Exercise 2

Deep Learning Course

Stav Bar-Sheshet (301558086) and Shlomo Spitzer (305143315)

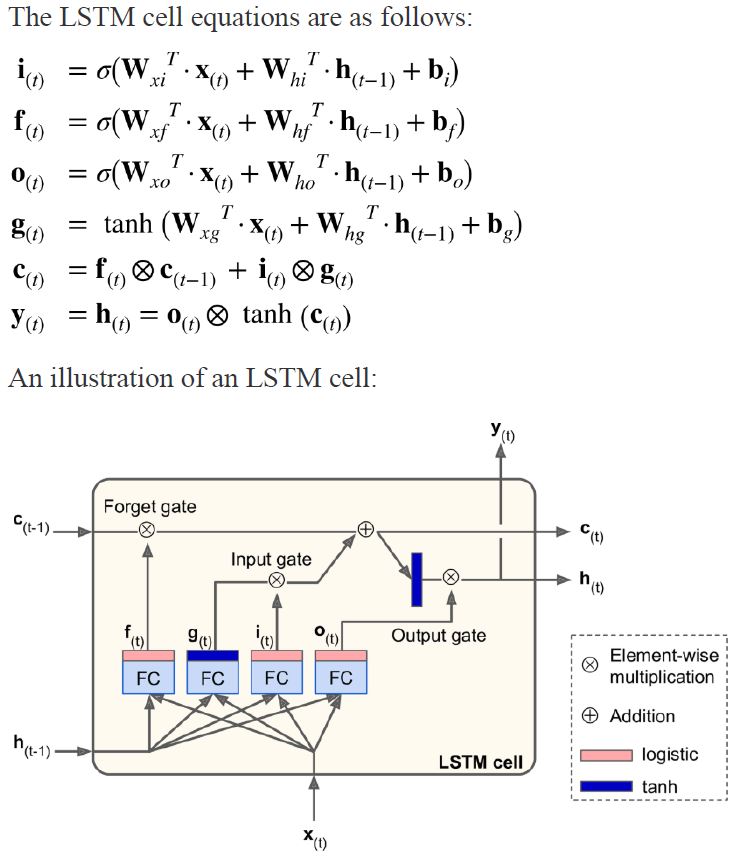
# Question 1

# Question 2

Name two advantages of GRU over LSTM.

Compared to the LSTM, the GRU can usually reach similar performance while having less trainable parameters & not requiring a memory unit. Making it lighter and easier to train. [Chung et al., 2014] Moreover, GRU being lighter and simpler makes it faster and easier to modify.

# Question 3

How many trainable parameters there are in an LSTM cell that uses a [200X1] size current state vector?

Noting that the size of h\y vectors is equal to the size of the state vector, namely [200X1], and assuming that the X vector has the size of [mX1], there are four weight matrices, each of the size of [(200+m)X200], and for each of them there is a bias vector with the size of [200X1]. In total, there are 4X200X(201+m) trainable parameters.

# Question 4

## Code

## Readme

To operate the code just CONNECT colab to a remote GPU and press Ctrl+F9 to run all models in a sequence. The saved weights training and testing accuracy results is running under the evaluation section for each model.

## Results

The Lenet-5 network has been implemented 4 times. Different regularization techniques were added in each implementation including:

1. Dropout
2. Weight Decay
3. Batch Normalization

All implementation used Adam optimization technique, Log loss, 100 example batch size and trained for 15 epochs.

The implemented networks are described below, together with their convergence (accuracy) and loss graphs for both the training and testing sets.

### Clean Lenet-5

The Lenet-5 is constructed with an input layer, 2 convolution layers and 3 fully connected layers. The network accuracy and loss are presented in the following graphs:

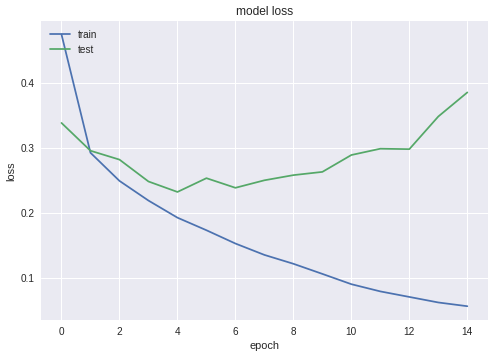
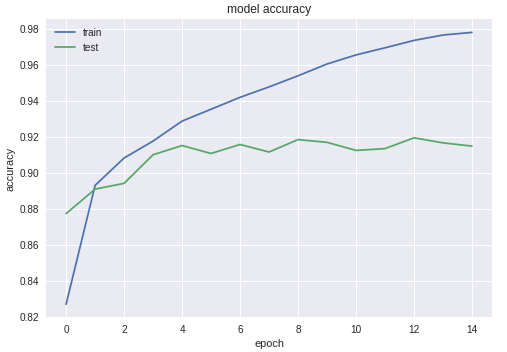


Figure 1 - Accuracy (left) and loss (right) graphs a clean Lenet-5 model

From the graphs it appears that overfitting happened in the training process, since the network got better and better over the training set while the testing set results got worse (this appears more dominantly over the loss graph). Therefore, it can be assumed that this architecture achieved its best results.

### Lenet-5 with dropout layers

On the clean Lenet-5 structure above, a dropout layer has been added for all the network’s hidden layers. Probability of p=0.25 has been chosen for all layers.

**Note -** The KERAS dropout layer implementation only operates on the training set. To measure dropout layer in the testing set, KERAS implement inverted dropout. This means that the weights are scaled up by (1/p) at the training phase and doesn’t transform in the testing phase.

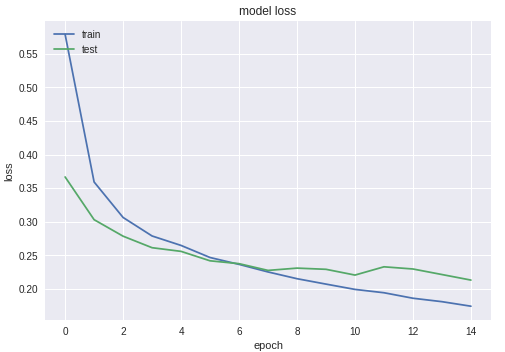
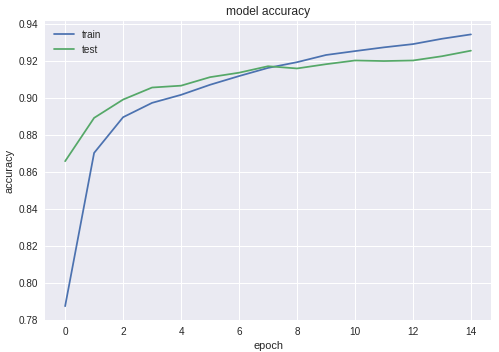


Figure 2 - Accuracy (left) and loss (right) graphs for a Lenet-5 model with dropout

From the graphs it appears that overfitting didn’t occur in the training process with the use of dropout regularization. This is a clear evidence that the use of dropout helped the training process from getting into overfitting. Yet, better result might have been achieved for more training epochs.

### Lenet-5 with weight decay

On the clean Lenet-5 structure above, an weight decay has been implemented for both Kernel and bias weights.

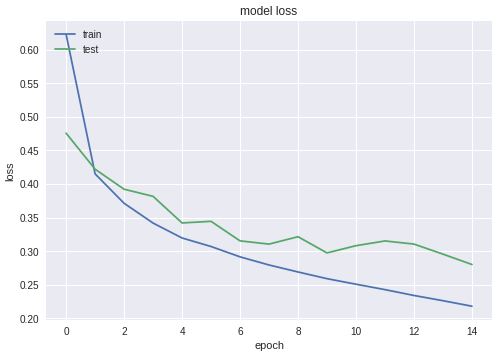
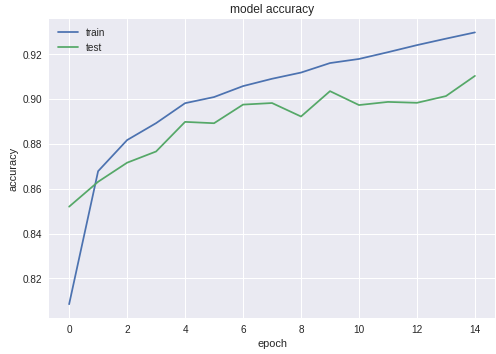


Figure 3 - Accuracy (left) and loss (right) graphs for a Lenet-5 model with weight decay

From the graphs it appears that the training process experienced some sort of resonance. In addition, the training process didn’t achieve overfitting. This is a clear evidence for the addition of weight decay generalization to the learning process. Due to the fact that the process penalizes big weight, the weight growth and penalty changes between epochs. Therefore, the weight decay option seems less smooth in comparison to the dropout generalization technique.

### Lenet-5 with batch normalization

On the clean Lenet-5 structure above, on each layer output batch normalization has been applied.

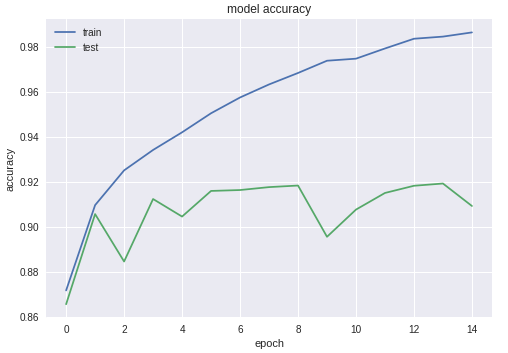
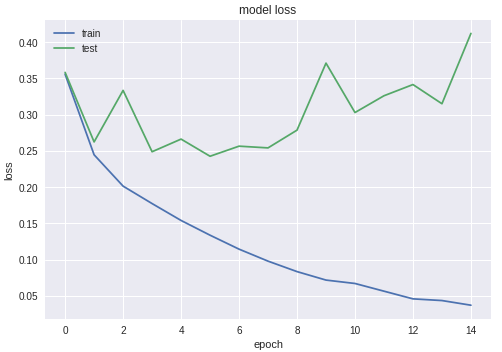
 

Figure 4 - Accuracy (left) and loss (right) graphs for a Lenet-5 model with batch normalization

From the graphs it appears that a larger amount of resonance occurred together with overfitting. The resonation of values over the training set indicates worse regulation in comparison to the weight decay and dropout techniques. Yet, it appears that BN force the values to resonate around some sort of a constant average value. This might indicate about the BN operation as a network stabilizer.

The following table summarize all 8 final accuracies:

|  |  |  |
| --- | --- | --- |
| Model | Training Accuracy | Testing Accuracy |
| Clean | 0.9436 | 0.9153 |
| Dropout | 0.9595 | 0.9255 |
| Weight Decay | 0.9378 | 0.9103 |
| Batch Normalization | 0.9596 | 0.9159 |

Table 1 - Testing set accuracy of the best trained networt for each model

From the table it clearly appears that the network with the dropout layers gained the highest accuracy over all other networks.

It appears that the other networks gained almost the same accuracy over the testing set but different accuracy over the training set. The changes in the training set accuracy indicate about the fact that each regularization technique affects the training process in a different manner. While the similarity in the testing set accuracy indicates that the network architecture affects the accuracy more than the regularization techniques.

In conclusion, the higher accuracy in the dropout testing set, together with the dropout training process graph above, indicates that the dropout technique is a preferred regularization technique over the others. It can be claimed dropout achieves some sort of ensemble learning. Since, dropping different neurons in every epoch actually result different networks. Therefore, better results are achieved from the same architecture.