Comparison the Results of Human Face Matching based on Bidirectional Neural Network and Siamese Neural Network

Yanzhi Ren[1]

[1] Research School of Computer Science, Australian National University, Australia
u6920387@anu.edu.au

Abstract: The bidirectional neural network is a special neural network which can be trained in two directions: from inputs to outputs, from outputs to inputs. In a bidirectional neural network, weights related to each hidden layer are shared whatever they are forward or backward training, however, the bias does not share when training. The traditional Siamese neural network is also a kind of weights shared neural network based on convolutional neural network, the input of this model is two images, then extracting the features of two images based on weights shared convolutional neural network and calculate the similarity of two images based on extracted features. The experiment is based on dataset related to sparse photographs of historic person, which is 'Facial Feature Markers'(FFMS). In this article, I build a bidirectional neural network model and a novel Siamese neural network which accept two images as input and output a label (0 or 1) to indicate match or un-match, then analysis and compare the results. The results indicate that the bidirectional neural network is much less effective than Siamese Neural Network, which probably because of the lack of a one-to-one mapping between input and output in backward training and the fact that Siamese Neural Network is more suitable for such face matching problem.

Keywords: Bidirectional Neural Network, Siamese Neural Network, Human Face Matching

1 Introduction

Face recognition and face matching are very hot fields in recent years, and the dataset I chose is about face features, which can be used in these fields very effectively. The dataset is derived from a subset of the National Archives of Australia "Bonds of Sacrifice exhibition" (Australia, n.d.), a collection of evocative WWI military personnel historic photographs [1]. We aim to compare and match an unknown natural person in a sparse historical photograph with a potential identical historical person, and if the match is successful that unknown person may have a name and be remembered by future generations.

Most neural networks can be applied to perform classification, pattern recognition or prediction tasks based on input data. However, these neural network models cannot produce any plausible input data if just given output data unless another network is trained specifically for that task. Bidirectional neural networks can remember input patterns as well as output vectors, given either of them. The connection between the input and the output is much closer through this bidirectional training [2]. Siamese neural network is created to compare the similarity of two images, it extracts the features of two images using the weights shared convolutional neural network, therefore, the extracted features would map to the same target space, and a simple distance (Euclidean distance, etc.) is used to compare the similarity in the target space [3].

The dataset comprises 12 sets of 3 photos each, the first two photographs (Photographs A and B) are of the same individual, and the third photograph (Photograph C) is of a different individual. FFMS provides three possible combinations in each set of 3 photographs. There are totally 36 rows, each row provides the x and y coordination of 28 important facial features (14 for each person) [1]. The figure 1 below shows the example of 14 facial features of a person in FFMS.

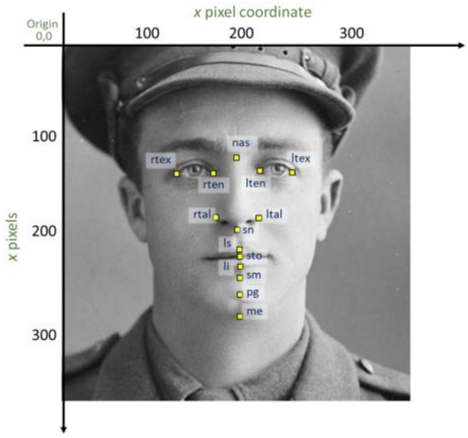

Figure 1. Example of 14 Facial Features of a Person [1]

The form of dataset used for bidirectional neural network is showed in table 1, the shape of the dataset would be 36*57 (56 coordinates of 28 facial features and 1 label).

Facial features of person A	Facial features of person B	1(match)
Facial features of person A	Facial features of person C	0(unmatch)
Facial features of person B	Facial features of person C	0(unmatch)

Table 1. Form of Dataset used for Bidirectional Neural Network

The form of dataset used for Siamese neural network is showed in table 2.

Photograph of person A	Photograph of person B	1(match)
Photograph of person A	Photograph of person C	0(unmatch)
Photograph of person B	Photograph of person C	0(unmatch)

Table 2. Form of Dataset used for Siamese Neural Network

2 Method

2.1 How to implement a Bidirectional Neural Network

In my experiment, I build two simple neural networks to perform a bidirectional neural network.
For forward neural network, it consists of 1 input layer, 1 hidden layer, 1 output layer and 1 dropout layer. I apply Relu activation function to the hidden layer and sigmoid activation function to the output layer. Then it is normal neural network based on back-propagation. It is trained by error back propagation

and it uses cross entropy as loss function to calculate loss and update the gradient. The dimension of input and output would be X and Y.

For the backward neural network, it consists of 1 input layer, 1 hidden layer and 1 output layer. There is no activation function, and it is more like a regression task to predict the input of forwarding neural network. It is also trained by error back propagation and it uses MSE as loss function to calculate loss and update the gradient. For the input for backward training (Back-inputs in figure 2), It is the new dimension added based on the original class. The dimension of input and output would be Y and X.

Forward and backward neural network will exchange their weights whenever the direction of the training changes. As shown in the figure 1, the forward network has 2 sets of weights, which are weights between input layer and hidden layer and weights between hidden layer and output layer. The backward network also has its own 2 sets of weights, which are weights between input layer and hidden layer and weights between hidden layer and output layer. These weights will share with each other whenever direction changes. In other words, when direction changes from forward training to backward training, the weights used in backward neural network are as same as the weights of the pervious forwarding training, when direction changes again, the weights of backward neural network will be used in the forwarding training. (The biases are not shown in Figure 2 for clarity)

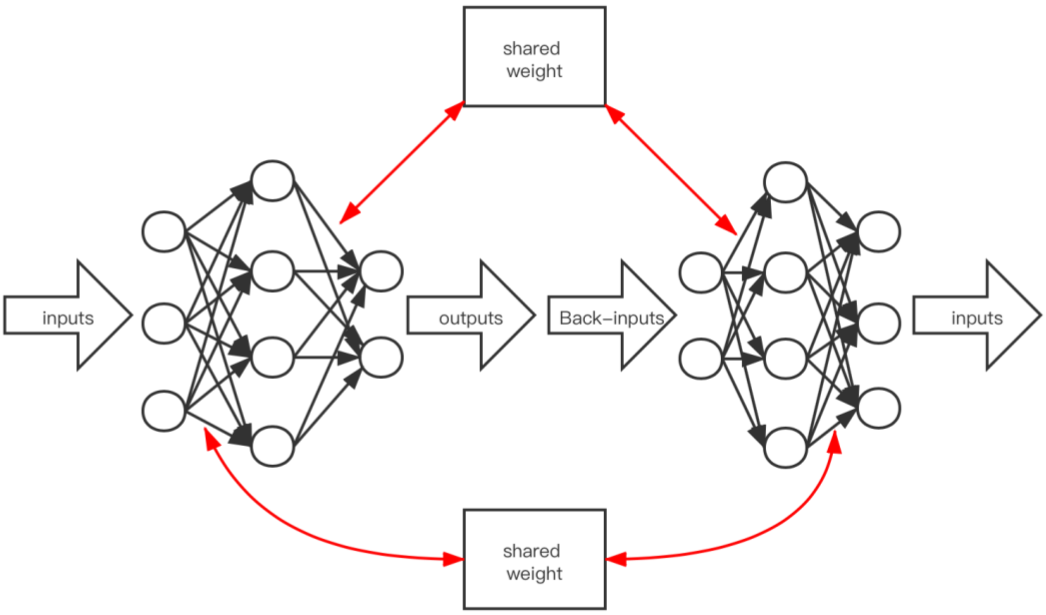

Figure 2. Topology of Bidirectional Neural Network

The parameters of the forward and backward neural network are showed in the table 3.

Layer	Parameter for neurons	activation function
Hidden layer for foward	(56,100)	Relu
Output layer for foward	(100,2)	Softmax
Hidden layer for backward	(2,100)	None
Output layer for backward	(100,56)	None

Table 3. Structure and parameter of Bidirectional Neural Network

2.2 How to Optimize the Bidirectional Neural Network

Preprocessing the Dataset

First, data augmentation: Manually adding 36 rows, in each row, the two sets of corresponding feature points to be matched came from the same person and labeled with 1; Shifting the coordinate of facial feature points to enlarge the dataset to prevent the neural network from overfitting. It would still represent the same person after shifting. Then normalize the dataset by subtracting the mean and dividing by the standard deviation. It can ensure that different features have the same value range, then the network does not need to learn to adapt to different distributions in each iteration, which will greatly reduce the training speed of the network, and the gradient descent can converge quickly. Last, adding new dimensions based on the original class, which is 2 (matched and unmatched) and these 2 new dimensions would be the input of backward neural network (If label of a row is 1, then the value for matched dimension would be 1 and the value of unmatched dimension would be 0, otherwise).

Choosing Hyperparameters, Optimizer and Loss Function

Hyperparameters can affect the performance of a model greatly. A good neural network can be built with an optimal set of hyperparameters. In my experiment, I adjusted a lot of hyperparameters: learning rate, number of epochs, number of hidden layers, number of neurons in each layer and activation functions in each layer. At last, I found a set of optimal hyperparameters. The hyperparameters in the experiments is: learning rate 0.01 with 300 number of epochs. The hyperparameters for bidirectional neural network is showed above in table 3.

I compared some optimizers like SGD optimizer, Adam optimizer and RMSProp optimizer. I chose the Adam optimizer at last because it performed best. The reasons are as following: high computational efficiency, very few memory requirements, the diagonal rescale of the gradient is invariant, suitable for very sparse gradient problems. And it is the most common optimizer people used nowadays.

I compared some loss functions and chose Cross entropy loss function in forward neural network and MSE loss function in backward neural network.

Adjusting Pattern of Changing Training Direction

If the network training switches direction after every pattern instead of every epoch, the results are much improved [3]. Changing training direction is very important in bidirectional neural network, I came up with three patterns, first one is to train a fixed number of times and then change direction, and the second one is to set a threshold, and change direction whenever loss is less than the threshold, and the last one is to change training direction randomly. After many experiments, I found that the first pattern worked best, and the third random pattern worked worst. Therefore, I used the first pattern in my experiment.

2.3 How to implement a Siamese Neural Network

Because the structure and weights of the two convolutional neural networks in the Siamese network are the same, I use a convolutional neural network which can accept two images at the same time to satisfy this characteristic. In this convolutional neural network, two images are processed the same time, and then their features are extracted. As shown in Figure 3, the two convolutional neural networks are the same actually, drawling like this aims to show the process of two inputs. The convolution neural network is used to extract the image features. Then the two feature vectors are subtracted and mapped into two-dimensional vectors as the final output.

It is trained by error back propagation, and it uses cross entropy as loss function to calculate loss and update the gradient.

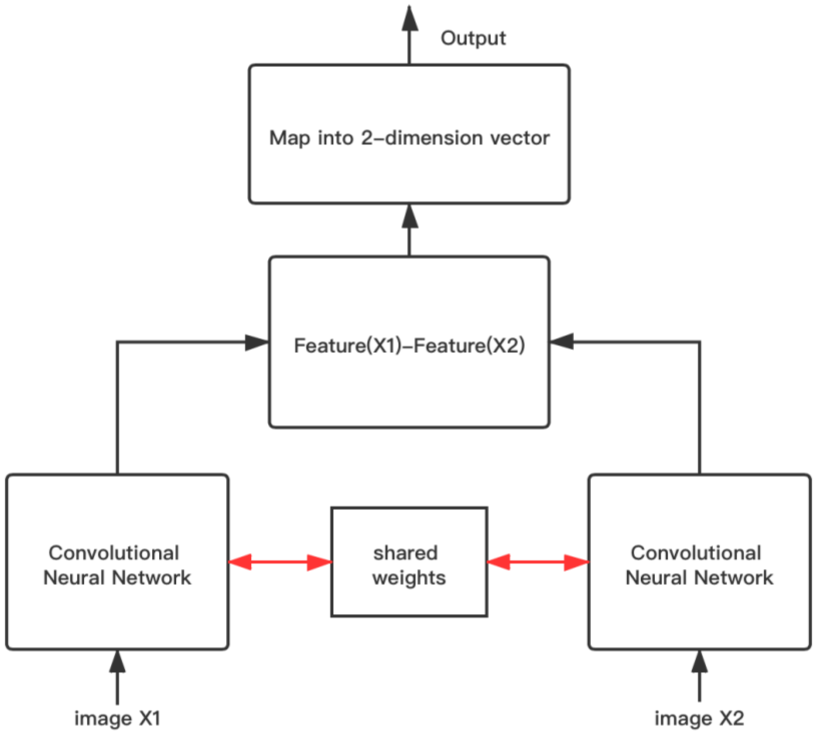

Figure 3. Topology of Siamese Neural Network

The convolutional neural network consists of 3 convolutional layers and 5 linear layers for output the result of matching. The image input uses RGB color images. The table 4 below shows the structure for 3 convolutional layers.

Layer	Parameter for Conv2d	activation function	Parameter for BatchNorm2d	Parameter for MaxPool2d
Conv Layer 1	(3,16,3,2,1)	Relu	16	(2,1,0,1,False)
Conv Layer 2	(16,32,3,2,1)	Relu	32	(2,1,0,1,False)
Conv Layer 3	(32,64,3,2,1)	Relu	64	(2,1,0,1,False)

Table 4. Structure and Parameter of Three Convolutional Layers

To fully connect with the traditional MLP (multilayer perceptron), each pixel of feature image in the final convolution layer is expanded in order and arranged in a row. Then the feature vectors of the two images are obtained through the first three linear layers, and then the two feature vectors are subtracted, and put the result into the final two linear layers to map into 2D vectors to complete the classification (matched or unmatched). The table 5 below shows the structure and parameters for 5 linear layers.

Layer	Parameter for neurons	activation function
Layer 1	(36864,1000)	Relu
Layer 2	(1000,500)	Relu
Layer 3	(500,16)	None
Layer 4	(16,8)	Relu
Layer 5	(8,2)	Softmax

Table 5. Structure and Parameter of Five Linear Layers

2.4 How to Optimize the Siamese Neural Network

Preprocessing the Dataset

Since the dataset is very small, data augmentation is an effective method to prevent overfitting, I apply data augmentation through these methods: Cropping the photographs to get the image of the center face as new data; Flipping the photographs horizontally; Rotating the photographs and changing the color contrast of the photographs. Next, rescaling all the photographs into [200,200] size then normalize the pixel from range 0-255 to range 0-1 in order to facilitate calculate.

Choosing Hyperparameters, Optimizer and Loss Function

The optimal set of hyperparameters in my experiment is: learning rate 0.0005 with 100 number of epochs; Adam optimizer, cross entropy loss function; The hyperparameters for Siamese neural network is showed above in table 4 and 5.

2.5 Evaluation

In this experiment, five metrics derived from confusion matrix is used to evaluate the results of bidirectional neural network and Siamese neural network, they are accuracy, precision, recall, specificity and F1-score [5].

$$Precision = \frac{tp}{tp + fp} \ , Recall = \frac{tp}{tp + fn}, Specificity = \frac{tn}{fp + tn}$$

$$F1score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \ , Accuracy = \frac{tp + tn}{tp + fn + fp + tn}$$

Meaning of TP, FN, FP and TN is showed below:

TP: true positive (truth is 1, prediction is 1); FN: false negative (truth is 0, prediction is 1); FP: false positive (truth is 1, prediction is 0); TN: true negative (truth is 0, prediction is 0).

3 Result and Discussion

3.1 Result of Bidirectional Neural Network based on FFMS

The test accuracy is 57.89%, which is not an acceptable result for such a binary classification problem, and figure 4 is the loss of bidirectional neural network based on FFMS. Loss figure is sometimes not so good, the reason may be that there is no one-to-one mapping, then the back propagation error is very large, resulting in the returned weights are not very suitable.

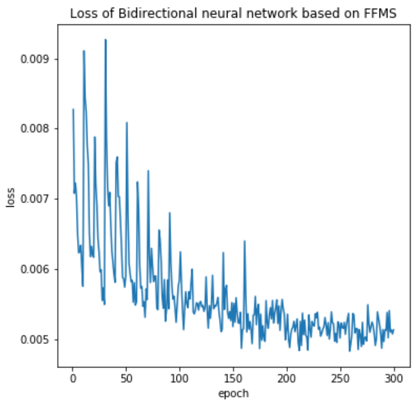

Figure 4. Loss of Bidirectional Neural Network based on FFMS

3.2 Result of Siamese Neural Network based on FFMS

The test accuracy is 74.07%, which is a very optimal result for such a binary classification problem. And figure 5 is the loss of Siamese neural network based on FFMS. The loss figure has been converted to about 0.042 after about 70 epochs. The reason why the fluctuation looks relatively large is that the ordinate scale is very small.

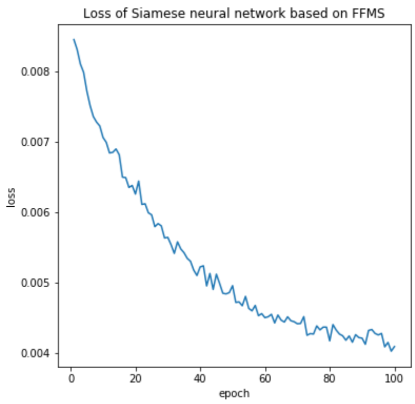

Figure 5. Loss of Siamese Neural Network based on FFMS

3.3 Compare Bidirectional Neural Network with Siamese Neural Network

The table 6 below shows the result of evaluation metrics in these two models. It is obvious that each evaluation metrics of Siamese neural network is significantly larger than that of bidirectional neural network. The performance of my novel Siamese neural network is much better than that of bidirectional neural network. For accuracy, novel Siamese neural network can reach about 74.07%, which is about 17% higher than that of bidirectional neural network. For F1-score, Siamese neural network is around 21% higher than bidirectional neural network. For other 3 metrics, Siamese neural network are higher with around 20% than Bidirectional neural network.

	Accuracy	Precision	Recall	Specificity	F1-score
Bidirectional NN	0.5789	0.4362	0.6833	0.4239	0.5325
Siamese NN	0.7407	0.6333	0.9048	0.6667	0.7451

Table 6. Results of Bidirectional and Siamese Neural Network

3.5 Compare Bidirectional and Siamese Neural Network and with Pervious Paper

From the previous paper [1], The test accuracy for Proportion Dataset based on SVM was 58%, this dataset is generated by proportions based on a reduced FFM set of 10. Accuracy of Bidirectional neural network is 57.89%. There is no significant difference between the result of previous paper and bidirectional neural network, and the results are both not good enough for human face matching problem. As for Siamese neural network, the accuracy is higher than that of previous paper, more specific, about 16% higher.

3.6 Analysis the Result

The results show that bidirectional neural network performs bad compared with Siamese neural network. Analysis from aspect of bidirectional neural network: Firstly, there may be other more meaningful method to preprocess the dataset like compute the angel between facial features. Secondly, the bidirectional neural network model is not complex enough. Lastly, there is no one-to-one mapping between inputs and outputs patterns, which may be the most essential factor. The dimensions of the input and output are too different, then using the parameters of backward training will cause a large error. If we do not have a one-to-one mapping in this image classification problem, it would be hard for backward neural network to distinguish which output fits the given input vector. Analysis from aspect of Siamese neural network: First, the image feature extracted from convolutional neural network is much better than using coordinates of facial features; Besides, Siamese neural network is specially used for human face matching problem, so it would be more suitable for such human face dataset than bidirectional neural network. Therefore, the performance of bidirectional neural network was bad and that of Siamese neural network was pretty good.

4 Conclusion and Future Work

The results of Siamese neural network are acceptable, but some improvements still need to be done, such as optimizing the structure of the whole model or making the network structure more complex or apply more data augmentation, using larger dataset to train a better model. The result of bidirectional neural network is not as good as expected, so some improvements I need to do in the future work. Firstly, I need to expand dataset by other technics, then a better model can be trained using large dataset. Thirdly, I need to find a more optimal pattern of changing training direction although fixed number patter performed best in my experiment. Lastly, bidirectional neural network can bring a lot of advantages because of the two-direction training, and one-to-one mapping between inputs and outputs pattern is extremely essential, however, there is no general method to construct such a one-to-one relationship. So, I'm going to try to figure out how to construct a universal mapping mechanism. If I can build such a mechanism to make bidirectional neural network be more common, then I believe this powerful neural network can help us deal with lots of problems.

References

1. Caldwell, S. (2021) "Human interpretability of AI-mediated comparisons of sparse natural person photos," CSTR-2021-1, School of Computing Technical Report, Australian National University.

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. S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 539–546. IEEE, 2005.

4. T. D. Gedeon: Stochastic bidirectional training, In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 1998, pp. 1968-1971 vol.2, San Diego, CA, USA (1998).

5. Lewis, H.G., Brown, M. 2001," A generalized confusion matrix for assessing area estimates from remotely sensed data", International Journal of Remote Sensing, vol. 22, no. 16, pp. 3223-3235.