig. 3 Backpropagation Result

Conclusion:

Hence the derived network is correct, as the backpropagation used along with the gradient descent minimizes the loss function and helps the model to learn better.

2 Question 2: Softmax Hypothesis

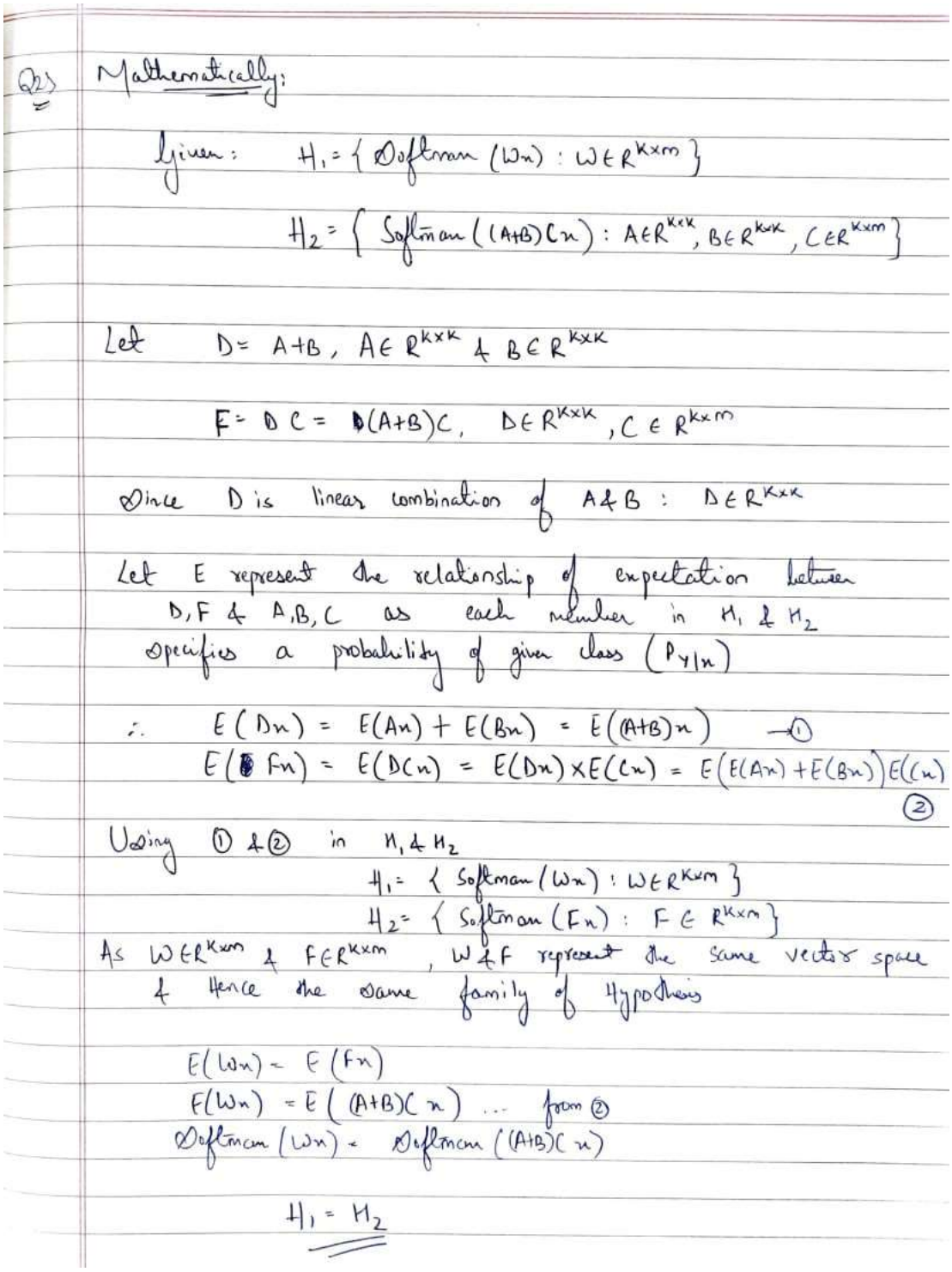

Fig. 4 Mathematical Proof

Q2)

K- Class Classification

Given: $H_1 = \{ Softmax(Wx) : W \in R^{K \times m} \}$

$H_2 = \{ Softmax((AB)(x)) : A \in R^{K \times K}, B \in R^{K \times K}, C \in R^{K \times m} \}$

To prove: $H_1 = H_2$

Theoritically

Given that 'W' represents all vectors in space $R^{K \times m}$. Also, A & B represents all vectors in $R^{K \times K}$ & C represents all vectors in $R^{K \times m}$

Let, $E = A \cdot B$. $E \in R^{K \times K}$ as it is the linear combination of A, B, E lies in the same space

Similarly; $F = E \times C$. $F \in R^{K \times m}$ & hence F represents all the vectors in space $R^{K \times m}$.

Since W & F represent the same vector space $W = F$

Hence, $H_1 = \{ Softmax(Wx) : W \in R^{K \times m} \}$

$H_2 = \{ Softmax(ECx) : E \in R^{K \times K}, C \in R^{K \times m} \}$

$= \{ Softmax(Fx) : F \in R^{K \times m} \}$

Hence $H_1 = H_2$

[Note: Even though $W \in R^{K \times m}$ each element in Wx can be represented as linear combination i.e. $(Wx)_i = W_i \cdot x$ where $i \in (0, .., K)$

Fig. 5 Theoretical Proof

3 Question 3: MNIST Classifiers

The three classifiers used are: Softmax Regression, Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN). The following results show the trend in the accuracy and loss/cost values as we combine different techniques like dropout and batch normalization respectively for each model.

3.1 Parameter Settings:

- Optimizer: Adam
- Dropout rate: 0.5
- Learning rate: 0.001
- Batch Size: 500
- Epochs: 20

1. Softmax:
    - Loss Function: Cross Entropy
    - Optimizer: Adam
2. MLP:
    - Layer 1: 512 nodes
    - Layer 2: 256 nodes
    - Layer 3: Softmax
    - Activation function: ReLU
3. CNN:
    - Layer order: Conv -> Max Pool -> Conv -> Max Pool -> Conv
    - Activation: ReLU

3.2 Comparison of Classifiers on vanilla settings

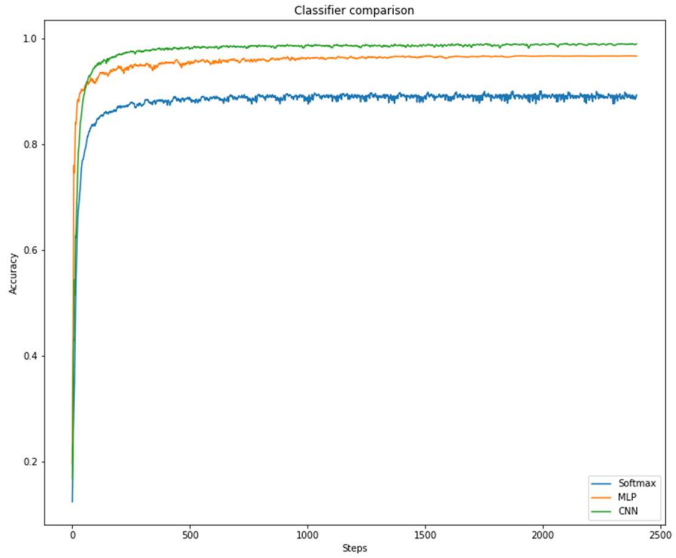

Fig. 6 Testing Accuracy of Classifiers

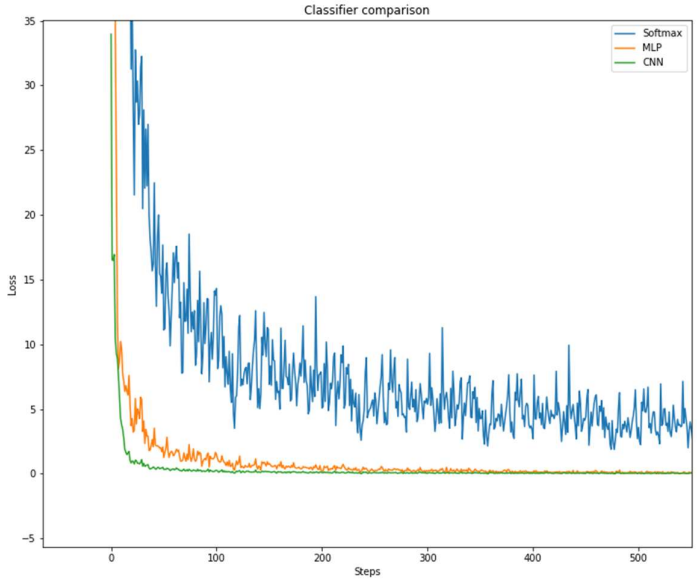

Fig. 7 Training Loss of Classifiers

Conclusion:

CNN outperforms Softmax Regression and MLP when initialized with same learning rate and in the same amount of epochs. While CNN and MLP show good performance, Softmax regression performs poorly in comparison, which can be attributed to the lack of model complexity or simplicity of the model. CNN prevailing over MLP can be attributed to lack of hidden layers in MLP or to the feature maps of CNN which, provide a better insight on the data.

3.3 Effects of Dropout and Batch-Normalization

3.3.1 Softmax Regression

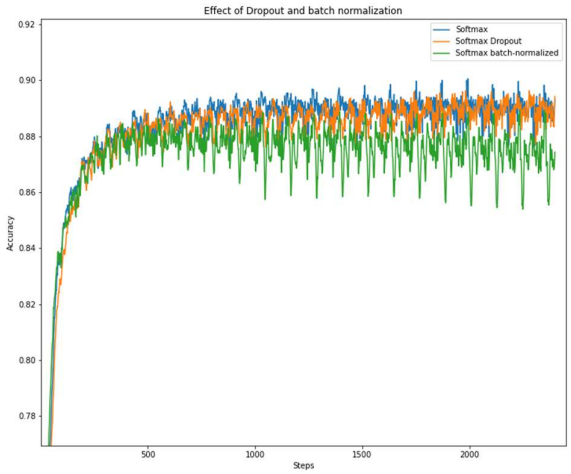

Fig. 8 Effect of Dropout and Batch Normalization on Softmax accuracy

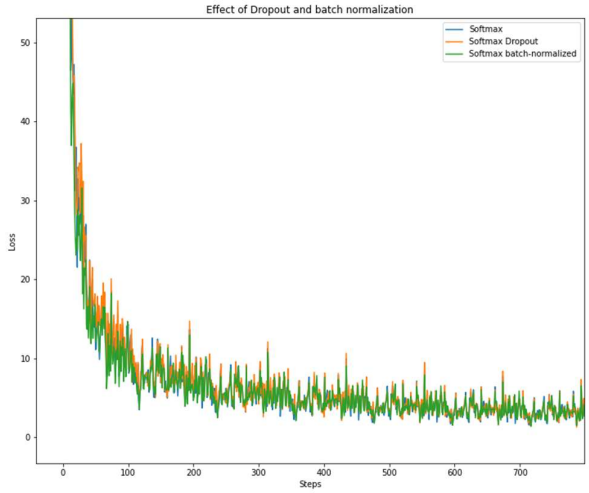

Fig. 9 Effect of Dropout and Batch Normalization on Softmax training loss

Conclusion:

It can be concluded that both dropout and batch normalization damage the performance of Softmax Regressor. The decrease in performance is due to the presence of only 1 layer. Dropping out of nodes just before the activation will change the result of probability distribution dramatically. Similarly, normalizing the weights just before prediction deviates weight vector for worse. Hence, it is not good to apply dropout and batch normalization on single layer models. Moreover, it is not a good idea to dropout nodes or normalize the weight vector just before the prediction.

3.3.2 MLP

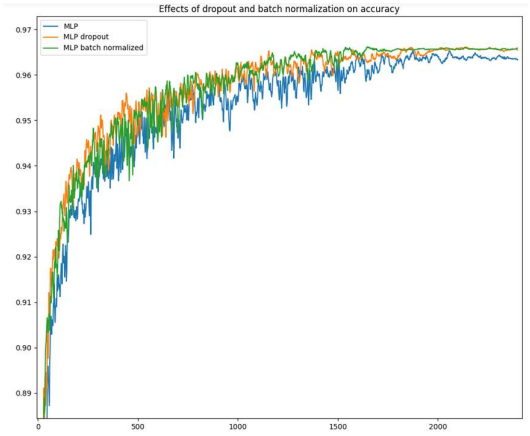

Fig. 10 Effect of Dropout and Batch Normalization on MLP test accuracy

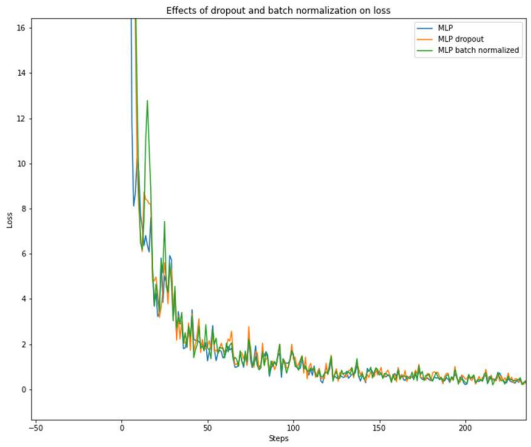

Fig. 11 Effect of Dropout and Batch Normalization on MLP training loss

Conclusion:

Although the three settings appear to have same loss, It can be inferred from Fig. 10 that, both Dropout and Batch normalization, increase the accuracy of the model. As the dropout is applied in every layer of the MLP, the dropout shows a significant increase in accuracy. Moreover, Batch-Normalization not only improves the accuracy but makes the model even faster, hence confirming to its theory.

3.3.3 CNN

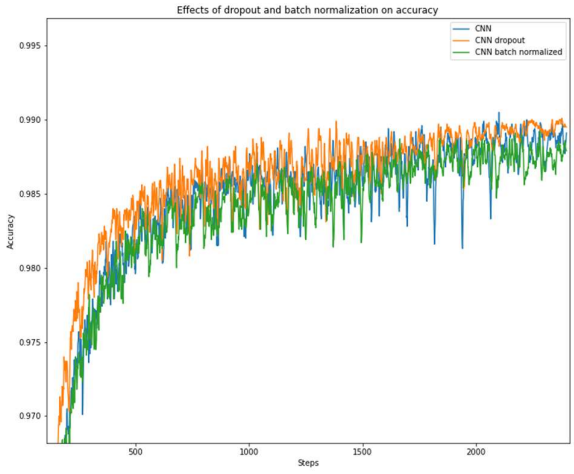

Fig. 12 Effect of Dropout and Batch Normalization on CNN test accuracy

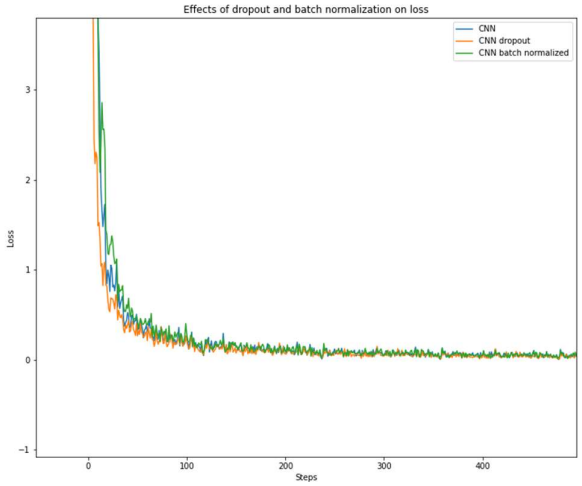

Fig. 13 Effect of Dropout and Batch Normalization on CNN training loss

Conclusion:

It can be clearly seen that the dropout increases the accuracy of the CNN, although the dropout can be applied in only 1 layer, the dropout technique proves its worth, as not only it increases the accuracy, it also converges faster.

It is surprising to see Batch-normalized CNN performing inferiorly to the dropout model. While both normalized and dropout surpass the default model, the normalized model is slower than the dropout model. One factor which can be the reason for this, is the fact that normalization occur in every layer whereas, the dropout is only on the last layer, hence explaining the faster convergence of dropout.

Overall: CNN prevails over the other models due to its complexity and the use of feature maps. Both dropout and batch normalization can significantly improve the performance of a given model, given that they are used correctly as seen above.