Improvement in Face-emotion Analysis replacing CasPer with CNN

Ran Ji[1]
u6447167@anu.edu.au

Australian National University, Acton ACT 2601, AU

Abstract. The CasPer neural network and convolutional neural network were proposed to the Static Facial Expressions in the Wild (SFEW) dataset to analyse the connection between human facial features and the emontion expressed from it. The CasPer networks were examined that it had the tendency of making more accurate predictions and it generalises better than CasCor networks. The CNN was compared to have better performances than CasPer technique and it was believed that classification with high accuracy could be made through CNN with the same level of generality CasPer has.

Keywords: artificial neural network · face-emotion · deep learning .

1 Introduction

1.1 CasPer

The Cascade correlation algorithm [1] was proven to be useful in the construction of artificial neural network. It contains a 2-phase training process which allows random starting weights. The cascade architecture adds hidden units once at a time and the learning algorithm install the weight of new hidden units with maximized correlation between its output and the residual error of the whole network while all weights of previous hidden neurons are frozen. Then, weights connected with the output neurons are trained to minimize the loss after the insertion of a new neuron. Both phases of training are ceased when the loss stops decreasing. Recalculation of network values when a new neuron is inserted is not neccesary for CasCor networks because of the freezing weights. Errors are not back propogated since just one layer is involved in a step of training.

Dropbacks of CasCor networks are critical. The freezing weights would lead to extremely large networks since early stages frozen neurons are much less detective which might introduce additive errors. These extra tasks could be solved by later layers added to teh network, but the training cost spent on them are enormous. Also, it has appeared that the generality of CasCor, especially in regression and classification problems.

The CasPer contains a modified RPROP algorithm [2] which uses individualistic learning rate for every weight in gradient descent. The CasPer networks are contructed similarly to the CasCor networks and RPROP takes places when

a hidden neuron is inserted. 3 subdivisions inside the network are assigned with different initial learning rates. Layer 1, 2 and 3 cover the weights which are connecting the new neuron and previous hidden/input neurons, the output of new neuron and the output neuron, the old hidden neurons and the input neurons, respectively.

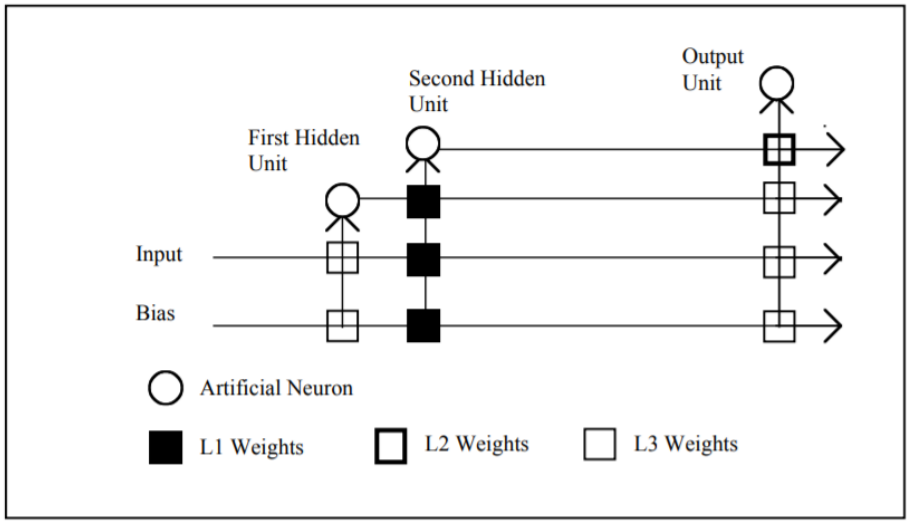

Fig. 1. The architecture of CasPer network - vertical lines sum up the inputs

The values inside each of the layers are set such that layer 1 has much larger values than layer 2 does, and layer 2 has slightly larger values than layer 3 does. New hidden neurons are considered more efficient to learn the network error with higher layer 1 values and less likely to produce network error with slightly higher layer 2 values.

To differ from the CasCor networks, CasPer does not introduce the freezing steps. Changes will be applied on old weights if the algorithm regards them benifical, but in a slower rate than the changes in weights connected to the new neuron. Advantages of freezing are hence partially preserved while major problems, namely saturation and poorly performing early stage neurons, are removed. Weight decay is introduced in CasPer for better generalisation. After the insertion of a new neuron, the weight decays proportionally to its magnitude squared. Additionaly, the installation of new neurons occur when the rate of decreasing in RMS error fall below a certain amount within a time period, which is given by 15 + P * N [4]. N represents the nunmber of installed neurons and P is a prior parameter. Casper network increases this period automatically as the size of the network grows with more hidden layers.

1.2 CNN

A Convolutionary Neural Network (CNN) is a type of space invariant artificial neural networks based on the convolution kernels, which is proven as an useful tool in many applicational fields of deep learning including image/video recognition and classification, recommandation system, natural language processing and more. The spatial independence feature ultimately ensures that the analysis around facial emotion is completely irrelated to the position where the face locates in an image. The most beneficial aspect of CNNs is reducing the number of parameters in artificial neural networks [5], which has prompted developers to approach larger models for solving sophisticated tasks that were once regarded impossible with classic neural networks. A CNN usually contains an input layer, and output layer and a bunch of hidden layers just as other neural networks do. The hidden layers in CNN would at least involve one layer where a dot product of the convolution kernel with its input matrix is performed, mostly the Frobenius inner product, with an activation function, mosly the ReLU. The convolution operation would generate a feature map contributing to the input of the next layer. Following the convolutional layers are the other functional layers, namely the pooling layers, fully connected layers or normalization layers.

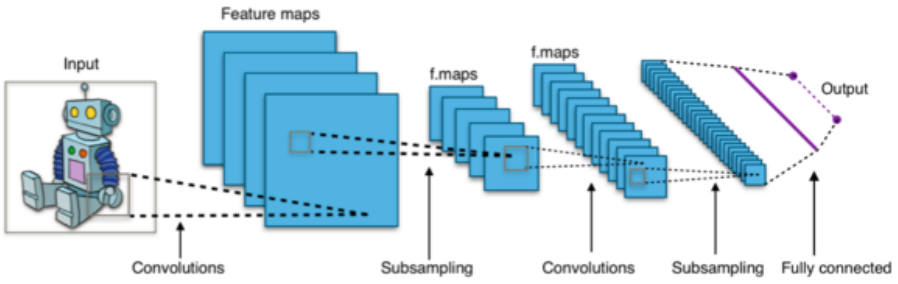

Fig. 2. Typical CNN architecture

In convolutional layers, a feature map, also called the activation map, is generated throught abstraction of the image. Hyperparameters that related to this layer are: width and height of the convolutional kernels, the number of channels and convolution operation parameters (padding, stride, dilation). In pooling layers, either local or global, the outputs of neuron clusters are combined into a single neuron at the next layer for the purpose of reducing dimensions of data. Max or average pooling takes the maximum or mean value of each local neuron cluster in the feature map. Fully connected layers work just as a typical multi-layer perceptron neural network that each neuron in a layer is connected to each neuron in another layer. It classifies the images from flattened matrices. Receptive field ensures that each neuron only receives input from a limited region in the previous layer, commonly a square region. A pyramid shaped structure is built since each neuron in the previous layer has its own receptive field. This

architecture is extended in the neural abstraction pyramid [6] by lateral and feedback connections.

1.3 Dataset and Tools

The recognition of human facial expressions became possible in recent years due to the developments in advanced computer vision. Internal affective state, intentions and communications could be reflected from changes in facial features. It is highly valuable in various fields of industry, namely lie detection, human behavior analysis, pain/stress/anxiety relief therapy, commercial promoting et al. Datasets of facial expressions were built using camera captured human faces and various deep leanring algorithms were proposed when it comes to analyze them.

The dataset on where the CasPer algorithm was performed in this project is the Static Facial Expressions in the Wild (SFEW) [3]. It contains unconstrained human facial expressions extracted from movies which includes people of large age range, various head poses, different resolutons and realistic illuminations. SFEW has been developed by selecting frames from Acted Facial Expressions in the Wild (AFEW) and accede its seven categories of expressions: angry, disgust, fear, happy, sad, surprise and neural. The dataset on where the CNN was performed is the source images, screenshots with human face, of SFEW.

The SFEW dataset was stored in .xlsx file using Microsoft Office. The SFEW source dataset was stored in .png files. The project was encoded in Python. Pandas was used for loading data, OpenCv and TorchVision were used for image reading and PyTorch was used for the implementation of CasPer and convolutional neural network.

2 Method

2.1 CasPer

According to the structure of dataset, the input size was set to 10 and the number of classes was set to 7. The batch size was set to 128 considering a total size of 675 points. Values inside the dataset were loaded and stored in a padas dataframe. It was later normalized by applying the z-score method. This method is calculated by subtracting the population mean from an individual raw score and then dividing the difference by the population standard deviation. It was further divided to a train subset and a test subset, which takes 80% and 20% of the whole dataset respectively.

Layers of neurons were built and stored in seperate torch.nn.ModuleDict(). 3 layers of CasPer network, as mentioned previously, were assigned to different parameters as shown in table 1. Connections related to each of the layers were stored in a customized CasperNet class which also introduced leaky ReLU technique.

	Layer 1	Layer 2	Layer 3
learning rate	0.2	0.005	0.001

A customized torch dataset class was constructed with adding randomized noises implemented. Additionaly, the weights were balanced with the utility of WeightedRandomSampler from torch.utils.data.sampler due to the order differences among several dimensions. Cross entropy loss was introduced when measuring the network error as it provides the distance from an estimated probability distribution to the true distribution, which more precisely indicates the performance of the network. Number of hidden layers was limited to prevent overfitting and huge network depth. Time period limits were also accumulated to the epoch limit in every loop.

K-fold method was later introduced for better stability of the CasPer neural network. As k set to 10, the dataset was divided into 10 folds and one of the folds would be selected as the validation set while others became the train set. The learning process repeated 10 times such that each fold was validated by each other 9.

2.2 CNN

The image files were loaded with OpenCv and TorchVision and a composation of transformations were applied: resizing, grayscaling and normalizing with 0.5 mean and 0.5 STD. After grayscaling, single channel images were sent to a PyTorch Tensor object and were histogram equalized then stored under a new directory. A customized CNN class with leaky ReLU and kernel size of 5 was constructed to store the neural network. The computing force of an individual Graphic Processing Unit (GPU) would be detected and made to effort with torch.device(). Otherwise, CPU would be activated as a backup plan. K-fold technique was also introduced to examine the stablity with K equals 10. Cross entropy loss method was introduced as it was in CasPer networks while Adadelta replacing RMSProp was used in the optimizer with a learning rate of 0.1.

3 Result

Results for the CasPer neural network would be shown below. The epoch limit was set as 1000 with a batch size of 128. When using default parameters of RMSprop functions in the optimizer, the results were not satisfying as the test accuracy reached 17.04% which was close to the accuracy of completely random guesses. After monitering around the momentum and weight decay, the combination that would give a better result was found. As the momentum set to 0.9 and the weight decay set to 0.00001, due to the micro-scale order of original data, the network was trained more properly. During training, we could see that the accuracy rate grows as the number of hidden layer increases as shown in fig 2. It shall be noticed that the rate started decreasing at the tail which indicated an overfitting occurs when the number of hidden layers were too high.

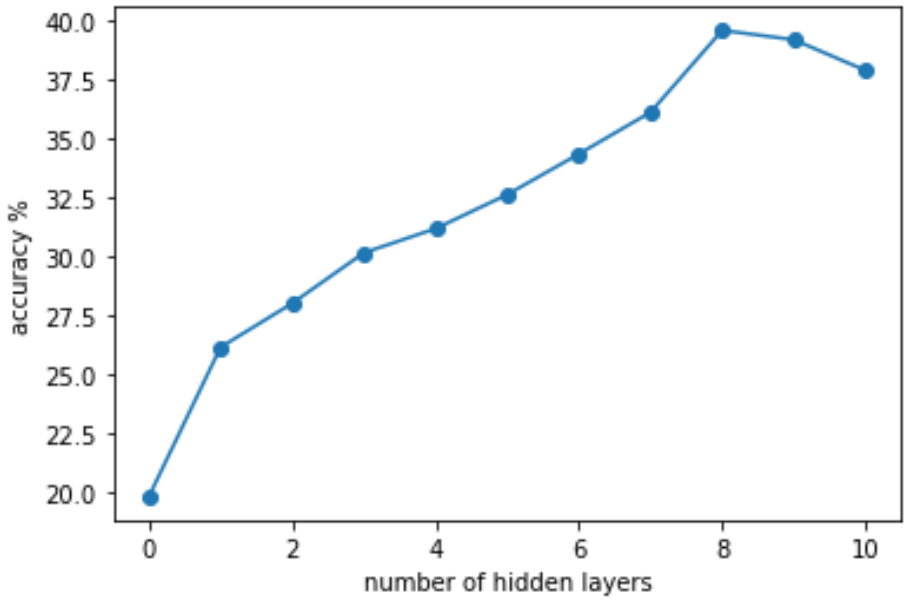

Fig. 3. accuracy against the number of hidden layers

The test accuracy was 28.35% which could be still considered as incomparable to the results given by [3]. After applying the k-fold method, the peak test accuracy reached 33.34% with the mean accuracy 25.06%, the median accuracy 24.62%. And the standard deviation of 10 accuracies was 5.67% which indicates there was only around 1.5 standard deviation from the peak to the mean. This showed that the CasPer network produced a relatively good generalization.

For the convolutional networks, the epoch limit was set as 30 with a batch size of 64. It was discovered that the training accuracy reached over 99% after mostly 22 epoches. The following table exhibits the test accuracy of the CNN with various weight decaydences:

weight decay	0.0001	0.00001	0.00002	0
test acc (mean)	42.39	42.54	41.19	41.04
test acc (median)	42.54	41.79	42.54	40.30
test acc (STD)	5.43	5.09	5.09	4.02

It was clearly shown that the weight decay only has insignificant correlation with the test accuracy which stabled around 42% after 10-fold cross validation. It also indicated that CNN has a crucial advantage in classification of facial emotions from images than the CasPer networks. The generality of CNN maintained on the same level with CasPer neural networks.

4 Conclusion

In the classification of facial expressions, the CasPer network successed to grant more generality over the CasCor network and it shows the tendency of making predictions based on the SFEW dataset. Experimentally, we found that the number of hidden layers is positively related with the accuracy before it surpass a certain limit. However, the test accuracy was not as satisfying as expected. A better solution to the tasks would be convolutional neural network whose performances were critically more satisfying with higher test accuracy and similar generality. For further improvements, the inner structure of CNN could be designed more carefully and other types of optimizer could be plugged in.

References

1. Fahlman, S.E., and Lebiere, C. (1990): The cascade-correlation learning architecture. In Advances in Neural Information Processing II, Touretzky, Ed. San Marco, CA: Morgan Kauffman, 1990, pp. 524-532.
2. Riedmfller, M. and Braun, H. (1993) A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: Ruspini, H., (Ed.) Proc. of the ICNN 93, San Francisco, pp. 586-591.
3. Dhall, A., Goecke, R., Lucey, S., Gedeon, T. (2011, November). Static facial expressions in tough conditions: Data, evaluation protocol and benchmark. In 1st IEEE International Workshop on Benchmarking Facial Image Analysis Technologies BeFIT, ICCV2011.
4. N. K. Treadgold and T. D. Gedeon, "A cascade network algorithm em-ploying progressive rprop," in Int. Work Conf. Artif. Natural NeuralNetw., 1997, pp. 733–742.'
5. Albawi, S., Mohammed, T. A., Al-Zawi, S. (2017): Understanding of a Convolutional Neural Network. Antalya, Turkey, ICET2017.
6. Behnke, Sven (2003). Hierarchical Neural Networks for Image Interpretation (PDF). Lecture Notes in Computer Science. 2766. Springer.