1. **Double cross validation:**

def doDoubleCrossValidation (D, k, H):

allIdxs = np.arange(len(D))

# Randomly split dataset into k folds

idxs = np.random.permutation(allIdxs)

idxs = idxs.reshape(k, -1)

accuracies = []

acc =0

h\_star = np.random.choice(H) # Initially taking a random hyper parameter set as the best one

for fold in range(k):

# Get all indexes for this fold

testIdxs = idxs[fold,:]

# Get all the other indexes

trainIdxs = idxs[list(set(allIdxs) - set(testIdxs)),:].flatten()

for h in H:

new\_acc = doCrossValidation (D[trainIdxs], k, h)

if new\_acc > acc:

acc = new\_acc

h\_star = h

# Train the model on the training data

model = trainModel(D[trainIdxs], h\_star)

# Test the model on the testing data

accuracies.append(testModel(model, D[testIdxs]))

return np.mean(accuracies)

2. **Convexity:**

Hessian Matrix:

Eigen values of the Hessian matrix can be computed by:

Here the numerator of has two components: that is equal to , and that is slightly larger than . Therefore, the numerator is either positive or negative.

Therefore, the Hessian matrix has one positive eigen value and one negative eigen value. This means the is not Positive semi definite or PSD.

Therefore function  **is not convex**. This can also be verified by plotting this function. This plot is given by the following figure.

Chart, surface chart

Description automatically generated



For , the Hessian matrix is given by:

**Proof:**

Let , , and .

, can be simplified to the following:

(

Gradient of is :

Jacobian of is Hessian:

For any matrix , it is defined as positive semi definite, only if, for any real valued nonzero vector ,

Here, is the squared Euclidian norm of the vector . Squared Euclidian norm is always greater than or equal to zero for any vector. Hence

is positive semi definite, which means the is convex.

3. **Stochastic Gradient Descent:**

Hyperparameters that are tried:

n=[100,200,300]

epsilon=[0.1,0.01,0.001,0.000001,0.000001,0.00000001]

epochs=[1,2,3,4,5]

alpha=[0.01,0.1,1,2,3,4]

After the validation I found the following parameters to be best:

Batch Size: 300.0

Learning Rate: 1e-06

Epochs: 5.0

Regularization: 0.01

Errors are:

Train Error: 293.38360497

Test Error: 151.05474931

4**. Logistic Sigmoid Identities:**