**EX1 - DL Basics**

**Theory**

Question 1

1. The shape on the input X is .   
   [m - training batch size as given]
2. The shape of the hidden layer’s weight vector is and the length of the bias vector is .
3. The shape of the output layer’s weight vector is and the length of the bias vector is .
4. The shape of the network output Y is

Question 2

The input images size is: .

The first convolutional layer outputs 100 feature maps.  
The number of weights for 1 feature map is (we added 1 for the bias term), therefore the number of weights of 100 feature maps will be .

Similarly, we can calculate the number of parameters for the 2nd and 3rd convolutional layers.

The second convolutional layer output 200 feature maps then the number of weights will be .

For the third convolutional layer we the number of weights is .

The total number of weights is :

Question 3

**batch mean :**

**batch variance :**

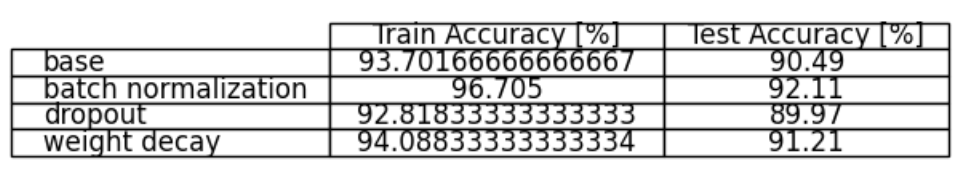
4. Note:
5. Note:
6. We can see that:  
     
     
     
   Back to the original expression:

**Practical**

Below is a table displaying the graphs for test versus train results for each technique utilized in the training of the LeNet5 network.

| **Without Regularization** | **Dropout** |
| --- | --- |
| **Train:** train/lenet5\_bnFalse\_dpFalse\_wdFalse **Test:** test/lenet5\_bnFalse\_dpFalse\_wdFalse | **Train:** train/lenet5\_bnFalse\_dpTrue\_wdFalse **Test:** test/lenet5\_bnFalse\_dpTrue\_wdFalse |
| **Batch Normalization** | **Weight Decay** |
| **Train:** train/lenet5\_bnTrue\_dpFalse\_wdFalse **Test:** test/lenet5\_bnTrue\_dpFalse\_wdFalse | **Train:** train/lenet5\_bnFalse\_dpFalse\_wdTrue **Test:** test/lenet5\_bnFalse\_dpFalse\_wdTrue |

Summarized Table

****

Conclusions

| **Train Set Accuracies** | **Test Set Accuracies** |
| --- | --- |
| 1. Batch Normalization 2. Weight Decay 3. No Regularization 4. Dropout | 1. Batch Normalization 2. Weight Decay 3. No Regularization 4. Dropout |

1. The model with **batch normalization has the highest accuracy** on both the training and test sets, indicating that this technique helped to reduce overfitting and improve generalization of the model.
2. The model with **weight decay also performed well**, with high accuracy on both the training and test sets. This suggests that adding weight decay regularization helped to prevent overfitting of the model.
3. The model with **dropout performed slightly worse than the model with no regularization**, with lower accuracy on both the training and test sets. This suggests that dropout regularization may not have been as effective in preventing overfitting for this particular model.
4. The model with **no regularization had high accuracy on the training set but lower accuracy on the test set**, indicating that the model **may have overfit to the training data** and may not generalize well to new data.

Overall, the results suggest that using batch normalization or weight decay regularization can help to improve the performance of the neural network, while dropout regularization may not be as effective for this particular model. It's important to note that these conclusions are based on our specific dataset and LeNet5 model architecture used, and may not necessarily generalize to other datasets or models.Therefore, it's always important to perform rigorous experimentation and cross-validation to draw solid conclusions about the effectiveness of different regularization techniques on different models and datasets.