The Effect of PCA Decomposition on Supervised Learning

Matthew Hoffman
Daniel Silver

April 24, 2019

1 Introduction

We propose that dimensionality reduction may assist in different supervised machine learning tasks by eliminating some of the noise in a dataset. Our project seeks to analyze the effect of different levels of PCA on a simple neural network based classification task by analyzing the trade-offs with reduced images and accuracy as a function of number of epochs. We attempt to have the best of both worlds by combining unsupervised learning approaches with supervised learning ones. We also seek to understand the relationship between entropy and classification accuracy.

2 Background

2.1 Dimensionality Reduction

The complexity of supervised learning training is greatly impacted by the amount of features in the data sets, considering the amount of random variables that need to be considered for each classification case. Additionally, high-dimensional data is susceptible to organization issues, such as memory or processing speed [1]. As such, dimensionality reduction aims to lower the amount of features used, to mitigate problems involved with storing and processing hundreds or thousands of features, at the cost of the information those features would provide.

2.2 Principal Component Analysis

Principal component analysis, or PCA, is a form of dimensionality reduction that maintains the features with the highest amount of variance, thereby maximizing variance by reducing to a lower-dimensional set of principal components, the features with the highest variance. High variance is desirable in features since classification depends on information that differentiates data, therefore performing dimensionality reduction by eliminating low variance features does little to interfere with the performance of classification.

In PCA, the total variance is is found by summing the variance of all features. A fraction of the total variance called the explained variance is selected, and the amount of features in the higher dimension set is reduced (in increasing order of individual variance) until the explained variance is satisfied.

2.3 Entropy

A measure of how one probability distribution is different from another is entropy. For our experiment, entropy is calculated using Kullback-Leibler divergence, defined as

$$D_{KL}(P||Q) = -\sum_{x \epsilon X} P(x) log(\frac{Q(x)}{P(x)}) \tag{1}$$

where P and Q are the original and reduced data sets, respectively [7]. Essentially, entropy here is the measure of lost information when Q is used to approximate P following principal component analysis.

2.4 Neural Networks

A neural network is a combination of perceptrons that seek to learn to approximate a function. According to the Universal Approximation Theorem [2], a neural network with only a single hidden layer can approximate any arbitrary continuous function. A neural network attempts to approximate a function, then iteratively iterates through the same network, attempting to minimize an objective function.

2.5 MNIST Dataset

The MNIST dataset is very commonly used to establish baselines for different machine learning algorithms. It contains 60,000 28x28 pixel images (in grayscale) of handwritten digits. This dataset is further split into 48,000 training images and corresponding labels and 12,000 test images and corresponding labels. There is roughly an even distribution of digits, all labeled. The labeling of these numbers allows for supervised learning such as classification tasks. This is the dataset we will be using to analyze our model. An example of a single MNIST image is in Figure 1.

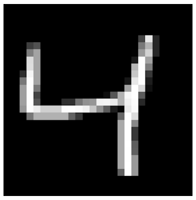


Figure 1: Example of an MNIST Image.

2.6 Related Works

In "The Role of Dimensionality Reduction in Classification" researchers were able to "train a combination of non-linear DR and a classifier, and apply it to a RBF mapping with a linear

SVM."[11] They used alternating steps to train a RBF mapping and a linear SVM, and were successful in allowing a user to select a trade-off between training accuracy and overall run time. They used multiple steps of training per iteration where they optimize one variable in their objective function at a time with alternating steps. This implementation does not use PCA and instead has the network learn its own dimensionality reduction. This is different from our implementations as we used PCA to perform our dimensionality reduction offline in a pre-processing network.

3 Our Model

3.1 Overview

Our model involves performing offline PCA at several levels of explained variance on the MNIST data set, before feeding the decomposed data sets into our neural network. By training the network against sets with different dimensions, we can analyze how the accuracy and loss are affected by the PCA, and conduct a cost benefit analysis for performing this sort of pre-processing technique. Figure 2 illustrates the overall structure of our project, specifically the inputs and outputs of different processes we perform.

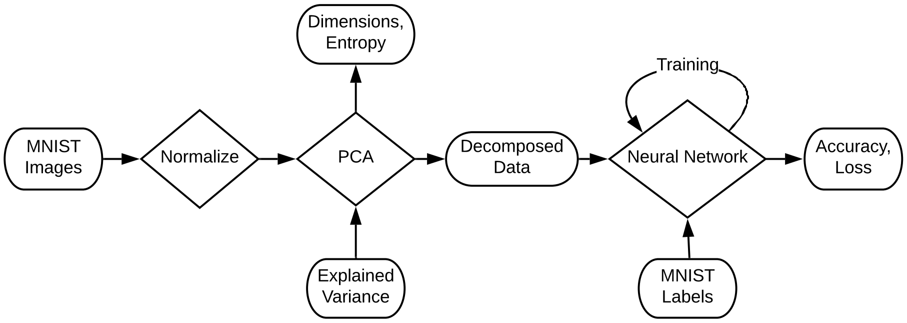


Figure 2: Illustration of the several components of our model.

3.2 Unsupervised Learning Component

The goal of our unsupervised learning component was to reduce the amount of dimensions while maintaining a specified level of variance within the set, called the explained variance.

Using a normal, vectorized data set (the MNIST database), we expect to perform PCA with several levels of variance, transforming the original set into lower dimensionality sets, while maintaining variance. We use scikit-learn's implementation of PCA in order to transform our original set according to the variances we provide [9].

3.3 Neural Network Architecture

In order to establish baselines, we used the same neural network architecture to asses the accuracy of our model for each different number of input dimensions, with the only difference being the number of input nodes to our network. We vectorize each input image before we pass it into our network. We use a fully connected network with 2 hidden layers each containing 256 nodes and a relu activation function. The relu activation function is defined as:

$$\text{relu(x)} = \max(x, 0). \tag{2}$$

We found relu activation to be the most consistent with training. We then have a final layer of size 10 that represents the network's guess as to what to classify the input image as. Each node at the final layer represents one digit [0-9] that the image can be. We use a softmax activation function on the final layer to smooth our and normalize the data. The softmax function is defined as:

$$\text{softmax(x)} = \frac{1}{1 + e^{-x}}. \tag{3}$$

We then calculate the cross entropy loss and use Adam Optimizer[6] to minimize our loss. We trained over 200 epochs. Our network is depicted in Figure 4. Our ultimate goal in training was to maximize the accuracy of our classification. We show in Figure 3 that by minimizing training loss, we maximize testing accuracy.

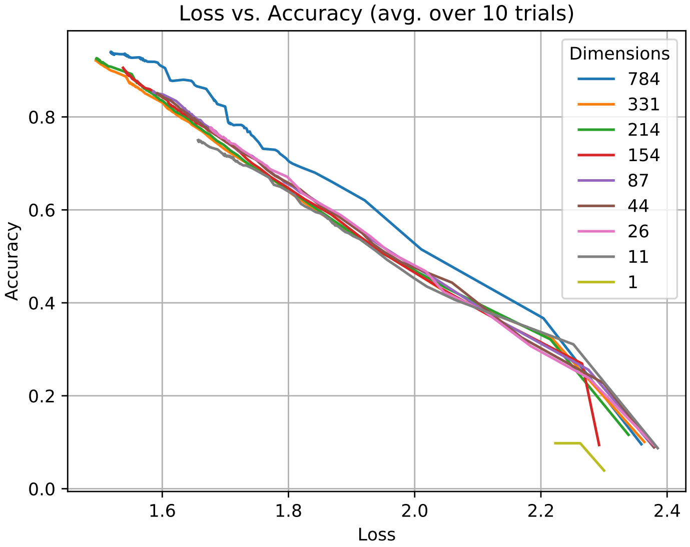


Figure 3: Graph of Loss of epoch vs Accuracy

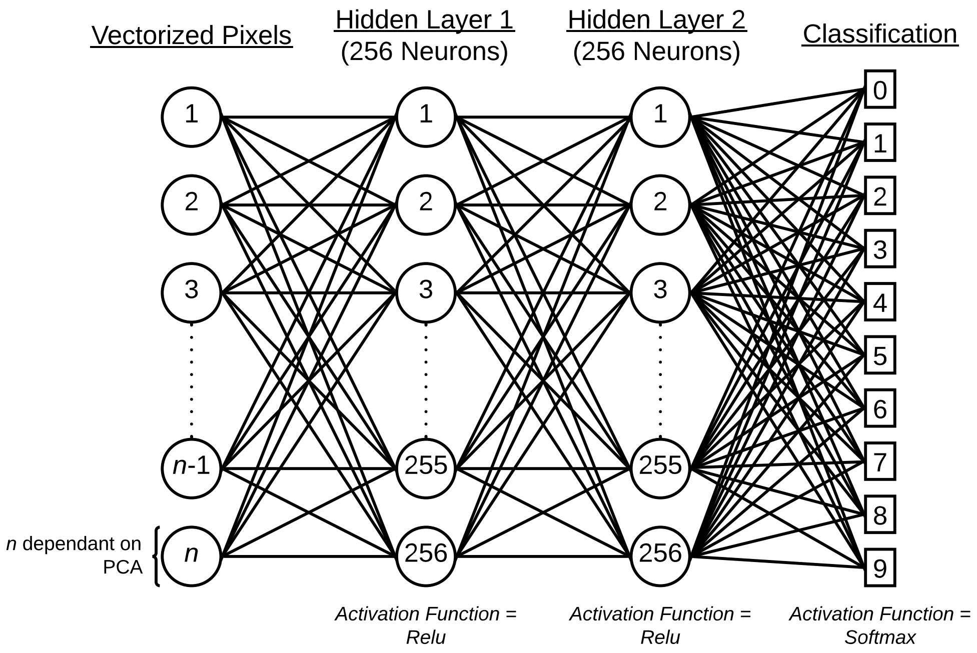


Figure 4: Neural network with 2 hidden layers containing 256 nodes at each layer

4 Experiments

4.1 Application of PCA

The validity of our PCA analysis depends on its ability to reduce the original sets dimensionality while maintaining features of high variance. Using the explained variances displayed in Figure 5 as the inputs for our PCA on the MNIST data sets, the original 784 features were reduced exponentially according to the variance selected, with the amount of features rapidly decreasing: at 99% explained variance, only 331 dimensions remain out of the original 784. In addition, PCA is meant to maximize entropy; Figure 6 shows a linear relationship between entropy of the sets and the explained variance used to produce them.

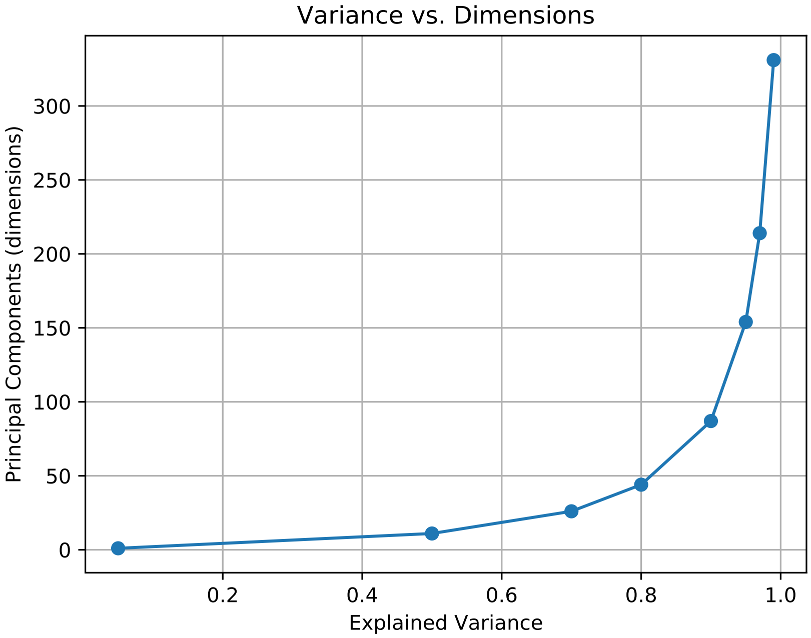


Figure 5: Relationship between explained variance and the principal components maintained during PCA

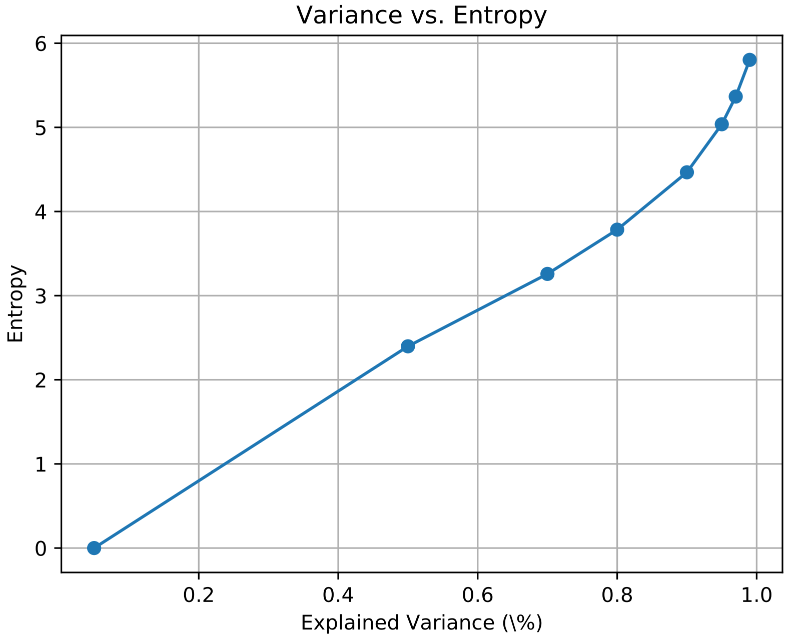


Figure 6: Relationship between explained variance and entropy of reduced sets following PCA

4.2 PCA effect on Supervised Learning

Through our experiments, we have concluded that on average, decreasing the number of dimensions of an input image/datapoint lowers the training accuracy of a supervised learning classification problem. This is supported by Figure 7.

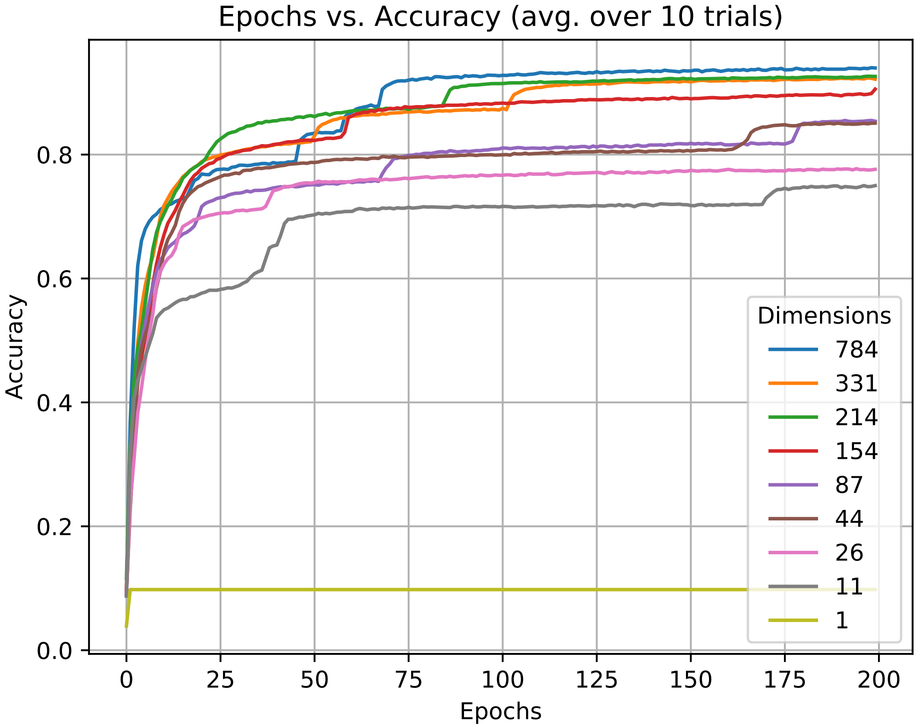


Figure 7: Accuracy vs. number of epochs in training for multiple different decompositions of MNIST images.

However, this drop in accuracy is very small compared to the dimensions lost in the image reduction. We believe this to be due to the entropy maximization and therefore image preserving nature of PCA. For example, when going from a full MNIST image of 784 pixels, to a smaller image of 331 pixels, 99% of the variance is preserved in the dataset. Intuitively, this can be explained by removing many unnecessary pixels that do not add much to the classification such as redundant black pixel border that accounts for roughly half of the image surrounding each digit in the MNIST dataset.

We found the relationship between accuracy and decomposed input image dimensions to be logarithmic as show in Figure 3. This shows that a large number of dimensions of an input image can be sacrificed for a small change in accuracy. For our particular experiment, where we testing the decomposition of MNIST images, we found a best fit representation of the relationship between accuracy of training at a set number of epochs to be represented as:

$$\text{y} = .1168 \ln x + .3027. \tag{4}$$

The values used to calculate this equation are show in Figure 8.

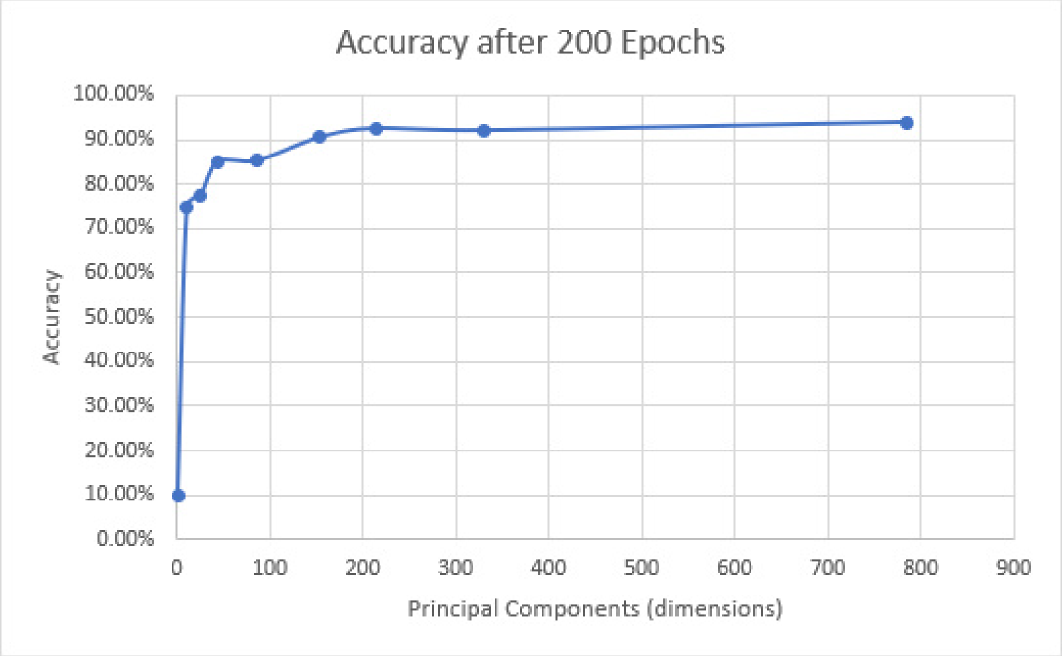
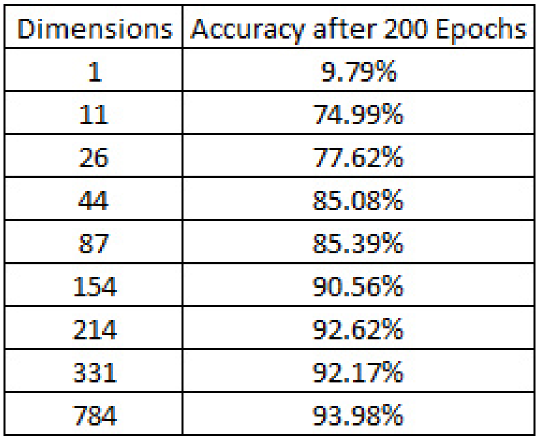


Figure 8: Graph of the relationship between accuracy after training and the number of dimensions of the decomposed input images



Dimensions	Accuracy after 200 Epochs
1	9.79%
11	74.99%
26	77.62%
44	85.08%
87	85.39%
154	90.56%
214	92.62%
331	92.17%
784	93.98%

Figure 9: Table of accuracy values for different levels of decomposition (Avg. over 10 trials)

5 Our contributions

To the best of our knowledge, the application of PCA as an offline pre-processing step has never been tested against the effectiveness of a supervised classification problem. Our contribution is this novel approach to finding a trade off between size of input and accuracy of an online neural network. We observed a logarithmic relationship between the accuracy of our online neural network and the number of dimensions in the decomposed inputs. This leads us to believe that there could be many benefits from trading off a small amount of accuracy for a large reduction in memory and speed cost of having a much larger dataset.

5.1 Implications for the research community

We believe our project can contribute to research on understanding the relationship between information theory and machine learning. We demonstrated a correlation between accuracy in classification in a supervised learning problem and the dimensionality reduction of the original dataset. We believe this can lead researchers to develop a rough estimate of the accuracy loss that will be incurred as a function of the variance reduced from the dimensionality reduction. This could lead to selecting what order dimensionality reduction should be used and this could be used to optimize computing resources to obtain the goal accuracy without over using computing resources.

5.2 Implications for industry

We believe the results of our project may help provide insight to industry leaders in machine learning. Our experiments lead us to believe that high values of accuracy, can be obtained with only a mere fraction of the dataset, if the dimensions are selected using PCA. An institution that may be running supervised machine learning algorithms over massive datasets may benefit from trading 1% accuracy for the cost of storing half of the data in each image. We believe this could have big implications on how large institutions store large amounts of data.

6 Conclusion/Future Work

Throughout our experiments, we found a link between the entropy of our reduced input images due to PCA and the eventual training accuracy of our model. As entropy and the number of dimensions decreases, so does the accuracy, but not by much. Almost the same exact accuracy results are achieved by lowering the number of input dimensions from 784 to 331. As there are many problems with having a larger dimensionality space, see The Curse of Dimensionality [1], this may be a beneficial technique to simplify many advanced supervised learning problems in machine learning.

References

[1] Richard Ernest Bellman. *Dynamic Programming.* New York, NY, USA: Dover Publications, Inc., 2003. ISBN: 0486428095.

[2] B.C. Csaji. “Approximation with Artificial Neural Networks”. In: *M.S.’Thesis, Dept. Science, Eotvos Lorand Univ., Budapest, Hungary* (2001). URL: https://ci.nii.ac.jp/naid/20001716508/en/.

[3] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository.* 2017. URL: http://archive.ics.uci.edu/ml.

[4] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (Second Edition).* New York, NY, USA: Wiley-Interscience, 2000. ISBN: 0471056693.

[5] Shuiwang Ji and Jieping Ye. “Linear Dimensionality Reduction for Multi-label Classification”. In: Pasadena, California, US: Twenty-First International Joint Conference on Artificial Intelligence, 2009.

[6] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization.* cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. 2014. URL: http://arxiv.org/abs/1412.6980.

[7] S. Kullback and R. A. Leibler. “On Information and Sufficiency”. In: *Ann. Math. Statist.* 22.1 (Mar. 1951), pp. 79–86. DOI: 10.1214/aoms/1177729694. URL: https://doi.org/10.1214/aoms/1177729694.

[8] Oky Dwi Nurhayati. “Principal Component Analysis with Mean and Entropy Values for Thermal Images Classification”. In: *International Journal of Computer Science and Telecommunications* 3.3 (Mar. 2012). ISSN: 0976-8491 (Online) — 2229-4333 (Print).

[9] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[10] Mark Richardson. “Principal Component Analysis”. In: (2009).

[11] Weiran Wang and Miguel Á. Carreira-Perpiñán. “The Role of Dimensionality Reduction in Classification”. In: Québec City, Québec, Canada: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014.

[12] Adrian Mondry Xiaoxing Liu Arun Krishnan. “An Entropy-based gene selection method for cancer classification using microarray data”. In: 6.76 (2005). ISSN: 1471-2105.