Implementation of Logistic and Softmax Regression for Classification of Emotions and Faces

Gitika Meher
Department of Electrical and Computer Engineering
University of California San Diego
9500 Gilman Dr, La Jolla, CA 92093
gkarumur@eng.ucsd.edu

Nasha Meoli
Department of Electrical and Computer Engineering
University of California San Diego
9500 Gilman Dr, La Jolla, CA 92093
nmeoli@eng.ucsd.edu

Abstract

There were 3 different tasks associated with this project with the aim of classifying emotions on faces. A dataset of 80 images was provided for training and testing of the classification algorithms. To this end a two different regression algorithms were employed, logistic regression for the two class (emotion) problem and softmax regression for the multi-class problem. To train the regression algorithm, the high dimensional image data (features) were reduced to a lower dimensional space by Principal Component Analysis (PCA). For the two class problem an accuracy of was achieved and for the multi-class problem, an accuracy of was achieved.

1 Pre-processing the Data

Every image in the dataset is of size 380x240 pixels. We continue with the classification algorithms with the re-sized data of size 91200x1 pixels. We use Principal Component analysis for dimensionality reduction so that we can work with a smaller data size.

1.1 Using Training Data to compute Principal Components

Only Training Data is supposed to be used for calculating the principal components as the network will learn the features of the Test and the validation data if they were included in calculating the principal components and as a result, the classification would not produce reliable results. When using PCA with the test sets, the principal components will include the features of the test set also. So, when test data is projected onto these components, the transformed data obtained will lack the features which are not present in the training set. And therefore, influence the results in an undesirable way.

1.2 Dataset

The dataset contains 8 different facial emotions: happy, happy with teeth, maudlin, surprise, fear, anger, disgust and neutral. It contains 10 subjects with the above mentioned 8 emotions. It has a total of 80 images.

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

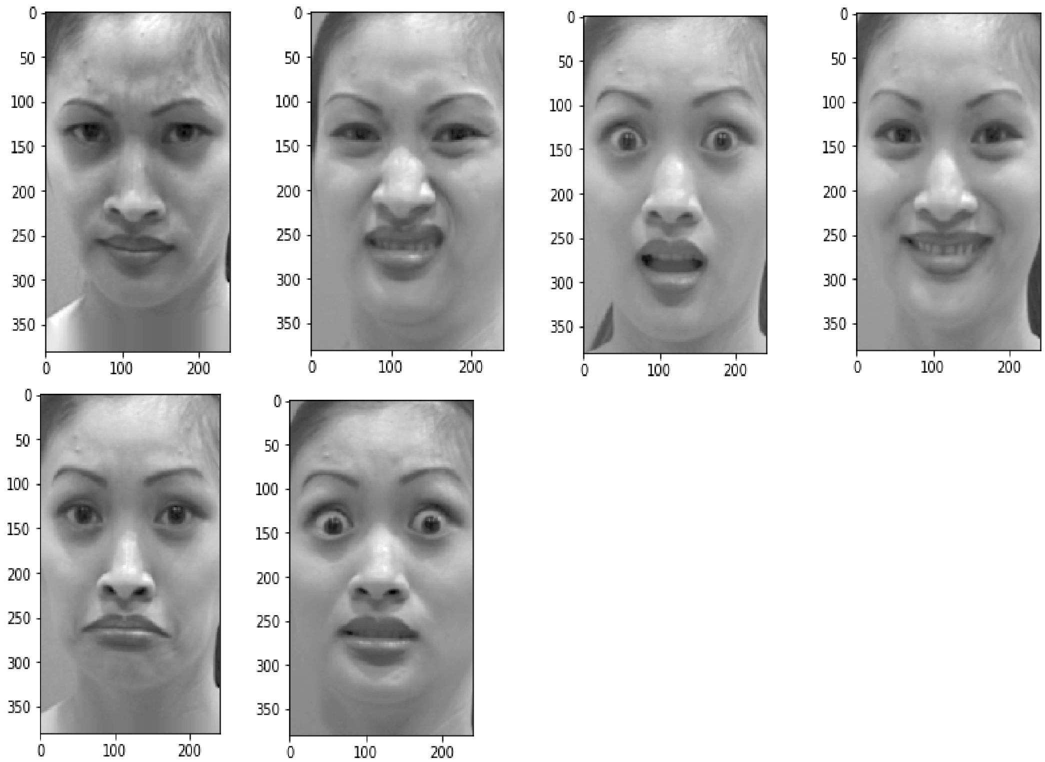


Figure 1: Images showing 6 different emotions: angry, disgust, surprise, happy, sad and fear respectively

1.3 Principal Components after PCA

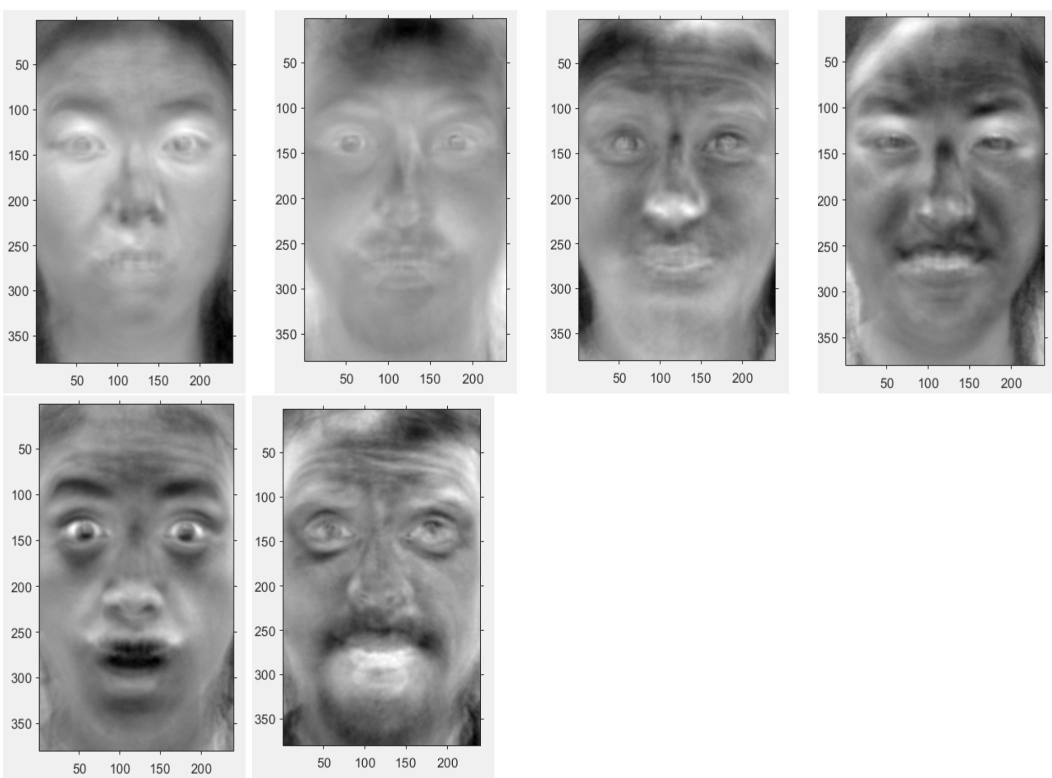


Figure 2: Images showing the first 6 eigenvectors

Figure 2 shows the six eigenvectors corresponding to the six highest eigenvalues of the given dataset containing 80 images.

2 Logistic Regression

Logistic Regression is used for a binary classification problem in this assignment. There are 2 emotions to be classified and a set of 20 images was used for training (80%), hold-out(10%) and test set(10%). The prescribed method of training and testing was to find the ideal weights and learning rate for this problem and the results are presented in the subsequent sections.

2.1 Implementation

Logistic regression is implemented using the batch gradient descent method. Code for the same is submitted separately as logistic.py

2.2 Evaluation on Happy vs Maudlin

Each subject/face among the 10 subjects is used as the test set once, and for each test subject, a random subject is selected to be the holdout. And so, the error and accuracy are reported as the average over the 10 subjects. Ten principal components are used for dimensionality reduction

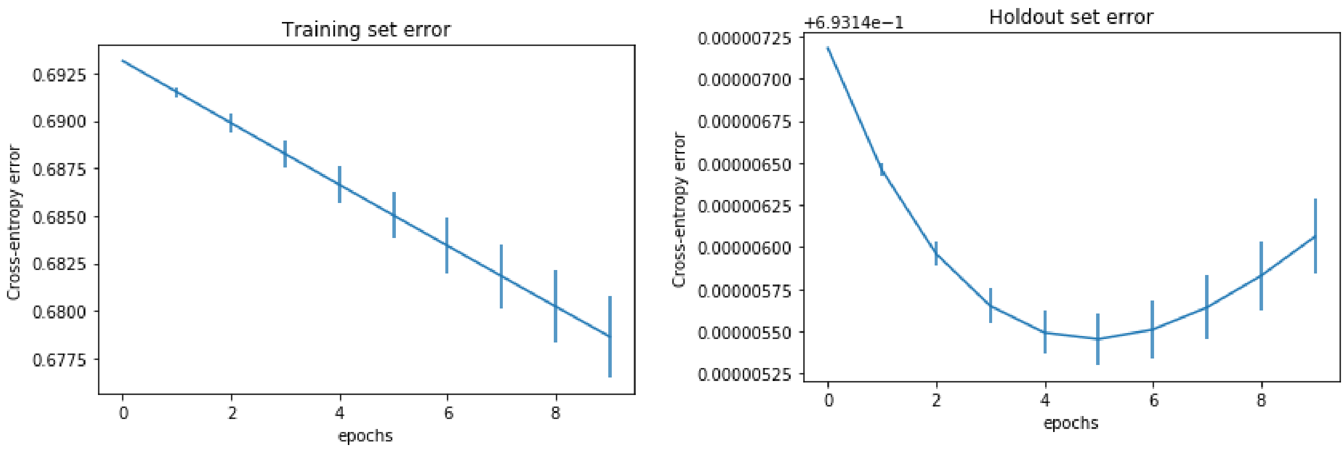


Figure 3: Plots showing the cross-entropy error of the training and the hold out set with an ideal learning rate of 0.0015

2.2.1 Test Accuracy

Average accuracy obtained over the Test set is 86.2%(3.4).

2.2.2 Different Learning rates

The plots below present the cross-entropy errors when too small or too large learning rates are used. Plots for the ideal learning rates are presented in Figure-3.

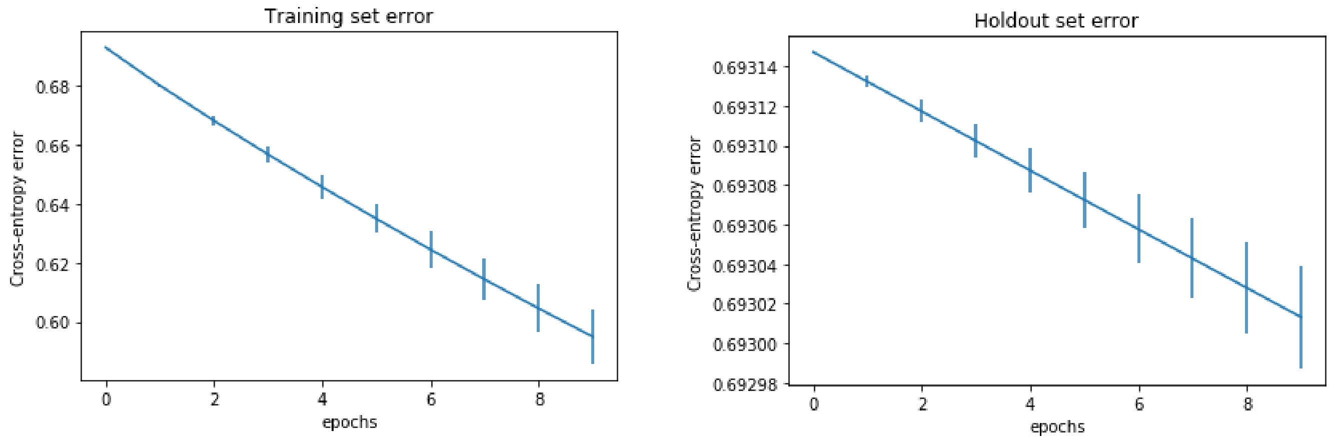


Figure 4: Plots showing the cross-entropy error of the training and the hold out set with a small learning rate of 0.00001

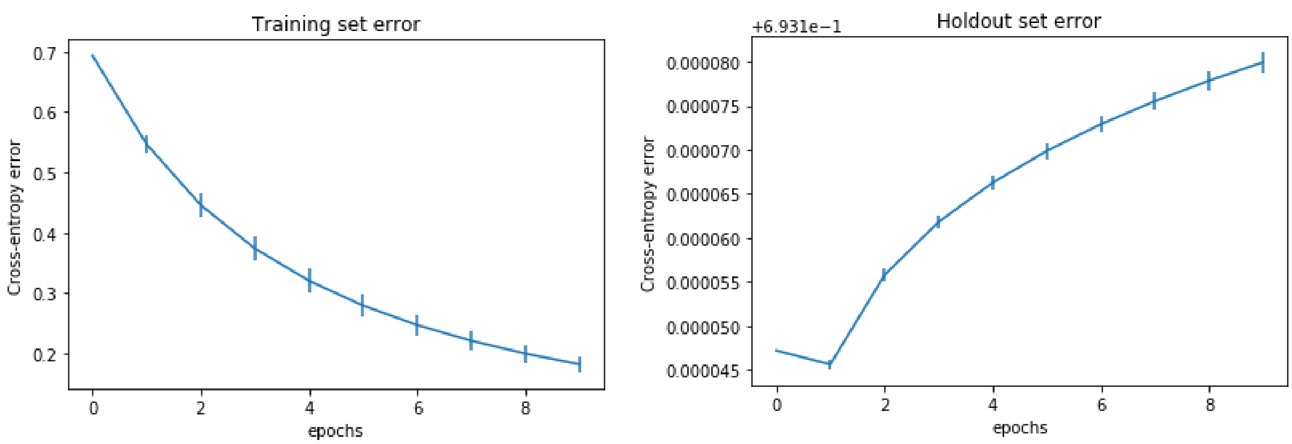


Figure 5: Plots showing the cross-entropy error of the training and the hold out set with a high learning rate of 0.1

2.3 Evaluation on Afraid vs Surprised

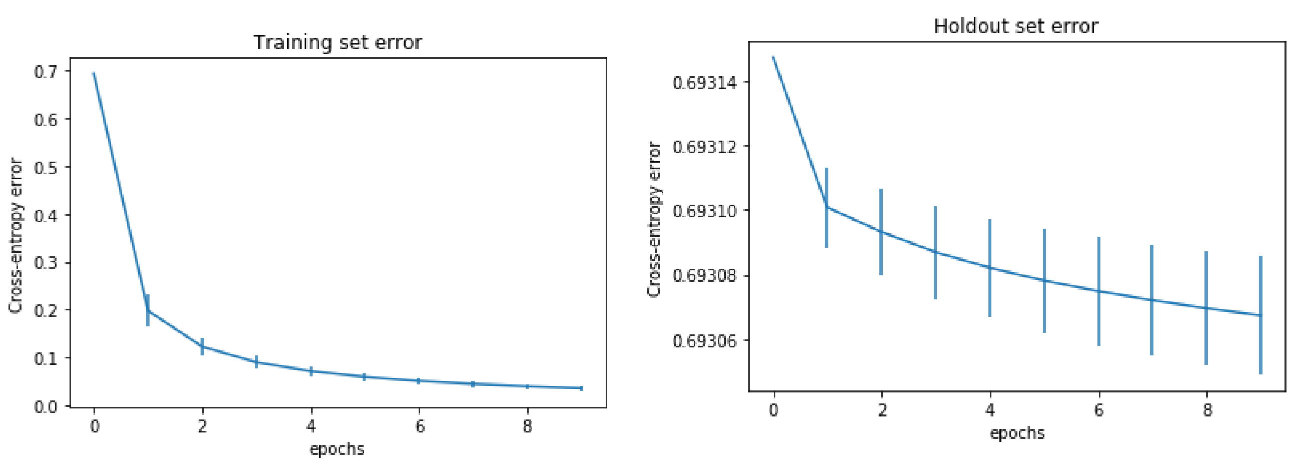


Figure 6: Plots showing the cross-entropy error of the training and the hold out set with a ideal learning rate of 0.0015

2.3.1 Accuracy

Average accuracy obtained over the Test set is 71.25%(6.7). The accuracy obtained is much worse than that of the happy vs maudlin. This is because features extracted in this case(Fear and surprised) are considerably similar when compared to that of happy vs maudlin. Eyes and mouth in both the Fear and the surprised case are similar(larger or open)

3 Softmax Regression

There are 6 different emotions to be classified and a set of 60 images was to be used as the training (80%), hold-out(10%) and test set(10%).

The prescribed method of training and testing was to find the ideal weights and learning rate for this problem and the results are presented in the subsequent sections. 20 principal components were used to report the same.

3.1 Training and Holdout Loss

For each test set, the cross-entropy loss was computed for the training and holdout sets for each of the 50 training epochs. This was repeated 10 times for each test set and the average of these ten runs was plotted with the standard deviations as shown below for training and hold out losses. Figures below show the cross-entropy losses for three different learning rates.

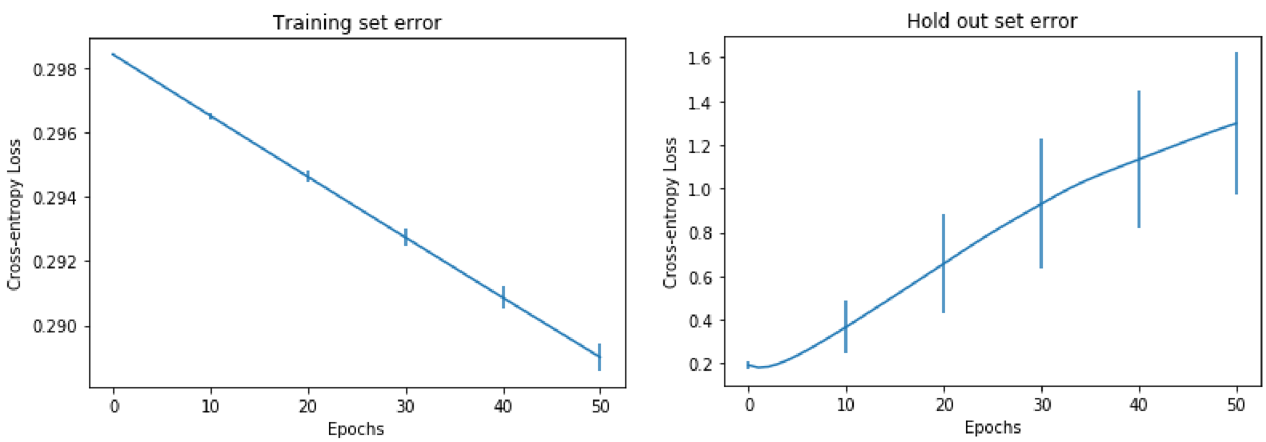


Figure 7: Training set error and the hold out error for a high learning rate of 0.1

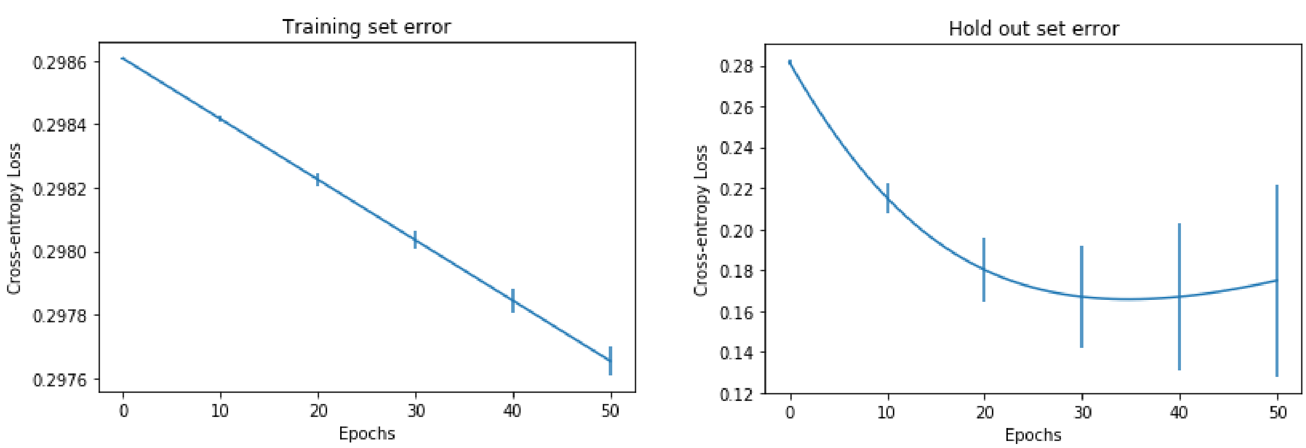


Figure 8 : Training set error and the hold out error for the best learning rate of 0.0019

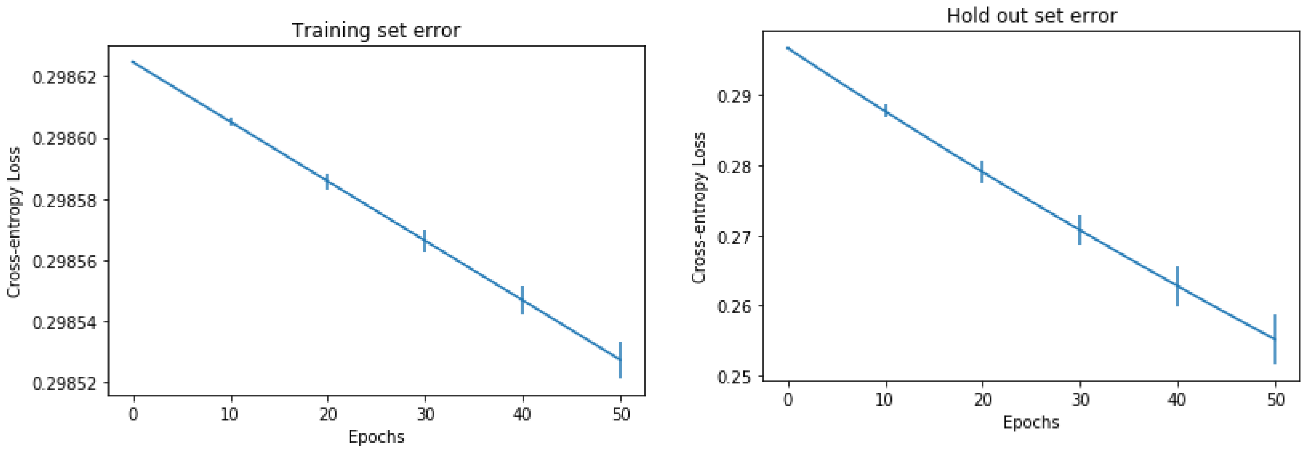


Figure 9 : Training set error and the hold out error for a low learning rate of 0.00001

3.1.1 Different number of principal components

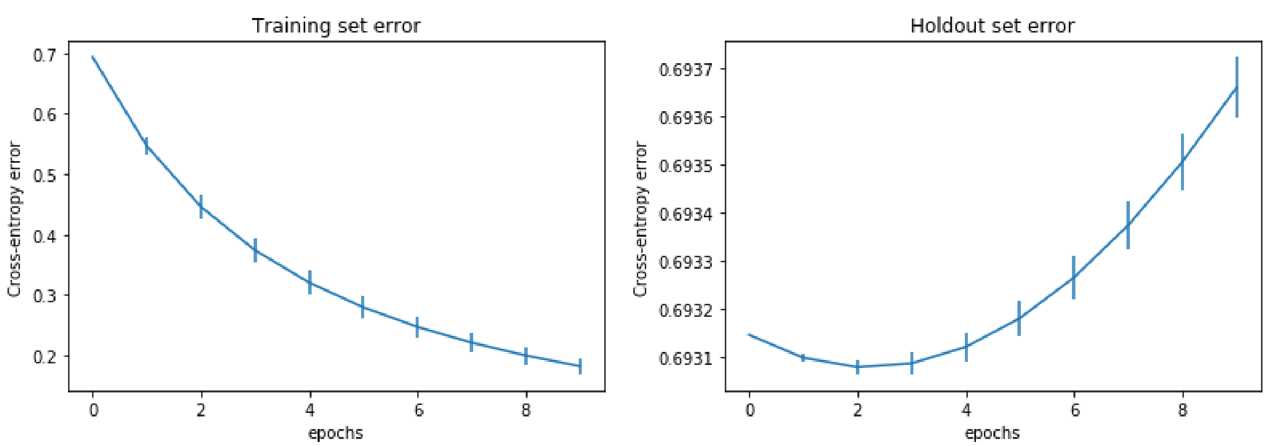


Figure 10 : Training set error and the hold out error for a learning rate of 0.0019 with 50 principal components

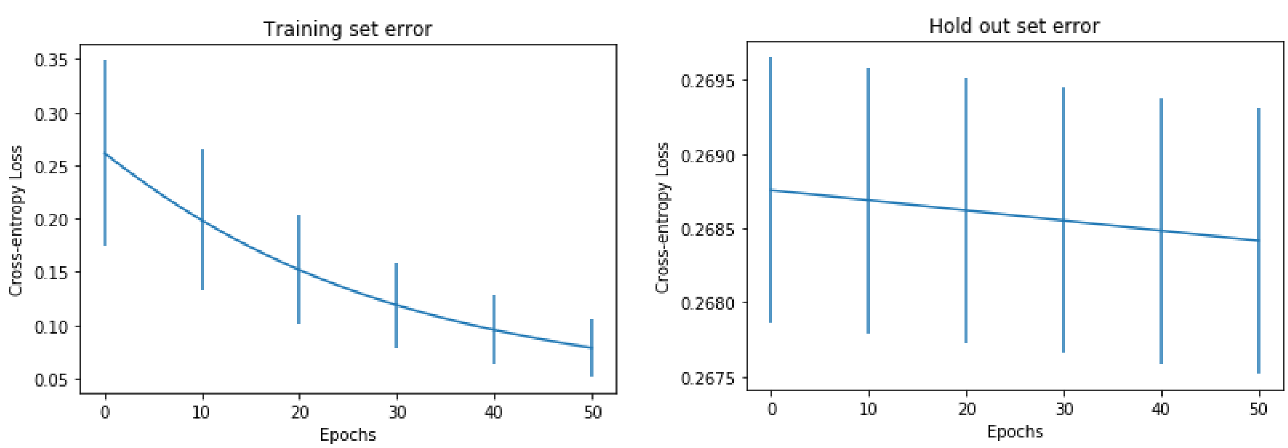


Figure 11 : Training set error and the hold out error for a learning rate of 0.0019 with 10 principal components

3.2 Confusion Matrix for the Test Set

The Table below (Confusion Matrix) shows the confusion matrix with a learning rate of 0.0019 and the best weights selected as the ones resulting in the lowest holdout set error in the 50 epochs.

	0	1	2	3	4	5
0	100	0	0	0	40	0
1	0	90	0	0	0	0
2	0	0	70	0	0	10
3	0	10	0	100	10	0
4	0	0	0	0	50	0
5	0	0	30	0	0	90

Table 1: Confusion Matrix showing the test results computed using 20 principal components

	0	1	2	3	4	5
0	40	10	20	0	20	20
1	10	20	0	20	20	10
2	0	0	10	20	10	0
3	10	20	0	10	10	10
4	30	10	50	0	20	20
5	0	30	10	40	10	30

Table 2: Confusion Matrix showing the test results computed using 10 principal components

	0	1	2	3	**4**	5
0	40	30	50	20	40	40
1	30	40	40	50	20	40
2	40	40	20	20	30	40
3	20	20	20	30	40	20
4	30	30	30	30	30	30
5	30	30	30	40	30	20

Table 3: Confusion Matrix showing the test results computed using 50 principal components

3.3 Batch vs Stochastic Gradient Descent

The training set loss was plotted over 50 epochs for both batch and stochastic gradient with same learning rate of 0.0019. The training set loss of the two was plotted and the figure below shows the results. Stochastic gradient achieves a better error in fewer epochs as in the batch mode, weights are updated only after going through the whole training set once while stochastic gradient method updates weights for every training sample reducing the number of epochs for learning.

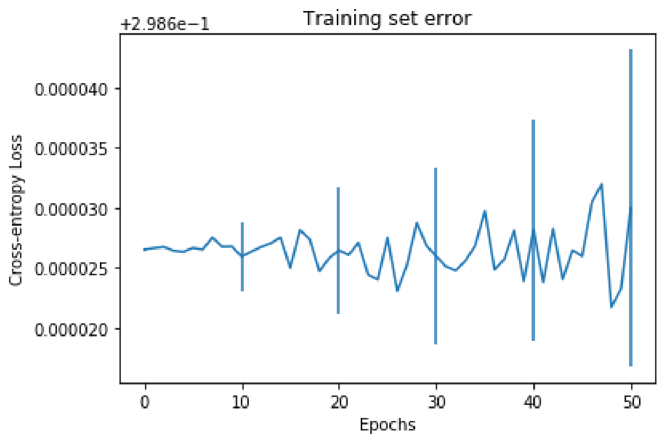


Figure 12: Cross entropy error when Stochastic gradient is used

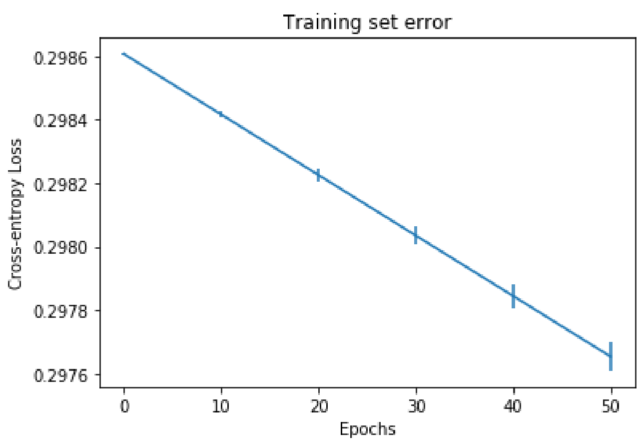


Figure 13:Cross entropy error when Batch gradient is used

3.4 Visualizing the weights

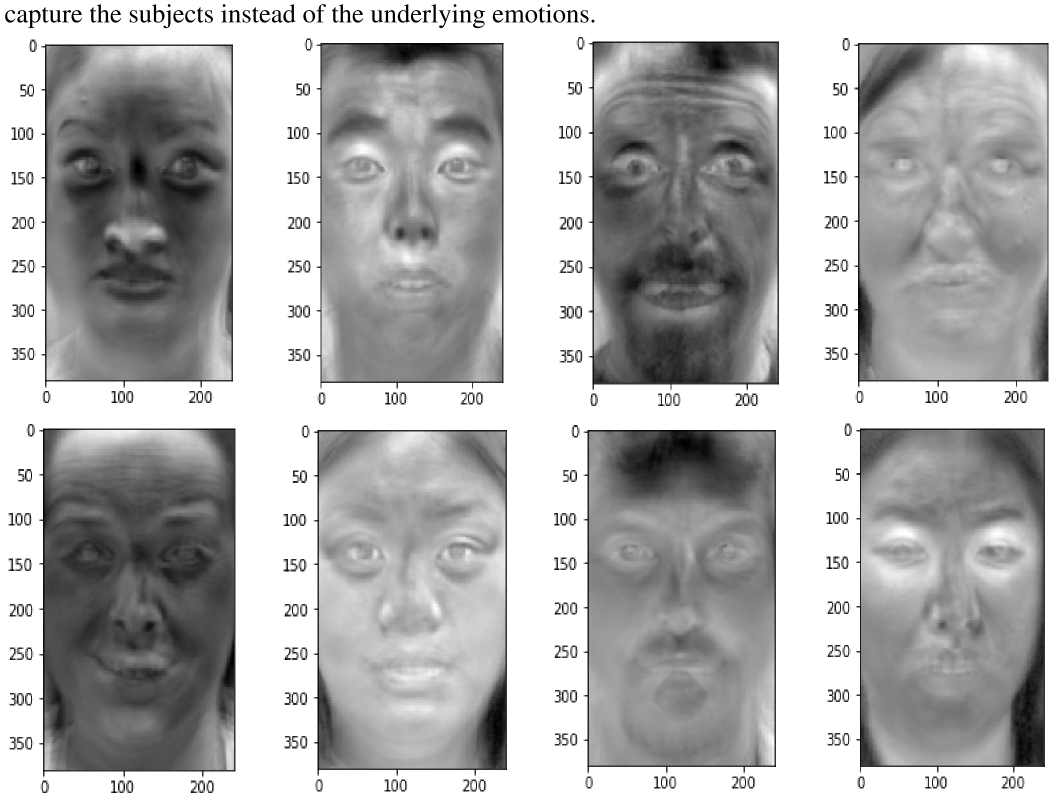
Using the best weights from the training of the regression algorithm, we project this back to face space by multiplying them by their respective eigenvectors and summing these images. This results in the images shown below. We notice that each of these projected weights resembles an emotion. This is because the best weights capture the features in the training set and when projected onto the principal components, they represent the feature they represent which is the emotion in this case.




Figure 14: Visualizing weights for different emotions: anger, disgust, fear, happy, maudlin, surprise respectively

3.5 Classification into 10 identities

Using the best weights from the training of the regression algorithm with labels representing the subjects, we project this back to face space by multiplying them by their respective eigenvectors and summing these images. This results in the images shown below. The following projected weights capture the subjects instead of the underlying emotions.



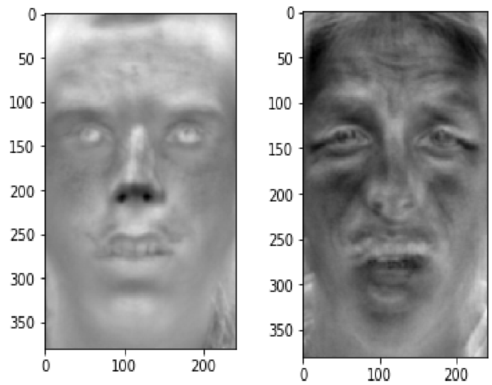


Figure 15: Visualizing weights for different subjects

4 Individual Contributions

Gitika Meher Karumuri - Worked on questions 2, 3.3 and 3.5 of the assignment. Worked on question 1- PCA, logistic regression and stochastic gradient descent for softmax and weight visualization for 10 subjects using softmax.

Nasha Meoli - Worked on question 3 (excluding 3.3, 3.5). Worked on question 1- PCA, softmax function.