Image Classification of Static Facial Expressions in the Wild Based on Bidirectional Neural Networks (Based on Pytorch)

Shixuan Liu[1]

[1] Research School of Computer Science, Australian National University, Australia
u6920173@anu.edu.au

Abstract. Quality data recorded in varied realistic environments is vital for effective human face related research. The experiment will be based on a static facial expression database, which is the "Static Facial Expressions in the Wild" (SFEW) extracted from the temporal facial expression database.[1] The bidirectional neural network technology is to train the neural network in reverse direction to achieve the purpose of strengthening the neural network. In this article, I use a bidirectional neural network to enhance the image classification model and compare it with the ordinary back propagation neural network and convolutional neural network based on Softmax. The results show that the bidirectional neural network model is less effective than back propagation neural network and convolutional neural network. The reason may be there is no one-to-one mapping between input and output in reverse training and there are noises from the wild environment in the facial expression database.

Keywords: Bidirectional Neural Network, Image Classification, Convolutional Neural Network

1 Introduction

Facial expression is one of the most effective way to recognize a person's emotional state and intention. These facial expressions are produced by changes in human facial muscles, and these changes convey personal influence to the observer. The experiment will be based on a static facial expression database " Static Facial Expressions in the Wild " (SFEW) which was developed by selecting a framework from AFEW. Extract the frame from the AFEW sequence and mark it based on the sequence label. Overall, SFEW is marked as six basic expressions, including anger, disgust, fear, joy, sadness, surprise and neutral class. [1]

Most neural networks predict tasks based on input data and perform classification or pattern recognition, which are widely used in the real world. However, given output data, these neural network models will not be able to produce any reasonable input data unless another network is specifically trained for the task. Bidirectional neural network differs from most of neural network models in that it enables the model to remember given input patterns and output vectors. And reverse training the input mode through the output vector to strengthen the conversion between the input mode and the output vector. [2]

The dataset included a total of 675 different facial expressions in SFEW. The dataset contains the categories of these facial expressions and different figures of facial expressions. For convolutional neural network, the figures are used as input. For bidirectional neural network and back propagation, The inputs are extracted by the first 5 main principal components of the local phase quantization (LPQ) features and the first 5 main principal components of the "gradient histogram pyramid" (PHOG) features.

2 Method

2.1 Data Extraction by LPQ and PHOG

LPQ is based on calculating the short-term Fourier transform (STFT) on the local image window, to a certain extent, the blur and lighting remain unchanged, which can effectively describe the feature information of face images. [3] Directional gradient histogram pyramid (PHOG) [6] descriptor also shows good performance in object recognition. [4] Then using the principal component analysis extracts the top 5 features of LPQ and PHOG respectively. I combine a total of ten features of the LPQ operator and the PHOG operator for classification training of the bidirectional neural network and the back propagation neural network. Then using the principal component analysis extracts the top 5 features of LPQ and PHOG respectively.

2.2 Method for Implement Back Propagation Neural Network Image Classification

The experiment will implement Softmax classification based on traditional back propagation neural network. The back propagation neural network consists of 1 input layer, 1 hidden layer and 1 output layer. The activation function of the hidden layer uses the Relu activation function. The activation function of the output layer is Softmax activation function.

It is trained by error back propagation and use cross entropy calculate loss function and update gradient. The ten features input in the network and the one category of total seven will be output.

2.3 Method for Implement Bidirectional Neural Network Image Classification

The implement of bidirectional neural network bases on the traditional back propagation neural network. In general, the back propagation neural network only needs to be trained by forward passing and error back propagation. In bidirectional neural network, the forward passing is like the back propagation neural network. The backward passing need to firstly convert the original output of forward passing into an input vector, which dimension is as same as the number of total categories. In the input vector, I enhance the dimension corresponding to its origin category. The backward passing does not have activation function and is like regression problems to predict the input patterns of forward passing. In the experiment, I set two neural networks and their layers structure are reverse in total. After one network has finished training, it would share its weights (excluding bias) with the other and the other network would continue to train. As figure 1 shows below, the left image in Figure 1 means forward training, which will use the connection weights between the input layer and the hidden layer and the weights between the hidden and output layer. The picture on the right means that in reverse training, the same weight is used between the input layer and hidden layer and the hidden layer and output layer of forward training. The same weight is used between the hidden layer and the output layer and the input layer and hidden layer of the forward training. These two networks do not share bias. For the forward propagation, the network uses cross entropy loss function and for backward propagation, the network uses MSE loss function. The reason why we use two different loss function is that for forward propagation, this is a multi-classification problem and for backward propagation, it is more like a linear regression problem. Finally, I applied the error back propagation technique in both the reverse and forward directions to adjust the weight matrix of the network.

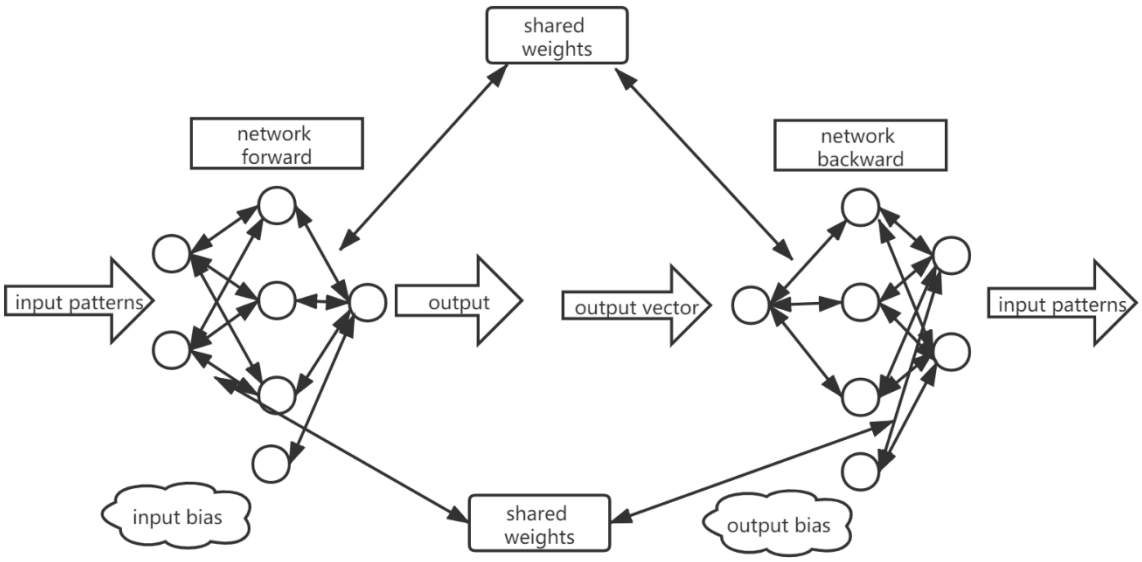

Figure 1. How to implement a bidirectional neural network

2.4 Method for Implement Convolutional Neural Network Image Classification

The experiment will implement Softmax classification based on traditional convolutional neural network. The convolutional neural network consists of 4 convolutional layers and 2 linear layers for output the result of classification. The image input uses RGB color images. Rescale the image into [28, 28] size and input RGB three channels of the original image as the input image. For the input image pixel component is [0, 255], we need to normalize to [0, 1] in order to facilitate calculation. The convolution layer performs convolution calculation on the output image of the previous layer by the weighted value of the convolution kernel (weight parameter) of this layer and adding the bias, obtains the feature images through Relu activation function, and then normalizes the feature images. In order to fully connect with the traditional multi-layer perceptron MLP, each pixel of all Feature Images in the upper layer is expanded in order and arranged in a row. The last layer is the classifier, which uses Softmax function to classify.

2.5 Optimization of the Neural Network

When applying the model, it is necessary to use some methods to increase the accuracy

Normalize the Input Data

Normalization turns the data into decimals between (0, 1) or (1, 1). It is mainly proposed for the convenience of data processing. It is convenient and fast to map the data to the range of 0 to 1 for processing. Another advantage is that turning dimensional expressions into non-dimensional expressions makes it easier for indicators of different units or magnitudes to be compared and weighted. Normalization is a way to simplify calculation, that is, a dimensional expression, after transformation, is transformed into a dimensionless expression and becomes a scalar quantity. In the experiment, the values of LPQ and PHOG dataset are too small and would cause large deviation and the image pixels should be normalized from [0, 255] to [0, 1] for Softmax function, so normalization is a good way to preprocess dataset.

Use Adam Optimizer

Adam, a stochastic objective function optimization algorithm based on one-step degree, which is based on adaptive estimation of low-order moments. This method is easy to implement, has high computational efficiency, requires very little memory, and is very suitable for problems with large data or parameters. This method is also applicable to the problem of non-stationary targets and very noisy and sparse gradients. Hyperparameters have intuitive explanations and usually require very little adjustment.[5]

Adjust Hyperparameters

A better model can be obtained by adjusting the hyperparameters. In the experiment of back propagation neural network and bidirectional neural network, we adjusted the number of iterations, the number of hidden layers, the number of neurons in each layer, and the learning rate. In order to prevent overfitting and underfitting, we compared the loss functions of the models under different hyperparameters. In the experiment of convolutional neural network, the main work is to adjust the number of convolutional layers, image size after normalization and kernel size. Finally determine a set of models with optimal hyperparameters.

Adjust the Method of Changing Direction

In the bidirectional experiment, the forward passing and backward passing process alternately, so it is important to decide when it should change the propagation direction. At first, I change the direction through fixed times and it seems performs not good. Because if the backward network has been trained for numbers of epochs, it will cause large error. Then I set a threshold for backward propagation loss function, once the value of loss function is lower than the threshold, the network will change its direction to forward passing. This method has a greater performance than fixed times changing direction.

2.6 Evaluation

To evaluate the BDNN, BPNN and CNN models, four metrics called SPI baseline [1] were used in this paper. They are accuracy, precision, recall and specificity. Their definitions are listed below.

$$Accuracy = \frac{tp + tn}{\mathrm{tp} + \mathrm{tn} + \mathrm{fp} + \mathrm{fn}}$$

$$Precision = \frac{tp}{\mathrm{tp} + \mathrm{fp}}$$

$$Recall = \frac{tp}{\mathrm{tp} + \mathrm{fn}}$$

$$Specificity = \frac{tn}{\mathrm{tn} + \mathrm{fp}}$$

Here, tp = true positive, fp = false positive, fn = false negative, and tn = true negative. There is an example of previous work in Figure 2 [1]. We apply these metrics to evaluate the performance of our models on the whole dataset for each face expression and compare with previous work. Besides, we also plot the losses of each epoch for back propagation neural network and bidirectional neural network. Because the convolutional neural network only needs few epochs much less than back propagation neural network and bidirectional neural network, we do not discuss about its loss polyline.

Emotion	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
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
Specificity	0.48	0.66	0.64	0.51	0.61	0.60	0.66

Figure 2. Previous Work

3 Results and Discussion

3.1 Result of Back Propagation Neural Network

After the back propagation neural network is optimized, the SPI baseline of the model for the test dataset shows as below Figure 3. This shows that compared with previous work of the precision, recall and specificity, it shows a little better than the previous work although the neutral expression shows worse. This may benefit of the principal component analysis which extracts the main components of the figures and decrease the noises caused by the wild environment.

	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
precision	0.25	0.29	0.27	0.23	0.11	0.35	0.22
recall	0.33	0.27	0.33	0.17	0.07	0.42	0.25
specificity	0.71	0.74	0.71	0.68	0.55	0.71	0.62

Figure 3. Evaluation on BPNN based on the SPI protocol

3.2 Result of Bidirectional Neural Network

After bidirectional neural network is optimized, the SPI baseline of the model for the test dataset shows as below Figure 4. This shows that compared with previous work and back propagation neural network of the precision, recall and specificity, it shows worse than the previous work and back propagation neural network. This shows that the backward propagation may not enhance the classification of the image. On the contrary, it makes the model not accuracy. This may cause by that we have not find the one to one mapping function for input vectors when implement backward training so that it may have some bias about the information in the network.

	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
precision	0.29	0.08	0.15	0.24	0.1	0.32	0.43
recall	0.29	0.11	0.11	0.31	0.18	0.24	0.18
specificity	0.57	0.76	0.6	0.66	0.62	0.51	0.62

Figure 4. Evaluation on BDNN based on the SPI protocol

3.3 Result of Convolutional Neural Network

After convolutional neural network is optimized, the SPI baseline of the model for the test dataset shows as below Figure 5. This shows that compared with previous work and both back propagation and bidirectional neural networks of the precision, recall and specificity, it shows the best in the average of this three SPI baseline. This may cause by that after rescaled and normalized, the figures lose some information and meanwhile decrease some noises.

	Angry	Disgust	Fear	Happy	Neutral	Sad	Surprise
precision	0.5	0.36	0.58	0.21	0.29	0.45	0.25
recall	0.24	0.4	0.28	0.23	0.56	0.5	0.3
specificity	0.73	0.77	0.69	0.82	0.85	0.79	0.75

Figure 5. Evaluation on CNN based on the SPI protocol

3.4 Comparison of the losses of BPNN and BDNN

As we could see the figure 6 below, both of the plots of the loss function are not stable. These results indicate that both of the model does not fit the dataset well. For back propagation neural network, it may because the loss has already converged although the accuracy is not good enough. For bidirectional neural network, it may because during the backward training the input vector of none one to one mapping function causes the information bias and the forward training attempts to fix it. Therefore, the dataset of the ten features may should use more complexity model.

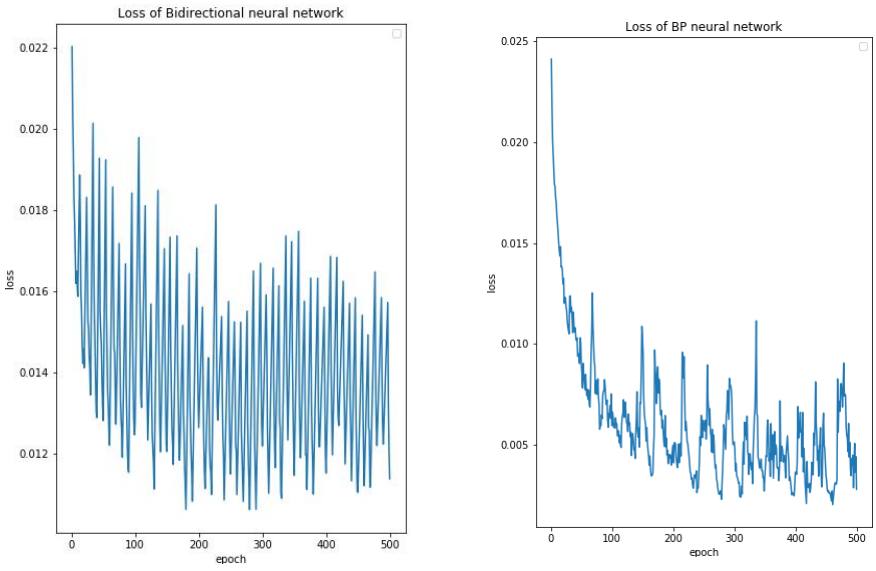

Figure 6. Comparison of the losses of BPNN and BDNN

3.5 Result of the Comparison with the Previous Paper

From previous paper [1], the classification accuracy for JAFFE is 69.01% for LPQ and 86.38% for PHOG. For SFEW, it is 43.71% for LPQ and 46.28% for PHOG. And in the figure 7, the accuracy is about 20% for back propagation neural network and bidirectional neural network and 36% about convolutional neural network. This indicates that my model of using LPQ and PHOG principle features does not perform well compared with previous experiment in accuracy. There may be three reasons for that. Firstly, the backward propagation process does not use one to one mapping function. Second, the structure of the model is not complexity enough to classify the image. Third, there may have more effective method to preprocess the dataset of LPQ and PHOG. And for convolutional neural network, it performs better than back propagation neural network and bidirectional neural network. This may because although the principle components of LPQ and PHOG already have much information, the image trained through convolutional neural network may still perform better. Besides for the SPI baseline, the back propagation neural network and convolutional neural network have a better performance than previous work. The reason may be that the extracted features of both images and principle components discard some of the information while they also discard some of the environmental noise. So it may be helpful for the SPI baseline.

It seems the one to one mapping function may offer the most support in the experiment. For most of the situation in real world, many images could map for one category. However, in reverse, one category would map for many images in general. So if we could not construct an one to one mapping function, when we train the dataset by backward passing, one input vector which is the output in forward passing would map to many output and that makes the neural network hard to distinguish the which output will fit the input vector. But in this experiment, it is hard to find a suitable one to one mapping function for the dataset. We could also inspire that for the figures of face which have few environment noises, it is easy to classify the face expressions. However, for the figures which exposed to the wild environment, due to the complexity factors, it is hard to perform classification well on them.

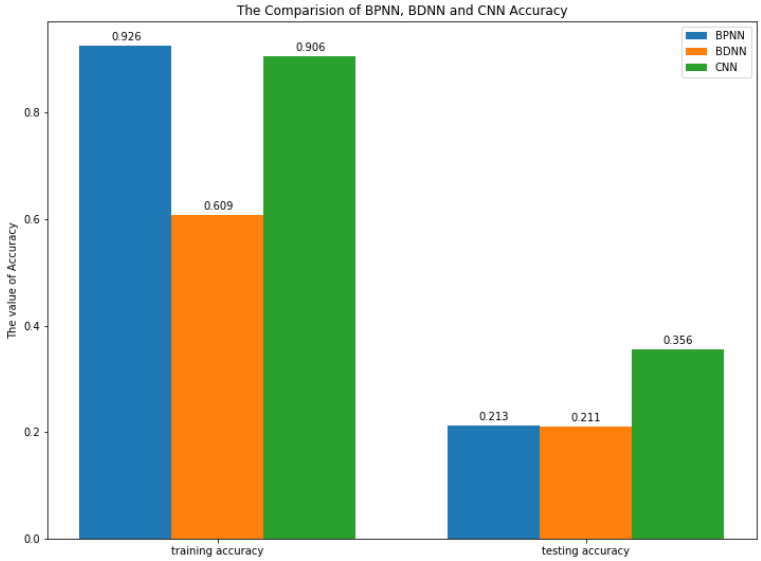

Figure 7. Training and testing accuracy of three NNs

4 Conclusion and Future Work

The result of our experiment does not have good effect. In the further work, we need to make more improvement. We know that for the figures influences by nature conditions, due to the complexity factors, it is hard to perform classification well on them. Therefore, for the SFEW dataset, it is necessary to find a method to eliminate the environment efficiently. As the one to one mapping function could offer a better effect for us, we could explore the generalized method to construct this function for bidirectional neural network. At present work, the bidirectional neural network also has some advantages than the ordinary backward propagation neural network. Our method of training a bidirectional neural network to learn both forward and backward tasks simultaneously will produce a more powerful neural network. The two-way learning aggregation gradient tends to be flat. We believe that this may make the network less susceptible to noise and provide better generalization capabilities [2]. Another advantage of using a bidirectional neural network is that the ideal change in output during the training of the neural network can provide an appropriate change in the input value. In the exploration of future work, a well-trained bidirectional neural network will help us solve the bottleneck of neural network reception and use.

References

1. Abhinav, D., Roland, G., Simon, L., Tom, G.: Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In: 2011 IEEE International Conference on Computer Vision Workshops, pp. 2106-2112, Barcelona (2011)
2. A. F., Nejad, T. D. Gedeon: Bidirectional neural networks and class prototypes, In: Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, pp. 1322-1327 vol.3, Perth, WA, Australia (1995).
3. V., Ojansivu, J., Heikkil.: Blur Insensitive Texture Classification Using Local Phase Quantization. In: Proceedings of the 3rd International Conference on Image and Signal Processing, ICISP'08, pages 236–243 (2008).
4. A., Bosch, A., Zisserman, X., Munoz.: Representing Shape with a Spatial Pyramid Kernel. In: Proceedings of the ACM International Conference on Image and Video Retrieval, CIVR '07, pages 401–408 (2007).
5. Diederik, P., K., Jimmy, B.: Adam: A Method for Stochastic Optimization. In: The 3rd International Conference for Learning Representations, San Diego (2015)
6. N., Dalal, B., Triggs.: Histograms of oriented gradients for human detection. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, CVPR'05, pages 886–893 (2005).