-机器学习和深度学习中值得弄清楚的一些问题

SIGAI 飞跃计划答疑精华问题汇总

SIGAI

2018.8.2

SIGAI 飞跃计划第一期已经进行 4 周了，在这 4 周的学习中，同学们提出了不少好问题。在这里，我们将每周直播答疑的问题进行筛选和整理，写成今天的公众号文章，供大家参考。相信会对大家的学习和实践有所帮助！

问题 1：线性回归的损失函数是凸函数的证明

假设有 l 个训练样本，特征向量为 x_i，标签值为 y_i，这里使用均方误差（MSE），线性回归训练时优化的目标为：

$$L=\frac{1}{2l}\sum_{i=1}^{l}\left(\mathrm{w}^{\mathrm{T}}\mathrm{x}_i-y_i\right)^2$$

损失函数对权重向量 w 的一阶偏导数为：

$$\frac{\partial L}{\partial w_i}=\frac{1}{2l}\sum_{k=1}^{l}2\left(\mathrm{w}^{\mathrm{T}}\mathrm{x}_k-y_k\right)\frac{\partial \mathrm{w}^{\mathrm{T}}\mathrm{x}_k}{\partial w_i}=\frac{1}{l}\sum_{k=1}^{l}\left(\mathrm{w}^{\mathrm{T}}\mathrm{x}_k-y_k\right)x_{ki}$$

损失函数对权重向量 w 的二阶偏导数为：

$$\frac{\partial^2 L}{\partial w_i\partial w_j}=\frac{\partial\frac{1}{l}\sum_{k=1}^{l}\left(\mathrm{w}^{\mathrm{T}}\mathrm{x}_k-y_k\right)x_{ki}}{\partial w_j}=\frac{\partial\frac{1}{l}\sum_{k=1}^{l}\mathrm{w}^{\mathrm{T}}\mathrm{x}_k x_{ki}}{\partial w_j}=\frac{\partial\frac{1}{l}\sum_{k=1}^{l}\left(\sum_{p=1}^{n}w_p x_{kp}\right)x_{ki}}{\partial w_j}$$

$$=\frac{1}{l}\sum_{k=1}^{l}x_{kj}x_{ki}$$

因此目标函数的 Hessian 矩阵为：

$$\frac{1}{l}\sum_{k=1}^{l}\begin{bmatrix} x_{k1}x_{k1} & \cdots & x_{k1}x_{kn} \\ \cdots & \cdots & \cdots \\ x_{kn}x_{k1} & \cdots & x_{kn}x_{kn} \end{bmatrix}=\frac{1}{l}\begin{bmatrix} \sum_{k=1}^{l}x_{k1}x_{k1} & \cdots & \sum_{k=1}^{l}x_{k1}x_{kn} \\ \cdots & \cdots & \cdots \\ \sum_{k=1}^{l}x_{kn}x_{k1} & \cdots & \sum_{k=1}^{l}x_{kn}x_{kn} \end{bmatrix}$$

写成矩阵形式为：

$$\frac{1}{l}\begin{bmatrix} \mathrm{x}_1^{\mathrm{T}} \\ \cdots \\ \mathrm{x}_l^{\mathrm{T}} \end{bmatrix}\begin{bmatrix} \mathrm{x}_1 & \cdots & \mathrm{x}_l \end{bmatrix}=\frac{1}{l}\mathrm{X}^{\mathrm{T}}\mathrm{X}$$

其中 X 是所有样本的特征向量按照列构成的矩阵。对于任意不为 0 的向量 x，有：

$$\mathrm{x}^{\mathrm{T}}\mathrm{X}^{\mathrm{T}}\mathrm{X}\mathrm{x}=\left(\mathrm{X}\mathrm{x}\right)^{\mathrm{T}}\left(\mathrm{X}\mathrm{x}\right)\geq 0$$

因此 Hessian 矩阵半正定，目标函数是凸函数。

问题 2：L1 和 L2 正则化的选定标准？

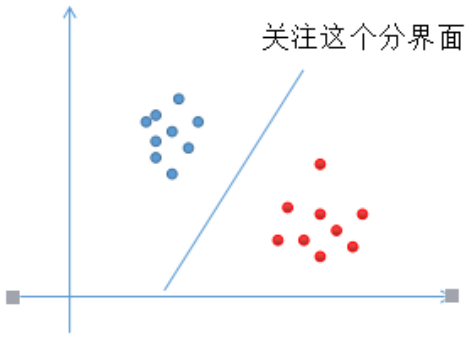
这个问题没有理论上的定论。在神经网络中我们一般选择 L2 正则化。以线性回归为例，使用 L2 正则化的岭回归和和使用 L1 正则化的 LASSO 回归都有应用。如果使用 L2 正则化，则正则化项的梯度值为 w；如果是 L1 正则化，则正则化项的梯度值为 sgn(w)。一般认为，L1 正则化的结果更为稀疏。可以证明，两种正则化项都是凸函数。

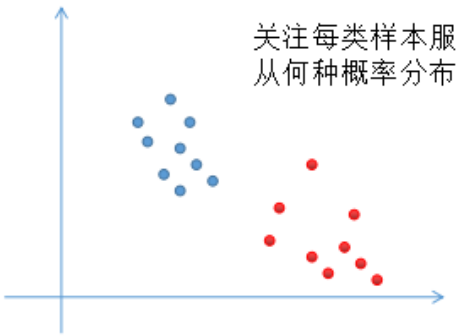
问题 3：什么时候用朴素贝叶斯，什么时候用正态贝叶斯？

一般我们都用朴素贝叶斯，因为它计算简单。除非特征向量维数不高、特征分量之间存在严重的相关性我们才用正态贝叶斯，如果特征向量是 n 维的，正态贝叶斯在训练时需要计算 n 阶矩阵的逆矩阵和行列式，这非常耗时。

问题 4：可否请雷老师讲解一下 discriminative classifier 和 generative classifier 的异同?

判别模型直接得到预测函数 f(x)，或者直接计算概率值 p(y|x)，比如 SVM 和 logistic 回归，softmax 回归。SVM 直接得到分类超平面的方程，logistic 回归和 softmax 回归，以及最后一层是 softmax 层的神经网络，直接根据输入向量 x 得到它属于每一类的概率值 p(y|x)。判别模型只关心决策面，而不管样本的概率分布。生成模型计算 p(x, y)或者 p(x|y) ，通俗来说，生成模型假设每个类的样本服从某种概率分布，对这个概率分布进行建模。





问题 5：雷老师下回可以分享一下自己的学习方法吗？ 机器学习的内容又多又难，涉

及理论与实践，很容易碰到问题卡壳的情况

首先要确定：卡壳在什么地方？数学公式不理解？算法的思想和原理不理解？还是算法的实现细节不清楚？

如果是数学知识欠缺，或者不能理解，需要先去补数学。如果是对机器学习算法本身使用的思想，思路不理解，则重点去推敲算法的思路。如果是觉得算法太抽象，则把算法形象化，用生动的例子来理解，或者看直观的实验结果。配合实验，实践，能更清楚的理解算法的效果，实现，细节问题。

问题 6：流形学习，拉普拉斯特征映射，证明拉普拉斯矩阵半正定

假设 L 是图的拉普拉斯矩阵，D 是加权度对角矩阵，W 是邻接矩阵。对于任意不为 0 的向量 f，有：

$$
\begin{aligned}
\mathrm{f}^{\mathrm{T}}\mathrm{Lf} &= \mathrm{f}^{\mathrm{T}}\mathrm{Df} - \mathrm{f}^{\mathrm{T}}\mathrm{Wf} \\
&= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j \\
&= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n} d_j f_j^2 \right) \\
&= \frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n}\sum_{i=1}^{n} w_{ji} f_j^2 \right) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(w_{ij} f_i^2 - 2 w_{ij} f_i f_j + w_{ji} f_j^2 \right) \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\left(f_i - f_j \right)^2 \geq 0
\end{aligned}
$$

因此拉普拉斯矩阵半正定。这里矩阵 D 的对角线元素是矩阵 W 的每一行元素的和。

问题 7：线性判别分析：优化目标有冗余，这个冗余怎么理解呢？

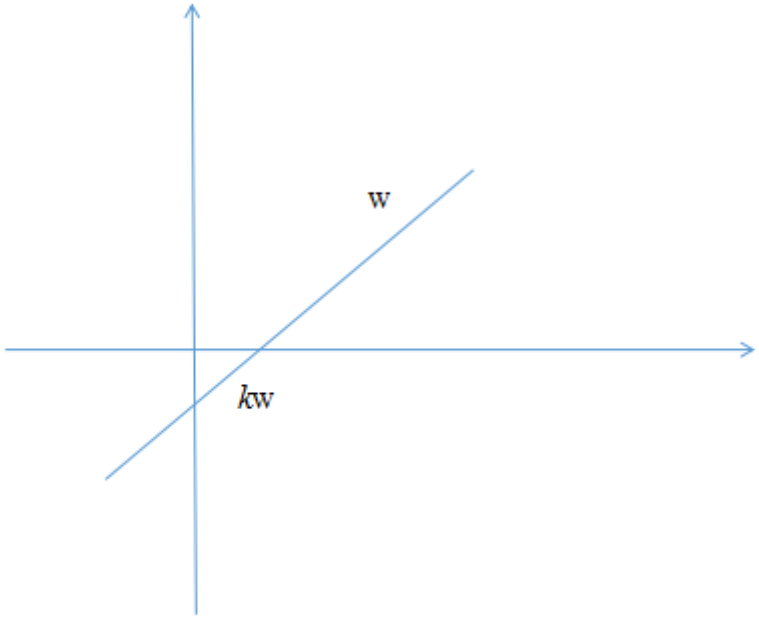
线性判别分析优化的目标函数为：

$$
L(\mathrm{w}) = \frac{\mathrm{w}^{\mathrm{T}}\mathrm{S}_B\mathrm{w}}{\mathrm{w}^{\mathrm{T}}\mathrm{S}_W\mathrm{w}}
$$

如果向量 w 是最优解，则将其乘以不为 0 的系数 k 之后，向量 kw 仍然是最优解，证明如下：

$$
L(k\mathrm{w}) = \frac{(k\mathrm{w})^{\mathrm{T}}\,\mathrm{S}_B\,(k\mathrm{w})}{(k\mathrm{w})^{\mathrm{T}}\,\mathrm{S}_W\,(k\mathrm{w})} = \frac{k^2\mathrm{w}^{\mathrm{T}}\mathrm{S}_B\mathrm{w}}{k^2\mathrm{w}^{\mathrm{T}}\mathrm{S}_W\mathrm{w}} = \frac{\mathrm{w}^{\mathrm{T}}\mathrm{S}_B\mathrm{w}}{\mathrm{w}^{\mathrm{T}}\mathrm{S}_W\mathrm{w}}
$$

从几何上看，w 可 kw 这两个向量表示的是一个方向，如果 w 是最佳投影方向，则 kw 还是这个方向：



问题 8：决策树，如果是回归树，在寻找最佳分裂时的标准

对于回归树，寻找最佳分裂的标准是分裂之后的回归误差最小化。这等价于让分裂之前的回归误差减去分裂之后的回归误差最大化：

$$E = E(D) - \frac{N_L}{N}E(D_L) - \frac{N_R}{N}E(D_R)$$

展开之后为：

$$E = \frac{1}{N}\left(\sum_{i=1}^{N} y_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} y_i\right)^2\right) - \frac{N_L}{N}\left(\frac{1}{N_L}\left(\sum_{i=1}^{N_L} y_i^2 - \frac{1}{N_L}\left(\sum_{i=1}^{N_L} y_i\right)^2\right)\right) -$$

$$\frac{N_R}{N}\left(\frac{1}{N_R}\left(\sum_{i=1}^{N_R} y_i^2 - \frac{1}{N_R}\left(\sum_{i=1}^{N_R} y_i\right)^2\right)\right)$$

$$= -\frac{1}{N^2}\left(\sum_{i=1}^{N} y_i\right)^2 + \frac{1}{N}\left(\frac{1}{N_L}\left(\sum_{i=1}^{N_l} y_i\right)^2 + \frac{1}{N_R}\left(\sum_{i=1}^{N_R} y_i\right)^2\right)$$

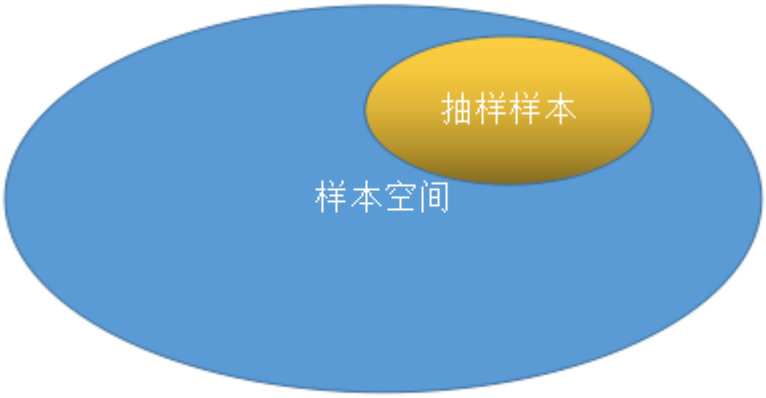
由于前面的都是常数，因此这等价于将下面的值最大化：

$$E = \frac{1}{N_L}\left(\sum_{i=1}^{N_l} y_i\right)^2 + \frac{1}{N_R}\left(\sum_{i=1}^{N_R} y_i\right)^2$$

具体细节可以参考 SIGAI 之前的公众号文章“理解决策树”。

问题 9：抽样误差是怎么判定的？能否消除抽样误差？

只要抽样的样本不是整个样本空间，理论上就会有抽样误差，只是是否严重而已。对于一个一般性的数据集，无法从理论上消除抽样误差。在机器学习中，我们无法得到所有可能的训练样本，只能从中抽取一部分，一般要让样本尽量有代表性、全面。



问题 10：卷积神经网络中的 w 到底是怎么更新的，我知道利用梯度下降法和误差函数可以更新 w 值，但是对具体更新的过程还不是很理解。比如每次怎么调整，是一层一层调整还是整体调整，调整的结果是遵循最小化误差函数，但是过程中怎么能体现出来

反向传播时对每一层计算出参数梯度值之后立即更新；所有层都计算出梯度值之后一起更新，这两种方式都是可以的。所有层的参数都按照梯度下降法更新完一轮，才算一次梯度下降法迭代。

OpenCV 的实现，train_backprop 函数

```
// _dw = a * x1 * grad1 + b * _dw，计算权重的更新量
// 在这里考虑了动量项
 cvGEMM( x1, grad1, params.bp_dw_scale, &_dw, params.bp_moment_scale,
     &_dw );
// _w = _w + _dw，更新权重
cvAdd( &_w, &_dw, &_w );
 if( i > 1 ) // 如果不是输入层，传播误差
 {
          grad2->cols = n1;
          _w.rows = n1;
          // 计算 grad2 =   grad1*_w^T，即
          cvGEMM( grad1, &_w, 1, 0, 0, grad2, CV_GEMM_B_T );
 }
```

Caffe 的实现

```
void SGDSolver<Dtype>::ApplyUpdate() {
    Dtype rate = GetLearningRate();
    if (this->param_.display() && this->iter_ % this->param_.display() == 0) {
        LOG_IF(INFO, Caffe::root_solver()) << "Iteration " << this->iter_
        << ", lr = " << rate;
    }
    ClipGradients();
    for (int param_id = 0; param_id < this->net_->learnable_params().size();
        ++param_id) {
        Normalize(param_id);
        Regularize(param_id);
        ComputeUpdateValue(param_id, rate);
    }
    this->net_->Update();
}
```

问题 11：对于凸优化问题的理解，我自己感觉这个很难实现，首先实际问题中有许多问题是不知道约束问题和目标函数的，不知道是不是我做的图像识别的问题，我之前对于目标函数的认识就是使用 softmax 的交叉损失函数，这里可能是我自己的理解不够吧，还需要老师给点提示。

所有机器学习算法的优化目标函数都是确定的，如果带有约束条件，约束条件也是确定的，不会存在不知道目标函数和约束条件的算法

问题 12：如何选择机器学习算法是映射函数 f(x)

映射函数的选取没有一个严格的理论。神经网络，决策树可以拟合任意目标函数，但决策树在高维空间容易过拟合，即遇到维数灾难问题。神经网络的结构和激活函数确定之后，

通过调节权重和偏置项可以得到不同的函数。决策树也是如此，不同的树结构代表不同的函数，而在训练开始的时候我们并不知道函数具体是什么样子的。其他的算法，函数都是确定的，如 logistic 回归，SVM，我们能调节的只有它们的参数。每类问题我们都要考虑精度，速度来选择适合它的函数。

问题 13：梯度下降法的总结

1.为什么需要学习率？保证泰勒展开在 x 的邻域内进行，从而可以忽略高次项。

2.只要没有到达驻点，每次迭代函数值一定能下降，前提是学习率设置合理。

3.迭代终止的判定规则。达到最大迭代次数，或者梯度充分接近于 0。

4.只能保证找到梯度为 0 的点，不能保证找到极小值点，更不能保证找到全局极小值点

梯度下降法的改进型，本质上都只用了梯度即一阶导数信息，区别在于构造更新项的公式不同。

问题 14：牛顿法的总结

1.不能保证每次迭代函数值下降.

2.不能保证收敛。

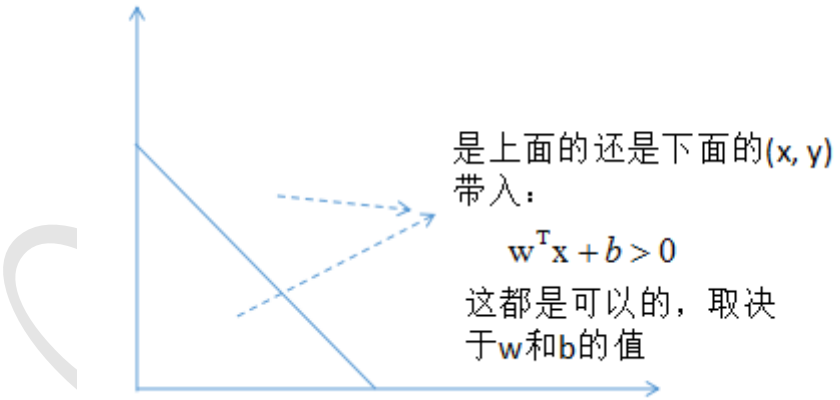
3.学习率的设定-直线搜索。

4.迭代终止的判定规则。达到最大迭代次数，或者梯度充分接近于 0。

5.只能保证找到梯度为 0 的点，不能保证找到极小值点，更不能保证找到全局极小值点

问题 15：为什么不能用斜率截距式的方程？

无法表达斜率为正无穷的情况-垂直的直线。直线方程两边同乘以一个不为 0 的数，还是同一条直线。



问题 16：神经网络的正则化项和动量项的比较

正则化项的作用：缓解过拟合，迫使参数尽可能小。以 L2 正则化为例：

$$L(w) = \frac{1}{N}\sum_{i=1}^{N} L(w, x_i, y_i) + \lambda w^T w$$

动量项的作用：加速收敛，减少震荡。计算公式为：

$$V_{t+1} = -\alpha \nabla_W L(W_t) + \mu V_t$$

$$W_{t+1} = W_t + V_{t+1}$$

这相当于累积了之前的梯度信息，并且呈指数级衰减。实现时，先加正则化项，计算动量项。

推荐阅读

[1] 机器学习-波澜壮阔 40 年 SIGAI 2018.4.13.
[2] 学好机器学习需要哪些数学知识？SIGAI 2018.4.17.
[3] 人脸识别算法演化史 SIGAI 2018.4.20.
[4] 基于深度学习的目标检测算法综述 SIGAI 2018.4.24.
[5] 卷积神经网络为什么能够称霸计算机视觉领域？ SIGAI 2018.4.26.
[6] 用一张图理解 SVM 的脉络 SIGAI 2018.4.28.
[7] 人脸检测算法综述 SIGAI 2018.5.3.
[8] 理解神经网络的激活函数 SIGAI 2018.5.5.
[9] 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 SIGAI 2018.5.8.
[10] 理解梯度下降法 SIGAI 2018.5.11.
[11] 循环神经网络综述—语音识别与自然语言处理的利器 SIGAI 2018.5.15
[12] 理解凸优化 SIGAI 2018.5.18
[13] 【实验】理解 SVM 的核函数和参数 SIGAI 2018.5.22
[14] 【SIGAI 综述】行人检测算法 SIGAI 2018.5.25
[15] 机器学习在自动驾驶中的应用—以百度阿波罗平台为例(上) SIGAI 2018.5.29
[16] 理解牛顿法 SIGAI 2018.5.31
[17] 【群话题精华】5 月集锦—机器学习和深度学习中一些值得思考的问题 SIGAI 2018.6.1
[18] 大话 Adaboost 算法 SIGAI 2018.6.2
[19] FlowNet 到 FlowNet2.0：基于卷积神经网络的光流预测算法 SIGAI 2018.6.4
[20] 理解主成分分析(PCA) SIGAI 2018.6.6
[21] 人体骨骼关键点检测综述 SIGAI 2018.6.8
[22] 理解决策树 SIGAI 2018.6.11
[23] 用一句话总结常用的机器学习算法 SIGAI 2018.6.13
[24] 目标检测算法之 YOLO SIGAI 2018.6.15
[25] 理解过拟合 SIGAI 2018.6.18
[26] 理解计算：从√2 到 AlphaGo ——第 1 季 从√2 谈起 SIGAI 2018.6.20
[27] 场景文本检测——CTPN 算法介绍 SIGAI 2018.6.22
[28] 卷积神经网络的压缩和加速 SIGAI 2018.6.25
[29] k 近邻算法 SIGAI 2018.6.27
[30] 自然场景文本检测识别技术综述 SIGAI 2018.6.27
[31] 理解计算：从√2 到 AlphaGo ——第 2 季 神经计算的历史背景 SIGAI 2018.7.4
[32] 机器学习算法地图 SIGAI2018.7.6
[33] 反向传播算法推导-全连接神经网络 SIGAI2018.7.9
[34] 生成式对抗网络模型综述 SIGAI0709.
[35] 怎样成为一名优秀的算法工程师 SIGAI0711.
[36] 理解计算：从根号 2 到 AlphaGo——第三季 神经网络的数学模型 SIGAI0716
[37]【技术短文】人脸检测算法之 S3FD
[38] 基于深度负相关学习的人群计数方法
[39] 流形学习概述
[40] 关于感受野的总结

[41] 随机森林概述

[42] 基于内容的图像检索技术综述——传统经典方法

[43] 神经网络的激活函数总结