Effectiveness Assessment of Feedforward and BiDirectional Neural Network for Facial Expression Classification with Feature Selection

Han Jingyu

Research School of Computer Science, Australian National University
U6764688@anu.edu.au

Abstract. Effective neural networks can be used as a prediction layer in a deep learning architecture to classify facial expressions. A 2-hidden-layer feedforward (FF) model achieves 23.8% average accuracy compared to the 19% benchmark. BDNN technique fails to improve the performance. It achieves 20.3% accuracy but is less stable than the FF model and unable to overcome dataset imbalance. A genetic algorithm is used to select more informative features. The FF model trained with the selected features achieves 28.7% accuracy while those features do not always help in training the BDNN model.

Keywords: Facial Expression Classification, BiDirectional Neural Network, Feature Selection

1 Introduction

Classification of realistic facial expression images is a challenging task, not only because of the complexity of image data but also the noise caused by the complex nature of real-world conditions. Static Facial Expression in Wild (SFEW) database provides such images, accompanied by protocols and baseline evaluations for a model trained on it. The SFEW paper [1] suggests a 19% classification accuracy benchmark for support vector machine (SVM) trained on local phase quantisation (LPQ) and pyramid of histogram of gradients (PHOG) descriptors of images. Those descriptors extract the appearance and shape information of images and encode that into numerical values [2].

This report intends to compare the classification accuracies of a 2-hidden-layer feedforward neural network (FF) and a bi-directional neural network (BDNN) [3], and to investigate whether the two models are effective in performing the classification task. Such investigation is relevant as a simple multiplayer model can be incorporated in a large deep learning model as a predicting layer, which receives processed data from the previous layer and conducts classification. In addition, BDNN trained with no restrictions on the input-output relation provides a way to find the cluster centre of each class [3], which can be used as a down-sampled layer in a larger model.

2 Method

2.1 Data Preparation

A down-sampled SFEW dataset, which contains 674 patterns, will be used. Each pattern is composed of the first 5 Principal Component Analysis (PCA) values of the LPQ and PHOG descriptors respectively, labelled as from 1 to 7 to represent emotions: angry, disgust, fear, happy, neural, sad, surprise.

Outlier removal is performed to reduce noise in the dataset and thus the variance of it. Outliers with extreme values in training dataset can introduce large error signals to a model, making it bias to an incorrect solution and unable to generalise to unseen testing dataset (with or without outliers) [4]. To detect outliers, all patterns are analysed using local outlier factor (LOF) technique, which assigns a degree of being an outlier with respect to the surrounding neighbourhood [5]. The number of neighbours is set to 7 for the k-distance neighbourhood calculation used by the technique.

Scaling is required to prevent a model being bias towards some features of larger value ranges (Figure 1 Left) and taking longer training time to overcome the bias. While detecting outliers among patterns, LOF could not always identify extreme values in one particular feature. Robust scaler (RS) is used to scale feature values according to the quartile range instead of the sample mean which is sensitive to extreme values. With the same model settings trained in 40 epochs (described later), the FF model learns better as shown by increased training accuracy and generalises better as shown by increased validation accuracy.

accuracy	Unprocessed	LOF+RS
Training	21.3	48.1
Validation	13.7	24.0

Table 1. Performance of FF with unprocessed and pre-processed data

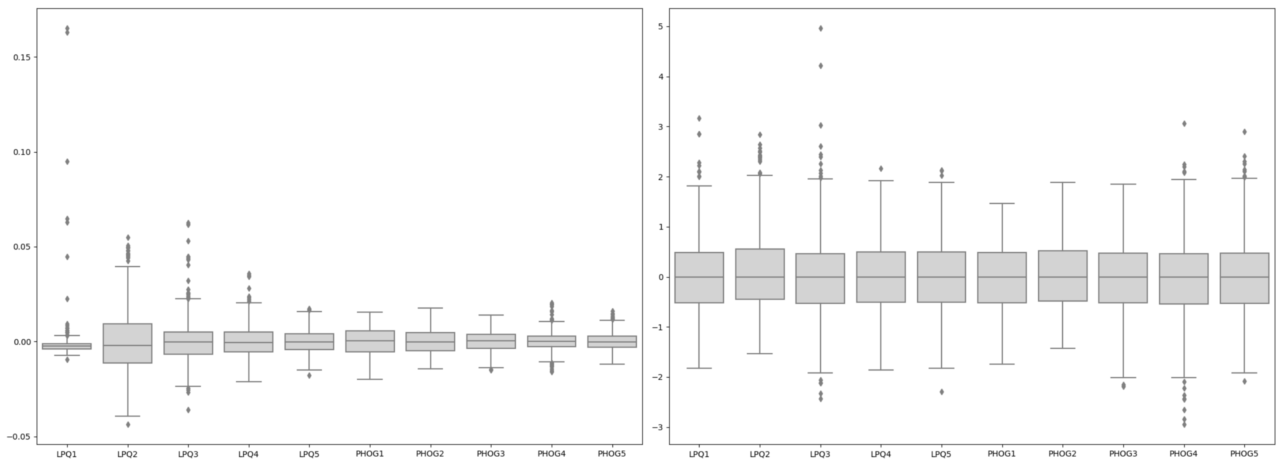

Fig. 1. Distribution of the unprocessed features (Left) and distribution of the processed features (Right)

The process has removed 68 patterns from the dataset without introducing significant imbalance. 11% of the remaining patterns are from the smallest subset, Class 2, compared to 14% if the dataset if perfectly balanced. In addition, the distributions of the original and the new datasets are shown in Figure 1. The dataset with processed features has less outliers and similar interquartile range compared to its unprocessed counterpart.

Labels are encoded as equilateral vectors (Table 1). A model using sparse one-hot encoding of 7 classes generates low error signals by pushing the output units to all 0, causing difficult learning. Equilateral encoding reduces the output dimension by one and all target units hold non-zero values, which would receive higher error signals upon incorrect predictions [6]. Cosine distance can be computed between a predicted vector and each of the equilateral vectors, and the corresponding label of the nearest equilateral vector is retrieved.

Label	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6
1	0.19	0.32	0.38	0.40	0.42	0.43
2	0.81	0.32	0.38	0.40	0.42	0.43
3	0.50	0.85	0.38	0.40	0.42	0.43
4	0.50	0.50	0.87	0.40	0.42	0.43
5	0.50	0.50	0.50	0.89	0.42	0.43
6	0.50	0.50	0.50	0.50	0.89	0.43
7	0.50	0.50	0.50	0.50	0.50	0.90

Table 2. 7-class equilateral encoding

The processed dataset is randomly split into test and train-validation subsets. Only the train-validation subset is involved in training and hyper-parameter tuning and 20% of data is left out for testing and evaluation. The train-validation subset if further randomly split for the 2-fold cross-validation approach suggested by the strictly person independent protocol [1].

2.2 Feedforward and bidirectional neural network

Model Architecture Both models consist of 2 hidden layers and can take arbitrary number of features as input which by default all 10 features are used. 2-hidden-layer architecture can learn convex or concave decision regions with arbitrary complexity [4]. The FF model forms a baseline to assess the effectiveness of the BDNN model, and its ability of learning is restricted to allow comparison. While more hidden units usually result in better model predictability [4], the numbers of hidden units are set to 16 and 32. The BDNN paper suggests that a model should not use more hidden units than the input and output units [3]. To achieve maximum learning ability under the constraint, the number of hidden units for both hidden layers is set to the number of input units.

Training Direction Both models are trained fowared and backpropagate the errors to update weights and biases. The BDNN model takes another reverse step, where the predicted vectors in the forward step are used as input and the input vectors are used as output. An additional set of bias parameters are included in the BDNN model architecture ot allow reverse learning. The parameters are initialised using uniform distribution $U(-k, k)$ where

$$k = \sqrt{(1/(number\ of\ hidden\ units))}$$

This initialisation takes the same approach as that for initialising the weights and other bias parameters of the model.

One-to-one relation The BDNN paper emphasises that the prediction model requires one-to-one relationship between the input patterns and the output classes [3]. Sequentially generated values are attached as extra nodes to the equilateral vectors to make each vector unique to the input.

Training setting To compare the performance, most of the hyperparametes have to be predefiend. With 242 patterns in each training set, the batch size is set to 50 for smoother converging steps yet to avoid being caught in local minimum [4]. The FF model first converges at 20 epochs per fold (total 40 epochs) with 0.01 learing rate. Increased number of epochs causes overfitting as shown by the increased validation loss in Figure 2. The number of epochs per fold required by the BDNN model doubles in order to train weights and parameters in two directions. Other hyperparameters of the BDNN model should be the same as those of the FF model for comparison.

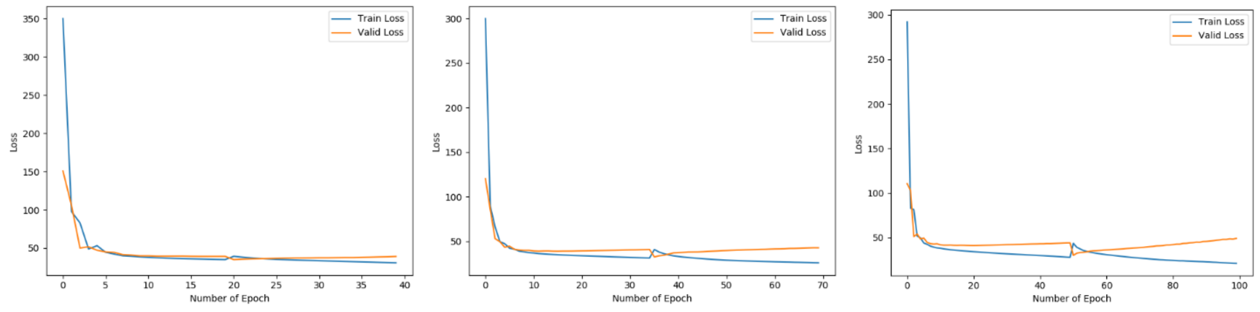

Fig. 2. FF model training loss for number of epochs per fold: 20 (Left), 35 (Middle), 50 (Right)

2.3 Genetic algorithm (GA) for feature selection

Chromosomes and Randomness Setting Each chromosome of an individual is represented by 10 binary bits, and each bit represents the selection of an input feature. A chromosome has 10% mutation probability and among those chromosomes to be mutated, each bit has 10% probability of being flipped [7]. Two chromosomes have 80% probability to exchange their binary bits [7]. One-point crossover is chosen as the recombination technique for fast computation, investigation on the performance of other crossover techniques remains interesting.

Fitness and Reproduction Test accuracy of the FF model (same settings as in the previous section) for each individual set of features selected is the fitness of that individual. Each generation is set to have a population consist of 100 such individuals [7]. For binary strings of length 10, there are 2^{10} possible chromosomes in total. Considering the randomness introduced by one-point cross over, at least 20 generations, each having 100 individuals, are required to cover the 2^{11} possible combinations.

3 Results and Discussion

Each model is trained and tested 50 times to get the average prediction accuracy. With small training dataset, taking the average of models' performance helps to reduce the bias caused by random initialisation [4]. An instance of dataset splitting is saved and loaded to ensure that models are trained with the same data to allow comparison. The GA selects features: LPQ2, LPQ4, LPQ5, PHOG2, PHOG3, PHOG5. The same set of features are used to train BDNN+GA.

The results are generally better than 14% accuarcy for classifying 7 classes by random chance and improve slightly compared to the 19% benchmark [1]. The FF model has a higher accuracy compared to the BDNN and both models are better than the benchmark. This shows that BDNN does not improve the predicition accuracy for this classification task.

The FF model trained with selected featues improves noticeably while the BDNN model trained with the same features impoves a little. This shows that the same set of features does not always improve performance regardless of the model architecture.

	Benchmark	FF	FF+GA	BDNN	BDNN+GA
accuracy	19.0	23.8	28.69	20.3	20.5

Table 3. Test accuracy for different models

Precison and recall are used to evalutate the performance of FF and BDNN besides prediction accuray. The values are calculated by the following formula using confusion matrix of a model which has an accuracy close to the average.

$$Class\ Wise\ Precision = tp/(tp + fp)$$
$$Class\ Wise\ Recall = tp/(tp + fn)$$

Precisions indicate how many data with predicted as an emotion is actually labelled as that emotioin. Recalls indicate how many data labelled with an emotion is predicted accordingly.

	Emotion	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
Benchmark	Precision	0.17	0.15	0.20	0.28	0.22	0.16	0.15
	Recall	0.21	0.13	0.18	0.29	0.21	0.16	0.12
FF	Precision	0.19	0.25	0.29	0.33	0.33	0.15	0.25
	Recall	0.38	0.13	0.28	0.24	0.17	0.29	0.18
BDNN	Precision	0.15	0	0.1	0.23	0.33	0.25	0.20
	Recall	0.44	0	0.06	0.32	0.25	0.71	0.12

Table 4. FF and BDNN class wise evaluations

The FF model have generally higher precisions and recalls compared to the benchmark. With higher accuracy, these evaluations show that the FF model is more effective in performing the classifcation task compared to the benchmark model.

The precisions and recalls of the BDNN model reveal imbalance of dataset. The Disgust emotion, which is labelled as 2, corresponds to the smallest subset of data. The model might have reduced the training loss by always predicting other classes, resulting in better prediction accuracy overall but 0 precision and recall values for the Disgust subset.

By investigating in the predictions made during each batch of learning, the ineffectivenss could also be caused by difficult learning. The model alwasy tries to predict the mean values of the output vectors to minimise the total loss. It is less stable compared to the FF model in terms of the prediction accuracy. Test accuracies higher than the benchmark and random classificaton can only be achieved with good random initializations. In addition, a large amount of loss is caused by the predictionis of extra node values, which are generated with no relation to the input or output vectors. This suggests that a better generation technique should be further researched on.

4 Conclusion and Future work

In conclusion, the FF model could be used to classify SFEW data and the BDNN model is less effective in overcoming dataset imbalance. Other crossover and mutation operations used by the genetic algorithm can be tested on. The original SFEW images and other encoding techniques can be future work as a recent convolutional network approach has shown its ability to achieve 56.4% accuracy on the original data [8]. In addition, BDNN without input-output relation restricts or extra nodes could be implemented and compared with the current models.

References

[1] A. Dhall *et al*, "Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark," in 2011, . DOI: 10.1109/ICCVW.2011.6130508.

[2] A. Dhall *et al*, "Emotion recognition using PHOG and LPQ features," in 2011, . DOI: 10.1109/FG.2011.5771366.

[3] A. F. Nejad and T. D. Gedeon, "Bidirectional neural networks and class prototypes," in 1995, . DOI: 10.1109/ICNN.1995.487348.

[4] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning.* Cambridge, Massachusetts: The MIT Press, 2016.

[5] M. M. Breunig *et al*, "LOF: identifying density-based local outliers," *SIGMOD Record,* vol. 29, *(2),* pp. 93-104, 2000.

[6] T. Masters, Practical Neural Network Recipies in C++. 1993.

[7] O. H. Babatunde *et al*, "A Genetic Algorithm-Based Feature Selection," in 2013.

[8] K. Wang *et al*, "Region Attention Networks for Pose and Occlusion Robust Facial Expression Recognition," *IEEE Transactions on Image Processing,* vol. 29, pp. 4057-4069, 2020.