Face Matching Based on Bidirectional Neural Network and deep exploration with Convolutional Neural Network (Based on Pytorch)

Hongche Wang [1]

[1] Research School of Computer Science, Australian National University, Australia
u6920817@anu.edu.au

Abstract. This project mainly explores the effects of Bidirectional Neural Network(BDNN) by comparing the face matching result and use Convolutional Neural Network(CNN) to explore in a deeper level. The experiment trains the two neural network structures by using the features in the sparse photos, and then perform mathematical analysis, statistical analysis on these two neural network model. Bidirectional neural network technology is to carry out reverse training of neural network to achieve the purpose of improving the effect of the neural network. CNN avoids the displayed feature extraction, but implicitly learns from the training data, and avoids the complexity of data reconstruction in the process of feature extraction and classification. After a lot of testing and optimization, these two model are evaluated through the loss and accuracy. The results show that the CNN has better results(85.7% accuracy) than the previously published paper [1](75% accuracy) and the BDNN(81.82% accuracy). The reason may be that CNN shares the convolution kernel and adapts to high-dimensional data, so it can better extract the features in the data set.

Keywords : Bidirectional Neural Network , Face Matching , Convolutional Neural Network

1 Introduction

Photographs are used extensively in preserving knowledge of history. Photographs memorialise human history, and also personal memories.However, many people in the photos have no names, and no one can recognize them. Therefore, many historical photos have become orphans, and at the same time we have lost the memory of these historical photos.This is even more problematic for photographs from the 19th and early 20th centuries. So in this project, a way will be found for an unknown natural person to compare himself in a sparse historical photo with (possibly) the same person in another sparse historical photo. And let the two-way neural network complete face matching in sparse photos and complete it in an interpretable way.By identifying people on sparse photos, it helps us restore the memory of these photos to the greatest extent.

Most neural networks based on input data can be applied to real world systems to perform classification, pattern recognition or prediction tasks . Given the output data, however, these neural network models are not able to produce any plausible input data unless another network is trained specifically for that task. A new model of training neural networks is suggested that enables them to make up for the shortcomings of traditional neural networks and to remember input patterns as well as output vectors, given either of them.After obtaining the output vector, train the input vector in reverse according to the output vector to strengthen the connection between the input and output vectors. They may be trained as associative memories and cluster centroid finders and are capable of classification or prediction, in

real world problems. So here I use a bidirectional neural network to classify the face, and use the loss function and model accuracy to evaluate and optimize the model.

As for CNN, Convolutional network is essentially a kind of input to output mapping, it can learn a large number of mapping relations between input and output, without any precise mathematical expressions between input and output, as long as the known The model trains the convolutional network, and the network has the ability to map between input and output pairs. Since the feature detection layer of CNN learns through training data, when CNN is used, explicit feature extraction is avoided, and learning is implicitly performed from training data; in addition, due to the neuron weights on the same feature mapping surface The same, so the network can learn in parallel, which is also a major advantage of convolutional networks over networks that connect neurons to each other. So in this article I use CNN to conduct a deeper research on face matching

The facial feature markers(FFMs) pays particular attention to the capulometric relationship between soft tissue and skeletal structure, as well as the differences between 3-dimensions and 2-dimensions[1].So it has a vital role in matching different faces, it provides many unique features on the faces of different people.Therefore, these features can help to judge whether the person in two photos is the same person to a large extent. Furthermore, intuitively speaking, the performance of facial recognition algorithms using the FFMs dataset is much better, and this intuition has been confirmed by many research papers[4]. So in this project, I use the data set of FFMs to train a bidirectional neural network. And for convolutional neural network, the figures are used as input.

2 Method

2.1 Data Inspection

The FFMs data set comprise 12 sets of 3 images, of which the first two in each set are positively identified as being the same person, and the third photograph is positively identified as not being a match. FFMs data set provided as x,y coordinates in pairs of FFMs each of the three possible combinations in each set of three images (n-1 & n-2, n-2 & n-3, and n-3 & n-1). Output targets are listed in the last column on the right. In the figure below, each two columns represent a facial feature markup (each column is the x coordinate and the y coordinate), and a total of 14 facial feature markups are collected in the data set for every sparse photo.

image pairs	1rtex–x	1rtex–y	1rten–x	1rten–y	1lten–x	1lten–y	1ltex–x	1ltex–y	1nas–x	1nas–y	1sn–x	1sn–y	1rtal–x	1rtal–y	1ltal–x	1ltal–y	1ls–x	1ls–y	1sto–x	1sto–y	1li–x	1li–y	1sm–x	1sm–y	1pg–x	1pg–y	1me–x	1me–y
1–1 & 1–	140	127	182	131	233	133	273	129	214	117	221	206	184	188	241	188	212	226	211	235	212	244	206	266	209	283	207	313
1–2 & 1–	77	146	100	144	135	141	163	137	111	136	114	190	98	181	137	174	120	203	121	210	121	218	125	231	125	244	129	261
1–3 & 1–	144	110	191	117	247	119	286	125	225	104	220	202	187	178	247	181	215	230	214	238	213	255	212	273	211	290	206	322
2–1 & 2–	302	552	403	569	541	578	673	585	456	536	414	753	357	702	516	720	420	823	416	845	408	888	409	911	391	941	399	1025
2–2 & 2–	119	92	151	94	195	95	226	97	177	80	175	144	152	132	195	134	174	164	173	171	172	182	170	193	170	203	169	233
2–3 & 2–	68	69	87	73	120	75	139	78	100	66	99	113	85	103	114	105	97	127	97	130	97	136	98	145	99	152	98	166
3–1 & 3–	128	140	169	141	221	137	257	138	195	123	196	202	170	189	221	190	197	225	198	233	199	244	200	258	198	272	202	298
3–2 & 3–	159	150	200	147	263	142	306	144	223	123	233	220	199	208	260	204	235	248	236	256	237	271	241	286	240	301	249	334

Figure 1. The structure of the FFMs dataset

For the CNN network, the data set used for training is the original picture corresponding to the sparse photo in the FFM data set. In this data set, the three pictures are a group corresponding to each row in the FFM data set.

Figure 2. The structure of the Facial Features

2.2 BDNN Theory

The realization of the bidirectional neural network is based on the ordinary one direction neural network. On the basis of the one direction neural network, a reverse neural network is added to strengthen the relationship between the input vector and the output vector. In the bidirectional neural network, the former forward neural network is like a traditional neural network to train the neural network through forward pass and error back propagation, while the backward neural network is to train the output vector in the forward pass in the backward neural network and then convert it into the input vector of forward neural network. So, The dimension of input vector of the backward neural network is the same as the dimension of output vector of the forward neural network, and the dimension of output vector in the backward neural network is the same as the dimension of input vector in the forward neural network. In this way, when the bidirectional neural network is repeatedly trained, the connection between the sample input vector and the output vector will be strengthened.[1]

2.3 BDNN Experiment

In the experiment, I set up two neural networks, namely a forward neural network and a backward neural network. After the forward network training is completed, the parameters of this training are passed into the backward training (excluding bias), then continue the training of the backward network, and after that pass the parameters of this backward training into the forward network for the next epoch training. The forward neural network is like the left half of the figure, and the backward neural network is the right half in the figure. The structures of the two neural networks are symmetrical and opposite. The weight between the input layer and the hidden layer in the forward network is the same as the weight between the hidden layer and the output layer of the backward network. The weight between the hidden layer and the output layer in the forward network is the same as that between the input layer and the hidden layer in the backward network.The two networks do not share the bias. In the final stage, I use the cross-entropy loss function for the forward network, the MSE loss function for the backward network and use the error back propagation technique to adjust the weight of the network.

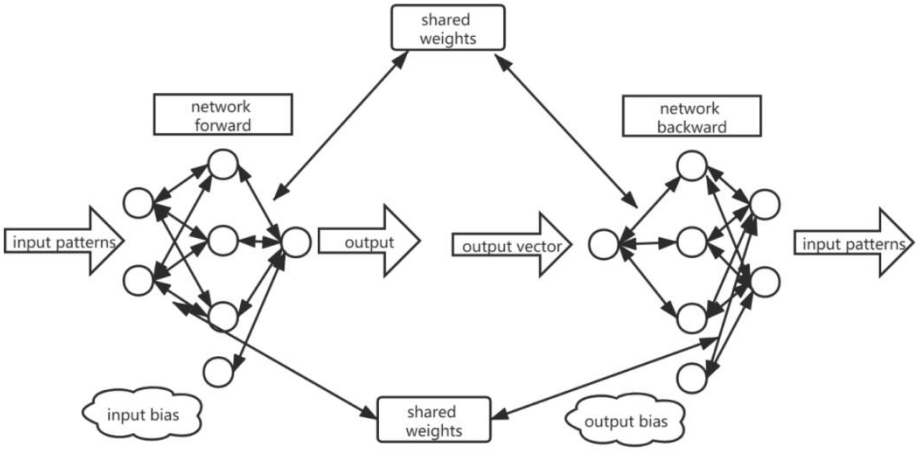


Figure 3. The structure of the bidirectional neural network

2.4 CNN Theory

CNN is mainly composed of three parts, 1. Convolutional layer 2. Pooling layer 3. Fully Connected layer. In the convolutional layer, it is mainly through multiple filters to continuously extract features, from local features to overall features, so as to carry out the face Recognition. In the pooling layer, it is to extract the main features of a certain area and reduce the number of parameters to prevent the model from overfitting. In the fully connected layer, the previous local features are reassembled into a complete graph through the weight matrix, and all local features are used.[7]

2.5 CNN Experiment

The experiment will be based on traditional convolutional neural networks to achieve face matching. CNN consists of 4 convolutional layers and 2 linear layers, which are used to output matching results. The image input uses RGB color images. Rescale the image to the size of [28, 28], and input the RGB three channels of the original image as the input image. In order to make the pixel component of the input image [0, 255], we need to normalize it to [0, 1] for easier calculation. The convolutional layer uses the weighted value of the convolution kernel of this layer to convolve the output image of the previous layer and add bias, obtain the feature image through the Relu activation function, and then normalize the feature image. In order to be fully connected with the traditional multi-layer perceptron MLP, each pixel of all the feature images of the upper layer is expanded in sequence and arranged in a row. The last layer is the classifier, which uses the Softmax function for face matching

3 Optimize

3.1 Normalize

Since the data given by the data set meets the standards for two-way neural network training, there are no other negative effects (for example, the data has missing values), and the unique id attribute is not entered into the neural network as input, so here just normalize the data set to achieve the best data set effect

Normalization the data into decimals between (0, 1)
(1) Improve convergence speed
For linear models, after the data is normalized, the optimization process of the optimal solution will obviously become smooth, and it will be easier to correctly converge to the optimal solution.
(2) Improve the accuracy of the model
Let each feature contribute the same to the result.
(3)Data normalization in deep learning can prevent model gradient explosion.

3.2 Optimizer

After trying many optimizers and comparing the results of different optimizers, it is concluded that Adam is the best performing optimizer.The biggest advantage of the Adam algorithm is that in the initial stage the step size will be limited to a certain range. The updated step size calculation can be adaptively adjusted from the gradient mean and the gradient square, rather than directly determined by the current gradient[3]. As a result, there are many places where

Adam can be applied and have better performance compared to other algorithms. For example, it is suitable for sparse gradients or the problem which has very noisy gradients.Sparse gradients is also the characteristic of our FFMs data set. So adam optimizer is very suitable for use in this project.

3.3 Adjust Hyperparameters

Adjusting the hyperparameters of the neural network will make our neural network achieve better results, so in this experiment, I adjusted the number of iterations of the neural network, the number of hidden layer nodes, and the number of hidden layers. I adjusted the parameters in the experiment based on the SPI baseline (the content described later). And through the neural network to compare the four matrices(the content described later) of the test set to compare different parameters, and select the optimal parameters.

The optimal hyperparameters in BDNN
num_epoch = 50
learning_rate = 0.1
net_forward = forward(n_feature=56,n_hidden=10,n_out=2)
net_backward = backward(n_feature=2,n_hidden=10,n_out=56)

The CNN Architecture

```
CNNnet(
  (conv1): Sequential(
    (0): Conv2d(3, 16, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (1): BatchNorm2d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
  )
  (conv2): Sequential(
    (0): Conv2d(16, 32, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
  )
  (conv3): Sequential(
    (0): Conv2d(32, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
  )
  (conv4): Sequential(
    (0): Conv2d(64, 64, kernel_size=(2, 2), stride=(2, 2))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU()
  )
  (mlp1): Linear(in_features=256, out_features=100, bias=True)
  (mlp2): Linear(in_features=100, out_features=12, bias=True)
)
```

3.4 Evaluation

In order to evaluate the BDNN and CNN models, four metrics called SPI baseline [8] were used in this paper. They are accuracy, precision, recall and specificity. Their definitions are listed below.

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

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

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

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

tp = true positive, fp = false positive, fn = false negative, and tn = true negative.

Accuracy refers to the degree to which the average value of multiple measurements under certain experimental conditions coincides with the true value. It is expressed as an error and used to indicate the size of the system error. Precision is the description of random error, which is used to measure the degree of agreement between the measured values when the same amount is repeatedly measured. recall is the fraction of the relevant documents that are successfully retrieved.[8]

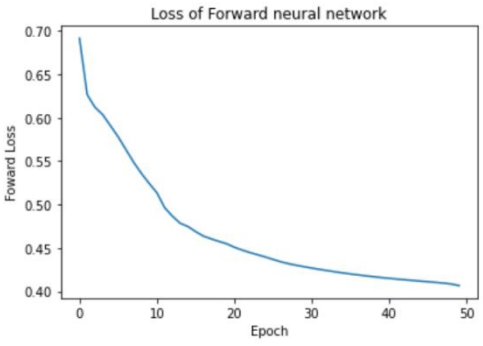
4 Results and Discussion

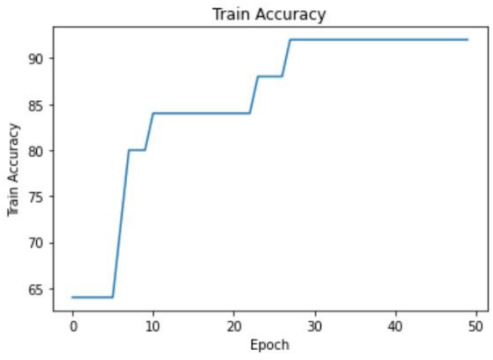
4.1 BDNN Result

After training and optimizing the data set with a bidirectional neural network, the loss of the forward and backward neural network and the change in the accuracy of the training set during the training process are shown in the following figure. These pictures can be very intuitive to see, in the process of training the neural network, the effect of the loss of the forward neural network and the accuracy during the training process gradually increase with the number of training.The average accuracy of the model on the test data is 81.82%. And the other three evaluation method also show a good effect. These data also shows that the two-way neural network has a good performance.

```
Testing precision: 0.40
Testing recall: 0.67
Testing Specificity: 0.62
```

(Since in this face matching system we mainly want to achieve whether the output face matches, instead of predicting the change of the face in the FFM of the model, we will not discuss the loss change in the backward neural network for the time being.)





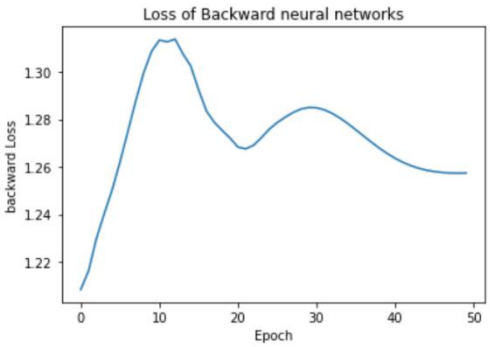


Figure 4. The Judging criteria of the bidirectional neural network

4.1 CNN Result

After pre-processing the picture, optimizing the network structure and adjusting the hyperparameters, what is shown in the picture is the loss, train accuracy, and test accuracy obtained after training through each epoch. These evaluation data can be clear It is concluded that CNN has very good adaptability to face matching. Each data will gradually increase with each epoch training, and finally reached an accuracy of 85.7%.

```
epoch 1, loss 2.5129, train acc 0.103, test acc 0.143
epoch 2, loss 2.3471, train acc 0.172, test acc 0.429
epoch 3, loss 2.1917, train acc 0.414, test acc 0.571
epoch 4, loss 2.0412, train acc 0.724, test acc 0.714
epoch 5, loss 1.8969, train acc 0.828, test acc 0.857
```

Figure 5. The Judging criteria of the Convolutional Neural Network

4.2 Discussion and Comparison of the Result

Compared with the previous paper, the accuracy in the previous paper[1] of face matching is 75% and the average accuracy of BDNN is 81.82% and the average accuracy of CNN is 85.7%. This data shows that CNN performs better in the FFM data set.The loss in the two-way neural network gradually decreases with the training of each epoch and finally flattens. The bidirectional neural network has shown better results in all kinds of data compared with the previous paper.There may be many reasons for this result. I think the main reason that makes the model perform better is that the bidirectional neural network enhances the relationship between the input vector and the output vector.Because the FFM data of a face in different photos may be different (for example, in frontal and side shots, the position of Right exocanthion in the image is different). Although the magnitude of the FFM change will not be large, it also has a certain impact on the neural network's learning of FFM and face matching. The bidirectional neural network happens to be able to adapt to this difference to a large extent. Because in the process of training the bidirectional network, the neural network can learn the changes in FFM in the output face image, and provide appropriate input changes according to the ideal output change, relying on the feature of the bidirectional network to adapt to the different FFM data of the same person on different photos. Secondly, the bidirectional neural network can not only obtain previous information, but also obtain future information, which provides good adaptability for model prediction[6]. At a

deeper level, this feature increases the robustness and generalization ability of the model. As for the CNN,This kind of convolutional neural network, sharing the convolution kernel, can better process high-dimensional data and can automatically perform feature extraction. This will greatly reduce the impact of different angles and expressions on face matching and recognition. Here On the basis, each area has its own exclusive characteristics during training and will not be affected by other areas.

5 Conclusion and Future Work

5.1 Conclusion

In this paper, we proposed to use a two-way neural network and CNN instead of a traditional one-way neural network for facial recognition and matching. Use the FFMs and sparse photo data set to train it, and use the results to verify the performance. According to the results of our experiment, the CNN is the best neural network for judging whether the people in two historical sparse photos are the same person. The reason is that the bidirectional neural network can adapt well to the impact due to the difference in front and side photos of the same person.

5.2 Future Work in BDNN area

Although the current effect of the bidirectional neural network is good, it can be applied to the currently used data set to a large extent. However, we should develop a deeper and more extensive application of the bidirectional neural network, for example using sensitivity analysis with a BDNN, perform acoustic analysis, speech recognition and natural language understanding to jointly form human-machine voice interaction[5]. Our method of training bidirectional neural networks to learn both the forward and reverse tasks at the same time will yield more powerful neural networks. The bidirectional learning aggregate gradient tends to be flatter. We believe this may result in networks less susceptible to noise and providing better generalisation.in future work,We believe that a well-trained Bidirectional neural network will help us solve the bottleneck of neural network acceptance and use.

5.2 Future Work in CNN area

At this stage, CNN already has good training capabilities, but in future work, we still have a lot of optimization to deal with. For example, when building the base layer of the volume for feature extraction, we can use hole convolution and deformable convolution Optimize the existing convolutional layer. In the training time, CNN has high memory requirements and takes a long time, which makes them unable to be deployed on mobile platforms. How to reduce complexity and obtain fast-executing models without reducing accuracy is an important research direction. Secondly, we found that one of the main obstacles to the application of CNN to new tasks is: how to choose appropriate hyperparameters? Such as learning rate, kernel size of convolution filtering, number of layers, etc., which require a lot of technology and experience. These hyper-parameters have internal dependencies, which can make adjustments very expensive. Recent studies have shown that there is huge room for improvement in the selection of learning deep CNN architecture.

Finally, regarding CNN, there is still a lack of a unified theory. The current mode of operation of the CNN model is still a black box. We don’t even know how it works or how it works. At the moment, it is worth putting more energy into studying the basic rules of CNN. At the same time, just as the early development of CNN was inspired by the biological visual perception mechanism, both deep CNN and computer neuroscience require further in-depth

research[7].

References

1. Caldwell, S., n.d. *Human interpretability of AI-mediated comparisons of sparse natural person photos*. Research School of Computer Science The Australian National University, pp.1-35.

2. Nejad, A. and Gedeon, T., n.d. *BiDirectional Neural Networks and Class Prototypes*. Sydney 2052 AUSTRALIA: Department of Artificial Intelligence School of Computer Science and Engineering The University of New South Wales, pp.1-6.

3. Jianshu. 2021. *A simple understanding of Adam optimizer*. [online] Available at: <https://www.jianshu.com/p/aebcaf8af76e> [Accessed 21 April 2021].

4. Mallick, S., 2015. Facial Landmark Detection. [ebook] Learn OpenCV.

5. Schuster M, Paliwal K K. Bidirectional recurrent neural networks[M]. IEEE Press, 1997.

6. Wu, E., 2019. *Bidirectional NN*. [ebook] DeepLearning.ai. Available at: <https://baozoulin.gitbook.io/neural-networks-and-deep-learning/di-wu-men-ke-xu-lie-mo-xing-sequence-models/di-wu-men-kexulie-mo-578b28-sequence-models/recurrent-neural-networks/111-shuang-xiang-xun-huan-shenjing-wang-luo-ff08-bidirectional-rnn> [Accessed 23 April 2021].

7. MA, W., 2016. Recent Advances in Convolutional Neural Networks. [ebook] Available at: <https://blog.csdn.net/maweifei/article/details/52984920> [Accessed 19 May 2021].

8. LI, X., 2021. Clearly understand the relationship between accurac, recall, and precision. [ebook] CSDN. Available at: <https://blog.csdn.net/qq_30022055/article/details/106623871> [Accessed 28 May 2021].