Machine Learning Assignment Group 71

(18CS10021) Hardik Aggarwal and (18CS30040) Sriyash Poddar

Assignment - 3 - Support Vector Machines and ANN

Procedure :

SVM

1. Preprocess Data - Normalize the data to z = (z - u) / s

2. Split the dataset into an 80 : 20 ratio of Train and Test data .

3. Train the Support vector classifier across the 3 kernels [‘linear’, ‘Quadratic’, ‘Radial Basis Function’]
4. Vary the hyperparameters (C in this case) to find out the best set of hyperparameters according to the corresponding test accuracy.

5. We Report the plot of accuracy vs C for all the three kernels and corresponding values resulting in maximum accuracy.

ANN:

1. Preprocess Data - Normalize the data to z = (z - u) / s

2. Split the dataset into an 80 : 20 ratio of Train and Test data and prepare the batches of size 32.

3. For each Model architecture and each value of learning rate do the following :
    a. Feed the inputs, batchwise to the model
    b. Calculate the Y_preds
    c. Backpropagate the loss and update the weights
    d. We then calculate the accuracy over the data.
    e. We continue the training the model for a maximum of 500 iterations, but stop early if the loss has converged and does not improve for a period of 10 epochs
4. We then plot the corresponding graphs of accuracy vs various hyper parameters like learning rate , hidden layer , device(CPU or GPU).

Observations : (SVM)

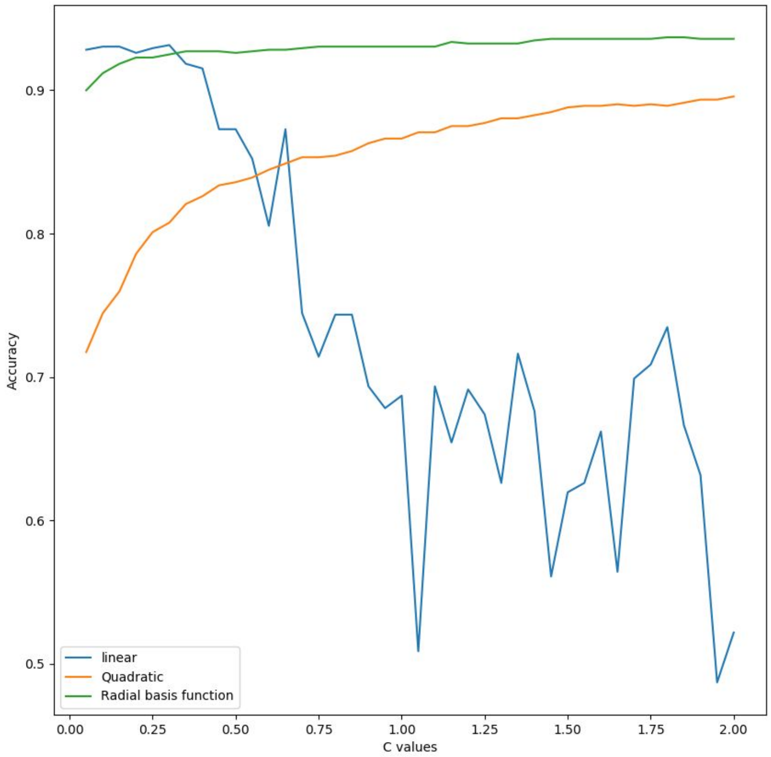

Fig . 1 Plot of Test Accuracy vs C for 3 different kernels

Kernel	Best C value	Train Accuracy	Test Accuracy
Linear	0.2	0.93532	0.93152
Quadratic	1.8	0.89293	0.87173
Radial Basis Function	1.55	0.95135	0.93695

Fig (a): Accuracy For different Kernels

Results : The data is clearly not linearly separable and hence the accuracy for the linear kernel is fluctuating. However for other kernels we are getting an general increase in accuracy vs C. The radial basis kernel is performing the best of 3 .

Final Best test accuracy is achieved for

- **Kernel : Radial Basis Function**
- **C : 1.55**

Best Test Accuracy : 0.93695

Observations (ANN) :

Other Hyperparameters of the model :

Number of epochs : 500
Weight decay parameter : 0.01
Batch size : 32
Optimizer : Stochastic Gradient Descent (SGD)
Activation : Relu (for hidden) and sigmoid (for output)
Loss function : BCE Loss

Nodes in input layer : 57 (number of attributes)
Nodes in output layer : 1 (binary classification)

Plots :

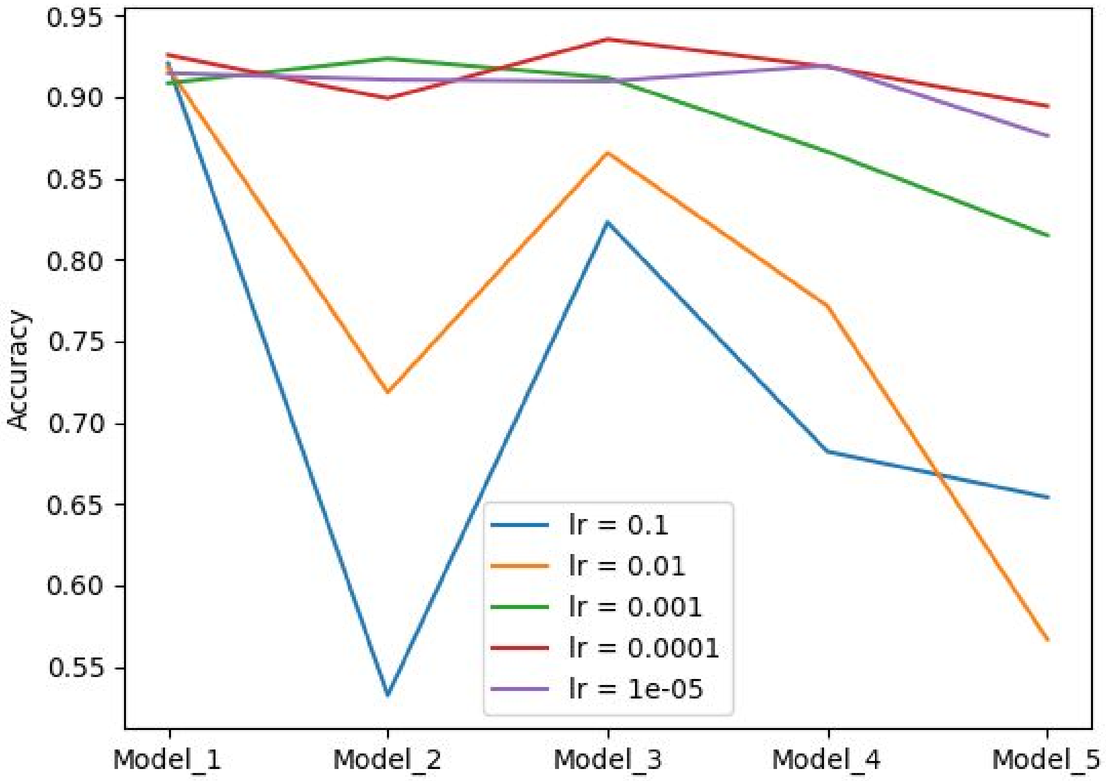

Fig 2. Variation of Test accuracy vs Models.(Learning rates also shown)

Inference : A general trend is that the accuracy is decreasing as the model complexity is increasing and we can infer that this is due to overfitting of training data most probably. So according to the Occasm's razor which states that use the most simple model , our results stand in strict accordance to that rule.

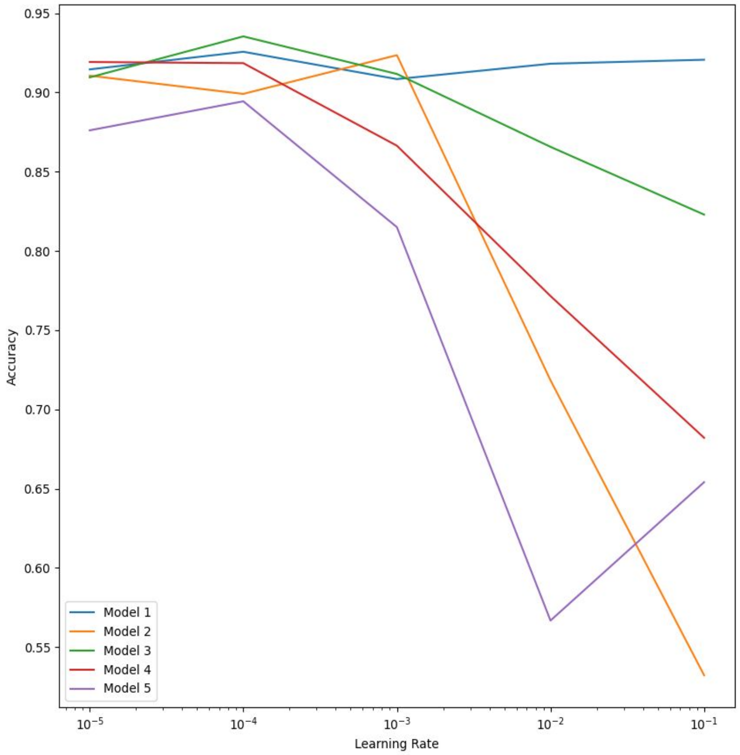

Fig 3. Test Accuracy vs the learning rate.

Inference : As the learning rate is increasing the accuracy is decreasing this is due to the fact the gradients are too large and if we increase the learning rate we are missing the optimum point and hence resulting in a low accuracy. So for almost all the models we are getting the maximum test accuracy at lr = 10^-4.

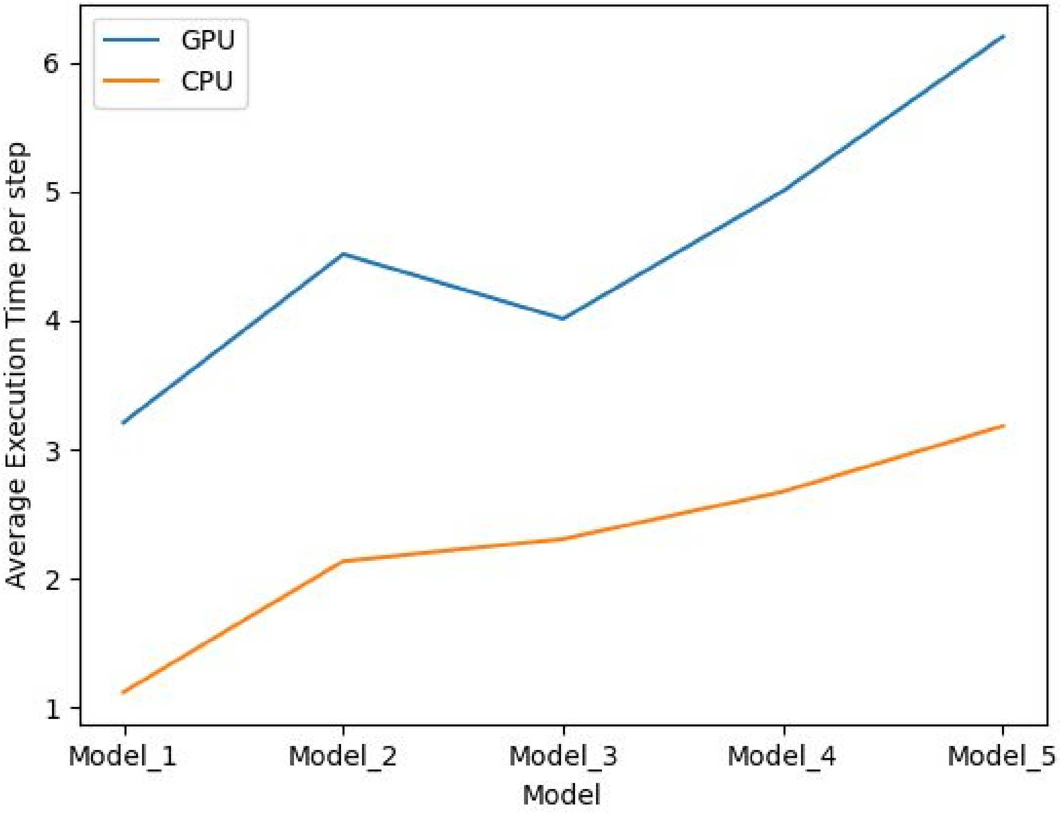

Fig. 4 Comparison of running time on CPU and GPU

Inference : The running time of the CPU is less as compared to GPU which is in contradiction to what we expected. One possible reason for this anomaly can be the improvement in running time because the GPU is less than the deterioration due the copy overhead to the GPU.

BEST MODEL :

"architecture": [6] , (1 hidden layer with 6 nodes)
"trainable_params": 355,
"lr": 0.0001,
"loss": 0.4428991650894549,
"accuracy": 0.9353448275862069

Best Test Accuracy : 0.9353448275862069

Final Comparison of Models : Though the final test accuracies of both SVM and ANN are pretty much similar (0.9369 and 0.9353 respectively) we see that both models perform relatively poor when either the kernel is too simple for eg linear and too poor when the number of hidden layers increases.
Also the ANN is more computationally expensive as compared to SVM and hence it takes more time to see the results due to costly operations like backpropagation etc.
Also one possible reason for the accuracy being almost similar is due to the noise present in the data and hence some saturation around the achieved value.