CSIC 5011: Mini-Project 1: Performance Improvement of FNN based Image Classification using PCA

Neel Kanth KUNDU[1] and Avik Kumar Das[2] {nkkundu, akdas}@connect.ust.hk
[1]: Department of Electronic and Computer Engineering, HKUST [2]: Department of Civil Engineering, HKUST

1. Introduction

In the first part of the course, we have learnt PCA as a popular technique for dimensionality reduction. Most datasets contain correlated variables, hence PCA helps to extract the most important information from the data by finding new set of orthogonal variables called principal components. In this project we use PCA and perform Horn's parallel analysis on MNIST dataset to reduced the dimension of each image and then train a feedforward neural network (FNN) for image classification. The aim of the project is to reduce the number of parameters in the FNN and reduce the inference time of the FNN.

2. MNIST Dataset

This dataset contains gray-scale images of 10 hand-written digits namely 0,1...9. The images are 28x28 pixels i.e. p=784-dimensional inputs with n=70000 total images in the dataset. The data matrix X is of dimension $n \times p$. First the data matrix is normalized in the range of [0,1] by dividing it by 255 and further it is centered by subtracting the mean from X.

3. PCA

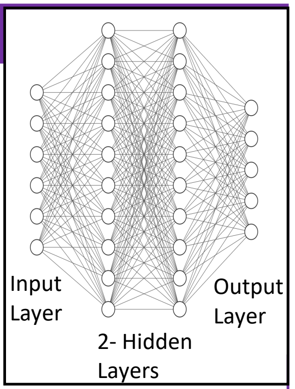
PCA can be carried out, either by using singular value decomposition (SVD) of the data matrix X or by using eigenvalue decomposition of the covariance matrix $\Sigma = \frac{1}{n}(X^T X)$. Here we use SVD to reduce the dimension of our data points. SVD of $X = U \Lambda V^T$, where U and V are orthonormal matrices of dimension $n \times p$ and $p \times p$ called the left and right singular matrices respectively. Λ is a diagonal matrix of dimension $p \times p$. The dimension of the data matrix can be reduced to $k < p$, by taking the first k columns of U and the upper-left $k \times k$ part of Λ. $X_k = U_k \Lambda_k$ is the transformed data matrix of dimension $n \times k$. The appropriate choice of k can be carried out by using Horn's parallel analysis which gives an empirical method to find the appropriate dimension of the transformed data matrix X_k.

4. Horn's Parallel Analysis

Horn's parallel analysis is based on random data simulation to determine the optimum number of dimension k. The data matrix is randomly permuted R times and for each permuted matrix Y^r , the eigenvalues of covariance matrix Σ_Y is calculated. For each eigenvalue $pval_i = \frac{\#\{\lambda_i(Y^r) > \lambda_i(X)\}}{R}$ is calculated which is regarded as the probability that λ_i is indistinguishable from noise. Smaller the p-value, the higher the confidence of λ_i belonging to the true data matrix X. We select the smallest 'k ' whose p-value is 1.

4. FNN Architecture

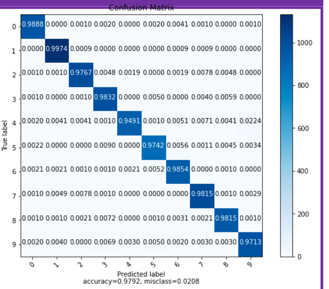
We use a FNN with 2 hidden layers each having N_h= 128 neurons and an output layer having 10 neurons corresponding to the 10 classes of MNIST as shown in the right figure. The activation function in the hidden units is ReLU while the output layer has softmax activation which gives the probability distribution over the 10 possible classes. The FNN trained by minimizing the categorical cross entropy loss function between the ground truth values and the predicted values by the FNN. We use Adam optimization algorithm which gives faster convergence.



5. Results and Analysis

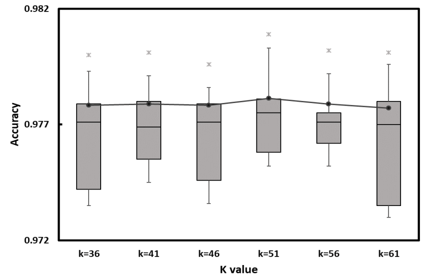
Method	No. of Parameters	Inference Time	Accuracy
FNN-Vanilla	~118k	440 ms	97.7%
FNN-PCA (k=51)	**~24k**	**372 ms**	**97.92%**

Confusion Matrix for FNN-PCA



	0	1	2	3	4	5	6	7	8	9
0	0.9888	0.0000	0.0010	0.0020	0.0000	0.0020	0.0041	0.0010	0.0000	0.0010
1	0.0000	0.9974	0.0009	0.0000	0.0000	0.0000	0.0009	0.0009	0.0000	0.0000
2	0.0010	0.0010	0.9767	0.0048	0.0019	0.0000	0.0019	0.0078	0.0048	0.0000
3	0.0010	0.0000	0.0010	0.9832	0.0000	0.0050	0.0000	0.0040	0.0059	0.0000
4	0.0020	0.0041	0.0041	0.0010	0.9491	0.0010	0.0051	0.0071	0.0041	0.0224
5	0.0022	0.0000	0.0000	0.0090	0.0000	0.9742	0.0056	0.0011	0.0045	0.0034
6	0.0021	0.0021	0.0010	0.0010	0.0021	0.0052	0.9854	0.0000	0.0010	0.0000
7	0.0010	0.0049	0.0078	0.0010	0.0000	0.0000	0.0000	0.9815	0.0010	0.0029
8	0.0010	0.0010	0.0021	0.0072	0.0000	0.0010	0.0031	0.0021	0.9815	0.0010
9	0.0020	0.0040	0.0000	0.0069	0.0030	0.0050	0.0020	0.0030	0.0030	0.9713

- By using dimensionality reduction technique on the input data, the number of parameters of the FNN are reduced by an order for the same depth and width of the FNN.
- PCA extracts the most important features using linear transformation from the 784-diemsional data. For a fixed width and depth of the FNN, the PCA-FNN gives better accuracy as compared to vanilla FNN. The inference time on the test dataset is also reduced by 68 ms.
- From Horn's parallel analysis we get k=46, we then experiment with k ranging from 36 to 51 to find the optimum k for the given FNN architecture.



This `box and whisker' plot shows statistical results of FNN-PCA for different choices of 'k'. The mean (black dot) and maximum (cross) for each each 'k' with 25 Monte-Carlo runs is shown. It shows that the mean inference accuracy for $k = 51$ is the highest.

- To achieve similar performance as FNN-PCA, the vanilla FNN would require more width and depth since the number of features of the input data is more hence a more complex network is required.

6. Conclusion and Future Work

In this project we used a linear dimensionality reduction technique called PCA on MNIST. The reduced dimension data is used in an FNN classifier. We see performance improvement of the FNN-PCA when compared to the vanilla FNN for a fixed width and depth. The inference time is also reduced due to reduced number of parameters in PCA-FNN.
In future work, the performance of CNN can be explored on the PCA reduced data.

7. References

- A. Gulli, A. Kapoor, and S. Pal, *Deep learning with TensorFlow 2 and Keras*. Birmingham: Packt Publishing Ltd, 2019.
- Yao, Yuan, "*A Mathematical Introduction to Data Science*", Lecture Notes (2019)

8. Contribution and GitHub link

- **Neel Kanth Kundu:** Problem Formulation, Implementation of PCA , Horn's Parallel Analysis, FNN and Report writing.
- **Avik Kumar Das:** Experimentation with PCA, FNN, Statistical Analysis Box Plot, Confusion matrix and Report writing.

GitHub: https://github.com/nkkundu/CSIC_5011_Project-1.git