全连接神经网络实现Mnist字符识别

作者：瑶瑶乐 联系邮箱：[bulong\_yaoyaole@qq.com](mailto:bulong@qq.com)

这段时间在学习NLP，要用到神经网络就去研究了下，刚开始的时候是在网上查资料看，只要看懂就没在意，往往过两天又给忘了，或者是编程实现时发现之前一直觉得简单而忽略的细节现在却怎么也搞不明白。于是决定将之前学习的知识点都一一记录同时分享出来，文章都是按我个人对神经网络的理解讲解，有理解不对的地方欢迎大家指正。本博文是一个系列，后续会继续完善CNN、RNN、LSTM等神经网络原理及在NLP中的应用和Tensorflow实现。

1．Mnist数据集

MNIST数据库是一个手写数字的数据库，它提供了六万的训练集和一万的测试集。它的图片是被规范处理过的，是一张被放在中间部位的28px\*28px的灰度图，如图1所示：

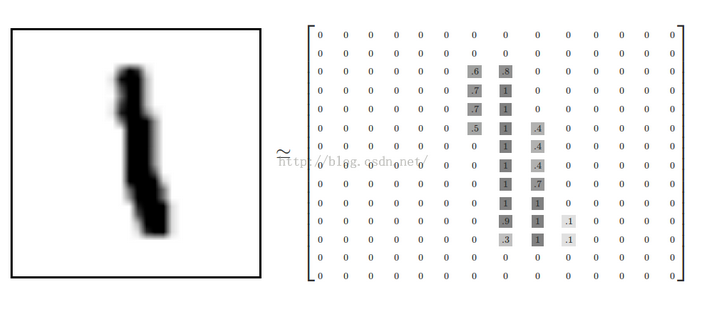
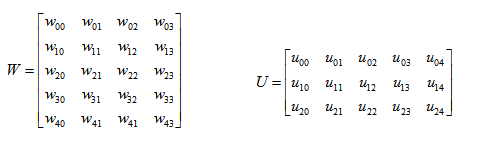


图1（每张灰度图28行，每行28个像素点）

一张mnist样式图片存储为28 X 28的矩阵，矩阵每个元素代表对应像素点的像素值。每个训练/测试样本为一张图片数据和一条标定好的类别组成（共有0、1、2、3、4、5、6、7、8、9）10个类别。本文实现的全连接神经网络目的是输入一张图片，判断是什么数字。

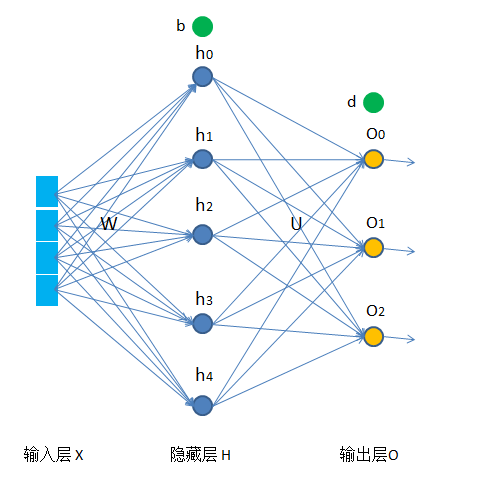
二．全连接神经网络

下图是一个最简单的全连接神经网络，由输入层、隐藏层和输出层构成。图中输入层样本X是一个4维向量，即X=[x0,x1,x2,x3],隐藏层有5个神经元（绿色的为偏置）。隐藏层输出是一个5维向量（一个神经元一个输出,相当于），输出层为有3个输出（3维向量）。

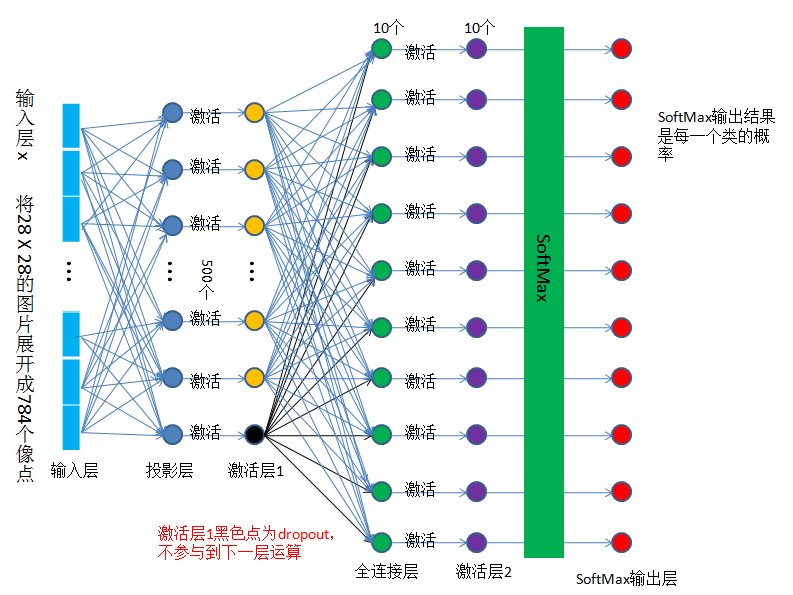


Wij代表输入层第j维到第i个神经元的权值，Uij代表隐层第j个神经元到第i个输出的权值

隐层输出： H = WX + b 输出层输出： O = UH + d



2．Mnist字符识别全连接神经网络



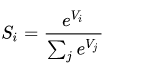
本文实现的全连接神经网络如上图，输入层X为一张图片的展开（每一行连接起来成一条线），为一个784维的向量。投影层有500个神经元。（此图未画偏置）。投影层输出经过激活层后进入全连接层，全连接层有10个神经元，再将全连接层的输出激活后，进行SoftMax输出。该网络不仅深度加深，还添加了DropOut和SoftMax输出。

**2.1 DropOut**

Dropout是指在模型训练时随机让网络中节点按照设置的概率不工作，不工作的那些节点可以暂时认为不是网络结构的一部分，但是它的权重得保留下来（只是暂时不更新而已），因为下次样本输入时它可能又得工作了（有点抽象，具体实现看后面的实验部分）。以上激活层1为例，若设置dropout为0.9，则每个节点在每次训练中不工作的概率为（1-0.9）。某次训练中一个节点不工作（如上图中的黑色节点），并不代表它下次不参与训练。

**2．2 SoftMax**

SoftMax没什么难的，就是输入N个值，输出这N个值的E指数占总和的比例。假设我们有一个数组，V，Vi表示V中的第i个元素，那么这个元素的Softmax值就是：

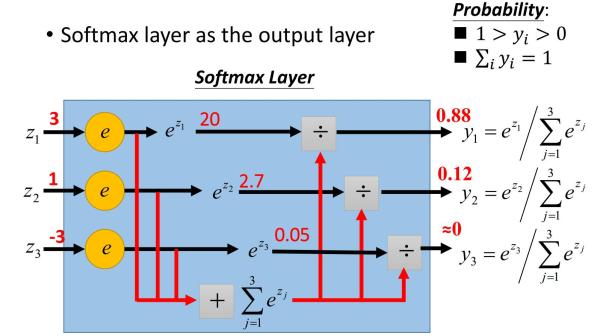


也就是说，是该元素的指数，与所有元素指数和的比值，结合下图更直观的理解softmax机制。

此例中因为是mnist数字字符分类，有10个类别，分别是0，1,2,3,4,5,6,7,8,9.所以softmax输出y\_为一个型为

y\_=[0.05,0.05,0.5,,0.1,0.05,0.05,0.05,0.05,0.05,0.05,]

的10维向量，每一维代表了图片属于该类的概率，所以维相加和为1。



**2.3 交叉熵**

交叉熵为图片真实类别y与神经网络分类得到的图片类别y\_的导数的对数乘积，公式下：



由于mnist数据集每张图片的类别标签为one-hot向量表示，例如一张图标签类别y为3，则表示为y=[0,0,1,0,0,0,0,0,0,0],即第3位为1，其余全为0.

且神经网络输出的分类类别y\_也是一个10维向量，则交叉熵表示为：

****

由公式可以看出，交叉熵越小，分类越准确。因为熵是指不确定度，当一张图片被准确分类时，交叉熵为0.

**2.4 损失函数**

神经网络不是每前向处理完一张图片就进行反向更新梯度的，那样太耗时，而是每次处理完一批（batch）的数据后，统一进行反向计算、更新权值参数，对这一批图片的交叉熵求平均（或求和）遍得到了我们的损失函数：



在训练时反复梯度下降更新权值，当loss最小时模型收敛。本文采用adam优化.

1. 运行结果

源码:

<https://github.com/bulong/mnist/blob/master/mnist_full_connect.py>

结果:

