Implementing Neural Networks from Scratch as a Means of Achieving High Accuracy

Lucas Nguyen, Paul Pan*
Department of Computer Science and Engineering
University of California - San Diego
La Jolla, CA 92093
lnguyen.professional@gmail.com, jpan@ucsd.edu

Abstract

Many introductory neural networks leverage libraries without regard to the math behind the models. Here we show that it is possible to create models the perform well from scratch given appropriate preprocessing. We apply principle componenent analysis and create two classification models, Logistic and Softmax Regression from scratch using NumPy. We then train both models on the Fashion MNIST Dataset, achieving accuracies of 99%, 85% and 84% for all three tasks measured.

1 Introduction

In this paper we implemented two neural networks from scratch for static image classification: logistic and softmax regression. The former is used for binary classification, the latter for multi-class classification. We train and evaluate the models on three tasks based on the popular machine learning benchmark, Fashion MNIST. Fashion MNIST has various images of clothing described later in Dataset. We distinguish between items for two different sets, first ankle boots and t-shirts then pullovers and coats. Finally we open to a multi-class task classifying all 10 classes from Fashion MNIST. To maximize accuracy, we first project the image data onto the most important pixels using principle component analysis. We encode the data appropriately for each task, set aside a holdout set, then perform k-fold cross validation with mini-batch learning, or learning on a subset of the training examples. During training we record the loss of train and validation sets for each epoch, averaging over fold and plotting losses after training. Finally we evaluate on the test set, achieving higher than expected accuracy of 99%, 85%, and 84% for the three tasks.

2 Related Work

We used lecture 2 primarily for logistic and softmax regression, including delta rules, activation functions, and loss functions, although we also referred to discussion 2, Bishop's Neural Network for Pattern Recognition, and the Coursera Machine Learning Course. For PCA we used discussion 1.

3 Dataset

The Fashion MNIST Dataset is a benchmark dataset containing 60,000 28 x 28 grayscale images of 10 different clothing items displayed below. We normalized all of the training data. For logistic regression, we indexed the target labels for classes (0, 9) and (4, 2) corresponding to T-shirt, Ankle Boot and Coat, Pullover respectively, then encoded 1s for the first and 0s for the second class. For

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

softmax regression, we one-hot encoded all of the targets so the resulting label matrix had dimension (10, N). We performed k-fold cross validation with k = 10, thus with 60000 training examples, we have 6000 examples as the validation set per fold and the rest as training.

4 PCA

4.1 Figure 4.1: decompose -> restructure

This is the set of figures for images reconstructed from 2, 10, 50, 100, 200, and 784 principle components. Also, the first image on the left is the original image. The reason why I am seeing increasingly less blurry images is because PCA reduces the dimensionality of the images, meaning that it is trying to represent the images with less pixels since each pixel represents a dimension in the data. With more components, more dimensions are used to represent the original image.

4.2 Figure 4.2: Ankle Boots and T-shirts

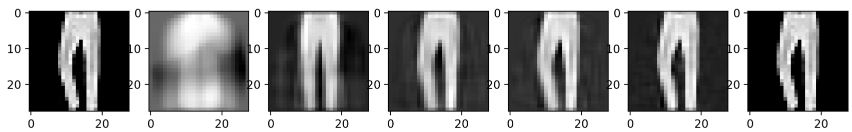
For the ankle boots and t-shirts, the images seem to be a blend of both the ankle boots and the t-shirt. Since these are the principle components of both of them, it makes sense that they are shown this way.

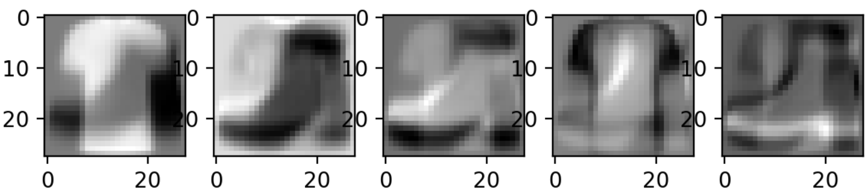
4.3 Figure 4.3: Coats and Pullovers

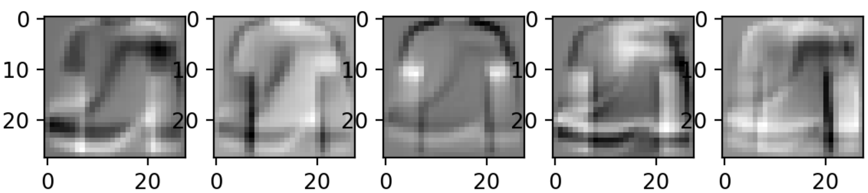
For the pullover and coats, there seems to be less of a blend but rather a type of clothing that is pretty distinct. The reason why it is not like the components groups in the previous set is because coats and pullovers have similar shapes, and they tend to complement each other.

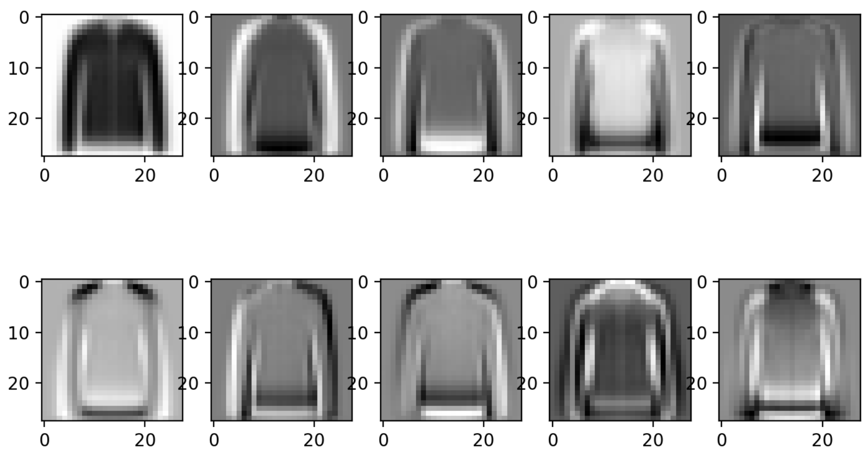
4.4 Figure 4.4: All classes

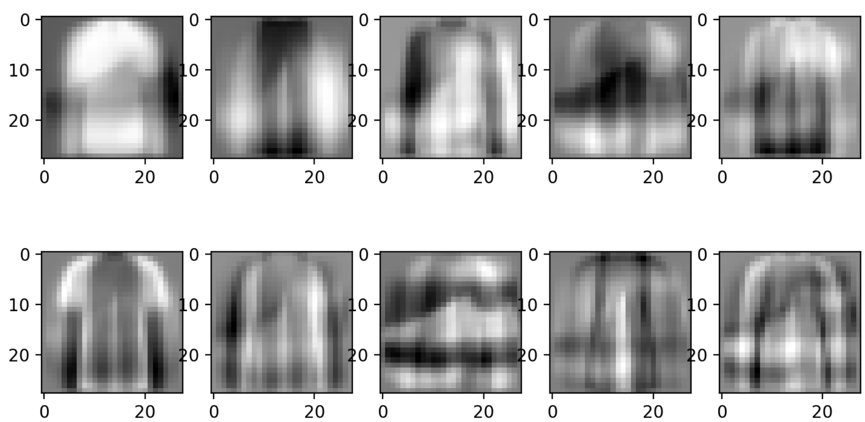
For the top 10 principle components of all classes, the images seem to be a blend of everything put together, which makes sense since those principle components are reduced from all images. There doesn't seem to be any pattern unlike the previous two groups.





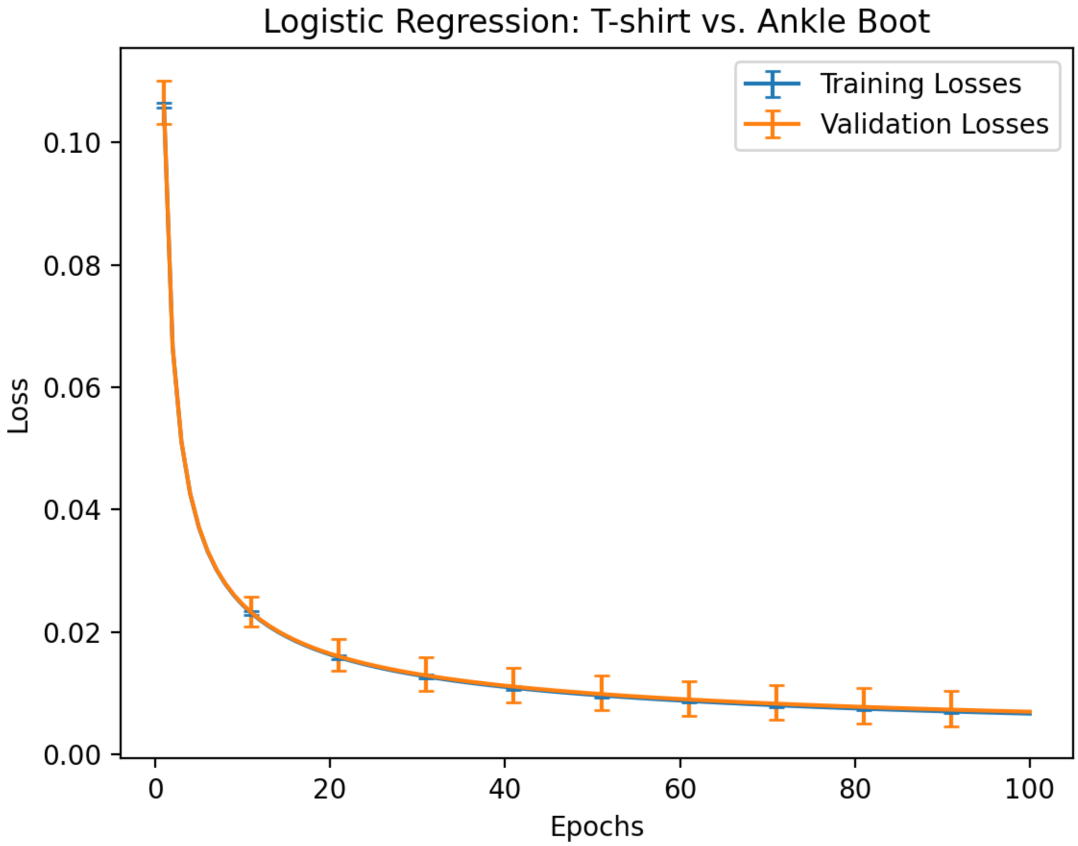


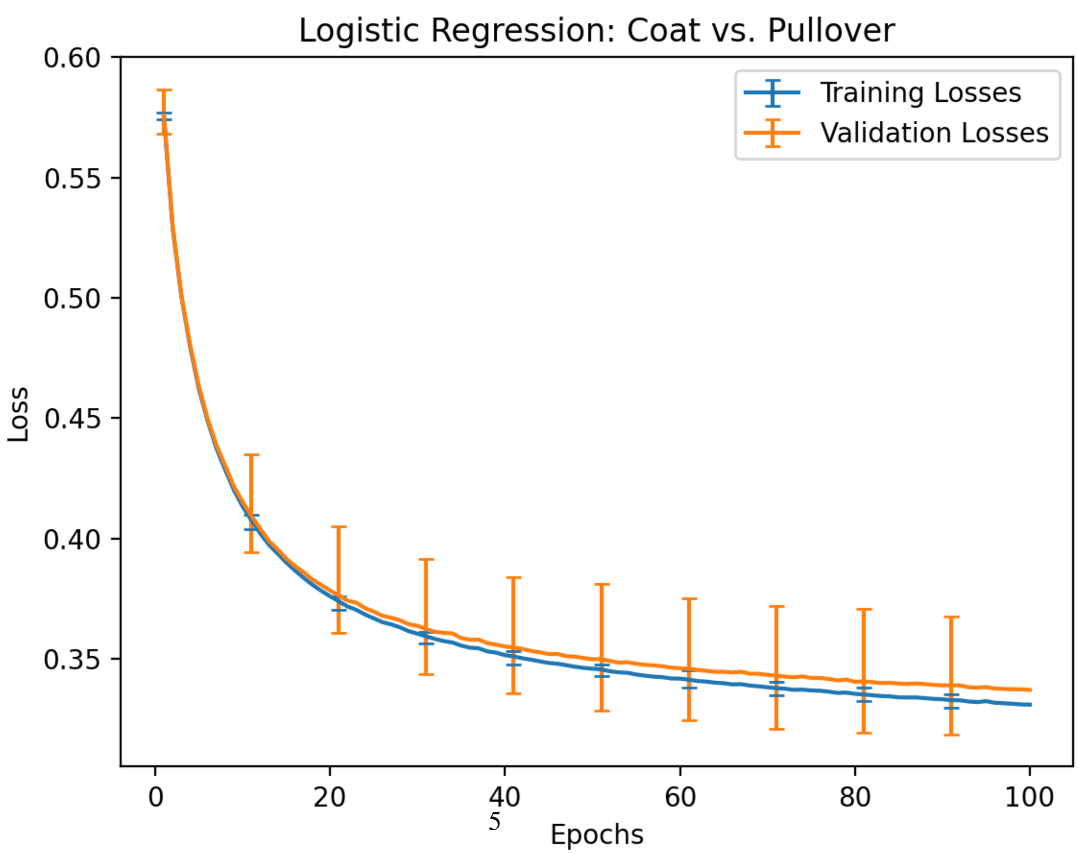




5 Logistic Regression

Logistic regression is a binary classification model which uses the sigmoid function to predict the probability of an input fitting or not fitting a class. Note that as the input approaches infinity/- infinity, the output approaches 1/0. We use logistic regression to classify between T-shirt vs Ankle Boot and Coat vs Pullover.


Logistic Regression: T-shirt vs. Ankle Boot
Training Losses
Validation Losses
Loss
Epochs
0.00
0.02
0.04
0.06
0.08
0.10
0
20
40
60
80
100


Logistic Regression: Coat vs. Pullover
Training Losses
Validation Losses
Loss
Epochs
0.35
0.40
0.45
0.50
0.55
0.60
0
20
40
60
80
100

During our first time training the model our model using a learning rate of 0.01 returned good loss curves and accuracy for both sets. However, we tried the following learning rates: 0.01, 0.1, 0.03, 0.075, 0.05 to see if we could improve our model, and we ended up with 0.03. We decided not to change PC components and batch size since our model performed significantly better than the corresponding homework accuracies. Upon finally evaluating the test set for both models we got 1999/2000 = 99.95% and 1713/2000 = 85.65%. This is interesting because both are approximately 15% higher than suggested accuracies with little hyperparameter tuning. It is also interesting that we achieved these results with such a small dataset. We believe the second set was more difficult to classify than the first because the two clothing types are more similar; pullovers and coats have the same outline, while ankle boots resemble footwear unlike t-shirts.

6 Softmax Regression

Softmax regression is a generalization of logistic regression to predict multiple classes. It uses the softmax function, a generalization of the sigmoid function to output probabilities of multiple classes. We use softmax regression to classify between all 10 of the clothing types found in Fashion-MNIST.

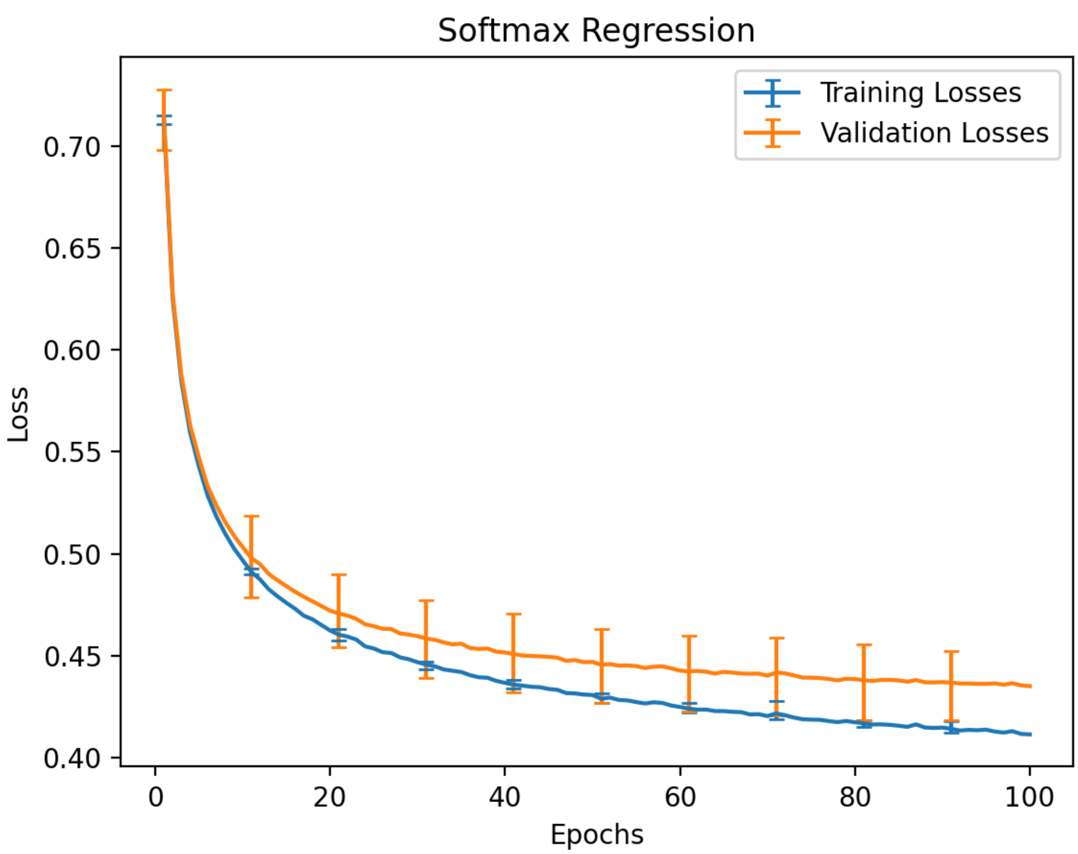
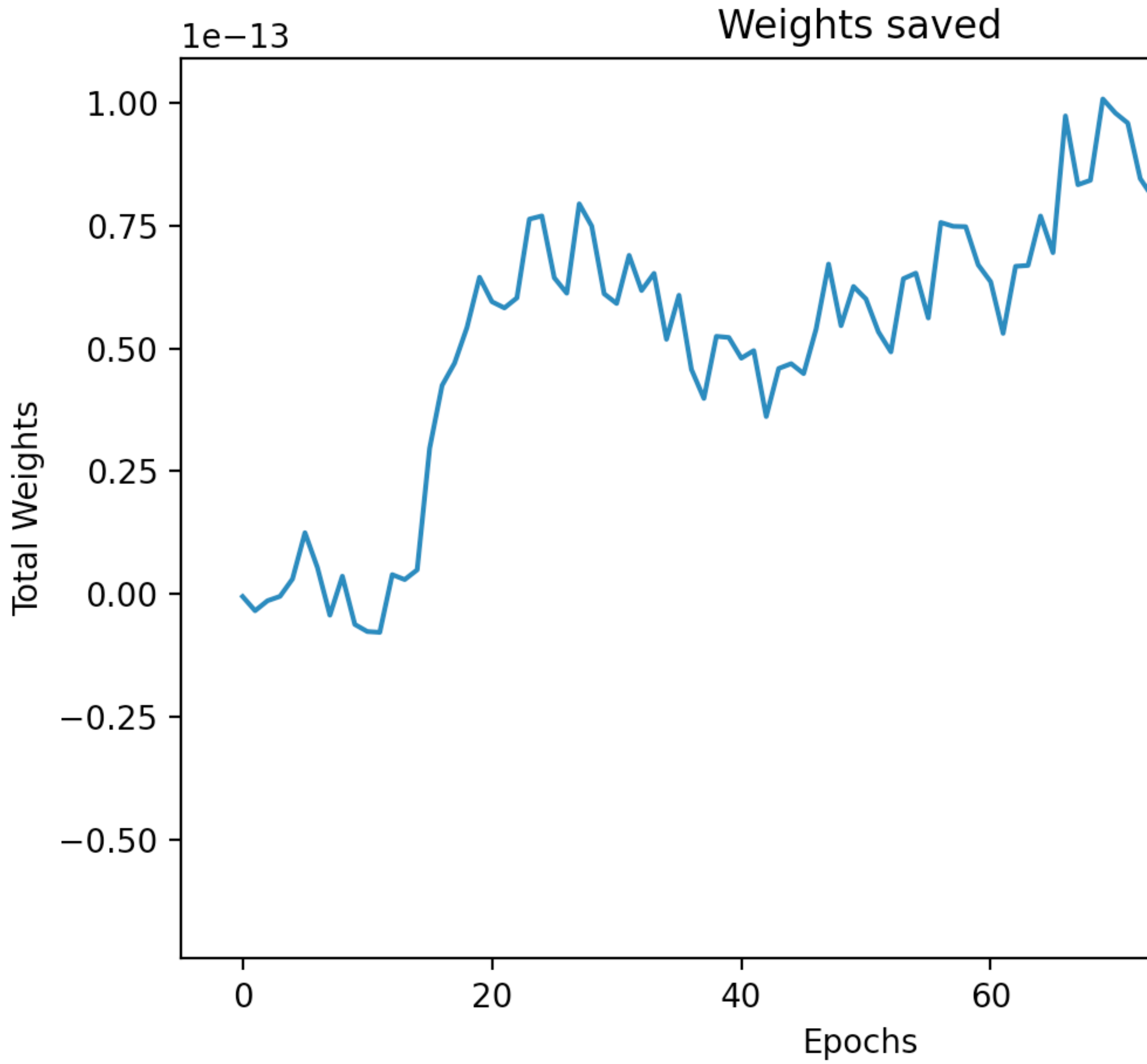


Figure 6.1: Softmax Regression

Like logistic regression, our first model did well with hyperparameters batch size of 512, learning rate of 0.01, and top 500 principle components. We also tested learning rates of 0.05, and 0.1 and found that while validation loss strayed more from training loss, the accuracy increased by a few percent. When we evaluated the test set we got an accuracy of 84.06%



7 Contributions

Lucas: I coded the frame (folds, epochs, batches) for gradient descent, the train and validation loss plotting, both loss functions, and the accuracy measurement. I also wrote the majority of the Latex file.

Paul: I wrote most of and converted to class the Softmax and logistic regression functions. I also did PCA and preprocessed/one=hot encoded images for logistic and softmax regression.

We both worked on cross validation at the beginning, and helped eachother debug