

University of South China

**课程作业**

|  |  |
| --- | --- |
| **题 目** | **循环神经网络概述** |
| **学 院** | **计算机科学与技术学院** |
| **课 程** | **现代系统工程导论** |
| **学 号** | **201620810172** |
| **学生姓名** | **孙 溢** |

2016年12月20日

目录

一、 循环神经网络简介 1

1、 神经网络简介 1

1.1 神经网络的结构 1

1.2 神经网络的计算公式 2

1.3 神经网络的分类 2

1.4 神经网络的应用 3

2、 循环神经网络简介 3

3、 小结 4

二、 循环神经网络的训练算法 6

1、 前向计算每个神经元的输出值 6

2、 反向计算每个神经元的误差项 6

2.1 误差反向传递到上一层 6

2.2 误差反向传递到上一时刻 9

3、 计算每个权重的梯度 11

4、用随机梯度下降算法更新权重 12

三、RNN的梯度爆炸和消失问题 13

# 循环神经网络简介

前提声明，本文所有内容都是本人在查阅相关资料之后，再加上本人自身的理解写出，所以文中难免有疏漏之处，还请海涵。

## 神经网络简介

在介绍循环神经网络之前，先简单地介绍一下基本的神经网络。

### 1.1 神经网络的结构

人工神经网络（ANN，Artificial Neural Network），简称神经网络（NN，Neural Network）或类神经网络，是一种模仿生物神经网络的结构和功能的数学模型或计算模型，用于对函数进行估算或近似。神经网络目前没有一个统一的正式定义，粗略地说，神经网络是一组连接的输入/输出单元，其中每个连接都与一个权重相关联。

简单的神经网络的结构如图1.1所示：

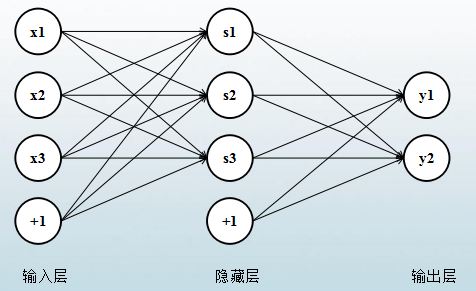


图1.1 简单的神经网络展开图（此图为自制）

输入层（Input layer）：由一个个神经元（Neuron）构成，神经网络的输入对应于每个训练元组的观测属性。

隐藏层（Hidden layer）：简称“隐层”，是输入层和输出层之间众多神经元和链接组成的各个层面。它们对于外部来说是不可见的，所以叫作隐藏层。隐藏层的数量是任意的，拥有两个以上隐藏层的神经网络叫作深度神经网络。

输出层（Output layer）：消息在神经元链接中传输、分析、权衡，形成输出结果，输出的消息称为输出向量。

神经元：图1.1中每个圆圈都是神经元。神经元也叫作感知器，只不过说感知器的时候，它的激活函数是阶跃函数，说神经元的时候，它的激活函数往往选择sigmod函数（）或tanh函数（双曲正切，）。

输入权重：一个神经元可以接收多个输入，每个输入上有一个权重，图1.1中神经元之间的每条连线对应一个权重。网络的权重被初始化为小随机数（例如，由1.0到-1.0，或由-0.5到0.5）。

偏置项：图1.1中“+1”所对应的权重就是偏置项，偏置项的输入恒为1，但它的权重b是一个超参数，由人为设置。偏置项用于给激活函数设置阈值，偏置单元没有输入，即没有连线指向偏置项。

激活函数：它在神经网络中的功能是通过对加权的输入进行非线性组合产生非线性决策边界。激活函数可以有很多种选择，例如阶跃函数、sigmod函数、tanh函数、relu函数（f(x)=max(0,x)），以及softmax函数（）。

图1.1中的神经网络被称为两层神经网络，因为它具有两层输出单元，输入层则不算在内，因为它只用来传递输入值到下一层。

神经网络是全连接的，如果每个单元都向下一层的每个单元（偏置单元除外）提供输入。

### 1.2 神经网络的计算公式

将图1.1收拢后如图1.2所示：

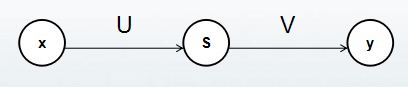


图1.2 简单的神经网络（此图为自制）

在图1.2中，x是输入层的输入向量，s是隐藏层的输出向量，y是输出层的输出向量，U是由输入层到隐藏层的权重矩阵，V是隐藏层到输出层的权重矩阵。

计算公式为：

 （式1）

 （式2）

上述两式中，（式1）是隐藏层的计算公式，其中f是隐藏层的激活函数，（式2）是输出层的计算公式，其中g是输出层的激活函数，b是偏置项。

### 1.3 神经网络的分类

神经网络按模型结构，可以分为前馈型网络（也称为多层感知机网络）和反馈型网络(也称为Hopfield网络）两大类，前者在数学上可以看作是一类大规模的非线性映射系统，后者则是一类大规模的非线性动力学系统。

按学习方式，可以分为有导师学习和无导师学习两类，也称为有监督学习和无监督学习。

按工作方式，可以分为确定性和随机性两类。

按时间特性，可以分为连续型或离散型两类。

……

### 1.4 神经网络的应用

神经网络已经被用于解决各种各样的问题，例如机器视觉和语音识别。

## 循环神经网络简介

循环神经网络（Recurrent Neural Network，以下简称RNN）的目的使用来处理序列数据。在传统的神经网络模型中，是从输入层到隐藏层再到输出层，层与层之间是全连接的，每层内部的节点之间是无连接的。

但是这种普通的神经网络对于很多问题却无能无力。例如，你要预测句子的下一个单词是什么，一般需要用到前面的单词，因为一个句子中前后单词并不是独立的。RNN之所以被称为循环神经网络，是因为在RNN中一个序列当前的输出与前面时刻的输出也有关。具体的表现形式为网络会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层内的节点之间不再无连接而是有连接的。需要注意的是，这里仅指隐藏层，不包括输出层。

简单的循环神经网络如图1.3所示：

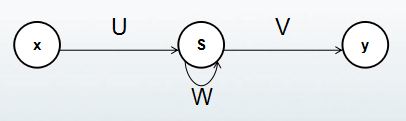


图1.3 简单的循环神经网络（此图为自制）

在图1.3中，x是输入层的输入向量，s是隐藏层的输出向量，y是输出层的输出向量，U是由输入层到隐藏层的权重矩阵，V是隐藏层到输出层的权重矩阵，W则由上一时刻隐藏层到当前时刻隐藏层的权重矩阵。

在RNN中，隐藏层也叫作循环层。

对于W，假设隐藏层有3个节点s1，s2，s3，则W是3×3阶矩阵.其中，对于当前时刻的s1，对它的输出有影响的除了输入层的输入值，还有上一时刻本层的输出值，即上一时刻的（s1，s2，s3），也就是说，在上一时刻本层的输出值中，除了s1以外，s2和s3也对当前时刻s1的输出有影响，所以说隐藏层内的节点之间是有连接的，这种连接是指不同时刻之间的连接。

从图1.3还是无法对RNN有一个比较直观的理解，将其在时间维度上展开后如图1.4所示：

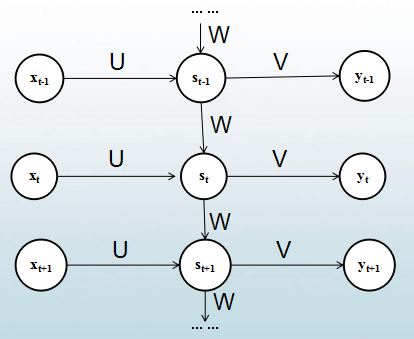


图1.4 循环神经网络结构展开图（此图为自制）

图1.4中的t表示时刻，xt-1是t-1时刻输入层的输入向量，st-1是t-1时刻隐藏层的输出向量，yt-1是t-1时刻输出层的输出向量；xt是t时刻输入层的输入向量，st是t时刻隐藏层的输出向量，yt是t时刻输出层的输出向量；xt+1是t+1时刻输入层的输入向量，st+1是t+1时刻隐藏层的输出向量，yt+1是t+1时刻输出层的输出向量。

U是由输入层到隐藏层的权重矩阵，V是隐藏层到输出层的权重矩阵，W则由上一时刻隐藏层到当前时刻同一隐藏层的权重矩阵。

所以，RNN 的计算公式为：

 （式3）

 （式4）

上述两式中，（式3）是隐藏层的计算公式，其中f是隐藏层的激活函数，（式4）是输出层的计算公式，其中g是输出层的激活函数，b是偏置项。

将（式3）、（式4）与（式1）、（式2）进行比较可以发现，与基本的神经网络相比，RNN只是在计算隐藏层的输出时多加了上一时刻本层的输出值作为输入的一部分。

与传统的神经网络相比，RNN有一个特点，所有的隐藏层共享一套参数（U，V，W），整个网络只用这一套参数。

## 小结

RNN与普通的神经网络的区别在于，RNN的隐藏层的激活函数的输入除了上一次的输出，还包括本层上一时刻的输出。

对于RNN，除了循环神经网络，它还有一个解释是递归神经网络（Recursive Neural Network）。它们还有另外的说法是时间递归神经网络（Recurrent Neural Network）和结构递归神经网络（Recursive Neural Network），时间递归神经网络的神经元连接构成有向图，而结构递归神经网络利用相似的神经网络结构递归构造更为复杂的深度网络。

在本文，RNN特指循环神经网络，即时间递归神经网络。

# 循环神经网络的训练算法

循环神经网络的训练算法BPTT（BackPropagation Through Time）算法包含四个步骤：

1. 前向计算每个神经元的输出值；
2. 反向计算每个神经元的误差项，它是误差函数E对第j个神经元的加权输入netj的偏导数；
3. 计算每个权重的梯度；
4. 用随机梯度下降算法更新权重。

## 前向计算每个神经元的输出值

在开始计算之前，将网络的所有权重初始化为小随机数（例如，有-1.0到1.0，或由-0.5到0.5）。同时，偏置项也被初始化为小随机数。

在初始时刻，由于不存在上一时刻的输出，所以根据（式1）（式2）进行计算，从下一时刻开始，则根据（式3）（式4）进行计算得出各个时刻隐藏层的输出值，以及输出层的输出值。

## 反向计算每个神经元的误差项

反向计算每个神经元的误差项，它是误差函数E对第j个神经元的加权输入netj的偏导数。

注：加权输入也叫作净输入（net input）。

BPTT算法将误差向两个方向传播：一个方向是将其逐层传递到上一层，一个方向是将其沿时间线传递到初始时刻。

### 2.1 误差反向传递到上一层

误差反向传递到上一层即为BP（BackPropagation）算法，称为反向传播算法。

设误差函数为，是关于输出层输出值的函数，例如，其中为训练样本的目标输出，y为神经元的实际输出。

对于的计算分为两种情况，一种情况是第j个神经元是输出层的单元，一种情况是第j个神经元是隐藏层的单元。

1、当第j个神经元是输出层的单元时，根据（式4）得：

 （式5）

 （式6）

上述两式中，t表示时刻，为t时刻输出层的各神经元的加权输入构成的列向量，为t时刻上一层隐藏层的各神经元的输出值构成的列向量，为t时刻输出层的各神经元的输出值构成的列向量。

所以，根据（式5）（式6），由连续求导法则，即链式法则，得：



其中，表示t时刻输出层的误差项构成的列向量。

对于，具体函数具体分析，例如，则。

对于，同样是具体函数具体分析，例如激活函数为sigmod函数（），则：



所以，当误差函数为，激活函数为sigmod函数时，t时刻输出层的第j个神经元的误差项为。

2、当第j个神经元是隐藏层的单元时，根据（式3）得：

 （式7）

 （式8）

上述两式中，t表示时刻，为t时刻本隐藏层各神经元的加权输入构成的列向量，为t-1时刻本隐藏层各神经元的输出值构成的列向量，为t时刻本隐藏层各神经元的输出值构成的列向量。

所以，根据（式7）（式8），由连续求导法则，得：



对于，由于E是关于输出层输出值的函数，而隐藏层的输出值则作为下一层的输入值，逐层传递，最终影响到E，所以，设downstream(j)为与第j个神经元相连的下一层神经元的集合，为t时刻本层第j个神经元的输出值，为t时刻下一层第k（k∈downstream(j)）个神经元的加权输入，则：



因为，t时刻下一层第k神经元的误差项为，所以，其中为本层第j个神经元到下一层第k个神经元的权重。

对于，它是本层各神经元对激活函数求导结果构成的列向量，具体函数具体分析，例如激活函数为sigmod函数（），则。

所以，对于本隐藏层第j个神经元的误差项，



因此，当激活函数为sigmod函数时，t时刻隐藏层的第j个神经元的误差项为



补充：

用矩阵表示误差反向传递到上一层的计算公式：

用表示t时刻第r层的误差项构成的列向量，表示的转置。设第z层为输出层，第r层为隐藏层，要计算第r层的误差项，则：



所以，要想得到第z层之前任意第r层的误差项向量，先要得到误差项传递到上一层的计算公式。

设为t时刻第r-1层的各神经元的输出值构成的列向量，为t时刻第r层的各神经元的加权输入构成的列向量，则由（式3）得：

 （式9）

 （式10）

所以



对于，由（式9）得：

对于，设第r-1层有n个神经元，则其结果为一个n×n阶雅可比矩阵，用表示t时刻第r-1层的第n个神经元的输出值，用表示t时刻第r-1层的第n个神经元的加权输入，则：



所以，



所以，误差项传递到上一层的计算公式为：



误差项从输出层开始逐层传递到第r层的计算公式为：



### 2.2 误差反向传递到上一时刻

由于只有隐藏层的输出值会受到上一时刻本层的输出值的影响，所以误差项反向传递到上一时刻只是在隐藏层上进行，输出层不参与。

设第r层为隐藏层，t时刻为最后时刻，表示k（k<t）时刻第r层的误差项构成列向量，表示的转置。k时刻的输入值通过影响下一时刻的输入值，逐级传递，最终影响到最后时刻的输入值，所以：



所以，要想得到t时刻之前任意k时刻的第r层的误差项向量，先要得到误差项传递到上一时刻的计算公式。

设为t-1时刻第r层的各神经元的输出值构成的列向量，为t时刻第r层的各神经元的加权输入构成的列向量，则由（式3）得：

 （式11）

 （式12）

所以



对于，由（式11）得：

对于，设第r层有n个神经元，则其结果为一个n×n雅可比矩阵，用表示t-1时刻第r层的第n个神经元的输出值，用表示t-1时刻第r层的第n个神经元的加权输入，则：



所以，



所以，误差项传递到上一时刻的计算公式为：



误差项从最后时刻传递到k时刻的计算公式为：



## 计算每个权重的梯度

根据图1.3可知，对于基本的循环神经网络，需要更新的权重矩阵有U、V、W。

对于权重矩阵U，V，现有上下相邻两层，用表示t时刻上一层第i个神经元到下一层第j个神经元的权重，表示t时刻下一层第j个神经元的加权输入，表示t时刻上一层第i个神经元的输出值，表示t时刻下一层第j个神经元的误差项，则t时刻权重的梯度，最终的为各时刻的权重梯度之和。

对于权重矩阵W，用表示上一时刻本层第i个神经元到下一时刻本层第j个神经元的权重，表示t时刻本隐藏层第j个神经元的加权输入，表示t时刻本隐藏层第j个神经元的误差项，表示t-1时刻本隐藏层第i个神经元的输出值，则权重的梯度，最终的为各时刻的权重梯度之和。

## 4、用随机梯度下降算法更新权重

对于权重矩阵U，V，W，用表示更新前的权重，表示更新后的权重，则：



这里选用随机梯度下降算法（Stochastic Gradient Descent，SGD）作为优化算法，设最终的权重梯度为G，因为梯度指向函数值增长最快的方向，所以与梯度相反的方向即为函数值下降最快的方向，则：



上式中，是被称为“学习速率”的常数，由人为设置。

# RNN的梯度爆炸和消失问题

从理论上讲，RNN能够对任何长度的序列数据进行处理。但实际上，RNN并不能很好地处理较长的序列数据，一个主要原因是，RNN在训练中很容易发生梯度爆炸和梯度消失，这导致训练时梯度不能在较长序列中一直传递下去，从而使RNN无法捕捉到长距离的影响。

BP算法和BPTT算法都存在梯度爆炸和梯度消失问题，以BPTT算法为例，分析产生梯度爆炸和梯度消失的原因：



上式中的β定义为矩阵的模的上界，因为上式是一个指数函数，如果t-k很大（即向前看得很远）的话，会导致对应的误差项的值增长或缩小得非常快，从而导致相应的梯度爆炸和梯度消失问题（取决于β大于1还是小于1）。当发生梯度消失问题时，从k时刻再往前，得到的梯度（几乎为0）不会对最终的梯度值有任何贡献，也就是说，无论k时刻之前的网络状态如何，都不会对最终的权重矩阵的更新有任何影响，k时刻之前的信息被忽略了，这也是RNN无法处理长距离依赖的原因。

对于梯度爆炸问题，可以设置一个阈值，当梯度超过这个阈值时，直接截取梯度。

对于梯度消失问题，有三种应对方法：

（1）合理地初始化权重值。初始化权重值时，使其尽可能不取极大值或极小值，以避开梯度消失的区域；

（2）用relu函数（f(x)=max(0,x)）代替sigmod函数和tanh函数作为激活函数，因为对于relu函数，当x<0时，f’(x)=0，当x>0时，f’(x)=1；

（3）使用其他结构的RNN，例如长短时记忆网络（Long-Short Time Memory，LSTM）。LSTM是在基本的RNN的基础上，加入了一个变量来保存长期状态信息。基本的RNN，循环层神经元的输出除了受上一层的神经元的输出的影响，还受到本层上一时刻的输出的影响，但是，它只对短期的输入非常敏感，所以，LSTM又加入了一个长期状态信息来影响循环层神经元的最终输出。在LSTM中，对于长期状态信息，它加入了三个门来控制长期状态信息的保存：一个是“遗忘门”，控制上一时刻的状态信息有多少保留到当前时刻的状态信息中；一个是“输入门”，控制当前时刻的输入信息有多少保留到当前时刻的状态信息中；一个是“输出门”，控制当前时刻的状态信息有多少输出到当前时刻的输出值中。LSTM是通过强制让wf’(x)=1.0来避免梯度爆炸和梯度消失问题。