Logistic and Softmax Regression to Predict Vehicle Types from Images

Anshuman Dewangan
A59001372
adewanga@ucsd.edu

Margot Wagner
A53279875
mwagner@ucsd.edu

Abstract

We trained logistic and softmax regression models to predict vehicle types from 200x300 pixel images from the Comprehensive Cars dataset. We achieved 60% test accuracy classifying Convertible and Minivan vehicle types on the resized dataset. In our second model, we achieved 76.4% average test accuracy classifying Convertible and Minivan vehicle types on the aligned dataset. In our third model, we achieved 80.0% average test accuracy classifying Sedan and Pickup vehicle types. Lastly, we achieved 57.96% average test accuracy classifying all four vehicle types using multiclass softmax regression. We observed that model performance is related to the visual similarity of vehicle types in the prediction task. We also explored the effect of varying the number of principal components, learning rate, and using batch vs. gradient descent throughout our analyses.

1 Introduction

In this work, we explored machine learning methods to predict vehicle types from images in The Comprehensive Cars dataset with the general hypothesis that the model can learn visual characteristics representative of each vehicle type. All data was preprocessed by applying flattening the images into 1-dimensional vectors. Principal Component Analysis (PCA) was applied to further reduce the dimensionality of the data while maintaining the highest variance explained by the data. The testing procedure leveraged 10-fold cross-validation, with the average test loss and test accuracy reported across all folds.

First, logistic regression was used for the following binary classification tasks: Convertible vs. Minivan and Sedan vs. Pickup. For the Convertible vs. Minivan classification task, we explored the impact of aligning the images to make the relative size of the car the same. We then compared the difference in model performance between the Convertible vs. Minivan classification task and the Sedan vs. Pickup classification task with the hypothesis that one task will be easier than the other due to the stark visual differences between the target vehicle types.

Next, Softmax regression was used for classification among all vehicle types: Convertible, Minivan, Sedan, and Pickup. We see that the increase in the number of classes that we are trying to predict decreases the model's overall accuracy. Finally, we observed little difference in model performance between batch vs. stochastic implementations of gradient descent.

2 Methods

Images were loaded and preprocessed from the CompCars dataset. Each image, originally 200x300 pixels, was flattened to a 1D vector of pixels of length 60,000. The images for each category were then stacked into a data matrix with dimensions of the number of examples by the flattened number of pixels. Subsequently, PCA was applied to reduce the number of dimensions resulting in dimensions of number of examples by number of components. Unless otherwise specified, a 10-fold

cross-validation procedure with an 80/10/10 train, validation, and test set split was utilized to train the models. Further implementation details vary depending on the regression method.

2.1 Logistic Regression

Logistic regression is a binary classification strategy, so each class was assigned a one or zero depending on the output label as ground truth. We used the logistic regression model parameterized by weights, w, a 1D vector with length as the number of components.

$$y = P(C_1|x) = \frac{1}{1 + exp(-w^T x)} = g(w^T x) \tag{1}$$

$$P(C_0|x) = 1 - P(C_1|x) = 1 - y \tag{2}$$

We used the 80% training data split to train the model weights using gradient descent where the goal was to minimize the error with each iteration (or epoch). The error used was the cross-entropy loss defined by:

$$E(w) = -\sum_{n=1}^{N} \{t^n ln(y^n) + (1 - t^n) ln(1 - y^n)\} \tag{3}$$

Thus, the update rule for gradient descent in our model was:

$$w_{new} = w_{old} - \alpha \left(\frac{\partial E(w)}{\partial w_j} \right) = w_{old} - \alpha \sum_{n=1}^{N} (t^n - y^n) x_j^n \tag{4}$$

where α was the model learning rate. With each iteration, the cross-entropy loss was recorded as well as the accuracy (defined as the sum of the number of correct predictions between the predicted labels and the actual labels normalized by the number of examples). The cross-entropy loss was also normalized by the number of examples as well as the number of categories, which is 2 for this binary classification task. The algorithm was run for 300 epochs with performance metrics calculated at each iteration of the gradient descent algorithm for both the training and the validation sets. For trials with 10-fold cross validation, the loss and accuracy were averaged over the 10 runs. Leveraging early stopping, the iteration resulting in weights of the model with the lowest validation loss were then used to measure performance on the test set. The test set performance was also averaged over the number of folds.

We first evaluated the model on classifying between Convertible and Minivan images using the resized dataset. This provided a way to highlight the importance of alignment as a preprocessing step. There were 149 images of Convertibles and 148 images of Minivans used in total, representing relatively balanced classes. This evaluation was performed with no cross-validation, but the effect of number of principal components (PCs) and learning rate on the training loss was analyzed. Next, we used the logistic regression model to again classify Convertible vs. Minivan images but used the aligned image dataset. This model was performed with 10-fold cross-validation, including an analysis of the effect of the number of PCs and the learning rate. Lastly, we repeated the experiment but looked to classify Sedan vs. Pickup using the aligned dataset. We used 150 images of each class. This allows for analysis of how well the model functions in different types of binary classification tasks.

2.2 Softmax Regression

Softmax regression is a multiclass generalization of the binary logistic regression model, where the model is instead given by

$$y_k^m = \frac{exp(w_k^T x^n)}{\sum_{k'} exp(w_{k'}^T x^n)} \tag{5}$$

Similarly, we used cross-entropy loss to quantify error using the following formula

$$E = -\sum_{n} \sum_{k=1}^{c} t_k^n ln y_k^n \tag{6}$$

Loss was normalized by the number of examples and the number of categories. Again, accuracy was also measured as the sum of correct predictions normalized by the number of examples. We used the model to classify between all four car types: Convertible, Minivan, Pickup and Sedan. We initially trained the model using batch gradient descent with 10-fold cross validation and early stopping, similar to logistic regression. This model used the following weight updating scheme:

$$w_{jk,new} = w_{jk,old} - \alpha \left(\frac{\partial E(w)}{\partial w_j k} \right) = w_{jk,old} - \alpha \sum_{n=1}^{N} (y_k^n - t_k^n) x_j^n \tag{7}$$

We also trained the model using stochastic gradient descent as a comparison, where the weight was updated within each epoch using each example individual with the update order determined randomly. Predictions were made by selecting the output with the highest probability.

3 Results & Discussion

3.1 Logistic Regression

3.1.1 Resized Convertible vs. Minivan model

The model trained to distinguish between Convertible and Minivan images using the just resized dataset without any alignment and no cross-validation produced a test loss of 0.33 and test accuracy of 60.0% using 10 PCA components and a learning rate of 0.01. Loss and accuracy for both the training and holdout datasets can be seen in Figure 1.

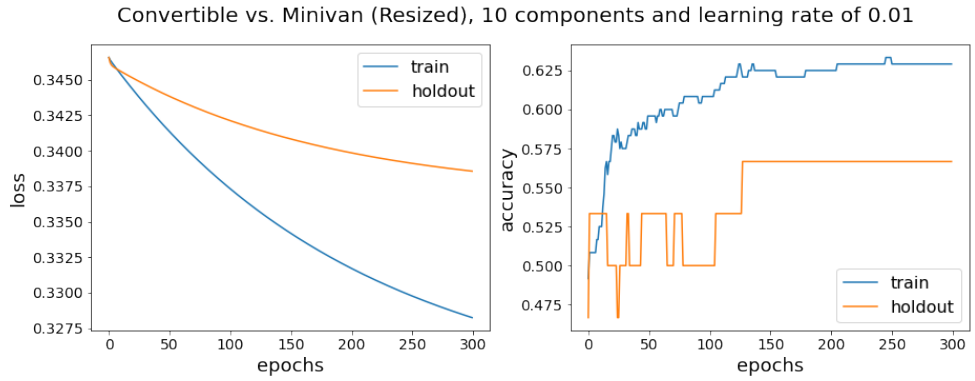

Figure 1: (Resized dataset) Convertible vs. Minivan model performance across training.

Both the training and holdout error drop as the training continues across the number of iterations, although the training error drops at a much higher rate. This is to be expected as the model is using this data for weight updates. Note, we might expect to see the holdout error to go down and go up again due to a certain point in which the model overfits the training data and performance decreases on the holdout data; while we did not see this for this run of the model, we did see the expected result if we changed the train/validation/test split (not shown here).

The training accuracy increases as the number of iterations increases until it levels out at around 0.625 halfway through the training (M = 150 epochs). The holdout accuracy increases and decreases as training progresses until about halfway through training, at which point it levels off as well.

In Figure 2, we vary the value of the the learning rate and the number of components used for PCA. Learning rate values of 0.1 or 0.05 introduce significant error into the data, but any learning rate 0.01 or below achieve a low error rate. Varying the number of PCA components shows a decrease in training error with an increase in number of components, which is expected as we are providing more information to our model to make predictions (Figure 2). The loss significantly drops between 2 to 7 components, but does not change much from 10 to 12 components. Consequently, we used

10 components in our final model to predict on the test set to speed up calculations without much decrease in the model's performance.

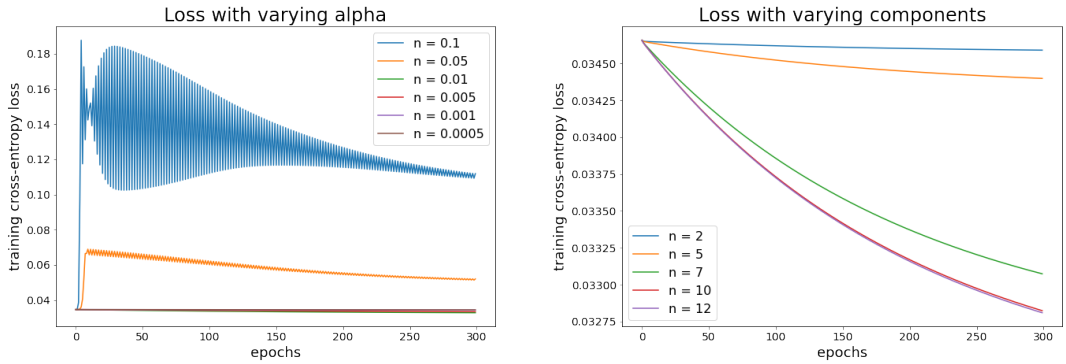

Figure 2: (Resized dataset) Convertible vs. Minivan cross-entropy loss across training with varying learning rate and number of principal components.

We can also visualize the PCA components as images to understand the shapes that capture the most variance in the training data. Figure 3 has the top 4 such components represented. In the resized (but not aligned) dataset, the cars appear to be different sizes in the images, which negatively affects prediction performance as the model has to also learn the relative size of the vehicle within the image. This explains why the model performance was poor. We will see in the next section that aligned images drastically improve performance.

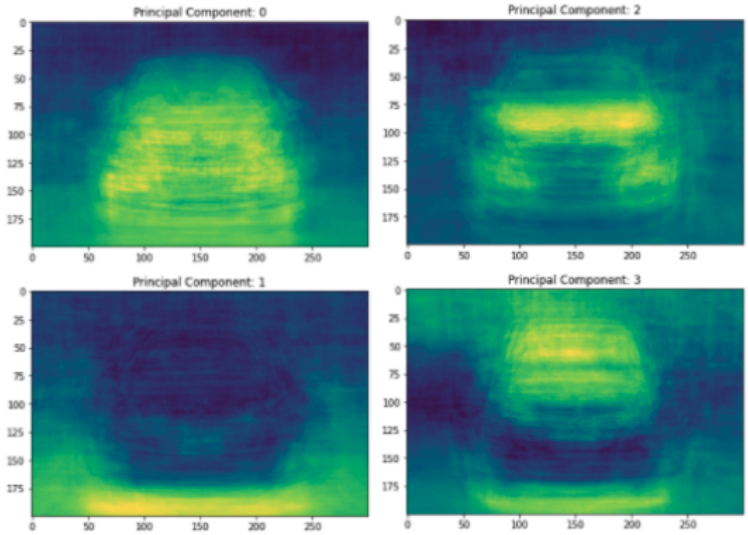

Figure 3: (Resized dataset) Top four principal components of Convertible vs. Minivan.

3.1.2 Aligned Convertible vs. Minivan model

Using 10-fold cross validation, our model achieved an average test loss of 0.24 (± 0.02) and an average test accuracy of 76.4% ($\pm 4.64\%$) using 12 PCA components and a learning rate of 0.05. The loss for both the training and holdout datasets increased during the first few iterations, then decreased monotonically at a similar rate over the model training (Figure 4). The training and holdout accuracy increase until 200 iterations, at which point it leveled off with the training accuracy being about 5% higher than the holdout accuracy.

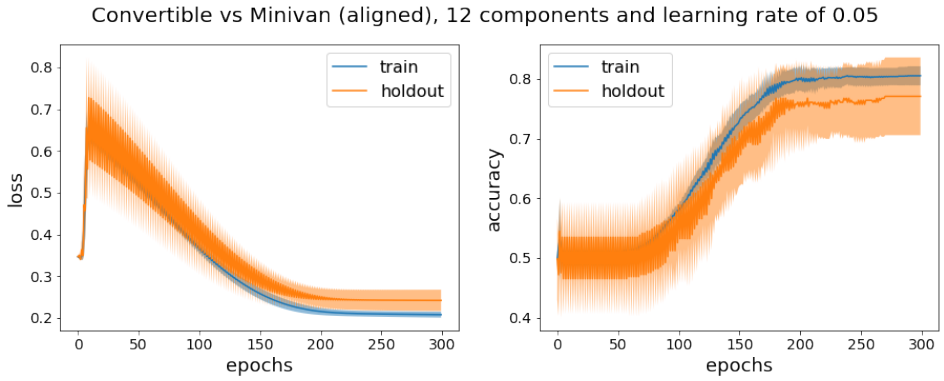

Figure 4: (Aligned dataset) Convertible vs. Minivan model performance across training.

Increasing the number of components used for training decreases the training loss with the loss significantly decreasing from 2 to 12 components (Figure 5). Since the difference in loss between 10 components and 12 components was appreciable, we used 12 components in the final model. Any learning rate less than 0.1 resulted in a sufficiently low loss, with 0.05 resulting in the best model performance.

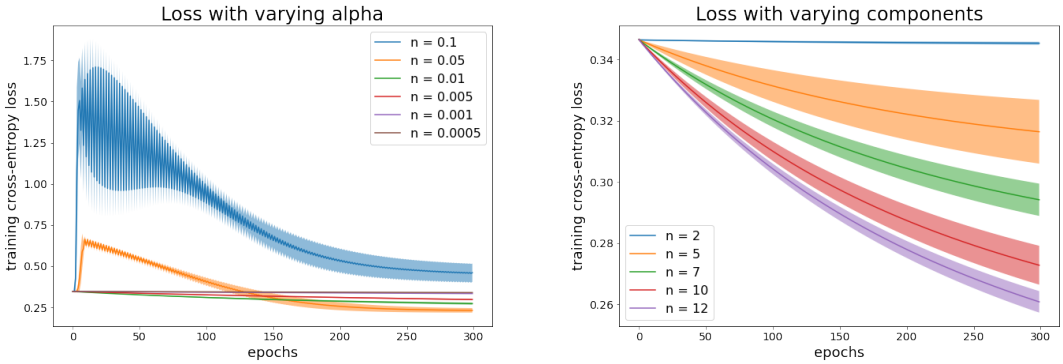

Figure 5: (Aligned dataset) Convertible vs. Minivan cross-entropy loss across training with varying learning rate and number of principal components.

Visualizing the PCA components (Additional Figures, Figure 12), we can gain intuition on why the model performs better on the aligned dataset than in the previous resized dataset. In the aligned dataset, all the cars are the same relative size within the image. The PCA components suggest that the model does not have to learn the difference in sizes of the vehicles as in the un-aligned dataset, and instead can focus on visual features that distinguish one car type from the other. This allows the model to perform better using the same number of principal components during training.

3.1.3 Aligned Sedan vs. Pickup model

Using optimized hyperparameters, we achieved a test loss of 0.23 (± 0.04) and test accuracy of 80.00% ($\pm 6.99\%$) using 10 PCA components and a learning rate of 0.05. The train and holdout loss both decrease with the number of epochs, but the training loss decreases at a slightly faster rate (Figure 6). Both the holdout and training accuracy increase rapidly at first then level out by about halfway through the training. In this model, the training and holdout accuracy are approximately the same, suggesting slightly less overfitting than in the Convertible vs Minivan model.

As shown in Additional Figures - Figure 13, the gradient descent algorithm does not work effectively with larger learning rate. The learning rate of 0.01 again provides the lowest loss while maintaining

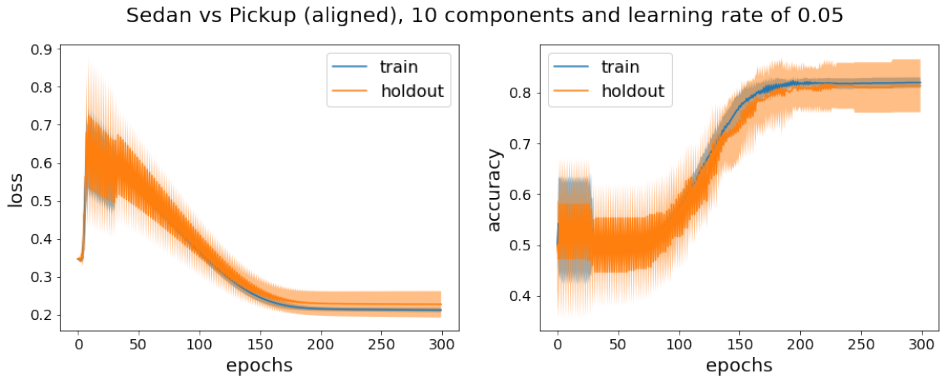

Figure 6: (Aligned dataset) Sedan vs. Pickup model performance across training.

a functioning training algorithm. The training loss again decreases with the number of components, and the change in loss per additional component decreases as the number of components increases.

The visualized PCA components vary drastically from the previous parts, corresponding to the unique shape of sedan and pickup vehicles (Figure 7). The components give insight into why the model might have performed best on this task: sedans and pickups are visually distinct, which makes it easier for the model to recognize the shape that would constitute the difference between the two. Convertibles and minivans seem harder to distinguish based on the principal components.

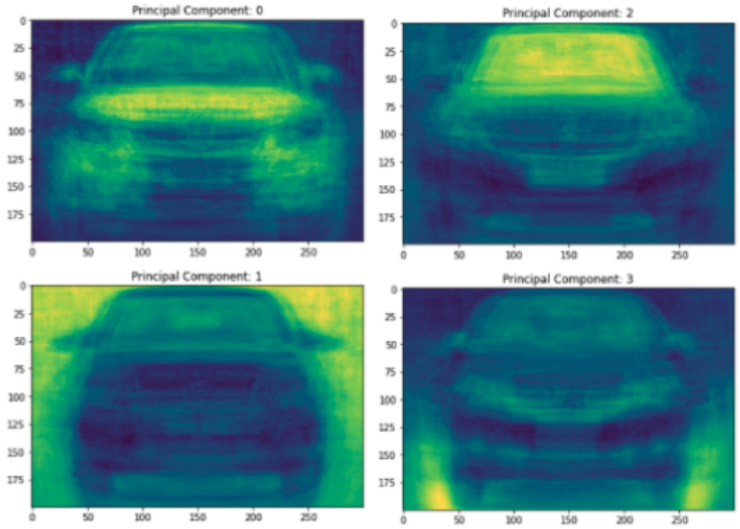

Figure 7: (Aligned dataset) Top four principal components for Sedan vs. Pickup.

3.2 Softmax Regression

Using the optimized hyperparameters, we achieved a test loss of 0.28 (± 0.01) and test accuracy of 55.14% ($\pm 7.77\%$) using 40 PCA components and a learning rate of 0.01. Similar to our first logistic model, the training and holdout loss decrease monotonoically. Both the training and holdout accuracy flatline at around 30 epochs, with the training accuracy approximately 10% higher than the holdout accuracy (Figure 8).

We can visualize the predicted classes of the model against the true labels of the test set in a confusion matrix (Figure 9). The values along the diagonal represent correct predictions. Our model

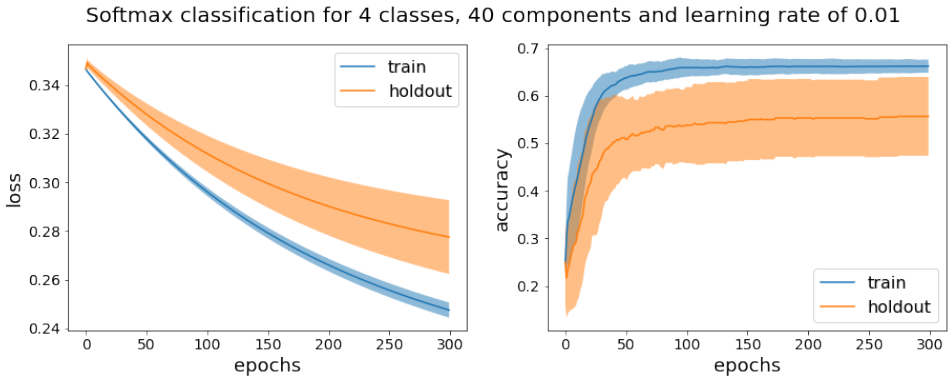

Figure 8: Performance of Softmax model with batch gradient descent across model training.

seems to predict many Convertibles incorrectly as Sedans, and many Sedans incorrectly as Minivans. For future work, we should consider shape feature representations that capture the unique characteristics of these vehicle models for better overall accuracy.

		Predicted			
		Convertible	Minivan	Sedan	Pickup
True	Convertible	9	0	6	1
	Minivan	0	10	1	4
	Sedan	2	5	8	0
	Pickup	1	1	1	11

Figure 9: Confusion matrix of true and predicted labels of Softmax model with batch gradient descent on test set.

As the number of components included in the model training increased, the loss decreased. The smallest cross-entropy loss was obtained with a learning rate of 0.01 (Additional Figures, Figure 14). The change in loss was still significant for larger number of components than in the logistic regression model case as the number of categories was larger.

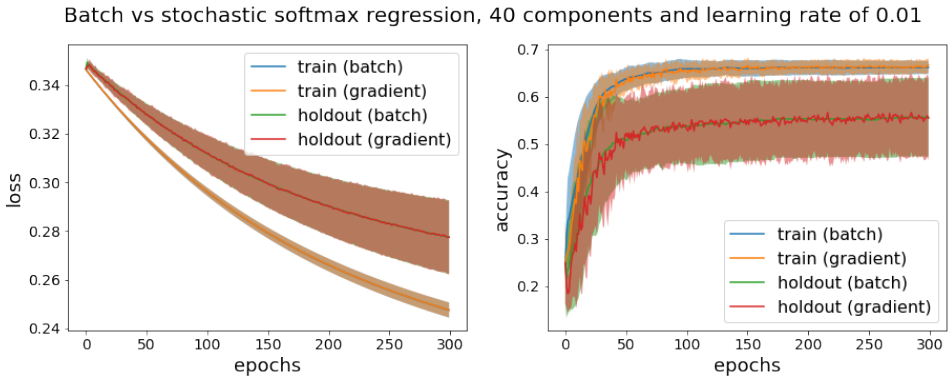

Figure 10: Comparison of Softmax model with batch vs. stochastic gradient descent across training, alpha = 0.01.

A model trained with stochastic gradient descent instead of batch gradient descent obtained very similar test loss (0.28 $\pm$0.01) and accuracy (54.97% $\pm$7.69%) (Figure 10). For comparison, we

also looked at the difference between the two algorithms when the learning rate is 0.05 (Additional Figures, Figure 15). Stochastic gradient descent did not prove to lead to faster convergence in a fewer number of iterations. An interesting next step would be to time both processes and compare the time, but this was not done as a part of this work.

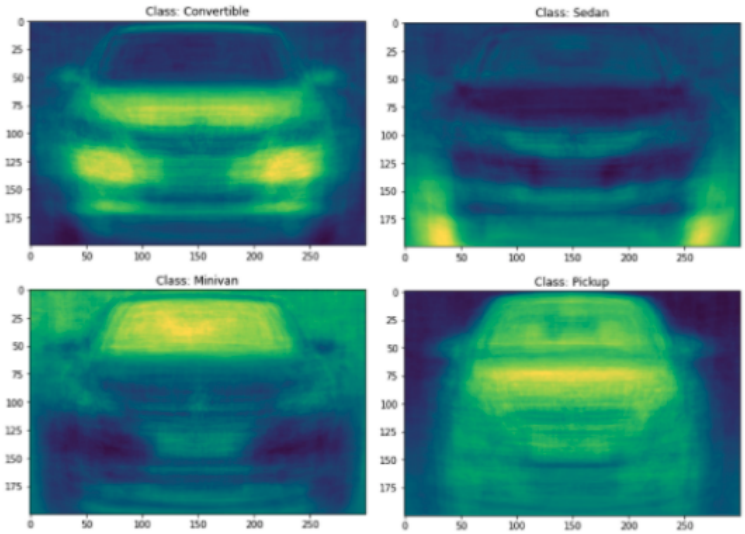

Figure 11: Visualization of weights of the Softmax model with batch gradient descent by vehicle type.

Finally, we can visualize the weights of each predicted class as an image by projecting the learning principal components back into the image space (Figure 11). We see that the learned weights visually represent what we would imagine for each of the vehicle types. However, there is a lot of similarity among the images, which would explain why the model does not result in a high accuracy as it would likely misclassify one vehicle type for another.

4 Conclusion

Our general hypothesis that machine learning models can learn visual characteristics representative of vehicle type was affirmed. The classification task predicting Convertible vs. Minivan using the resized (but not aligned) dataset had much lower accuracy compared to that of the other two binary prediction tasks. The visualization of the principal components suggests that the model has to learn the disparity in vehicle size within the image due to the lack of alignment, thus explaining its poor performance. Predicting Sedan vs. Pickup on the aligned dataset resulted in the highest accuracy; these vehicle types are quite visually distinct and the principal components highlight the differences. Finally, the visualization of the learned weights for each vehicle type from the softmax regression was intuitive and visually characteristic of what a person would imagine for each vehicle type.

5 Contributions

Margot programmed the softmax regression algorithm as well as the stochastic gradient descent. She also created the functions for all the plotting and for varying the hyperparameters. Additionally, she set up the LaTeX for the report and wrote the methods and results sections of the report.

Anshuman programmed the implementation of PCA, cross-validation, and logistic regression with batch gradient descent. He also completed the abstract, introduction, and discussion sections of the report.

Both affirm that the other's contribution was equal in effort and outcome.

6 Additional Figures

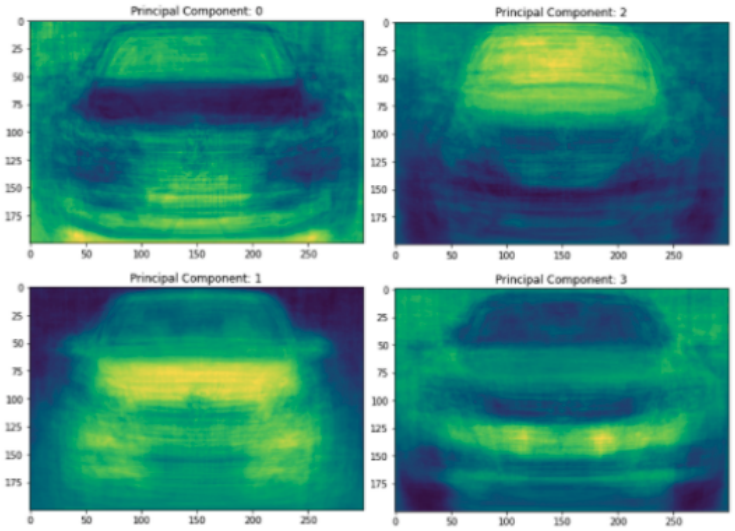

Figure 12: (Aligned dataset) Top four principal components for Convertible vs. Minivan.

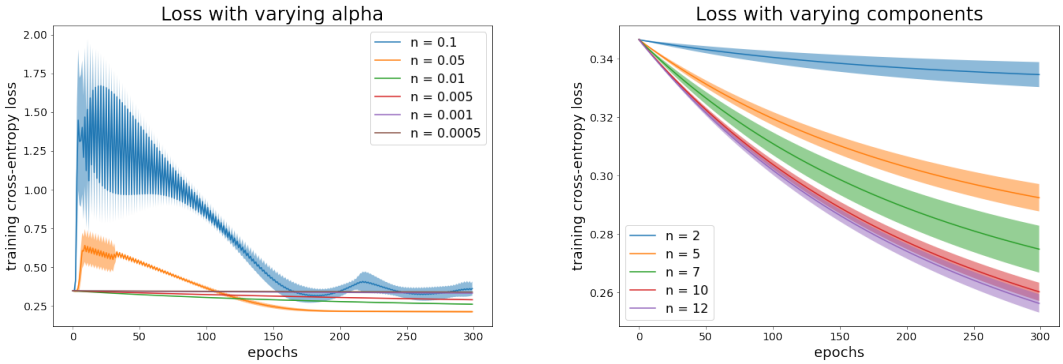

Figure 13: (Aligned dataset) Sedan vs. Pickup cross-entropy loss across training with varying learning rate and number of principal components.

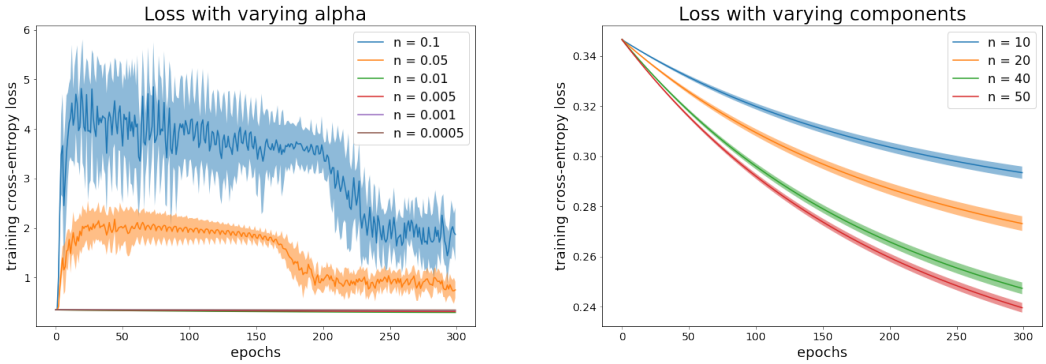

Figure 14: Softmax cross-entropy loss across training with varying learning rate and number of principal components.

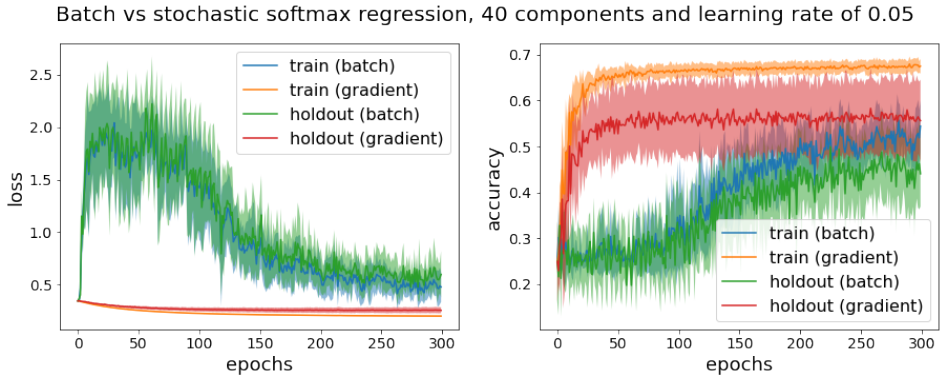

Figure 15: Comparison of Softmax model with batch vs. stochastic gradient descent across training, alpha = 0.05.