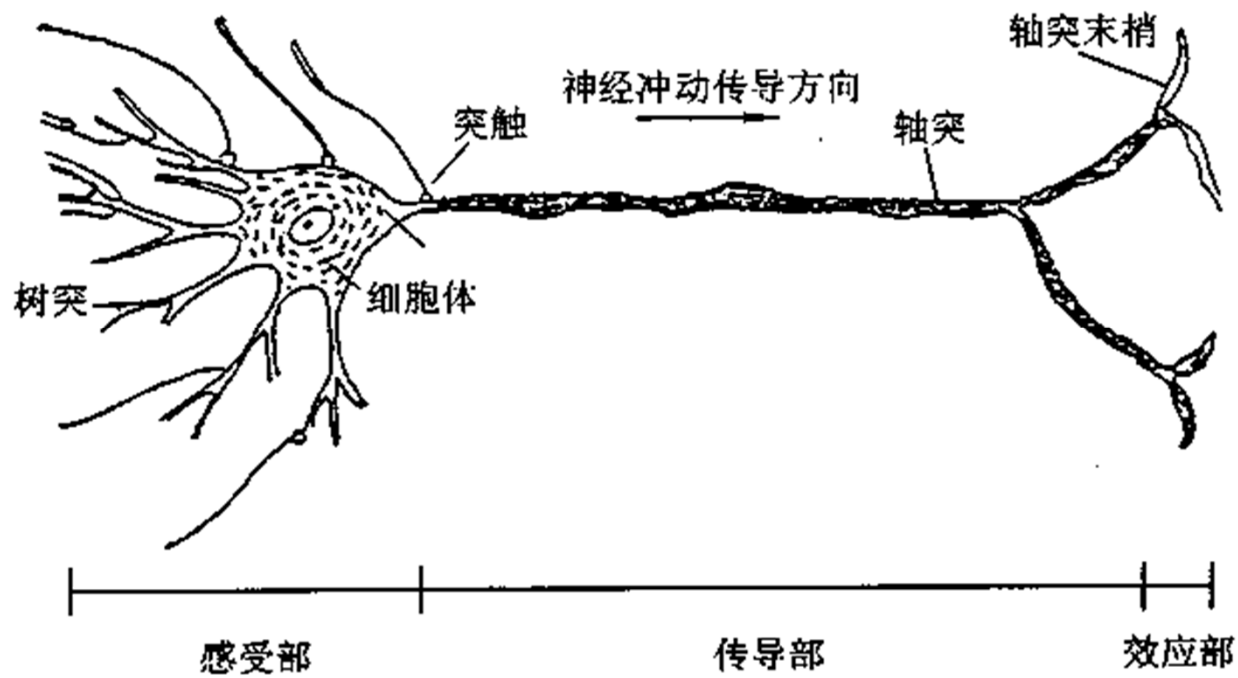


第5-4章 Deep Learning for Binding Specificity

- 人工神经网络
- 多层感知器模型及BP算法
- DeepBind算法介绍

生物神经元(I)



生物神经元(II)

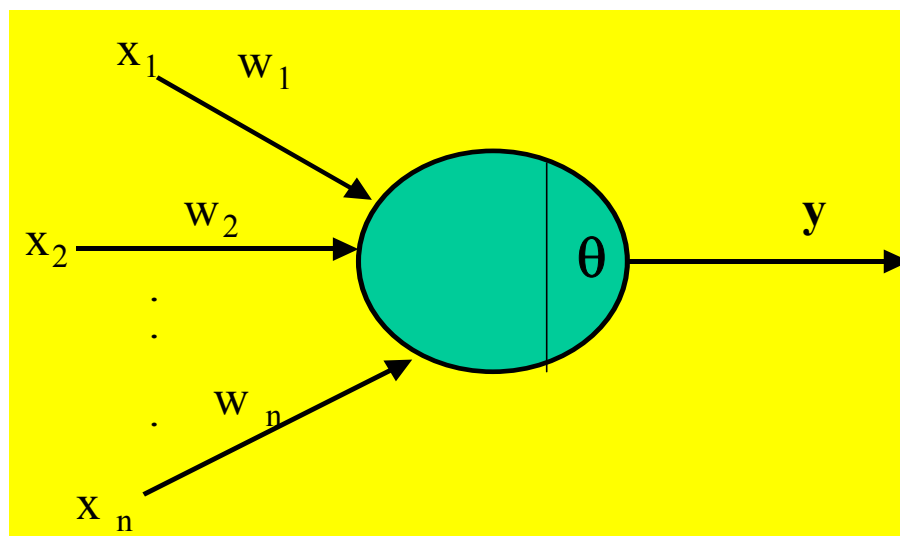
- 树突：接受信号，以突触为界面。
- 轴突：传递和输出信号，以突触为界面。
- 突触(Synapse):生物神经元之间的相互连接从而让信息传递的部位。
- 细胞体：信号处理器

生物神经元(III)

- 细胞的状态，激发和抑制。
- 细胞的状态与接受到的总信号强度有关，只有当总信号集聚到一定的程度才引起细胞的激发。
- 神经元之间的作用强度可以不同，而且具有可塑性(Hebb律)。
- 神经细胞数 10^{11} 量级。
- 每个神经元连接通道 10^4 量级。

MP模型(I)

- McCulloch, Pitts神经元模型(1943).



$$y = g\left(\sum_k \omega_k x_k - \theta\right)$$

MP模型(III)

- 符号函数

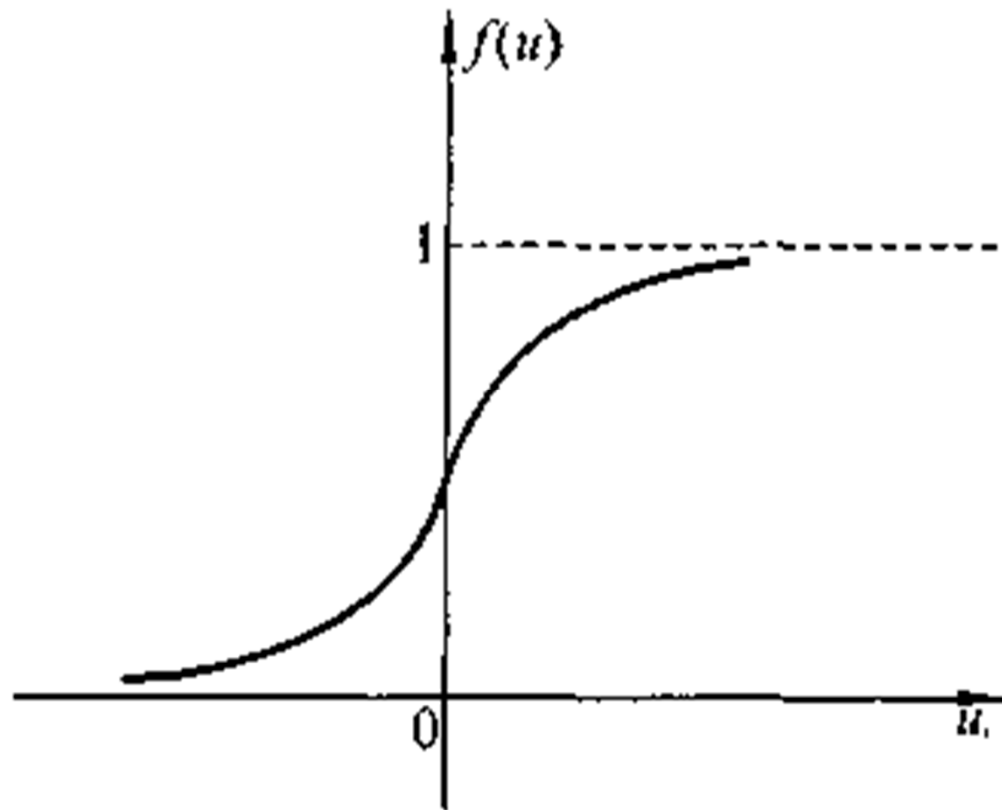
$$g(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

- Sigmoid函数

$$g(x) = \frac{1}{1 + \exp(-ax)}$$

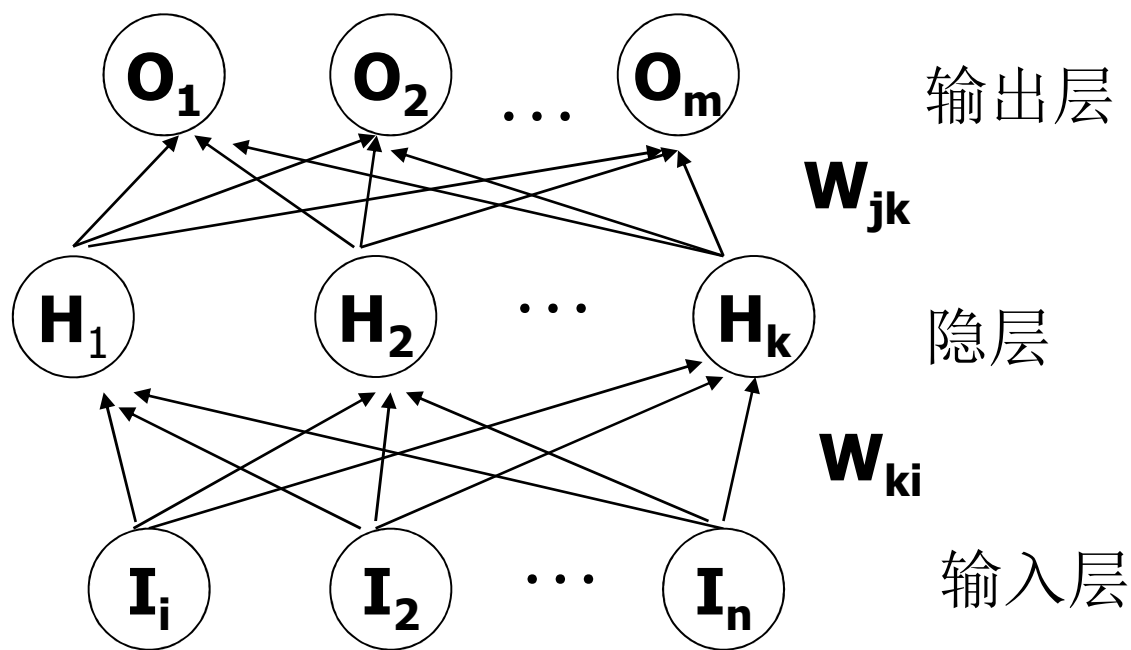
$$g(x) = \frac{2}{\pi} \arctan(ax)$$

Sigmoid 函数



多层感知器网络(MLP)

- 从下到上，底层的输出是高层的输入。
- 多层感知器至少要有三层：输入层、输出层与隐层。



一些记号

- o_j^μ : 样本 μ 对应于输出单元 j 的输入。
- O_j^μ : 样本 μ 对应于输出单元 j 的输出。
- h_k^μ : 样本 μ 对应于隐单元 k 的输入。
- H_k^μ : 样本 μ 对应于隐单元 k 的输出。

$$h_k^\mu = \sum_i w_{ki} I_i^\mu$$

$$H_k^\mu = g\left(\sum_i w_{ki} I_i^\mu\right)$$

$$o_j^\mu = \sum_k w_{jk} H_k^\mu = \sum_k w_{jk} g\left(\sum_i w_{ki} I_i^\mu\right)$$

$$O_j^\mu = g\left(\sum_k w_{jk} H_k^\mu\right) = g\left(\sum_k w_{jk} g\left(\sum_i w_{ki} I_i^\mu\right)\right)$$

有监督学习

- 理想输出 (Teacher)

$$\{T_j^\mu, j = 1, \dots, m; \mu = 1, \dots, N\}$$

- 实际输出

$$O_j^\mu = g\left(\sum_k w_{jk} H_k^\mu\right) = g\left(\sum_k w_{jk} g\left(\sum_i w_{ki} I_i^\mu\right)\right)$$

- 理想输出与实际输出的误差函数;

$$E(w) = \sum_{j, \mu} (T_j^\mu - O_j^\mu)^2$$

误差后传算法推导(I)

- 梯度下降方法以修改权值。

$$\left\{ \begin{array}{l} E(w) = \sum_{j,\mu} (T_j^\mu - O_j^\mu)^2 \\ \Delta w_{jk} = -\eta \frac{\partial E(w)}{\partial w_{jk}} \\ \Delta w_{ki} = -\eta \frac{\partial E(w)}{\partial w_{ki}} \end{array} \right.$$

误差后传算法推导(II)

$$\left\{ \begin{array}{l} \Delta w_{jk} = \eta \sum_{\mu} (T_j^{\mu} - O_j^{\mu}) g'(o_j^{\mu}) H_k^{\mu} \\ \quad = \eta \sum_{\mu} \delta_j^{\mu} H_k^{\mu} \\ \Delta w_{ki} = \eta \sum_{\mu} \sum_j (T_j^{\mu} - O_j^{\mu}) g'(o_j^{\mu}) w_{jk} g'(h_k^{\mu}) I_i^{\mu} \\ \quad = \eta \sum_{\mu} \left(g'(h_k^{\mu}) \sum_j \delta_j^{\mu} w_{jk} \right) I_i^{\mu} \\ \quad = \eta \sum_{\mu} \delta_k^{\mu} I_i^{\mu} \end{array} \right.$$

误差后传算法推导(III)

- 输出单元上的误差

$$\delta_j^\mu = g'(o_j^\mu)(T_j^\mu - O_j^\mu)$$

- 隐单元上的误差

$$\delta_k^\mu = g'(h_k^\mu) \sum_j \delta_j^\mu w_{jk}$$

- 隐单元上的误差实际上是输出单元上误差的线性组合(向后传播)，这就是误差后传算法的由来。

误差后传算法步骤 (I)

- a) 设定参数的初始值;
- b) 逐个或按随机顺序将样品的输入部分, 输进初始参数所对应的网络, 得到输出, 并将其与该样品应有的已知输出进行比较;
- c) 校正最后一层(设为第 n 层)的网络参数, 得到使得输出最接近于应有输出第 n 层的参数, (即将实际输出和期望输出的均方误差取为目标函数, 求出使目标函数达到最小的第 n 层的参数); 并解出使得误差最小的第 $n-1$ 层输出, 将它当作 $n-1$ 层的期望输出。校正的算法大都采用目标函数的梯度下降法。

误差后传法步骤 (II)

- d) 将第 $n-1$ 层当作第 n 层, 重复(c), 校正第 $n-1$ 层的参数, 并求出第 $n-2$ 层使第 $n-1$ 层的输出误差最小的期望输出;
- e) 重复(d), 向前面各层推进, 直到求出所有各层的参数;
- f) 重复步骤 (b) 至 (e), 直到看起来稳定.

神经网络通用逼近定理

- 神经网络只需1个隐藏层和 n 个隐藏节点，即能把任意光滑函数拟合到 $1/n$ 的精度
- 文献： Andrew R. Barron, Universal Approximation Bounds for Superpositions of a Sigmoidal Function. IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 39, NO.3, MAY 1993.

DeepBind

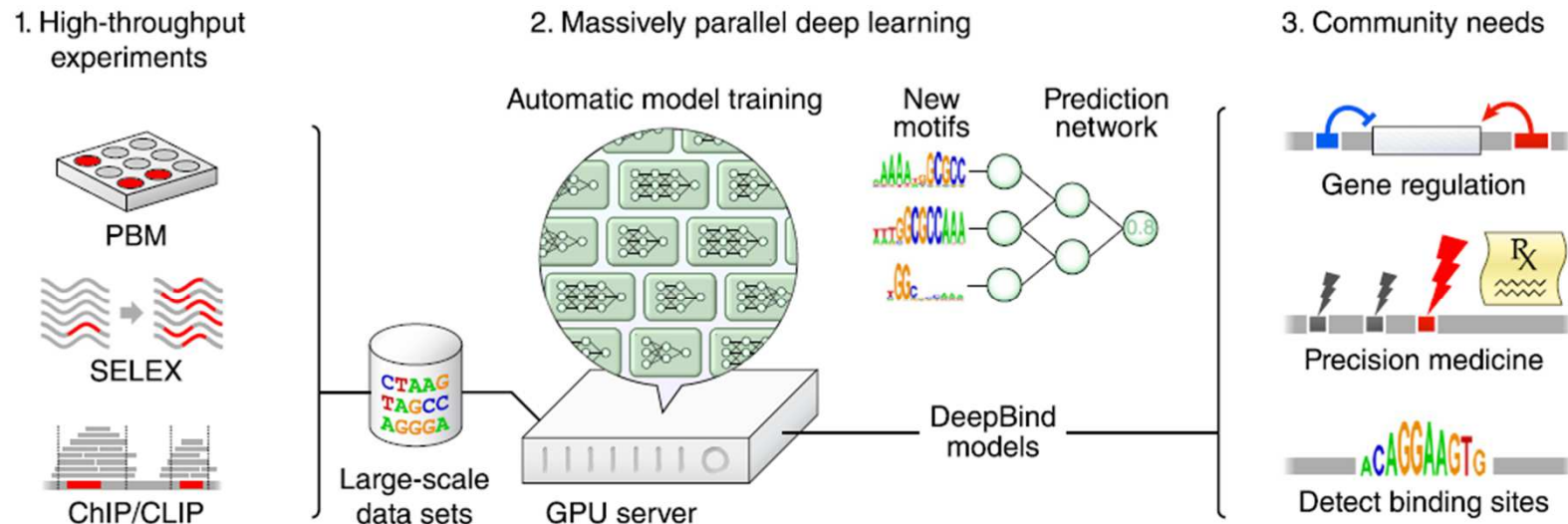


Figure 1 DeepBind's input data, training procedure and applications. 1. The sequence specificities of DNA- and RNA-binding proteins can now be measured by several types of high-throughput assay, including PBM, SELEX, and ChIP- and CLIP-seq techniques. 2. DeepBind captures these binding specificities from raw sequence data by jointly discovering new sequence motifs along with rules for combining them into a predictive binding score. Graphics processing units (GPUs) are used to automatically train high-quality models, with expert tuning allowed but not required. 3. The resulting DeepBind models can then be used to identify binding sites in test sequences and to score the effects of novel mutations.

DeepBind

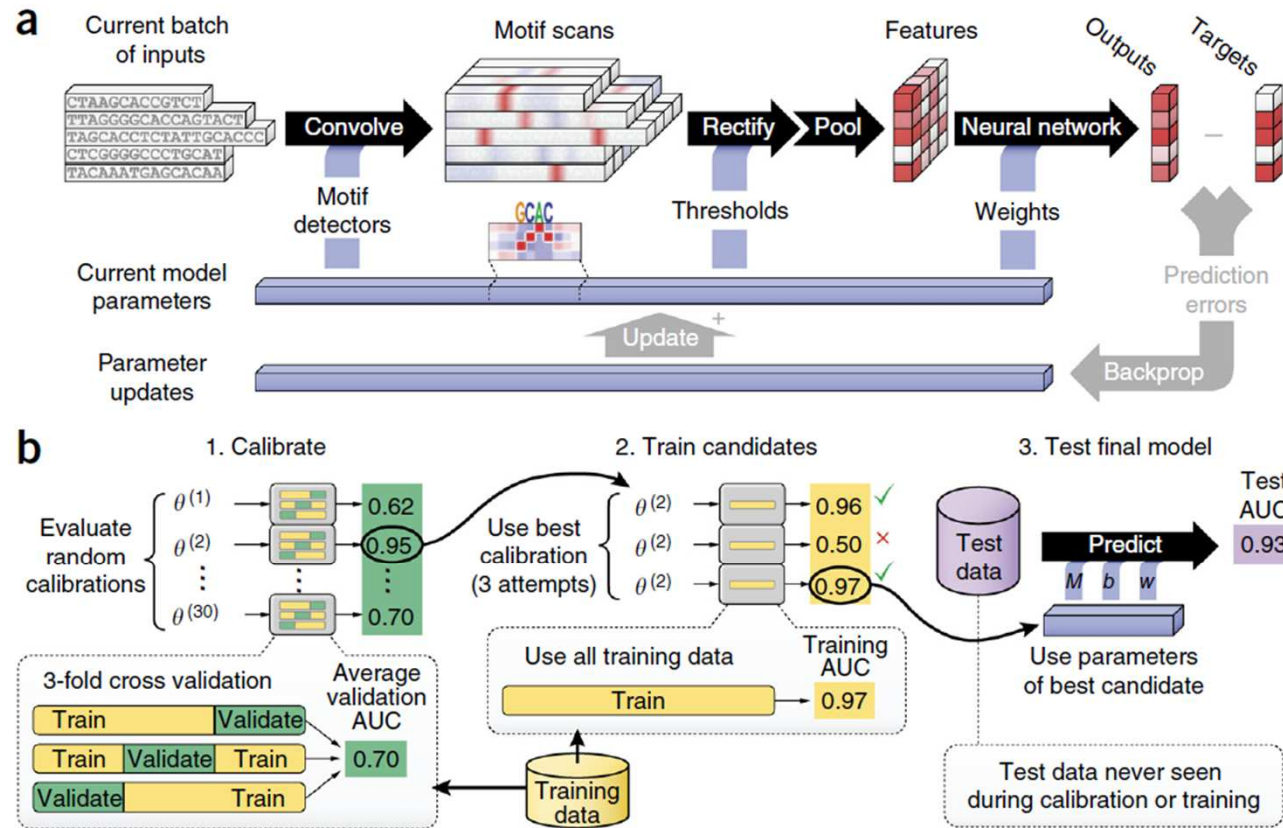


Figure 2 Details of inner workings of DeepBind and its training procedure. (a) Five independent sequences being processed in parallel by a single DeepBind model. The convolve, rectify, pool and neural network stages predict a separate score for each sequence using the current model parameters (**Supplementary Notes**, sec. 1). During the training phase, the backprop and update stages simultaneously update all motifs, thresholds and network weights of the model to improve prediction accuracy. (b) The calibration, training and testing procedure used throughout (**Supplementary Notes**, sec. 2).

预测问题

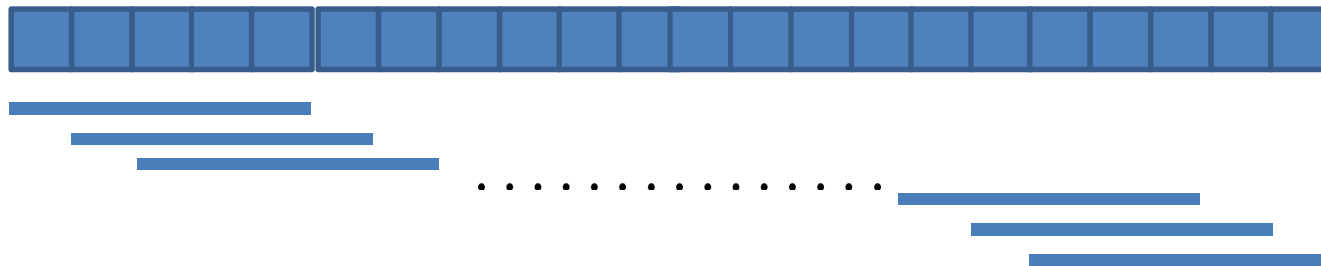
- 给定序列 $S = (s_1, s_2, \dots, s_n)$, 预测它是否和某个TF结合
- 即学习出一个序列S的函数f(S)
 - 最后一步的输出可以是logistic函数 $\text{logit}(f_1(S))$, 以表示某个概率
 - 最后一步的输出可以是某个实值函数, 以回归形式所给出

DeepBind算法设计

- Convolution(卷积)
- Rectification(修正)
- Pooling (池化)
- Neural network(神经网络)

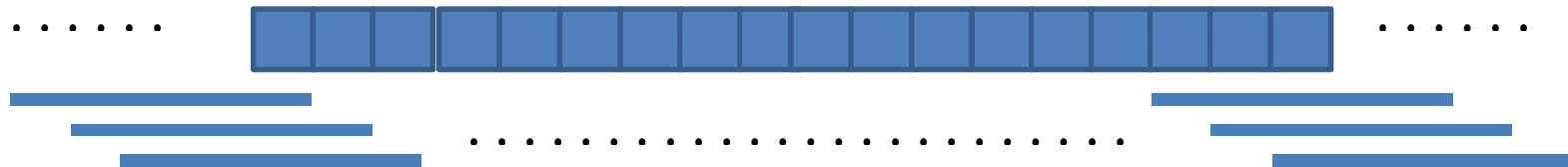
Convolution

- 给定一个长度为 w 的Motif, 在序列 S 上进行Motif扫描；通常的扫描方法是第一个字母作为第一个Motif起点位置，以最后一个字母最后一个Motif终点 f 位置滑动窗口，得到 $n-w+1$ 个位置



Convolution

- 论文中的扫描方式略有不同，是两头补齐以第一个字母作为第一个Motif的终点位置，以最后一个字母作为最后一个motif的起点位置滑动窗口，得到 $n+w-1$ 个位置
- 用等概率方式补齐



Convolution

- 若序列为 $S=AACGT$ ，如果 $w=3$ ，补齐后序列可用如下矩阵描述

$$S = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}$$

Convolution

- 如果用位置权重矩阵(PWM)对Motif建模, 那么第*i*个位置处对该Motif的得分表示为

$$X_i = \sum_{j=1}^w \sum_{l=1}^4 S_{i+j,l} M_{j,l}, \quad 1 \leq i \leq n + w - 1$$

- 该向量 (X_1, \dots, X_{n+w-1}) 称为序列S对该motif的 Motif Scan
- 从序列到这个得分向量的过程就是一个卷积过程, 是序列对该Motif的信息增强过程

Statistical Meaning of Sequence Convolution

- 如果Motif模型中的记分矩阵为似然比(这是Motif常用的打分方法)

$$M_{j,l} = \log \left(\frac{Pr(S_j = l)}{q(S_j = l)} \right)$$

- 那么 x_i 就是第 i 个位置对应的子序列在Motif模型相对于随机模型的Log似然比(Bayes因子)

为什么叫卷积

- 函数卷积

$$[f * g](x) = \int f(t)g(x - t)dt$$

- 图像处理中的卷积运算：采用模板对图像局部区域进行加权求和。比如：

$$R_5 = \sum_{i=1}^9 R_i * G_i \quad R = \begin{bmatrix} R_1 & R_2 & R_3 \\ R_4 & R_5 & R_6 \\ R_7 & R_8 & R_9 \end{bmatrix}, G = \begin{bmatrix} G_1 & G_2 & G_3 \\ G_4 & G_5 & G_6 \\ G_7 & G_8 & G_9 \end{bmatrix}.$$

- 不同的处理方法对应于不同的卷积核

常用卷积核—低通滤波

$$\frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \frac{1}{10} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\frac{1}{16} \begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$$

常用卷积核—高通滤波

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{pmatrix}$$

常用卷积核—边缘检测

$$\begin{pmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{pmatrix}, \quad \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}, \quad \begin{pmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

常用卷积核—梯度方向检测

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & - & -1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & -1 \end{pmatrix}$$

$$\begin{pmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{pmatrix}, \quad \begin{pmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{pmatrix}$$

Rectification

- Rectified motif scan

$$Y_i = \max\{0, X_i - b\}, \quad 1 \leq i \leq n + w - 1$$

- Rectified linear unit (ReLU修正线性单元)是神经网络中除了Sigmoid函数之外的另一种Activation函数

$$f(x) = \max(0, x)$$

- 其被认为更加符合神经元信号激励原理。ReLU在深度网络中已逐渐取代sigmoid而成为主流。

Rectification

- 其平滑解析函数为，称之为SoftPlus函数

$$f(x) = \ln(1 + e^x)$$

- 其导函数就是logistic函数

$$f'(x) = \frac{1}{1 + e^{-x}}$$

Rectification的统计含义

- 按照之前对卷积的理解 x 表示了 Bayes因子, 那么 Rectification 的含义是表示只有 Bayes因子超过某个阈值的才被留下, 否则将其抹为0;
- 其作用是信号的过滤
- 在 Deep Learning 中认为这一步非常关键

Pooling

- Max pooling: 对前面得到的向量取最大

$$z = \max(Y_1, Y_2, \dots, Y_{n+w-1})$$

- Max average pooling

$$\begin{cases} z_1 = \max(Y_1, Y_2, \dots, Y_{n+w-1}) \\ z_2 = \text{Avg}(Y_1, Y_2, \dots, Y_{n+w-1}) \end{cases}$$

Pooling的统计意义

- MAX Pooling表示相对于该Motif是否存在存在于该Motif的Instance, 如果存在, 记下其最好的得分(Bayes因子)
- MAX Average Pooling不仅仅记录最好的评分, 而且还要记录平均的得分

多个Motif 输入

- 前面介绍的只是一个Motif的处理
- 实际中使用了多个Motif

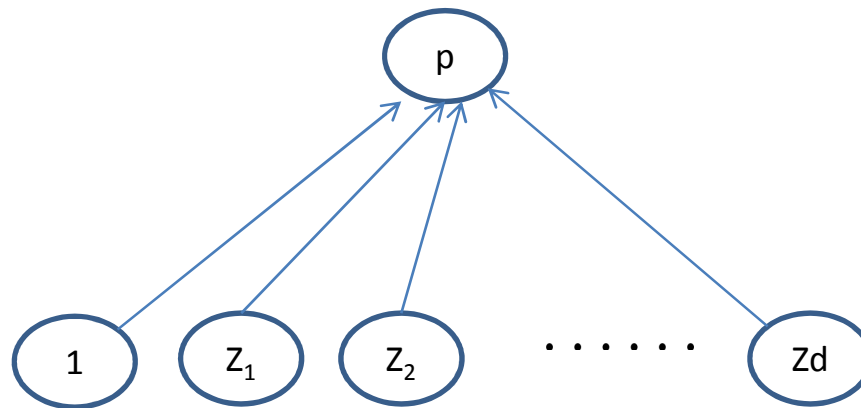
$$M_1, M_2, \dots, M_d$$

- 而这些抽象的Motif将作为下一步人工神经网络(ANN)的输入

Neural Network

- No hidden layer,即为简单的回归模型，包含了 $d+1$ 个参数

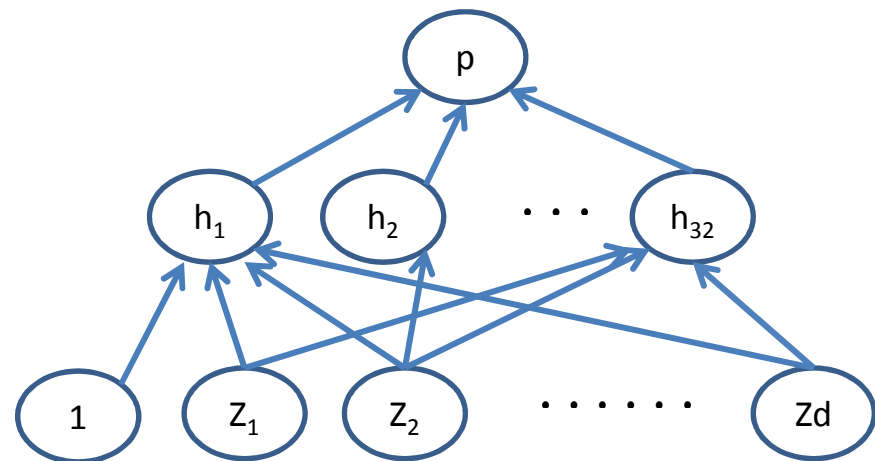
$$p = w_{d+1} + \sum_{k=1}^d w_k z_k$$



Neural Network

- One hidden layer with 32 rectified-linear unit (ReLU),

$$\begin{cases} h_j = \max \left(0, W_{j,d+1} + \sum_{k=1}^d W_{jk} z_k \right), & j = 1, \dots, 32. \\ f = w_{33} + \sum_{j=1}^{32} w_j h_j \end{cases}$$



Neural Network with Drop Out

- “drop out” intermediate values (e.g. the z_k values) by randomly setting them to zero during training.

- 随机生成: $m_k \sim \text{Bernoulli}(\alpha), 0 < \alpha < 1$ $f = w_{d+1} + \sum_{k=1}^d m_k w_k z_k$

- Training:

$$f = w_{d+1} + \sum_{k=1}^d m_k w_k z_k$$

- Testing

$$f = w_{d+1} + \alpha \sum_{k=1}^d w_k z_k$$

优化目标

- 目标函数为

$$\frac{1}{N}LOSS(f^{(t)}, t) + \beta_1 ||M||_1 + \beta_2 ||w||_1 + \beta_3 ||W||_1$$

- 对于像PDB Binding affinity测量数据，LOSS函数就是通常的均方误差(MSE)

优化目标

- 对于像ChIP等数据给出的是0,1数据，选取负对数似然做为LOSS函数

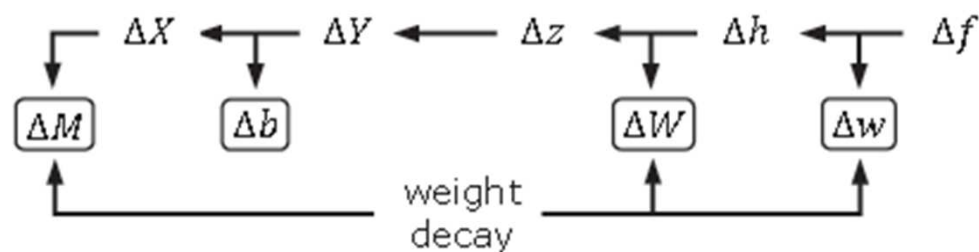
$$\begin{cases} NLL(f, t) = -t \log(\sigma(f)) - (1 - t) \log(1 - \sigma(f)) \\ \sigma(f) = \frac{1}{1 + e^{-f}} \end{cases}$$

优化方法

- 梯度方法(Stochastic Gradient Descent)
- No hidden layer



- One hidden layer with 32 rectified-linear unit (ReLU),



参考文献

- **Babak Alipanahi, Andrew Delong, Matthew T Weirauch, Brendan J Frey.** Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nature Biotech. 33(8), 831-839, 2015, doi:10.1038/nbt.3300.
- Andrew R. Barron, Universal Approximation Bounds for Superpositions of a Sigmoidal Function. IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 39, NO.3, MAY 1993.