Emotion recognition classification methods

Andrew Koch

Research School of Computer Science, Australian National University
u5371834@anu.edu.au

Abstract. In this paper I have presented a comparison of the performances of different classification methods for the task of classifying facial emotions given a 5-dimensional principal component reduction of the local phase quantization and Pyramid of Histogram of Orientation Gradients. These results were then compared to results obtained and presented in a paper on static facial expression analysis, with a comparison being made to the methods used within that paper.[1] It was found that a decision tree based method was better at dealing with overfitting than a deep neural network.

Keywords: Classification · Neural Networks · Random Forest · Discriminant Analysis

1 Introduction

For this project I chose to use the faces-emotion dataset, which comprises of labels for 7 different emotions as well as the compressed representations of the local phase quantization and Pyramid of Histogram of Orientation Gradients of 675 images of faces. This dataset seemed like it would be an easy dataset to train for, as I imagined the 7 emotions used to be fairly easily distinguishable at a glance. I therefore chose to use this dataset because I was interested in observing the differences in difficulty for classifying a visual image using your eyes vs the difficulty of classifying an image using the purely numeric representations of the image. From this dataset I attempted to create a model which would correctly classify the emotion shown when given an image of a face as an input. The model was trained using a training set comprised of 50.5% of the overall dataset, with a random state of 4 used for splitting the training and testing data from the original dataset. When training this neural network, I initially used the principal components of the LPQ and the PHOG for features but neglected to use the name of the image source as a feature. This resulted in relatively low training accuracy, with a higher training accuracy being reached when also using the image source name as a feature. This led me to believe that there might be some type of distinctness in the classification of an emotion depending on the person whose face is being analysed. This makes some sense as different facial structures should affect the LPQ and PHOG differently.

1.1 Problem Description

The problem of emotion recognition within the field of computer science is an important topic to consider. During human interactions the ability to discern the emotion of the person to whom one is talking is a very important ability to have, as it can help to ensure that one does not accidentally insult or otherwise damage relationships with another person. Similarly, within the field of computer science the ability to recognize emotions can be useful in many different ways, as it can allow a system to make adjustments to its behaviour based upon the emotions which a user may be experiencing. For example, you could apply emotional recognition to some form of an automatic teaching system, ensuring that the system can adjust the level of complexity of its descriptions or otherwise alter its teaching parameters based upon the emotional state of the user.[3] This paper focuses on the classification of the emotion being shown on the face of a person within an image. The information about each image has been compressed along the directions of greatest variance using principal component in order to obtain a 5-dimensional representation of the data which contains a majority of the information about the image. This data is then trained using different classification methods in order to obtain a classification for the emotion shown on each sample face.

1.2 Feedforward neural network

A feedforward neural network is one of the most basic kind of neural network architectures which exist. A feedforward network works by having a single input layer, which takes in your input data for which you would like to predict or classify some value, one or more hidden layers, which extract features from the input layer in order to obtain more information from each sample, and a single output layer, which will provide either the class of your model or some other value predicted using your model. For a feedforward neural network, the model learns sequentially with an order to the layers which does not loop back to previous layers at any point during the learning process. This can be represented visually as seen in the diagram below.

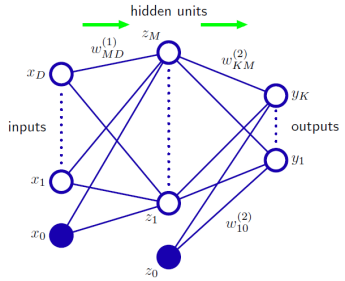


Fig. 1. Diagram of a single layer feedforward neural network.[4]

The above diagram represents a simple feedforward neural network with a single hidden layer. The inputs x_1 to x_D represent different sample inputs while the x_0 and z_0 represent the bias parameters for hidden layers z_1 to z_M and y_1 to y_K respectively. The network works by taking the inputs $x_1, \ldots, x_D$, applying some weight scaling to each of their values, and then finding a linear combination of weights and values of all input values for each neuron within the hidden layer. These hidden layer neuron values are then used similarly in a linear combination of weights and values in order to generate each possible output neuron within the output layer.

1.3 Deep Learning

When defining a feedforward neural network architecture for training it is possible to have multiple hidden layers, each with distinct weight values for each hidden neuron, which can help to extract more information for your model, generally resulting in a better model. Using two or more hidden layers in a neural network is generally referred to as deep learning. A deep neural network architecture can often be more difficult to optimize and as such it is often important when employing a deep learning approach to a problem to experiment with the parameters in order to find the best number of hidden neurons for each hidden layer.[5] Due to each subsequent hidden layer extracting more information from the input feature, a deep learning approach can often help to reduce the generalization error for your model; however, one potential problem which may occur when utilizing deep learning for a classification task is the issue of overfitting. If a deep learning model is overfitting to the input training data, then the best approach to take is to either decrease the number of hidden layers within the model or to increase the number of training samples.

1.4 Decision Trees

A decision tree is an algorithmic structure used to enable easy decision making related to the value of a numeric target value or class target value of a sample. Decision trees work by segmenting the possible target value into a number of simple regions of prediction. This segmentation process is usually represented as a tree structure with the rules for splitting segments shown at each split.[6] An example decision tree structure is illustrated below using the Hitters dataset. The Hitters dataset comprises of 322 sample observations with 20 variables related to the play records and salaries of baseball players in the 1986 Major League Baseball season. The tree attempts to determine the log salary for a baseball player when given as features the number of years theyve participated in the major leagues and the number of hits they made in 1986.

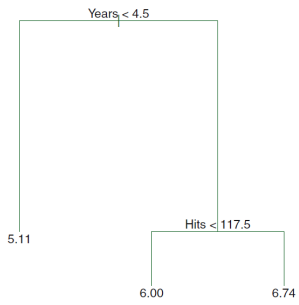


Fig. 2. An illustration of the decision tree structure using the Hitters dataset.[6]

In the above figure the sample values are first split into two distinct segments depending on whether or not they have participated in the major leagues for less than 4.5 years. If they have done so, then their log salary is predicted to be a value of 5.11. Otherwise, a further split is applied based on how many hits the sample player made in the year 1986, with their log salary predicted as 6.00 if they have hit less than 117.5 or 6.74 if they have hit more than that. While the above figure illustrates a prediction of a numeric value a decision tree can also be tasked with predicting the class output for a sample. This is done in a similar fashion to prediction with the only real change being the target values used.

1.5 Maximum Likelihood Classification

Maximum likelihood classification (otherwise known as discriminant analysis) is a method for determining the most probabilistically likely target class for an input sample given a set of features. Discriminant analysis works by modelling the distribution of the features of a dataset separately for each of the target classes and then using Bayes theorem to infer the posterior probability for the target value given the features used.[7] Discriminant analysis can be defined linearly or quadratically. Linear discriminant analysis (LDA) assumes that each class is taken from a multivariate normal distribution, with each class sharing the same covariance matrix despite each class having its own specific means; however, quadratic discriminant analysis (QDA) assumes that each different class has its own covariance. Due to this, the output target class for a specific sample is calculated by finding the class for which the following equation is maximized.

$$\begin{aligned}\delta_k(x) &= -\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\mathbf{\Sigma}_k| + \log\pi_k \\ &= -\frac{1}{2}x^T\mathbf{\Sigma}_k^{-1}x + x^T\mathbf{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\log|\mathbf{\Sigma}_k| + \log\pi_k\end{aligned}$$

Fig. 3. Formula for assigning a sample to a target class.[7]

Where Σ_k and μ_k are the covariance matrix and mean vector for the target class k, x is the sample being assigned. Due to the class-specific covariance matrices used in QDA, QDA is more flexible for multi-class classification than LDA. This is because due to the shared covariance matrices used in LDA it can often suffer from a large class bias if different classes do not share the same covariance values. This makes QDA generally a better choice for classification problems with relatively many target classes such as emotion recognition.

1.6 Random Forest

The random forest classification algorithm is an extension of the decision tree algorithm for classification. A random forest classifier works using a collection of decision tree classifiers, each generated using a random selection of features at each split in order to determine the segmentations used. During classification of a sample the random forest classifier will assign the sample to the class with the highest frequency of classifications based on the classifications of the point for the decision trees which make up the random forest.[8] Generally, when training a random forest model, the number of features used for each split is the square root of the total number of features for your training data. Choosing a number of features lower than the total number of features in your training dataset can be useful for eliminating the problem of overfitting which a single decision tree might encounter, allowing for a better overall model accuracy.[9] Since random forests retain internal estimates for impurity-based feature importance they can also be used as a way to estimate the importance of a feature variable when predicting a target value.

2 Method

The data was split into a training set and a testing set in order to investigate the feasibility of training a neural network to recognize facial emotions. Since the paper split the dataset into two groups of roughly 49.5% and 50.5%, the training set was set to a size of 50.5% of the original dataset.[1] For testing purposes, a random state of 4 was also used when splitting the dataset into training and testing sets. The class frequencies for the training dataset can be seen below:
After splitting the dataset, the data was first classified using a single-layer feed-forward neural network with a sigmoid activation function applied to the hidden layer. The loss value for this model over time during the training process can be seen below along with the confusion matrix generated by comparing the results to the training dataset and the ROC curve.

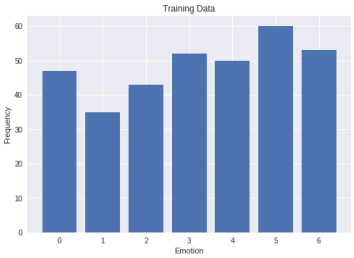
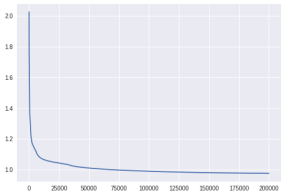


Fig. 4. The class distribution for the training dataset.



```
Confusion matrix for testing:
[[ 6.  3.  9.  6. 10.  8. 11.]
 [ 6.  4.  7.  4.  8.  2.  9.]
 [ 8.  4. 19.  7.  5.  8.  5.]
 [ 6.  1. 10. 17.  6.  6.  2.]
 [10.  1.  6.  9.  8.  9.  7.]
 [ 4.  0.  4.  5. 14. 10.  3.]
 [ 8.  5.  3.  6.  7.  7. 11.]]
```

Fig. 5. Loss over time and confusion matrix for a ffnn with 1 hidden layer.

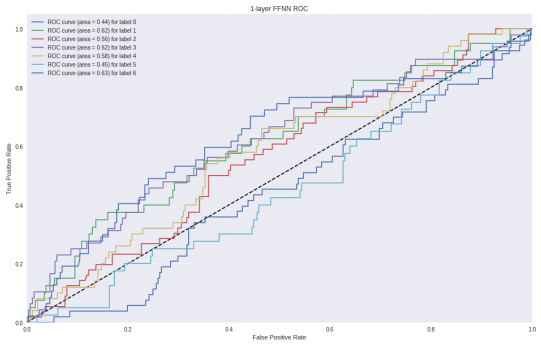
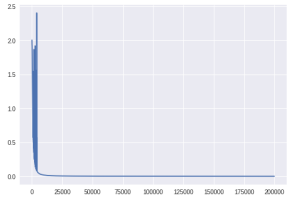


Fig. 6. ROC curve for a feedforward neural network with 1 hidden layer.

After this, the data was classified using a two-layer feedforward neural network with a sigmoid activation function applied to both hidden layers. The loss value for this model over time during the training process can be seen below along with the confusion matrix generated by comparing the results to the training dataset and the ROC curve.



```
Confusion matrix for testing:
[[15.  4.  4.  5.  8.  7. 10.]
 [ 5.  4.  9.  6.  7.  3.  6.]
 [ 6.  7. 16.  3.  2.  9. 13.]
 [ 9.  5.  4.  9.  6.  7.  8.]
 [ 3.  5.  9. 10.  6. 10.  7.]
 [ 5.  3.  3.  9.  3. 13.  4.]
 [ 6. 12. 11.  3.  5.  2.  8.]]
```

Fig. 7. Loss over time and confusion matrix for a ffnn with 2 hidden layers.

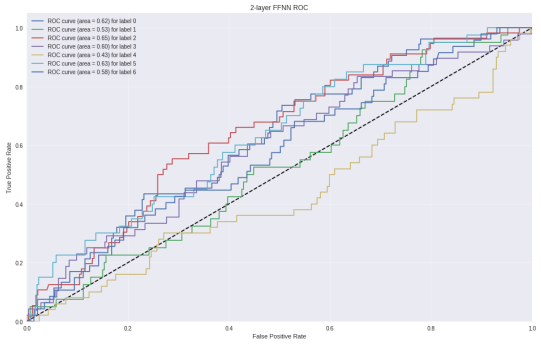
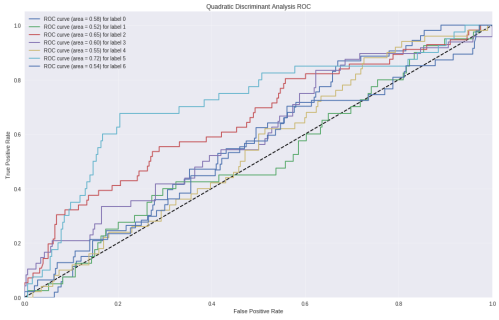


Fig. 8. ROC curve for a feedforward neural network with 2 hidden layers.

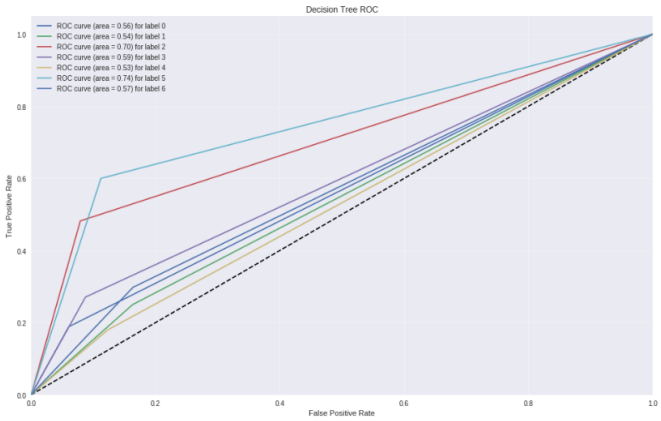
For all neural network architectures an initial learning rate of 1 was used, with the learning rate decaying by 15% every time a tenth of the total training time had passed. The models were then trained for 200,000 epochs. For the single-layer network 9 neurons were present in the hidden layer while for the two-layer network 11 neurons were present in each of the hidden layers. After attempting to classify the data using neural networks, the data was then classified using a maximum likelihood model. The confusion matrix and ROC graph for this model can be seen below.




```
Confusion matrix for testing:
[[ 6  6  3  8 18  2 10]
 [ 6  6  3  9  8  1  7]
 [ 9  6 15 11  8  2  5]
 [ 3  7  5 16 13  3  1]
 [ 5  5  7 13 12  4  4]
 [ 1  6  2 12  9  5  5]
 [ 7  8  4  9 13  0  6]]
```

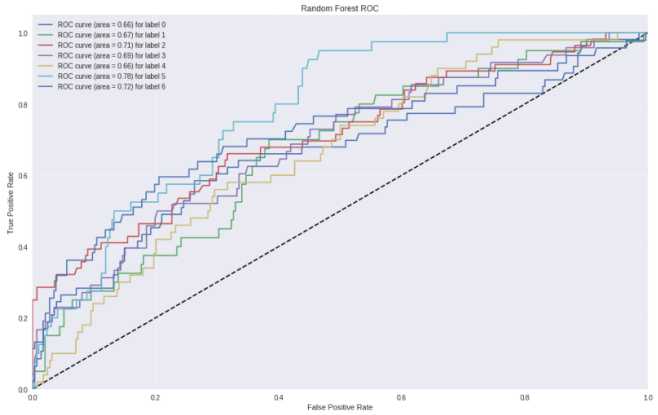
Fig. 9. ROC curve and confusion matrix for a maximum likelihood classification mode.

Finally, the data was classified using decision tree and random foresting methods, with the confusion matrix and ROC graph obtained for both models.




```
Confusion matrix for testing:
[[10 10  3  5  5  5 15]
 [ 4 10  8  2  6  4  6]
 [ 4  5 27  0  7  9  4]
 [ 1  9  3 13  9  3 10]
 [ 3  8  2 12  9  7  9]
 [ 2  2  1  3  5 24  3]
 [ 3 14  5  3  3  5 14]]
```

Fig. 10. ROC curve and confusion matrix for a decision tree classification mode.




```
Confusion matrix for testing:
[[11  5  1  6  4 13 13]
 [ 4  4  3  7  6  6 10]
 [ 0  3 18  6  4 16  9]
 [ 2  1  2 15  7 13  8]
 [ 4  1  1  8 13 17  6]
 [ 1  0  2  6  3 24  4]
 [ 2  5  4  1  4  7 24]]
```

Fig. 11. ROC curve and confusion matrix for a random forest classification mode.

The features used for training each of the models above were the 5 principal components of the Local Phase Quantization and the 5 principal components of the Pyramid of Histogram of Orientation Gradients. The name for the source of each image was also converted to a numeric value and used as a feature.

3 Results and discussion

The accuracy for each classification model is given in the table below.

Table 1. Classification model accuracies.

Method	Training accuracy (%)	Testing accuracy (%)	Loss
Single layer FF network	65	22.46	0.9746
Two layer FF network	100	21.26	0.0012
Maximum Likelihood	42.93	19.76	N/A
Decision Tree	100	32.04	N/A
Random Forest	100	32.63	N/A

From the above table it can be seen that attempting to use a simple feedforward deep neural network for classification of facial emotions is not a very good approach, giving a worse training accuracy than a single-layer feedforward network due to overfitting. While a decision tree approach to classification also suffers from overfitting, it is generally a better approach as it is able to better handle that overfitting. In fact, by comparing the ROC curves for each classification method if can be seen that the random forest classification methods are consistently better than the other methods explored, meaning that it is most likely to correctly identify an emotion. In particular it is very likely to correctly identify the emotion of sadness on a given individual's face.
It is worth noting that the emotion of sadness appears to be quite easy to clasify for not only a decision tree based classification method, but also for a discriminant analysis based method.
It is also worth noting that the emotion of anger is often mistaken for the emotion of surprise, with all of the models misclassifying at least 10 samples for faces which showed anger as showing surprise instead. This makes some sense as showing such emotions both often involve raising your eyebrows and opening your mouth.

4 Conclusion and future work

Within this report I have explored different methods for classifying emotions on a face given a dataset of facial images containing ground truth emotion values. It was found that decision tree based classification methods such as random foresting would give the best results for classification on testing data, with it being the model most likely to correctly identify an emotion. The maximum likelihood based classification method was found to be the worst method overall, which agrees with the findings of a similar paper on the comparison of classification methods related to visual data.[2]
An extension of this work may involve an investigation into classifying emotions using deeper levels of deep learning, with an investigation into the exact correlation between the number of hidden layers used when training a model and the

accuracy and loss values obtained. It may also be a good idea to investigate the usage of clustering algorithms based on similarity metrics and the possible classification accuracy obtained though their usage. For future work I believe that it would be good to explore the usage of different features and feature extraction processes when attempting to classify the dataset. This could potentially be extracted through the usage of a convolutional neural network. I also believe that it would be good to gain access to a dataset where each face is clearly labelled as belonging to a single individual, and each individual has images showing the full range of emotions on their face. This would allow the model to be properly trained on a wider range of facial structures and thus increase the classification accuracy.

References

[1] A. Dhall, R. Goecke, S. Lucey and T. Gedeon, "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark," 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, 2011, pp. 2106-2112, doi: 10.1109/ICCVW.2011.6130508.
[2] L. K. Milne, T. D. Gedeon and A. K. Skidmore, " Classifying Dry Sclerophyll Forest from Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood " Proceedings of 6th Australian Conference on Neural Networks, Sydney, NSW, Australia, 1995, pp. 160-163
[3] X. Alam-Pineda, E. Ricci and N. Sebe, Multimodal behavior analysis in the wild: Advances and challenges. Elsevier, 2018, pp. 387-406.
[4] C. Bishop, Pattern Recognition and Machine Learning. New York: Springer, 2006, pp. 228-230.
[5] I. Goodfellow, Y. Bengio and A. Courville, Deep learning. MIT Press, 2016, pp. 164-223.
[6] G. James, D. Witten, T. Hastie and R. Tibshirani, An introduction to statistical learning. New York: Springer, 2013, pp. 303-304.
[7] G. James, D. Witten, T. Hastie and R. Tibshirani, An introduction to statistical learning. New York: Springer, 2013, pp. 138-150
[8] J. Han, M. Kamber and J. Pei, Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann, 2011, pp. 138-150
[9] G. James, D. Witten, T. Hastie and R. Tibshirani, An introduction to statistical learning. New York: Springer, 2013, pp. 319-321