**神经网络大作业一：深度神经网络**

**姓名：宋昕 学号：6181611008 专业：软件工程专硕（数字媒体学院）**

# 实验目的

1. 熟悉深度神经网络结构。
2. 实现包含三层隐含层的深度神经网络，并用mini-batch的训练策略来训练手写数字。
3. 探究深度神经网络的不同参数对实验精度的影响：
   1. 不同的权值初始化方式
   2. 不同的激活函数与损失函数的组合方式
   3. 不同的隐含层个数

# 实验原理

## 前向传播

BP神经网络是一种这一层的输入均来源于上一层的输出的网络结构，即前馈型神经网络。从输入层开始，基于带权值和偏置的网络逐层更新直到最后的输出层。

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

…

|  |  |  |
| --- | --- | --- |
|  |  | (2) |

其中代表第二层神经元的带权输入，代表第二层神经元与第一层神经元之间连接的权值，代表第二层神经元与第一层神经元之间连接的偏置值，代表第二层的神经元经过激励函数后的输出值，代表激励函数。对于第二层神经元来说，输入就是输入层X。对于最后一层神经元L层即输出层来说，输入就是上一层L-1层的输出值。

## 误差传递

定义误差为，L层即为最后一层输出层，输出层误差方程为：

|  |  |  |
| --- | --- | --- |
|  |  | (3) |
|  |  | (4) |
|  |  | (5) |

将公式用Hadamard乘积符号⊙向量化表示为：

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

误差从最后一层前向更新。在神经网络中每一层神经元都有带权输入，每一层带权输入都和当层的权值*wL*与偏置*bL*有关。要从损失函数入手更新每一层的权值*wL*与偏置*bL*，就要更新每一层的带权输入，即更新每一层的误差即可。

使用下一层误差来表示当前层的误差的公式为：

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

## 参数更新

损失函数*C*对每一层神经元的权值*wL*与偏置*bL*的梯度为：

|  |  |  |
| --- | --- | --- |
|  |  | (8) |
|  |  | (9) |

权值*wL*与偏置*bL*的误差公式为：

|  |  |  |
| --- | --- | --- |
|  |  | (10) |
|  |  | (11) |

权值*wL*与偏置*bL*的更新公式为：

|  |  |  |
| --- | --- | --- |
|  |  | (12) |
|  |  | (13) |

当训练采取mini-batch策略时，每一次batch才更新一次。具体方法是一个batch内参数的梯度累加后求均值。采用mini-batch后权值*wL*与偏置*bL*的更新公式为：

|  |  |  |
| --- | --- | --- |
|  |  | (14) |
|  |  | (15) |

# 实验设计

## 数据集介绍

采用MNIST手写数字库进行手写数字识别实验 。MNIST 数据集来自美国国家标准与技术研究所，训练集由来自 250 个不同人手写的数字构成，其中50% 是高中学生，50% 来自人口普查局的工作人员。测试集也是同样比例的手写数字数据。

手写数字图片为28×28的灰度图像，其中训练集包含50,000张手写数字图片与50,000个标签值，测试集包含10,000张手写数字与10,000个标签值。

## 实现包含三层隐含层的深度神经网络并用于训练

### 网络结构

如图1所示，是深度网络的结构设计图。其中，输入层Input有784个神经元，中间有三个隐含层H1，H2，H3，每个隐含层都有100个神经元，输出层Output有10个神经元。

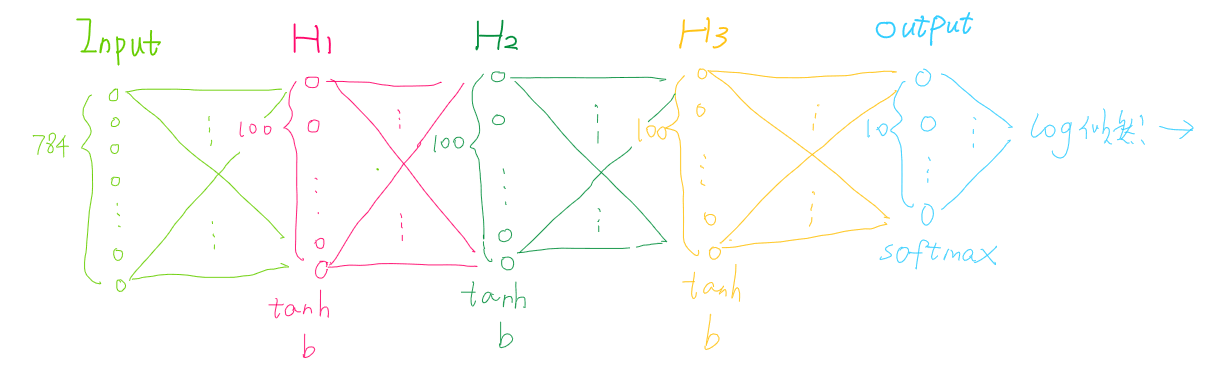


图 1 三个隐含层+tanh+softmax+log似然的网络结构

### 实验参数

输入50,000个图片训练样本。网络结构模型如以上介绍。中间的三个隐含层所使用的激活函数为tanh，输出层所使用的激活函数为sotfmax，输出层所用的损失函数为log似然函数。权值参数的初始化采用分布。采用mini-batch训练策略，mini-batch-size=10。学习速率0.1。最大迭代次数Max=30。

### 实验步骤

定义实验要用到的以上的参数值，初始化各隐藏层与输出层的权值参数。

for epoch in (1, Max):

for batch in (1, 批数=5,000):

输入每一批内的训练样本数进入深度神经网络;

累加这一批次的权重梯度值;

更新权重参数;

在包含10,000个测试样本的测试集上输出识别精度;

每迭代一次输出loss损失值;

## 探究神经网络的不同参数对实验结果的影响

### 权值初始化方式

探究符合高斯分布 的权值初始化方法与符合分布的权值初始化方法对实验结果的影响。

原实验使用的是 的权值初始化方法，现改变权值初始化的方式为其他参数不变并作实验。

### 激活函数与损失函数的组合方式

探究采用tanh（隐含层）+ sotfmax（输出层）+ log似然函数与sigmoid（隐含层+输出层）+最小二乘法的两种不同的激活函数与损失函数的组合方式对实验结果的影响。

原实验使用的组合方式是tanh（隐含层）+ sotfmax（输出层）+ log似然函数，现改变组合方式为sigmoid（隐含层+输出层）+ 最小二乘法，其他参数不变并作实验。

### 隐含层个数

探究采用含有一个隐含层与三个隐含层的深度神经网络对实验结果的影响。

原实验使用的是三个隐含层（100个隐含层神经元），现改变隐含层个数为一（100个隐含层神经元），其他参数不变并作实验。

# 实验结果

原3.2实验的实验结果为图2，图3。如图3所示，在测试集上的正确率达到了92.5%以上。如图2所示，损失曲线也逐渐下降。结果还是较令人满意的。

|  |  |
| --- | --- |
| C:\Users\Administrator\Documents\GitHub\NN-homework\src\dnn_loss.jpg  图 2 原3.2实验的损失曲线 | C:\Users\Administrator\Documents\GitHub\NN-homework\src\dnn_accuracy.jpg  图 3 原3.2实验的正确率 |

改进的3.3.1实验的实验结果为图4，图5。如图5所示，在测试集上的正确率最高能够达到98.02%，这对实验的结果有着显著的提升。损失曲线总体来说有所下降，但在第15次迭代后，损失曲线不再收敛，相应的正确率也不再提升。损失曲线与正确率曲线均比较陡峭。

|  |  |
| --- | --- |
| dnn_loss.1  图4 改进的3.3.1实验的损失曲线 | dnn_accuracy.1  图5 改进的3.3.1实验的正确率 |

改进的3.3.2实验的实验结果为图6，图7。如图7所示，在测试集上的正确率最高只有75%。虽然正确率达不到原3.2实验与改进的3.3.1实验，但损失曲线与正确率曲线却相对平缓得多，损失曲线收敛的更好。

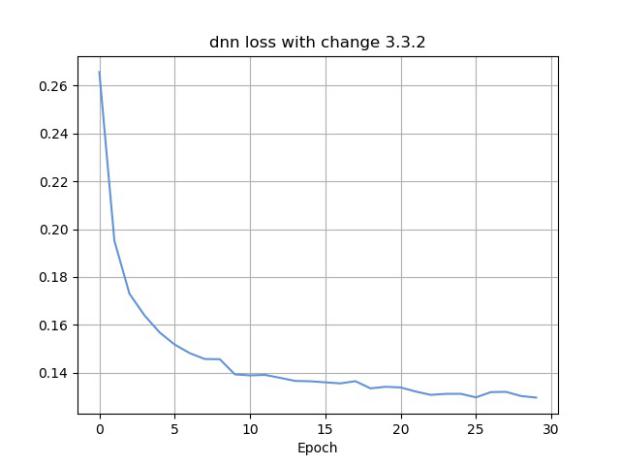
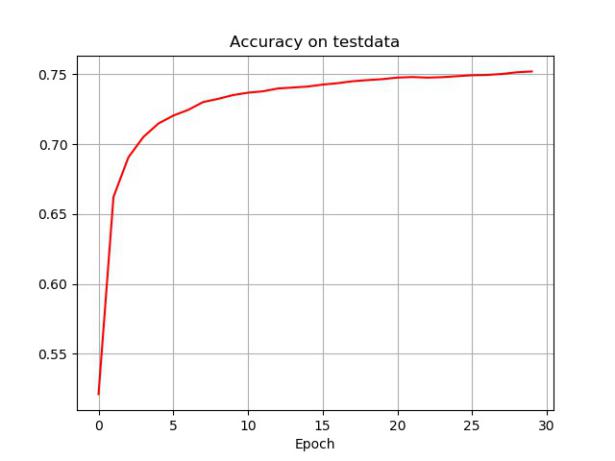
 

图6 改进的3.3.2实验的损失曲线 图7 改进的3.3.2实验的正确率

改进的3.3.3实验的实验结果为图8，图9。如图9所示，在测试集上的正确率达到了94%以上，相较于原实验3.2，识别结果反而还有一些提升。如图8所示，在第20次迭代后，损失曲线不再下降，同时正确率也没有显著提升，应该在此时终止程序。

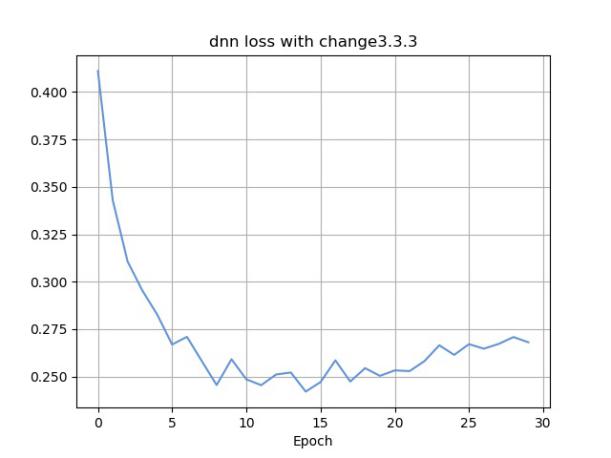
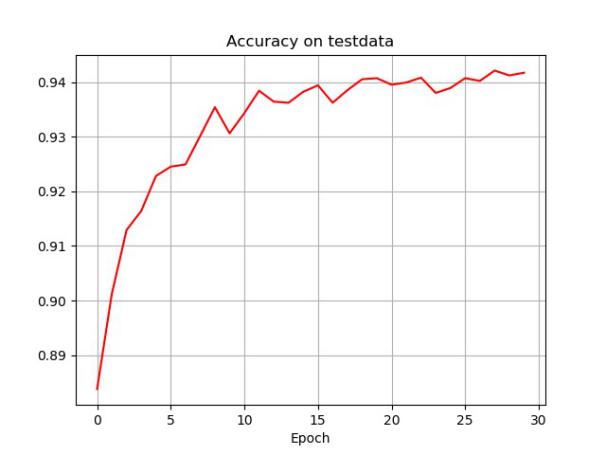
 

图8 改进的3.3.3实验的损失曲线 图9 改进的3.3.3实验的正确率

# 实验结论

1. 使用符合分布的权值初始化方法能够显著提高实验结果的识别正确率。但在第15代后损失曲线不再收敛，此时应该考虑减小学习速率，这可以为进一步的参数调整提供思路与方法。
2. 使用sigmoid（隐含层+输出层）+ 最小二乘法的激活函数与损失函数的组合方式对于实验识别精度的提升没有帮助，但却可以获得更为平缓的损失函数与正确率。使用tanh（隐含层）+ sotfmax（输出层）+ log似然函数虽然获得了更高的识别精度，但损失下降曲线与正确率提升曲线都很抖动，不平稳。
3. 使用含有一个隐含层的网络结构对于实验结果反而有一些提高，所以可以看出深度神经网络的结构并不是越深，实验结果越好。

# 参考文献

<https://www.cnblogs.com/pinard/p/6494810.html>

<http://cs231n.github.io/convolutional-networks/#pool>

<https://www.zybuluo.com/hanbingtao/note/485480>

<https://www.coursera.org/lecture/machine-learning/gradient-descent-8SpIM>