# DD2424 DEEP LEARNING IN DATA SCIENCE
# ASSIGNMENT 3

Rithika Harish Kumar(rihk@kth.se)

## AIM

To train and test multi-layer networks on a multi classification problem with batch normalization. The network is trained with stochastic mini batch gradient descent. Momentum is also applied to the training while the aim is to minimize the loss function called cross entropy with a regularization term. The dataset to be used is CIFAR-10.

## RESULTS

The network consists of the below architecture where N is the number of layers except the last layer.

Input – {Fully Connected – Batch Norm **-** ReLU} * N – Fully Connected – Softmax

### GRADIENT CHECK

Gradients are checked by comparing the backpropagated gradients to the numerically computed gradients. The centered difference method is used here while in the previous assignments I had used the faster and less accurate method of finite difference. In the centered difference method, the cost function is computed twice with a positive shift and also with a negative shift. This is more accurate compared to the previously used one because it computes element wise absolute difference scaled by the order of magnitude of individual values. Then it is checked whether every element in the resultant matrix is less than a threshold of $10^{-4}$.

**Gradients for networks without batch normalization and with regularization**

For **2-Layer** network

- W1: 5.70e-04
    - Analytical Grad 6.411e-08
    - Numerical Grad 6.404e-08
    - Diff 7.298e-11
- b1: 3.59e-09
- W2: 3.73e-09
- b2: 8.85e-10

For **3-Layer** network

- W1: 2.01e-03
    - Analytical Grad -9.542e-09
    - Numerical Grad -9.504e-09
    - Diff 3.835e-11

- b1: 7.92e-08
- W2: 1.25e-07
- b2: 1.29e-08
- W3: 2.40e-08
- b3: 3.26e-09

**Gradients for networks with batch normalization and without regularization** (lamda = 0)

For **2-Layer** network

- W1: 1.00e+00
    - Analytical Grad -1.357e-02
    - Numerical Grad 4.006e-03
    - Diff 1.757e-02
- b1: 1.00e+00
    - Analytical Grad -1.388e-18
    - Numerical Grad 3.213e-02
    - Diff 3.213e-02
- W2: 1.70e-08
- b2: 5.68e-09

For **3-Layer** network

- W1: 1.00e+00
    - Analytical Grad -7.359e-04
    - Numerical Grad 1.801e-02
    - Diff 1.874e-02
- b1: 1.00e+00
    - Analytical Grad -5.551e-18
    - Numerical Grad 2.182e-02
    - Diff 2.182e-02
- W2: 1.00e+00
    - Analytical Grad -8.254e-03
    - Numerical Grad 2.522e-02
    - Diff 3.348e-02
- b2: 1.00e+00
    - Analytical Grad 6.982e-18
    - Numerical Grad -1.082e-02
    - Diff 1.082e-02
- W3: 1.85e-08
- b3: 2.61e-09

For **4-Layer** network

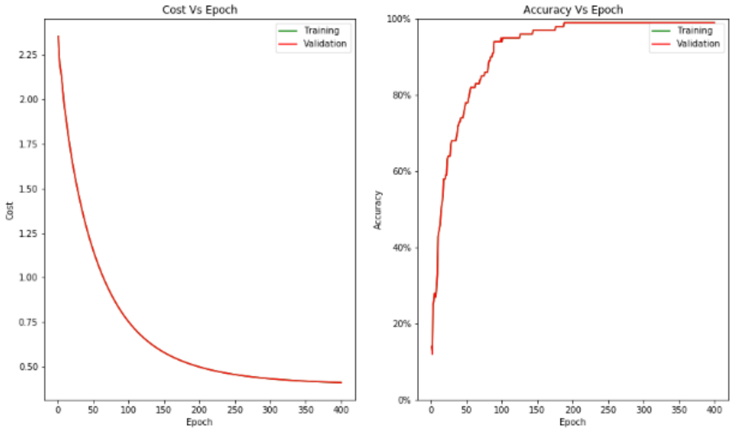
- W1: 1.00e+00
    - Analytical Grad 7.886e-06
    - Numerical Grad -1.110e-02
    - Diff 1.110e-02
- b1: 1.00e+00
    - Analytical Grad -9.592e-18
    - Numerical Grad 5.392e-02
    - Diff 5.392e-02
- W2: 1.00e+00
    - Analytical Grad -4.188e-03
    - Numerical Grad 5.094e-03
    - Diff 9.282e-03
- b2: 1.00e+00
    - Analytical Grad -3.197e-18

  - Numerical Grad 1.291e-02
  - Diff 1.291e-02
- W3: 1.00e+00
  - Analytical Grad 1.080e-04
  - Numerical Grad -6.330e-03
  - Diff 6.438e-03
- b3: 1.00e+00
  - Analytical Grad 2.593e-17
  - Numerical Grad -8.148e-03
  - Diff 8.148e-03
- W4: 1.97e-07
- b4: 7.03e-09

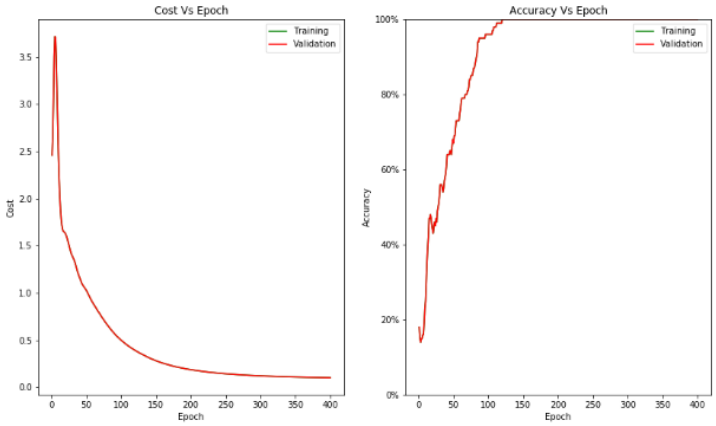
We see that the numerically computed gradients and the gradients computed by backpropagation differ much, sometimes even in sign. So, to test the gradients an empirical test was performed to see that the optimization process was able to overfit (400 epochs) a batch normalized network on a small subset. Below are two, three and four layer networks tested with and without batch normalization.

**2-layer network** (momentum=0.8)

Without Batch Normalization



With Batch Normalization

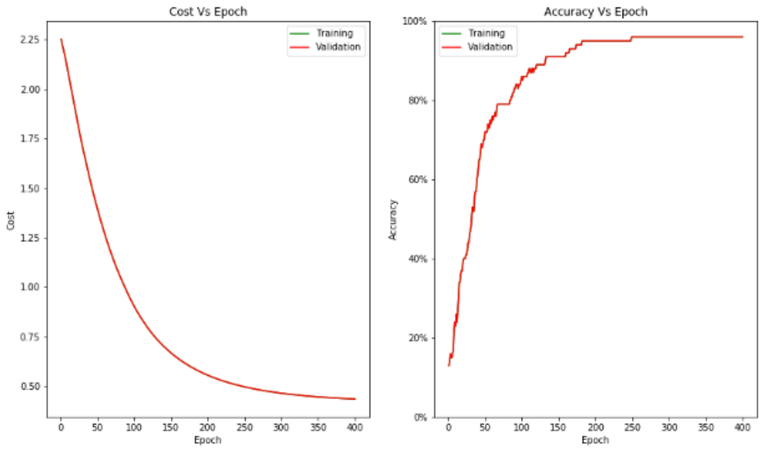


# DD2424 DEEP LEARNING IN DATA SCIENCE
# ASSIGNMENT 3

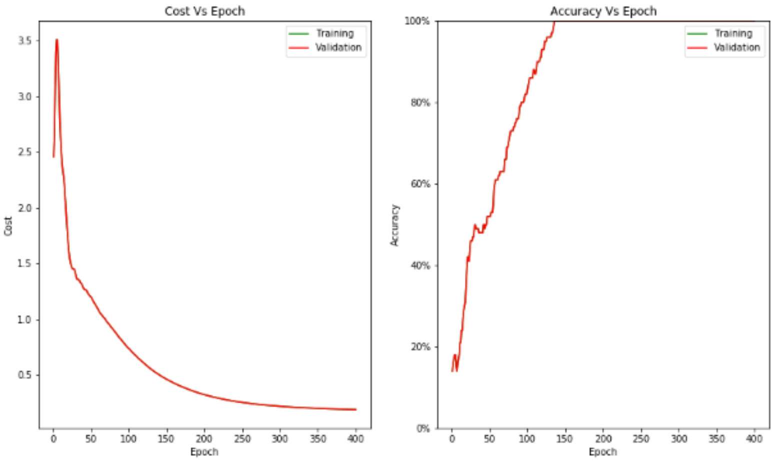
Rithika Harish Kumar(rihk@kth.se)

**3-layer network** (momentum=0.8)

Without Batch Normalization

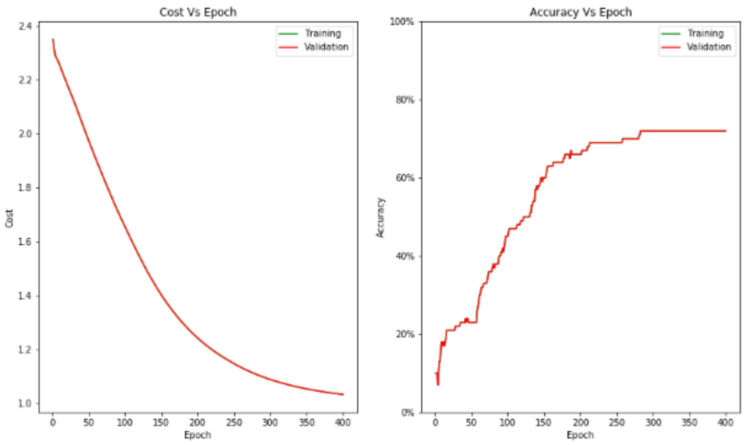


With Batch Normalization

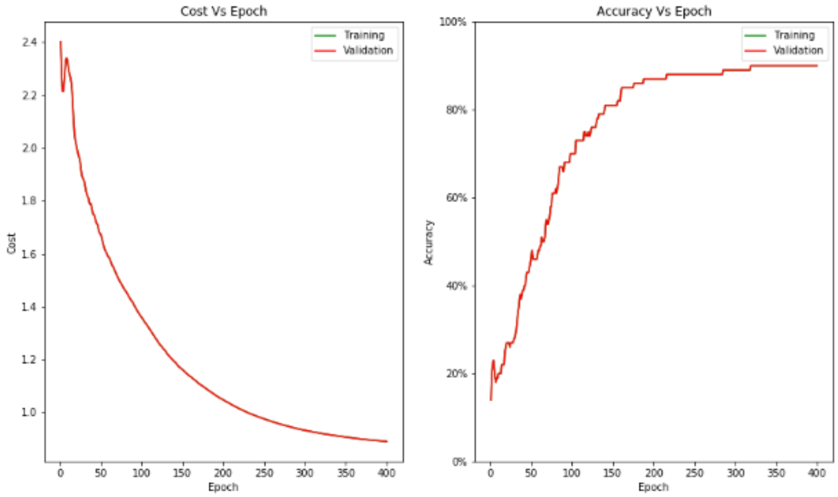


**4-layer network** (momentum=0.8)

Without Batch Normalization



With Batch Normalization



We see that momentumsgd works fine. The 4-layer network without batch normalization has failed to reach high accuracy. Momentum and Batch Normalization speeds up the learning process.

With these results it is evident that the backpropagation works fine.

## BATCH NORMALIZATION

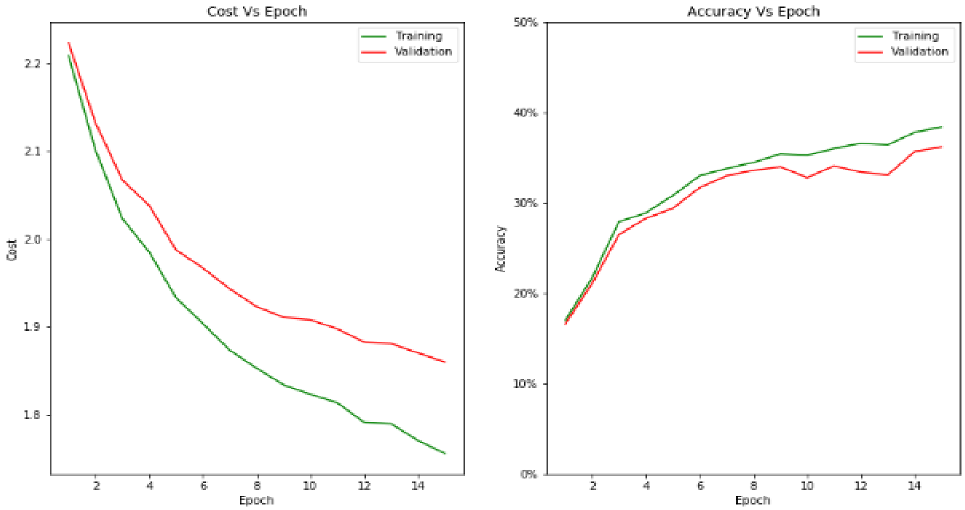
To train deeper networks batch normalization is necessary, otherwise due to gradient propagation and weight initialization their might be problems. In this assignment, He initialization is used for all the networks as it works well with ReLU compared to Xavier initialization.

### 3-layer network with and without batch normalization

Hyper parameters:

Eta = 0.01, decay factor = 0.99, rho = 0.8, lambda = 0, epochs = 15, training set = 10000, validation set = 1000, Hidden layer size = 50-30

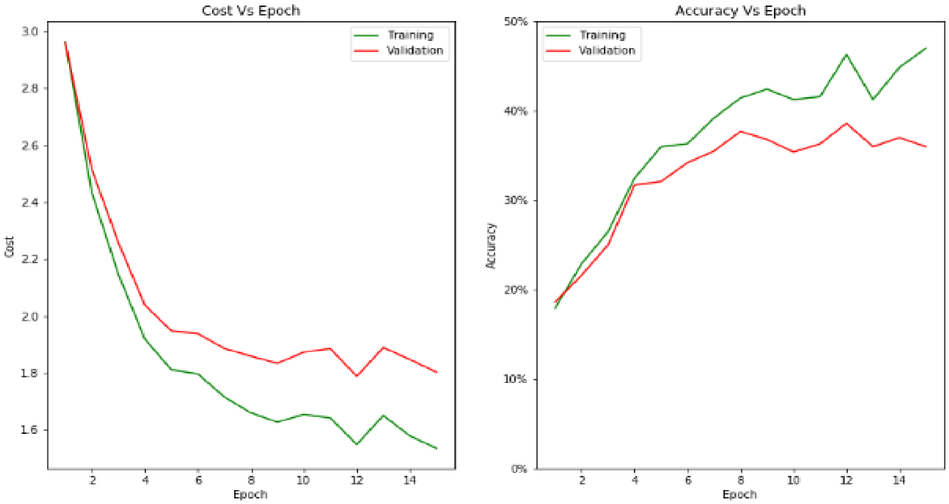
### 3-layer without Batch Normalization



Training accuracy = 0.384, Validation accuracy = 0.362

It is observed that the network is slower in training and also the accuracy is low.

### 3-layer with Batch Normalization



Training accuracy = 0.47, validation accuracy = 0.36

It is seen that the network trains faster and has a good performance compared to the one without batch normalization. Also, the network without batch normalization learns only after a few epochs whereas with batch normalization the cost function steadily decreases as well.

### 3-layer networks with batch normalization: coarse and fine search

To find the best hyper parameters settings to train the network, coarse and fine searches are performed.

**Coarse Search:**

Hyper parameters:

decay factor = 0.99, rho = 0.8, epochs = 10, training set = 5000, validation set = 1000, hidden layers = 3720 – 50 – 30

Lamda = [0.000192, 0.000672, 0.003772, 0.032430, 0.266132]

eta = [0.40573, 0.029202, 0.012721, 0.003215]

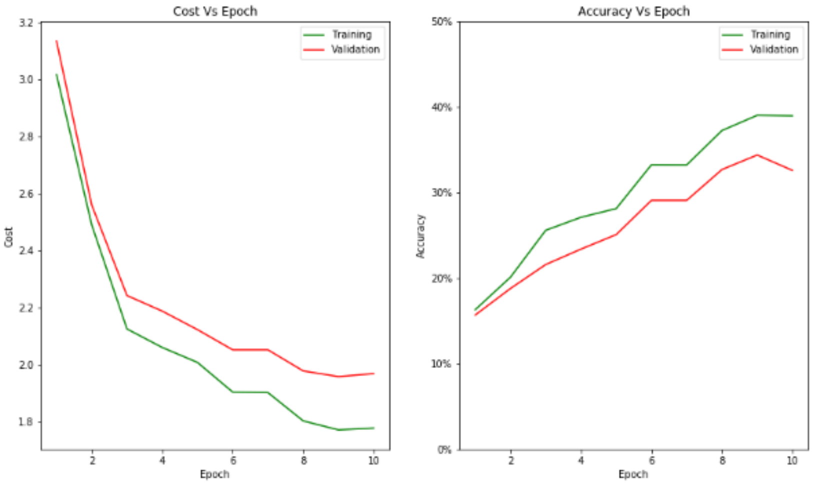
The lamda and eta values are randomly generated between $10^{-4}$ and 0.5.

The best performing 3-layer network on validation set are:

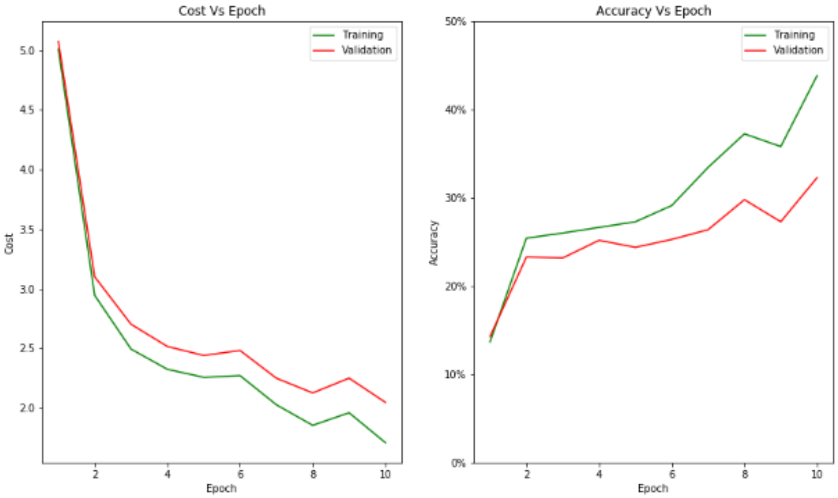
- Eta = 0.012721, lambda = 0.000192, Training accuracy = 0.389800, Validation accuracy = 0.3260

## DD2424 DEEP LEARNING IN DATA SCIENCE
## ASSIGNMENT 3

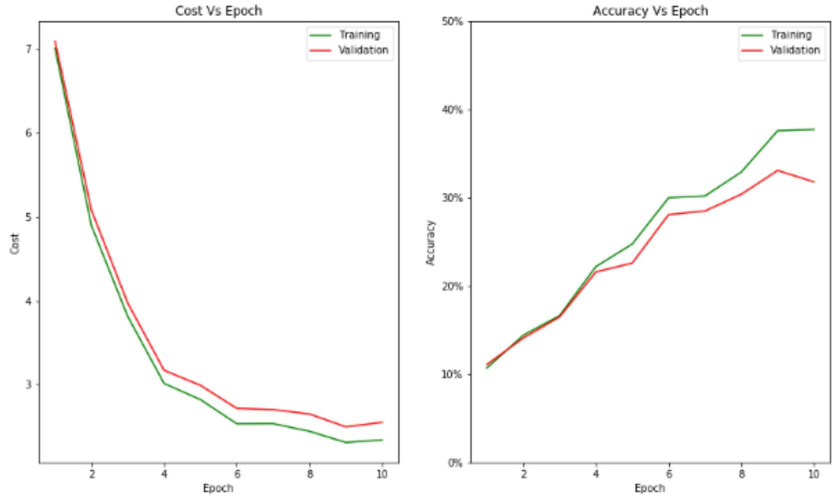
Rithika Harish Kumar(rihk@kth.se)



- Eta = 0.029202, lambda = 0.000672, Training accuracy = 0.438600, Validation accuracy = 0.3230



- Eta = 0.012721, lambda = 0.003772, Training accuracy = 0.377400, Validation accuracy = 0.3180



**Fine Search:**

Hyper parameters:

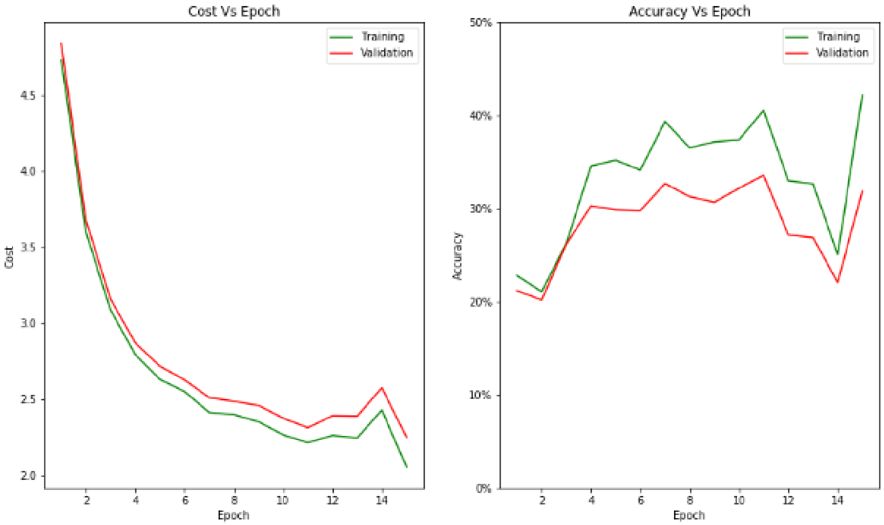
decay factor = 0.99, rho = 0.8, epochs = 15, training set = 10000, validation set = 1000, hidden layers = 3720 – 50 – 30

Lambda = [0.012361, 0.032430, 0.053913]

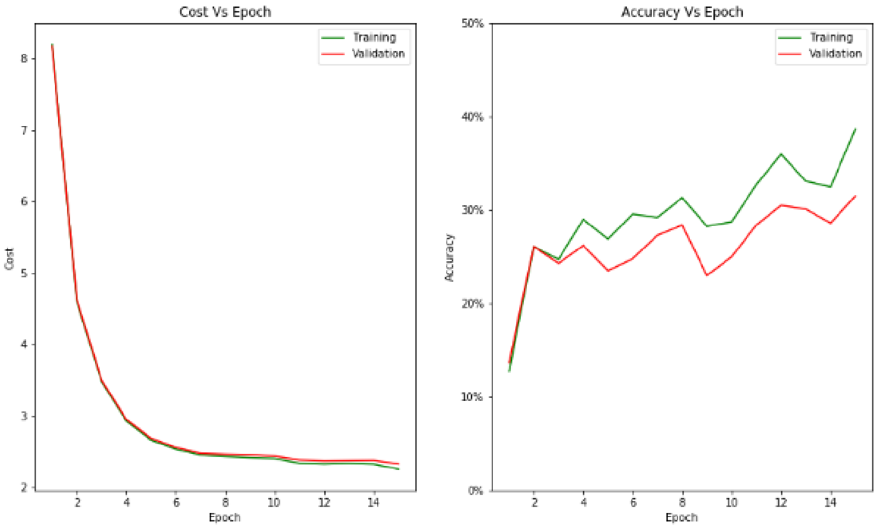
eta = [0.043773, 0.029202, 0.016231, 0.001051]

The best performing 3-layer network on validation set are:

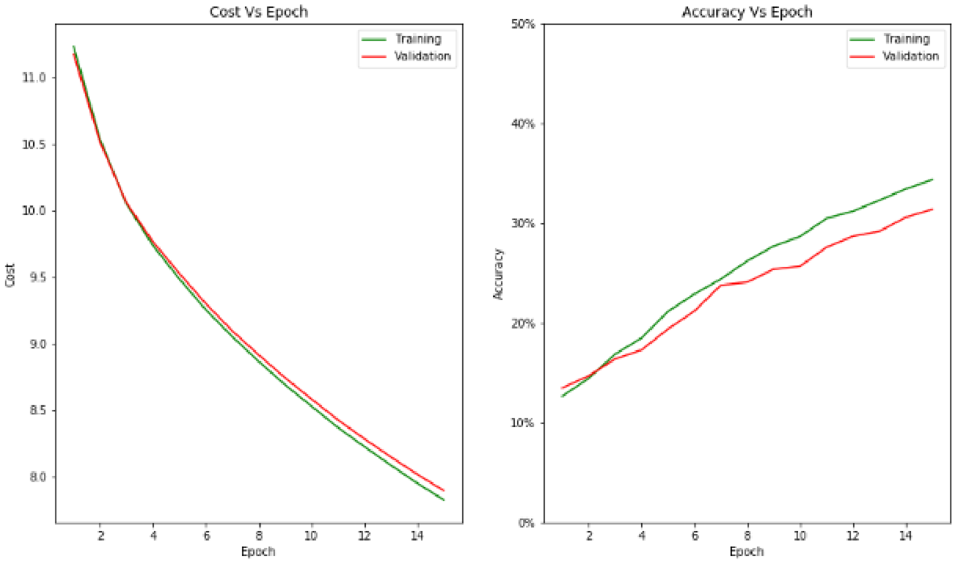
- Eta = 0.043773, lambda = 0.012361, Training accuracy = 0.422000, Validation accuracy = 0.3190



- Eta = 0.029202, lambda = 0.053913, Training accuracy = 0.386500, Validation accuracy = 0.3150



- Eta = 0.001051, lambda = 0.053913, Training accuracy = 0.343700, Validation accuracy = 0.3140

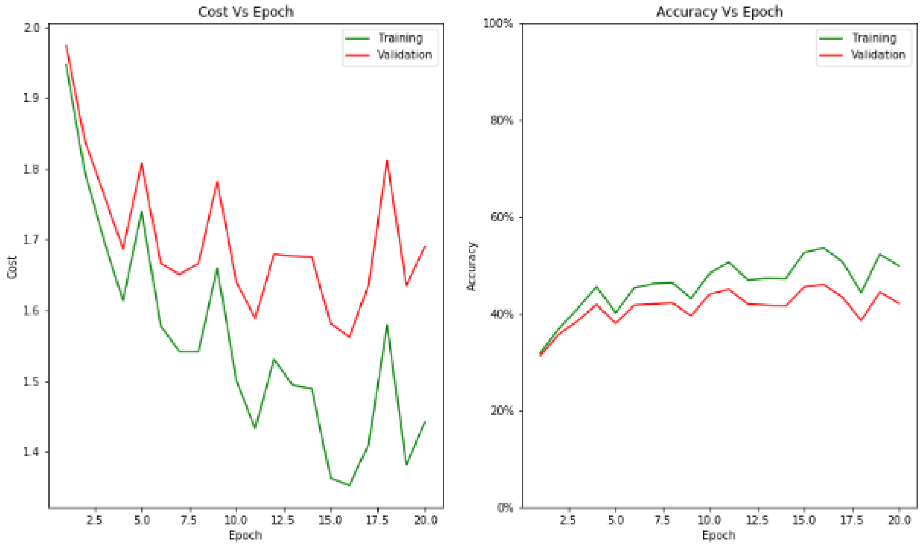


**Best performing network**:

Hyper parameters:

- Eta = 0.012721
- Eta decay = 0.99
- lambda = 0.000192
- epochs = 20
- momentum = 0.8
- Training set = 50000, Validation set = 10000, Test set = 10000

Training Accuracy 0.498625, **Testing Accuracy 0.4264**



Thus, the **test accuracy** achieved was **42.64%**

# DD2424 DEEP LEARNING IN DATA SCIENCE
# ASSIGNMENT 3

Rithika Harish Kumar(rihk@kth.se)

## COMPARING DIFFERENT LEARNING RATES WITH AND WITHOUT BATCH NORMALIZATION IN A 2-LAYER NETWORK:

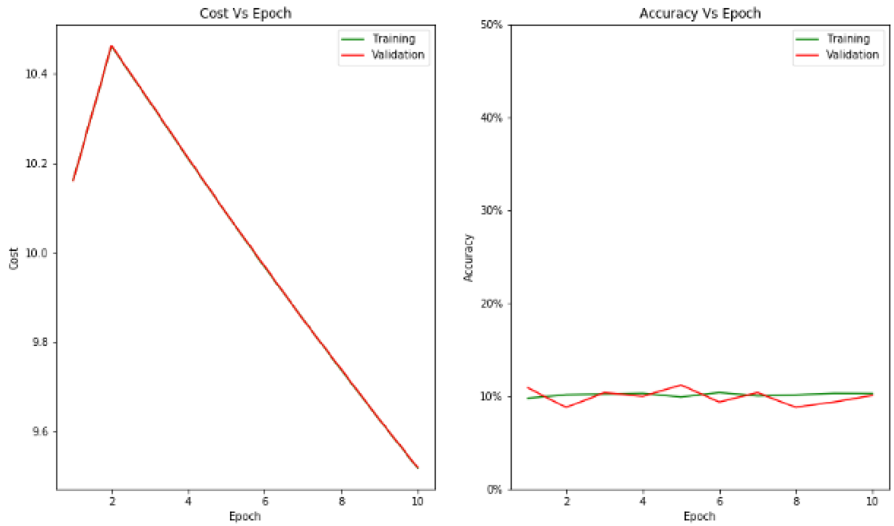
It is observed that the batch normalized network is able to reach higher performances in lesser iterations. Adding batch normalization could increase the learning ability i.e. the gradients descent faster. It also increases the stability i.e. overfitting occurs later.

Hyper parameters:

decay factor = 0.99, rho = 0.8, Lambda = 0.0001, epochs = 10, training set = 10000, validation set = 1000, hidden layers = 3720 – 50 – 10

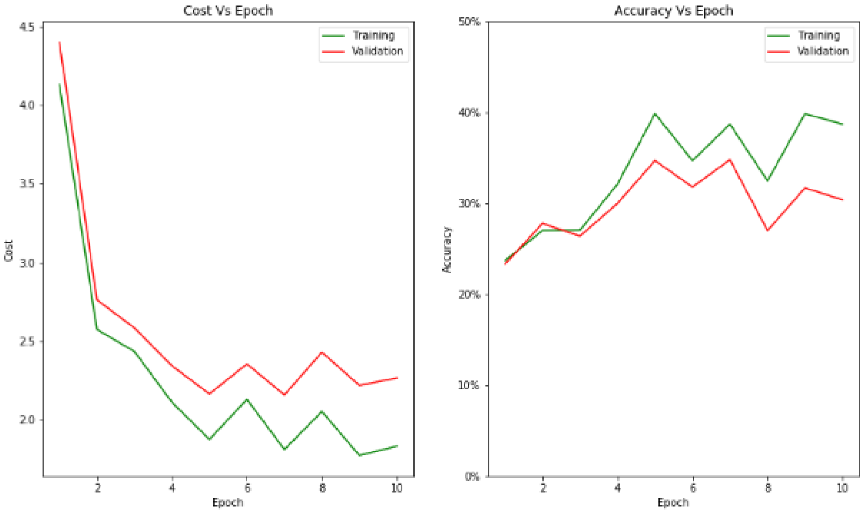
- **eta = 0.40573**

Without Batch Normalization



Training accuracy = 0.103, Validation accuracy = 0.101

With Batch Normalization



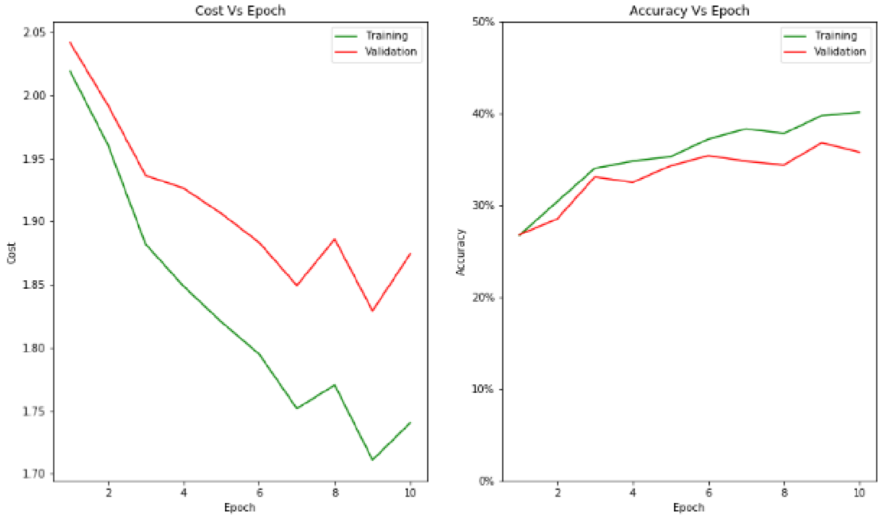
# DD2424 DEEP LEARNING IN DATA SCIENCE
# ASSIGNMENT 3

Rithika Harish Kumar(rihk@kth.se)

Training accuracy = 0.3872, Validation accuracy = 0.304

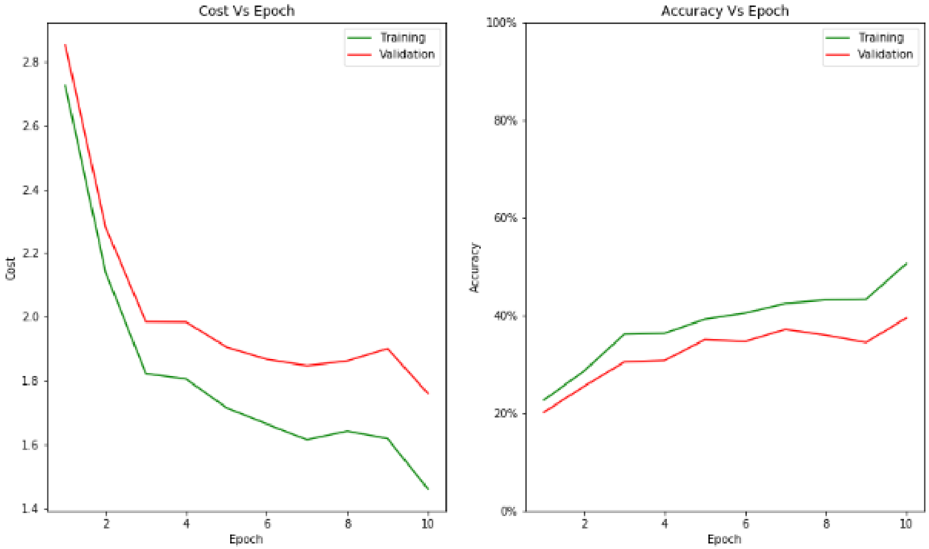
- **eta = 0.029202**

Without Batch Normalization



Training accuracy = 0.4014, Validation accuracy = 0.358

With Batch Normalization



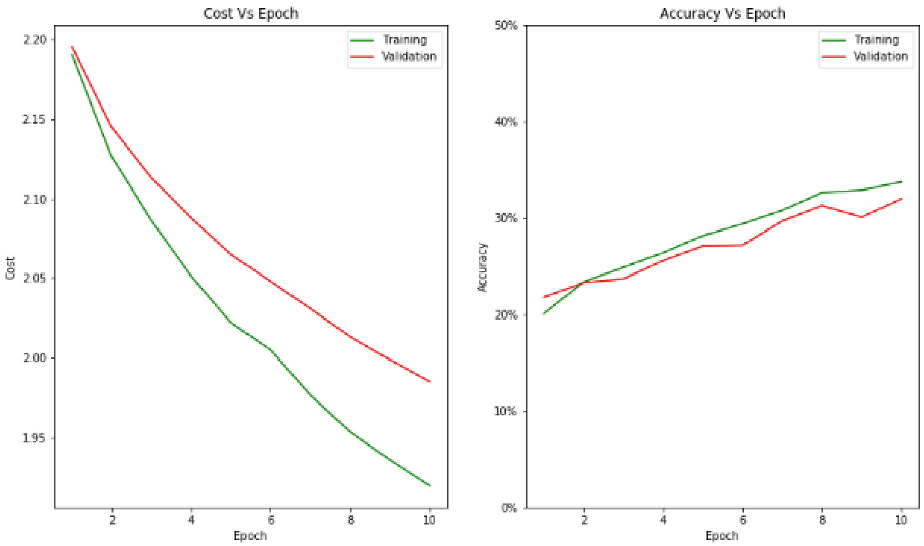
Training accuracy = 0.5063, Validation accuracy = 0.395

- **eta = 0.003215**

Without Batch Normalization

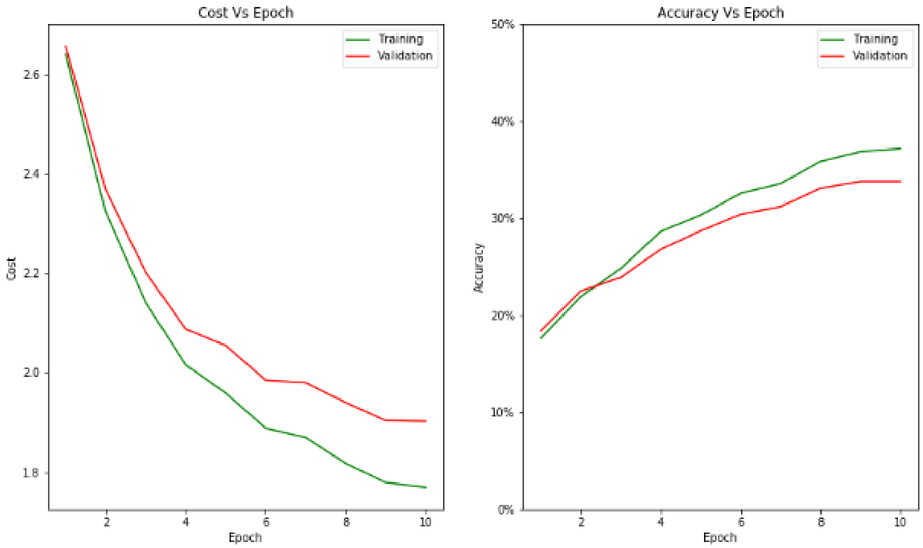
# DD2424 DEEP LEARNING IN DATA SCIENCE
# ASSIGNMENT 3

Rithika Harish Kumar(rihk@kth.se)



Training accuracy = 0.3376, Validation accuracy = 0.32

With Batch Normalization



Training accuracy = 0.3718, Validation accuracy = 0.338