CS7015: Deep learning for Computer Vision

Name: ***JOSE MOTI***

Roll No: ***CE17B118***

Date: ***16/09/19***

Programming Assignment 2

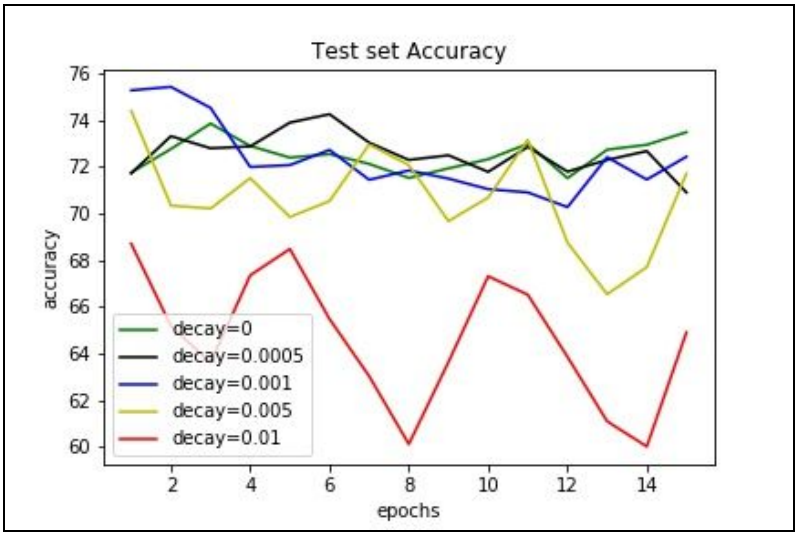
PART A - Introduction

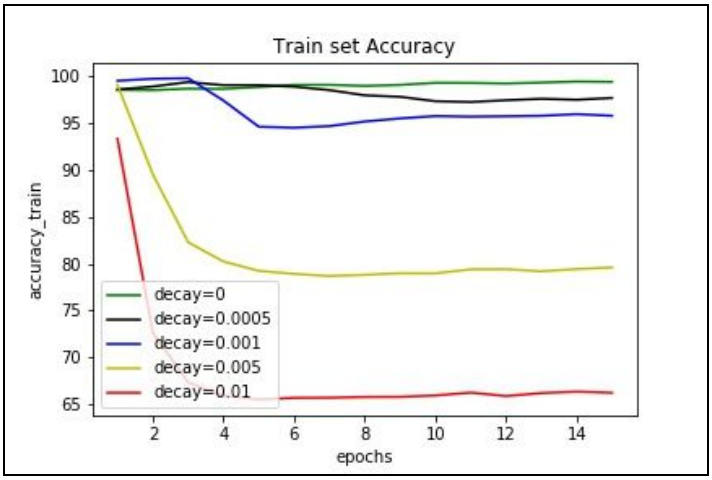
Regularization is provided in order to prevent the overfitting of the model in the training dataset. In this assignment, we would be training models with different regularization parameters and find its effect on training and test datasets. From the previous assignment, I got the best model with a test accuracy of 73.04% and I would be further training that model with different weight decay parameters. Also please note the initial model was trained for 15 epochs using SGD optimizer and keeping the regularization parameter as 0.

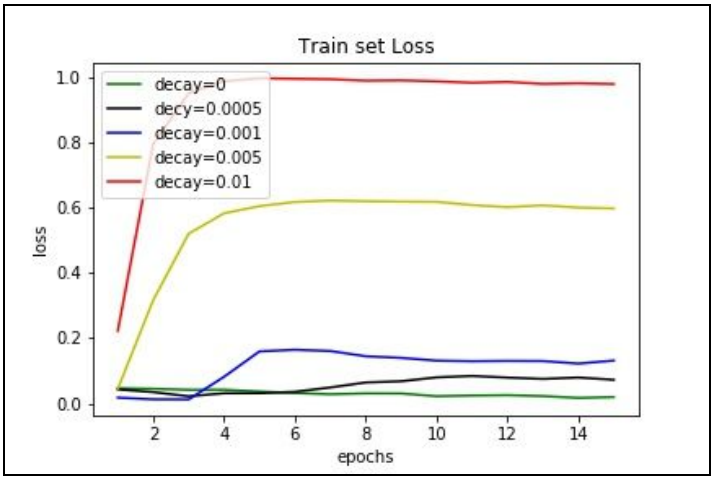
Regularization with Stochastic Gradient Descent

Plots

Model 3.3 as described above from assignment 1 is further trained for 5 different values of weight_decay 0,0.0005, 0.001, 0.005, and 0.01. The variation of test accuracy, train accuracy, train loss after each epoch is plotted. I have used 15 epochs for all these tests. I have used the SGD optimizer here.







Observation table

weight_decay	Max test accuracy	Final train accuracy	Final loss
0	73.8 (epoch 15)	99.3	0.03
0.0005	74.6 (epoch 6)	97.4	0.06
0.001	75.4 (epoch 2)	95.2	0.13
0.005	73.5 (epoch 11)	80.3	0.58
0.01	68.5 (epoch 5)	66.8	0.99

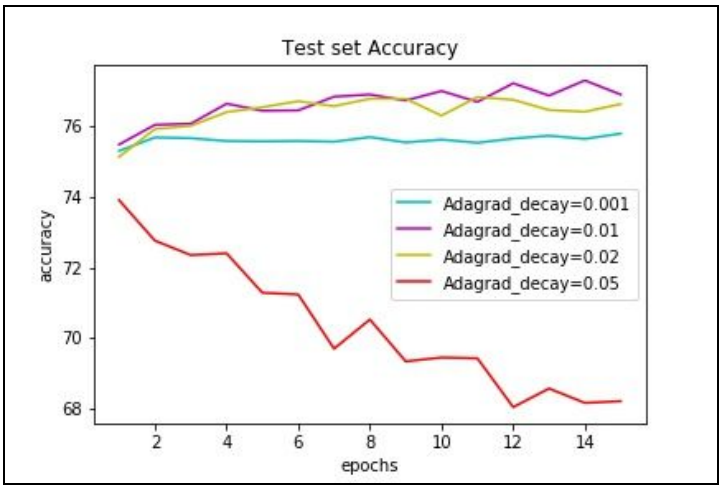
Inferences

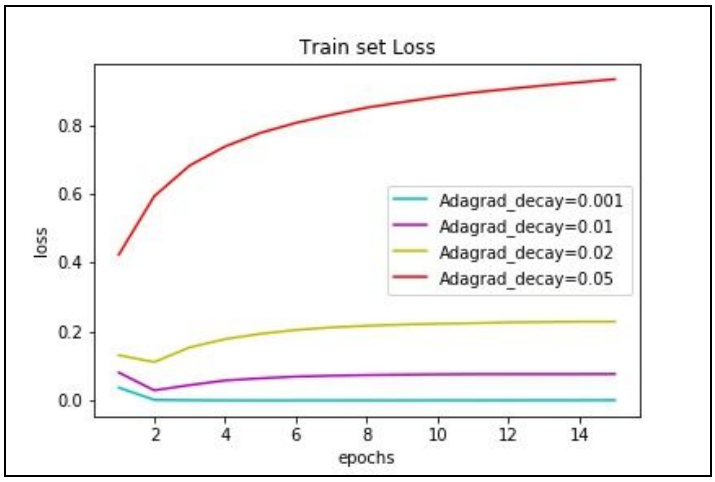
- We are able to see a common trend in train dataset from the table. As the regularization parameter increases the final train accuracy decreases and the final train loss increases.
- Also, the plots are showing a similar shape for all the regularization parameters used. There is a small gap between 1,2,3 because the difference in the regularization parameter is 0.0005. When 0.005 difference is brought we can see a greater decrease in train accuracy/ increase in training loss.
- What regularization does here is the prevention of overfitting in the train dataset. It can be considered similar to decreasing the parameters and weight decay provides the percent of parameters being made 0. Thus higher the weight decay less the parameters available for training and so lower the training accuracy/ higher the training loss.
- When coming to performance in the test dataset we can see clearly no trend is followed here. By using regularization parameter 0, 0.0005, 0.001 we are getting almost similar values in test set. While on the other hand using 0.005 and 0.01 gives us lower test accuracy. From the observation table values 3>2>1>4>5.
- This till for 1, 2, 3 the model is reducing the overfitting to the train set thus producing a slight increase in accuracy of test set with increase in regularization parameter. But a subsequent increase as in case 4, 5 nullifies so many parameters which were important thus causing a major drawback in test set prediction.
- Here we can conclude that 0.001 can be considered as the regularization parameter giving the best test/ train results.

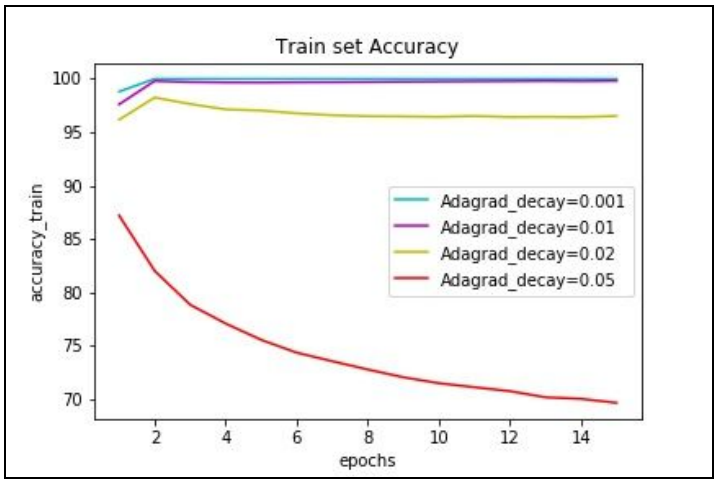
Regularization with Adagrad Optimizer

Plots

One of the problems we faced with SGD was the test results are showing large ups and downs with epochs. In order to decrease this, i am also tuning the hyperparameter with the Adagrad optimizer. The same Model 3.3 is used and it is run for an additional 15 epochs for 4 different weight decay parameters.







Observation table

weight_decay	Max test accuracy	Final train accuracy	Final loss
0	75.6 (epoch 15)	100	0.008
0.0005	78.1 (epoch 14)	99.6	0.06
0.001	76.4 (epoch 15)	97.2	0.21
0.005	73.9 (epoch 1)	69.3	0.98

Inferences

- We are able to see a common trend in train dataset from the table. As the regularization parameter increases the final train accuracy decreases and the final train loss increases similar to SGD.
- When coming to performance in the test dataset we can see clearly that the test accuracy increases for till regularization parameter of 0.01 and then decreases after that. While using 0.05 it is showing a massive decrease in test/ train accuracy because a lot of required parameters are diminished thus worsening the predictions made in test and train set.
- This till for 1, 2, 3 the model is reducing the overfitting to the train set thus producing a slight increase in accuracy of test set with an increase in regularization parameter. But a subsequent increase as in case 4 decreases the test accuracy. 2>3>1>4
- Here we can conclude that 0.01 can be considered as the regularization parameter giving the best test/ train results.
- An observation made here is that as we use different optimizers we are geeting different values of optimal regularization parameter. Also, the regularization parameter varies from model to model and also depends on several other factors in the model.

Results and conclusions

- I was able to understand the effect of different regularization parameters on test and train dataset.
- I was able to find out the best hyperparameter by tuning these hyperparameters.
- Also was able to understand the difference in behaviour of test/ train dataset when two different optimizers are used.

References

- Pytorch documentation
- Matplotlib documentation

Codes and models

The following code files should be run in the order given for this programming assignment. These codes are given in the folder ‘Codes’.

- **definitions.py** - contains all the required functions
- **regloadmodel1.py** - loads the model trained in the previous assignment. This model is taken from the ‘models’ folder.
- **regmain.py** - contains the training part. The weight_decay parameters should be changed for each testing. Also if required the optimizer can also be changed.
- **graphreg.py** - plotting graph of test accuracy vs epochs, train accuracy vs epochs, train loss vs epochs for the model with different regularization parameters trained using SGD optimizer.
- **graphreg2ada.py** - plotting graph of test accuracy vs epochs, train accuracy vs epochs, train loss vs epochs for the model with different regularization parameters trained using Adagrad optimizer.

Each model after each epoch is stored in the folder as follows.

- **models** - the initial model from the previous assignment.
- **models_reg_1** - SGD weight_decay = 0
- **models_reg_2** - SGD weight_decay = 0.001
- **models_reg_3** - SGD weight_decay = 0.01
- **models_reg_4** - SGD weight_decay = 0.005
- **models_reg_5** - SGD weight_decay = 0.0005
- **models_reg_6** - Adagrad weight_decay = 0.001
- **models_reg_7** - Adagrad weight_decay = 0.01
- **models_reg_8** - Adagrad weight_decay = 0.05
- **models_reg_9** - Adagrad weight_decay = 0.02

Since the space is limited i am giving final models of final 3 epochs only in each folder.

Thank you !!!