IMAGE CLASSIFICATION ON CIFAR-10 DATASET

YAN-YAN Wang
College of Science, Zhongyuan University of Technology, Zhengzhou, China

Abstract. In this project, we work on image classification of the CIFAR-10 datasetusing supervised machine learning techniques. The dataset consists of 60,000 32x32RGB images containing one of 10 object classes, with 6000 images per class. Weexperiment with various learning algorithms including nearest neighbor classifier,one-vs-all classification, Softmax classifier, two-layer fully connected artificial neural network (ANN), deep convolutional neural network (CNN), and deep residualnetworks (ResNet). We use cross-validation by splitting the 50,000training data into49,000 training samples and 1,000 validation samples to select the optimized hyperparameters for each parametric classifier. Among all methods, the 56-layer deepresidual network yields the best performance with a training accuracy above 99% anda validation accuracy of 93.6%.

Key words: *image classification; CIFAR-10; supervised machine learning algorithm; deep convolutional neural network (CNN); deep residual network (ResNet)*

1 INTRODUCTION

TheCIFAR-10datasetcontains of 60,00032x32colorimagesof10classes, with6000foreach class[1]. Figure 1 shows the sample images from the dataset. Over the years, a lot ofworks have been reported regarding the image classification problem with CIFAR-10dataset[2-4]. The highest accuracy so far is achieved by using modified convolutional neural network(CNN) with fractional max-pooling [5].

Figure 1: Examples of CIFAR-10 dataset[1].

In this project, the CIFAR-10 dataset was divided into 50,000 labeled training images and 10,000 unlabeled testing images. We further divided the training set into 49,000 training samples and 1,000 validationsamples to select the best model and hyperparameters. The classifier trained on the49,000 samples with the optimized parameters was then used to predict on the testing set andevaluate the prediction accuracy.During this project, we experimented with both non-parametric and parametricmethods for image classification. We started with the non-parametric nearest neighbor classifier (NN), which only gave a validation accuracy around23.9%. Then we experimented with various parametric methods, which included bothlinear and non-linear classifiers. The linear methods such as One-vs-all classifier andSoftmax classifier had validation accuracy around 40%. The non-linear methods, onthe other hand, gave a validation accuracy from 50% (two-layer artificial neural net-work (ANN)) to 80% (plain convolutional neural network (CNN)).

Finally, we experimented with the deep residual networks (ResNet)[6]. After carefully designing the network structures and tuning the hyperparameters, we got the best performance with a 56-layer ResNet. The network showed atraining accuracy above 99%, a validation accuracy of 93.6% and a testing accuracy of92.58%. The ResNet is therefore chosen as the final candidate for this project. The comparison of different classifiers' performances was shown in Figure 2.

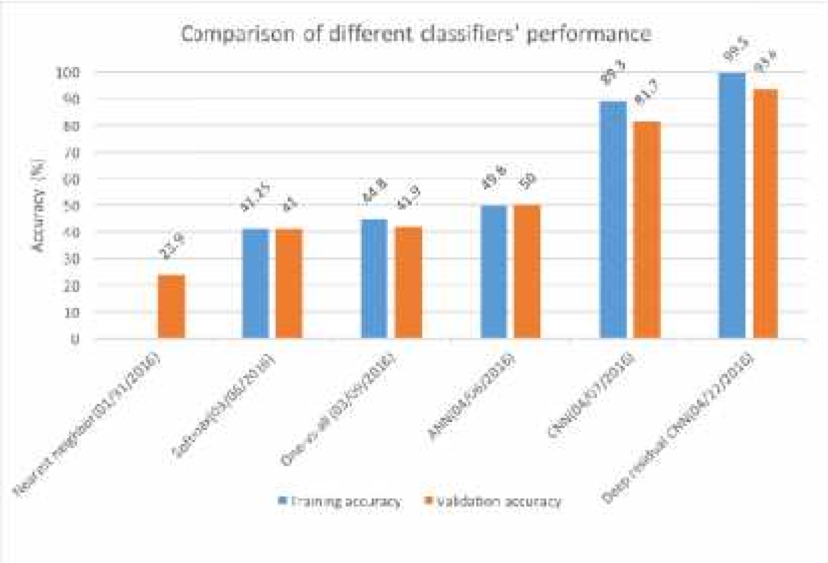


Figure 2: Training and validation accuracy of different classifiers.

2 METHODS

2.1 Deep Residual Network (ResNet)

The general convolutional neural network (CNN) consists of multiple layers that transform the input image volume into an output volume holding the class scores. The severaldistinct types of layers are convolutional layer, RELU layer, POOL layer and fully-connected layer. For a convolutional neural network, the most important layer is theconvolution layer, where each entry in the output volume can be interpreted as an output of a neuron that looks at only a small region in the input and shares parameterswith neurons in the same activation map[7].

A lot of breakthroughs for image classification have been made with deep CNNby stacking more and more convolutional layers. However, when deeper networks areable to start converging, the degradation problem occurs[6]. The accuracy gets saturated and then degrades rapidly, and adding more layers to a suitable deep model willlead to higher training error. The degradation problem was addressed by KaimingHe, et al by introducing a deep residual learning framework[6]. In this approach, insteadof directly fitting the desired mapping, the layers are made to explicitly fit a residualmapping.

Formally, suppose the desired mapping is H(x), the plain CNN directly fits themapping H(x),while the ResNet fits another mapping of F(x) = H(x) − x. Therefore, the original mapping is recast into F(x) + x, and it could be realized by feed forward neural networks with "short connections". The short connections simply performidentity mapping and their outputs are added to the outputs of the stacked layers[6].Figure 3 shows the comparison between the building block of plain CNN and ResNet.By stacking the building blocks together, very deep residual networks (30-100 layers)could be developed without the degradation problem.

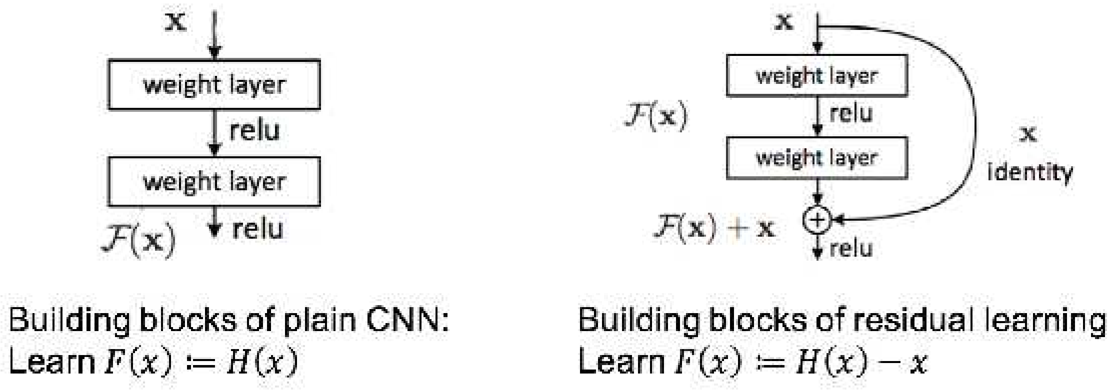


Figure 3: Comparison of the building block of plain CNN (left) and ResNet (right)[6].

2.2 Microsoft Azure Cloud Computing

Due to the high computational cost of the neural networks, we decided to use MicrosoftAzure cloud computing tool. The configuration of the virtual machine we used was:Dv2-series based on 2.4 GHz Intel Xeon E5-2673 v3 (Haswell) processor, Ubuntu14.04 system, 8 cores, 56 GB RAM, 400 GB disk sizes.

3 RESULTS AND DISCUSSION

3.1 Non-parametric Method

In this section we experimented with non-parametric Nearest Neighbor Classifier (NN).Basically the NN classifier will take a test image, compare it to every single one of thetraining images, and predict the label of the closest training image. The differencebetween two images is calculated pixel-wise using the L1 norm:

$$d_1(I_1, I_2) = \sum_p |I_1^p - I_2^p|$$

We evaluated the average validation accuracy on the 1,000 images. After repeatingthe experiments several times, we found that the performance of NN classifier is relatively poor since it only uses raw pixel values with no pre-processing. It has average validation accuracy around 23.90%, which is only twice that of random guess (10%).

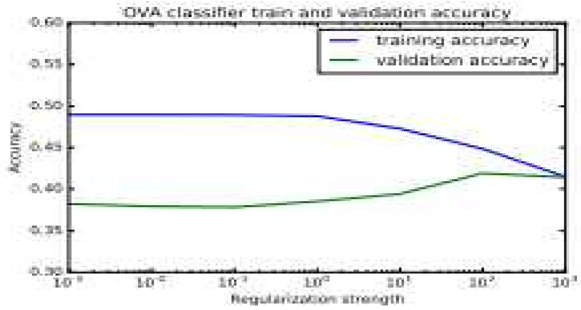
3.2Linear Methods

3.2.1 One-vs-All (OVA)

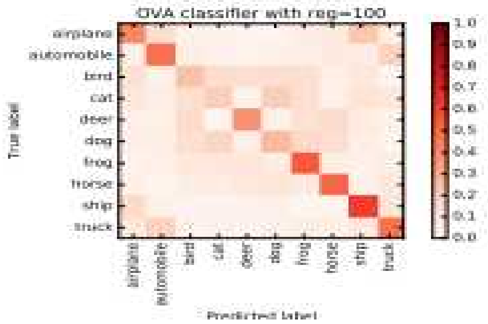
We used the One-vs-all classification (OVA) by training 10 different classifiers using python Scikit-learn module. The images were preprocessed to normalize the pixel values by dividing by 255anddeductingthemeanofthetrainingsetfromboththetrainingset, validation set and testing set. Then for linear models the 32x32x3 images wereflattened and transformed to 3072x1 vector. For the i^{th} classifier, we let the positiveexamples be all the images in class i and the negative examples be all the points not inclass i. Let h_i be the i^{th} classifier, the problem was to compute:

$$h(x) = \arg\max_i h_i(x)$$

We used L2 norm to control overfitting and tuned the regularization strength λfor best validation accuracy. The impact of λ on the training and validation accuracy is shown in Figure 4(a). The figure indicated that λ = 1e2 gave the best performance, with the training accuracy being 44.84%, the validation accuracy being 41.9%, andthe testing accuracy being 39.08%. The corresponding confusion matrix is shown inFigure 4(b). The confusion matrix indicates that the bird, cat, deer and dog classes' arerelatively harder to classify compared with the truck, automobile, etc.



(a)



(b)

Figure 4: (a) Impact of λ on the training and validation accuracy of OVA classification.
(b) Confusion matrix of OVA classification with λ = 1e2.

3.2.2 Softmax Classifier

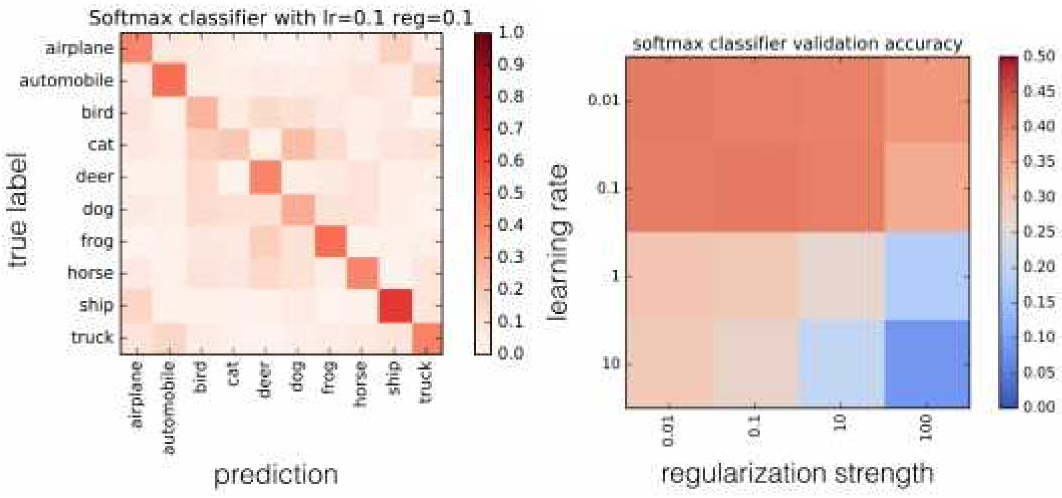
We implemented softmax logistic regression after applying the preprocessing technique as detailed previously. Then we computed the loss functionand gradient, with L2 regularization. In order to implement gradient descent on large-scale dataset, we used mini-batch approach, where we computed the gradient over a256 batch of the training data and then perform a single parameter update. Next, we swept over a certain range for several parameters in an attempt to find the optimal learning settings. The hyperparameters we tuned were:

Learning rate α = [1e2, 1e1, 1e0, 1e1]

Regularization strength λ= [1e-2, 1e-1,1e1, 1e2]

From the experiments we found out that the best soft max classifier had the learningrate of 1e-1 and regularization strength of 1e-1. The best soft max classifier yield an accuracy of 41.25% on training data set, 41.10% on validation data set, and 39.93% ontest data set.

The impact of learning rate and regularization strength on validation accuracy is shown in Figure 5(a). The confusion matrix with the best hyper-parameters is shownin Figure 5(b). From the confusion matrix we observe that the bird, cat and dog classesare more difficult to correctly classify than the transportation classes.



(a) （b）

Figure 5: Accuracy and confusion matrix of softmax classification. (a) Impact oflearning rate and regularization strength on validation accuracy. (b) Confusion matrixof OVA classification with best hyper-parameters $\alpha = 1e - 1$, $\lambda = 1e - 1$.

The visualization of theta under the best learning parameters is:

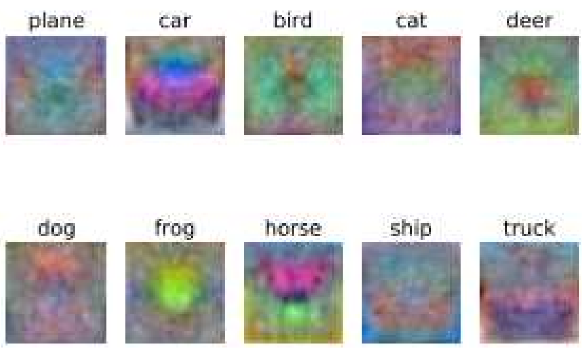


Figure 6: Visualization of theta under best learning parameters.

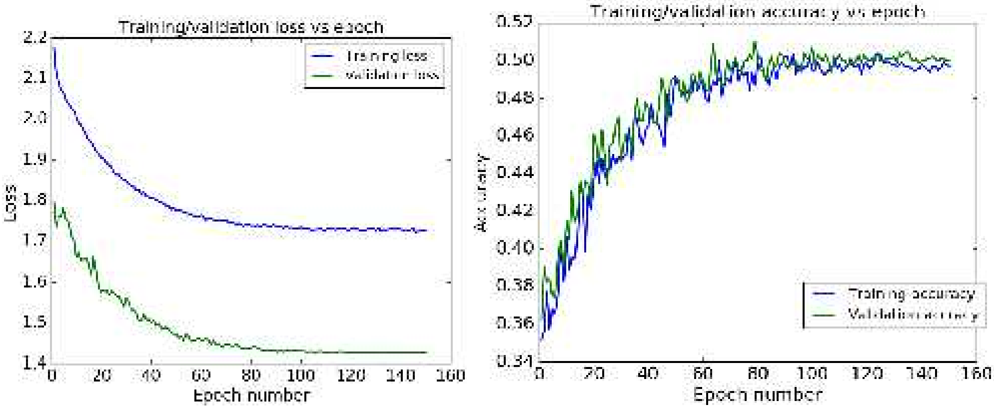
3.3 Non-linear Methods

For all non-linear classifiers, the images were preprocessed using the technique described in section 3.2.1.

3.3.1 Fully-Connected Artificial Neural Network (ANN)

In this section, we experimented with the fully-connected artificial neural network(ANN). The code was developed using Python Lasagne package. Wetuned the hyperparameters including the number of hidden layers, the number of unitsin each hidden layer, batch size and update rule. During the experiments we observedthat two-layer ANN generally had better performance than multiple-layer ANNsince the latter could be easily overfitted.

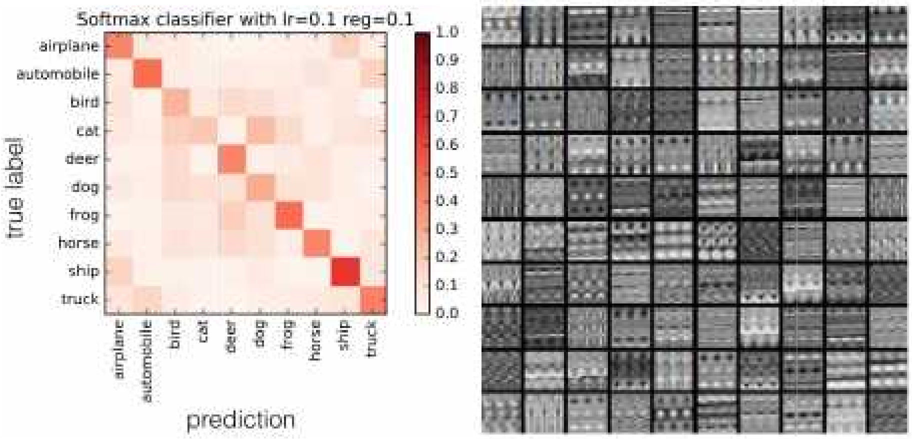
Therefore, the best network structure we selected was: two-layer ANN with 100hidden units. We set the dropout rate p=0.5 to control overfitting. The learning rate was 0.1at the begin, with a learning rate decay of 0.95. We tuned the batch size to be 128, 256and 500, and the highest validation was achieved using batch size 256, 150 epochs and SGD update rule. The training history is shown in Figure 7.



(a) (b)

Figure 7: Learning history of two-layer ANN. (a) Training and validation loss vs number of epochs. (b) Training and validation accuracy vs number of epochs.

The confusion matrix and the visualization of the first-layer weights are:



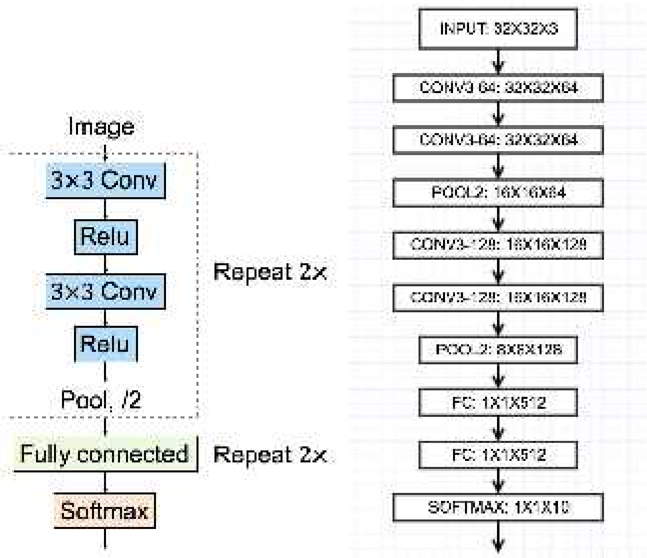
(a)(b)

Figure 8: Confusion matrix (a) and first layer weights visualization (b) of the two-layerANN.

3.3.2 Plain Convolutional Neural Network (CNN)

In the experiments, we evaluated the performance of a 15-layer CNN adapted from reference 8,where the input andoutput data types were modified to accommodate our dataset. The network structure wasbased on the codebase, while the training protocol, network performance evaluationmodule were developed by ourselves. The structure of the 15-layer CNN is shown inFigure 9.

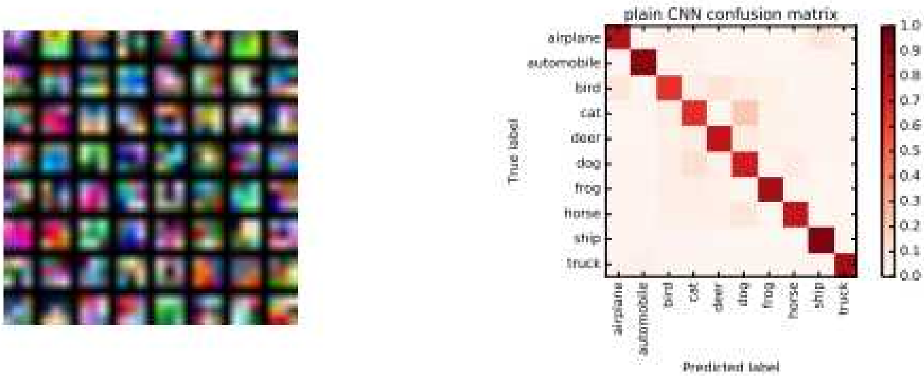
The average time to train a plain CNN was around 16 hours, therefore we didn'tget enough time to sweep for optimal parameters.



(a)(b)

Figure 9: Structure of the 15-layer plain CNN.

The best performance was achieved when we set learning rate=0.01, batchsize=500, epoch=300. The dropout rate of the last two fully connected layers p was set to 0.5 to control overfitting. The L2 normalization term was set to 0. The visualization offirst layer filters and the confusion matrix are shown in Figure 10(a), and Figure 10(b), respectively. The visualization of the filters indicates that CNN is able to capture moredetailed features of the images than ANN.



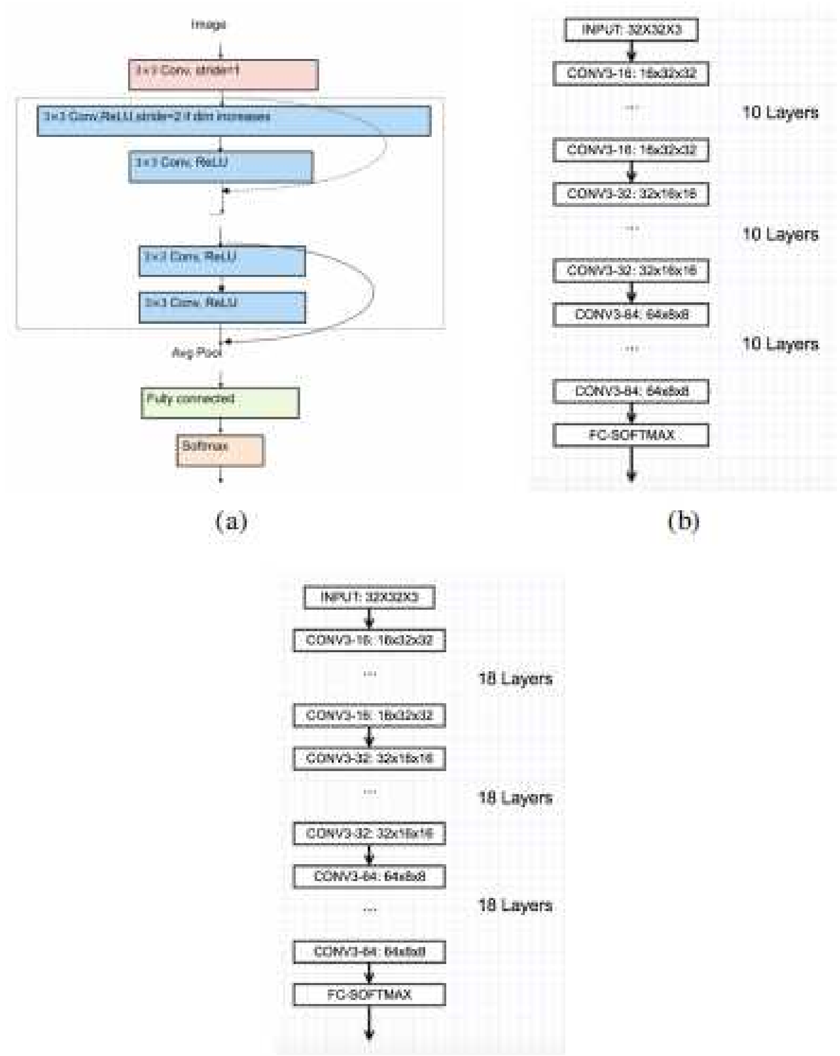
(a) (b)

Figure 10: The visualization of the first layer filters and the confusion matrix of 15-layer plain CNN.

The training accuracy with the network is 89.3%, the validation accuracy is 81.7%,and the testing accuracy is 80.01%. The confusion matrix indicated that the classifierhad a rather good sensitivity against all 10 classes.

3.3.3 Deep Residual Network (ResNet)

For the deep residual network (ResNet), we tuned the number of layers and experimented with both 32-layer ResNet and 56-layer ResNet. The structure of both networks is shown in Figure11. Thecodeswereadaptedfrom reference 9. Even with Microsoft cloud computing, it still took 48 hours to train the 32-layer ResNet and 160 hours for 56-layer ResNet. Due to time limit, wewere unable to do large scale parameter sweeping, and we only experimented with 1or 2 sets of parameters for each network architecture.

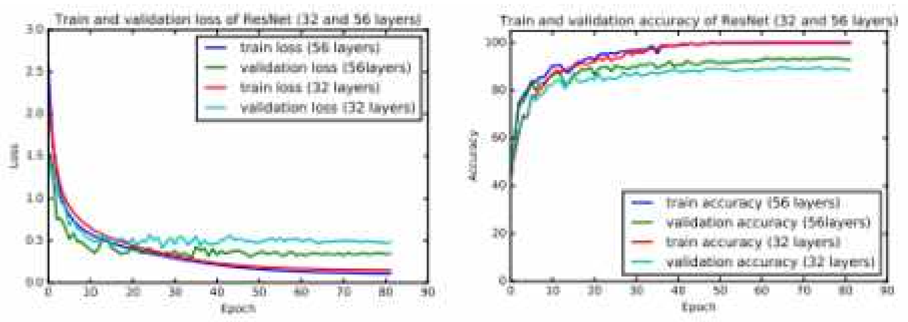


(a) (b)

(c)

Figure 11: The structure of a general ResNet (a), the 32-layer ResNet (b) and the56-layer ResNet (c).

To improve the performance of the algorithm, we mirrored the training images toget 98,000 training samples in total. We trained 82 epochs for both networks. Thelearning rate was set to 0.1 at the beginning, with a decay rate of 0.95. The traininghistory of the two deep ResNet is shown in Figure 12.



(a) (b)

Figure 12: (a) The training/validation loss vs number of epochs ; (b) training/validation accuracy vs number of epochs of ResNet.

We further analyzed the change of the confusion matrix and the first layer filtersduring the training process of the 56-layer ResNet, which is shown in Figure 13.

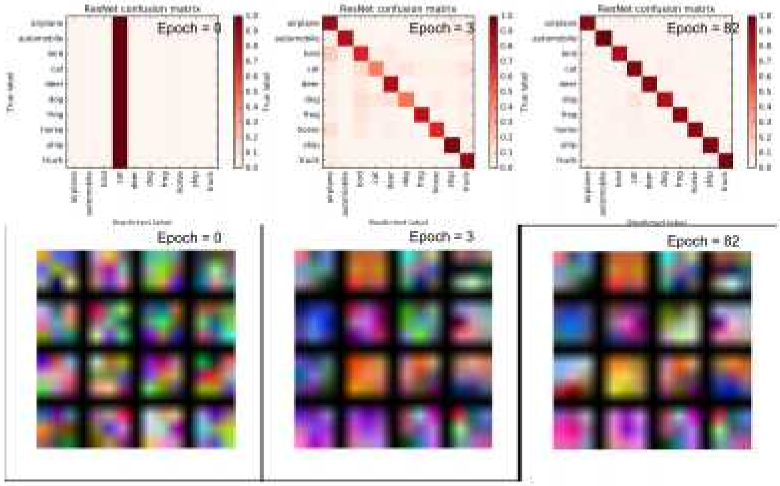


Figure 13: Change of the confusion matrix and the first layer filters during the trainingprocess of the 56-layer ResNet.

The above figures show that during the training process, the categorization of eachclass became more and more clear, and the filters gradually learned more detailed andspecific feature sets of the images.

The 32-layer ResNet had a training accuracy of above 99%, a validation accuracyof 91.6% and a testing accuracy of 90.95%. The 56-layer deep CNN had the bestperformance in all experiments. The network yields a training accuracy of 100%, avalidation accuracy of 93.6% and a testing accuracy of 92.6%.

4 CONCLUSION

In this project, we achieved the best classification accuracy of the CIFAR-10 datasetwith 56-layer deep residual network. Through this project, by experimenting with multiple linear/non-linear classifiers and tuning the hyper-parameters, we could concludethat linear classifiers (OVA, Softmax, etc.) normally have a bottleneck accuracy around40%. Non-linear neural networks have a much better performance, with fully-connectedANN having accuracy higher than 50%, and plain CNN around 80-90%. ModifiedCNN, such as fractional maxpooling and ResNet will further increase the accuracy toabove 90%.

Besides, we learned that the control of overfitting is important for classifiers withlarge set of parameters. By adding a penalty term for linear classifiers, or usingdrop-out technique, we could effectively control overfitting and improve the predictionaccuracy on the testing dataset.

ACKNOWLEDGMENT

This work is supported by the grants of the higher education development

REFERENCES

[1]Krizhevsky, Alex, Vinod Nair, and Geoffrey Hinton. "The CIFAR-10 dataset." *online: http://www.cs.toronto.edu/kriz/cifar. html* (2014).

[2]Lee, Chen-Yu, Patrick W. Gallagher, and Zhuowen Tu. "Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree." *Artificial Intelligence and Statistics*. 2016.

[3] Springenberg, Jost Tobias, et al. "Striving for simplicity: The all convolutional net." *arXiv preprint arXiv:1412.6806* (2014).

[4] Krizhevsky, Alex, and G. Hinton. "Convolutional deep belief networks on cifar-10." *Unpublished manuscript* 40 (2010): 7.

[5] Graham, Benjamin. "Fractional max-pooling." *arXiv preprint arXiv:1412.6071* (2014).

[6] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[7] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[8] jseppanen/cifarlasagne: Cifar-10 image classification with lasagne, https://github.com/jseppanen/cifar_lasagne. (Accessed on 04/24/2016).

[9] Recipes/deep residual learning cifar-10.py at master lasagne/recipes, https://github.com/Lasagne/Recipes/blob/master/papers/deep_residual_learning/Deep_Residual_Learning_CIFAR-10.py.(Accessed on 04/24/2016).