MNIST

Convolutional Neural Network

Author:
Vayne Lover

Supervisor:
Udacity

August 18, 2016

1 Introduction

1.1 Tools

- Caffe
- Visio
- LaTeX
- Python
- Matlab
- Adobe Illustrator CC

1.2 Background

It's known to us that the demand of people can greatly promoted the development of technology so nowadays deeping learning is becoming more and more popular.We may always hear Convolutional Neural Network,the most popular famous neural work,which has excellent perforcement for large image processing.

1.3 Target

In this project,my target is to use CNN algorithm in image recognition.I developed a simple CNN model and use MNIST dataset to train the model,you can see a simple related cnn below.Then i test the model using accuracy to judge if this model can work well with the data.
In Section 2 i describe the MNIST dataset.in Section 3 i developed a simple CNN model and show the results of test.In Section 4 i explain the key of the model and draw a conclusion.

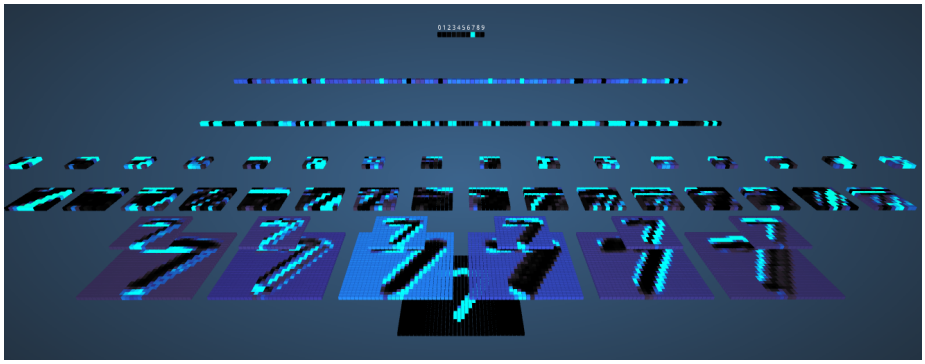

Figure 1: Visual CNN

2 Data Engineering

2.1 MNIST

The MNIST database of handwritten digits, in this project, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

2.2 Getting Data

You can download the data in http://yann.lecun.com/exdb/mnist/.

2.3 Constitution

This data set consist of four files.They are train-images-idx3-ubyte,train-labels-idx1-ubyte,t10k-images-idx3-ubyte and t10k-labels-idx1-ubyte.You can see the attributes below.

File	Description
train-images-idx3-ubyte	training set,pictures
train-labels-idx1-ubyte	training set,labels
t10k-images-idx3-ubyte	testing set,pictures
t10k-labels-idx1-ubyte	testing set,labels

Table 1: MNIST Origin Data Files

train-labels-idx1-ubyte			
Offset	Data Type	Value	Description
0000	32 bit integer	2049	magic number
0004	32 bit integer	60000	number of items
0008	unsigned byte	Unknown	label
0009	unsigned byte	Unknown	label
...	...	...	...

Table 2: TRAINING SET LABEL FILE

train-images-idx3-ubyte			
Offset	Data Type	Value	Description
0000	32 bit integer	2051	magic number
0004	32 bit integer	60000	number of images
0008	32 bit integer	28	number of rows
0012	32 bit integer	28	number of columns
0016	unsigned byte	Unknown	pixel
0017	unsigned byte	Unknown	pixel
...	...	...	...

Table 3: TRAINING SET IMAGE FILE

t10k-labels-idx1-ubyte			
Offset	Data Type	Value	Description
0000	32 bit integer	2049	magic number
0004	32 bit integer	10000	number of items
0008	unsigned byte	Unknown	label
0009	unsigned byte	Unknown	label
...	...	...	...

Table 4: TEST SET LABEL FILE

t10k-images-idx3-ubyte			
Offset	Data Type	Value	Description
0000	32 bit integer	2051	magic number
0004	32 bit integer	10000	number of images
0008	32 bit integer	28	number of rows
0012	32 bit integer	28	number of columns
0016	unsigned byte	Unknown	pixel
0017	unsigned byte	Unknown	pixel
...	...	...	...

Table 5: TEST SET IMAGE FILE

2.4 Transform

We know that Caffe just work with LEVELDB and LMDB,so we need to change these data into LMDB form.You can follow the procedure below.

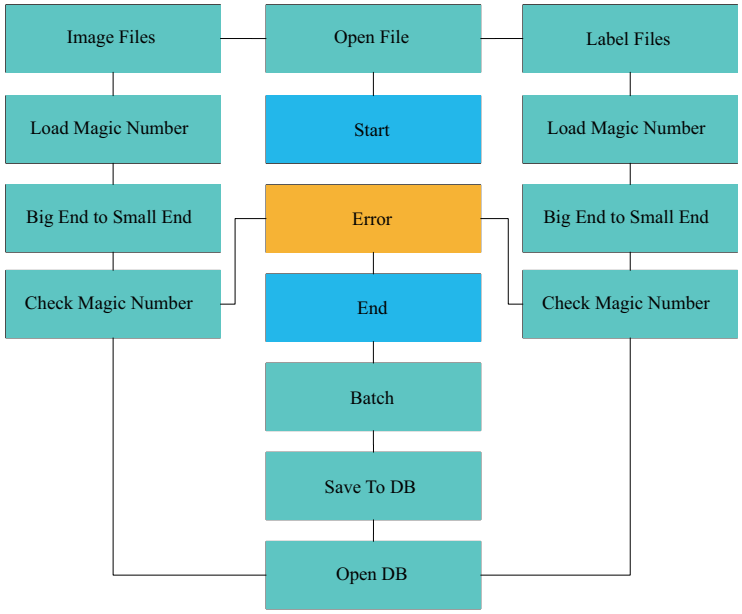

Figure 2: Transform

3 Modeling

3.1 Description

In this part,i will structure the famous LeNet-5 model.LeNet-5 is a convolutional network designed for handwritten and machine-printed character recognition.

3.2 Analysis

LeNet-5 has seven layers.However,this time i use caffe so i need to add other layers.In caffe it includes eleven layers:two data layers,two convolution layers,two pooling layers,two inner product layers,a ReLU layer,an accuracy layer and a loss layer.

LeNet-5 Model

Here are the LeNet-5 Model.

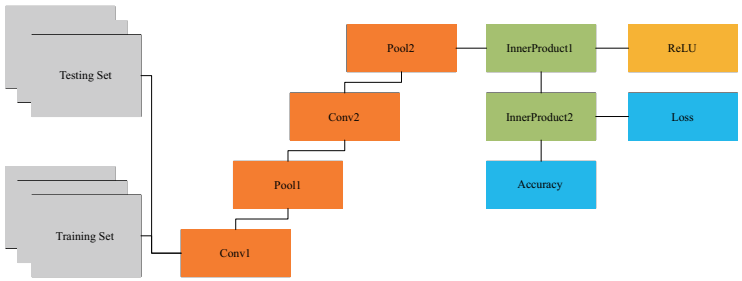

Figure 3: LeNet-5 Model

LeNet-5 Parameters

Here are the LeNet-5 Parameters.

Layer	Type	Output	Kernel	LR_mult	Bias_mult
conv1	convolution	20	5×5	1	2
pool1	pooling	NaN	2×2	NaN	NaN
conv2	convolution	50	5×5	1	2
pool2	pooling	NaN	2×2	NaN	NaN
ip1	inner product	500	NaN	1	2
ip2	inner product	10	NaN	1	2

Table 6: LeNet-5 Parameters

3.3 Training

In this part,i will train the hyper parameter of model.In order to get a good result,i choose iterations as 10,000 times with the hope of finding the global optimal value.Besides,i choose the base learning rate as 0.01,momentum as 0.9,weight decay as 5×10^{-4}.What's more,i set gamma as 10^{-4},power as 0.75.The procedure of training can be seen below.

3.4 Prediction

After long time training,we can get a model,then i will use this model to predict the test set.We can see that the accuracy surprising arrive at 99%.So i think

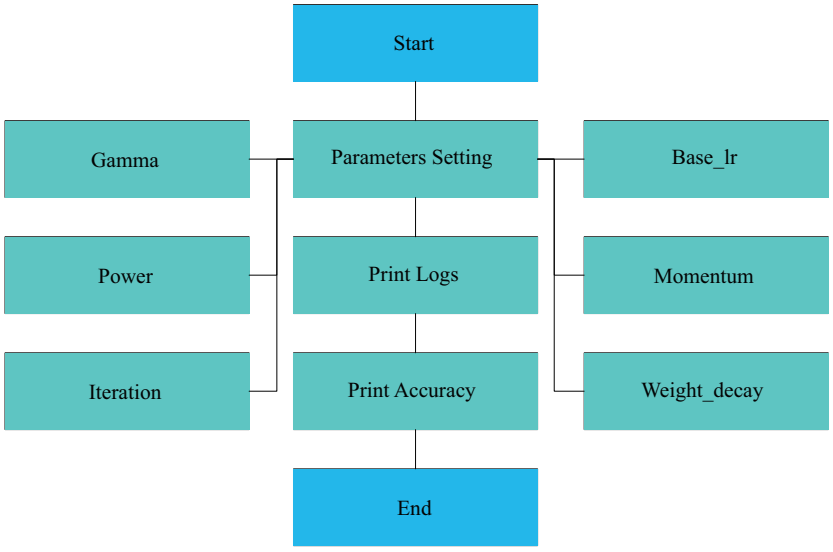

Figure 4: Training

this model can solve the hand-written image recognization problem well.The accuracy can be seen below.

```
I0818 17:50:19.965402 3332875200 caffe.cpp:308] Batch 79, accuracy = 1
I0818 17:50:19.965450 3332875200 caffe.cpp:308] Batch 79, loss = 0.00261825
I0818 17:50:19.999764 3332875200 caffe.cpp:308] Batch 80, accuracy = 0.99
I0818 17:50:19.999831 3332875200 caffe.cpp:308] Batch 80, loss = 0.0228167
I0818 17:50:20.049513 3332875200 caffe.cpp:308] Batch 81, accuracy = 1
I0818 17:50:20.049579 3332875200 caffe.cpp:308] Batch 81, loss = 0.00425161
I0818 17:50:20.092494 3332875200 caffe.cpp:308] Batch 82, accuracy = 1
I0818 17:50:20.092625 3332875200 caffe.cpp:308] Batch 82, loss = 0.000854463
I0818 17:50:20.135527 3332875200 caffe.cpp:308] Batch 83, accuracy = 1
I0818 17:50:20.135570 3332875200 caffe.cpp:308] Batch 83, loss = 0.0194295
I0818 17:50:20.182132 3332875200 caffe.cpp:308] Batch 84, accuracy = 0.99
I0818 17:50:20.182173 3332875200 caffe.cpp:308] Batch 84, loss = 0.0206927
I0818 17:50:20.230895 3332875200 caffe.cpp:308] Batch 85, accuracy = 1
I0818 17:50:20.230940 3332875200 caffe.cpp:308] Batch 85, loss = 0.00890107
I0818 17:50:20.269271 3332875200 caffe.cpp:308] Batch 86, accuracy = 1
I0818 17:50:20.269312 3332875200 caffe.cpp:308] Batch 86, loss = 0.000101268
I0818 17:50:20.316546 3332875200 caffe.cpp:308] Batch 87, accuracy = 1
I0818 17:50:20.316586 3332875200 caffe.cpp:308] Batch 87, loss = 0.000149978
I0818 17:50:20.361763 3332875200 caffe.cpp:308] Batch 88, accuracy = 1
I0818 17:50:20.361918 3332875200 caffe.cpp:308] Batch 88, loss = 3.29222e-05
I0818 17:50:20.406551 3332875200 caffe.cpp:308] Batch 89, accuracy = 1
I0818 17:50:20.406590 3332875200 caffe.cpp:308] Batch 89, loss = 3.77777e-05
I0818 17:50:20.452600 3332875200 caffe.cpp:308] Batch 90, accuracy = 0.96
I0818 17:50:20.452641 3332875200 caffe.cpp:308] Batch 90, loss = 0.0884745
I0818 17:50:20.486058 3332875200 caffe.cpp:308] Batch 91, accuracy = 1
I0818 17:50:20.486100 3332875200 caffe.cpp:308] Batch 91, loss = 3.99872e-05
I0818 17:50:20.530663 3332875200 caffe.cpp:308] Batch 92, accuracy = 1
I0818 17:50:20.530711 3332875200 caffe.cpp:308] Batch 92, loss = 0.000441549
I0818 17:50:20.565843 3332875200 caffe.cpp:308] Batch 93, accuracy = 1
I0818 17:50:20.565917 3332875200 caffe.cpp:308] Batch 93, loss = 9.33e-05
I0818 17:50:20.611857 3332875200 caffe.cpp:308] Batch 94, accuracy = 1
I0818 17:50:20.611896 3332875200 caffe.cpp:308] Batch 94, loss = 0.000334327
I0818 17:50:20.660082 3332875200 caffe.cpp:308] Batch 95, accuracy = 0.99
I0818 17:50:20.660141 3332875200 caffe.cpp:308] Batch 95, loss = 0.0177553
I0818 17:50:20.696465 3332875200 caffe.cpp:308] Batch 96, accuracy = 0.96
I0818 17:50:20.696527 3332875200 caffe.cpp:308] Batch 96, loss = 0.0732619
I0818 17:50:20.739668 3332875200 caffe.cpp:308] Batch 97, accuracy = 0.98
I0818 17:50:20.739706 3332875200 caffe.cpp:308] Batch 97, loss = 0.0838478
I0818 17:50:20.787011 3332875200 caffe.cpp:308] Batch 98, accuracy = 1
I0818 17:50:20.787051 3332875200 caffe.cpp:308] Batch 98, loss = 0.00312422
I0818 17:50:20.823093 3332875200 caffe.cpp:308] Batch 99, accuracy = 1
I0818 17:50:20.823192 3332875200 caffe.cpp:308] Batch 99, loss = 0.00318926
I0818 17:50:20.823226 3332875200 caffe.cpp:313] Loss: 0.0291732
I0818 17:50:20.823274 3332875200 caffe.cpp:325] accuracy = 0.99
I0818 17:50:20.823307 3332875200 caffe.cpp:325] loss = 0.0291732 (* 1 = 0.0291732 loss)
Vayne-Lovers
```

Figure 5: Accuracy

4 Conclusion

In this part i will tell what i learned and what i thought based on CNN.As we can learn from last section,CNN does a really good job in image recognization.But why it can get good results?In order to find the answer i read a lot of related papers and blogs.And finally i find these key points of CNN.

4.1 Local Perception

Convolutional Neural Network has a good way to decrease the number of features called local perception.We may think that we see a picture based on each pixels of it,however,our neural cells don't work like our thoughts.In face,each cell just take a part of a thing then the high layer neural cells combine the information together.
For example,if there is a 100×100 image,which can be presented a 10^4 vector.If hidden layer equals to input layer,the parameters will be 10^8.if each cell just need to collect 10×10 pixels of the image,the parameters will be 10^6.And this operation equals to convolution.

4.2 Weight Sharing

We may think that although we decrease 100 times,it still too large.So it comes to the second way,weight sharing.
For example,if we have 10^4 cells,and if each cell have 100 weights,it will be 100 weights total.It may hard to understand,you can think this way:We learn something in this part,but this thing we learn can be used in another part.So we can use the same features in all parts of the image.

4.3 Multi-Kernel

If we just use a 10×10 kernel,we can't get full features of the image.In order to solve this problem,we can add other kernels.For example,if we add another 15 kernels,we have 16 kernels,and we can use these kernels to learn 16 features of the image.

4.4 Down-Pooling

When we get all features we can start to classify using softmax classifier.However,we may face the problem that the features are still too large.For example,if we have a 32×32 image,and learned 100 features using 4×4 kernels,each feature will get $(32-4+1) \times (32-4+1)$ convolutional features,and will total get 841×100 convolutional features vector.It's hard to train the classifier.

Luckily,images have a good attribute that image's feature can be same in a part range. Therefore we can collect part of image together to decrease the features.And this operation is called pooling.

5 Reference

1. http://yann.lecun.com/exdb/lenet/
2. http://yann.lecun.com/exdb/mnist/
3. http://scs.ryerson.ca/ aharley/vis/conv/
4. http://www.cs.nyu.edu/ roweis/data.html
5. http://deeplearning.stanford.edu/wiki/index.php/Pooling
6. http://blog.csdn.net/qiaofangjie/article/details/16826849
7. http://caffe.berkeleyvision.org/gathered/examples/mnist.html
8. http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution