http://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf

This is a transcription of some ideas in Recifier Network Paper.

2 Background

2.1 Neuroscience Observations

对于生物学的神经元模型，激活函数是我们期望的发射率，因为它是一个当前突触的输入信号产生的总输入(Dayan and Abott, 2001)。一个激活函数被称为反对称或者对称的，当它对一个“非强刺激输入”的响应是强烈的抑制或者激活，当对非强烈刺激输入的响应是 0 的时候它就是单边的。我们期望在计算神经元模型和机器学习模型间的主要的差距包括以下：

关于脑力消耗的研究表明神经元在编码信息时是稀疏和分布式的(Attwell and Laughlin, 2001)，估计出同一时刻神经元被激活的比例介于 1~4%之间(Lennie, 2003)。这对应了表达的丰富度和小动作电位的能量消耗的折衷。在没有额外的规则话，例如 L1 惩罚因子，一般的前向神经元并不带这项属性。例如，sigmoid 激活在 0.5 附近有一个稳定的状态区间，因而在初始化了很小的权重之后，所有的神经元在它们饱和范围的一半处被激活。这在生物学上是不合理的并且不利于基于梯度的优化(LeCun et al., 1998; Bengio and Glorot, 2010)。

生物学和机器学习模型的重要分歧与非线性激活函数有关。一个常用的生物学神经元模型，leaky integrate-and-fire （LIF）(Dayan and Abott, 2001)， 给出下面关于激活率和输入流的关系

$$f(I) = \begin{cases} \left[\tau \log\left(\frac{E+RI-V_r}{E+RI-V_{th}}\right) + t_{ref}\right]^{-1}, & \text{if } E + RI > V_{th} \\ 0, & \text{if } E + RI \leq V_{th} \end{cases}$$

这里 t_{ref} 是不应期（refractory period，在两次激活动作间的最小时间），I 是输入流，Vr 是静息电位，Vth 是临界电位（满足 Vth > Vr），R, E, t 是膜电阻，势电位和时间常量。深度学习和神经网络最常用的激活函数是标准的罗杰斯第 sigmoid 和双曲正切（tanh），它们相当于一个线性变换。双曲正切在 0 处又一个稳定状态，故在优化的角度更可取(LeCun et al., 1998; Bengio and Glorot, 2010)，但是它在 0 附近强制制造出反对称，这在生物神经元里是反常的。

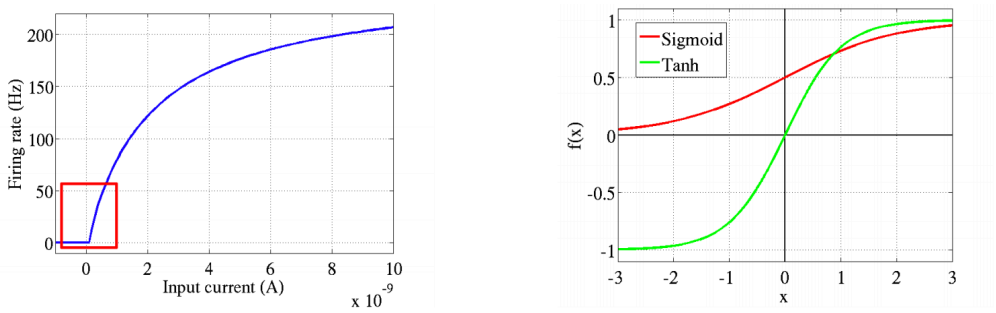


Figure 1: *Left:* **Common neural activation function motivated by biological data.** *Right:* **Commonly used activation functions** in neural networks literature: logistic sigmoid and hyperbolic tangent (*tanh*).

2.2 Advantages of Sparsity

稀疏不仅在计算神经科学和机器学习中开始大热，同时也是统计和信号处理上(Candes and Tao, 2005)。它首先由计算神经科学在视觉系统的稀疏编码场景中引入(Olshausen and Field, 1997)。 它是深度卷积网络一个主要的成分，借助稀疏分布的表达它引出了多种多样的自动编码器 s(Ranzato et al., 2007, 2008; Mairal et al., 2009)，同时成为 DBN 中的关键成分(Lee et al., 2008)。 稀疏惩罚的方法已经用在多种计算神经科学 (Olshausen and Field, 1997; Doi et al., 2006)和机器学习模型中(Lee et al., 2007; Mairal et al., 2009)，特别是深层结构上。然而后者的神经元最后携带着很小但非零的激活或发射概率。我们这里展示了采用一个非线性整流的方法可以最终让神经元达到真 0 激活，从而真正实现稀疏表达。从计算的观点看，这样的表达非常有吸引力，因为：

信息解析，一个深度学习算法声称的目标(Bengio, 2009)是解析出可以解释数据差异的因子。稠密的数据表达是高度缠绕的，因为几乎任何输入的改变都会修改大部分表达向量中的内容。相反，如果一个表达是稀疏并且对小输入的改变鲁邦的，非 0 特征集在输入的细微变化下基本上总是大致保持不变。

有效的变长表达，不同输入可能会包含不同量的信息并且用变长的数据结构来表达会更方便，这种情况在计算机信息表达上很常见。 改变被激活的神经元数允许一个模型对给定输入和需要的输出下，控制表达的有效维度。

现行可分性，稀疏表示更可能是线性可分的，或者更容易在更少的非线性影响下分离，因为信息被用高维空间表示了。另外这反映了原始数据的格式，在文本相关的应用中，譬如说，原始数据本来就非常稀疏。

既分布又稀疏，稠密分布表达是最丰富的表达，比单纯是本地数据具有潜在性的指数级更优效率(Bengio, 2009)。稀疏表达的效率仍然是指数级优，其指数是非零特征的数量。它们可以代表上述两者的一个不错的折衷。

然而强制产生太多的稀疏也许会不利于预测性能，因为对同等数量的神经元，它减少了模型的有效容量。

3 Deep Rectifier Networks

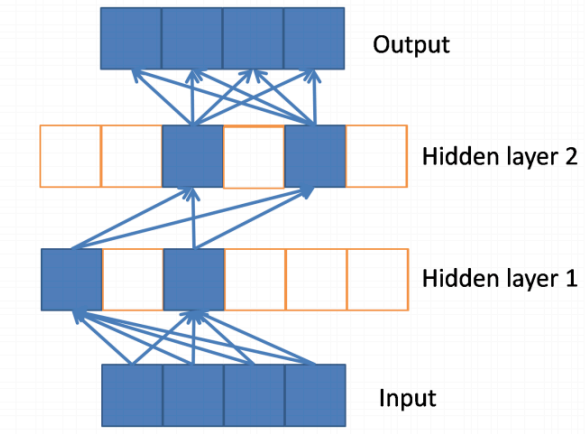
3.1 Rectifier Neurons

神经科学(Bush and Sejnowski, 1995; Douglas and al., 2003)指出皮层神经元很少处于起最大的饱和区间，提出它们的激活函数可以用整流器来近似。大部分之前的神经网络研究在递归网络中包含了一个整流激活函数 s (Salinas and Abbott, 1996; Hahnloser, 1998)。

整流激活函数 ref(x)=max(0, x) 是单边的，故并没有强制出符号对称或者反对称，反而，对非激活输入的响应是 0。然而我们可以通过组合两个共享参数的整流单元来获得对称或者反对称效果。

优点

整流激活函数允许一个网络很容易获得稀疏表达，譬如，经过权重平均分布初始化后，大概 50%的隐藏单元连续输出 0 值，并且这个比例会很容易随着稀疏诱导性正则化而增加。除了在生物学上更合理，稀疏同时可以产生数学上的优势。



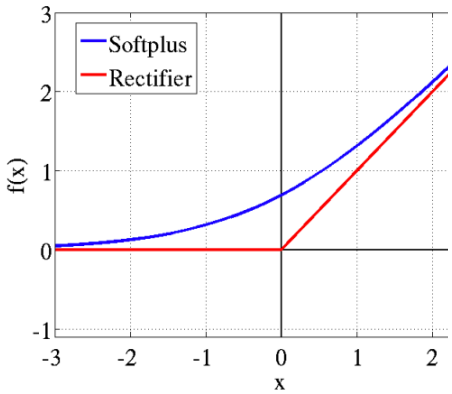


Figure 2: *Left:* **Sparse propagation of activations and gradients in a network of rectifie** input selects a subset of active neurons and computation is linear in this subset. *Right:* **Rectifier a activation functions.** The second one is a smooth version of the first.

图 2 左边所示，网络中唯一的非线性来自于单个神经元有没有被激活的路径选择。对于一个给定的输入，仅有一个神经元的子集是被激活的，在这个子集中，计算是线性的，一旦这个子集的神经元被选中，输出便是输入的线性函数，尽管一个足够大的改变可以催生出被激活的神经元集合的一个离散的变化。这个函数由每个神经元或者由网络基于输入的输出计算，故它是部分线性

的。我们可以把模型看作为一个指数数量个共享参数的线性模型 s (Nair and Hinton, 2010)。基于这种线性，梯度流可以很好的流经神经元激活路径，由于 sigmoid 或 tanh 单元的非线性激活，故它并没有梯度消失效应，同时这里的数学研究也很容易。计算消耗更低，激活时并不需要计算指数函数，并且可以利用稀疏进行计算。

潜在问题

也许会有假设说在 0 处硬饱和会不利于优化因为它阻断了梯度反向传播。为了估计这种潜在影响，我们同时研究了 softplus 激活，softplus(x)=log(1+exp(x)) (Dugas et al., 2001),它是整流非线性的光滑版本。我们失去了稀疏性，但是希望可以更容易训练。然而试验结果似乎与假设相反，结果表示，0 硬饱和可以帮助监督训练。我们假设硬非线性并不会不利，只要梯度可以沿着某个路径方向传播，譬如说这个路径上每一层有一些隐藏单元非 0。对这些 ON 单元赋值信用和谴责，而不是跟平均地分配，我们假设优化会更容易。另外一个问题是由于对激活的不设上界行为引起；也许会有人想用规则话来防止潜在的数值问题。因而我们对每个激活值用 L1 惩罚式，这样同时能提升更多的稀疏。另外别忘了为了更有效表达数据的对称和反对称行为，一个整流网络需要 2 倍于对称 / 反对称网络的隐藏单元的激活函数。

最后，整流网络易于受参数化的病态限制影响。偏移和权重可以用不同却一致的方式规则化，并维持同样的全局网络函数。更精确的讲，对一个 i 层网络的每层，一个尺度因子 a, 那么尺度化参数($W'i = \frac{Wi}{ai}, b'i = bi/\prod_{i=1}^{i} aj$(a)), 输出值会变成$s' = s/\prod_{j=1}^{n} aj$, 因而只要$\prod_{j=1}^{n} aj$ 为 1，网络函数就是唯一的。