摘要

* 脑机接口（Brain Machine Interface, BCI）是一个完成大脑神经元和外部机器互联，通信与控制的接口，为现代生物和医学的研究热点之一。脑机接口可以通过大脑信号采集，信号处理，解码为计算机指令，反馈这四步，为残障人士提供自动化服务。本文主要研究植入式脑电信号处理，压缩，以及哪种脑电信号的解码。在压缩方面，过去的神经信号压缩工作主要关注于非植入式脑电信号，如EEG（脑电图），EMG（肌电图）等，但这些同植入式信号有很大差异，不能直接套用。本文通过植入式电极研究大脑运动皮层信号特性，并提出一种高保真信号压缩方法，而后在信号解码方面分别在空间和时间维度建立深度神经网络，结合神经信号的时序性提高解码准确性，完善模型。

A BMI is an interface connecting neurons to current machine, accomplishing information interaction and control, as one of the hottest research point in modern biology and medicine. This is inspired by the requirements of substituting the BMI for impaired nervous systems [10, 11] and it has been

demonstrated that BMI can transform motor intentions into actions

by decoding the cerebral neuronal signals [4, 5, 6, 7, 8, 9].

1. 神经科学概述
   1. BCI 介绍

由于现代计算机技术和神经科学学科的迅速发展，人们已经可以将大脑中的运动与计算机设备相关联，通过机器捕捉大脑中各个通道的活动（Review见Van Gerven et al., 2009 和Wolpaw et al. 2002）。这种方法与应用统称为脑机接口（brain computer interface，或BCI），以探索大脑活动与特定神经状态的关系。其中特定的神经状态也叫做签名（signatures）。一个BCI需要包括：

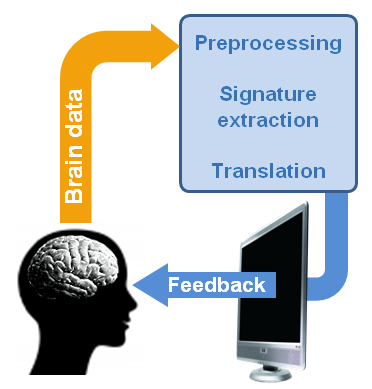
①记录大脑活动

②提取并处理签名

③将签名翻译成计算机指令

④最后返回给用户

整个过程的技术。



* 1. P300

现在有很多测量脑信号的技术，如fMRI （functional magnetic resonance imaging，功能性磁共振成像），NIRS（near-infrared spectroscopy，近红外光谱学），EEG（Electroencephalograph，脑电图），MEG（Magnetoencephalography，脑磁图）等。针对不同的采集信号，其信号预处理方法也各不相同。但是BCI最基本的任务都是正确地识别“签名”并翻译成机器指令以完成用户的意愿。在P300拼写中，可以通过“oddball task”（如图）产生P300信号。这个oddball task是一个关于字符注意的实验，展现一串字符（如SSTSSSSTSS），其中出现频率高的称为标准刺激（如该信号序列中的S），频率低的称为异常刺激（如该信号序列中的T）。当出现一个异常刺激时，300ms后就会在EEG信号中产生一个正向偏移。这里标准刺激和异常刺激的差异可以用来识别所给刺激的类别，然后基于刺激发送信号给计算机指令执行。



Figure 2. 标准刺激（S）中的异常刺激（T）

在P300实验中，字母都展现在一个matrix中，其中同一时刻只有一行或一列亮起，具体哪一行或哪一列随机闪现。如下图所示，当一行一列相继闪烁的交点为指定字母时，测试者可以集中精神在头脑中进行简单计数或者确认，相应就会有P300产生。

1. 神经信号压缩

这一章中，我们介绍神经信号的压缩方法，

1. 信号解码
2. LSTM

事件在时域上的间隔信息很大程度上反映了时序任务的重要信息，如运动控制和节奏检测。隐马尔可夫模型往往忽视这一信息，而循环神经网络（ RNNs ）可以从根本上学习应用这个信息。我们专注于长短期记忆方法Long Short-Term Memory（ LSTM ），因为它已被证明在长时间滞后任务上比其他RNN的方法强 。我们发现LSTM通过内部细胞到多门的“窥孔连接”，在没有任何段式训练样本的情况下可以很好的区分50个和49个时间间隔的spike序列。在没有外部复位和监督学习的情况下，LSTM的变种也能学习生成时间精确的稳定spike流和其他非线性周期模式。这使得LSTM对需要的精确测量或生成有时间间隔的任务，成为了一个很有前途的方法。

1. 介绍

人类可以很快地识别节奏模式序列，即那些有时间间隔的子模式。此外，鼓手等演奏者也能根据运动命令生成有精确时间节奏的节奏序列。这促使人们研究人工系统，去分离或产生通过事件间时间间隔长度传递信息的模式。

目前广泛使用的序列处理方法，如隐马尔可夫模型（HMM） ，一般丢弃这样的信息。他们成功地应用在语音识别，正是因为这种方法不喜欢特定口语单词语速产生的差异。而如节奏检测，音乐处理，和其他本文中所述的任务，都需要精确的时间测量。 虽然HMM可以通过给每个时间间隔单独创建一个内部状态来解决有限个时间间隔的问题，但是这样既麻烦又低效的，并且没有用到HMM的非线性时序伸展不变性。

循环神经网络（ RNNs ）在识别由时间距离定义的模式上更有希望。事实上，尽管隐马尔克夫模型和传统的离散符号语法学习设备被限制在离散状态空间， RNNs在根本上上适用于所有的序列学习任务，因为它们具有图灵能力（ Siegelmann和Sontag，1991）。典型的RNN学习算法（ Pearlmutter ， 1995年）在一个潜在抗噪算法的通用空间中进行梯度下降，这个通用算法采用分布式，用连续值表示的内部状态来将实值的输入序列映射为实值输出序列。

混合HMM的RNN方法（Bengio和Frasconi， 1995）可能能够结合两种方法的优点，但就我们所知，从未被用到我们这里讨论的 准确的事件计时的问题。

之前我们已经介绍了一种新型的RNN——长短期记忆Long Short-Term Memory（ LSTM - Hochreiter和Schmidhuber， 1997），它在涉及很长一段时间延时的任务上比传统RNNs更好。其体系结构允许LSTM容忍输入的事件之间有很长的时间延时（1000步及以上）；而传统RNN用代价更高的更新算法，比如BPTT （Williams and Peng,1990） ， RTRL （Robinson and Fallside ， 1987年，Williams and Zipser, 1992，1992） ，或它们的组合**(Schmidhuber, 1992, Williams and Zipser,1992),** 都无法学习哪怕10步时间延时的序列(Hochreiter, 1991,Bengio et al., 1994,Ho chreiter and Schmidhuber, 1997,Gers et al., 2000,Ho chreiter et al.,2001).

比如，一些我们以前任务需要用LSTM网络对发生于50个离散时间间隔之前的事件予以响应，而不受发生于之前49个时间间隔的事件的影响。就在关键时刻之前，如果有一个有用的“标记”输入通知网络，其下一步的行动将是至关重要的。因此，网络没有必要学习度量50个步骤的时间间隔；它只需要学会保存这50步的相关信息，而后一旦观察到这个标记就用这段信息即可。

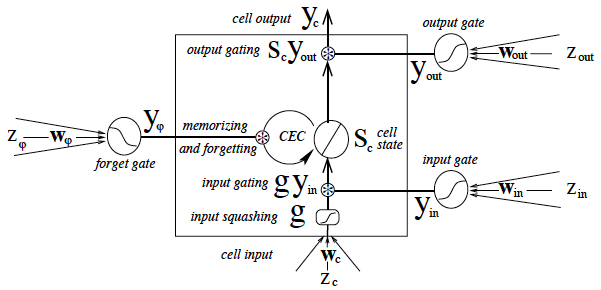
但是如果没有这样的标记呢？如果网络本身需要学习测量并在内部表示这段特定任务的间隔内信息，或生成由特定时间间隔分割的序列模式呢？在这里，我们讨论这个任务的可能性，以及到什么程度是可能的。本文件中的高度非线性的任务不涉及任何时间标志的输入; 相反，他们要求在长时间连续输入流中保持网络的时间精确行和鲁棒性。显然，这样的任务通常不能由基于共同时间窗口的方法解决的，因为它们的泛化能力受时间窗口大小的限制。不幸的是，这意味着我们不能用标准基准进行测试和计时，因为就我们所知，目前没有任何其他网络学会将15个时间滞后推广到45个等。因此，我们不得不以创建一组新的比较任务。

**提纲**：

Section 2 回顾了传统LSTM，参照我们之前发表的文章。Section 3 讲了其连接策略的弱点，介绍了” 窥孔连接”，并描述了LSTM所需的学习算法变化之处。Section 4 在以上阐述的计时任务上比较了窥孔LSTM和传统LSTM，Section 5对我们的方法做了后续讨论。

2. 传统LSTM

我们在LSTM中安置了忘记门（Gers et al., 2000）。LSTM网络中的基本单元是包含一个或多个存储单元（memory cell）和所有cell共享的三个自适应乘法门（图1）的存储块（memory block）。每个存储单元都在其中心有一个自连接线性单元，我们称之为“恒错误旋转木马”（ Constant Error Carousel， CEC ） 。通过无限循环地进行激活和误差信号传递，CEC可以在延长的时间段内提供短期存储。输入，忘记和输出门可以通过训练分别进行学习,即学会将什么信息存入，怎样保存，什么时候读出。将存储单元（cell）组合成块（block）可以使它们共享相同的门（如果任务允许的话） ，从而减少自适应参数的数目。



本文中用

j表示memory block 的下标；

v表示block j中memory cell的下标；

表示第j个block中第v个cell；

表示unit m到unit l之间的权重；其中m是源unit，取决于网络拓扑。

一个源unit激活用表示；

Memory cell c的输出由当前cell的状态和四个输入源确定：

，；

1. 实验