096260 - Deep learning course

Exercise 2

Submitted by: Danny Semiat 200068807 Sagitt Efrat 200888386

- **Model architecture description, training procedure:**

  Preprocessing:

  We altered the data according to Sergey's alternations in https://github.com/szagoruyko/cifar.torch.git

  - Normalization: normalized the data according to the mean and std of the train data
  - Converted colors data from RGB to YUV

  Our Model's Architecture:

  Inspired by Sergey's Network In Network.

  We have 28 layers in our Neural Network :

```
  (1): nn.SpatialConvolution(3 -> 64, 5x5, 1,1, 2,2)
(2): nn.SpatialBatchNormalization (4D) (64)
(3): nn.ReLU
(4): nn.SpatialConvolution(64 -> 32, 1x1)
(5): nn.SpatialBatchNormalization (4D) (32)
(6): nn.ReLU
(7): nn.SpatialConvolution(32 -> 16, 1x1)
(8): nn.SpatialBatchNormalization (4D) (16)
(9): nn.ReLU
(10): nn.SpatialAveragePooling(3x3, 2,2)
(11): nn.Dropout(0.125000)
(12): nn.SpatialConvolution(16 -> 32, 5x5, 1,1, 2,2)
(13): nn.SpatialBatchNormalization (4D) (32)
(14): nn.ReLU
(15): nn.SpatialConvolution(32 -> 64, 1x1)
(16): nn.SpatialBatchNormalization (4D) (64)
(17): nn.ReLU
(18): nn.SpatialAveragePooling(3x3, 2,2)
(19): nn.Dropout(0.250000)
(20): nn.SpatialConvolution(64 -> 32, 3x3, 1,1, 1,1)
(21): nn.SpatialBatchNormalization (4D) (32)
(22): nn.ReLU
(23): nn.SpatialConvolution(32 -> 10, 1x1)
(24): nn.SpatialBatchNormalization (4D) (10)
(25): nn.ReLU
(26): nn.SpatialAveragePooling(8x8, 1,1)
(27): nn.View(10)
(28): nn.LogSoftMax
```

Data Augmentation:

We used Horizontal Flip on some of the train data.

Training:

Optimization: We used optim.sgd

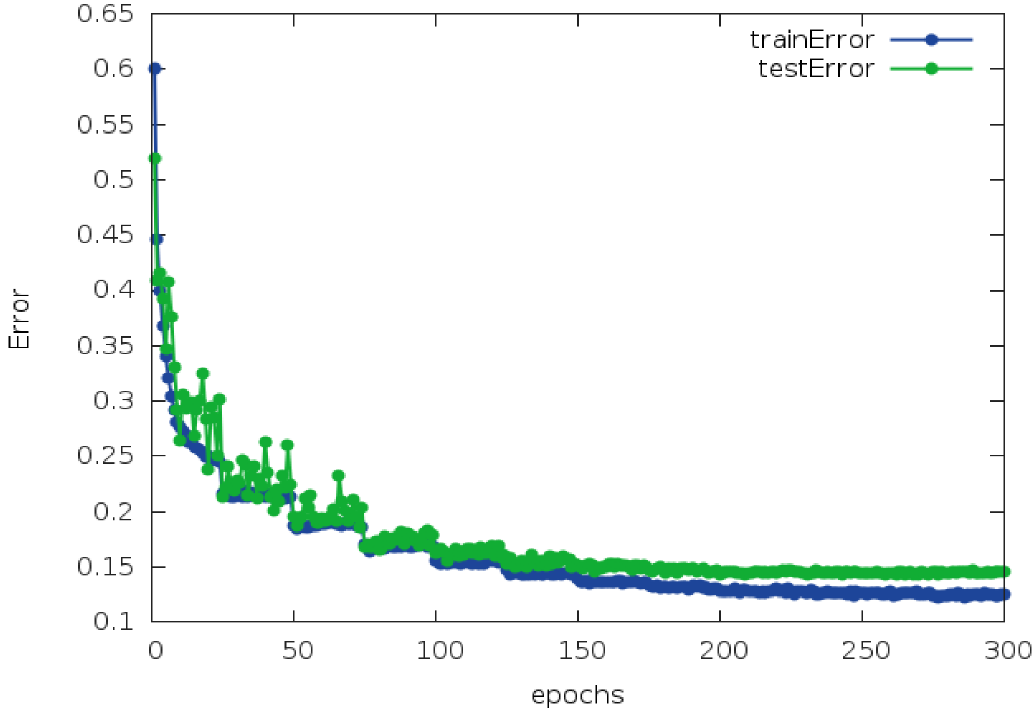
With learning rate that decays every 25 epochs.

Criterion (Loss Function): Cross Entropy.

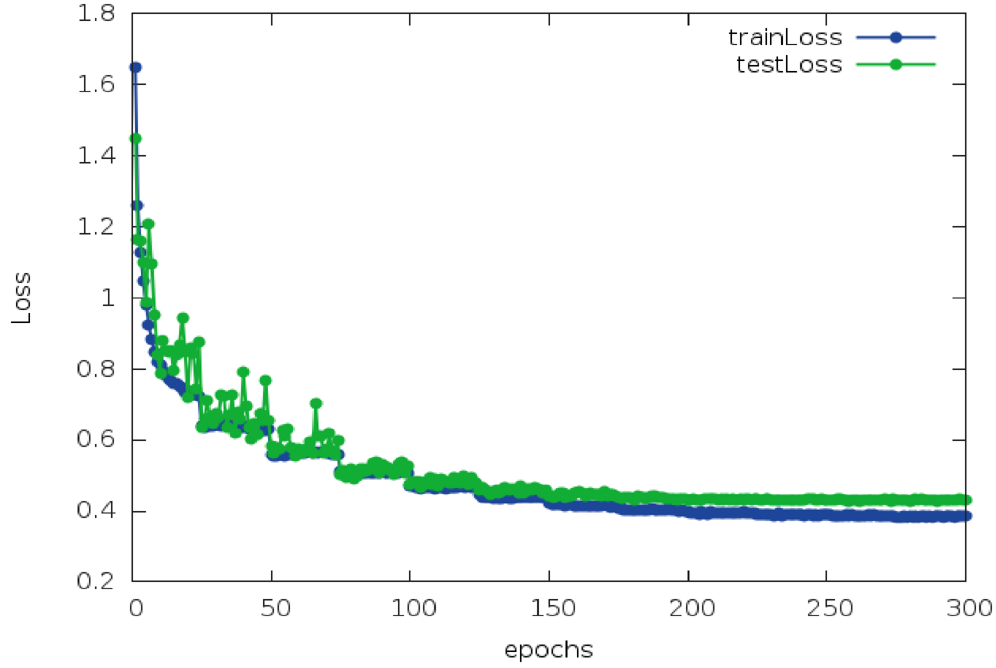
Number of epochs: 300.

- **Two convergence graphs of our model – using sgd optimization and horizontal flip:**

Error: *(Best Epoch: 200, Lowest Error: 0.1428)*

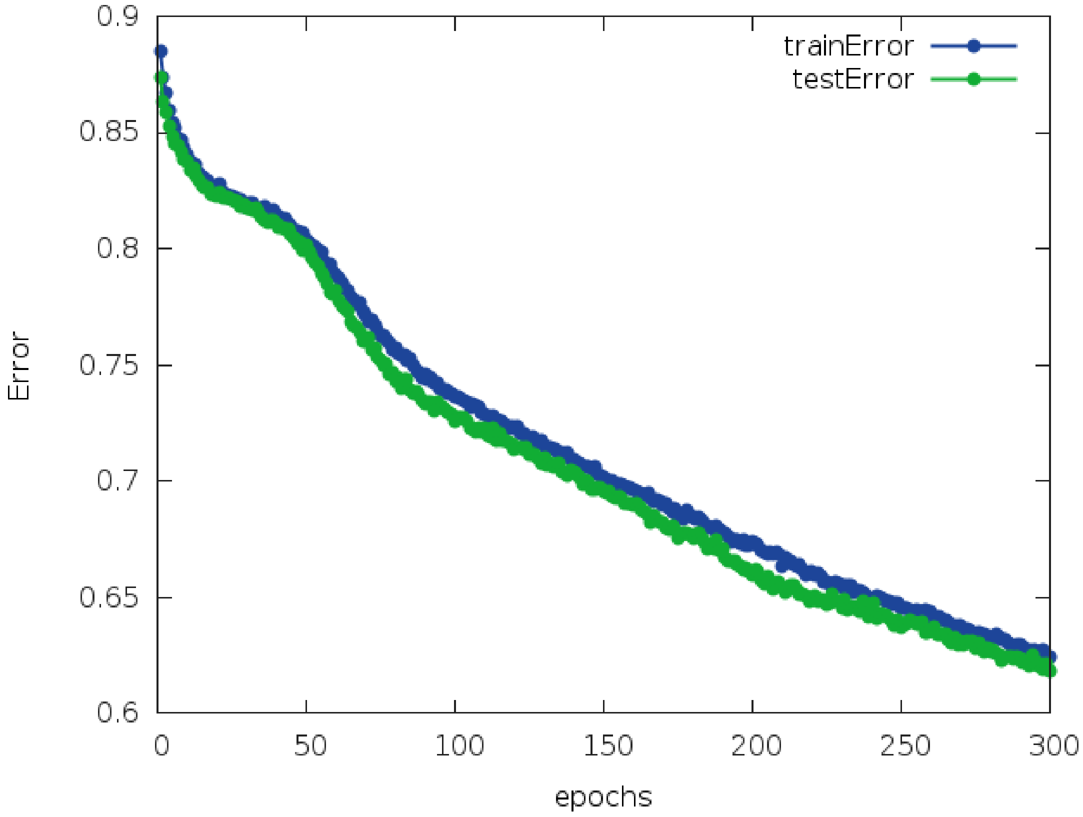


Loss:

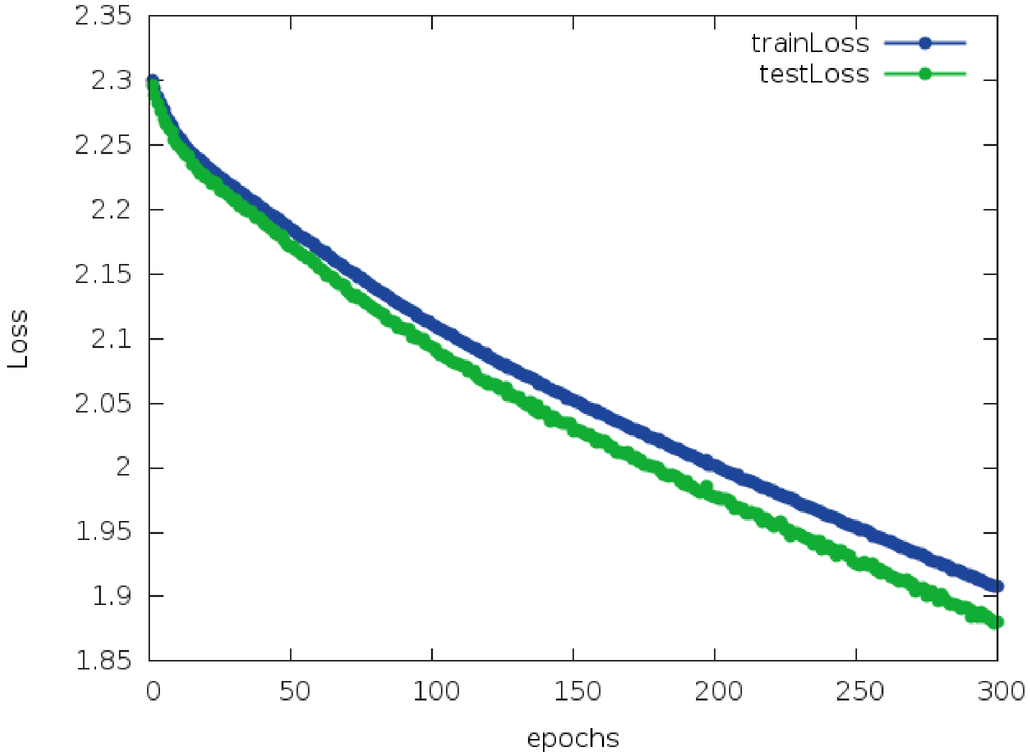


- **Two convergence graphs of our model when using asgd optimization instead of sgd and augmentation of horizontal flip**

Error: *(Best Epoch: 300, Lowest Error: 0.6181)*

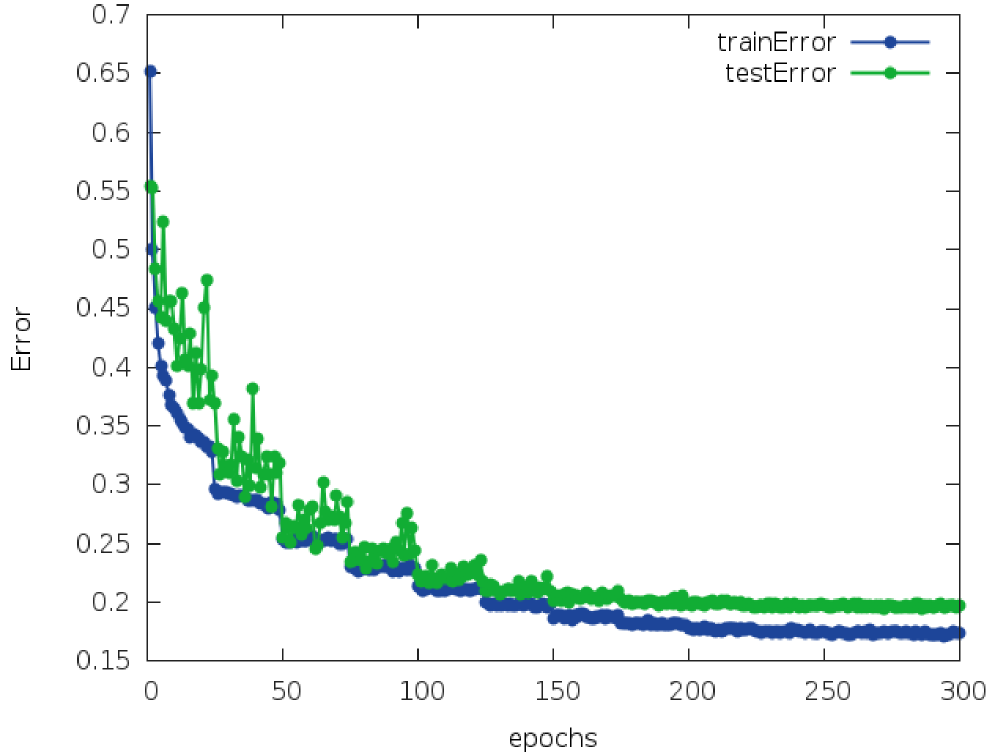


Loss:

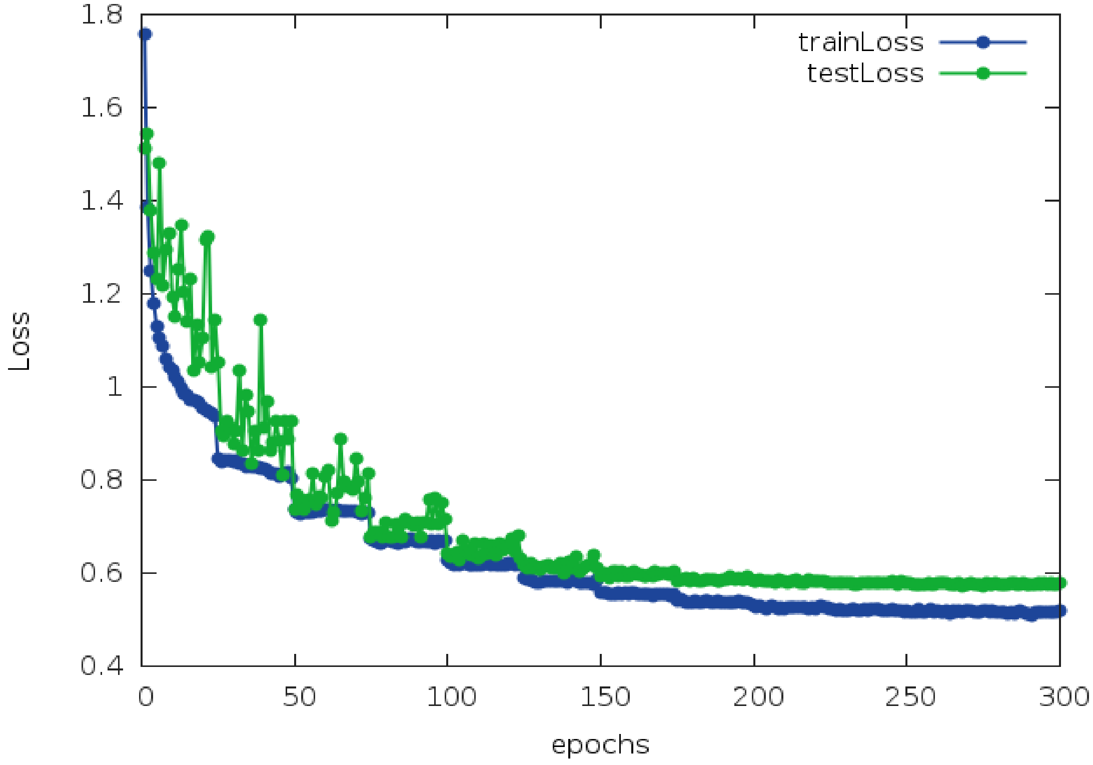


- **Two convergence graphs of our model when using <u>sgd</u> and augmentation <u>vertical flip</u> instead of horizontal flip:**

Error: *(Best Epoch: 272, Lowest Error: 0.1952)*

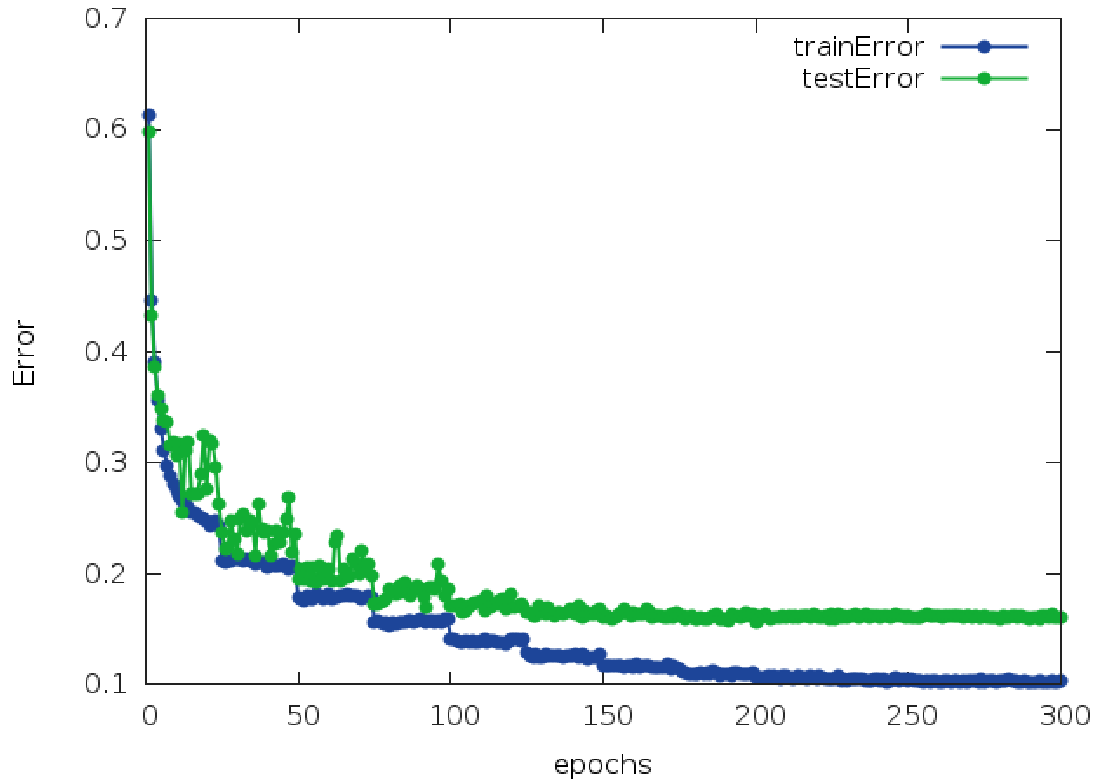


Loss:

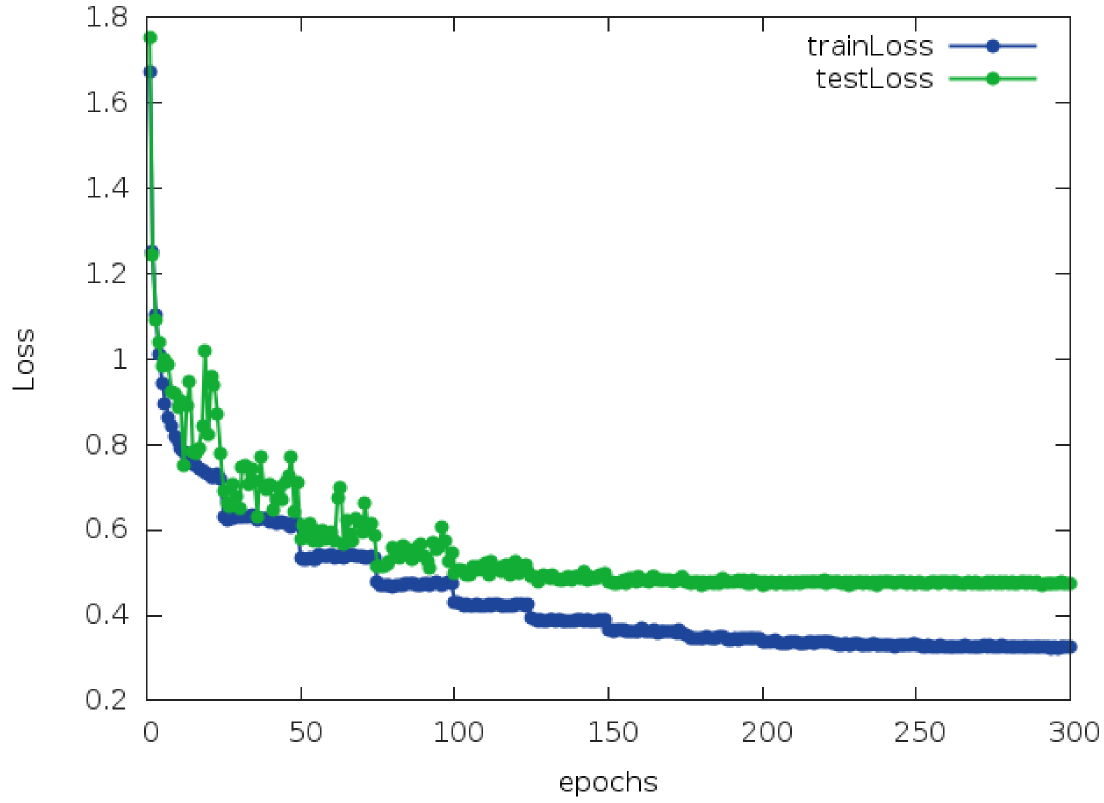


- **Two convergence graphs of our model when using sgd and augmentation of rotate instead of horizontal flip:**

Error: *(Best Epoch: 200, Lowest Error: 0.1555)*



Loss:



- **A short summary of our attempts and conclusions:**
  We tried different kind of variables numbers in the network layers, the number of the layers and also tried different kind of PCA's and then decided on the ones that gave us the lowest loss. The loss function is 'CrossEntropy' after trying also 'MSE' that gave us bigger loss.
  The number of epochs was also decided by trial and error.
  Also, decreasing *learning factor* over time and *normalizing* the data the way we did, improved the prediction accuracy.
- **Our code:**
  On github repository:
  https://github.com/HolyFalafel/DeepLearning.git
  Under the path:
    ex2\submission\
    train_2_sgd.lua - creates the model.
    cifar10_load_model.lua - loads our trained network, normalize the data.
    function ‘test()’ returns the average error on the test set.