**南京理工大学计算机科学与技术学院**

**智能计算技术论文**

|  |  |
| --- | --- |
| **班级** | **9191062302** |
| **学号** |  |
| **姓名** |  |

目录

[1](#_Toc125829407)

[1 算法的方法步骤 3](#_Toc125829408)

[2 分析与讨论 4](#_Toc125829409)

[3 优化和改进 7](#_Toc125829410)

# 算法的方法步骤

Positive pass对真实数据进行操作，并调整权重以增加每个隐藏层的goodness。Negative pass对"负数据"进行操作，并调整权重以降低每个隐藏层的goodness。

假设某一层的goodness函数简单地是该层修正线性神经元活动的平方和。学习的目的是使goodness在真实数据中远高于某个阈值，在负数据中远低于该阈值。更具体地说，目标是当输入向量为正(即真实的)的概率通过公式给出，将输入向量正确分类为正数据或负数据。将隐藏向量的长度用作下一层的输入之前对其进行归一化。这将删除用于确定第一隐藏层中的优缺点的所有信息，并迫使下一隐藏层使用第一隐藏层神经元的相对活动中的信息。



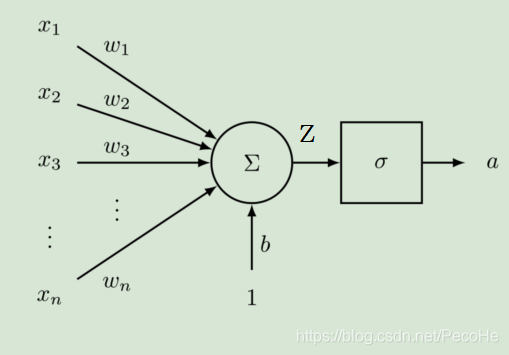
|  |
| --- |
| Forward-Forward Algorithm  Input real data,negative data  is the logistic function,  is threshold,  is the activity of hidden unit j before layer normalization |
| **For** each hidden layer **do**:  Initialize weights, threshold  Positive pass：  operates on real data and adjusts the weights to increase the goodness.    Negative pass：  operates on negative data and adjusts the weights to decrease the goodness.    Normalizes the length of the hidden vector  Output |
|  |

# 分析与讨论

下面将分析并讨论它与一般神经网络（如感知机）的异同；

## 相同点

均是由多个神经元组成的神经网络

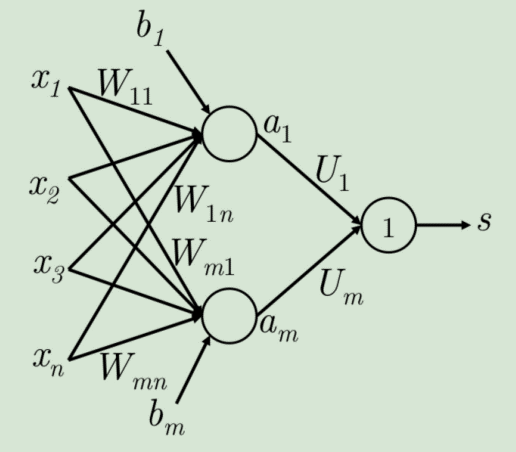


图表 1单个神经元结构

上一层的输出与权重对应相乘相加，然后再加上一个偏置b 之后的值经过sigmod激活函数处理后完成非线性变换。







图表 2神经网络结构

为 n维输入向量。

表示后一层第i个神经元与前一层第j个神经元之间的权重。



，为网络最终的输出结果。

## 不同点

### 一

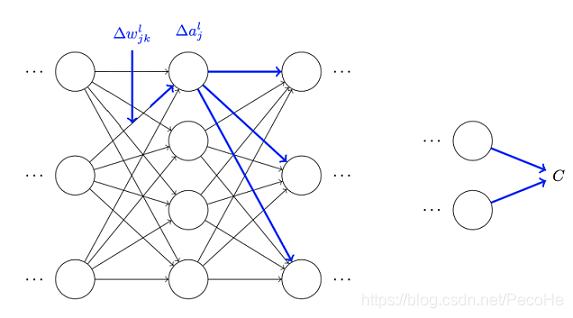
一般神经网络学习算法的学习过程就是优化调整参数，使得损失函数收敛，换句话说就是预测结果和实际结果的误差减小。使用的Backward Propagation Algorithm是个双向传播算法，包含两个步骤：

1. 前向传播的过程是一种推理计算的过程：给定一个输入数据，从神经网络的第一层开始往后计算，到最后一层，输出一个结果。
2. 反向传播的过程，则是在神经网络训练时，计算每次前向传播的结果与真实结果的误差，反向完成一些层的权重（参数）的更新，使得下一次前向传播能比上一次有更好的结果，周而复始，一轮轮的迭代，直到整个网络收敛，神经网络完成训练。

而Forward-Forward Algorithm学习的过程是使goodness函数在正样本中远高于某个阈值，在负样本中远低于该阈值。更具体地说，目标是当输入向量为正(即真实的)的概率通过公式给出，将输入向量正确分类为正样本或负样本。使用前向-前向传播代替前向-反向传播，这两个前向传播计算相同，但是使用的数据集以及最终的目标不同，如下：

1. Positive pass对正样本进行操作，并调整权重以增加每个隐藏层的goodness。
2. Negative pass对负样本进行操作，并调整权重以降低每个隐藏层的goodness。

### 二



图表 3

一般神经网络的Backward Propagation Algorithm中，如图表(3)，假设图中的网络中某个权重发生了一个小的改变，网络的最终损失函数输出为**C**。显然，的变化会引起下一层直接与其相连的一个神经元，以及下一层之后所有神经元直到最终输出C的变化，如图中蓝线标记的就是该权值变化的影响传播路径。

而Forward-Forward Algorithm中认为，如果第一隐藏层的活动随后被用作第二隐藏层的输入，则通过简单地使用第一隐藏层中的活动向量的长度来区分正数据和负数据是微不足道的。为了防止这种情况，FFA在将隐藏向量的长度用作下一层的输入之前对其进行归一化。这将删除用于确定第一隐藏层中的优缺点的所有信息，并迫使下一隐藏层使用第一隐藏层神经元的相对活动中的信息。

总的来说，Forward-Forward Algorithm不需要从最后一层将误差传递过来更新每一层的权值，而是每一层自己使用正负样本的计算来调整权值。

### 三

一般神经网络的反向传播过程中，从最后一层反传时的误差有时会很小，对数值的精度要求要比单纯的前向传播更高。因此依赖数字电路完成运算。这就要求在依靠反向传播进行神经网络的训练中，使用的硬件需要额外加一个AD转换（模拟到数字的转换）电路，将模拟信号转换为数字信号，然后完成计算。

而Forward-Forward Algorithm使用两个前向传播，放弃反向传播，不需要AD转换，只需要使用模拟电路完成运算，大大减少了能量消耗。

# 优化和改进

可用遗传算法优化Forward-Forward神经网络的初始权重和阈值，使优化后的Forward-Forward神经网络能够更好地进行样本预测。

要素包括种群初始化、适应度函数，选择算子、交叉算子和变异算子。

1. 种群初始化。个体编码使用二进制编码,每个个体均为一个二进制串，由隐藏层的权重和隐藏层的阈值连接而成,每个权值和阈值使用固定位的二进制编码。
2. 适应度函数。选择适当的适应度函数
3. 选择算子。选择算子采用随机遍历抽样。
4. 交叉算子。交叉算子采用最简单的单点交叉算子。
5. 变异算子。变异以一定概率产生变异基因数，用随机方法选出发生变异的基因。如果所选的基因的编码为1，则变为0；反之，则变为1。

遗传算法流程为：

1. 随机初始化种群。
2. 计算种群适应度值，从中找出最优个体。
3. 选择操作。
4. 交叉操作。
5. 变异操作。
6. 判断进化是否结束；若否，则返回步骤2）。