Project 2: A Comparative Study Between Traditional and Information Theoretic Neural Network Models for Fashion-MNIST Classification

Connor McCurley
Deep Learning, Fall 2019
University of Florida
Gainesville, FL, USA 32611
Email: cmccurley@ufl.edu

***Abstract*—**
***Index Terms*—Fashion MNIST, Autoencoder, Information Theoretic Learning, Support Vector Machine**

I. Introduction

II. Methodology

This section describes the methodology implemented in this work. Analysis of the data is performed, dimensionality reduction and classification procedures are described, various network architectures under analysis are elaborated on and experimental procedures are outlined.

A. Data Analysis

The following data analysis was originally described in [1], but is pertinent to this work and is thus re-analyzed here. The data was plotted as shown in Figure 1 to gain an understanding of the format. Each sample in the Fashion-MNIST dataset is a 28x28, gray-scale image of a clothing item belonging to one of ten classes [2]. This translates to 784 length feature vectors with values ranging between 0-255. There were exactly 60000 training images included in the training dataset and 10000 which were held-out for test. The 60000 samples were later sub-divided in the experimentation for cross-validation. As with Project 1, dimensionality reduction(DR) was incorporated to reduce data complexity. This DR was employed through the use of stacked autocoder neural networks (SAE). Elaboration on the SAE networks is provided in the following section.

B. Autoencoder

Description: An autoencoder is a specific taxonomy of artificial neural network which learns how to compress and de-compress representations of data [3], [4]. The first half of an autoencoder typically performs dimensionality reduction through non-linear transformations until the middle layer, known as the *bottleneck* or *latent* layer. The goal of the *encoder* is to learn efficient representations of the factors which govern variation in the data, or in terms of compression, *codes* which can be used to reconstruct the input data with high accuracy. The second half of an SAE (which is called the *decoder*) projects the data back into its original dimensionality in attempt to reconstruct the original sample.(See Figure 2.) Reconstruction loss is between the input and output is used to update the network's parameters. In practice, samples can be passed through the encoder to perform dimensionality reduction.

SAE Architecture: The SAE architecture tested in this work consisted of 5 hidden layers, along with the input and output. The layers were selected as $784 \rightarrow 500 \rightarrow 200 \rightarrow k \rightarrow 200 \rightarrow 500 \rightarrow 784$, where k is the arbitrarily chosen dimensionality of the bottleneck. In this work, k was tested at $[10, 25, 50, 75, 100]$ in order to provided a reasonable comparison of performance changes resulting from dimensionality reduction. ReLU activation functions were used to apply nonlinearity. A sigmoid activation, however, was used at the output layer to enforce image value constraints between $[0-1]$. This was done because the images were normalized between $[0-1]$ before passing through the network. An initial learning rate of $\eta = 0.01$ was selected, and was updated using the Adamax optimizer through training.

SAE Experiments: The SAE network was trained for 20 epochs using mini-batch sizes of 200 samples. The bottleneck layer's dimensionality was varied between $[10, 25, 50, 75, 100]$. Each network configuration was trained 5 times, and the model which provided the lowest reconstruction Mean-Squared Error (MSE) on the hold-out validation set was selected for further use in classification. Results are shown in section III-A.

C. Support Vector Machines

Description: A Support Vector Machine (SVM) is a specific class of sparse kernel machines whose objective is to learn a decision boundary which can adequately discriminate between classes in a high-dimensional space [5], [6]. Because of its sparsity constraints, a SVMs' predictions rely only on a subset of the training data known as *support vectors*. By design, support vectors tend to be examples in the training data

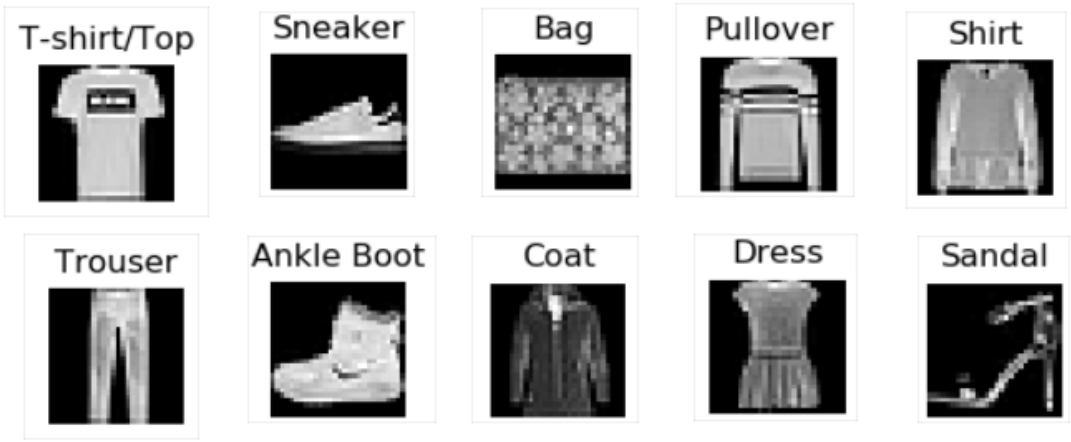


Fig. 1: Samples from the Fashion-MNIST dataset. One sample from each class was randomly chosen for visualization. The gray-scale images are size 28x28, each representing an article of clothing.

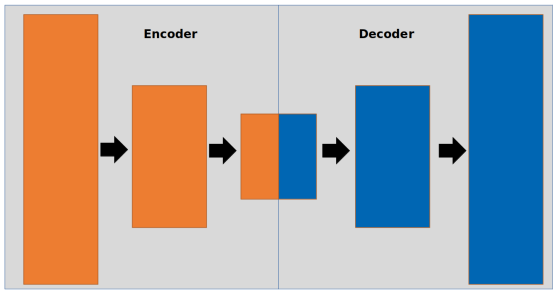
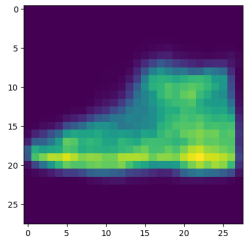
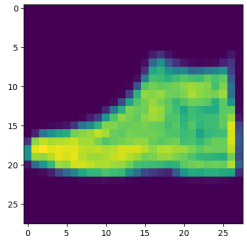


Fig. 2: Block diagram of an autoencoder neural network. The layers consecutively reduce dimensionality until the middle (bottleneck) layer. The second half of the network transforms the data back to the size of the input. The desired value of the network is the original image.



(a)



(b)

Fig. 3: Reconstructed images of a shoe after passing through an SAE with bottleneck dimensionality 10 (a) and 100 (b). The images' original dimensionality was 784.

which lie closest to the decision boundary, and are thus the most prone to mis-classification. A primary difference between SVM and methods relying on Information Theory is that vanilla SVM classification predictions are not probabilistic [7]. In other words, hard labels are assigned which do not capture the uncertainty of the prediction results. In this work, SVMs were trained on the data passed through the selected autoencoders. A comparison of classification performance on the various sizes of features is provided in section III-C.

Parameters: Non-linear support vector machines were used as the classifiers in this work. The necessary hyperparameters were the type of mapping kernel, kernel parameters, and a regularization (slack) parameter. A radial-basis function was arbitrarily chosen as the mapping function. Silverman's rule was used to provided a reasonable range for the kernel bandwidth. The regularization parameter was set to the Python Scikitlearn's default value of $C = 1$. These parameter choices provided reasonable results for comparison. More parameter variations for the SVM were not tested due to time limitations.

Experiments: SVMs were employed using Scikitlearn's SVM SVC package. The classifiers were trained using one-versus-one training. At test, a sample was applied to the classifier ensemble and the most-likely label was provided to the sample. Data was passed through each of the top 5 trained autoencoders, and a single SVM was trained to provide label predictions for each input feature dimensionality, $[10, 25, 50, 75, 100]$. Classification performance is presented in section III-C.

D. Baseline CNN

Description: The autoencoder + SVM classifiers were compared against a baseline convolutional neural network (CNN)

architecture. A CNN is a special type of neural network which utilizes weight sharing. A suit of kernels is convolved across the input feature map to detect interesting attributes in an image [3], [4]. CNNs have shown considerable success in a variety of image classification problems. For this reason, a CNN architecture is used as a standard for comparison. Classification performance of the baseline architecture is provided in section III-C.

CNN Architecture: The baseline CNN architecture was chosen based off previous performance on Fashion-MNIST classification [2]. The network consists of two convolutional/ max-pooling layers followed by two fully connected layers. The network provides an output size of 10. This is passed into a soft-max activation function to provide class probability scores. ReLUs are used to provide nonlinearity in each of the hidden layers.

CNN Experiments: The baseline CNN was initialized with a learning rate of $\eta = 0.01$. This was refined during training with the Adamax optimizer. The model was trained 5 times for 20 epochs each. Mini-batches of size 200 were implemented and class prediction scores were evaluated using cross-entropy loss. The model with the best classification accuracy on the hold-out validation set was selected for comparison against the alternative methods in this work.

E. Information Theoretic Learning

The methods previously described are what the author considers as "traditional" forms of learning. In this sense, these methods typically rely on low-order statistics to capture prediction error. Alternatively, Information Theoretic Learning (ITL) approaches have proven effective in a variety of learning applications [8]. These methods take advantage of class distributions to compare higher-order statistics and thus provide richer descriptions of classification error.

Minimum Cross-Entropy (xEnt): One such information-theoretic metric is Minimum Cross-Entropy (MinxEnt). As outlined by Principe in [8], MinxEnt is formulated as a constrained minimization of KL-divergence between the label and predicted-value distributions, p_d and p_y.

$$\min_{p_y} D_{KL}(p_d||p_y) \quad \text{subject to (constraints)} \tag{1}$$

To implement this comparison, a small amount of Gaussian noise is added to the one-hot encoded label vectors:

$$d_{new} = d + \mathcal{N}(0, \sigma^2\mathbf{I}) \tag{2}$$

where $d \in \mathbb{R}^{10\times 1}$, and σ^2 was arbitrarily chosen as 0.002.

ITL Network Architecture: In order to provide a comparison to the alternative methods, multi-layer perceptron networks were implemented and trained with MinxEnt. The network architectures were designed as $784 \to 500 \to 200 \to k \to 10$ which is consistent with the autoencoder networks defined earlier in this work. The value of k was selected to match the bottleneck sizes of the tested autoencoders and an output layer of size 10 was employed to allow label comparison. ReLU activations were utilized in each of the hidden layers and a softmax was applied at the output to produce class probability vectors.

ITL Experiments: As with the previous experiments, networks were trained 5 times each with an initial learning rate of $\eta = 0.01$ and updated with Adamax. For each of the sizes for k, $[10, 25, 50, 75, 100]$, models were trained using a range of bandwidth parameters on the parzen window estimator for xEnt. Specifically, bandwidths of $\sigma = [3, 30, 300, 300]$ were implemented to allow for broad generalization of effects on classification performance.

F. Experiments

Experiments

III. Results

A. Autoencoder Reconstruction

B. Confusion Matrices

In this section, classification results for each experimental method tested are presented in the form of confusion matrices. A confusion matrix demonstrates the discrepancies between predicted and true class values for groups of samples. Essentially, it is a way to measure how accurate a classifier is, while providing insight into how the network confuses samples. A diagonal matrix signifies zero mis-classifications among all categories.

C. Comparison of Cost Functions

TABLE I: Classification Accuracies for Different Neural Model/ Classification Systems

Classification Model	Accuracy
Baseline CNN	0.90
SVM 100D	0.86
SVM 75D	0.85
SVM 50D	0.84
SVM 25D	0.81
SVM 10D	0.76

D. Comparison of XEnt Kernel Bandwidths

TABLE II: Classification Accuracies for Different XEnt Kernel Bandwidths

Bottleneck Size	Bandwidth	Accuracy
Baseline CNN	0.90	
SVM 100D	0.86	
SVM 75D	0.85	
SVM 50D	0.84	
SVM 25D	0.81	
SVM 10D	0.76	

IV. Discussion

In this sections, observations are made on results and insight is given to potential influences.

A. Results

B. Potential Improvements

In

V. Conclusions

Three

Future research endeavors toward this topic include,

Honor Statement

* I confirm that this assignment is my own work, it is not copied from any other person's work (published or unpublished), and has not been previously submitted for assessment either at University of Florida or elsewhere.

References

[1] C. H. McCurley, "Project 1: Manifold learning for fashion-mnist classification with multi-layer perceptrons," 2019.

[2] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

[3] S. S. Haykin, *Neural networks and learning machines*, 3rd ed. Upper Saddle River, NJ: Pearson Education, 2009.

[4] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[5] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang, "Large-scale image classification: Fast feature extraction and svm training," in *CVPR 2011*, June 2011, pp. 1689–1696.

[6] J. Sanchez and F. Perronnin, "High-dimensional signature compression for large-scale image classification," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1665–1672.

[7] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[8] J. C. Principe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*, 1st ed. Springer Publishing Company, Incorporated, 2010.