CONVOLUTIONAL NEURAL NETWORK

卷积神经网络

吴嘉权
201625010224
*2016*级计算机科学与技术2班

历程

	两周前换题目
90%	阅读资料
8%	*Coding*
2%	*Debugging*

假设空间 HYPOTHESIS SPACE

策略 STRATEGY

算法 ALGORITHM

模型三要素

卷积神经网络(CNN)厉害在哪里?

相比一般的神经网络

特征工程

FEATURE ENGINEERING

CNN三法宝

Local receptive fields
Shared weights and biases
Pooling layers

c.f. http://neuralnetworksanddeeplearning.com/chap6.html

我的模型

~~残疾版CNN~~

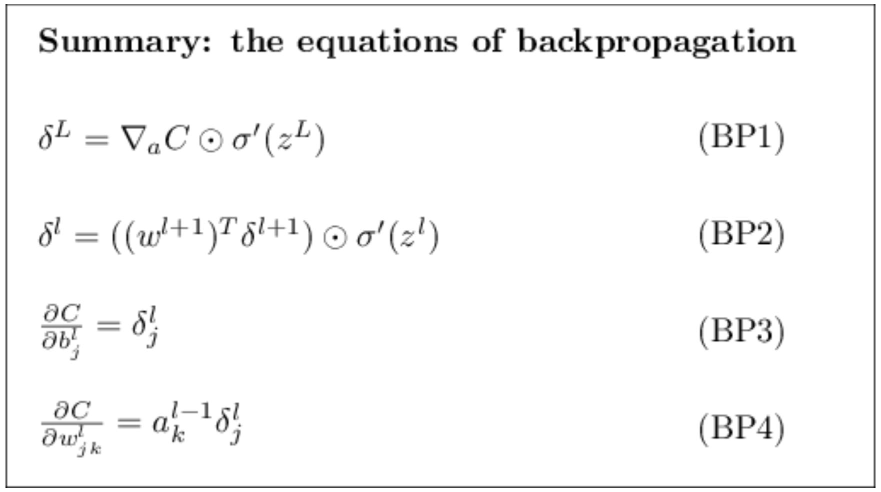
~~真·人工智障~~

三个一个不能少

一层*convolution*
一层*max pooling*
一层*fully connected*

怎么样 BACK PROPAGATION?

一般的*BP*



Summary: the equations of backpropagation

$$\delta^L = \nabla_a C \odot \sigma'(z^L) \quad \text{(BP1)}$$

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \quad \text{(BP2)}$$

$$\frac{\partial C}{\partial b^l_j} = \delta^l_j \quad \text{(BP3)}$$

$$\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j \quad \text{(BP4)}$$

CONVOLUTION, MAX POOLING 怎么BP?

其实很简单~~（并不）~~
运用基本微积分的知识
一步步推导
然后合并，发现规律

强烈建议搞懂BP

~~虽然框架满天飞~~
~~我本来也以为我自己懂的~~

为什么选这个*activation*?
不同的*activation*的区别和影响?
什么时候训练结束?
Gradient vanishing?
……

怎么知道我写得对不对?

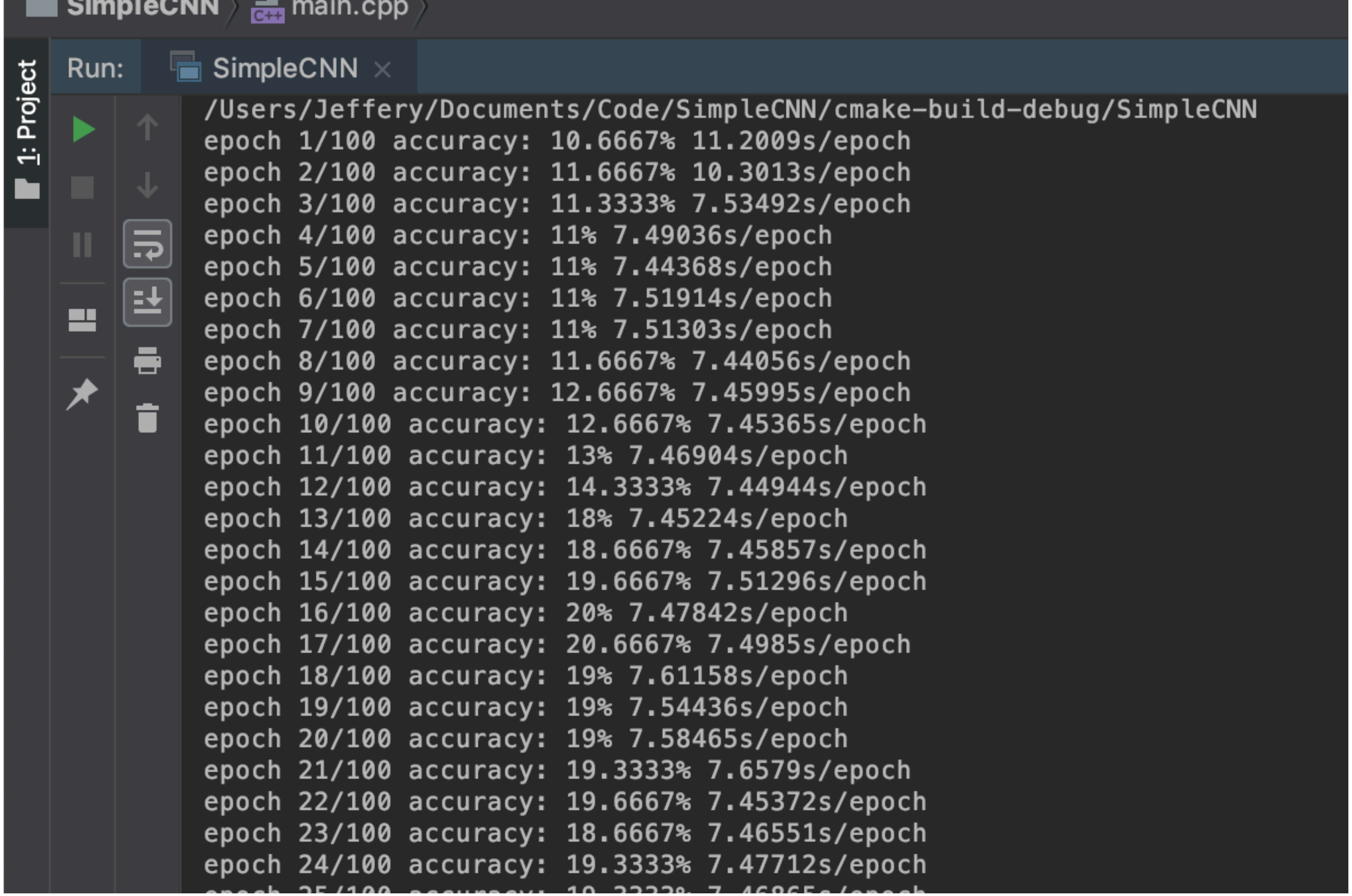
跑一个“HELLO WORLD”

MNIST

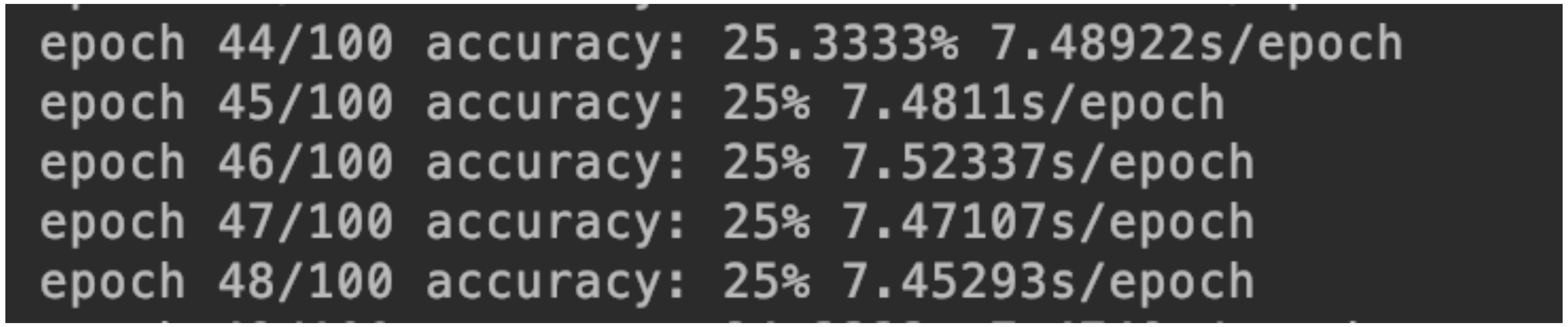
-数据来自KAGGLE的DIGIT RECOGNIZER　　(格式比官方的文件容易读)

首先定义一个基准BASELINE

10%



```
/Users/Jeffery/Documents/Code/SimpleCNN/cmake-build-debug/SimpleCNN
epoch 1/100 accuracy: 10.6667% 11.2009s/epoch
epoch 2/100 accuracy: 11.6667% 10.3013s/epoch
epoch 3/100 accuracy: 11.3333% 7.53492s/epoch
epoch 4/100 accuracy: 11% 7.49036s/epoch
epoch 5/100 accuracy: 11% 7.44368s/epoch
epoch 6/100 accuracy: 11% 7.51914s/epoch
epoch 7/100 accuracy: 11% 7.51303s/epoch
epoch 8/100 accuracy: 11.6667% 7.44056s/epoch
epoch 9/100 accuracy: 12.6667% 7.45995s/epoch
epoch 10/100 accuracy: 12.6667% 7.45365s/epoch
epoch 11/100 accuracy: 13% 7.46904s/epoch
epoch 12/100 accuracy: 14.3333% 7.44944s/epoch
epoch 13/100 accuracy: 18% 7.45224s/epoch
epoch 14/100 accuracy: 18.6667% 7.45857s/epoch
epoch 15/100 accuracy: 19.6667% 7.51296s/epoch
epoch 16/100 accuracy: 20% 7.47842s/epoch
epoch 17/100 accuracy: 20.6667% 7.4985s/epoch
epoch 18/100 accuracy: 19% 7.61158s/epoch
epoch 19/100 accuracy: 19% 7.54436s/epoch
epoch 20/100 accuracy: 19% 7.58465s/epoch
epoch 21/100 accuracy: 19.3333% 7.6579s/epoch
epoch 22/100 accuracy: 19.6667% 7.45372s/epoch
epoch 23/100 accuracy: 18.6667% 7.46551s/epoch
epoch 24/100 accuracy: 19.3333% 7.47712s/epoch
epoch 25/100 accuracy: 19.3333% 7.46865s/epoch
```



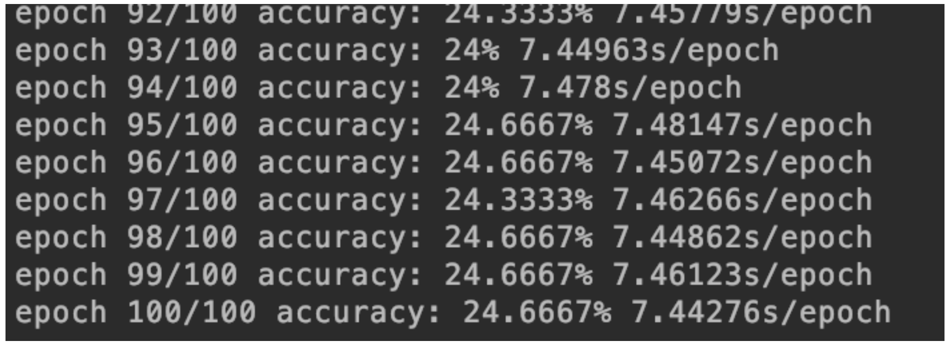
```
epoch 44/100 accuracy: 25.3333% 7.48922s/epoch
epoch 45/100 accuracy: 25% 7.4811s/epoch
epoch 46/100 accuracy: 25% 7.52337s/epoch
epoch 47/100 accuracy: 25% 7.47107s/epoch
epoch 48/100 accuracy: 25% 7.45293s/epoch
```

YES!

跑赢了基准

但其实很差很差很差很差…….

而且不幸地卡在了LOCAL MINIMUM （25%）

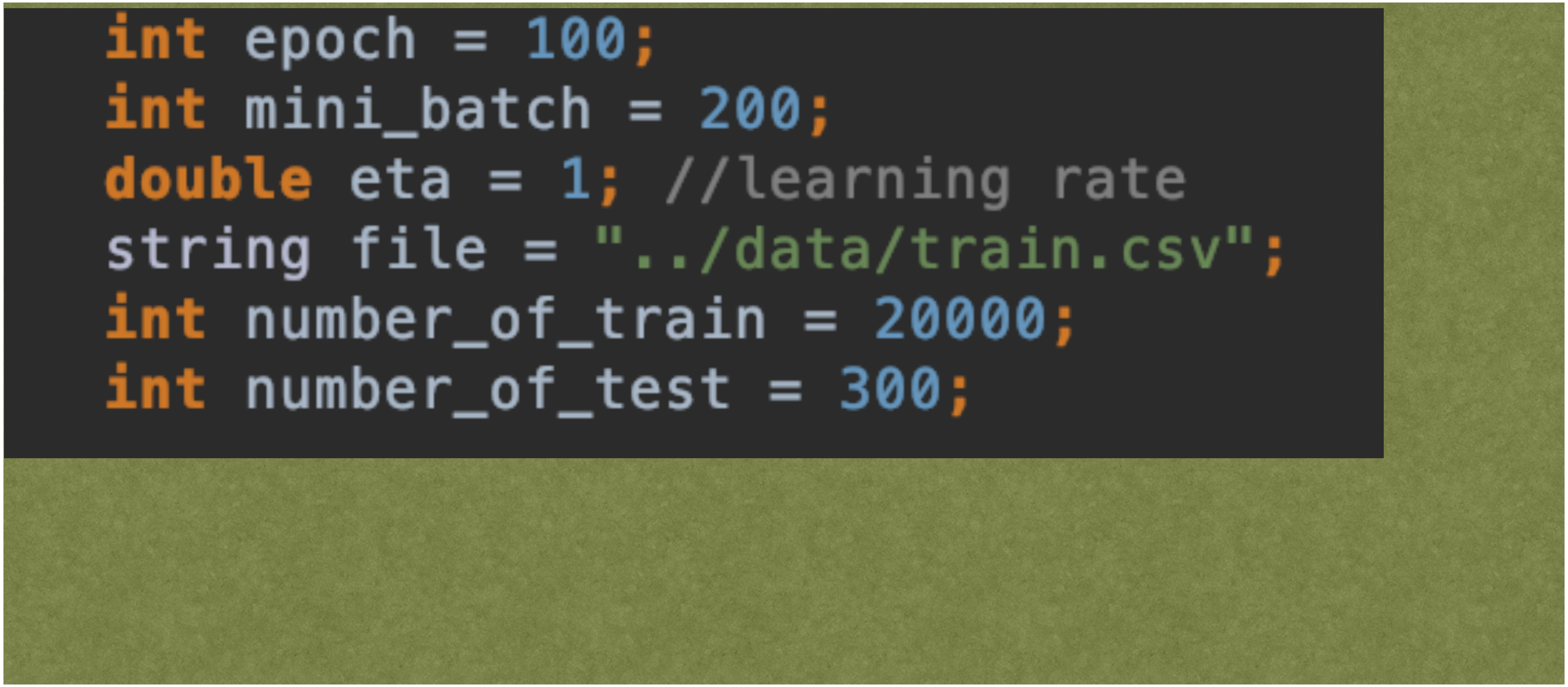


```
epoch 92/100 accuracy: 24.3333% 7.45779s/epoch
epoch 93/100 accuracy: 24% 7.44963s/epoch
epoch 94/100 accuracy: 24% 7.478s/epoch
epoch 95/100 accuracy: 24.6667% 7.48147s/epoch
epoch 96/100 accuracy: 24.6667% 7.45072s/epoch
epoch 97/100 accuracy: 24.3333% 7.46266s/epoch
epoch 98/100 accuracy: 24.6667% 7.44862s/epoch
epoch 99/100 accuracy: 24.6667% 7.46123s/epoch
epoch 100/100 accuracy: 24.6667% 7.44276s/epoch
```

事后诸葛亮环节

最基础的DNN有90%
LE-NET5(CNN) 有99%

模型结构不行
（三要素之假设空间）



```
int epoch = 100;
int mini_batch = 200;
double eta = 1; //learning rate
string file = "../data/train.csv";
int number_of_train = 20000;
int number_of_test = 300;
```

超参数不行

（三要素之算法）

LEARNING RATE?

MINI BATCH? SGD?

—论如何成为一名调参侠

误差函数不行
(三要素之策略)

CROSS ENTROPY?

L1 DISTANCE?

L2 DISTANCE?

…

~~我不行~~

流下了属于真正弱鸡的眼泪

实现方式（粗暴低效）

运气（LOCAL MINIMUM）

...

Reference

🤩Neural Networks and Deep Learning,　Michael Nielsen

🤩Machine Learning, 2017, Fall　李宏毅,国立台湾大学

🤩Deep learning with Python，François Chollet

🤩CS231N,　Serena Yeung, Fei-Fei Li, Stanford

🤩三个月教你从零入门深度学习，　Charlotte77

🤩Forward And Backpropagation in Convolutional Neural Network. , Sujit Rai

🤩统计学习方法，李航

Understanding Convolution in Deep Learning,　Tim Dettmers

Eigen Document(矩阵运算库)

. . .

谢谢