**Task1**

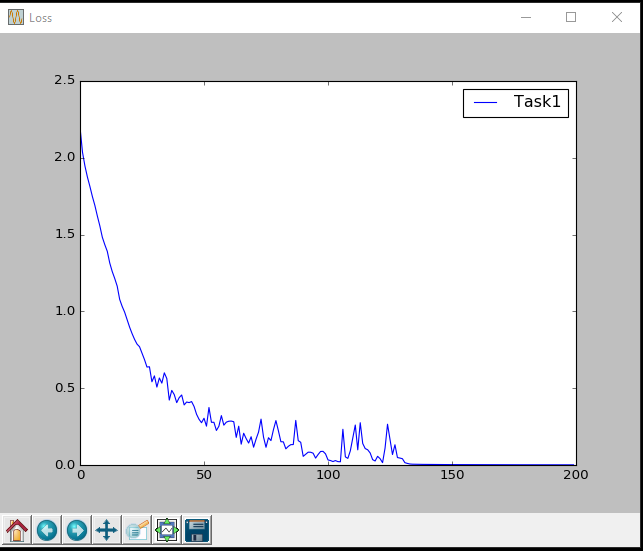
Learning rate = alpha = 0.005

Regularization factor = lambda = 0.0

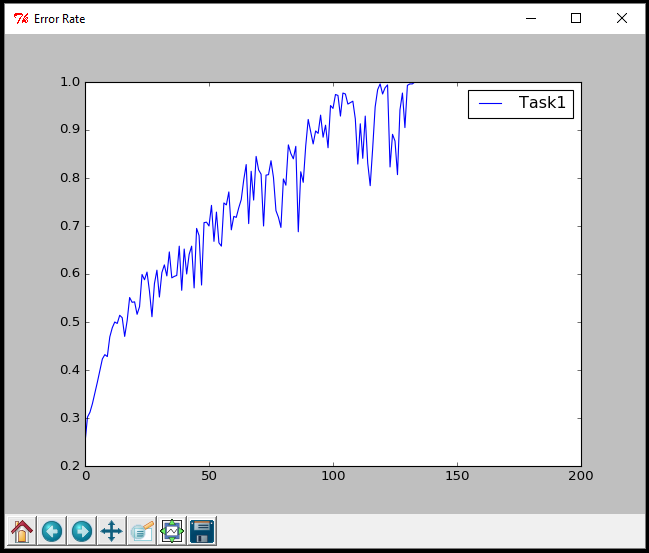
No of Nodes in hidden layer = 100

Activation Function = “relu”

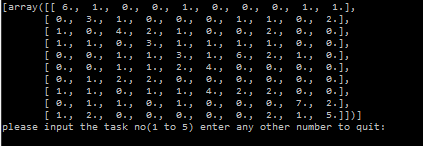
**Loss:**



**Error Rate:**



**Confusion Matrix:**



**We can see because of overfitting error rate is increasing and loss goes to zero.**

**Number of correct output: 37**

**Task2**

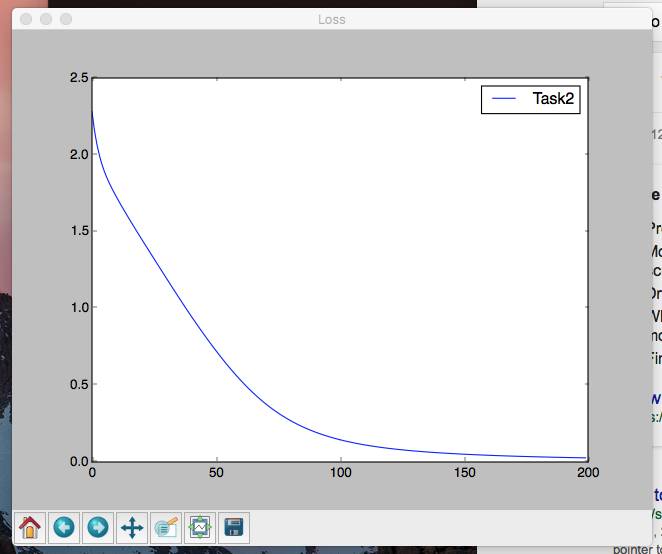
Learning rate = alpha = 0.005

Regularization factor = lambda = 0.0

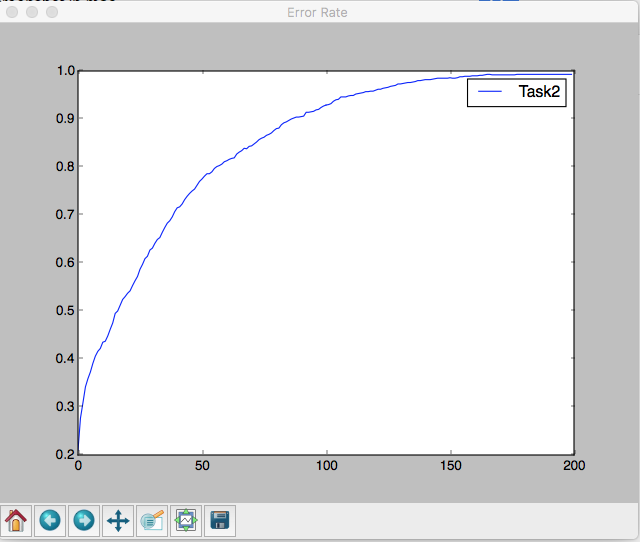
No of Nodes in hidden layer = 100

Activation Function = “sigmoid”

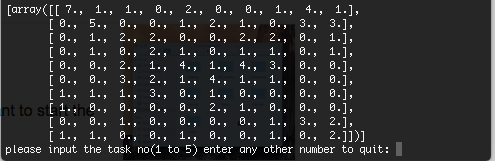
**Loss:**



**Error Rate:**



**Confusion Matrix:**



**Number of correct output: 29**

**Task3**

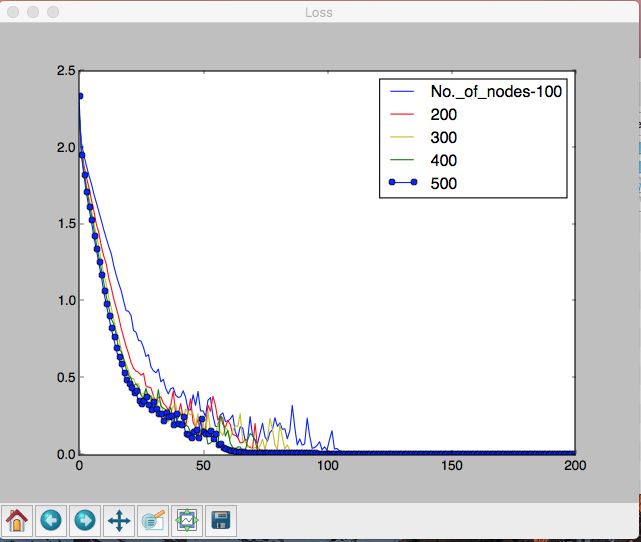
Learning rate = alpha = 0.005

Regularization factor = lambda = 0.0

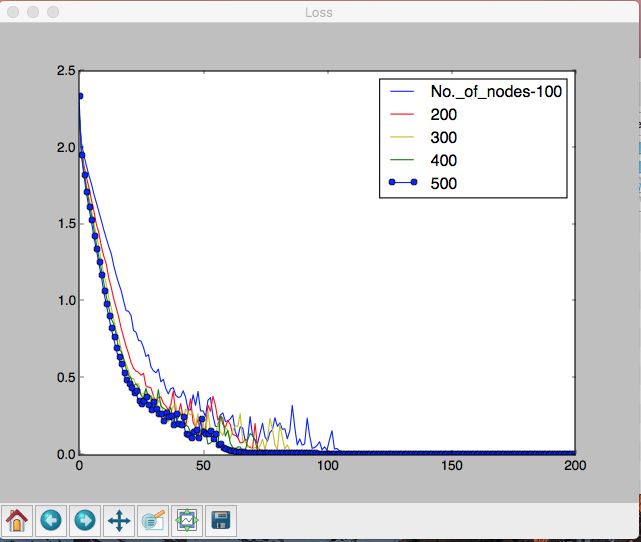
No of Nodes in hidden layer = 100,200,300,400,500

Activation Function = “relu”

**Loss:**



**Error Rate:**



**Confusion Matrix:**

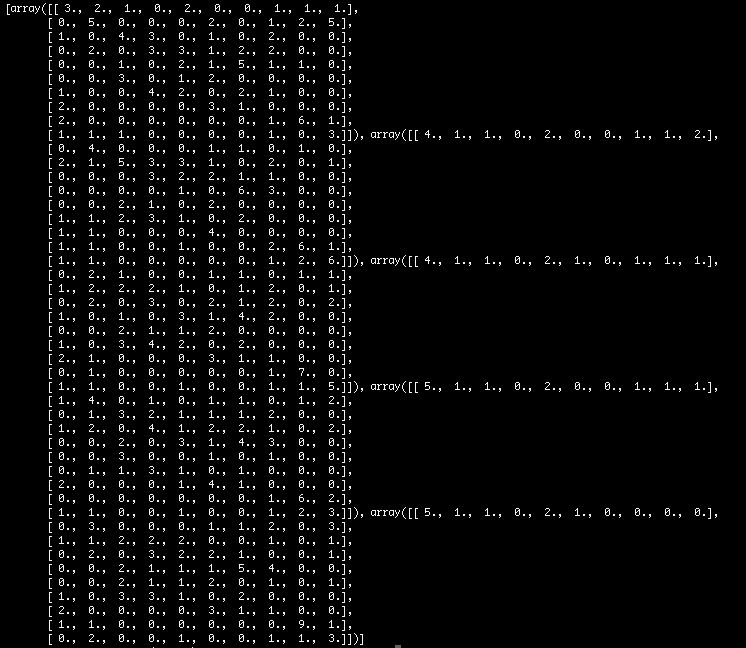
**No of Nodes in hidden layer = 100 :: No of correct output: 30**

**No of Nodes in hidden layer = 200 :: No of correct output: 33**

**No of Nodes in hidden layer = 300 :: No of correct output: 31**

**No of Nodes in hidden layer = 400 :: No of correct output: 30**

**No of Nodes in hidden layer = 500 :: No of correct output: 31**



**Task4**

Learning rate = alpha = 0.005

Regularization factor = lambda = 0.1,0.2,0.3,0.4,0.5

No of Nodes in hidden layer = 500

Activation Function = “relu”

**Loss:**

**Error Rate:**

**Confusion Matrix:**

**Regularization factor:0.1 :: No of correct output:**

**Regularization factor:0.2 :: No of correct output:**

**Regularization factor:0.3 :: No of correct output:**

**Regularization factor:0.4 :: No of correct output:**

**Regularization factor:0.5 :: No of correct output:**

**Task5**

Learning rate = alpha = 0.008

Regularization factor = lambda = 0.09

No of Nodes in hidden layer = 200

Activation Function = “sigmoid”

**Loss:**

**Error Rate:**

**Confusion Matrix:**

**No of correct output :**