

# 浓缩就是精华--SIGAI机器学习蓝宝书

原创： AI学习与实践平台



横屏观看哟

浓缩就是精华-SIGAI机器学习

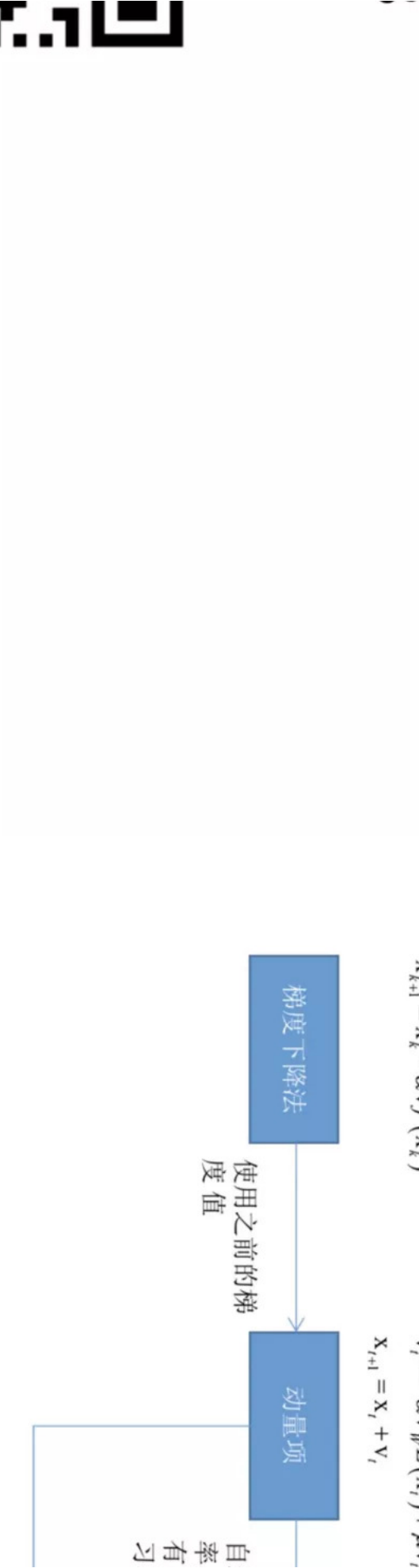
SIGAI-2018.08

[www.sigai.cn](http://www.sigai.cn)



SIGAI

# 学习蓝宝书



$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

$$V_t = -\alpha \nabla_w L(x_t) + \mu V_t$$
$$x_{t+1} = x_t + V_t$$

梯度下降法

动量项

使用之前的梯度值

自率有习

使用动:



$$(m_t)_i = \beta_1 (m_{t-1})_i + (1 - \beta_1) (g_t)_i$$
$$(v_t)_i = \beta_2 (v_{t-1})_i + (1 - \beta_2) (g_t)_i^2$$
$$(x_{t+1})_i = (x_t)_i - \alpha \frac{\sqrt{1 - (\beta_2)_i}}{1 - (\beta_1)_i} \frac{(m_t)_i}{\sqrt{(v_t)_i}} + \epsilon$$

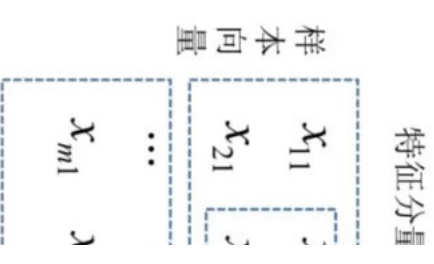
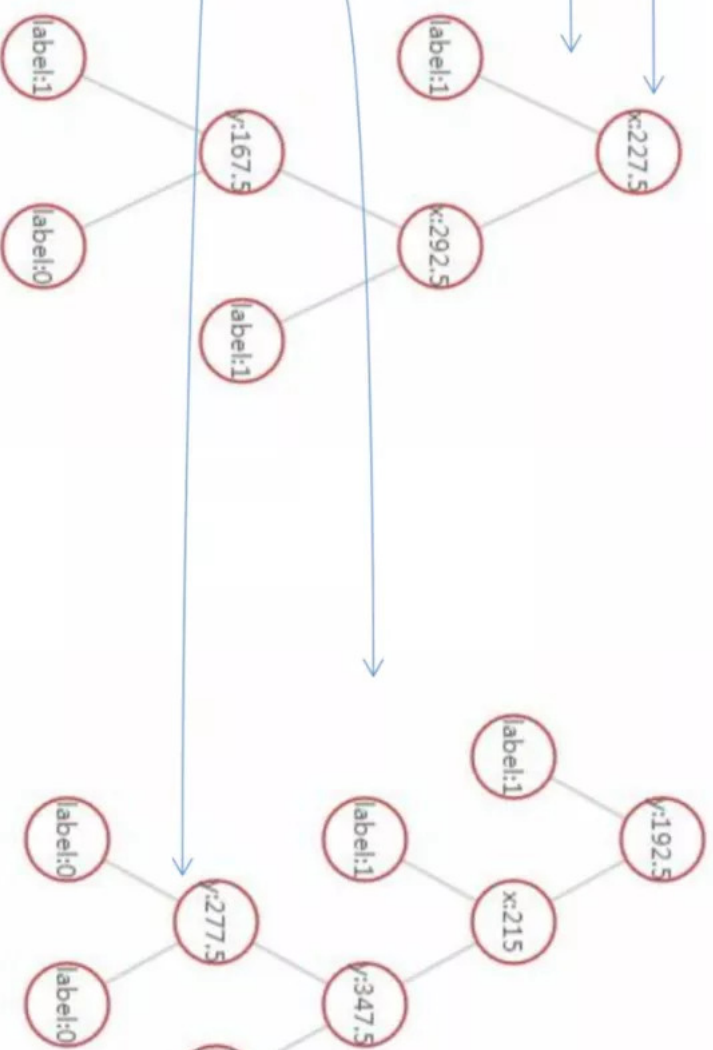


图 1

训练决策树的每个节点时，只用了随机抽取的部分特征分量

Bootstrap抽样，随机抽取出  
——部分样本训练决策树

东样本集



$$F(\mathbf{x}) = \sum_{t=1}^T \alpha_t f_t(\mathbf{x})$$

强分类器
弱分类器权重
弱分类器

样本权重初始化，所有样本权重均为  $1/N$   
 循环，依次训练每个弱分类器

训练一个弱分类器  $f_t(\mathbf{x})$ ，并计算它对训练样本集的错误率  $e_t$   
 计算弱分类器权重：

$$\alpha_t = \frac{1}{2} \log \left( (1 - e_t) / e_t \right)$$

更新所有样本的权重：

$$w'_t = w_t e^{-\gamma_t \alpha_t f_t(\mathbf{x}_i)} / Z_t$$

结束循环

得到强分类器： $\text{sgn}(F(\mathbf{x})) = \text{sgn} \left( \sum_{t=1}^T \alpha_t f_t(\mathbf{x}) \right)$

AdaBoost =  $\min_{\beta, f} \sum_{i=1}^I \exp(-\dots)$

迭代求解，每  
 $\min_{\beta, f} \sum_{i=1}^I w_i^{t-1} \exp$

分阶段优化，

弱分类器  
 正确分类的  
 错误分类的

定义加法模型+指数损失函数

$$y_i(F_{j-1}(x_i) + \beta f(x_i))$$

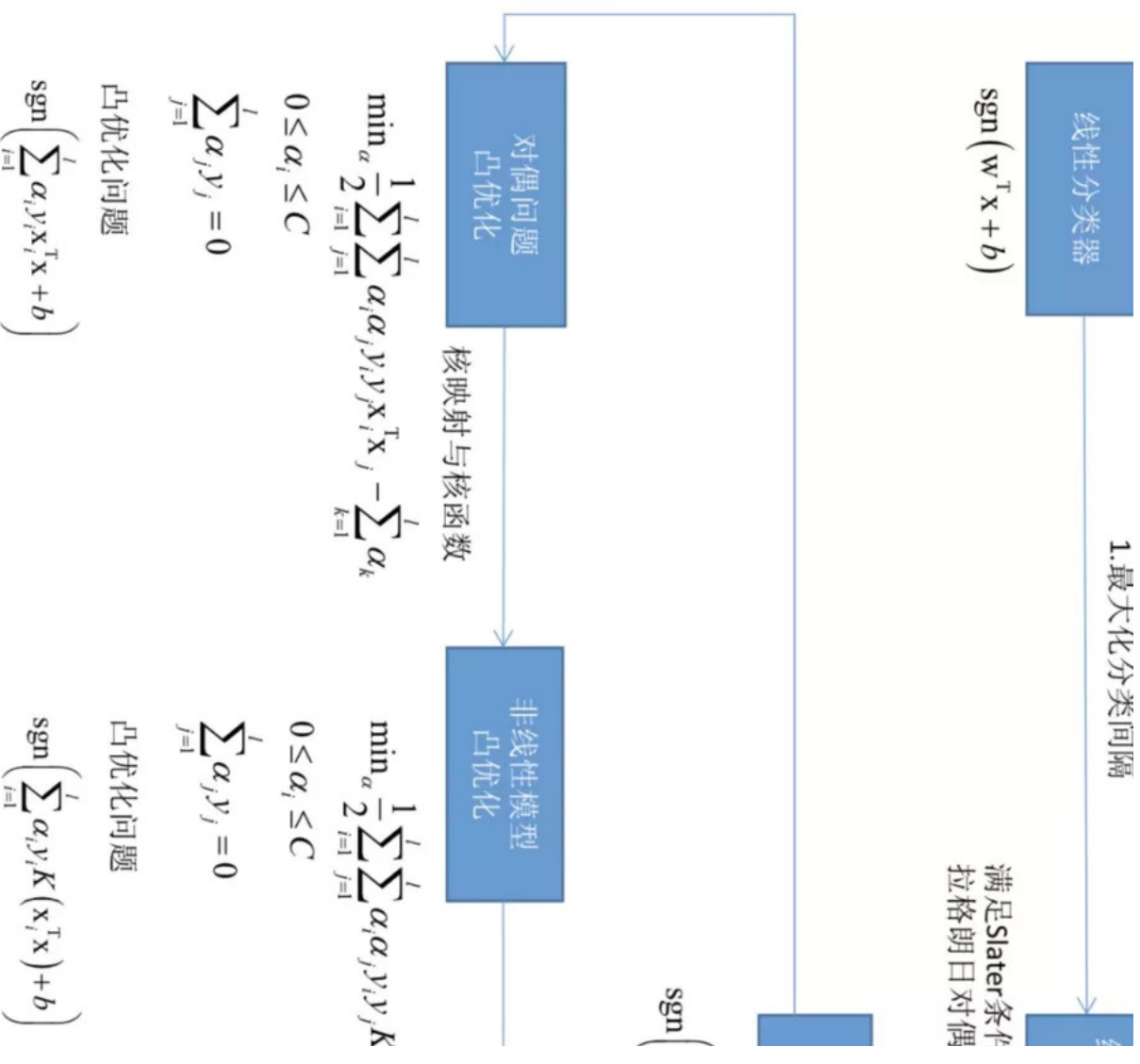
依次确定一个弱分类器及其权重

$$p(-\beta y_i f(x_i))$$

先优化弱分类器，然后确定其权重

样本权重是错误率的减函数

错误的样本权重会增加，  
正确的样本权重会减小



松弛变量和惩罚因子

线性可分的SVM  
凸优化

线性不可分的SVM  
凸优化

非, 强对偶成立

$$\min_w \frac{1}{2} w^T w$$

$$y_i (w^T x_i + b) \geq 1$$

$$\min_w \frac{1}{2} w^T w + C \sum_{i=1}^l \xi_i$$

$$y_i (w^T x_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0, i = 1, \dots, l$$

对偶问题  
凸优化

$$\sum_{i=1}^l \alpha_i y_i x_i^T x + b$$

$$\sum_{i=1}^l \alpha_i y_i = 0$$

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{k=1}^l \alpha_k$$

满足Slater条件, 强对偶成立  
拉格朗日对偶

SMO算法

$$(\mathbf{x}_i^T \mathbf{x}_j) - \sum_{k=1}^l \alpha_k$$

子问题也是凸优化, 求公式解, 二元二次函数极值

$$f(\alpha_i, \alpha_j) = \frac{1}{2} K_{ii} \alpha_i^2 + \frac{1}{2} K_{jj} \alpha_j^2 + s K_{ij} \alpha_i \alpha_j +$$

$$y_i y_j \alpha_i + y_j y_j \alpha_j - \alpha_i - \alpha_j + c$$

$$0 \leq \alpha_i \leq C$$

$$0 \leq \alpha_j \leq C$$

$$y_i \alpha_i + y_j \alpha_j = \xi$$

优化变量的选择-KKT条件

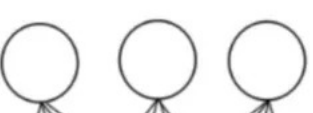
$$\alpha_i = 0 \Leftrightarrow y_i g(\mathbf{x}_i) \geq 1$$

$$0 < \alpha_i < C \Leftrightarrow y_i g(\mathbf{x}_i) = 1$$

$$\alpha_i = C \Leftrightarrow y_i g(\mathbf{x}_i) \leq 1$$

$$g(\mathbf{x}_i) = \sum_{j=1}^l \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) + b$$

正向传  
成的变



反向传  
的递封

$\delta^{(l)}$

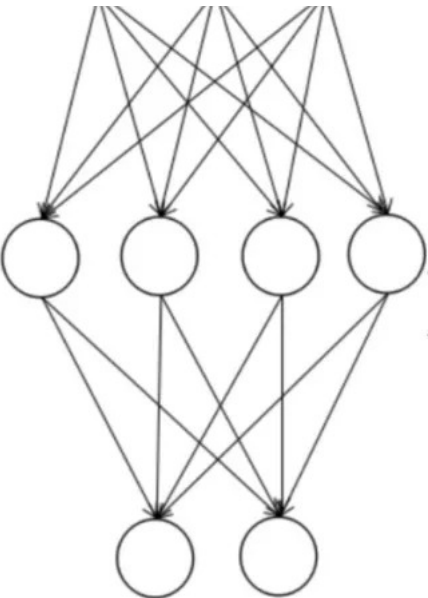


## 训练算法 = 复合函数求导 + 梯度下降法

传播时，从输入层开始，逐层向后计算，每个层完  
换为：

$$\mathbf{u}^{(l)} = \mathbf{W}^{(l)} \mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}$$

$$\mathbf{x}^{(l)} = f(\mathbf{u}^{(l)})$$



传播时，从输出层开始，逐层向前计算，误差项  
佳计算公式为：

$$\nabla_{\mathbf{u}^{(l)}} L = \begin{cases} \left( \mathbf{x}^{(l)} - \mathbf{y} \right) \odot f'(\mathbf{u}^{(l)}) & l = n_l \\ \left( \mathbf{W}^{(l+1)} \right)^T \left( \delta^{(l+1)} \right) \odot f'(\mathbf{u}^{(l)}) & l \neq n_l \end{cases}$$

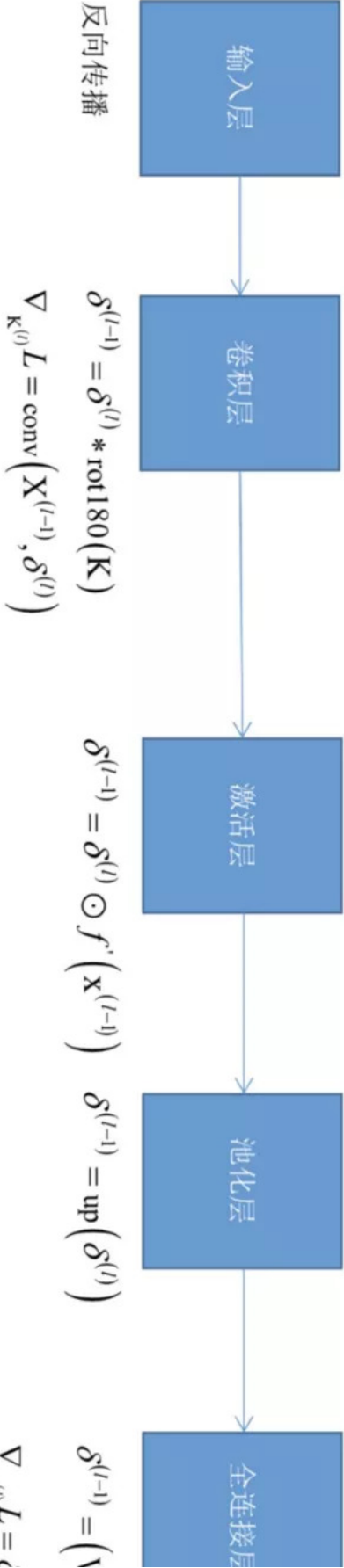
根据误差项可以计算出对权重，偏置的梯度值，  
然后用梯度下降法更新：

$$\begin{aligned} \nabla_{\mathbf{w}^{(l)}} L &= \delta^{(l)} \left( \mathbf{x}^{(l-1)} \right)^T \\ \nabla_{\mathbf{b}^{(l)}} L &= \delta^{(l)} \end{aligned}$$



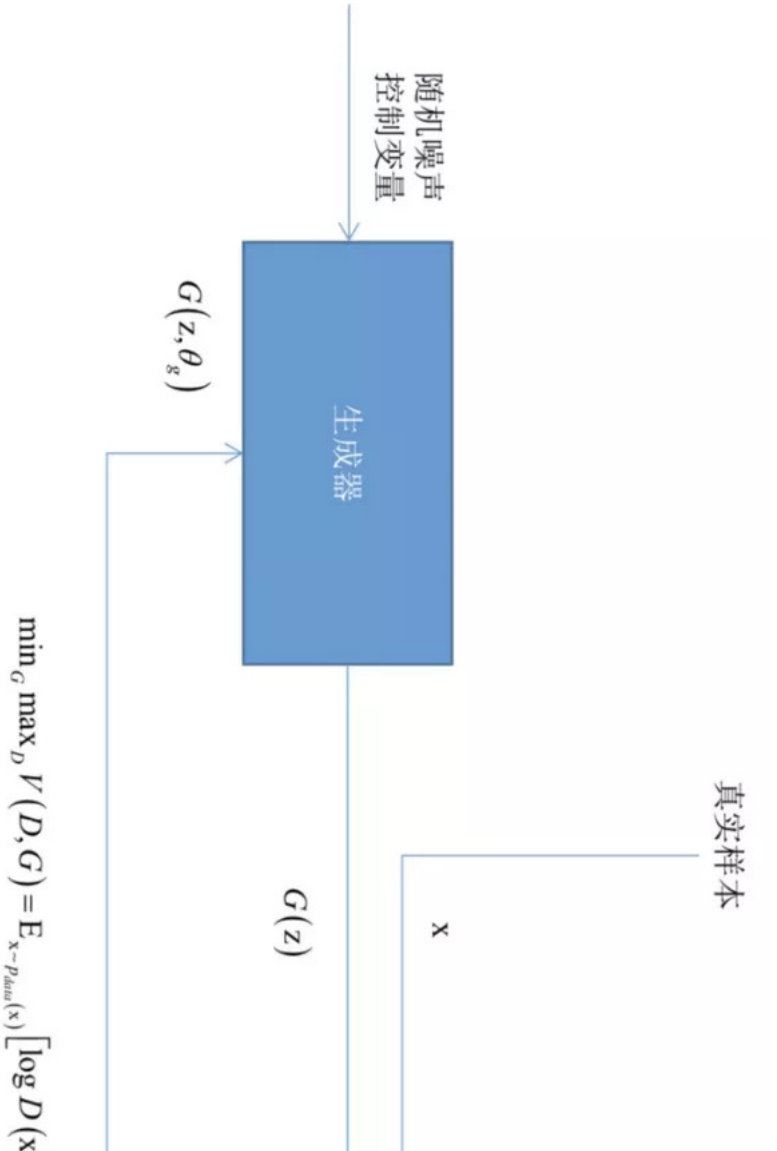
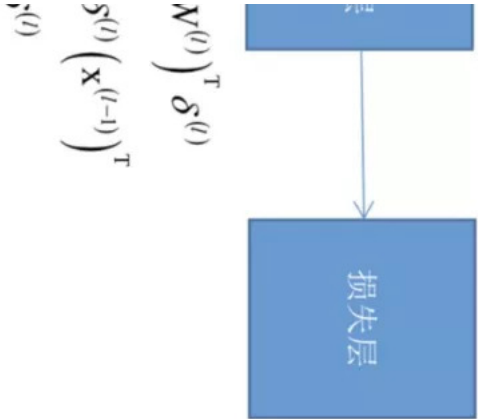
正向传播

$$x_{ij}^{(l)} = \sum_{p=1}^s \sum_{q=1}^s x_{i+p-1, j+q-1}^{(l-1)} \times k_{pq}^{(l)} + b_j^{(l)} \quad \mathbf{x}^{(l)} = f\left(\mathbf{x}^{(l-1)}\right) \quad \mathbf{X}^{(l)} = \text{down}\left(\mathbf{X}^{(l-1)}\right) \quad \mathbf{u}^{(l)} = \mathbf{W}^{(l)}$$

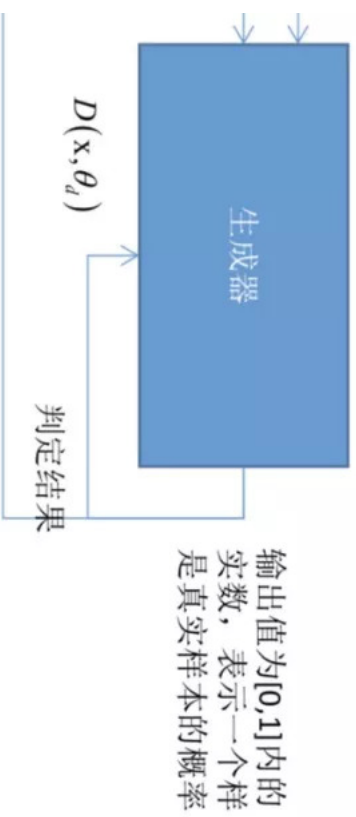


反向传播时，如果本层有参数，则根据从后一层传入的误差项计算损失函数对本层参数的梯度值  
 然后根据从后一层传入的误差项计算本层的误差项，并传播到前一层  
 后层传入的误差项，是损失函数对本层输出值的梯度值；本层的误差项，是损失函数对本层输入值的梯度值

$$\mathbf{x}^{(l-1)} + \mathbf{b}^{(l)}$$



判别模型要正确  
即真实样本要被  
器生成的样本尽  
最大化，这意味  
生成器要让它生  
即  $\log(1 - D(G(z)))$   
接近于1



$$)] + E_{z \sim p_z(z)} [\log (1 - D(G(z)))]$$

均区分真实样本与生成器生成的样本，判定为正样本， $\log D(x)$ 要最大化；生成量被判定为负样本，即  $\log (1 - D(G(z)))$  着  $D(G(z))$  最小化即接近于0

或的模型被判定为正样本的概率最大化， $D(G(z))$ 最小化，这意味着  $D(G(z))$ 最大化即

- 2.在  $\theta_1$  处，下函数和上函数相等
- 1.用参数的当前估计值构造下界函数

$$\sum_{i=1}^I \ln \sum_{z_i} p(x_i, z_i; \theta)$$

蓝色为目标函数，即对数似然函数。由于隐变量 $z$ 的存在，对数函数内部有求和项，不易求解

5. 在 $\theta_{t+1}$ 点处，新的下界函数和目標函数数值相等

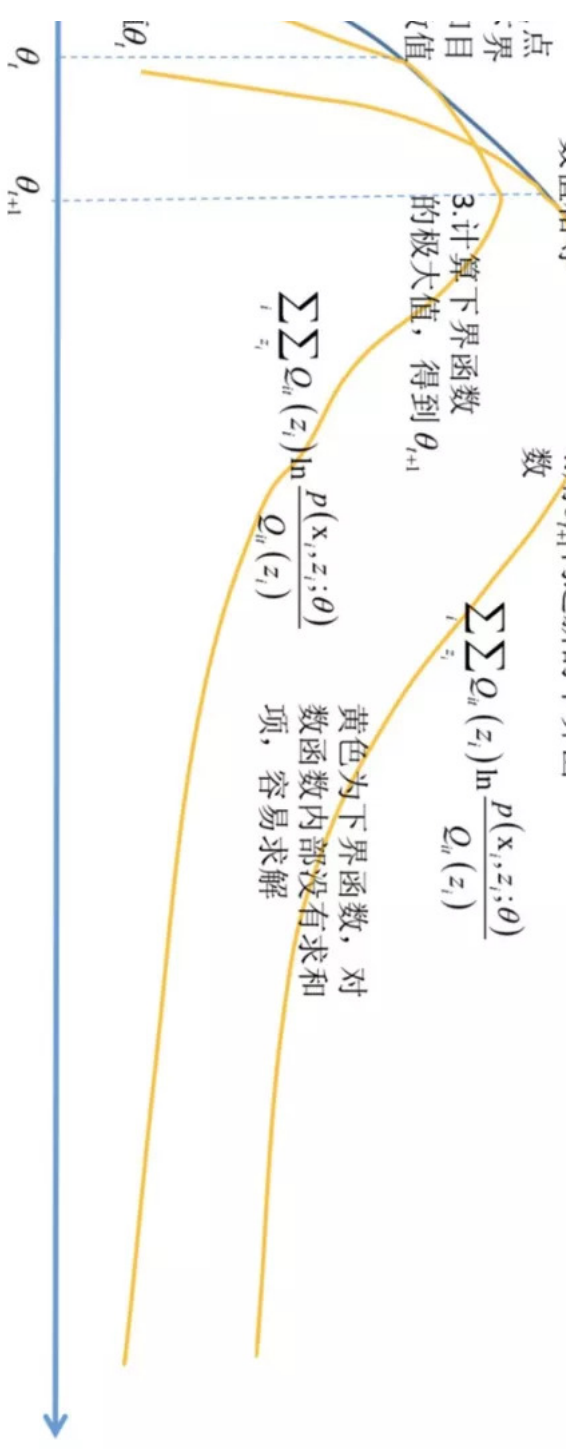
4. 用 $\theta_{t+1}$ 构造新的下界函数

$$\sum_i \sum_{z_i} Q_{ii}(z_i) \ln \frac{p(x_i, z_i; \theta)}{Q_{ii}(z_i)}$$

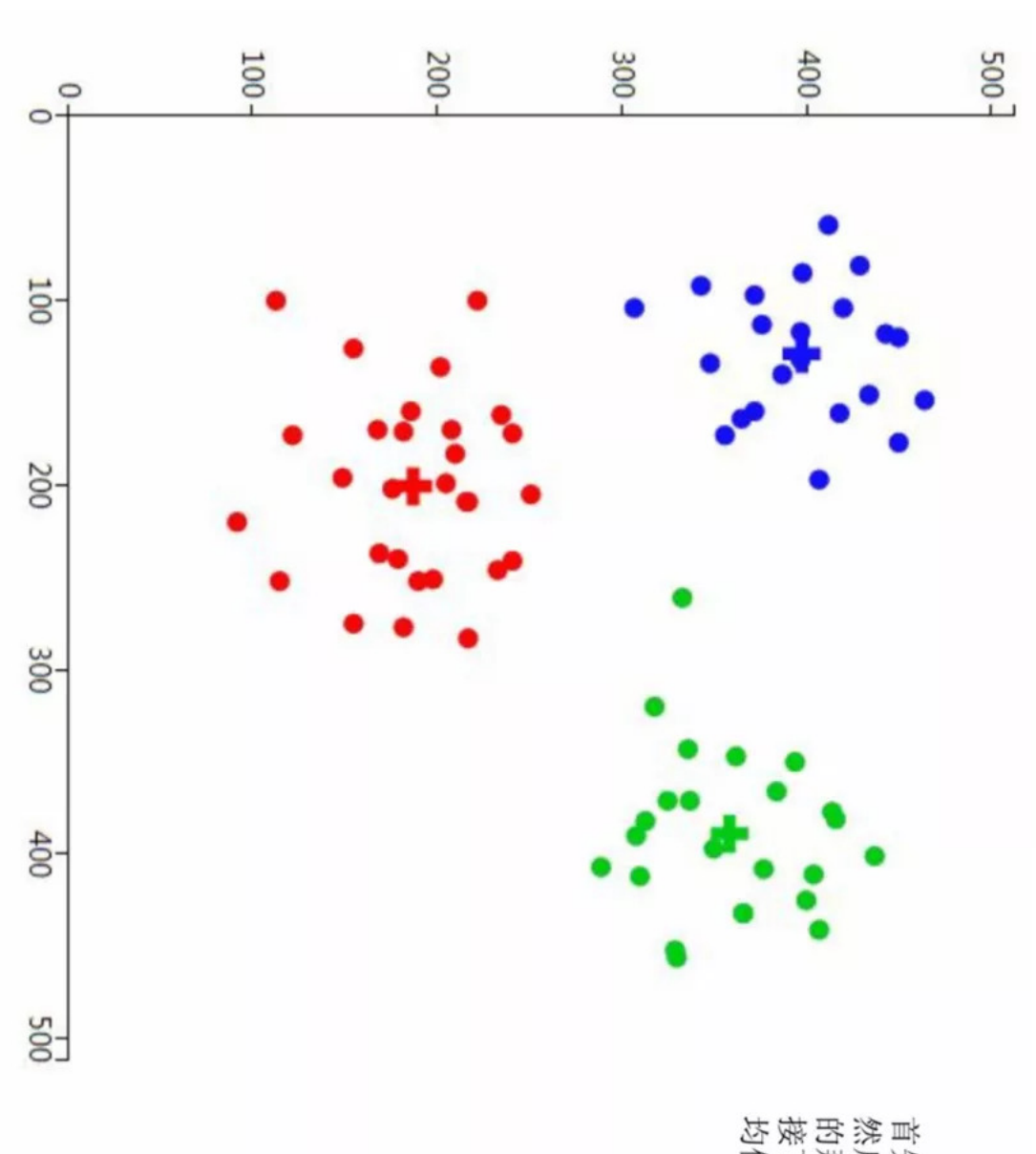
3. 计算下界函数的极大值，得到 $\theta_{t+1}$

$$\sum_i \sum_{z_i} Q_{ii}(z_i) \ln \frac{p(x_i, z_i; \theta)}{Q_{ii}(z_i)}$$

黄色为下界函数，对数函数内部没有求和项，容易求解



首先随机的生成每个类的类中心，  
然后计算每个样本离每个类中心的距  
的那个类  
接下来根据分配方案重新计算类中心  
均值



0左图中的+  
距离，将其分配到距离最近  
心，即本类所有样本向量的



科普类

【获取码】SIGAI0413

[机器学习——波澜壮阔四十年](#)

【获取码】SIGAI0620

[理解计算：从√2到AlphaGo ——第1季 从√2谈起](#)

【获取码】SIGAI0704

[理解计算：从√2到AlphaGo ——第2季 神经计算的历史背景](#)

【获取码】SIGAI0713

[理解计算：从√2到AlphaGo ——第3季 神经计算的数学模型](#)

【获取码】SIGAI0815

[理解计算：从√2到AlphaGo ——第4季 凛冬将至](#)

【获取码】SIGAI0802

[机器学习和深度学习中值得弄清楚的一些问题](#)



数学类

【获取码】SIGAI0417

[学好机器学习需要哪些数学知识](#)

【获取码】SIGAI0511

[理解梯度下降法](#)

【获取码】SIGAI0518

[理解凸优化](#)

【获取码】SIGAI0531

[理解牛顿法](#)



机器学习类

【获取码】SIGAI0428

[用一张图理解SVM的脉络](#)

【获取码】SIGAI0505

[理解神经网络的激活函数](#)

【获取码】SIGAI0522

【实验】[理解SVM核函数和参数的作用](#)

【获取码】SIGAI0601

【群话题精华】[五月集锦—机器学习和深度学习中一些值得思考的问题](#)

【获取码】SIGAI0602

[大话AdaBoost算法](#)

【获取码】SIGAI0606

[理解主成分分析（PCA）](#)

【获取码】SIGAI0611

[理解决策树](#)

【获取码】SIGAI0613

[用一句话总结常用的机器学习算法](#)

【获取码】SIGAI0618

[理解过拟合](#)

【获取码】SIGAI0627

[k近邻算法](#)

【获取码】SIGAI0704

[机器学习算法地图](#)

【获取码】SIGAI0706

[反向传播算法推导—全连接神经网络](#)

【获取码】SIGAI0711

[如何成为一名优秀的算法工程师](#)

【获取码】SIGAI0723

[流形学习概述](#)

【获取码】SIGAI0725

[随机森林概述](#)

深度学习类

【获取码】SIGAI0426

[卷积神经网络为什么能够称霸计算机视觉领域？](#)

【获取码】SIGAI0508

[深度卷积神经网络演化历史及结构改进脉络-40页长文全面解读](#)

【获取码】SIGAI0515

[循环神经网络综述—语音识别与自然语言处理的利器](#)

【获取码】SIGAI0625

[卷积神经网络的压缩与加速](#)

【获取码】SIGAI0709

[生成式对抗网络模型综述](#)

【获取码】SIGAI0718

[基于深度负相关学习的人群计数方法](#)

【获取码】SIGAI0723



[关于感受野的总结](#)

【获取码】SIGAI0806

[反向传播算法推导——卷积神经网络](#)

【获取码】SIGAI0810

[理解Spatial Transformer Networks](#)

## 机器视觉类

【获取码】SIGAI0420

[人脸识别算法演化史](#)

【获取码】SIGAI0424

[基于深度学习的目标检测算法综述](#)

【获取码】SIGAI0503

[人脸检测算法综述](#)

【获取码】SIGAI0525

[【SIGAI综述】行人检测算法](#)

【获取码】SIGAI0604

[FlowNet到FlowNet2.0：基于卷积神经网络的光流预测算法](#)

【获取码】SIGAI0608

[人体骨骼关键点检测综述](#)

【获取码】SIGAI0615

[目标检测算法之YOLO](#)

【获取码】SIGAI0622

[场景文本检测——CTPN算法介绍](#)

【获取码】SIGAI0629

[自然场景文本检测识别技术综述](#)

【获取码】SIGAI0716

[人脸检测算法之S3FD](#)

【获取码】SIGAI0727

[基于内容的图像检索技术综述——传统经典方法](#)

【获取码】SIGAI0817

[基于内容的图像检索技术综述——CNN方法](#)

自然语言处理

【获取码】SIGAI0803

[基于深度神经网络的自动问答概述](#)

【获取码】SIGAI0820

[文本表示简介](#)

工业应用类

【获取码】SIGAI0529

[机器学习在自动驾驶中的应用-以百度阿波罗平台为例【上】](#)

[本文为SIGAI原创](#)

[如需转载，欢迎发消息到本订阅号](#)