Machine Learning Note

BrightHush

2014 年 12 月 24 日

目录

1 Restricted Boltzmann Machine

RBM 是Boltzmann Machine 的一种，在RBM 中，同一层中的单元不存在连接。

1.1 Architecture of RBM

RBM 其实就是一个二分图，只有两层，一个可见层和一个隐藏层，其结构如图1。

RBM的变量表示方法在1已经描绘出来，可见层是用**v**表示，隐层使用**h**表示，其中需要注意的是权值矩阵$W \in R^{n_h \times n_v}$，也就是说W_{ij}表示隐层第i个单元与可见层第j个单元的连接权重。模型的参数为$\theta = (W, a, b)$，其中(a, b)分别表示可见层和隐藏层的偏置。

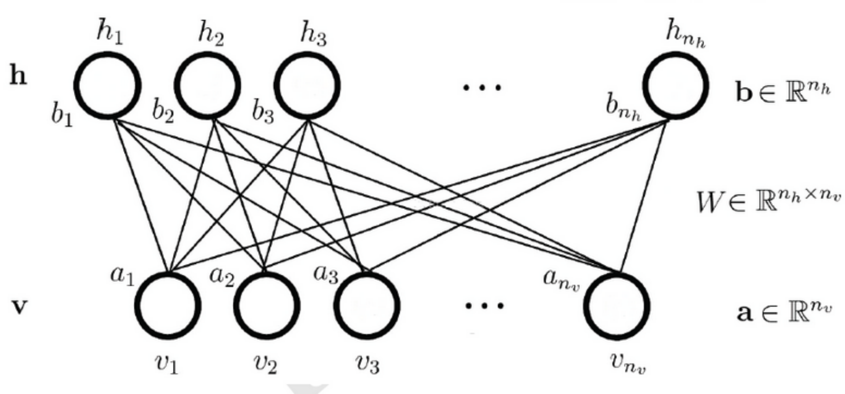

图 1: RBM Architecture

通常来说，我们讨论的RBM的每个神经元的取值为二元的，也就是取值为0或者1。

1.2 Energy Function and Probability Distribution

RBM是一个基于能量的物理模型，通过定义能量函数，我们就可以表示该系统的联合概率了。RBM中，对于给定的状态($\mathbf{v}$, $\mathbf{h}$)，其能量函数表示为（1）。

$$E_\theta(\mathbf{v}, \mathbf{h}) = -\sum_{i=1}^{n_v} a_i v_i - \sum_{j=1}^{n_h} b_j h_j - \sum_{i=1}^{n_v} \sum_{j=1}^{n_h} h_j W_{ji} v_i \tag{1}$$

表示成矩阵向量形式为（2）。

$$E_\theta(\mathbf{v}, \mathbf{h}) = -a^T v - b^T h - h^T W v \tag{2}$$

在能量函数如（1）给定的情况下，定义状态($\mathbf{v}$, $\mathbf{h}$)的联合概率表示为（3）。

$$P_\theta(\mathbf{v}, \mathbf{h}) = \frac{e^{-E_\theta(v,h)}}{Z_\theta} \tag{3}$$

$$in\ which : Z_\theta = \sum_v \sum_h e^{-E_\theta(v,h)} \tag{4}$$

有了联合概率分布，我们可以得到边缘概率分布$P_\theta(v), P_\theta(h)$，也就是我们常说的似然函数（likelihood function），其表示为5

$$P_\theta(v) = \sum_h P_\theta(v,h) \tag{5}$$
$$= \frac{1}{Z_\theta}\sum_h e^{-E_\theta(v,h)} \tag{6}$$
$$P_\theta(h) = \sum_v P_\theta(v,h) \tag{7}$$
$$= \frac{1}{Z_\theta}\sum_v e^{-E_\theta(v,h)} \tag{8}$$

通过定义能量函数以及使用能量函数表示的联合概率分布，以及RBM中的条件独立关系，可以推导出下面的两个类似普通神经网络中的forward propagation 的计算公式（9）。

$$P(h_k = 1|v) = sigmoid(b_k + \sum_{i=1}^{n_v} W_{ki}v_i) \tag{9}$$
$$p(v_k = 1|h) = sigmoid(a_k + \sum_{j=1}^{n_h} h_j W_{jk}) \tag{10}$$

由于已知可见层的情况下，隐层各单元之间条件独立；已知隐层的情况下，各可见层单元条件独立。于是可以得到在已知某一层的情况下，另一层状态的条件概率表达式。

$$p(h|v) = \Pi_{j=1}^{n_h} p(h_j|v) \tag{11}$$
$$p(v|h) = \Pi_{i=1}^{n_v} p(v_i|h) \tag{12}$$

1.3 Likelihood Function

通常来说一个模型需要优化的目标要么是最小化损失函数，要么是最大化似然函数，目的只有一个，那就是要最好的拟合当前的训练样本集合。在RBM中，我们需要最大化的是模型计算的样本似然。

假设我们的样本集合为

$$S = v^1, v^2, ..., v^n$$

其中v^i表示第i个样本，显然我们认为这些样本是独立同分布的，那么我们训练RBM的目标就是最大化下面的似然函数(13)

$$L_{\theta,S} = \Pi_{v^i \in S} P(v^i) \tag{13}$$

将上述似然转化成为log似然，于是取对数便得到

$$ln(L_{\theta,S}) = \sum_{v^i \in S} ln(P(v^i)) \tag{14}$$

1.4 Gradient Computation

最大化上述的log似然，我们通常采用梯度上升法，表示为

$$\theta := \theta + \eta \frac{\partial ln(L_S)}{\partial \theta} \tag{15}$$

不管是Batch Gradient 还是Statistic Gradient 算法，我们都需要分别计算每一个样本下的梯度变化，因此我们需要首先推导单个样本下的梯度计算公式，对于单个样本，我们计算

$$\begin{aligned}
ln(L_S) &= ln(P(v)) && (16)\\
&= ln(\sum_h p(v,h)) && (17)\\
&= ln(\sum_h \frac{e^{-E(v,h)}}{Z}) && (18)\\
&= ln(\sum_h e^{-E(v,h)}) - ln(\sum_v \sum_h e^{-E(v,h)}) && (19)
\end{aligned}$$

上式对参数求导可以表示为

$$\frac{\partial ln(P(v))}{\partial \theta} = -\sum_h P(h|v) \frac{\partial E(v,h)}{\partial \theta} + \sum_{v,h} P(v,h) \frac{\partial E(v,h)}{\partial \theta} \tag{20}$$

(20)前一项可以看成为对应条件概率的期望，第二项看成是对于联合概率的期望。(20)可以进一步表示为

$$\frac{\partial ln(P(v))}{\partial \theta} = -\sum_h P(h|v) \frac{\partial E(v,h)}{\partial \theta} + \sum_v P(v) \sum_h P(h|v) \frac{\partial E(v,h)}{\partial \theta} \tag{21}$$

于是我们只需要计算θ中各个参数对应$\sum_h P(h|v)\frac{\partial E(v,h)}{\partial \theta}$的情况，下面进行推导。对$W_ij$进行求导，

$$\sum_h P(h|v)\frac{\partial E(v,h)}{\partial W_{ij}} = -\sum_h P(h|v)h_i v_j \tag{22}$$
$$= -v_j \sum_h P(h_i|v)P(h_{-i}|v)h_i \tag{23}$$
$$= -v_j \sum_{h_i}\sum_{h_{-i}} P(h_i|v)h_i P(h_{-i}|v) \tag{24}$$
$$= -v_j \sum_{h_i} P(h_i|v)h_i \sum_{h_{-i}} P(h_{-i}|v) \tag{25}$$
$$= -v_j \sum_{h_i} P(h_i|v)h_i \tag{26}$$
$$= -v_j(P(0|v)0 + P(1|v)1) \tag{27}$$
$$= -v_j P(h_i = 1|v) \tag{28}$$

对a_i进行求导，

$$\sum_h P(h|v)\frac{\partial E(v,h)}{\partial a_i} = -\sum_h P(h|v)v_i \tag{29}$$
$$= -v_i \tag{30}$$

对b_i进行求导，

$$\sum_h P(h|v)\frac{\partial E(v,h)}{\partial b_i} = -\sum_h P(h|v)h_i \tag{31}$$
$$= -P(h_i = 1|v) \tag{32}$$

将上述式子分别代入到21 中，我们可以得到各项梯度计算函数为

$$\frac{\partial ln(P(v))}{\partial W_{ij}} = v_j P(h_i = 1|v) - \sum_v P(v)P(h_i = 1|v)v_j \tag{33}$$

$$\frac{\partial ln(P(v))}{\partial a_i} = v_i - \sum_v P(v)v_i \tag{34}$$

$$\frac{\partial ln(P(v))}{\partial b_i} = P(h_i = 1|v) - \sum_v P(v)P(h_i = 1|v) \tag{35}$$

上述三个公式中，都需要计算$\sum_v$，这个复杂度是$O(2^{n_v})$的。对于这种求期望的形式，我们通常可以使用采样的方法进行计算，获得样本之后直接计算$\sum_v$即可。如果我们使用Gibbs 采样方法，需要足够次数的状态转移才能保证采样到的样本符合目标分布，然而采集大量的样本则加重了RBM训练的复杂度，所以在求解RBM参数的时候，通常采用的Contrastive Divergence 算法来降低RBM参数训练的复杂度。

1.5 Contrastive Divergence Algorithm

在式子(33, 34, 35)中的$\sum_v$的项如果使用MCMC算法，需要经过较多的步数才能收敛到目标分布。在RBM模型中，我们的目的是要你和训练样本的分布，那么如果让MCMC从训练样本分布出发，是否会更快的收敛到目标分布呢？基于这个想法，2002年Hinton提出了Contrastive Divergence 算法。在CD-k算法中，依次按照下面的两个步骤进行采样：

- 通过$P(h^t|v^{t-1})$采样得到h^t
- 通过$P(v^t|h^t)$采样得到v^t

然后使用$P(v^k|h^k)$代替式子(33, 34, 35)中的$\sum_v$ 对应的期望项，其实相当于说我们使用第k次采样得到的样本来近似期望。

$$\frac{\partial ln(P(v))}{\partial W_{ij}} = v_j^0 P(h_i = 1|v^0) - P(h_i = 1|v^k)v_j^k \tag{36}$$

$$\frac{\partial ln(P(v))}{\partial a_i} = v_i^0 - v_i^k \tag{37}$$

$$\frac{\partial ln(P(v))}{\partial b_i} = P(h_i = 1|v^0) - P(h_i = 1|v^k) \tag{38}$$

在实际应用中k取1的时候，就能达到非常好的效果了。通过CD-k算法，现在梯度是可计算的了。

1.6 Evaluate RBM

对于已经学习得到或者正在学习的RBM，我们应该如何评价RBM呢？如果我们使用$ln(L_S)$的话，分母中存在Z难以计算。于是我们可以考虑**重构误差**，也就是说以训练样本作为初始状态，经过一次RBM定义的分布进行一次Gibbs采样得到的v'与原始v的差异，这个差异可以定义为1范数或者2范数。显然，重构误差能在很大程度上描述RBM对训练数据拟合的似然情况。

2 References

1 Convolutional Neural Networks,
http://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/
.

2 数据挖掘系列（10）卷积神经网络算法的一个实现,
http://www.cnblogs.com/fengfenggirl/p/cnn_ implement.html.

3 受限波兹曼机（RBM）学习笔记,
http://blog.csdn.net/itplus/article/details/19168937