"CMSC828C/ENEE633 Project 2"

Statistical Pattern Recognition

PROJECT REPORT

Akshay Kurhade-116914529

Part 1-

Problem Statement-

Implement linear and kernel SVM on MNIST dataset using different kernels (linear, polynomial, RBF) and compare results.

Approach-

Reduce the dimensionality of the dataset using Principal Component Analysis(PCA), search for the best parameters and then implement SVM using best fit parameters. Finally fit it to classify the MNIST dataset.

Dimensionality Reduction-

It is the transformation of data from a high-dimensional space into a low-dimensional space so that dimension reduction retains meaningful properties of the original data, ideally close to its intrinsic dimension.

PCA-

In the sklearn implementation of PCA, Linear dimensionality reduction is done using Singular Value Decomposition(SVD) of the data to project it to a lower dimensional space. The input data is centered but not scaled for each feature before applying the SVD. It uses the randomized truncated SVD by the method of Halko et al. 2009, depending on the shape of the input data and the number of components to extract.

Parameter selection-

Hyperparameters for the Support Vector Machine(SVM) were found using GridSearchCV.

GridSearchCV does exhaustive search on the estimator parameters, The Sklearn implementation of SVC is based on libsvm.

The primary goal of SVM is to optimize-

Given a training set of instance-label pairs $(\mathbf{x}_i, y_i), i = 1, \ldots, l$ where $\mathbf{x}_i \in R^n$ and $\mathbf{y} \in \{1, -1\}^l$, the support vector machines (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995) require the solution of the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l} \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{1}$$

1

Fig- Boser et al 1992

Results-

The Radial Basis Function(RBF) performs better on the MNIST dataset giving an accuracy of almost 100%.

The RBF kernel can be defined as-

$$K(X_1, X_2) = exp(-\frac{||X_1 - X_2||^2}{2\sigma^2})$$

Sr. No.	Kernel	Score
1	Linear (C=0.5,gamma=0.1)	88.52%
2	rbf (C=010gamma=0.1)	99.85%
3	Polynomial (C=0.5,gamma=0.1)	98.11%

CNN Model-

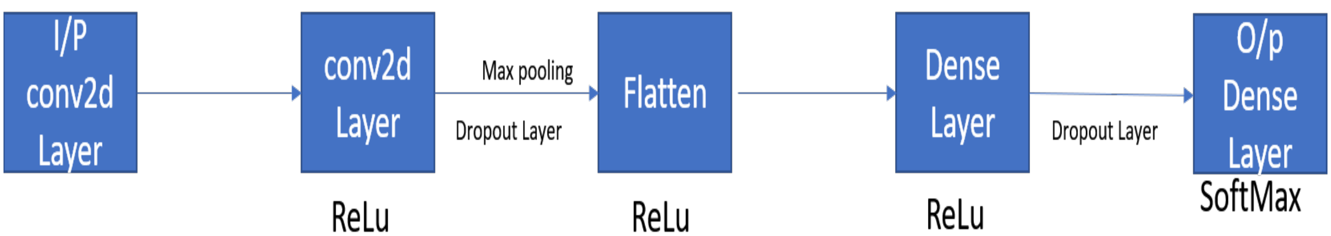


Fig. CNN model to train on MNIST dataset.

Hyperparameters-

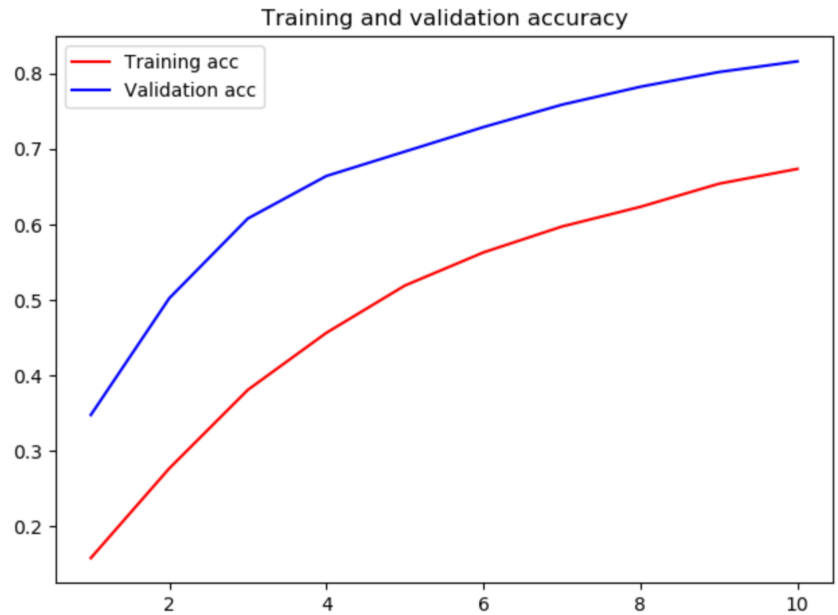
Epochs-10

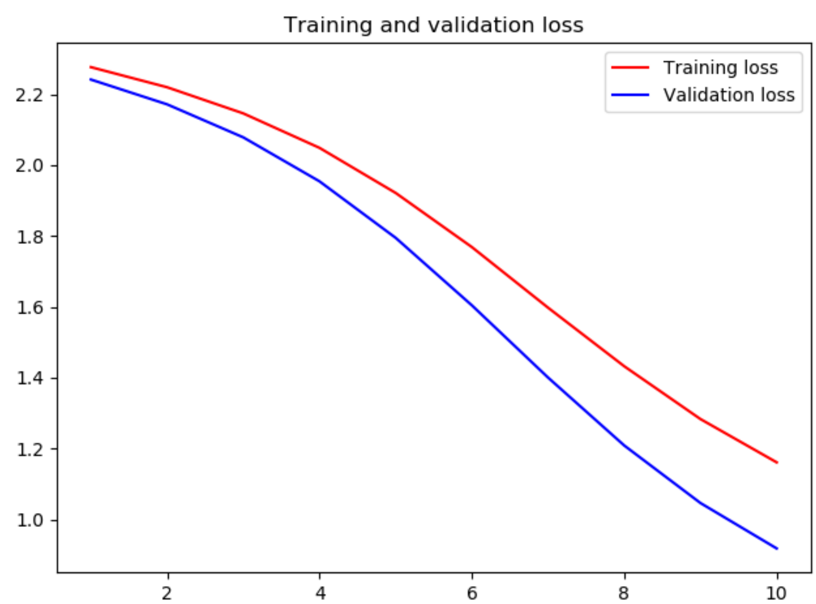
Batch size- 128

Activation- Relu

Optimizer- Adadelta

Validation-loss Curves-





Here the training loss is greater than validation loss inferring that the data split is unfair and would yield better results if trained on validation dataset.

The model achieved an accuracy of **~88%** on the validation dataset

Validation results-

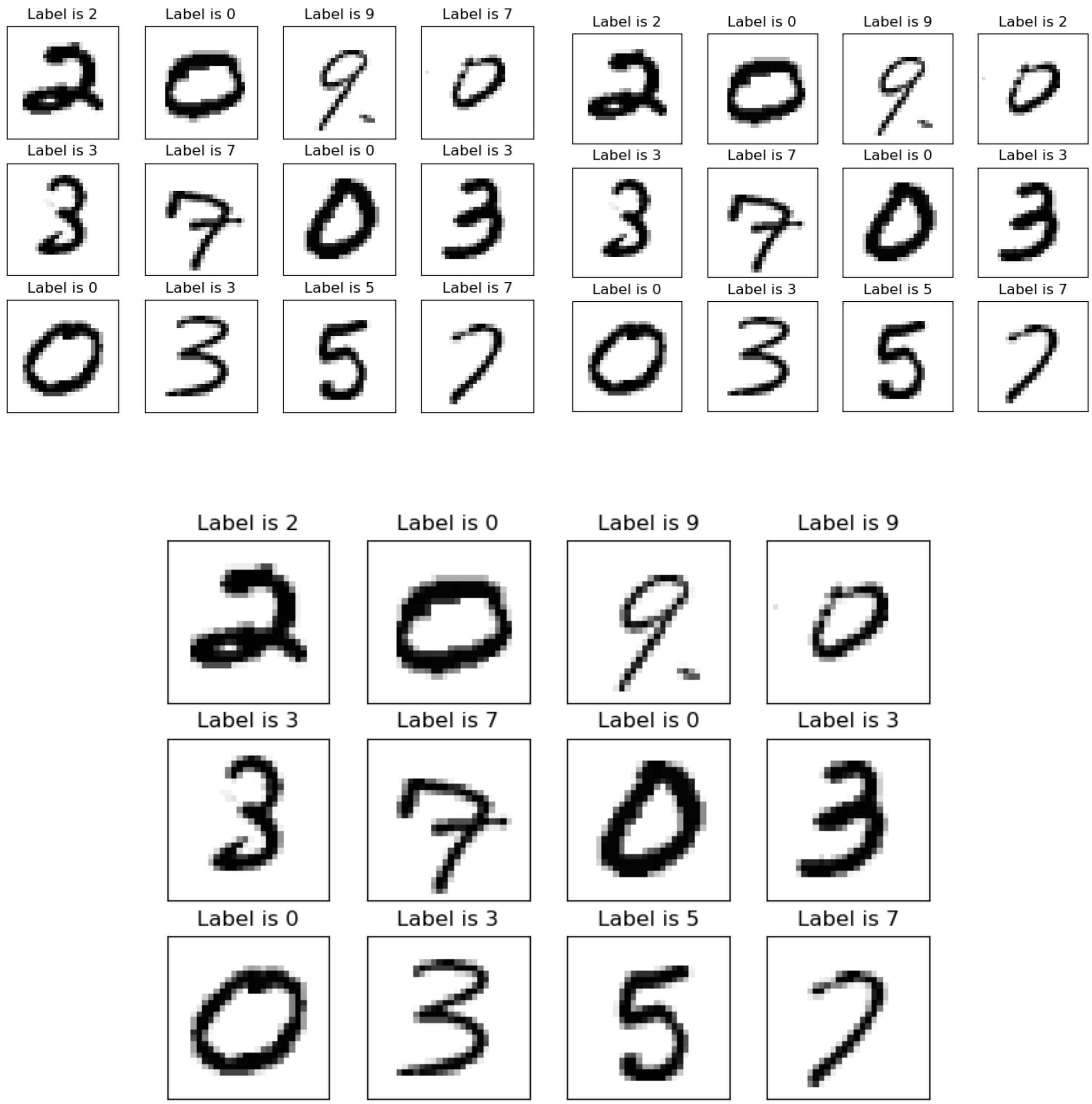


Fig: Results of SVM using linear, rbf, polynomial kernels respectively

Part 2-

Problem Statement-

To build and train a Convolutional Neural Network from scratch and train it on a small dataset (10 Dogs) and compare the validation results to a pretrained model on a bigger dataset like IMAGENET via transfer Learning-

Approach-

Build a simple convolutional neural network and try to optimize it based on the validation and accuracy curves and then compare its results to features extracted from a pretrained VGG16 model.

Model from Scratch-

Hyperparameter selection-

The hyperparameters were optimally selected using grid search method.

```
parameters = {'batch_size': [16, 32],
              'epochs': [10, 50, 100],
              'optimizer': ['adam', 'adamw',’rmsprop’],
              'kernel_initializer': ['random_uniform', 'normal'],
              'activation': ['relu', 'leaky-relu', 'elu'],
              }

grid_search = GridSearchCV(estimator = classifier,
                           param_grid = parameters,
                           scoring = 'accuracy', cv = 5)

grid_search = grid_search.fit(x_train_orig, y_train_orig)

best_param = grid_search.best_params_
best_param
best_acc = grid_search.best_score_
best_acc
```

Based on the results, following hyperparameters were selected-

Batch size=32

Epochs=10

Optimizer=rmsprop

Activation= ReLu

Experiment-

Initially developed a 3 convolutional layer network but the model severely underfitted, so increased the complexity to 5 layers and instead of a sigmoid at the classification layer switched to SoftMax.

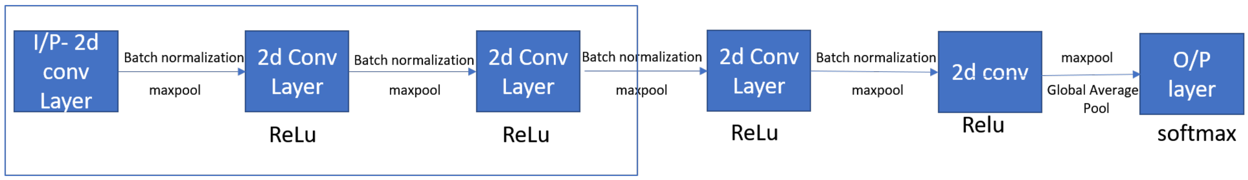
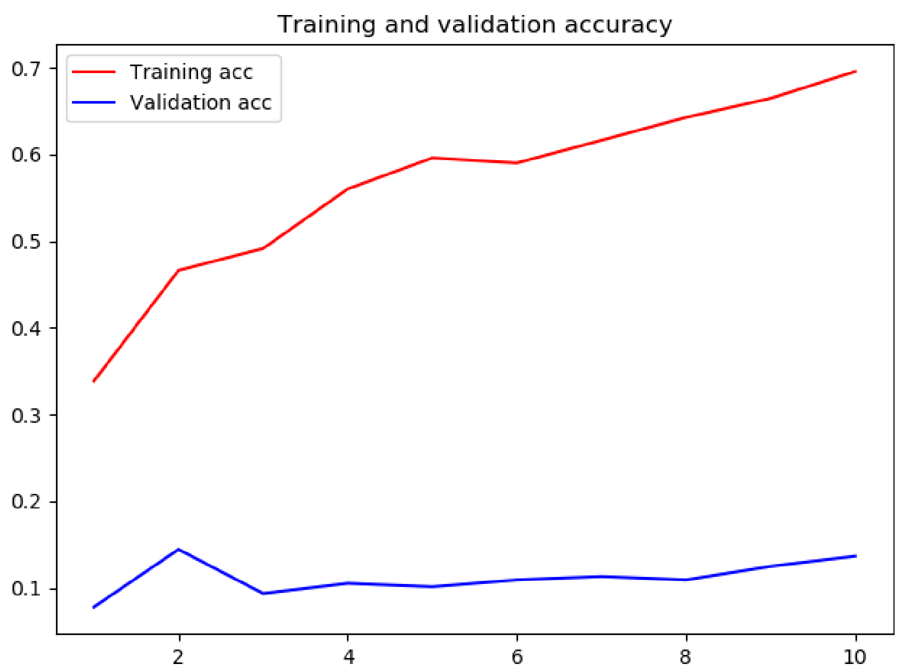
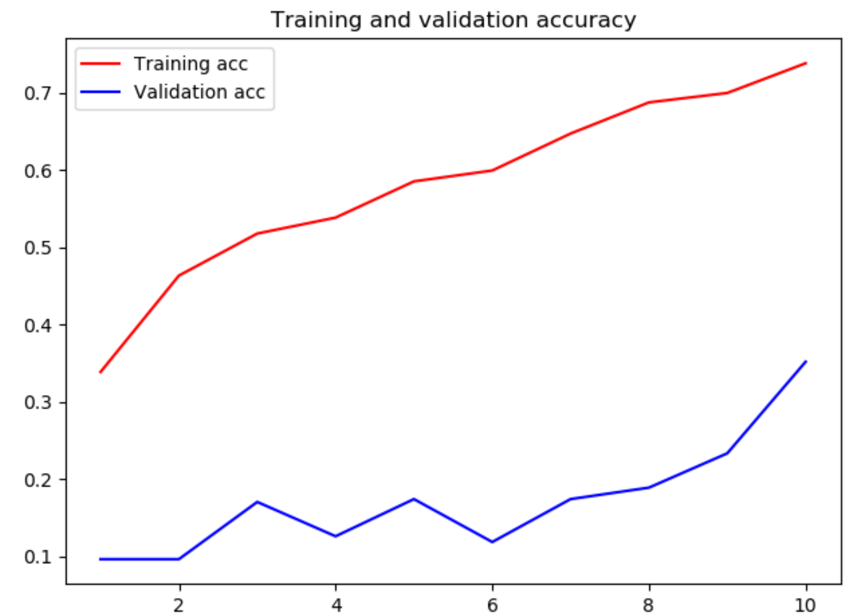


Fig- Model developed from scratch, (initial model bounded by the box)

3-layer model results-



5- layer tuned model results-



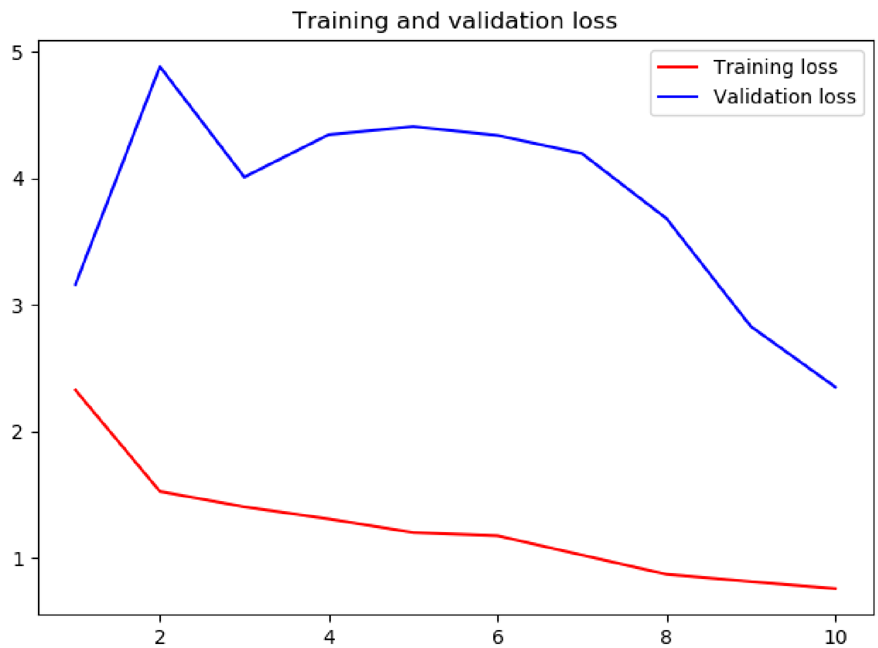


Fig- Training, Validation curves for model trained from scratch.

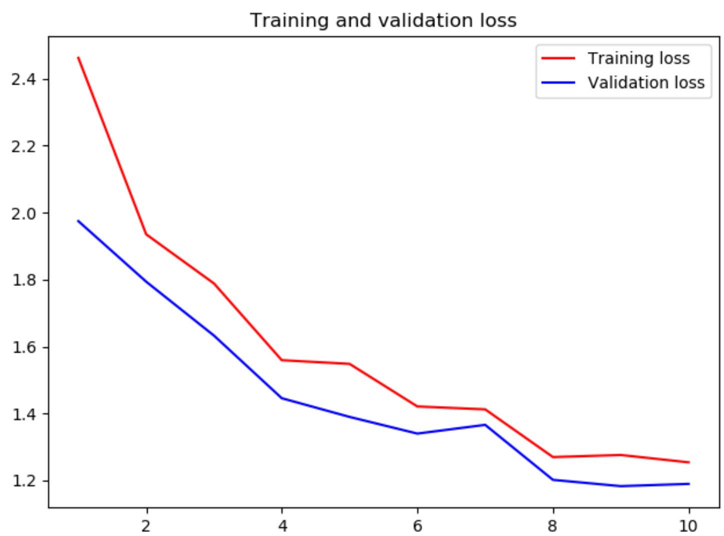
Although there were some improvements over accuracy the results were moderately satisfactory. The model achieved an accuracy of **~36%** on the validation set.

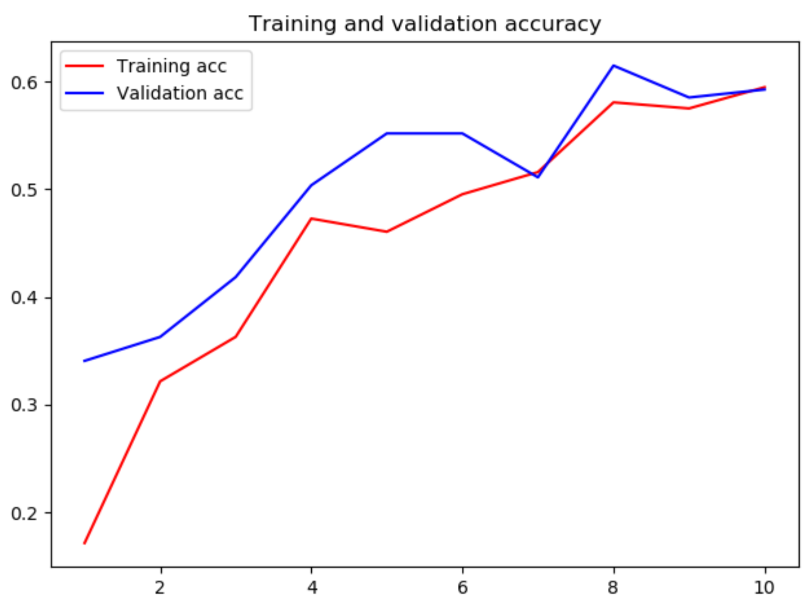
Transfer Learning-

A pretrained model built in keras based on vgg16 was used for classification by replacing its last layer with a fully connected layer for feature extraction.

The hyperparameters were kept same for consistency.

Validation-loss Curves-





The curves indicate that the pretrained model when used to train on a smaller dataset yielded better results and is a good fit.

The model achieved an accuracy of **~60%** on the validation set.

References-

1) https://keras.io/guides/sequential_model/
2) https://machinelearningmastery.com/5-step-life-cycle-neural-network-models-keras/
3) https://www.csie.ntu.edu.tw/~cjlin/libsvm/
4) Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on Computational learning theory (COLT '92). Association for Computing Machinery, New York, NY, USA, 144–152. DOI: https://doi.org/10.1145/130385.130401
5) https://www.kaggle.com/slothkong/10-monkey-species/home
6) http://yann.lecun.com/exdb/mnist/
7) Y. Lecun, L. Bottou, Y. Bengio, and P. Haner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, Nov 1998.
8) Nathan Halko, Per-Gunnar Martinsson, Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev., Survey and Review section, Vol. 53, num. 2, pp. 217-288, June 2011