# 符号说明

## 1.1全连接层

|  |  |
| --- | --- |
| 符号意义 | 符号表示 |
| 输入特征图 |  |
| 输出特征图 |  |
| 权重矩阵 |  |
| 投影矩阵 |  |
| 压缩因子 | b |

## 1.2卷积层

|  |  |
| --- | --- |
| 符号意义 | 符号表示 |
| 输入特征图 |  |
| 输出特征图 |  |
| 权重矩阵 |  |
| 投影矩阵 |  |
| 压缩因子 | b |

# 算法思想

## 2.1计算参数梯度

### 2.1.1前向传播

在前向传播过程中，我们将每个神经网络层的输入特征图和权重矩阵都通过相同的投影矩阵进行压缩，对压缩后的矩阵相乘得到的矩阵作为输出特征图的近似，假设共有L层网络，每一层即k=1到L的网络运算如下。其中全连接层计算公式如下：







同理，卷积层计算公式如下：







### 2.1.2后向传播

因为在前向传播过程中，对输入特征图和输出特征图做了一些变换，所以我们在后向传播时，参数的梯度也会发生相应的变化，并且由于投影矩阵的存在，我们需要额外求出投影矩阵P的梯度，以使得我们在训练过程中能够同时训练投影矩阵P，一次得到更好的准确率。

卷积神经网络的最终输出层的梯度是可以直接得到的，由此我们能很容易的得到每一层错误率对于输出特征图的梯度，对k=1到L，都有 ，根据链式法则由此可得到、、，如下：











根据前向传播的公式带入可求得、

在上述的式子中，、、、因为涉及到sgn函数的求导，但是sgn函数并不连续可导，所以无法直接给出。故一般我们在对其求导时需要做一些不同的处理，大多数都用一些近似的连续函数来做逼近，所以在下面我们根据不同的近似函数做不同的求导处理。

* 对y=sgn(x)的逼近用y=x，即直接去sgn逼近，y’=1

即可求出、、、

所以最终梯度的表达式为：



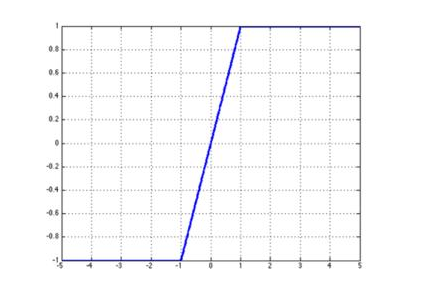








* 对y=sgn(x)的逼近用y=Htanh(x)=max(-1,min(1,x))，即用hard tanh函数逼近，hard tanh函数如图



当x>=-1并且x<=1时，y’=1，否则y’=0，用 表示Htanh函数的梯度

即可求出、、、

所以最终梯度的表达式为：











# 算法1随机生成P

|  |
| --- |
| Algorithm 1: Mini-Batch SGD training with Random Projection.  is the cost function for mini-batch and the functions. L is the number of layers. |
| Require: a mini-batch of (inputs, targets), previous parameters (weights) and (biases),  and learning rate .  Ensure: updated parameters and.   1. Forward propagation:       For k = 1 to L, compute  knowing ,  and    2. Backward propagation:  Initialize output layer’s activations gradient  For k = L to 2, compute 、 if y = sgn(x) approximate y = x      if y = sgn(x) approximate y = Htanh(x)      3. Parameter update: |

# 算法2训练P

|  |
| --- |
| Algorithm 2: Mini-Batch SGD training with Random Projection.  is the cost function for mini-batch and the functions. L is the number of layers. |
| Require: a mini-batch of (inputs, targets), previous parameters (weights) and (biases), (projection matrix), and learning rate .  Ensure: updated parameters  、 、  .   1. Forward propagation:       For k = 1 to L, compute  knowing ,  and    2. Backward propagation:  Initialize output layer’s activations gradient  For k = L to 2, compute 、 if y = sgn(x) approximate y = x        if y = sgn(x) approximate y = Htanh(x)        3. Parameter update: |

# 测试目标

本文根据投影矩阵是否进行训练，提出了神经网络压缩的两种算法。本项目希望利用这两种算法对神经网络进行压缩，并测试在不同压缩比下，神经网络在MNIST数据集上的表现（准确率）。