CSCI 567 – MACHINE LEARNING
ASSIGNMENT 4
AJAY KUMAR LOGANATHAN RAVICHANDRAN
USC ID : 1669468906

1) **Boosting**

Least Square Loss $(L(y_i, \hat{y}_i)) = (y_i - \hat{y}_i)^2$

A)

To find: *Gradient* (g_i)

Gradient $(g_i) = \frac{\partial L(y_i, \hat{y}_i)}{\partial \hat{y}_i}$

$= 2\,(y_i - \hat{y}_i)\,(-1)$

$= 2(\hat{y}_i - y_i)$

B)

Given :

Step size $= \gamma$

$$h^* = arg\ min_{h \in H}\left(min_{\gamma \in R} \sum_{i=1}^{n} (-g_i - \gamma h(x_i))^2 \right)$$

To prove:

γ can be computed in the closed form

Substituting for g_i,

$$h^* = arg\ min_{h \in H}\left(min_{\gamma \in R} \sum_{i=1}^{n} (-2(\hat{y}_i - y_i) - \gamma h(x_i))^2 \right)$$

Let $\gamma^* = min_{\gamma \in R} \sum_{i=1}^{n} (-2(\hat{y}_i - y_i) - \gamma h(x_i))^2$

In order to find the optimal value of γ^*, differentiate it with respect to γ and equate it to zero.

$$\frac{\partial \gamma^*}{\partial \gamma} = \sum_{i=1}^{n} (2\,((2(y_i - \hat{y}_i) - \gamma h(x_i))\ (-h(x_i)))) = 0$$

$$\Rightarrow \sum_{i=1}^{n} ((4\,(y_i - \hat{y}_i) - 2\,\gamma h(x_i))(-h(x_i))) = 0$$

$$\Rightarrow \sum_{i=1}^{n} ((2\gamma h((x_i))^2 - 4\,(y_i - \hat{y}_i)(h(x_i)) = 0$$

$$\Rightarrow \sum_{i=1}^{n} ((2\gamma h((x_i))^2 = \sum_{i=1}^{n} 4\ (y_i - \hat{y}_i)(h(x_i))$$

$$\Rightarrow \gamma = \frac{\sum_{i=1}^{n} 4\,(y_i - \hat{y}_i)(h(x_i))}{\sum_{i=1}^{n}((2h((x_i))^2}$$

C)

Given:

$$Update\ step: \ \hat{y}_i \leftarrow \widehat{y_i} + \alpha h^*(x_i)$$

$$\alpha^* = arg\ min_{\alpha \in R} \sum_{i=1}^{n} L(y_i + \alpha h^*(x_i))$$

$$\Rightarrow arg\ min_{\alpha \in R} \sum_{i=1}^{n} (y_i - (\hat{y}_i + \alpha h^*(x_i)))^2$$

Differentiating it with respect to α,

$$\frac{\partial \alpha^*}{\partial \alpha} = \sum_{i=1}^{n} 2(y_i - (\hat{y}_i + \alpha h(x_i)))\,(-h^*(x_i)) = 0$$

$$\Rightarrow \sum_{i=1}^{n} 2((h^*(x_i)\left(y_i - \widehat{y_i} - \alpha h^*(x_i)\right) = 0$$

$$\Rightarrow \sum_{i=1}^{n} 2\alpha(h^*(x))^2 = \sum_{i=1}^{n} 2(h^*(x_i))(y_i - \widehat{y_i})$$

$$\Rightarrow \alpha = \frac{\sum_{i=1}^{n} 2(h^*(x_i))(y_i - \widehat{y_i})}{\sum_{i=1}^{n} 2(h^*(x))^2}$$

2)

a)

Given:

The hidden layers have linear activations.

Therefore for the first hidden layer,

Activation function for the j^{th} neuron $(\alpha_j) = \sum_i v_{ji} x_i$

Where $v_{ji} \rightarrow$ *weight of the* j^{th} *neuron w.r.t to input* x_i

Since the given neural network has a single logistic output, considering the sigmoid function, the output y can be shown as:

$$\Rightarrow \sigma(z) = \sigma(\sum_k \sum_j w_k \alpha_j)$$

Where α_j *is the input from the previous layer* (*activation function*)

w_k *is the weight of* k^{th} *neuron in the current layer from the* j^{th} *neuron from the previous layer*

As we know that a sigmoid function $\sigma(z) = \frac{1}{1+exp(-z)}$,

$\Rightarrow \quad y = \frac{1}{1+exp(-\sum_k \sum_i w_k v_{ki} x_i)}$

$\Rightarrow y = \frac{1}{1+exp(-\sum_i w_i' x_i)}$, $where \;\; w_i' = \sum_k w_k v_{ki}$

Since, the hidden layers have a linear function, the above output equation holds true even if there are multiple hidden layers. Thus, the neural network with single logistic output with linear activation in the hidden layers is equivalent to logistic regression.

b)

Given:

Squared loss function : $L(y, \hat{y}) = \frac{1}{2}\sum_{i=1}^{2}(y_j - \hat{y}_j)^2$

Hidden layer $(z_k) = tanh(\sum_{i=1}^{3} w_{ki} x_i) \;\; for \; k = 1, 2, .., 4$

Outputs $(y_j) = \sum_{k=1}^{4} v_{jk} z_k \;\; for \; j = 1, 2$

Solution:

The gradient of w_{ki} gives the back-propagation updates for the estimation of w_{ki}

$\Rightarrow \quad \frac{\partial L}{\partial w_{ki}} = \frac{\partial L}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial z_k} \frac{\partial z_k}{\partial w_{ki}}$

$\frac{\partial L}{\partial \hat{y}_j} = -\sum_{i=1}^{2}(y_j - \hat{y}_j)$

$\frac{\partial \hat{y}_j}{\partial z_k} = v_{jk} \quad ; \quad \frac{\partial z_k}{\partial w_{ki}} = (1 - tanh^2(\sum_{i=1}^{3} w_{ki} x_i)) \, x_i$

Therefore,

$$\frac{\partial L}{\partial w_{ki}} = \sum_{i=1}^{2}(y_j - \hat{y}_j) \, v_{jk} \, (1 - tanh^2(\sum_{i=1}^{3} w_{ki} x_i)) \, x_i$$

The gradient of v_{jk} gives the back-propagation updates for estimation of v_{jk},

$\Rightarrow \quad \frac{\partial L}{\partial v_{jk}} = \frac{\partial L}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial v_{jk}}$

Now, calculating the terms in above equation, we get:

$\frac{\partial L}{\partial \hat{y}_j} = (y_j - \hat{y}_j)$

$\frac{\partial \hat{y}_j}{\partial v_{jk}} = z_k$

Therefore the gradient of v_{jk} can be given by :

$$\frac{\partial L}{\partial v_{jk}} = (y_j - \hat{y}_j)\, z_k$$

PROGRAMMING

D) LINEAR ACTIVATIONS

1)

Score for architecture = [50, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = linear: 0.835120901088
Score for architecture = [50, 50, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = linear: 0.841386998808
Score for architecture = [50, 50, 50, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = linear: 0.842348056741
Score for architecture = [50, 50, 50, 50, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = linear: 0.84350132626
Best Config: architecture = [50, 50, 50, 50, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = linear, best_acc = 0.84350132626
Time elapsed :38.4965770245

Observation:

- Test accuracies increase as the number of hidden layers increases in the network.
- This increase is due to the fact that the neural network will try to fit the data better as the number of hidden layer increases until it'll try to overfit the data.
- In our case, the increase in the test accuracies shows that overfit hasn't occurred yet.
- We get the best accuracy when the architecture is **[50, 50, 50, 50, 2]** with a value of **0.84350132626**

2)

Score for architecture = [50, 50, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = linear: 0.84065659478
Score for architecture = [50, 500, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = linear: 0.84307846077
Score for architecture = [50, 500, 300, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = linear: 0.846961134817
Score for architecture = [50, 800, 500, 300, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = linear: 0.850997578134
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = linear: 0.851535770576

Best Config: architecture = [50, 800, 800, 500, 300, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = linear, best_acc = 0.851535770576
Time elapsed :396.607818842

Observation:

- Similar to the previous section, the test accuracies increases with increase in the number of hidden layers in the network.
- The decrease in the number of neurons across the hidden layers helps to prune the nodes with lower weights and thus help to get a faster convergence.
- We notice that the time elapsed increases with the increase in the number of neurons in the hidden layer.
- We get the best accuracy when the architecture is **[50, 800, 800, 500, 300, 2]** with a value of **0.851535770576**

E) SIGMOID ACTIVATION

Score for architecture = [50, 50, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = sigmoid: 0.740821896744
Score for architecture = [50, 500, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = sigmoid: 0.767500864952
Score for architecture = [50, 500, 300, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = sigmoid: 0.717372083189
Score for architecture = [50, 800, 500, 300, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = sigmoid: 0.717372083189
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = sigmoid: 0.717372083189
Best Config: architecture = [50, 500, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = sigmoid, best_acc = 0.767500864952
Time elapsed :1263.39859009

Observations:

- We get the best accuracy when the architecture is **[50, 500, 2]** with a value of **0.767500864952.**
- The above architecture has a single hidden layer containing a many neurons thereby accounting to maximum testing accuracy.
- As the number of hidden layer increases in the sigmoid activation, the increase in the number of neurons in each layer have very little effect on the output of the sigmoid activation. This is due to the fact that the sigmoid activation(nonlinear function) tends to

constrain every input to a small output range[0,1]. This leads to the vanishing gradient problem.

- Due to vanishing gradient problem, Sigmoid activation provides the best accuracy for the architecture with less number of hidden layers which is in contrast with the performance of the linear activation model.
- In sigmoid activation, the magnitude of the gradient decreases exponentially (due to small output range) and the number of steps near the point of convergence to reach the point of convergence will be higher which directly attributes to the slower convergence to the output. Thus, it performs slower than linear activation.

F) ReLU ACTIVATION

Score for architecture = [50, 50, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = relu: 0.819167339407
Score for architecture = [50, 500, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = relu: 0.819974628071
Score for architecture = [50, 500, 300, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = relu: 0.810940683504
Score for architecture = [50, 800, 500, 300, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = relu: 0.812516818514
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = relu: 0.798024064891
Best Config: architecture = [50, 500, 2], lambda = 0.0, decay = 0.0, momentum = 0.0, actfn = relu, best_acc = 0.819974628071
Time elapsed :518.185555935

Observations:

- We get the best accuracy when the architecture is **[50, 500, 2]** with a value of **0.819974628071.**
- Rectified Linear Unit (ReLU) activation solves the vanishing gradient problem faced by the sigmoid activation as the activation function of ReLu is ($f(x) = max(0,x)$).
- ReLU activation is also a gradient based activation. Because of its activation function, it tends to choose the architecture with less number of hidden layers so as to avoid the

gradient explosion. Thus, the selection of architecture is in contrast to that of the linear activation.

- The irregularity in the test accuracies when the number of hidden layer increases is due to the fact that ReLU tends to blow up activations as it's output ranges from zero to infinity.
- Since ReLU doesn't have the vanishing gradient problem, it converges faster than the sigmoid activation function.
- The higher training time for ReLU when compared to the linear function is due to its complex activation function when compared to the linear function.

G) L2 REGULARIZATION

Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-07, decay = 0.0, momentum = 0.0, actfn = relu: 0.806366047745
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 0.0, momentum = 0.0, actfn = relu: 0.816783915734
Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-06, decay = 0.0, momentum = 0.0, actfn = relu: 0.804367047246
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-06, decay = 0.0, momentum = 0.0, actfn = relu: 0.810479375697
Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-05, decay = 0.0, momentum = 0.0, actfn = relu: 0.810825356552
Best Config: architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 0.0, momentum = 0.0, actfn = relu, best_acc = 0.816783915734
Time elapsed :786.397320032

Observations:

- We get the best accuracy when the architecture is **[50, 800, 500, 300, 2]** and for lambda value **5e-07** with a value of **0.816783915734.**
- The fluctuation in the accuracies is due to the combination of ReLU activation function and the regularization parameter. Same trend was seen when the lambda is set was zero (previous section)

H) EARLY STOPPING AND L2 REGULARIZATION

Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-07, decay = 0.0, momentum = 0.0, actfn = relu: 0.8035982009
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 0.0, momentum = 0.0, actfn = relu: 0.798562257333

Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-06, decay = 0.0, momentum = 0.0, actfn = relu: 0.791373543997
Epoch 00008: early stopping
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-06, decay = 0.0, momentum = 0.0, actfn = relu: 0.76211894053
Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-05, decay = 0.0, momentum = 0.0, actfn = relu: 0.800099950025
Best Config: architecture = [50, 800, 500, 300, 2], lambda = 1e-07, decay = 0.0, momentum = 0.0, actfn = relu, best_acc = 0.8035982009
Time elapsed :631.252521038

Observations:

- We get the best accuracy when the architecture is **[50, 800, 500, 300, 2]** with a value of **0.8035982009.**
- **Best lambda** is **1e-07** which is different from L2 regularization.
- Though Early stopping has decreased the accuracy, the variation is the accuracies is very minute.
- Because of early stopping the time taken for computation has decreased.

I) SGD WITH WEIGHT DECAY

Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 1e-05, momentum = 0.0, actfn = relu: 0.780994118325
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 5e-05, momentum = 0.0, actfn = relu: 0.716026602084
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 0.0001, momentum = 0.0, actfn = relu: 0.774997116826
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 0.0003, momentum = 0.0, actfn = relu: 0.748587244839
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 0.0007, momentum = 0.0, actfn = relu: 0.741859839311
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 0.001, momentum = 0.0, actfn = relu: 0.614923307577
Best Config: architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 1e-05, momentum = 0.0, actfn = relu, best_acc = 0.780994118325
Time elapsed :3230.63407516

Observations:

- The best value of decay is **1e-05**

J) MOMENTUM

Score for architecture = [50, 800, 500, 300, 2], lambda = 0.0, decay = 1e-05, momentum = 0.99, actfn = relu: 0.850382501057
Score for architecture = [50, 800, 500, 300, 2], lambda = 0.0, decay = 1e-05, momentum = 0.98, actfn = relu: 0.825471879445
Score for architecture = [50, 800, 500, 300, 2], lambda = 0.0, decay = 1e-05, momentum = 0.95, actfn = relu: 0.771960173759
Score for architecture = [50, 800, 500, 300, 2], lambda = 0.0, decay = 1e-05, momentum = 0.9, actfn = relu: 0.761503863453
Score for architecture = [50, 800, 500, 300, 2], lambda = 0.0, decay = 1e-05, momentum = 0.85, actfn = relu: 0.732364586937
Best Config: architecture = [50, 800, 500, 300, 2], lambda = 0.0, decay = 1e-05, momentum = 0.99, actfn = relu, best_acc = 0.850382501057
Time elapsed :1265.07936907

Observations:

- The best value of decay is **0.99**

K) COMBINATION OF ABOVE

Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-07, decay = 1e-05, momentum = 0.99, actfn = relu: 0.870564717641
Best Config: architecture = [50, 800, 500, 300, 2], lambda = 1e-07, decay = 1e-05, momentum = 0.99, actfn = relu, best_acc = 0.870564717641
Time elapsed :488.091462135

Observations:

- **This accuracy (0.870564717641) is better than all the accuracies that has been seen for this particular architecture ([50, 800, 500, 300, 2]).**

L) GRID SEARCH WITH CROSS-VALIDATION

Score for architecture = [50, 50, 2], lambda = 1e-07, decay = 1e-05, momentum = 0.99, actfn = relu: 0.840733479414
Score for architecture = [50, 50, 2], lambda = 1e-07, decay = 5e-05, momentum = 0.99, actfn = relu: 0.839234229039
Score for architecture = [50, 50, 2], lambda = 1e-07, decay = 0.0001, momentum = 0.99, actfn = relu: 0.843001576135
Score for architecture = [50, 50, 2], lambda = 5e-07, decay = 1e-05, momentum = 0.99, actfn = relu: 0.842540268327
Epoch 00013: early stopping
Score for architecture = [50, 50, 2], lambda = 5e-07, decay = 5e-05, momentum = 0.99, actfn = relu: 0.825471879445
Score for architecture = [50, 50, 2], lambda = 5e-07, decay = 0.0001, momentum = 0.99, actfn = relu: 0.838388498059
Epoch 00012: early stopping
Score for architecture = [50, 50, 2], lambda = 1e-06, decay = 1e-05, momentum = 0.99, actfn = relu: 0.798139391843
Score for architecture = [50, 50, 2], lambda = 1e-06, decay = 5e-05, momentum = 0.99, actfn = relu: 0.840118402337
Score for architecture = [50, 50, 2], lambda = 1e-06, decay = 0.0001, momentum = 0.99, actfn = relu: 0.846384500058
Score for architecture = [50, 50, 2], lambda = 5e-06, decay = 1e-05, momentum = 0.99, actfn = relu: 0.842578710645
Score for architecture = [50, 50, 2], lambda = 5e-06, decay = 5e-05, momentum = 0.99, actfn = relu: 0.839349555991
Score for architecture = [50, 50, 2], lambda = 5e-06, decay = 0.0001, momentum = 0.99, actfn = relu: 0.843347556991
Epoch 00079: early stopping
Score for architecture = [50, 50, 2], lambda = 1e-05, decay = 1e-05, momentum = 0.99, actfn = relu: 0.841848306616
Score for architecture = [50, 50, 2], lambda = 1e-05, decay = 5e-05, momentum = 0.99, actfn = relu: 0.835274670357
Score for architecture = [50, 50, 2], lambda = 1e-05, decay = 0.0001, momentum = 0.99, actfn = relu: 0.838503825011
Score for architecture = [50, 500, 2], lambda = 1e-07, decay = 1e-05, momentum = 0.99, actfn = relu: 0.848114404336
Score for architecture = [50, 500, 2], lambda = 1e-07, decay = 5e-05, momentum = 0.99, actfn = relu: 0.845615653712

Score for architecture = [50, 500, 2], lambda = 1e-07, decay = 0.0001, momentum = 0.99, actfn = relu: 0.843001576135
Score for architecture = [50, 500, 2], lambda = 5e-07, decay = 1e-05, momentum = 0.99, actfn = relu: 0.851689539845
Score for architecture = [50, 500, 2], lambda = 5e-07, decay = 5e-05, momentum = 0.99, actfn = relu: 0.845269672856
Score for architecture = [50, 500, 2], lambda = 5e-07, decay = 0.0001, momentum = 0.99, actfn = relu: 0.844731480414
Score for architecture = [50, 500, 2], lambda = 1e-06, decay = 1e-05, momentum = 0.99, actfn = relu: 0.849690539346
Score for architecture = [50, 500, 2], lambda = 1e-06, decay = 5e-05, momentum = 0.99, actfn = relu: 0.846192288471
Epoch 00089: early stopping
Score for architecture = [50, 500, 2], lambda = 1e-06, decay = 0.0001, momentum = 0.99, actfn = relu: 0.838542267328
Epoch 00008: early stopping
Score for architecture = [50, 500, 2], lambda = 5e-06, decay = 1e-05, momentum = 0.99, actfn = relu: 0.812632145466
Score for architecture = [50, 500, 2], lambda = 5e-06, decay = 5e-05, momentum = 0.99, actfn = relu: 0.847768423481
Score for architecture = [50, 500, 2], lambda = 5e-06, decay = 0.0001, momentum = 0.99, actfn = relu: 0.845384999808
Score for architecture = [50, 500, 2], lambda = 1e-05, decay = 1e-05, momentum = 0.99, actfn = relu: 0.847653096529
Epoch 00010: early stopping
Score for architecture = [50, 500, 2], lambda = 1e-05, decay = 5e-05, momentum = 0.99, actfn = relu: 0.81128666436
Score for architecture = [50, 500, 2], lambda = 1e-05, decay = 0.0001, momentum = 0.99, actfn = relu: 0.846422942375
Score for architecture = [50, 500, 300, 2], lambda = 1e-07, decay = 1e-05, momentum = 0.99, actfn = relu: 0.856917694999
Epoch 00009: early stopping
Score for architecture = [50, 500, 300, 2], lambda = 1e-07, decay = 5e-05, momentum = 0.99, actfn = relu: 0.794718025603
Epoch 00009: early stopping
Score for architecture = [50, 500, 300, 2], lambda = 1e-07, decay = 0.0001, momentum = 0.99, actfn = relu: 0.798100949525
Score for architecture = [50, 500, 300, 2], lambda = 5e-07, decay = 1e-05, momentum = 0.99, actfn = relu: 0.859454887941

Score for architecture = [50, 500, 300, 2], lambda = 5e-07, decay = 5e-05, momentum = 0.99, actfn = relu: 0.854726482912
Score for architecture = [50, 500, 300, 2], lambda = 5e-07, decay = 0.0001, momentum = 0.99, actfn = relu: 0.852496828509
Score for architecture = [50, 500, 300, 2], lambda = 1e-06, decay = 1e-05, momentum = 0.99, actfn = relu: 0.859800868796
Score for architecture = [50, 500, 300, 2], lambda = 1e-06, decay = 5e-05, momentum = 0.99, actfn = relu: 0.853534771076
Score for architecture = [50, 500, 300, 2], lambda = 1e-06, decay = 0.0001, momentum = 0.99, actfn = relu: 0.850843808865
Score for architecture = [50, 500, 300, 2], lambda = 5e-06, decay = 1e-05, momentum = 0.99, actfn = relu: 0.857840310614
Score for architecture = [50, 500, 300, 2], lambda = 5e-06, decay = 5e-05, momentum = 0.99, actfn = relu: 0.85518779072
Score for architecture = [50, 500, 300, 2], lambda = 5e-06, decay = 0.0001, momentum = 0.99, actfn = relu: 0.854495829009
Epoch 00008: early stopping
Score for architecture = [50, 500, 300, 2], lambda = 1e-05, decay = 1e-05, momentum = 0.99, actfn = relu: 0.794871794872
Score for architecture = [50, 500, 300, 2], lambda = 1e-05, decay = 5e-05, momentum = 0.99, actfn = relu: 0.853880751932
Score for architecture = [50, 500, 300, 2], lambda = 1e-05, decay = 0.0001, momentum = 0.99, actfn = relu: 0.850536270326
Epoch 00008: early stopping
Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-07, decay = 1e-05, momentum = 0.99, actfn = relu: 0.779994618076
Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-07, decay = 5e-05, momentum = 0.99, actfn = relu: 0.864644600777
Epoch 00008: early stopping
Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-07, decay = 0.0001, momentum = 0.99, actfn = relu: 0.774305155115
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 1e-05, momentum = 0.99, actfn = relu: 0.86745088994
Epoch 00008: early stopping
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 5e-05, momentum = 0.99, actfn = relu: 0.763310652366
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-07, decay = 0.0001, momentum = 0.99, actfn = relu: 0.861799869296

Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-06, decay = 1e-05, momentum = 0.99, actfn = relu: 0.863991081382
Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-06, decay = 5e-05, momentum = 0.99, actfn = relu: 0.86587475493
Epoch 00008: early stopping
Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-06, decay = 0.0001, momentum = 0.99, actfn = relu: 0.749471418137
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-06, decay = 1e-05, momentum = 0.99, actfn = relu: 0.866951139815
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-06, decay = 5e-05, momentum = 0.99, actfn = relu: 0.864759927728
Score for architecture = [50, 800, 500, 300, 2], lambda = 5e-06, decay = 0.0001, momentum = 0.99, actfn = relu: 0.863145350402
Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-05, decay = 1e-05, momentum = 0.99, actfn = relu: 0.866682043594
Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-05, decay = 5e-05, momentum = 0.99, actfn = relu: 0.859570214893
Score for architecture = [50, 800, 500, 300, 2], lambda = 1e-05, decay = 0.0001, momentum = 0.99, actfn = relu: 0.858570714643
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 1e-07, decay = 1e-05, momentum = 0.99, actfn = relu: 0.875485334256
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 1e-07, decay = 5e-05, momentum = 0.99, actfn = relu: 0.86987275593
Epoch 00008: early stopping
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 1e-07, decay = 0.0001, momentum = 0.99, actfn = relu: 0.771306654365
Epoch 00008: early stopping
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 5e-07, decay = 1e-05, momentum = 0.99, actfn = relu: 0.734056048899
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 5e-07, decay = 5e-05, momentum = 0.99, actfn = relu: 0.869757428978
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 5e-07, decay = 0.0001, momentum = 0.99, actfn = relu: 0.859839311114
Epoch 00007: early stopping
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 1e-06, decay = 1e-05, momentum = 0.99, actfn = relu: 0.744397032253
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 1e-06, decay = 5e-05, momentum = 0.99, actfn = relu: 0.865951639565

Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 1e-06, decay = 0.0001, momentum = 0.99, actfn = relu: 0.861915196248
Epoch 00008: early stopping
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 5e-06, decay = 1e-05, momentum = 0.99, actfn = relu: 0.753161880598
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 5e-06, decay = 5e-05, momentum = 0.99, actfn = relu: 0.864836812363
Epoch 00007: early stopping
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 5e-06, decay = 0.0001, momentum = 0.99, actfn = relu: 0.741052550648
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 1e-05, decay = 1e-05, momentum = 0.99, actfn = relu: 0.872755929727
Epoch 00007: early stopping
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 1e-05, decay = 5e-05, momentum = 0.99, actfn = relu: 0.733710068043
Epoch 00007: early stopping
Score for architecture = [50, 800, 800, 500, 300, 2], lambda = 1e-05, decay = 0.0001, momentum = 0.99, actfn = relu: 0.7260600469
Best Config: architecture = [50, 800, 800, 500, 300, 2], lambda = 1e-07, decay = 1e-05, momentum = 0.99, actfn = relu, best_acc = 0.875485334256

Time elapsed :17153.4044189

Observations:

- **Best Lambda : 1e-07**
- **Best decay : 1e-05**
- **Best Momentum : 0.99**
- **Best accuracy : 0.875485334256**

COLLABORATORS : SINDHUJHA SETHURAMAN, LAVANYA KUMAR