Pattern Classification Using CNN

Mikhail Podvalkov

Due to March 24, 2022

1. Introduction

The data for learning was a CIFAR-10 which consists of color images divided into three parts training, validation, and testing. The computer assignments aim to design the architectures of a stacked three-layer neural network and find its parameters. The computer assignments created a CNN model in MATLAB code [1] with different learning algorithms. Moreover, the model weights were defined. After that, the best model was trained several times to ensure that final validation accuracy did not depend on the random initiation. The computer assignments provided the learning curve of the designs models.

2. Theory

The convolution neural network is the unique neural network where the input data is plane (image) [2] (fig 1).

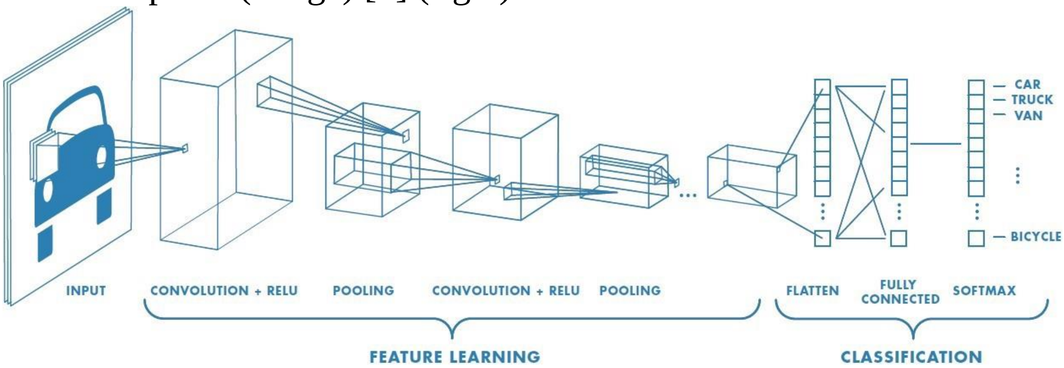

Figure 1 The CNN model

The CNN contains two parts: the feature learning layers and the fully connected layers used for classification. The feature learning layers extract the feature map of the input image. For this extraction, use the two types of layer convolution and pooling. Convolution layers are square matrices where all elements are weights. The number of each matrix in one layer could be several, which increases the width of the input image. The pooling layer is a matrix that averages the input, i.e., the layer compresses information incoming on the layer.

The mathematical part of the network is similar to a neural network except for planes and transforming into the 3D matrix.

$$net_{i,j,k} = w_k * x_{i,j} + b_k$$
$$y_{i,j,k} = f(net_{i,j,k})$$
$$v_{i,j,k} = pool(y_{m,n,k})$$

The computer assignments had two types of learning algorithms Stochastic gradient algorithms and RMSprop

The stochastic gradient algorithms:

1. Initiate the 0 parameters w_k randomly
2. Find gradient vector $\frac{\partial J(w(k))}{\partial w(k)}$
3. Update parameters w_{k+1}=w_k +μ*$x_{i,j}$*e(W)
4. Go back to step 2

The RMSprop algorithm:

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1-\beta)\left(\frac{\partial C}{\partial w}\right)^2$$

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}}\frac{\partial C}{\partial w}$$

where $E[g^2]_t$-a moving average of the squared matrix, $\frac{\partial C}{\partial w}$ – gradient of cost function respective to weights, η-learning rate, β- moving average parameters.

3. Results and discussion

The Matlab code provides several CNN architectures a training test on validation data using Deep Learning Toolbox [3]. The architecture of these models is presented in table 1.

Model 1	Model 2	Model 3
Conv(3x3 filters 32)	Conv(3x3 filters 32)	Conv(3x3 filters 32)
Pool(2 relu)	Pool(2 relu)	Pool(2 relu)
Conv(3x3 filters 64)	Conv(3x3 filters 64)	Conv(3x3 filters 32)
Pool(2 relu)	Pool(2 relu)	Pool(2 relu)
Conv(3x3 filters 128)	Conv(3x3 filters 64)	Conv(3x3 filters 64)
Fully connected (128)	Fully connected (64)	Fully connected (64)
Fully connected (10)	Fully connected (10)	Fully connected (10)
softmax	softmax	softmax

Table 1 The architecture of models.

The learning curves are presented in figure 2.

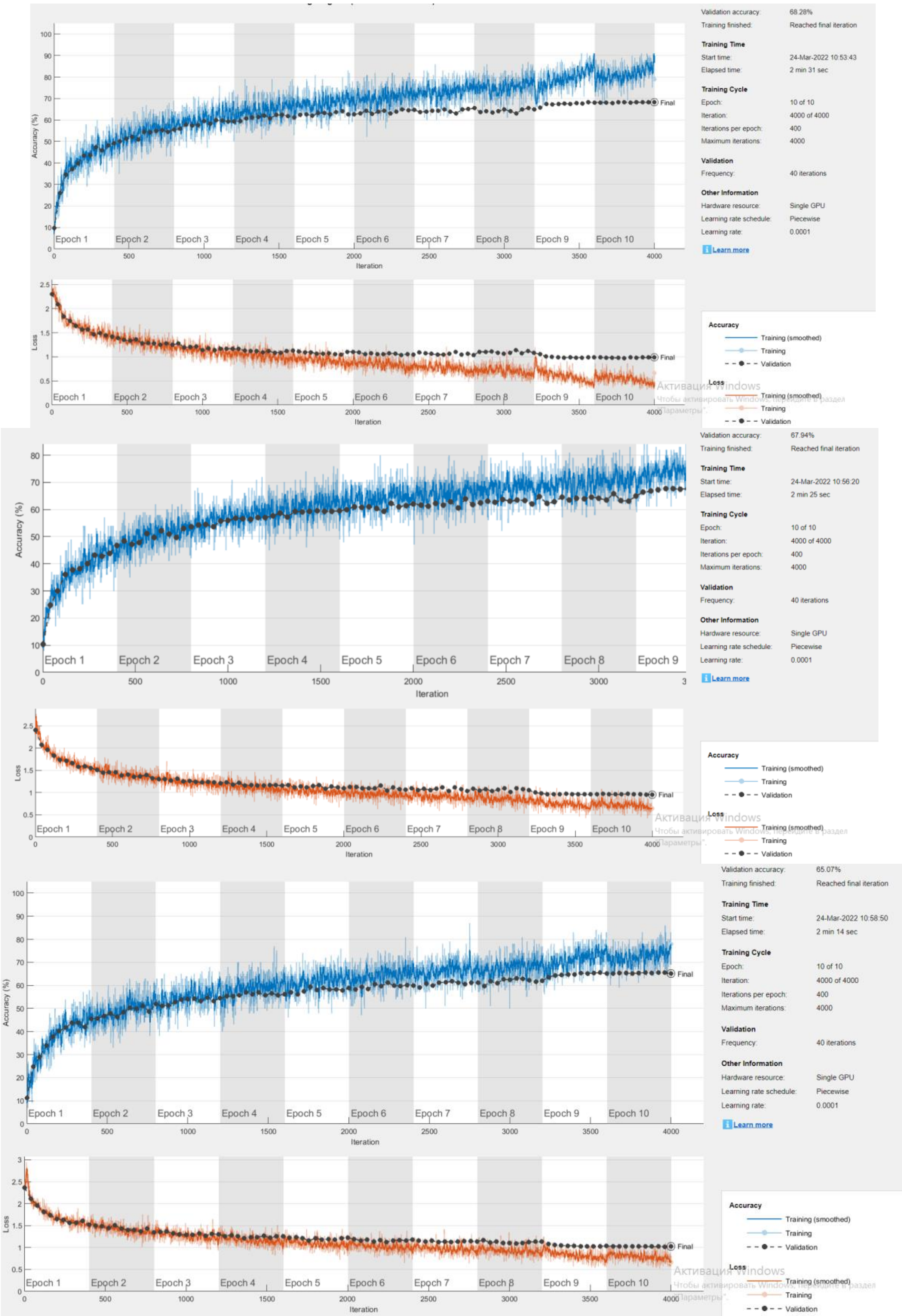

Figure 1 The learning curve of the first, second and third model

The best accuracy on the validation set was admired on the first model, which allows us to conclude that the architectures where the number of filters in the layer increase is better than the other. Also, the fully connected layers significantly influenced classification because this part of the network works with the feature map and chooses in which class the input images.

After that, the best network was tested on the different training options like training algorithms. The result is presented in figure 2.

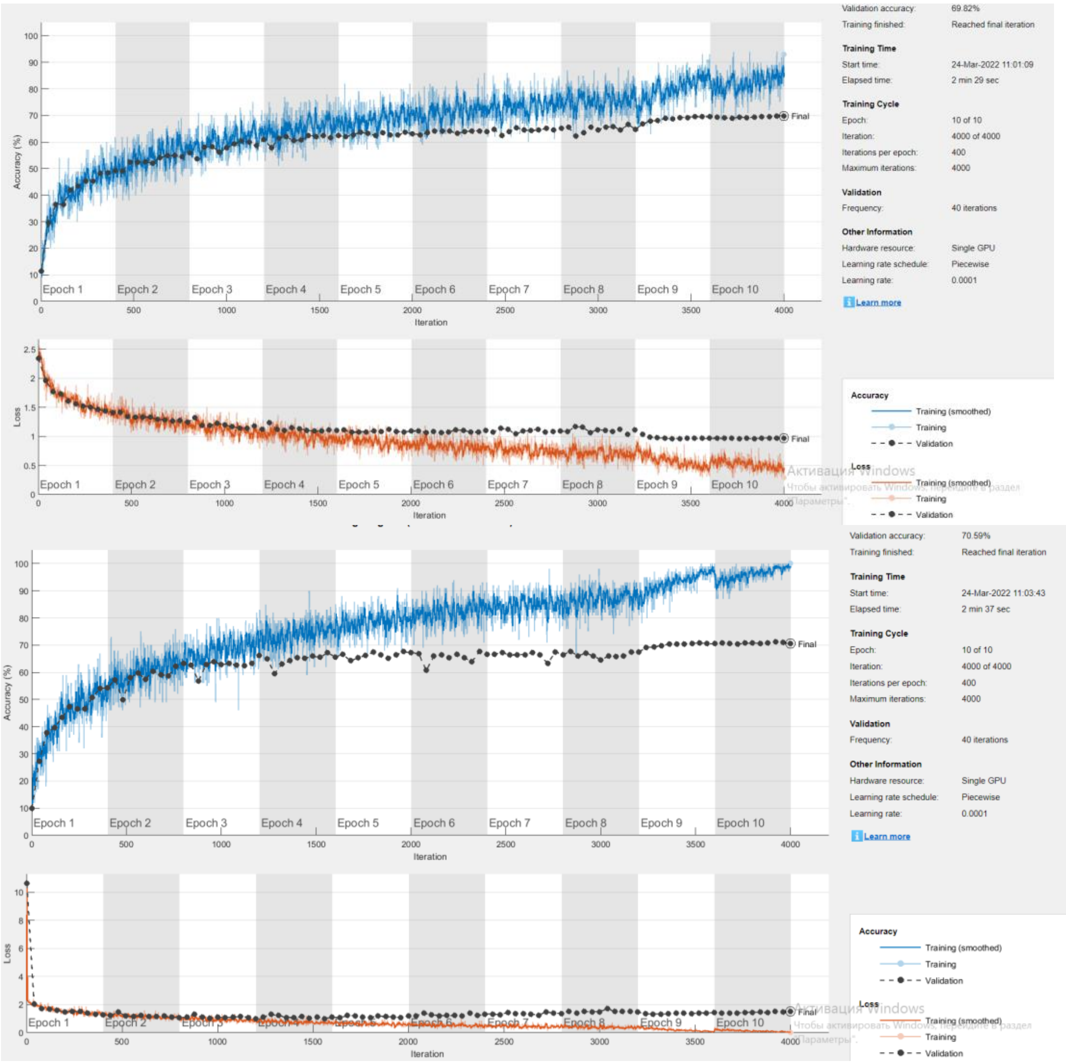

Figure 2 The learning curve of SGDM and RMSprop model

The higher accuracy on the validation set was reached on the RMSprop algorithm because the SGM is not allowed to change gradient with others parameters of the system, which could influence networks like the RMSprop. RMSprop considers a gradient of cost function respective to weights, allowing better performance on the more complex data.

The weights of the model were initialized randomly. It could influence the accuracy; that is why in computer assignments, trained the best model several times to check the influence of initialization. The result is presented in Table 2.

Number of trains	1	2	3	4	5
Validation accuracy	71.8%	70.66%	71.73%	71.14%	70.98%

Table 2 The Validation accuracy of the same model on different initial conditions of weights

The changes in validation accuracy due to weight initialization are less than 1%, which has no significant impact on the final performance of the network. The test accuracy is equal to 71.6%, within the normal range for this type of task

4. Conclusion.

The advantages of the convolution neural network are robust to mirroring and pre-processing the input data that allows creating more training data and getting an accurate result. The second is the CNN east to optimize such as paralyzed on several GPU or optimize the matrix multiplication. The third is that the convolution network is the best way to find the input features, and it does not matter what data type is. The main disadvantage of CNN is that it needs a lot of data and rotations, reflections this data to get high trained. In the second network, the training process is relatively slow. The third CNN needs much more complicated computation, which requires high-performance hardware.

Reference

1. The official MATLAB site: https://www.mathworks.com/
2. The professor M.R Azimi lecture 13-14 page 7-10
3. The official MATLAB Deep Learining toolbox site: https://www.mathworks.com/products/deep-learning.html