Predicted whether two photos are matched through CNN and three -layer neural network and explain result by causal index and characteristic input pattern

Di Li

Research School of Computer Science
The Australian National University
Canberra ACT 2601
u7039884@anu.edu.au

Abstract. In this paper, by training a three-layer neural network and a CNN to predict whether two photos match. The provided data set includes 36 images, Facial Feature Markers (FFMs) provided as x, y coordinates in pairs of FFMs and FFM distances provided as lengths (in pixels). The final results show that the accuracy of CNN can reach 86% to 90%, and sometimes even 100%, while the traditional three-layer neural network can only reach 70% to 80%. In addition, I calculated the gradient of the output with respect to the input as a causal index to explain the output of the neural network, and extracted the rule through the causal index, and used these rules to predict results. However, the accuracy of using the rule to predict the result is very low, only 45%. Finally, the paper discussed the results of the three methods and further work

Keywords: Match of two pictures · CNN · Three-layer-neural network, causal index · Extract rules from characteristic pattern.

1 Introduction

1.1 Dataset introduction and choice

This paper provides 36 images and two sets of data related to the images. These images are sparse historical photos. There are a total of 12 groups of photos, each group has three photos in which the first two photos are match, and the last photo does not match. Each comparison is only for three photos within the group, therefore, for each group of three photos, the first photo matches the second photo, the second photo does not match the third photo, and the third photo does not match the first photo. This project provides two kinds of dataset with repect to the photos, one is using the coordinates of the facial feature markers of the photos, called **FFMs**, another is **FFM** distance as data, called **distances**. Each photo has a total of 28 facial feature coordinates, including the right exocanthion, right endocanthion, left endocanthion, left exocanthion, nasion, subnasale, right alare, left alare, labiale superius, stomion, labiale inferius, supramentale, pogonion, menton . Therefore, FFMs is a dataset with 56 features and the coordinate of these facial features as dataset. **distances** is a dataset with 182 features and each feature represents the Euclidean distance between any two facial features. I finally decided to use distances as the data of my three-layer neural network and the pixel values of 36 photos are taken as data of my CNN. This is because the data expresses the distance between facial features. Compared with the coordinates of facial features, the distance of facial features is more rigorous, because coordinates of the facial feature may change due to the different shooting angles, but the distance of the facial feature will not change. Secondly, the distance feature has more features than the coordinate feature, the model trained with more features is general more accurate than fewer features, so I choose distances as my data set. The pixel value of the photo is selected as the input of the CNN model because the advantage of the CNN model is that the model can directly take the pixel value of the image as input and extract the feature value from the picture.

1.2 Model goal and investigations

The goal of this paper is to predict whether the two photos are the same person by comparing the distance between the facial features of the two photos. The output result is a match or a mismatch. It is very meaningful to complete this goal. Although AI has successfully solved the problem of facial recognition in the modern sense, that is, hundreds or even thousands of personal photos can exist through digital photography and video capture, but the help in determining personal identity is limited, especially some old photos, because of number of old photos is very small, and it is difficult to judge its personal identity. In addition, people also want to understand how neural networks determine the personal identity of photos [9]. Through analysis, we can see that the problem is a binary classification problem, and there is no correlation between each feature. Therefore, the traditional multilayer neural network is a good choice to solve this problem. The main problem is to adjust the hyperparameters of the neural network to achieve the best training results. Although traditional neural networks can be used to learn features and classify data, this architecture is generally impractical for larger inputs such as high resolution images. It would require a very high number of neurons, even in a shallow architecture, due to the large input size of images, where each pixel is a relevant input feature. Therefore, this paper further introduces a convolutional neural network that extracts features directly from pictures. CNN is mainly composed of convolution and pooling. The objective of the convolution operation is to extract the high-level features such as edges, from the input image.Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational Power required to process the data through dimensionality reduction. Finally, the feature space output by CNN is used as the input of the fully connected neural network, and the fully connected layer is used to complete the mapping from the input image to the label set.

1.3 Characteristic pattern and Causal index

Although neural networks have powerful learning capabilities, generally, as long as there is enough data, the results of neural networks are more accurate, which greatly reduces a lot of expensive expert time. However, unlike conventional expert systems, the neural network cannot provide any explanation about its output. Therefore, we need to find a way to intuitively understand the reason why the neural network gets the output. Through the provided paper [1], I learned that one of the methods is to prune the neural network to the smallest size, but this extreme pruning does reduce the robustness of the neural network solution. Another method is to encode activation values to generate rules, but the number of rules will also increase exponentially with the number of inputs. Therefore, I used the another method mentioned in the paper [1] to explain the output of the neural network, that is, using the **causal index**. The causal index refers to the mathematical method to find the rate of change of output with respect to input, which means to find the derivative of the output with respect to the input, and use gradient changes to extract the rules. In addition, in order to calculate the causal index, we also need to find the **characteristic pattern** by calculating the average value of the input, which means that a pattern representing each input class is created as a characteristic pattern for that class. For my data, because my output results only match or not match, so there are only two characteristic patterns, that is, **characteristic ON pattern** (match) and **characteristic OFF pattern** (mismatch).

2 Method

2.1 Data preprocess

In order for my 3-layer neural network to be able to train the provided data, I preprocessed the data. First, I used **pandas.read_excel** to read the provided data. According to my explanation of data selection above, I only read data about the distance between facial features. According to the observation of the data, the first column of the data is the index number of the two pictures in the current row. This index number has no value to the neural network and will reduce the accuracy of the neural network. Therefore, I deleted this column. In addition, in order to train the neural network with different data each time, I randomly sorted the data so that the training set and the test set used by the neural network are different each time. In order to know the quality of my neural network, I divided the data into two parts, the training set and the test set. The training set accounts for 80% of the total data, and the test set accounts for 20% of the total data. In order to calculate the accuracy of the training set and the test set, the training set and the test set are divided into two parts, the last column of the data is used as the true value of the data, and the rest is used as the input of the neural network. Finally, in order to improve the accuracy of the model, the test set and training set were normalized.

For CNN, since my original data is 12 groups of color photos, each group contains 3 photos. First of all, in order to speed up the convergence and reduce the calculation, I converted all 36 color photos to gray photos. Secondly, in order to input the photos into the model, I changed all the photos to the same size, 64*64. Since the purpose of this paper is to predict whether two photos match, I will compare the three photos in each group, merge the first photo with the second photo, the second with the third, and the first with the third one, and then I got a 3*2*64*64 array. After I merged all the 12 groups of photos, I finally got a 36*2*64*64 data set. After that, I divided the data set into a training set (80%) and a test set (20%). Finally, in order to improve the accuracy of the model, the test set and training set were normalized.

2.2 Neural Network

This paper uses a 3-layer neural network, including an input layer, a hidden layer, and an output layer. The number of neurons in the input layer is the same as the number of features in the data. Since my task is to identify whether two photos match, there is only a match or a mismatch, so there are 2 neurons in the output layer. The number of neurons in the hidden layer requires a lot of experiments to adjust. In addition, in order to adapt to the nonlinear classification problem, I used the **Sigmoid** function as the activation function of the hidden layer and the output layer. This is because the sigmoid function can smoothly map the real number domain to the $[0, 1]$ space. The function value can be interpreted as the probability of belonging to the class. In addition, the sigmoid function is monotonically increasing, continuous and derivable, and the derivative form is very simple, which is a more suitable function for binary classification problems. For the choice of optimizer, I chose to use **Adam** as the optimizer of the neural network. This is because it combines the advantages of the two optimization algorithms **AdaGrad** and **RMSProp**. Comprehensively consider the first moment estimation and second moment estimation, and calculate the update step length. The most important thing is that the optimizer has a good interpretability for the hyperparameters, and only needs to be fine-tuned or even no adjusted. Simple implementation, efficient calculation, and low memory requirements are also the reasons why I choose it as the optimizer. The loss function I chose to use cross-entropy loss, this is because compared with other loss functions, the speed of convergence of the cross-entropy loss function is stable, and the speed of convergence will not change with the change of the difference. In addition, the cross entropy loss is often used in conjunction with the sigmoid function. The choice of learning rate is the most important hyperparameter of the neural network, which greatly affects the quality of the model, and it is difficult to choose theoretically, but requires a lot of experiments to adjust. In order to make the accuracy of my model convincing, I call the model 10 times and calculate the average of these 10 accuracy as the final accuracy of my model. In the end, after a lot of experiments, I chose the following values as the hyperparameters of my model

Table 1. Hyperparameters of my 3-layer neural network

Name	Value
Number of input neural	182
Number of hidden neural	120
Number of output neural	2
Learning rate	0.001
Number of epochs	500

2.3 CNN

For the CNN model design, in order to better extract the features of the picture, I added 2 convolutional layers and set the kernel size to 5. Since my input is a picture with 2 channels, the number of channel of the convolutional layer is set to 2. In addition, since most of the information contained in the output of the convolutional layer is redundant after each convolution calculation, the output value is pooled in order to make the feature map smaller, simplify network calculation complexity, perform feature compression, and extract main features. I chose max pooling as my pooling layer function. This is because max pooling selects brighter pixels from the image relative to average pooling. It is useful when the background of the image is very dark and we are only interested in the brighter pixels of the image [13]. Because most of my photos are white people, and the background of the photo is black, so I choose max pooling as the pooling function. Secondly, unlike traditional neural networks, I choose ReLU as the activation function of the CNN model. This is because the use of ReLU helps prevent the amount of computation required to operate neural networks from increasing exponentially. ReLUs also prevent the emergence of the so-called "vanishing gradient" problem, which is common when using sigmoidal functions [12]. While sigmoidal functions have derivatives that tend to 0 as they approach positive infinity, ReLU always remains at a constant 1. This allows backpropagation of the error and learning to continue, even for high values of the input to the activation function. After extracting the features of the picture, I used 2 fully connected layers to map the features to the classification, and added a dropout layer after the first fully connected layer. The effect of using this dropout layer is that Dropout will randomly disable certain neurons, thereby invalidating their contribution to the output, thereby avoiding overfitting. For the choice of optimizer, I chose the same optimizer Adam as the traditional neural network. For CNN, the learning rate is also the most important hyperparameter of the model. I used the same method as the traditional neural network, through many experiments, continuously optimized the hyperparameters, and saved the best model according to the accuracy of the output. In the end, I got the following hyperparameters of the CNN model

Table 2. Hyperparameters of my 3-layer neural network

Name	Value
kernal size	5
Number of channel	2
Learning rate	0.01
Number of epochs	20

2.4 Technique

The provided paper [1] describes the use of characteristic patterns and causal indexes to explain neural networks. The specific calculation method is to group the data according to the output of the neural network, that is, use the trained neural network to predict the output of the test set. According to the output, the test set is divided into ON input (output equal to 1) and OFF input (output equal to 0), and then calculate the average values of ON input and OFF input as characteristic inputs, called characteristic ON input and characteristic OFF input. Then, for a new test data point, calculate the distance between the point and the characteristic ON input and characteristic OFF input, and take the closest one as the predicted value of the point. Finally, the predicted result is compared with the actual result of the neural network and calculate accuracy.

According to this paper [2], we learned that the causal index is the derivative of the output with respect to the input. The specific method of calculating the causal index is: first, create a new data set for each feature, that is, for each data set, except for the feature that needs to be calculated causal index to be changed to a random value from 0 to 400, other all features are replaced by the value of the characteristic patterns, so that for each feature, we have 1000 new data. Then, put these 1000 data into the trained neural network to get the predicted value. Finally, the causal index of each feature is calculated separately in the on and off states. I draw the result into a picture to facilitate the extraction of rules later. Since my data has 182 features, it is difficult to put the causal index of the entire feature in one chart. Therefore, I drew the causal index of the first 10 features.

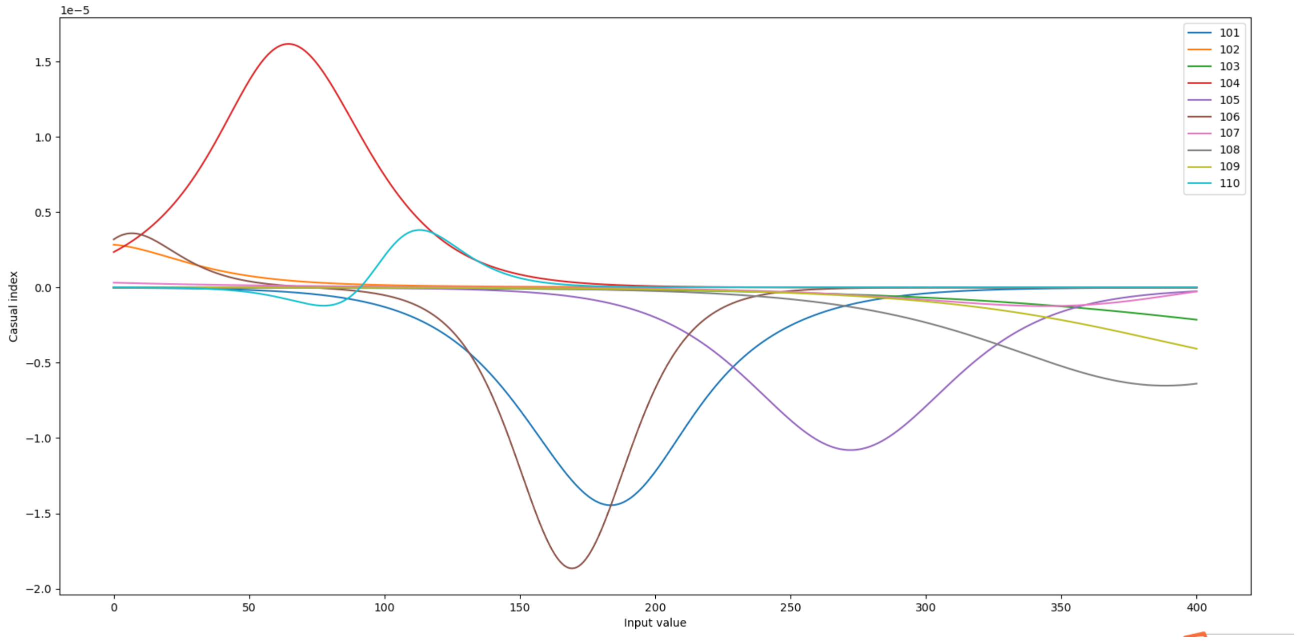

Fig. 1. characteristic ON pattern

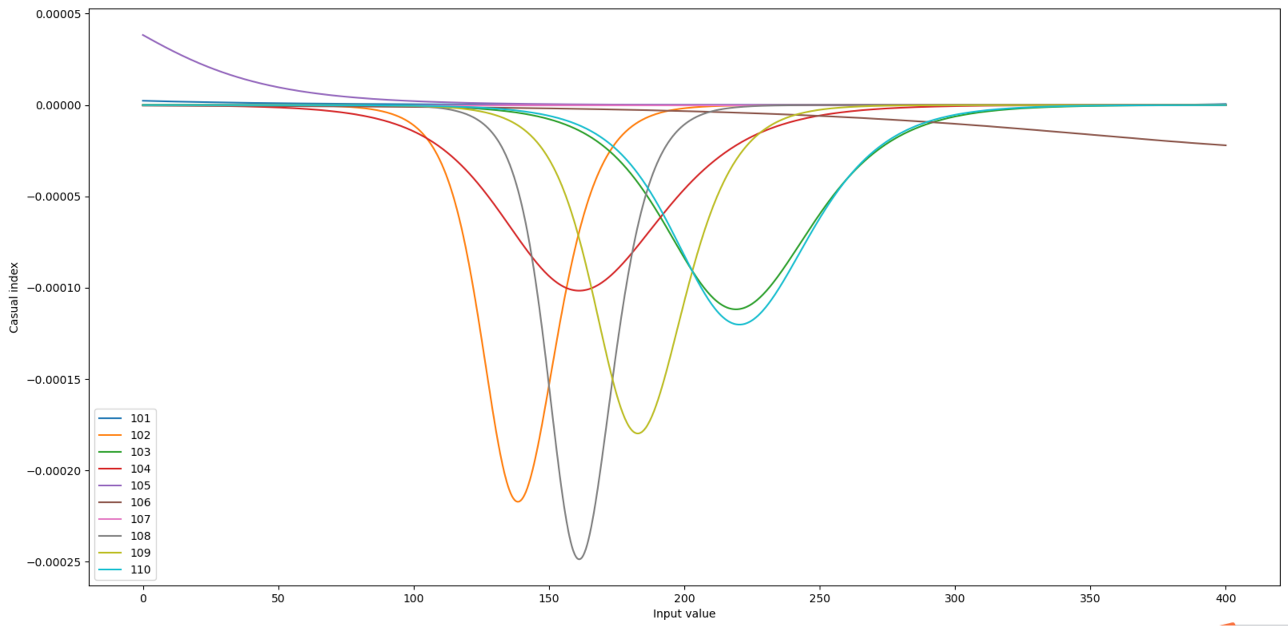

Fig. 2. characteristic OFF pattern

For a three-layer neural network, I can extract the rules from the two images above. By observing the above two pictures and the relationship between the causal index and rules described in [2], I found the characteristic value of the 10 features, and found the point where the characteristic value tend to be flat. The characteristic value is used as the feature's rule. For example, by observing the feature ”104” in the above two images, we can find that for the characteristic OFF pattern, a large negative peak in the rate of change for feature 104 which then returns to a constant level (zero) at 300, therefore, ”104”$<$ 300 is a sub-rule of the characteristic OFF pattern. For the characteristic ON pattern, a large positive peak in the rate of change for feature 104 which returns to a constant level (zero) at 180, so ”104”$>$ 180 is a sub-rule of the characteristic ON pattern. Through the above method, I can find all the sub-rules of the characteristic pattern for the 10 features. Finally, all the sub-rules of the characteristic ON pattern are combined using the conjunction operator, and all the sub-rules of the characteristic OFF pattern are combined using the disjunction operator.

Since the input of CNN is a pair of picture, rather than several features of traditional neural networks, the method of calculating the causal index of traditional neural networks cannot be used. So this article will compare the three accuracy of traditional neural network, CNN and the extracted rules.

After all the sub-rules are extracted, we can bring the test set into the rule to calculate the accuracy of the rule.
The explanation of the neural network can be divided into the following steps:
1. The output of the 3-layer neural network
2. Calculate feature patterns and present the most similar output.
3. extract rule by calculating the causal index.
4. Output according to the rules

3 Results and Discussion

Table 3. Predicted result by 3-layer neural network, characteristic pattern and rules

Name	Accuracy
3-layer neural network	75%
CNN	86%
Rules	45%

According to **Table3**, we can see that the CNN has an accuracy of 86%, while the accuracy of the result generated by the 3-layer neural network is only 75%, which is a decrease of 11%. This shows that the CNN is more suitable for image recognition than using the traditional neural network to predict. This is because each neuron in each layer of CNN takes a group of small local neighbor units of the previous layer as input, that is, weight sharing and local connection. The neuron extracts some basic visual features, such as edges, corners, etc. , These features will later be used by higher-level neurons. CNN obtain feature maps through convolution operations. A convolutional layer usually contains multiple feature maps with different weight vectors, so that the image can retain richer features. The convolutional layer will be connected to the pooling layer for downsampling. On the one hand, it can reduce the resolution of the image and reduce the amount of parameters. On the other hand, it can obtain the robustness of translation and deformation. The alternating distribution of convolutional layer and pooling layer makes the number of feature maps gradually increase, and the resolution gradually decreases[6]. Although the accuracy of CNN has reached 86%, in the real world, the accuracy of face recognition should reach more than 90%. Therefore, the CNN model is not good enough. One of the very important reasons is that the provided data is too small, and only 36 photos have been trained. Therefore, the trained model is not generic enough to perform well in the test set.

However, for the traditional three-layer neural network, since each input is a pair of photos, the size of the photo is 2*64*64, then the feature dimension reaches $64 \times 64 \times 2$, then our neural network needs $64 \times 64 \times 2$ inputs. If the hidden layer of the neural network has 1,000 hidden units, then the parameter matrix between the input layer and the first hidden layer has millions parameters. With such a large number of parameters, it is difficult to obtain enough data to prevent the neural network from overfitting, and there is still huge pressure.

From the third row of **Table3**. We can see that the accuracy of using the extracted rules to predict is only 45%, which is far less than the CNN and the 3-layer neural network, and also far less than the provided paper [1]. According to the two causal index graphs in Fig1, we can find that although most of the curves have a process from 0 to apex and then to 0, there are some curves that do not drop or rise significantly. These curves are considered unimportant. characteristics. Although the rule extraction can explain the neural network well, so that the user can intuitively observe which features are important to the neural network, it is not accurate enough to use the rules to predict results, which indicates the limitation of this method to a certain extent

4 Conclusion and Future Work

Although the accuracy of CNN has reached 86%, there are still some areas that need to be improved. For example, when I tuning hyperparameters, I often encounter overfitting and underfitting. One of the ways to avoid this situation is cross validation. The specific method is to randomly divide the training set into k equal parts, take one part as the validation set to evaluate the model, and the remaining k-1 parts as the training set, repeat this step k times, and take a different subset each time as a validation set, finally, k different models and k accuracy are obtained, and the performance (average score or other) of these k models is integrated to evaluate the pros and cons of the model in the current problem [10].

Another improvement method is early stopping. Divide the data into training set and validation set. After each epoch (or after every N epoch): Obtain the test results on the validation set. As the epoch increases, if the test error is found to rise on the validation set, stop training and use the weight after the stop as the final parameter of the network [11].

References

1. Harry,S. T., Tamás,D.G.: Extracting Meaning from Neural Networks
2. Gedeon T. D. and Turner H. S.:Explaining student grades predicted by a neural network. In Proceedings of 1993 International Joint Conference on Neural Networks(1993)
3. Quora, https://www.quora.com/Why-do-neural-networks-need-more-than-one-hidden-layer
4. ResearchGate,https://www.researchgate.net/post/How-to-decide-the-number-of-hidden-layers-and-nodes-in-a-hidden-laye
5. Machine Learning Mastery, https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/
6. Samer, H. Rishi K. and Chris,R.: Using Convolutional Neural Networks for Image Recognition. https://ip.cadence.com/uploads/901/cnn_wp-pdf
7. Investopedia, https://www.investopedia.com/terms/n/neuralnetwork.asp#:~:text=Neural%20networks%20are%20a%20series,fraud%20detection%20and%20risk%20assessment.
8. Quning Z.: Predicted the authenticity of anger through LSTMs and three -layer neural network and explain result by causal index and characteristic input pattern
9. Sabrina, C.: Human interpretability of AI-mediated comparisons of sparse natural person photos
10. Machinelearningmastery, https://machinelearningmastery.com/k-fold-cross-validation/
11. Towardsdatascience, https://towardsdatascience.com/early-stopping-a-cool-strategy-to-regularize-neural-networks-bfde
12. Baeldung, https://www.baeldung.com/cs/ml-relu-dropout-layers
13. Towardsdatascience, https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks--eli5-way-