

A research report on the classification of FER2013 dataset

by Youqian Chen

Submission date: 04-May-2023 06:25PM (UTC+0100)

Submission ID: 204762822

File name:

40627_Youqian_Chen_A_research_report_on_the_classification_of_FER2013_dataset_498640_1374404507.docx
(465.15K)

Word count: 2613

Character count: 13005

A research report on the classification of FER2013 dataset

2216701 Youqian Chen

Abstract— Emotion recognition is the process of recognizing human emotions, and the development of deep learning makes it possible to access human emotions through intelligence. In this paper, facial features in the FER2013 dataset are extracted using related networks such as VGG and ResNet to complete the classification of emotion. The results are also compared and analyzed, including confusion matrix, recall, precision and F1 score.

keywords— *FER2013 dataset, VGG, ResNet, emotion recognition*

1. Introduction

We first analyze the dataset and divide it into a training set, a validation set and a test set. After that, we compare different network models for feature extraction of the data and select the model with the highest accuracy in the validation set for confusion matrix and ROC (Receiver Operating Characteristic curve) plotting under different learning rates and activation functions.

2. Literature related work

2.1 FER2013 dataset

The data set used in this experiment is the FER2013 dataset [1], which was collected by psychologists Ekman and Friesen and divided into six categories, namely, angry, disgust, fear, happy, neutral, sad and surprise. the data for this experiment were selected from a subset of this dataset. The data for this experiment were selected from a subset of this dataset, mainly consisting of the following four categories angry, fear, happy and sad, of which the details of the dataset are shown in the table below. We use the training dataset to train the model, the validation dataset to verify the validity of the model, and 20% of the test dataset to compare the performance. Where the image size is 48*48 and the channel is 3 RGB color images.

Class name	Train num	Val num	Test num	Size*channel
angry	2797	1198	958	48*48*3
fear	2868	1229	1024	48*48*3
happy	5051	2164	1774	48*48*3
sad	3381	1449	1247	48*48*3

Table 1: Experimental dataset

2.2 DNNs

DNN deep neural networks[2] perform related tasks mainly by extending the perceptron, also called MLP (Multi-Layer perceptron). DNNs have many drawbacks mainly due to the fact that the connection method used is fully connected, which leads to an excessive amount of parameters, the size of the whole model and the speed of training is extremely slow, and it is particularly easy to fall into local optima. The specific structure of DNNs is shown below.

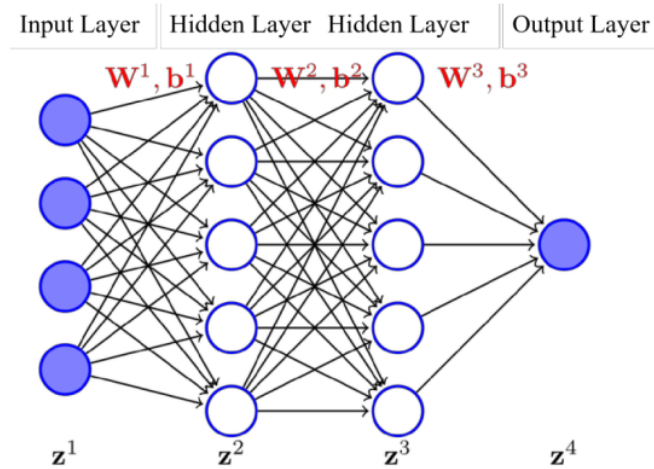


Figure 1: DNN network structure diagram

2.3 CNNs

CNNs (Convolutional Neural Networks) [3] are neural networks that perform general operations by using convolutional operations. The main relevant layers are the input layer, convolutional layer, pooling layer, activation function, and Affine layer. Convolutional neural networks use convolutional layers except for the Affine layer near the fully connected layer. The specific network structure is shown in the following diagram:

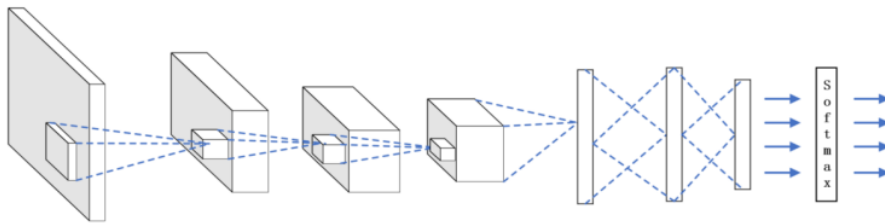


Figure 2:Convolutional neural network structure diagram

In general, incoming data need to be processed at this layer. This step can significantly optimize the image processing results, effectively prevent the input of different data units, significantly shorten the neural network training elapsed time and improve the network convergence rate; when running artificial neural networks, the effect of the underlying parameter update has a significant effect on them, causing effects that can lead to hypothesis conditions that are difficult to satisfy. Scientists engaged in related research adopt the normalization operation in neural networks, which can effectively solve the problem of unsatisfiable hypothesis conditions. This method usually scales up and down the data and is generally used to pre-process the data or to normalize the data to different units of measurement data.

The convolutional layer [4], the most important component of the convolutional neural network, is mainly the convolutional operation of the convolutional layer, which preserves the shape of the original data compared to the fully connected. The operation

is performed on the input data filter, also known as the convolution kernel, and is a multiplicative accumulation operation, with the bias added to the result mainly in the form of an addition by a broadcast method.

When performing the convolution operation, there are several important parameters when the filter performs the convolution calculation through the sliding window each time. The first one is the step length, which is the length of each slide, for the position interval of the filter; the second one is the depth, which is the number of neurons; the third one is the padding, which is the operation required when the length of the feature map cannot be completely traversed by the convolution kernel during the slide, before the processing of the convolution layer, a fixed amount of data is filled around the data, and the size of the output can be maintained after the padding and the The size of the output and the original data remain unchanged. The number of channels of the input data and the number of channels of the filter should be the same; the size of the filter can be set to any value, but the size of the channels should be the same; the number of feature maps of the output data is determined by the number of filters.

The pooling operation [5], also known as downsampling operation, mainly aims to reduce the parameters of the neural network by using the relevant features of the neighboring location regions as the output, i.e., a certain value instead of a certain region, mainly including maximum pooling and average pooling, etc. The parameters of the pooling layer mainly include the size of the pooling window and the size of the pooling step, in general, the pooling window size will be set to the same value as the step size, and the maximum pooling is used more than the average pooling.

The activation function as [6] is an important component of neural networks, mainly through nonlinear operations to determine the output of the relevant neuron more intuitively. When the input data changes in a small range, we do not expect the output value to change significantly. The activation function is a nonlinear operation that takes our input values and increases the expressiveness of the linear model. Earlier we

introduced convolution and found that convolution is actually a linear operation. For sample data, the vast majority of them are linearly inseparable, so to make them linearly separable, we can either perform a linear change operation or increase the nonlinearity by introducing an activation function. Since neural networks are differentiable everywhere, the design of the activation function must also require that the inputs and outputs of the data are also differentiable. Here we present several activation functions used in our experiments, ReLU and Leaky_ReLU.

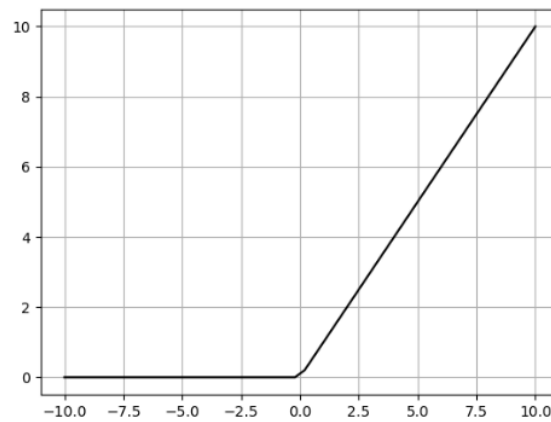


Figure 3: ReLU function diagram

The emergence of the ReLU function is of great significance, as shown in the figure. This function is one of the more popular and commonly used, the main advantages of the following two, the first is not a saturated function, when the input is positive, the output does not tend to a certain value; the second is that the linear relationship brings a very fast calculation.

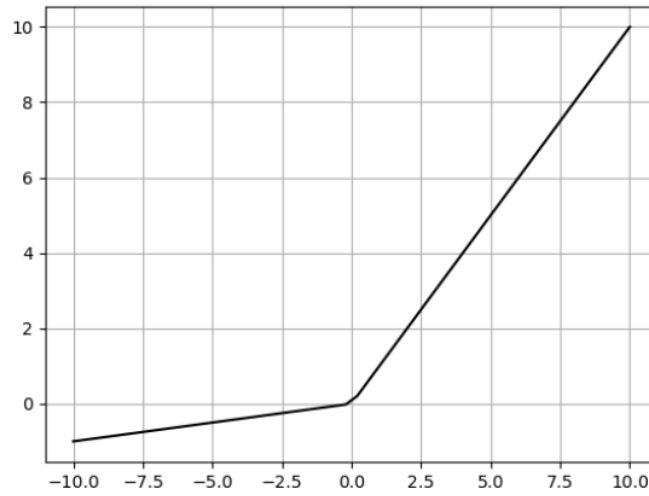


Figure 4: Leaky ReLU function diagram

The above ReLU function also has drawbacks, the most obvious being that the ReLU function is completely deactivated when the value of the input is negative. To solve this problem, the Leaky ReLU function was designed as shown in the figure. It can be noticed that Leaky ReLU regulates the problem of zero gradient by assigning a very small component of x to a negative input.

2.4 performance metrics

P is the precision rate (precision), also called the accuracy rate, which tends to focus on the accuracy of the lookup, and R is the recall rate (recall), also called the full rate, and the main formula is shown below. In order to evaluate the effectiveness of this model for express identification system, we use precision (P) and recall (R) together to evaluate the algorithm[7].

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

From the above equation, the precision rate P is the ratio of the number of positive cases

predicted accurately by the model to the number of positive cases predicted, which is used to measure the ability of the model to discriminate between positive and negative cases. The recall rate R is the ratio of the number of positive cases accurately predicted by the model to the number of positive cases in the sample, which is used to measure the model's ability to recall positive cases. Research shows that the relationship between P and R is somewhat inversely proportional, and when the accuracy rate increases, the recall rate decreases. Therefore, accuracy and recall cannot ensure the optimum at the same time, so it would be unfair to evaluate the overall performance purely from accuracy or recall, and the P-R graph is usually plotted to show the performance of the model with recall as the horizontal coordinate and accuracy as the vertical coordinate. The confusion matrix is also used to measure the performance of the tested images, and the specific binary confusion matrix is shown in the following table.

Reference \ Prediction	1	0
1	Ture Positive	False Negative
0	False Positive	Ture Negative

Table 2: Meaning of TP, FN, FP, TN

3. Submission

3.1 ResNet

ResNet (Residual Network) [8] is a network structure proposed by Kaiming He in 2015, and the idea of this network and network has been widely used in various feature extraction. The most important thing in ResNet is the proposal of residual block, before the residual block is proposed, when the number of layers is accumulated to a certain level, the overfitting phenomenon of the model will appear and the problem of accuracy degradation will occur. The proposed residual block not only strengthens the expressiveness of the network, but also can solve the problem of gradient disappearance and explosion.

3.2 VGG

The design concept of the VGG (Visual Geometry Group) network [9] is to double the number of channels when the resolution of the image is reduced, and all the feature extraction networks that follow adopt this idea, as shown in Figure 2.9 for the network structure of VGG16.

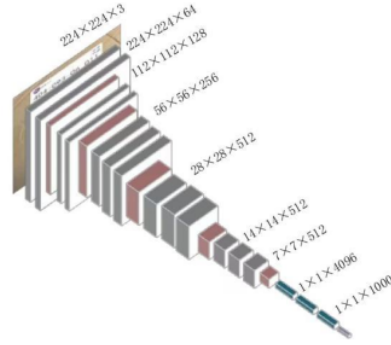


Figure 5: Network structure diagram of VGG16

The network structure of VGG16 mainly consists of 13 convolutional layers, 5 pooling layers and 3 fully connected layers. Each convolutional layer has the same size, 3×3 convolutional kernels, and the step size and padding are 1. Two 3×3 convolutional kernels are equivalent to the perceptual field of a 5×5 convolutional kernel, and three 3×3 convolutional kernels are equivalent to the perceptual field of a 7×7 convolutional kernel. vgg16 reduces the number of parameters by stacking small convolutional kernels, while keeping the perceptual field unchanged. The first one is the small convolutional kernels, and the size of all the convolutional kernels used is 3×3 ; the second one is that the pooling kernels in the pooling layer are all 2×2 ; the third one is that the structure is deeper and wider compared to the previously proposed networks.

3.3 Modification of C-related parameters

For training, the pixel values of the images are first normalized by a normalization operation and randomly flipped and regularized using the method of transforms. A cross-entropy loss function is used and the optimizer is mainly used by the stochastic gradient descent method [10]. Different models are compared, mainly including

ResNet18, VGG11, and a convolutional neural network sCNN (small CNN) that was built by myself.

Input dimension	Kernel	Stride	Pooling
3	7×7	2	3
64	3×3	1	1
64	3×3	1	1
64	3×3	1	1
64	3×3	1	1
128	3×3	2	1
128	3×3	1	1
128	3×3	1	1
128	3×3	1	1
256	3×3	2	1
256	3×3	1	1
256	3×3	1	1
256	3×3	1	1
512	3×3	2	1
512	3×3	1	1
512	3×3	1	1
512	3×3	1	1
Fully connected (512×4)			

Table 3 ResNet18 model parameters

Input dimension	Kernel	Stride	Pooling
3	3×3	1	1
64	3×3	1	1
128	3×3	1	1
256	3×3	1	1
256	3×3	1	1
512	3×3	1	1
512	3×3	1	1
512	3×3	1	1

Fully connected (512×64)
Fully connected (64×64)
Fully connected (64×4)

Table 4 VGG11 model parameters

Input dimension	Kernel	Stride	Pooling
3	3×3	1	1
16	3×3	1	1
32	3×3	1	1
64	3×3	1	1
Fully connected (2304, 500)			
Fully connected (500×4)			

Table 5 sCNN model parameters

Different activation functions, Relu and Leak Relu, in resnet were compared and pre-trained models of ResNet were loaded for comparison. And since the validation set of VGG model has the highest accuracy, the starting learning rate of VGG model is modified and 0.001 and 0.01 are selected for the experiments, and the related experimental results are shown in the following table.

Models	Learn rate	activation function	Epoch	Accuracy	Loading pre-training
VGG11	0.01	Relu	30	63.82%	No
	0.001	Relu	30	65.11%	No
	0.001	Relu	150	65.69%	Yes
ResNet18	0.001	Leaky_ReLU	30	60.12%	No
	0.001	Relu	30	59.46%	No
	0.001	Relu	30	60.53%	Yes
Small network	0.001	Relu	30	49%	No

Table 6 Model experimental results

From the table, we can find that the accuracy of VGG11 network is high compared to

that of ResNet, and the accuracy of the small convolutional network built is not high on this dataset, but the skills and methods of model building are mastered to build the corresponding feature extraction network.

4. Results and Discussion

It was found that the final best network on the validation set was VGG11 with a learning rate at 0.001. Validation was performed on the test set.

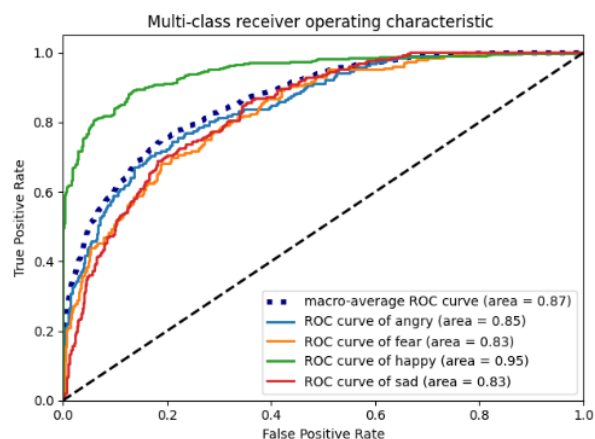


Figure 6: ROC curve

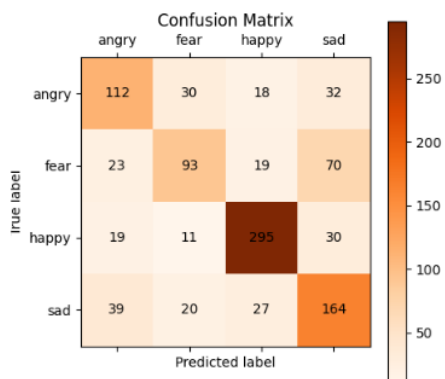


Figure 7: Confusion Matrix

accuracy: 0.6626746506986028
precision: 0.6399970145694451
recall: 0.6309944463529142
F1 Score: 0.6317467910864601

Figure 8: Accuracy, accuracy, recall, F1

5. Conclusion

In this paper, by comparing different models, different learning rates and activation functions, the classification of sentiment is completed and an accuracy rate of about 66% is obtained. And in the given test set, the model is completed and the test results are as expected, which confirms the correctness and validity of the model and the experiment.

References:

- [1] Zahara L, Musa P, Wibowo E P, et al. The facial emotion recognition (FER-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (CNN) algorithm based Raspberry Pi[C]//2020 Fifth international conference on informatics and computing (ICIC). IEEE, 2020: 1-9.
- [2]Zhang J, Zheng Y, Qi D, et al. DNN-based prediction model for spatio-temporal data[C]//Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems. 2016: 1-4.
- [3]Alzubaidi L, Zhang J, Humaidi A J, et al. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions[J]. Journal of big Data, 2021, 8: 1-74.
- [4]O'Shea K, Nash R. An introduction to convolutional neural networks[J]. arXiv preprint arXiv:1511.08458, 2015.
- [5]Sun M, Song Z, Jiang X, et al. Learning pooling for convolutional neural network[J]. Neurocomputing, 2017, 224: 96-104.
- [6]Nwankpa C, Ijomah W, Gachagan A, et al. Activation functions: Comparison of trends in practice and research for deep learning[J]. arXiv preprint arXiv:1811.03378, 2018.
- [7]Lei X, Pan H, Huang X. A dilated CNN model for image classification[J]. IEEE Access, 2019, 7: 124087-124095.
- [8]He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
- [9]Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
- [10]Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

Video link

https://www.youtube.com/watch?v=Eqq_Tdg7WwA

A research report on the classification of FER2013 dataset

GRADEMARK REPORT

FINAL GRADE

72/100

GENERAL COMMENTS

Instructor

In abstract, there is no harm in including some summary results.

Refs are appropriate and used well in general.

Some minor typos, etc but in general well structured and presented report.

Regarding related work, did you find any other papers that experimented on the same dataset with similar networks? These could be used to validate your results.

Appropriate theory and related concepts identified and discussed and used to guide experimental design. Good.

Please spend some time explaining/justifying your class selections- ie happy, sadness, fear, disgust. This is important since your choices can affect the the training success - accuracy, etc of your network. Would using additional classes in training help improve the accuracy of these 4?

A nice experiment comparing your own with other established architectures. Good.

Discussion of using pre-trained network weights (transfer learning) could be developed further here.

Use of further pre-processing and potential of dimensionality reduction could be included.

Maybe discuss further the relevance of validation data in optimising no of epochs of training.

Overall a very good effort. Well done.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7



Comment 1

Choose a more meaningful section title here.

PAGE 8



Comment 2

reference your source in the figure title

PAGE 9

PAGE 10

PAGE 11

PAGE 12



Comment 3

state your references used to assert this.

YOUQIAN CHEN

0

RESEARCH-INFORMED LITERATURE

KNOWLEDGE AND UNDERSTANDING

ANALYSIS

PRACTICAL APPLICATION AND DEP.

SKILLS FOR PROFESSIONAL PRAC.