ENEE 633 - PROJECT 2

Jing Xie
12/18/2020

Part I: Hand-written Digit Recognition

Requirement: Implement linear and kernel SVM on MNIST dataset. You have to try different kernels (linear, polynomial, RBF) and compare results in your report.
Method:
LDA is applied here for dimensionality reduction. SVM is trained with "one vs. All" algorithm. Due to the large size of the data and the limited training time, only 10 different SVMs are trained, where the SVM with highest score is chosen.
Then, the SVM is trained to classify images between 2 of the 10 classes above. In this algorithm, 45 different SVMs are tried. During the testing, images are sent to all SVMs and the predict label is the one with highest score.

Experiments:
Table 1 The performance of kernels and algorithms

Kernels and algorithms	One vs. All	One vs. One
Linear	86.9%	89.53%
Polynomial (0.05)	87.03%	89.55%
RBF	93.39%	93.36%

Discussion:
From the table, we can find the RBF kernel is the best regarding to the accuracy. Different params also have an impact on the performance. Generally speaking, the One vs. One is better than One vs. All because the better one trains more SVMs than the other while the size of each training set is actually smaller. So more smaller training sets seem better than less bigger sets.

Auxiliary functions:
The loadMNISTImages function reads the images and reformat them to a vector. It checks the correctness of the data by magic number and normalize images to (0, 1). The output is a matrix with all image vectors.
The loadMNISTLabels function read labels and convert them to a vector. It also checks the data with magic number.

Requirement: Build a Convolutional Neural Network. Train it on MNIST training set and test it on testing set. You can design your architecture or use the architecture introduced in LeCun's paper [1].

LightNet is chose as the toolbox. Learning method is SGD. Learning rate is selected by the LightNet. Below is the structure of designed LeNet.

layers	Filter dimension	Image size after filtering
Conv	5x5	28x28
ReLu		28x28
Pool		14x14
Conv	5x5	10x10
ReLu		10x10
Pool		5x5
Conv	5x5	1x1
ReLu		1x1
Conv	1x1	1x1
Softmax loss		1x1

Auxiliary functions:
The getMINSTData, PrepareData_MNIST, Netlnit_MNIST function read the data from MNIST dataset and adapt the data to the input of LightNet.

Experiments:

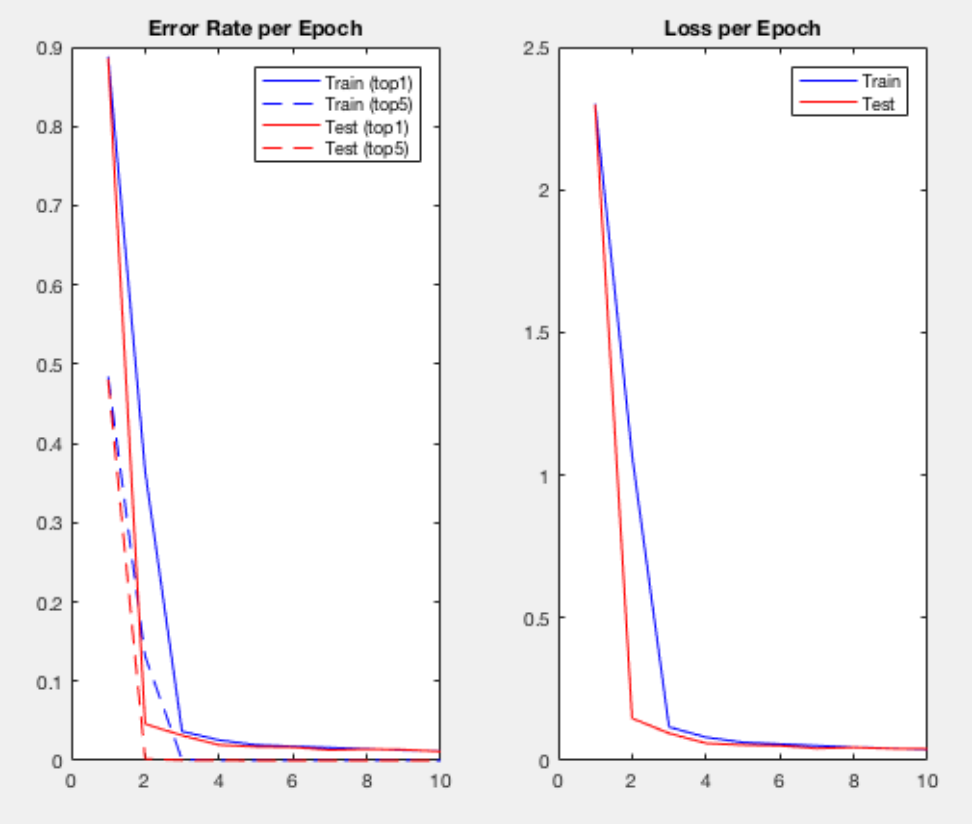

Figure 1 Deep learning on hand written digits

Discussion:

1. **From the training data, we can find the most error is 0.011867 and the top 5 error is 0.00011667. From the testing data, the top 1 error is 0.0114 and the top 5 error is 0.**

Part II: Transfer Learning

- Task 1

Requirement: You will be applying a deep learning technique commonly referred to as Transfer Learning. The first task will be to train a simple convolutional neural network using these images (a very simple 3-5 convolutional network will suffice) and test the accuracy of the model using the validation set. Because of the low number of training samples, you will see that the test accuracy of the model will be lower than expected.

Method: A simple convolutional neural network is trained and tested using the training images.

Experiments: Shown below.

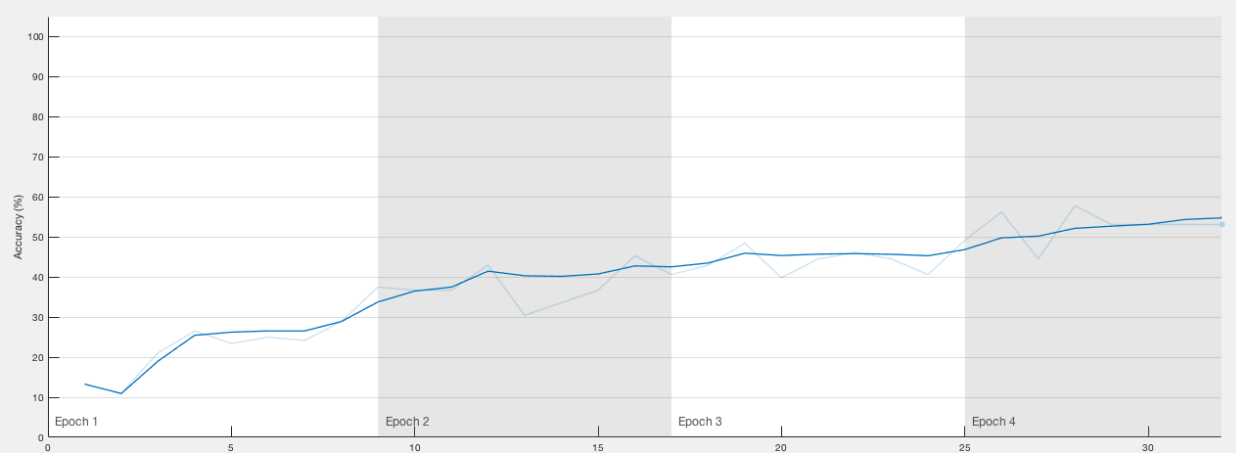

Figure 2 Training result of the simple CNN

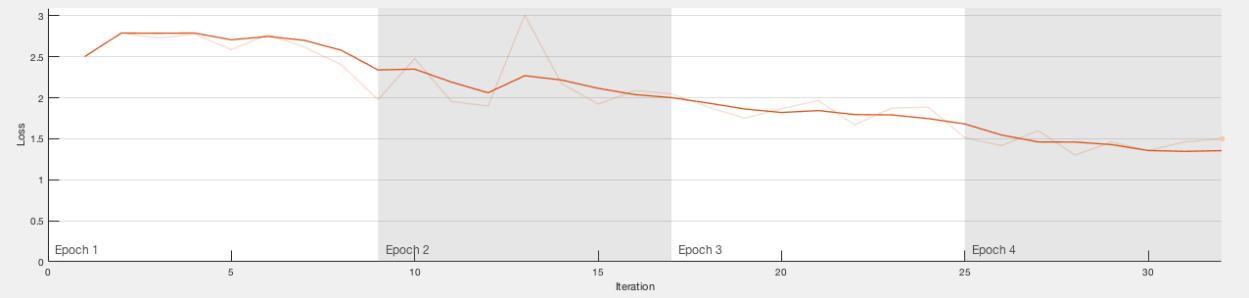

Figure 3 Testing result of the simple CNN

Discussion:
The accuracy on the test data is 43.98%, which is lower than expected. It is because of the low number of training samples.

- Task 2

Requirement: The next task will be to download a pretrained model for image classification (for example VGG, ResNet, InceptionNet) and use it as a feature extractor. To do this, you will remove the last fully connected layers of the pretrained model and replace it with untrained fully connected layers to classify the monkey species. During training, you will freeze the convolutional layer parameters so that they remain the same and only update the fully connected layers at the end. In this way, the convolutional layers act as generalized feature extractors that have already been pretrained on millions of other images (that were not necessarily all monkeys) while the fully connected layers are able to take these features and classify our images. You should see a substantial boost in accuracy even though we have the same amount of training samples. To further boost the performance of the network, you can unfreeze the convolutional layers and train for a few more epochs with a small step size to fine tune the network to extract even more predictive power.

Method: The features extractor is a pretrained AlexNet. Only the parameters in the last fully connected layer is changed. All previous layers' params are fixed.

Experiments: Shown below

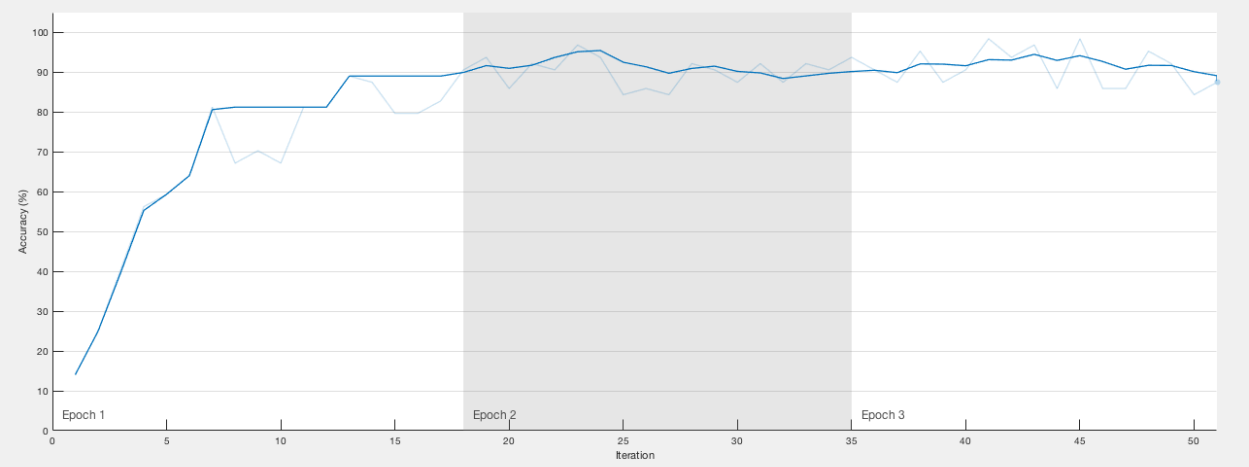

Figure 4 Training accuracy of transfer learning

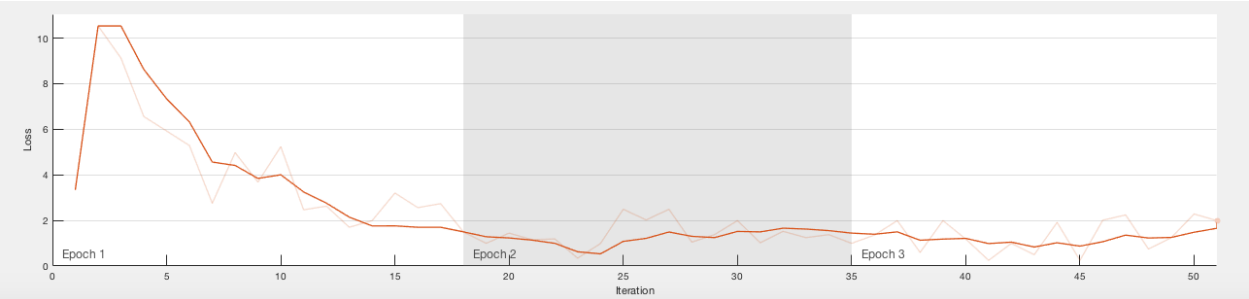

Figure 5 Test accuracy of transfer learning

Discussion:
With transfer learning, the accuracy increased from 43.98% to 90.32%. When the data size is same, transfer learning still gives a better result because the feature extractor (The pretrained Convolutional NN).

- Task 3

Method: Then the params in convolutional layers are unfreeze and tuned with a new small learning rate 0.0001.

Experiment:

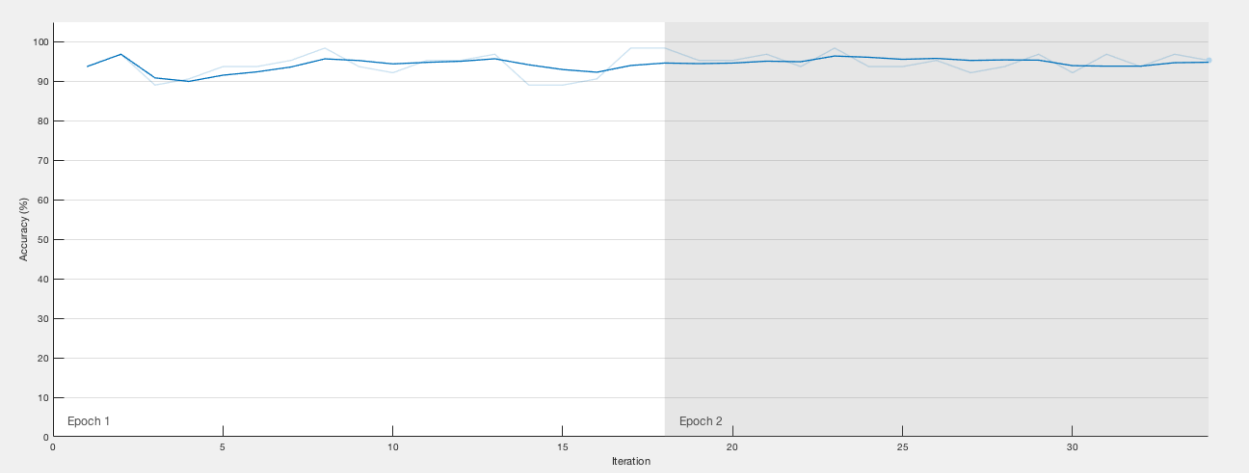

Figure 6 Training accuracy of fine-tuned network

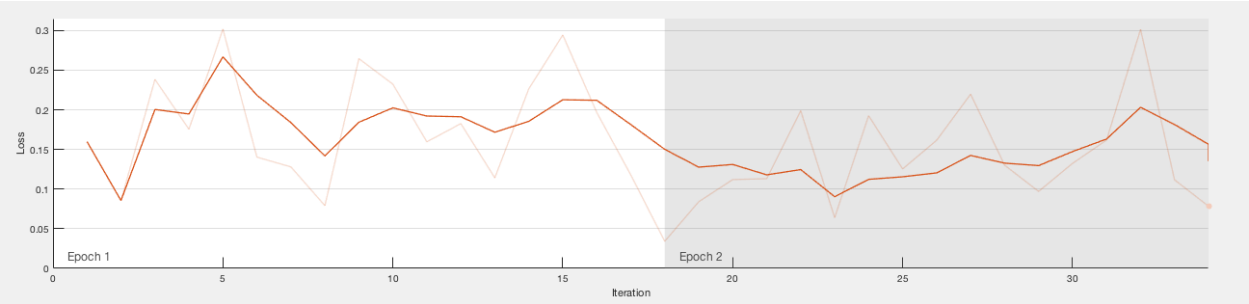

Figure 7 Testing accuracy of fine-tuned network

Discussion:
Testing accuracy increases from 90.32% to 93.01%. Fine-tune parameters helps the network be a little bit better.

Other hyper-parameters in three tasks above: Training method: SGD; batch size is 64; learning rate is 0.01

Auxiliary functions: The loadTrainingImages function reads and resizes the image regarding to the AlexNet. The loadTestingImages function does the same thing.