实验报告（提纲）

－－－－BP 神经网络和卷积神经网络 CNN

姓名：陈恩婷　　学号：19335015　　日期：2020/1/2

摘要：本次实验中，我通过创建三层的 BP 神经网络和一个卷积神经网络，完成了手写的 0~9 数字识别，其中 BP 神经网络在测试集上达到了 92.88%的正确率，CNN 神经网络在测试集上达到了 95.84%的正确率。

1．导言

(1) 问题描述：

构造一个三层的 BP 神经网络和一个卷积神经网络，完成手写 0-9 数字的识别：

1. 设计网络的结构，比如层数，每层的神经元数，单个神经元的输入输出函数；
2. 根据数字识别的任务，设计网络的输入和输出；
3. 实现 BP 网络的错误反传算法，完成神经网络的训练和测试，最终识别率达到 70%以上；
4. 数字识别训练集可以自己手工制作，也可以网上下载，要求具有可视化图形界面，能够输入输出。
5. 进一步的，用卷积神经网络实现以上任务，对比深度学习与浅层模型。

(2) 背景介绍：

反向传播（英语：Backpropagation，缩写为 BP）是“误差反向传播”的简称，是一种与最优化方法（如梯度下降法）结合使用的，用来训练人工神经网络的常见方法。该方法对网络中所有权重计算损失函数的梯度。这个梯度会回馈给优化方法，用来更新权值以最小化损失函数。

卷积神经网络（Convolutional Neural Network, CNN）是一种前馈神经网络，对于图像处理有出色表现。

MNIST 数据库是一个大的手写数字数据库，广泛用于机器学习领域的训练和测试。

2．实验过程

(1) 算法思想流程：

BP 算法的流程如下：

function BACK-PROP-LEARNING(*examples*, *network*) **returns** a neural network
inputs: *examples*, a set of examples, each with input vector **x** and output vector **y**
network, a multilayer network with L layers, weights $w_{i,j}$, activation function g
local variables: Δ, a vector of errors, indexed by network node

repeat
for each weight $w_{i,j}$ in *network* **do**
$w_{i,j} \leftarrow$ a small random number
for each example $(\mathbf{x}, \mathbf{y})$ **in** *examples* **do**
/* *Propagate the inputs forward to compute the outputs* */
for each node i in the input layer **do**
$a_i \leftarrow x_i$
for $\ell = 2$ **to** L **do**
for each node j in layer ℓ **do**
$in_j \leftarrow \sum_i w_{i,j} a_i$
$a_j \leftarrow g(in_j)$
/* *Propagate deltas backward from output layer to input layer* */
for each node j in the output layer **do**
$\Delta[j] \leftarrow g'(in_j) \times (y_j - a_j)$
for $\ell = L - 1$ **to** 1 **do**
for each node i in layer ℓ **do**
$\Delta[i] \leftarrow g'(in_i) \sum_j w_{i,j} \Delta[j]$
/* *Update every weight in network using deltas* */
for each weight $w_{i,j}$ in *network* **do**
$w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta[j]$
until some stopping criterion is satisfied
return *network*

对于本次实验中的数据集，在输入层和隐层之间我选用了 ReLU 函数：

```
a1 = max(0, in1); %ReLU
```

在隐层与输出层之间我选用了 softmax 函数：

```
a2 = exp(in2)/sum(exp(in2)); %softmax
```

损失函数为交叉熵函数：

```
errors(1, ind) = -sum(y.*log(a2));
```

BP 算法的误差求导公式如下：

计算误差对权重 w_{ji} 的偏导数是两次使用链式法则得到的：

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial \mathrm{net_j}} \frac{\partial \mathrm{net_j}}{\partial w_{ji}}$$

在右边的最后一项中，只有加权和 $\mathrm{net_j}$ 取决于 w_{ji}，因此

$$\frac{\partial \mathrm{net_j}}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} \left(\sum_{k=1}^{n} w_{jk} o_k \right) = o_i .$$

关于误差的求导，softmax/交叉熵的求导如下：

$$\frac{\partial E}{\partial w_{ij}} = y_i(o_j - t_j)$$

总的来说，误差反传的公式如下：

隐层－输出层权值更新

$$w = w - \alpha \frac{\partial E}{\partial w_{ji}}$$

$$= w - y_i(o_j - t_j) \cdot$$

softmax/crossentropy 推导结果

$$w_2 = w_2 - \alpha \cdot (a_2 - y) \cdot a_1'$$

输入层－隐层权值更新

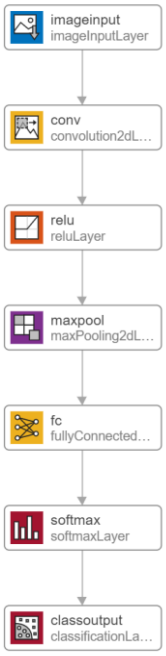
$$w_{ij} = w_{ij} + \alpha \cdot a_i \Delta[j]$$

$$= w_{ij} + \alpha \cdot a_i \, g'(in_j) \sum_j w_{ij} \Delta[j]$$

$$w_1 = w_1 - \alpha \cdot (w_2' * d_2) .* (in_1 > 0) \cdot x'$$

(2) CNN 神经网络流程

本实验的神经网络结构如下：



3．结果分析

本实验在 Matlab R2020b 下运行，两种神经网络的参数分别如下：

(1) BP 算法

学习率 alpha = 0.00005
隐层维度 hidden_dim = 100
迭代次数 epoch = 30

(2) CNN 神经网络

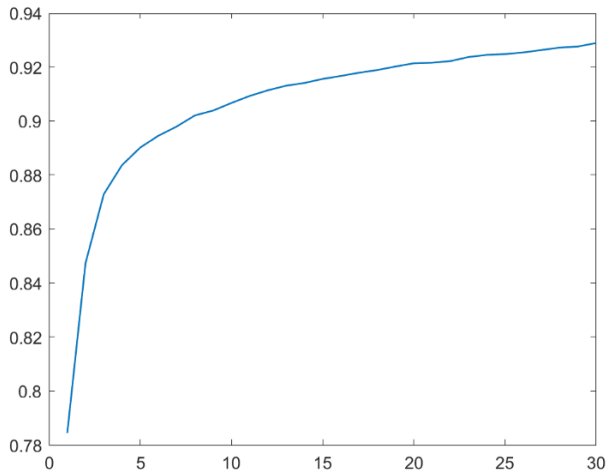
本实验 CNN 各层的设计参数如下：

```
layers =

  具有以下层的 7×1 Layer 数组:

     1   ''   图像输入    28×28×1 图像: 'zerocenter' 归一化
     2   ''   卷积        20 5×5 卷积: 步幅 [1  1]，填充 [0  0  0  0]
     3   ''   ReLU        ReLU
     4   ''   最大池化    2×2 最大池化: 步幅 [2  2]，填充 [0  0  0  0]
     5   ''   全连接      10 全连接层
     6   ''   Softmax    softmax
     7   ''   分类输出    crossentropyex
```

运行结果：

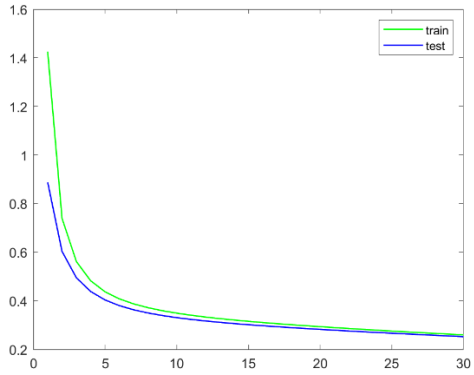
(1) BP 算法

a. 测试的准确率

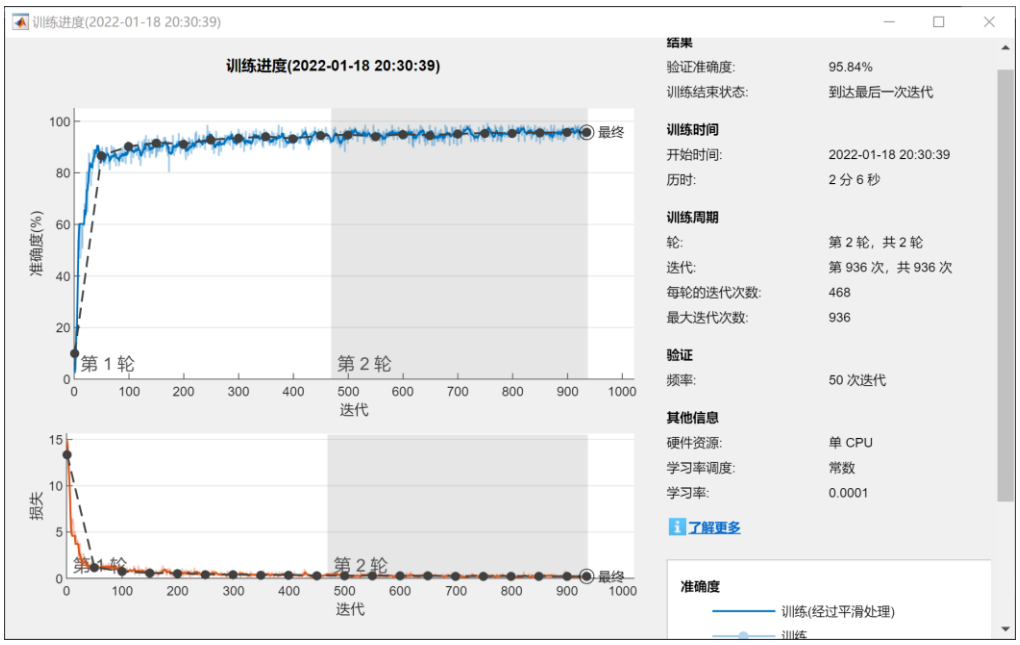


最终准确率为 92.88%。

b. 训练与测试的损失函数值（每个 epoch 将所有的样例求平均值）

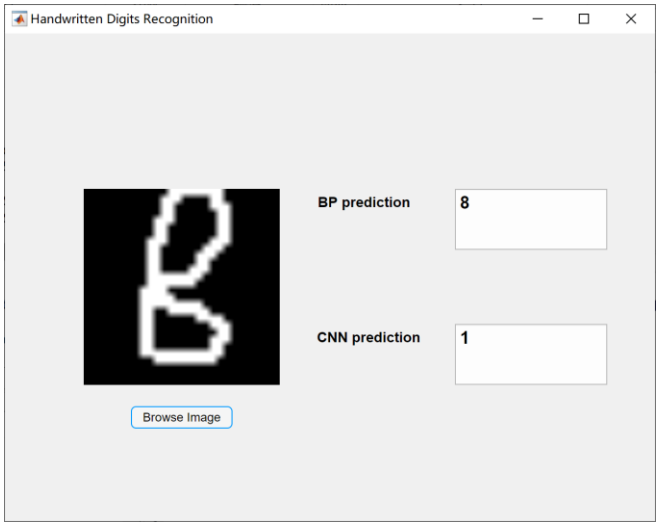


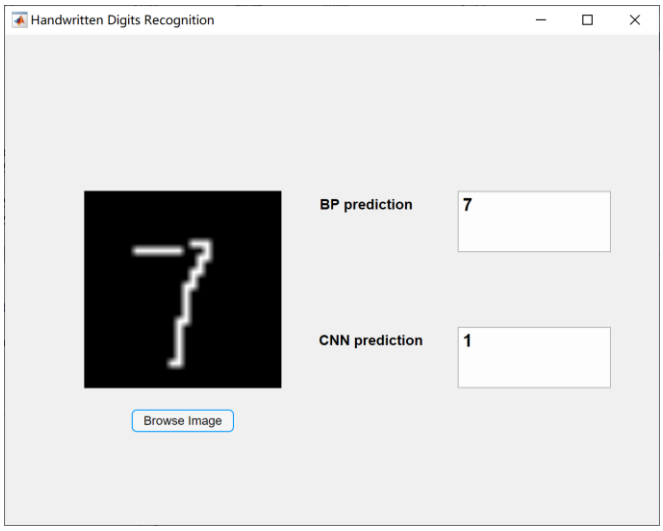
(2) CNN 神经网络

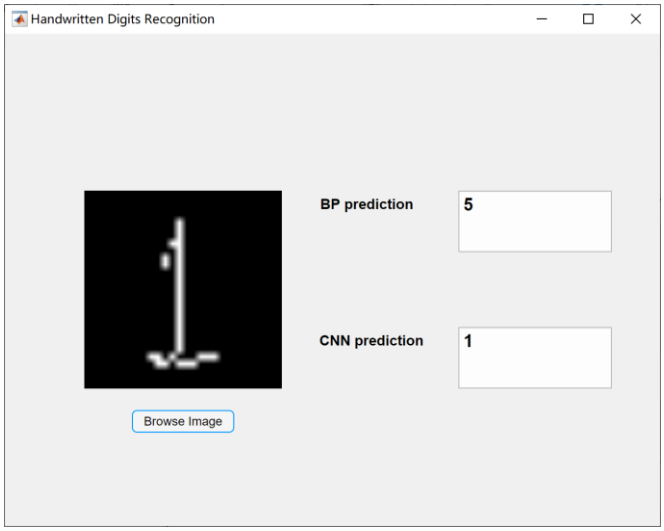


最终准确率为 95.84%。

(3) 在自己写的数字上面验证：







可以看到，两种神经网络一般至少有一个会给出正确结果。前两张图中 BP 算法给出了正确结果而 CNN 网络不行，可能是数字的字体粗细产生了影响。最后一张 CNN 给出了正确结果。可见这两个神经网络在自己创建的数据上表现并不是特别好，泛化能力不是很强。

4．结论

本次实验中，我通过创建三层的 BP 神经网络和一个卷积神经网络，完成了手写的 0~9 数字识别，其中 BP 神经网络在测试集上达到了 92.88%的正确率，CNN 神经网络在测试集上达到了 95.84%的正确率。这次实验也让我在自己实现 BP 算法的过程中，复习了 BP 神经网络的结构与误差反传的公式，收获很多。

主要参考文献(三五个即可)

1. https://zh.wikipedia.org/wiki/%E5%8F%8D%E5%90%91%E4%BC%A0%E6%92%AD%E7%AE%97%E6%B3%95