**《神经科学前沿问题及信息学方法》结课大作业**

**——循环神经网络简介及HDL在BCI中的应用**

**F2103803 何鋆林**

1. **循环神经网络（Recurrent Neural Network）**
2. **RNN简介**
3. **RNN的产生**

CNN，即卷积神经网络的发明通过卷积层和池化层两个关键层结构实现了高维静态输入数据的压缩变形，即通过一系列线性的和非线性的数学运算的叠加对源数据进行处理，从而尽可能地滤过贡献度低的、无关紧要的信息数据，并完成了原始数据从高维度向低维度的高保真坍缩，其结果的关键点在于：既可靠地保存了信息个体与个体之间的关系，同时有极大程度上规避了冗余信息对空间的占用。但是在CNN一系列数学运算的过程中，其逻辑架构忽略了一个因素，那就是时间。换句话说，CNN的设计忽略了原始数据在时间轴上的连续性，从而使得CNN仅对静态的、在时间轴上不连续的原始数据类型起到高效率的滤过作用。CNN的输出只考虑了前一个输入而不考虑其他时刻的输入，于是RNN应运而生。

RNN是一种特殊的神经网络结构。

我非常喜欢这句话，也倾向于将它理解为RNN参考的本源——

“人的认知是基于过往的经验和记忆。”

事实上我们应当尝试去理解、去欣赏任何一句诸如此类的话语，因为这是在为一些计算机科学里的发现赋予一种人性化的解释，就好像神经网络之父福岛邦彦基于人体神经元之间信号传递的方式发明了卷积神经网络CNN一样，在计算机科学的架构里发现人的本源性。

言归正传，RNN与CNN不同之处在于，它不仅考虑前一时刻的输入,而且赋予了网络对前面的内容的一种'记忆'功能。RNN之所以称为循环神经网路，是因为一个序列当前的输出与前面的输出也有关。具体的表现形式是，RNN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，即隐藏层之间的节点不再是无连接的而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出。

图示

描述已自动生成

1. **对RNN隐藏层的分析**

应当认为隐藏层是RNN的核心，实际上每一个隐藏层都是每一种神经网络的核心之所在。对于RNN隐藏层的深入分析也许可以更好地解释上段对RNN的文字叙述。

图示, 示意图

描述已自动生成

Hidden Layer的展开图中有一个相较于CNN显著的不同，即自循环的存在，而这个自身循环就是RNN实现记忆功能的方法。

在Hidden Layer的层级展开图中，t-1, t, t+1表示时间序列。X表示输入的样本。St表示样本在时间t处的记忆，St = ***f*** (W·St-1+U·Xt)。W表示输入的权重，U表示此刻输入的样本的权重，V表示输出的样本权重。

在t =1时刻, 一般初始化输入S0=0, 随机初始化W,U,V, 进行下面的公式计算：

H1 = U·x1+W·s0

S1 = ***f***(h1)

O1 = ***g*** (V·s1)

其中,f和g均为激活函数。其中f可以是tanh,relu,sigmoid等激活函数，g通常是softmax也可以是其他。

时间向前推进，此时的状态s1作为时刻1的记忆状态将参与下一个时刻的预测活动，也就是:

H2 = U·x2+W·s1

S2 = ***f***(h2)

O2 = ***g*** (V·s2)

……

以此类推, 可以得到最终的输出值为:

Ht = U·xt+W·st-1

St = ***f***(ht)

Ot = ***g*** (V·st)

需要注意的是：

1.此处的W，U，V在每一个时刻都是相等的——权重共享

2.可以将隐藏状态简单地理解为：S=***f*** (现有的输入+过去记忆总结)

3.权重的优化依然可以通过类似于CNN的反向传播算法完成，在此不再展开赘述。

1. **基于RNN的两种变形：LSTM和GRU**

RNN处理时间序列的问题的效果很好, 但是仍然存在着一些问题, 其中较为严重的是容易出现梯度消失或者梯度爆炸的问题(BP算法和长时间依赖造成的)。

注意: 这里的梯度消失和BP的不一样，这里主要指由于时间过长而造成记忆值较小的现象。

因此, 就出现了一系列的改进的算法, 这里介绍主要的两种算法: LSTM和GRU。

LSTM 和 GRU对于梯度消失或者梯度爆炸的问题处理方法主要是：

对于梯度消失: 由于它们都有特殊的方式存储“记忆”，那么以前梯度比较大的“记忆”不会像简单的RNN一样马上被抹除，因此可以一定程度上克服梯度消失问题。

对于梯度爆炸：用来克服梯度爆炸的问题就是gradient clipping，也就是当你计算的梯度超过阈值c或者小于阈值-c的时候，便把此时的梯度设置成c或-c。

1. **LSTM**

和RNN相比，LSTM重复的模块拥有一个不同的结构。具体来说，RNN是重复单一的神经网络层，LSTM中的重复模块则包含四个交互的层，三个Sigmoid 和一个tanh层，并以一种非常特殊的方式进行交互。

图示

中度可信度描述已自动生成

上图中，σ表示的Sigmoid 激活函数与 tanh 函数类似，不同之处在于 sigmoid 是把值压缩到0~1之间而不是-1~1之间。这样的设置有助于更新或忘记信息：

1. 因为任何数乘以 0 都得 0，这部分信息就会剔除掉；
2. 同样地，任何数乘以 1 都得到它本身，这部分信息就会完美地保存下来。

相当于要么是1则记住，要么是0则忘掉，所以还是这个原则：**因记忆能力有限，记住重要的，忘记无关紧要的**。

LSTM有通过精心设计的称作为“门”的结构来去除或者增加信息到细胞状态的能力。门是一种让信息选择式通过的方法。他们包含一个sigmoid神经网络层和一个pointwise乘法的非线性操作。

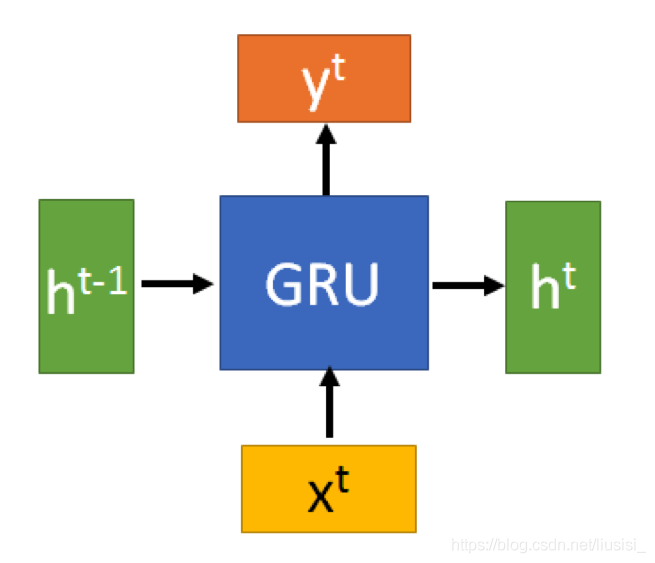
如此，0代表“不许任何量通过”，1就指“允许任意量通过！”

从而使得网络就能了解哪些数据是需要遗忘，哪些数据是需要保存。

1. **GRU**

相比LSTM，使用GRU能够达到相当的效果，并且相比之下更容易进行训练，能够很大程度上提高训练效率，因此很多时候会更倾向于使用GRU。

GRU的输入输出结构与普通的RNN相同：



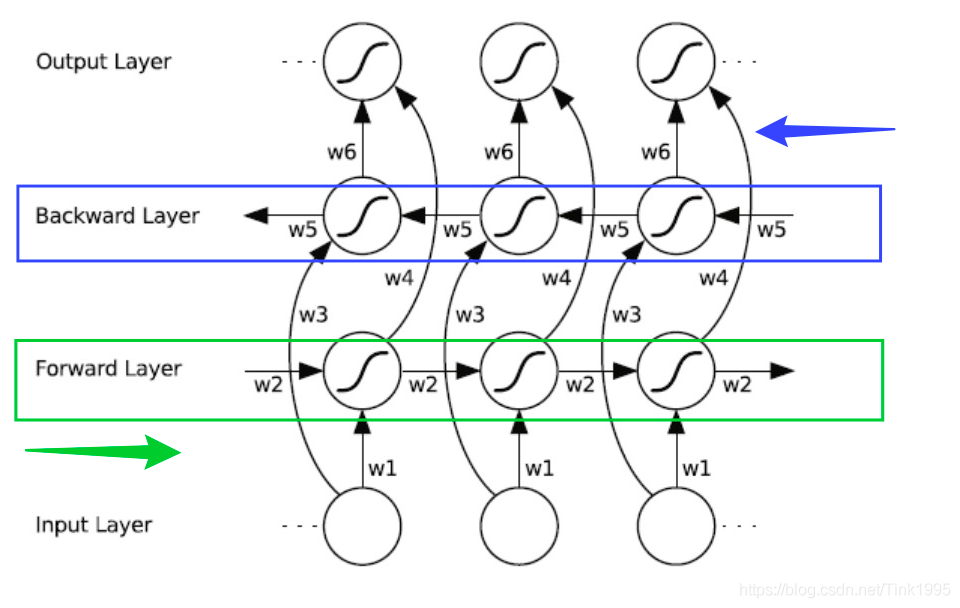
输入：t时刻的输入xt，和t-1时刻的隐藏层状态ht-1，这个隐藏层状态包含了之前节点的相关信息。

输出：t时刻隐藏节点的输出yt和传递给下一个节点的隐藏层状态ht

内部细节在此不展开赘述。

**3.Bi-RNN**

虽然RNN达到了传递信息的目的，但是只是将上一时刻的信息传递到了下一时刻，也就是只考虑到了当前节点前的信息，没有考虑到该节点后的信息。于是就有了Bi-RNN（Bidirectional RNN）。



上面是Bi-RNN的结构图，蓝框和绿框分别代表一个隐藏层，Bi-RNN在RNN的基础上增加了一层隐藏层，这层隐藏层中同样会进行信息传递，两个隐藏层值地计算方式也完全相同，只不过这次信息不是从前往后传，而是从后往前传，这样就不仅能考虑到前文的信息而且能考虑到后文的信息了。

1. **HDL（Hybrid Deep Learning）在BCI中的应用**

在《混合CNN-RNN在脑机接口运动图像识别中的应用》一文中，该研究提出了混合CNN-RNN作为一种基于多通道EEG信号的图像运动识别方法。CNN可以在多通道上提取空间信息，RNN可以从顺序数据中学习时序信息。因此，使用HDL即混合深度学习，具体为混合CNN-RNN进行识别和多通道处理，可以有效地保留时序信息。Hybrid CNN-RNN有可观的准确率，同时比其他方法的学习时间更短。

**三．参考文献**

<https://blog.csdn.net/v_JULY_v/article/details/89894058>

<https://blog.csdn.net/liusisi_/article/details/116236917>

Hybrid CNN-RNN in Motor Imagery Identification of Brain-Computer Interface