代码说明：

网络结构为3层，较低的两层为CRBM，最高的一层为RBM，整个网络采用DBM的训练方式进行学习。

说明：预留但未实现的，用背景色表示，在全局范围内制定用背景色表示

主要数据结构

* opts

|  |  |
| --- | --- |
| 变 量 名 | 功 能 |
| batchsize | the number of input image batches |
| MC | the number of Markov Chain |
| MFstep | the number of mean-field steps |
| epoch | the number of iterations |
| CDstep | the steps of Contrastive Divergence sampling |
| alpha | learning rate of weights |
| momentum | momentum of weights |
| decay | the weights’ decay rate |
| sparsity | sparsity of layer units |

* CDBM

1. 数据成员

|  |  |
| --- | --- |
| 变 量 名 | 功 能 |
| layers | configuration about levels of CDBM |

1. 函数

|  |  |
| --- | --- |
| 函 数 名 | 功 能 |
| cdbmSetup | initialize and configure the net |
| cdbmPretraining | do pretrain the cdbm |
| cdbmTrain | train cdbm |
| cdbmBottomup | do bottom-up forward propagate |
| cdbmMeanfield | do mean-field updates to obtain the mean-field approximate posterior QMF |

* CRBM

1. 数据成员

|  |  |
| --- | --- |
| 变量名 | 功能 |
| scale | pooling factor |
| outputmaps | number of output maps |
| kernelsize | size of kernel |
| w | convolution kernel, size(h{i}{j}) = (kernelsize, kernelsize)  i: inputmaps  j: outputmaps |
| h | detection units, size(h{j}) = (w,h,n); j: outputmaps |
| p | pooling units, size(p{j}) = (w,h,n); j: outputmaps |
| c | biases of visible units |
| b | biases of hidden units |

1. 函数

|  |  |
| --- | --- |
| x = crbmUp(crbm, x) | do bottom-up train |
| x = crbmDown(crbm) | do up-down train |

* 传递公式







* 预训练阶段







V

V’

ph1

h1

ph2

h2

W1

W2

h1’

ph3

h3

h2’

W3

2W1

2W2

W3

W1

W2

2W3

Figure 网络参数的预训练过程

data

V

S

h1p+

h1+

Vp-

S

p+

p\_

V-

h1p-

正相

负相

Figure 第一层RBM训练规则(CD1)

V=data

h1p

h1

S

h2p+

h2+

h1p-

正相

负相

p\_

h2p-

h1-

Figure 第二层RBM参数训练情况(CD1)

* 均值场估计



V

h1

h1’

h2

h2’

h3

h3’

Figure 均值场参数更新关系

* 权值调整公式







* 概率最大池化



subject to , 

第k组隐含层中的元素接收来自层V的信号可以表示为：







权值更新







* algorithm structure

|  |
| --- |
| Algorithm learning a Deep Boltzmann Machine |
| 1: Given : a training set of N binary data vectors , M (the number of Markov chains), and K (the number of mean-field steps).  2: //Pretraining:  3: Pretraining parameters  of a DBM.  4: Initialize the recognition model  to the value of .  5: Randomly initialize M sample particles: , where .  6: for t = 0 to T (number of iterations) do  7: //Variational Inference:  8: for each training example vn , n = 1 to N do  9: In a single deterministic bottom-up pass, use the recognition model to  obtain the mean-field approximate posterior QMF.  10: Set  and run the mean-field updates for K steps to obtain the mean-field approximate posterior QMF.  11: Adjust the recognition parameters by taking a single gradient step:    12: Set  13: end for  14: //Stochastic Approximation:  15: for each sample m = 1 to M do  16: Sample  given  by running a Gibbs sampler.  17: end for  18: //Parameter Update:  19:  20:  21:  22: end for |

* 后期需要考虑的工作：

1. 对原始数据进行中心化，并采用随机抽样抽取样本；
2. 添加稀疏正则项；
3. 采用可变学习速率；
4. 在每一层中添加偏差项。

256\*256→252\*252→126\*126→122\*122→61\*61→56\*56→28\*28→类别标签

* 总结

1. 在预训练阶段，类别标签信息参与最后一个隐含层的预训练（其本身也被重建和参与计算）；
2. 在均值场训练阶段，类别标签信息参与均值场训练，先进行自底向上的均值场训练以产生所有的中间隐含层（正相），然后MF-N学习；
3. 在反向传播阶段，类别标签信息不参与均值场训练；
4. 在反向传播阶段结束后，使用共轭梯度法对各层权值和偏差值进行训练。需要注意的是，在均值场训练阶段，在自底向上训练时两倍化的权值也仍然参与共轭梯度的训练。
5. 输出映射和卷积核并不是一一对应的。如果L-1层的输出映射层数为K，L层的输出映射层数为M，则L层所拥有的卷积核的个数为K\*M。这在一定程度上丰富了模型的表达能力。

* 代码阅读总结

1. 权重初始用0.001作为系数；
2. 凡是计算激活概率之前，都需要用sigmoid函数对传入值进行处理；
3. 激活值只需要进行简单的累加；
4. 误差使用输入值和输入反馈值做差求得；
5. 权值增量采用惯性动量法；
6. 动量采用可变动量，在训练周期达到一定次数之后改变动量大小；
7. 权值调整时，正相权值和负相权值必须对batch数求平均；