

深度学习大作业

**Cifar10图像分类**



* 实验科目：深度学习
* 老师：金长龙
* 实验者:刘君怡201500800513(软2)
* 实验者:吴晓彤201500800556数媒
* 实验者:夏雨柔201500800560(软1)
* 完成时间：2018年1月15日

# CIFAR-10 介绍

## Cifar背景介绍

Cifar-10 是由 Hinton 的两个大弟子 Alex Krizhevsky、Ilya Sutskever 收集的一个用于普适物体识别的数据集。Cifar 是加拿大政府牵头投资的一个先进科学项目研究所。Hinton、Bengio和他的学生在2004年拿到了 Cifar 投资的少量资金，建立了神经计算和自适应感知项目。这个项目结集了不少计算机科学家、生物学家、电气工程师、神经科学家、物理学家、心理学家，加速推动了 Deep Learning  的进程。从这个阵容来看，DL 已经和 ML 系的数据挖掘分的很远了。Deep Learning 强调的是自适应感知和人工[智能](http://lib.csdn.net/base/aiplanning)，是计算机与神经科学交叉；Data Mining 强调的是高速、[大数据](http://lib.csdn.net/base/hadoop)、统计数学分析，是计算机和数学的交叉。

## Cifar数据集介绍

CIFAR10数据集共有5个train\_batch及一个test\_batch,

每个batch是10000\*3072大小，

也就是说每个bacth共含有10000张图片，每张照片存储为一行即3072大小，

这3072是三个通道总和，所以每个通道是3072/3=1024个像素，即1024=32\*32大小。

Cifar10总共10个输出，即10个分类

这10个分类是：飞机, 摩托车, 鸟, 猫, 鹿, 狗, 青蛙,马, 船, 卡车

Cifar图片展示

train\_batch\_1.mat 第88张图片-----马

# 2. 所用算法及网络

## 有监督的NN网络

原因：因为每一个图像都有labels标签，就想尝试一下有监督的NN网络

正确率：35.42%

### 训练方法：

建立一个五层神经网络，五层的神经元个数分别为3072，10000，500，100，10

，

输出函数为sigm

激活函数为tan\_opt

总共训练5次，每个小batch大小为1000

## 自我学习

### 自我学习所用算法

pca算法（主成份分析）

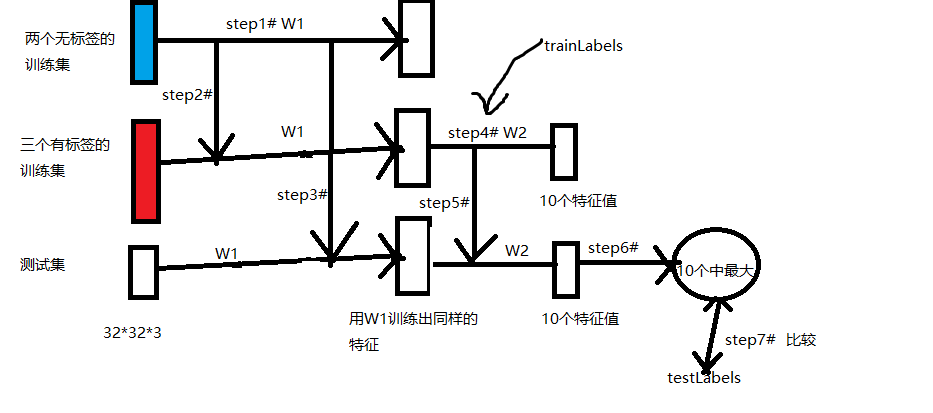
自编码器算法

Softmax分类算法

L-BFGS算法

### 自我学习训练方法

一层自编码器，一层softmax



Step1#:对两个无标签的训练集使用自编码器的方法无监督训练的得到最优W1

Step2#:使用W1\*三个有标签的训练集得到同样的特征

Step3#:同理使用W1\*测试集得到同样的特征

Step4#:对三个有标签训练集进行softmax分类得到10个特征，和最优W2

Step5#:使用最优W2\*测试集得到训练之后的测试集的10个特征值

Step6#:在10个特征值选取最大的即为预测的labels

Step7#:预测labels与testLabels做比对即可知道正确率

（所有训练集与测试集都pca到144维）

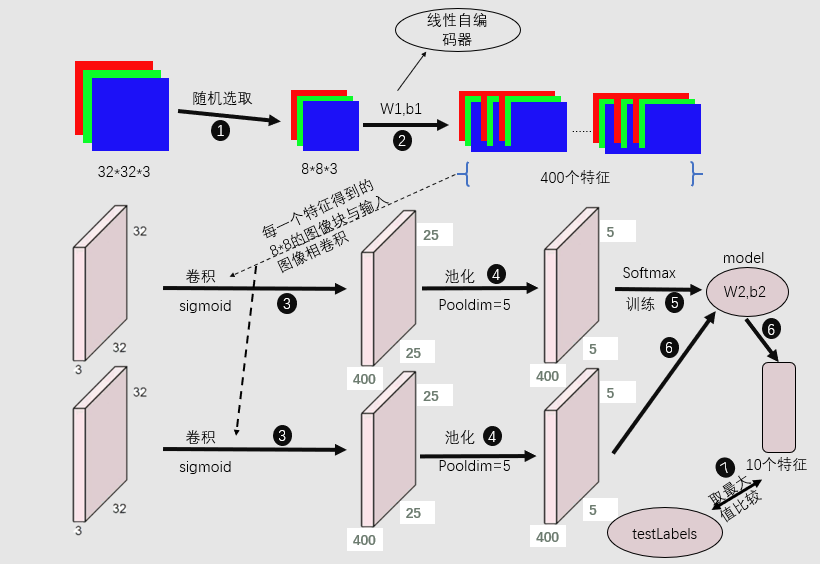
## 无监督的单层CNN网络

### CNN所用算法

1. 线性自编码算法
2. ZCA白化算法
3. 卷积图像算法
4. 池化（最大池，平均池）
5. Softmax分类算法
6. L-BFGS算法

### CNN训练方法

用一层自编码器训练的到的小patch去卷积图像和池化卷积之后的数据

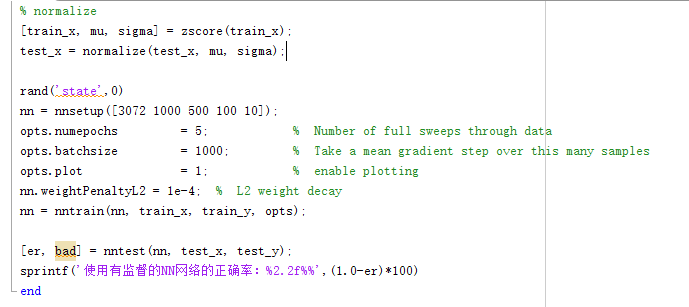


1. 从32\*32\*3的彩色图片（从所有样本集随机抽取60000个小patch）随机选取8\*8\*3的彩色小块
2. 使用线性自编码器提取所有小patch，得到400个特征，并得到可以得到同样的特征的最优W1，b1
3. 用每一个特征所得到8\*8的特征块去卷积原数据中的图像（32\*32\*3）得到卷积图像（25\*25）（该步是对测试集和训练集同时进行）
4. 为了减少数据量，对卷积后得到的数据进行池化，得到5\*5的池化后的特征结构（该步是对测试集和训练集同时进行）
5. 用softmax去训练训练集得到最优的参数W2，b2
6. 然后使用最优W2\*测试集得到训练之后的测试集的10个特征值
7. 在这10个特征值取最大值与testLabels比较

# 3. 实验及结果分析

## 有监督的NN网络实验

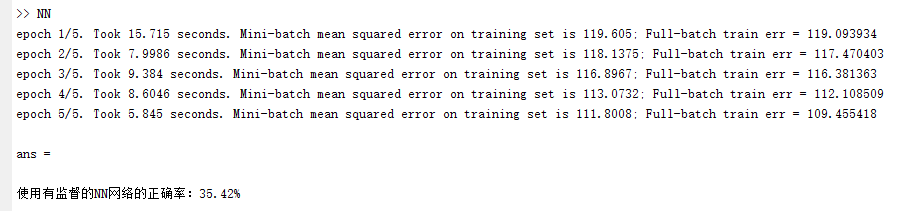
### 部分代码展示



### 代码流程图



### 结果展示（最好的一次）（35.42%）



### 坏结果原因（训练调整）

层数不够多（刚开始设置3072 1000 10）

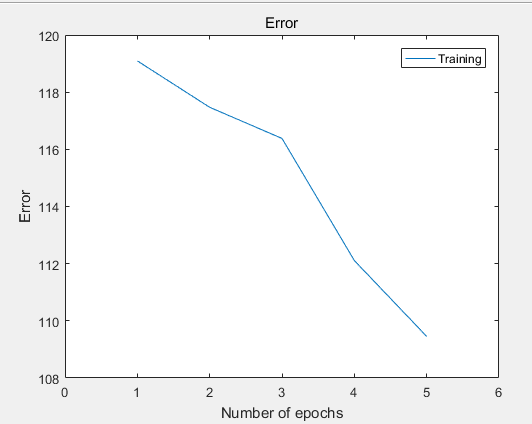
训练次数小于5或者大于5

输出函数为softmax时偏差过大

batchsize过大

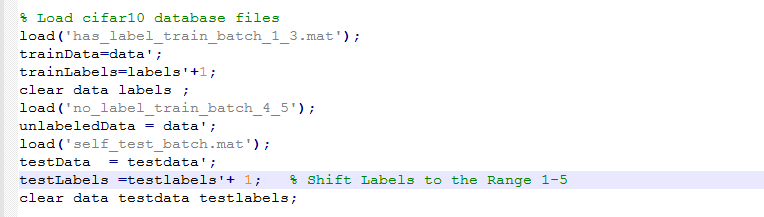
以上原因都只有小于10%的正确率，其中有些还会出现训练次数过大不收敛的情况

每个batch平均误差太大



## 有监督的自我学习

### 部分代码展示



1. 数据都进行pca
2. 因为要softmax，所以类别需要从1开始
3. 把train\_batch\_1\_2\_3当作有标签的训练集，留作softmax训练
4. 把train\_batch\_4\_5当作无标签的训练集，留作自编码训练

### 改进过程

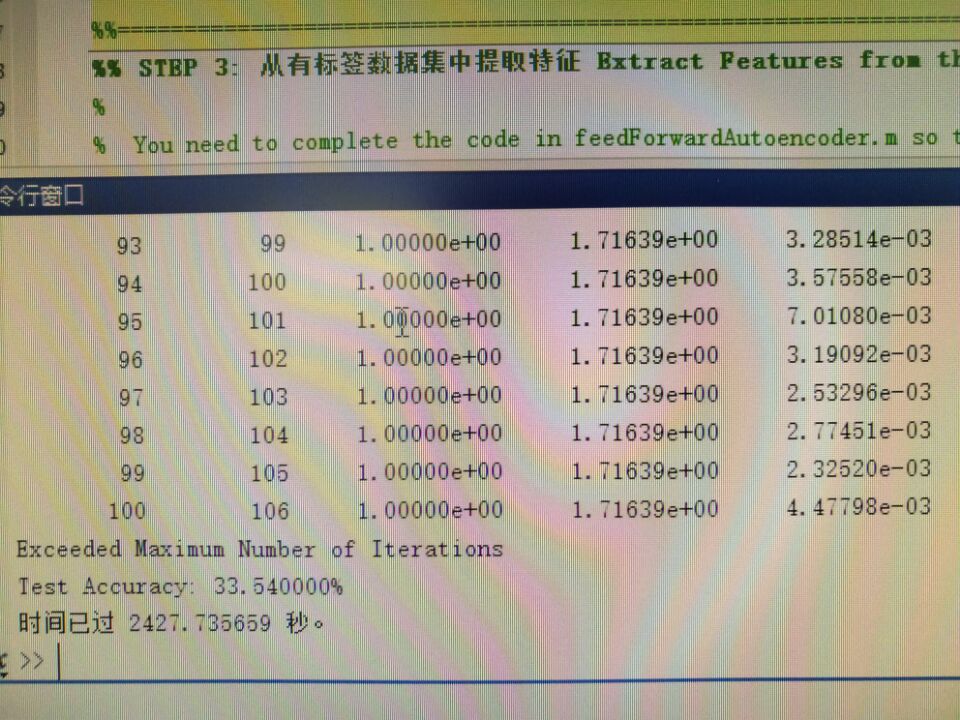
因为该开始的所有小组成员电脑运行自我学习代码都会死机，所以放弃了该方法，去进行卷积训练，后来去了机房就试试了，总共进行了两次实验

第一次有标签集，无标签集，测试集使用10000样本

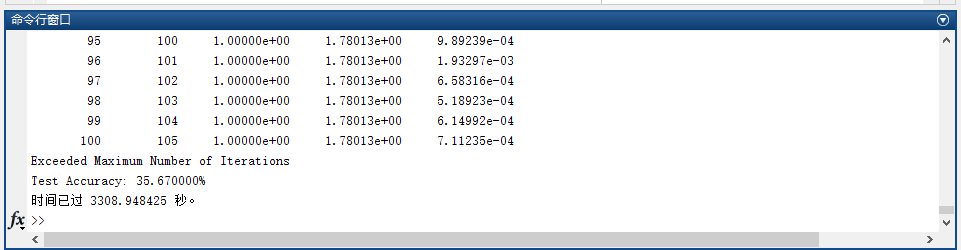
第二次有标签集，无标签集，测试集分别使用30000，20000，10000样本，但正确率远不如卷积神经网络，就每进行进一步训练

### 结果展示(35.67%)

**各使用10000个样本**



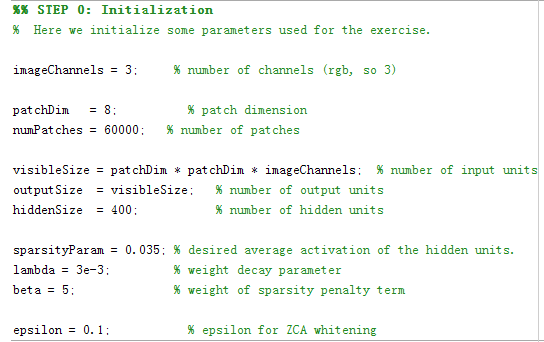
**有标签集，无标签集，测试集分别使用30000，20000，10000样本**

****

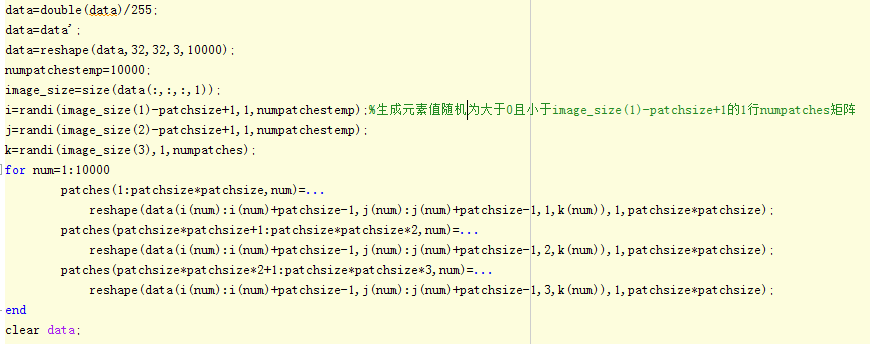
## 无监督的单层CNN实验(主要)

### 部分代码展示

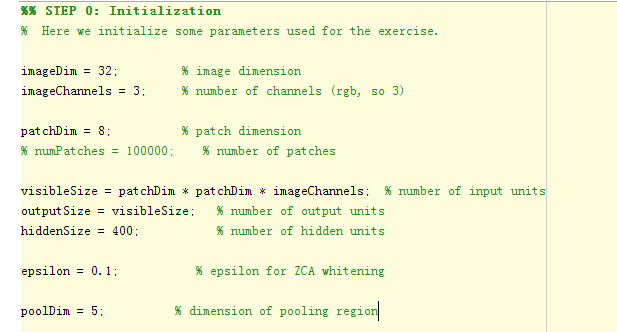
线性自编码参数设置



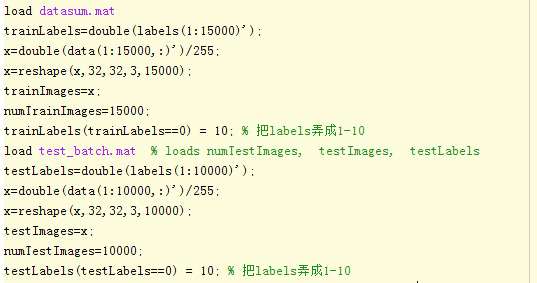
彩色图片随机选取图片



卷积网络参数设置



Cnn的加载数据



### 遇到的问题及解决

1.问题：可以直接用黑色图片训练吗？

解决：不能，不用彩色图片会失去很多特征

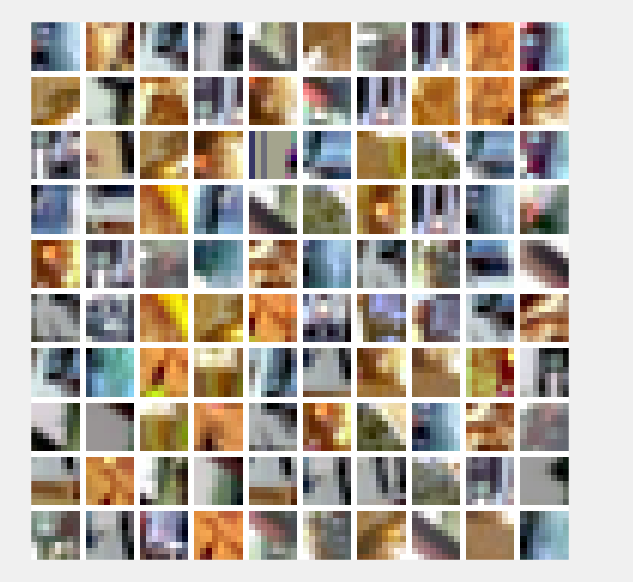
2.问题：随机选取patch时候的问题：数据集有10个分类，随机选取patch是从每个图片里面都随机选patch，还是所有图片随机选patch？

解决：是所有图片里面随机选取patch，因为训练的patch是所有图像共有的隐藏层属性，所有选取时图片也应该随机

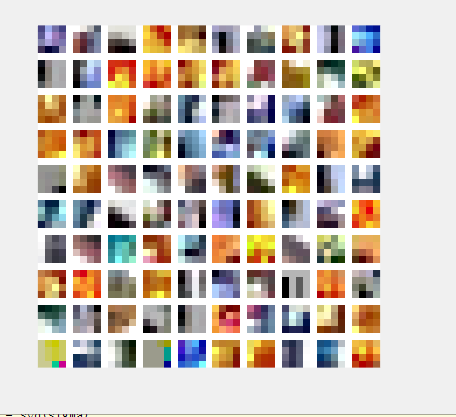
3.问题：patchsize 不知道定多大合适？

动手实验，最后选取patchsize为8\*8\*3

Patchsize=8\*8\*3

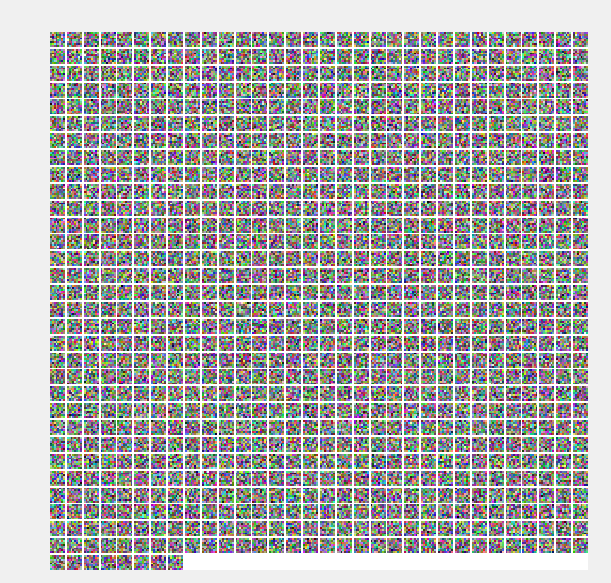


Patchsize=4\*4\*3



4.问题：使用relu构造自编码器？

解决：不行，训练结果如下（不知原因）



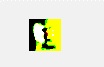
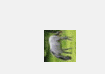
1. 问题：对于hiddensize的大小问题？

解决：使用控制变量法将hiddensize作为变量，我们试验hiddensize=200，400，500，实验结果表明200的正确率较于400，500的正确率偏低一点

1. 问题：机器内存不够

解决：经过仔细思考我们想了四个解决方法，我们先采取对输入数据的pca，将3072维的数据压缩到144维，让电脑尽可能地多跑一点数据；因为卷积过的特征矩阵特别大，所以我们设置了一个stepsize，stepsize=50，一次对50个特征进行卷积和池化，总共卷积和池化8次，来避免运行内存的不足；还有pooldim参数调节，pooldim过小了会使得池化后的数据量依旧很大，但是pooldim过大又会影响到我们所得到数据的真实性，在权衡下选择了pooldim为5；调节参数patchdim的大小，如果采用16\*16或者8\*8的特征来进行卷积会比用4\*4的特征卷积所占的内存更小，而且8\*8得到的特征具有边缘特征而且非常明显，故采用patchdim为8.

以下为pca前后



1. 问题：卷积函数用relu代替sigmoid

解决：使用relu之后虽然收敛较快，但正确率下降一半，故不采用

1. 问题；样本个数对训练结果有影响吗

解决：我们分别试验了1000样本，5000样本，10000样本，15000样本，正确率确实在逐步提升，但用机房电脑最大只能跑15000的样本，但我们相信50000的样本正确率会更高的

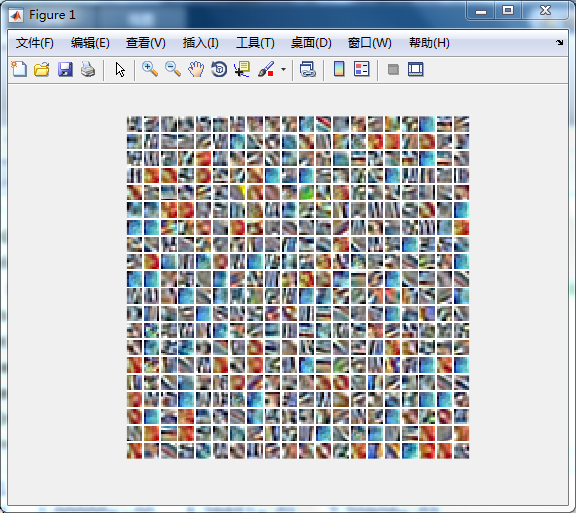
1. 问题：softmax迭代次数对正确率有影响

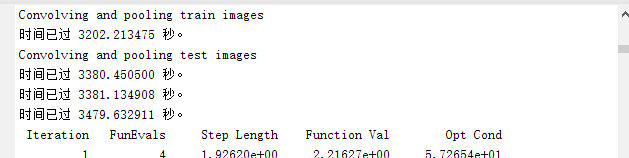
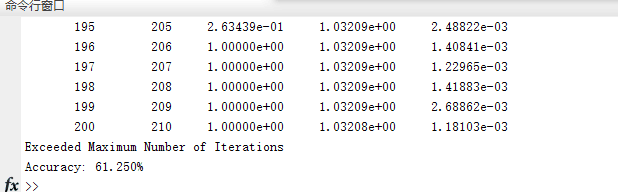
解决：测试过200，400次的迭代次数，发现softmax在200以内就都收敛了，所以无影响

1. 问题：一开始使用的训练集和测试集都是1000个，得到的正确率只有百分之41点多？

解决：使用更多的训练集，因为训练集的数量不够会导致softmax训练得到的模型的误差较大，只有通过更多数据的学习才能是模型的性能上去。

### 结果展示（最好的一次）（61.25%）





使用15000的训练样本，patchdim=8，pooldim=5

# 4. 总结及展望

总结与展望

经过短短几天紧张地完成一个小小的cifar10分类项目，我们对于课堂上讲的自动编码器、主成分分析与白化等等有了更加深刻的理解，讲课堂上学习的内容与实践结合起来，

而且之前接触的都是MINIST的28\*28的灰度图，但cifar10使用的是3通道的彩色图，所以需要对原来的代码进行相对应的修改。这也是我们第一次使用cnn进行实验，

之前只是在ppt上了解了卷积和池化的原理，而进行实验后明白了对于数据进行卷积和池化的意义何在，以及如何选择卷积核的大小与池的大小。

因为机器内存的限制，我们只使用了百分之30的数据进行训练和学习，然后正确率最高也只有60出头，所以我们也想过使用多层卷积来提高正确率，

关于两层卷积我们只是有个初步的想法，我们之前一直卡在怎么把第一次卷积和池化后的输出结果作为第二层卷积的输入呢？

因为输入是32\*32\*3但是通过特征块卷积后的输出是特征块数量（即隐层个数400）\*池的个数（5\*5），

首先我们明确了第二层应该用一个patchsize\*patchsize\*hiddensize的特征块来进行卷积，但是怎么通过线性编码器来得出这个相应的w和b呢，

我们觉得是对特征再提取特征，将现在的400（hiddensize）看作原来的(3)颜色通道来对待，我们想是这样想的，

但是感觉计算量一下又变大了而且感觉不得劲，碍于机器内存不足、时间不够、脑子不够机灵的原因所以双层卷积只是思考了一下就放弃了。

在这之后还是想再参考一些论文实现一下双层卷积吧。

# 5. 参考文献

**1. Deep Learning 学习随记（四）自学习和非监督特征学习**

**http://blog.csdn.net/zhoubl668/article/details/24800819**

**2. 机器学习实验报告：利用3层神经网络对CIFAR-10图像数据库进行分类**

**https://www.cnblogs.com/lvye-song/p/4029750.html**

**3. 卷积网络训练太慢？Yann LeCun：已解决CIFAR-10，目标 ImageNet**

**http://m.blog.csdn.net/llp1992/article/details/48057419**

**4. CNN训练Cifar-10技巧**

**http://www.cnblogs.com/neopenx/p/4480701.html**

**5. Deep LearningUFLDL教程（多篇po文）**

**http://www.cnblogs.com/dmzhuo/tag/deep%20learning/default.html?page=2**

**6. 卷积神经网络\_（1）卷积层和池化层学习**

**https://www.cnblogs.com/zf-blog/p/6075286.html**

**7. 多通道(比如RGB三通道)卷积过程**

**http://blog.csdn.net/u014114990/article/details/51125776**

**8. 非监督学习算法--K均值聚类**

**https://zhuanlan.zhihu.com/p/21558539**

**9. 深度学习浅层理解（三）--- 常用模型之自编码器**

**https://www.cnblogs.com/caocan702/p/5665972.html**

**10. STL-10 dataset**

[**http://cs.stanford.edu/~acoates/stl10/**](http://cs.stanford.edu/~acoates/stl10/)

# 6.贡献率

|  |  |  |  |
| --- | --- | --- | --- |
| 小组成员 | 班级 | 学号 | 贡献率 |
| 夏雨柔（队长） | 2015级软件1班 | 201500800560 | 38% |
| 刘君怡 | 2015级软件2班 | 201500800513 | 36% |
| 吴晓彤 | 2015级数媒班 | 201500800556 | 26% |

# 7.结果截图以及改进过程

请点击查看[结果截图+数据比较\实验数据(包括图片链接).xlsx](结果截图+数据比较/实验数据(包括图片链接).xlsx)