# Assignment 1 - report - Intro To ML

Mtech - 2nd Sem

---

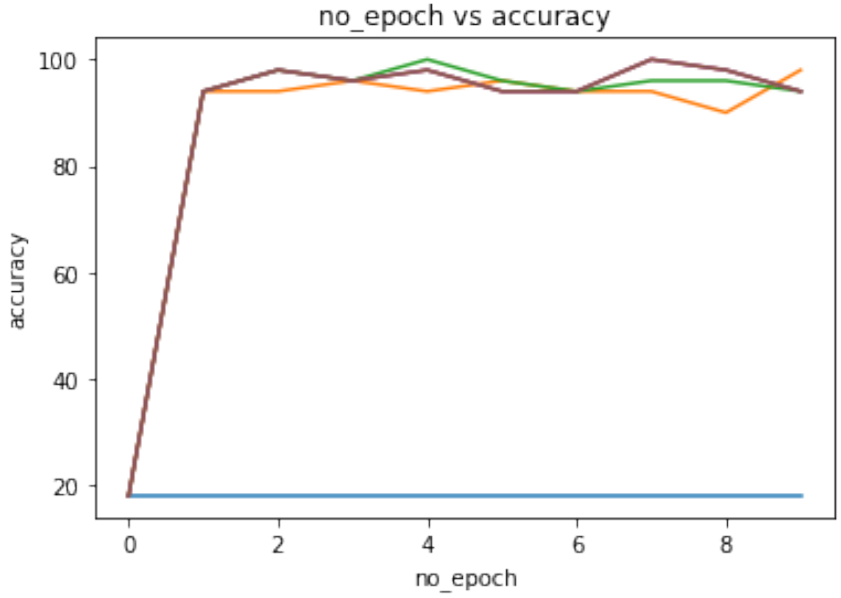
K.Ranjith - MIT2020017

21-02-2021

**IIIT Allahabad**

# problem 1b - accuracy with respect to number of epoch - for Logistic Regression - for varying $\alpha$ - for train data = 600

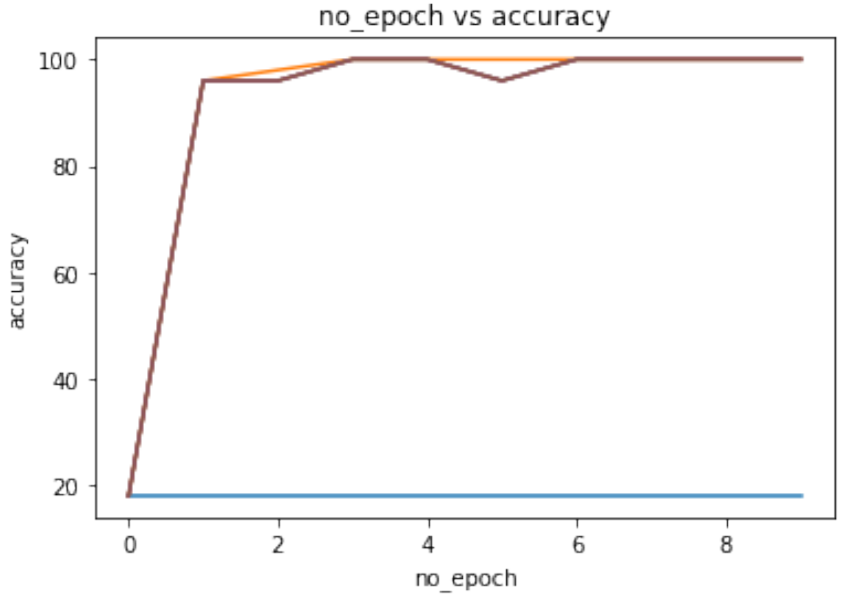
hyperparameters[no-train $= 600$, no-test $= 50$,
$\alpha = [0, 0.0001, 0.001, 0.01, 0.1, 1]$, $no - epoch = [0, 9]$]
x-axis-epoch , y-axis-accuracy, $\alpha = [blue, orange, green, red, voilet, brown]$




for any value $\alpha$ there are minute flutuations after epoch 1
how ever curves with respect to $\alpha = [0.01, 0.1, 1]$ overlaps

# problem 1b - accuracy with respect to number of epoch - for Logistic Regression - for varying $\alpha$ – for train data = 200

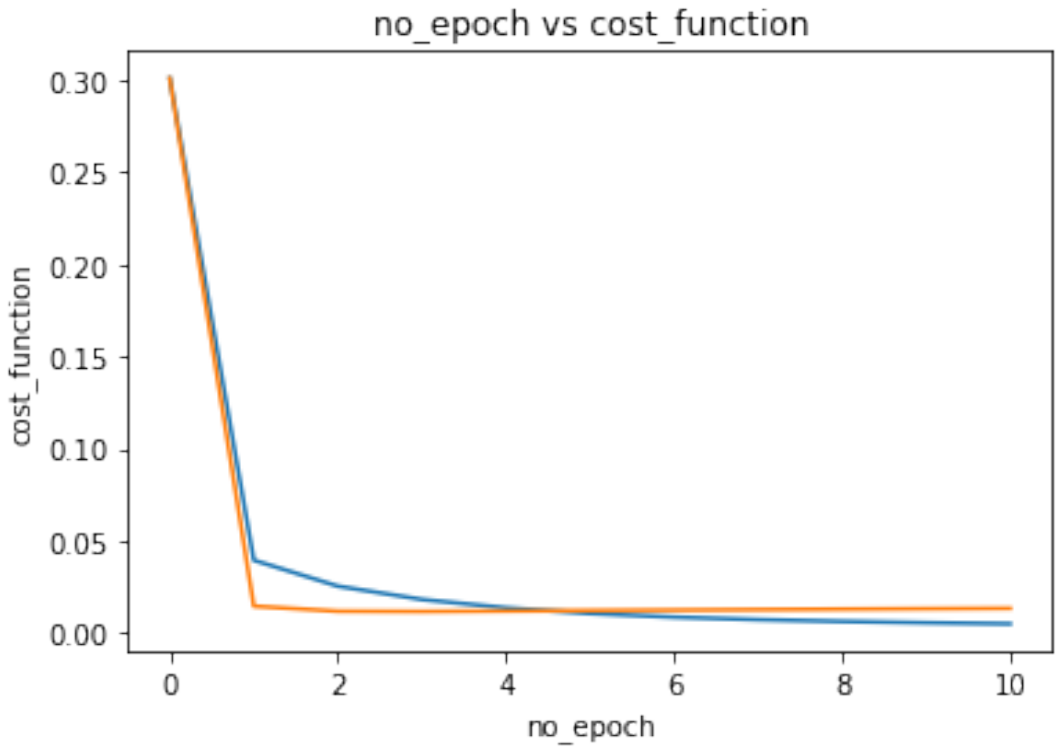
hyperparameters[no-train $= 200$, no-test $= 50$,
$\alpha = [0, 0.0001, 0.001, 0.01, 0.1, 1]$, $no - epoch = [0, 9]]$
x-axis-epoch , y-axis-accuracy, $\alpha = [blue, orange, green, red, voilet, brown]$




for $\alpha$= 0.0001 we can see a smooth orange curve
for remaining values of $\alpha$ the curves overlap, These curves has little fluctuations wrt noepoch due to oscillations at minimum point.

# problem 4a - Loss Function 1 - $\alpha$=0.000001

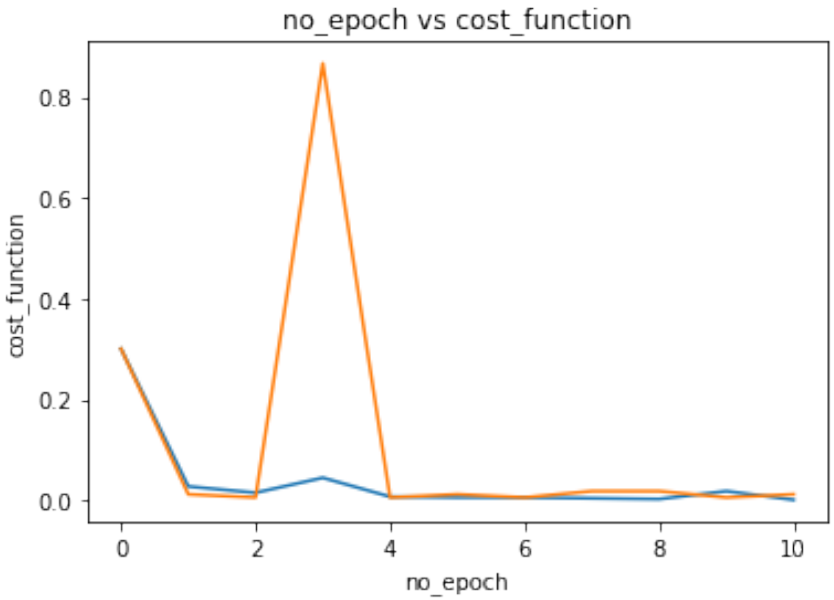
hyperparameters[no-train = 600, no-validation = 240, no-epoch = 10, $\alpha = 0.000001$]
orange - Validation Set, blue - Train Set




when less $\alpha$ is taken model is good(less error) but model should be trained with more number of epoch(10) because train and validation curves goes parallel after 10 epoch.

# problem 4a - Loss Function 1 - $\alpha$=0.1

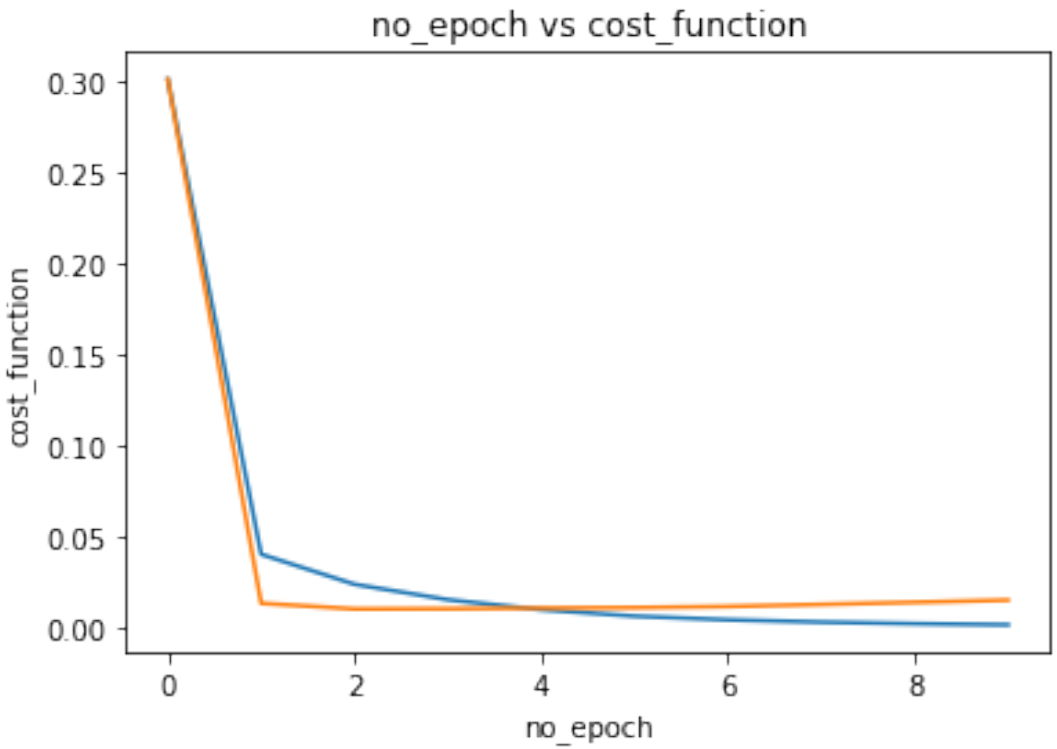
hyperparameters[no-train = 600, no-validation = 240, no-epoch = 10, $\alpha = 0.1$]
orange - Validation Set, blue - Train Set




when greater $\alpha$ is taken model is relatively bad. but model can be trained with just epoch(1). bacause train and validation curves goes parallel just after first epoch. That spike in validation curve at epoch(3) may be due to missing of minimum value during gradient decent due to large learning rate

# problem 4a - Loss Function 2 - $\alpha = 0.000001$

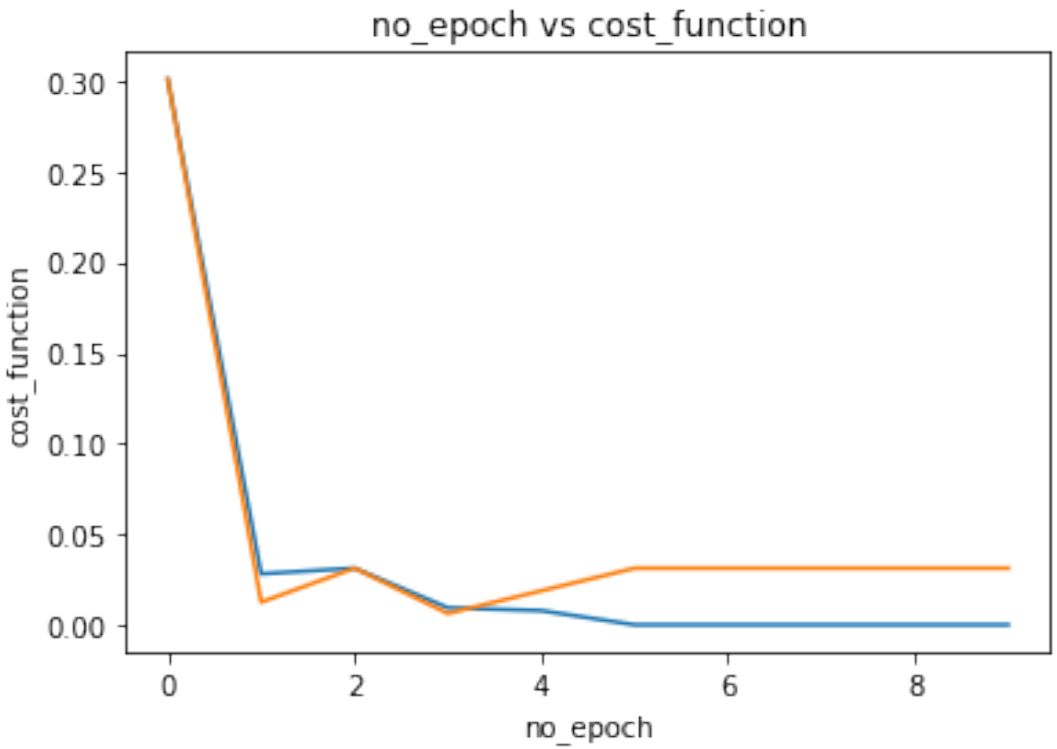
hyperparameters [no-train $= 600$, no-validation $= 240$, no-epoch $= 10$, $\alpha = 0.000001, \lambda = 0.1$]
orange - Validation set, blue - train set




After L2 regulization the cost function of train data crosses validation data at 4th epoch vs for normal cost funtion it crosses at 5th epoch. So with L2 regulization we are getting better model for less epoch

# problem 4a - Loss Function 2 - $\alpha = 0.1$

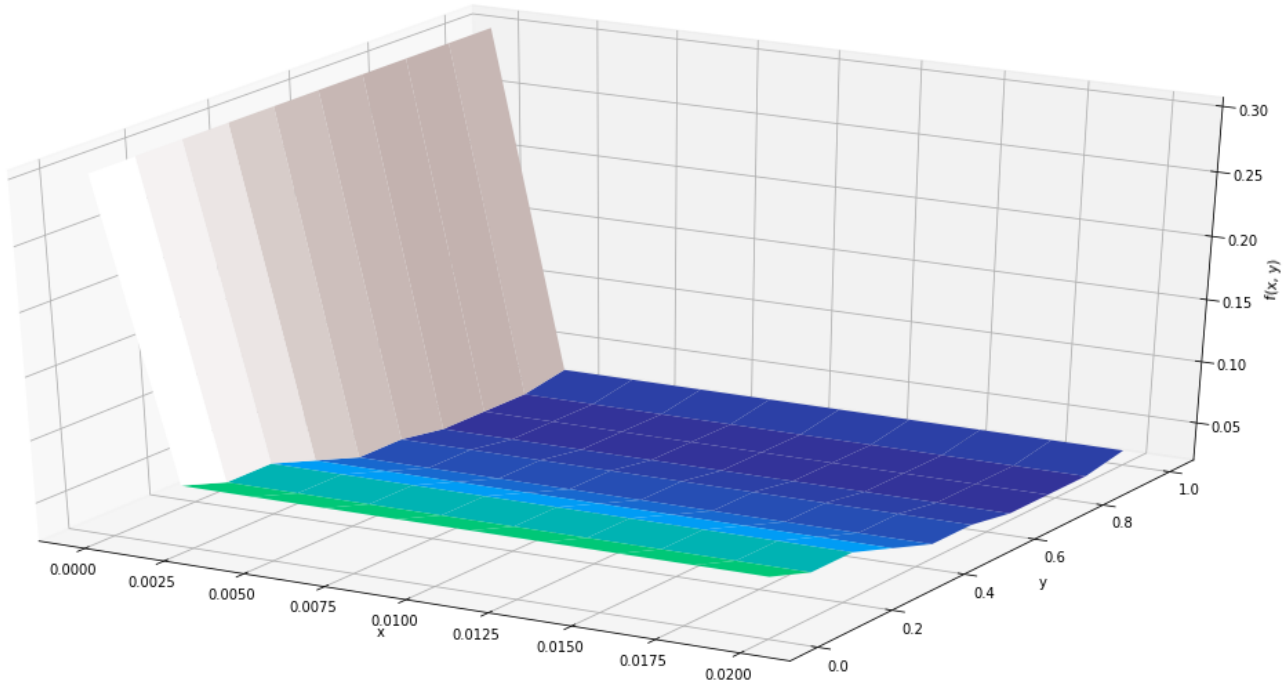
hyperparameters [no-train $= 600$, no-validation $= 240$, no-epoch $= 10$, $\alpha = 0.1, \lambda = 0.1$]
orange - Validation set, blue - train set




After L2 regulization the train curve and test curve goes parallel from epoch 5, where as in normal cost functions curves overlap and there is fluctuations also. so after L2 regulizations we are getting better stable model

# problem 4b - lambda wrt learning rate

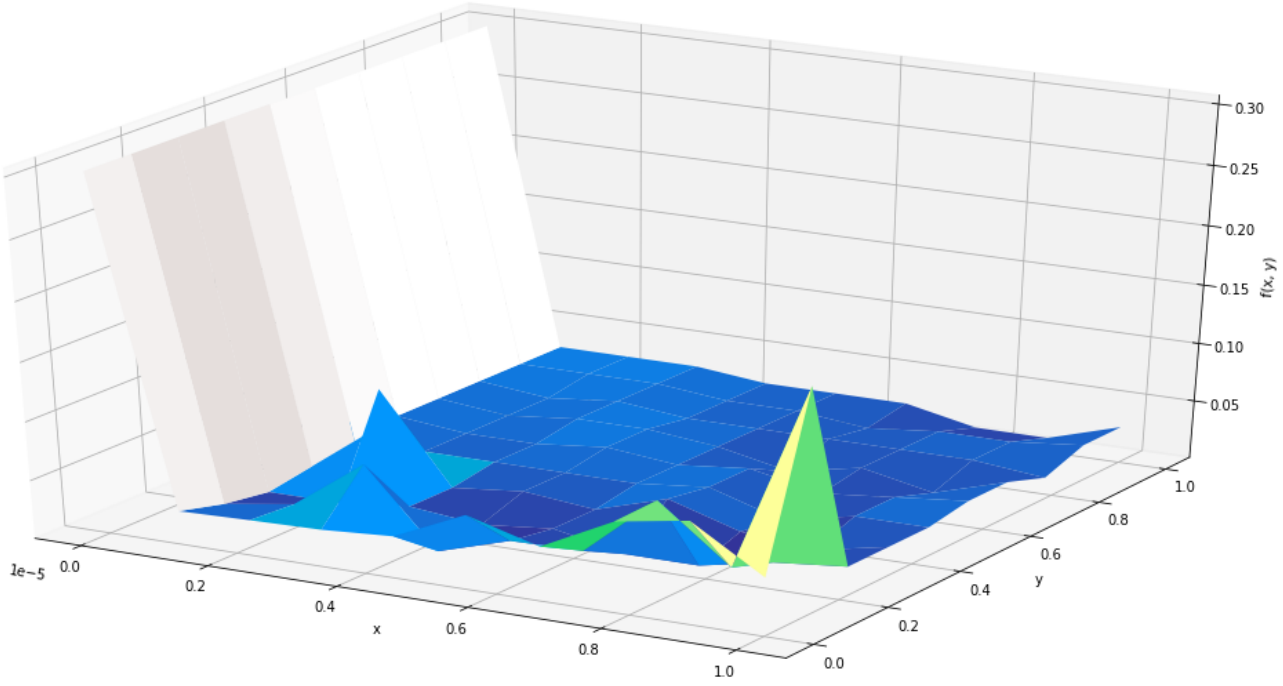
hyperparameters [no-train $= 250$, no-validation $= 240$, no-epoch $= 25$, $\alpha = [0, 0.02], \lambda = [0, 1]]$




when $\lambda = [0.5,0.9]$ the model is relatively great as cost funcions converges to low values compared to model with $\lambda = 0$. This is a classic example where model overfits to trian data, through L2 regulization we get best model

# problem 4b - lambda wrt learning rate

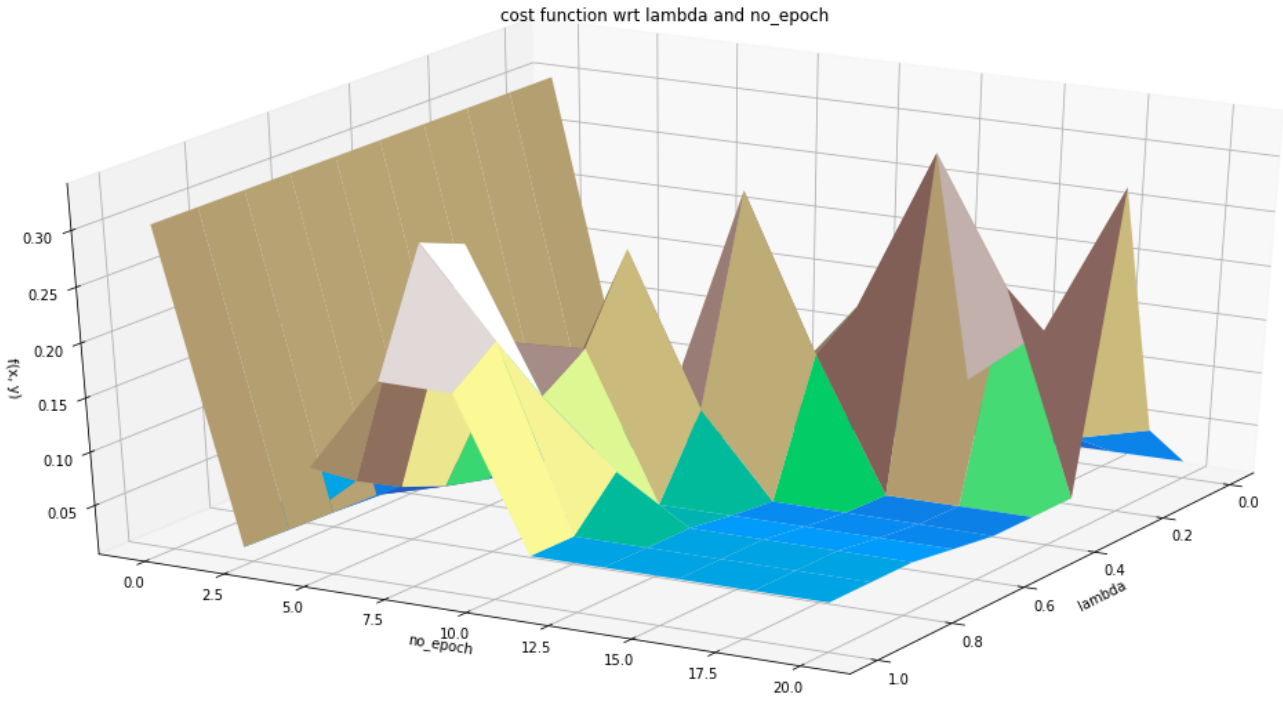
hyperparameters [no-train $= 250$, no-validation $= 240$, no-epoch $= 25$, $\alpha = [0, 0.00001], \lambda = [0, 1]]$




The model becomes less sensitive to $\alpha$ and becomes more stable after L2 regulization

# problem 4b - lambda wrt no-epoch

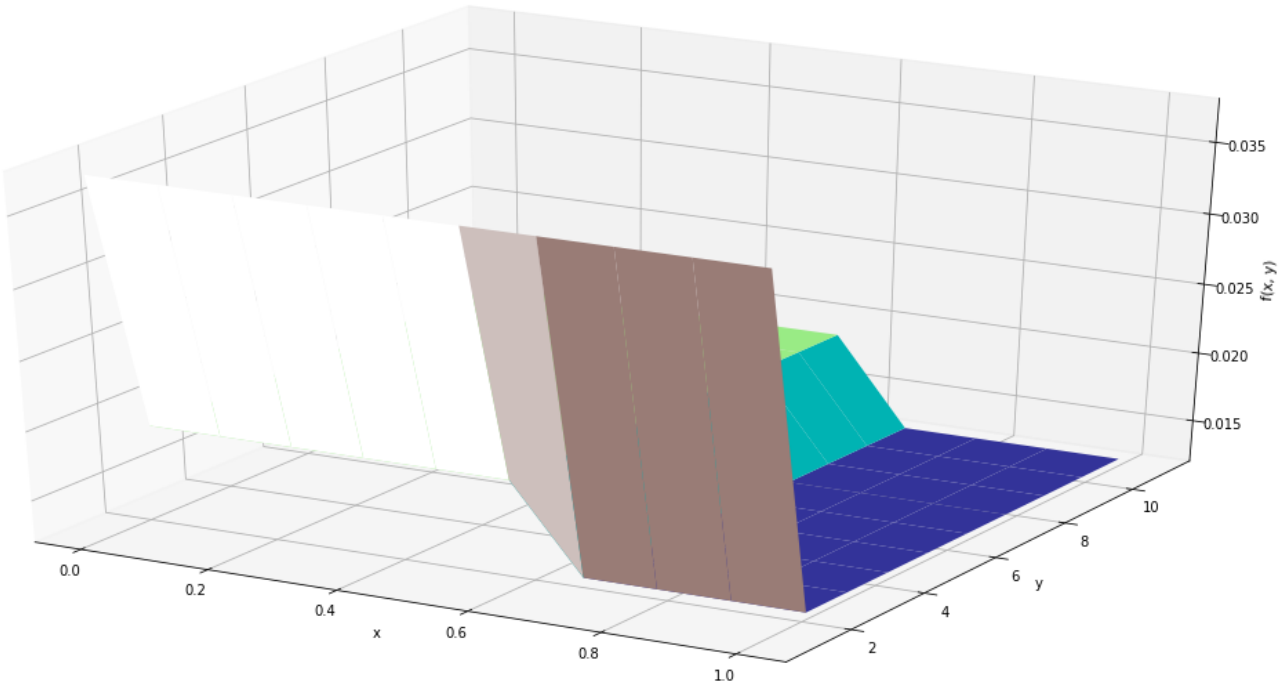
hyperparameters [no-train $= 600$, no-test $= 160$,
$\alpha = 0.000001, no - epoch = [0, 20], \lambda = [0, 1]]$




As you can see with normal cost function ($\lambda$ =0) there are lot of fluctuations in cost functions as model overfit to train data and there is more variance wrt to test data . This problem is solved with L2 regulizations the model optimizes for $\lambda$ = [0.5,1].

# problem 4b - lambda wrt no-epoch

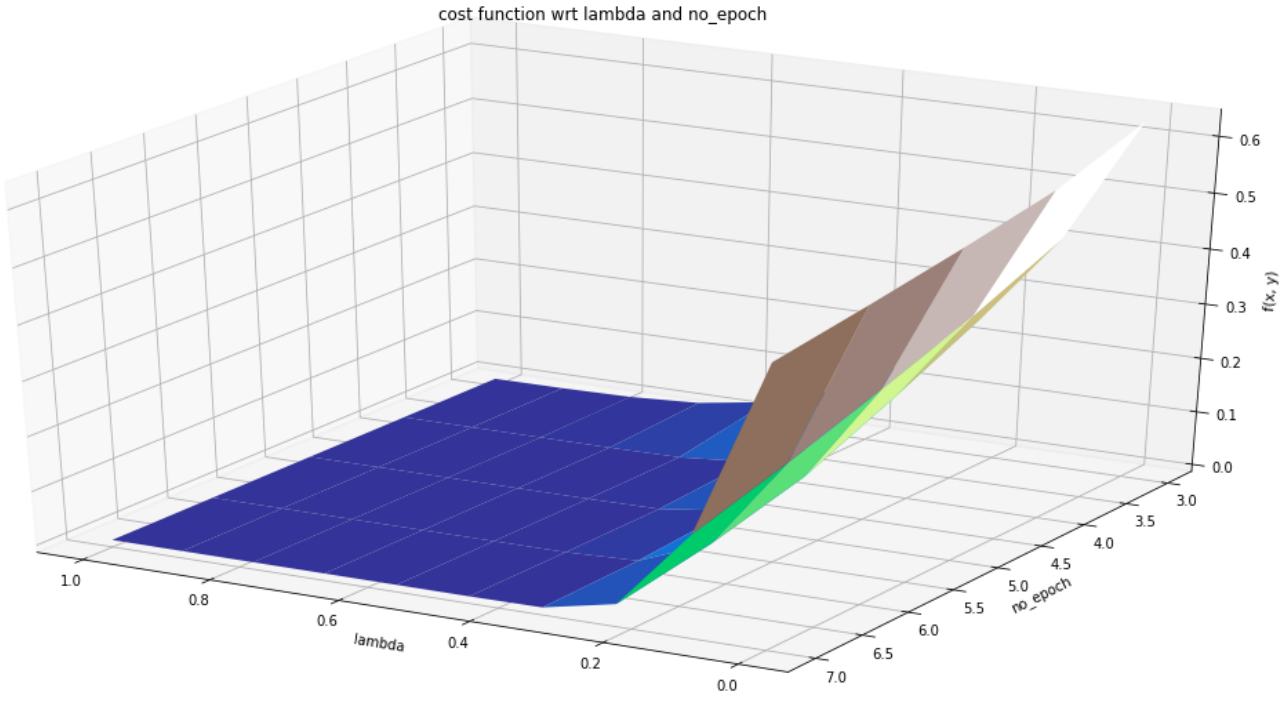
hyperparameters [no-train $= 600$, no-test $= 160$,
$\alpha = 0.01, no - epoch = [0, 20], \lambda = [0, 1]]$




after L2 regulization the model converges faster(less epoch) and better(less cost functions vaues) compared to using normal cost function

# problem 6 - final evaluations with test data

hyperparameters [totalSamples $= 420$,testdata $= 75$,
$\alpha = 0.000001$, $no - epoch = [3, 7]$, $\lambda = [0, 1]$]




*lambda* $= 0.44$, epoch $=5.0$, error is around 0.00104 .Important observation is with out regulization i.e for $\lambda = 0$, The model behaves badly due to over fitting with train data as test data data is set aside from training.When we are testing data with new data even with *lambda* $= 1$ model works great