Echo state network

Author: Herbert Jaeger, Jacobs University Bremen, Bremen, Germany

介绍

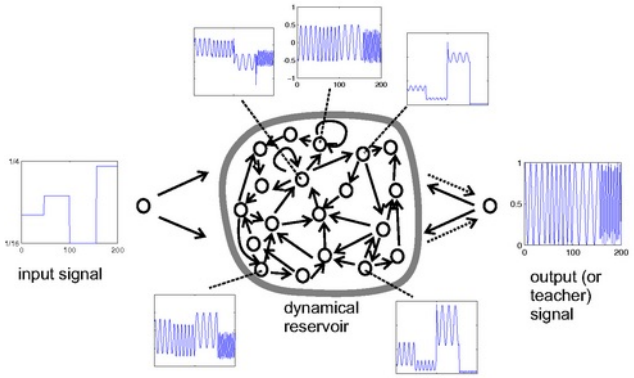
对于前馈神经网络，一种便捷的训练方式是将其input->hidden和hidden->hidden之间的连接权重随机且固定，然后仅训练最后一层，即仅训练hidden->output间的连接权重，而最后的hidden层与output层之间实际为线性模型。通过这种方式，我们得到了一个具有非线性能力同时训练又非常快速的随机前馈神经网络。

在随机前馈神经网络的启发下，类似的思想可以应用在RNN上。相同的，我们将RNN中的input->hidden和hidden->hidden间连接权重随机固定，仅学习hidden->output间的连接权重。这种具有随机机制的RNN便是Jaeger(2001)所提出的ESN。

更具体一点讲，ESN的主要思想是建立一个随机的、固定的、多循环神经元的循环神经网络来进行有监督的学习。在ESN的观点中，循环神经元的组合被视为循环池Reservoir，输入的信号在循环池中的各神经元处各得到一种非线性的前馈信号，然后将这些非线性的信号加以组合得到Reservoir的最后一层状态。此时，Reservoir与output之间的训练过程即为训练Reservoir中产生的前馈信号组去逼近目标信号的过程，即一种线性组合的训练过程。

举例

下面将举一个例子。如图所示，训练ESN去生成一个Sin函数。输入$u(n)$是一组低频信号，目标$y(n)$是一组对应当前输入的sin信号。假设训练集为$D = (u(1), y(1)), \ldots, (u(n_{max}), y(n_{max}))$，其中输入是频率从1/16到1/4Hz的频率信号，输出（目标）为一组由输入频率确定的sin信号。图中，实线表示随机且固定的连接，虚线表示可训练的连接。



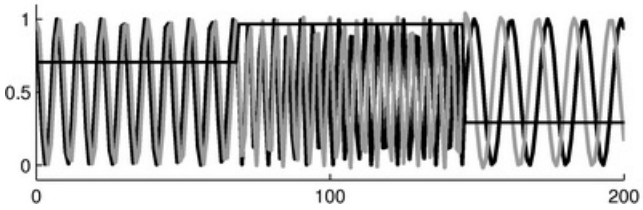
在实现ESN的逼近过程中，可大致分为以下三个阶段：

1. 创建一个随机的ＲＮＮ
    - 创建一个随机的具有Reservoir的RNN。其中，Reservoir的大小（即Reservoir中的神经元个数）是作为超参指定的。
    - 附加RNN的input layer。其中input层与Reservoir之间的连接是all-to-all的，连接权重是随机且固定的。
    - 附加RNN的output　layer。如果训练任务需要output的反馈，则在output与Reservoir之间建立all-to-all的连接，这些连接也同样是固定且随机的；如果任务不需要，则不在output与Reservoir之间建立连接。
2. 计算Reservoir状态
    - 利用训练数据D驱动Reservoir计算Reservoir最后一层的状态。在图中的例子中，训练任务需要利用output的反馈，因此训练数据包括了input的$u(n)$和output的$y(n)$。在不需要考虑output反馈的训练任务

中，只需要训练input的$u(n)$。REservoir的计算结果为一个维度为N维的Reservoir状态，其中的N为Reservoir中神经元的总数。在此用$\mathbf{x}(n)$表示Reservoir中的状态。每一个$x(n)$都是由input计算得到的不同的非线性映射。

3. 训练输出权重
    - 通过第2步中计算出的Reservoir状态$\mathbf{x}(n)$和目标输出$y(n)$对$\mathbf{x}(n)$做线性回归从而得到Reservoir->output之间的权重。

训练结果如图所示:



图中的阶跃函数表示的是input信号。黑色的sin函数表示的是目标即正确的输出。灰色的sin函数为ESN的output结果。其中output与teacher signal间的差异是不可避免的。

状态更新公式

在ESN中，假设有N个reservoir units，K个inputs和L个outputs。输入输出均为序列数据。ESN的Reservoir状态更新公式为：

$$\mathbf{x}(n+1) = f(\mathbf{W}\mathbf{x}(n) + \mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}^{fb}\mathbf{y}(n))$$

其中$\mathbf{x}(n)$为N维的reservoir状态对应N个reservoir units，f是sigmoid激活函数或其他类型的激活函数，$\mathbf{W}$为$N \times N$的reservoir权重矩阵，$\mathbf{W}^{in}$是$N \times K$的输入权重矩阵，$\mathbf{u}(n)$是K维的输入信号，$\mathbf{W}^{fb}$是$N \times L$的输出反馈权重矩阵（在不考虑是输出反馈的训练任务中则无此部分）。

系统状态 $\mathbf{z}(n) = [\mathbf{x}(n); \mathbf{u}(n)]$为reservoir和input在序列数据的第n处的concatenation拼接。

输出output的计算公式为:

$$\mathbf{y}(n) = g(\mathbf{w}^{out}z(n))$$

其中g为sigmoid或其他激活函数，$\mathbf{w}^{out}$是$L \times (K+N)$维输出矩阵。