如何翻译和解释机器学习术语？请看Google官方答案。

比特小组 机器学习与数学 2019-11-18

机器学习术语表

谷歌官方出品的机器学习术语中英文对照，有了它，你还怕面试官说你用语不专业吗，还怕看不懂英文资料吗？

这张长长的术语表中列出了一般的机器学习术语和 TensorFlow 专用术语的定义。为了方便大家阅读，分成上下两篇排版，**本篇列出首字母从 A 到 K 的前半部分术语**。

1 A

A/B 测试 (A/B testing)

一种统计方法，用于将两种或多种技术进行比较，通常是将当前采用的技术与新技术进行比较。A/B 测试不仅旨在确定哪种技术的效果更好，而且还有助于了解相应差异是否具有显著的统计意义。A/B 测试通常是采用一种衡量方式对两种技术进行比较，但也适用于任意有限数量的技术和衡量方式。

准确率 (accuracy)

分类模型的正确预测所占的比例。在**多类别分类**中，准确率的定义如下：

$$\text{准确率} = \frac{\text{正确的预测数}}{\text{样本总数}}$$

在**二元分类**中，准确率的定义如下：

$$\text{准确率} = \frac{\text{正例数} + \text{负例数}}{\text{样本总数}}$$

请参阅正例和负例。

激活函数 (activation function)

一种函数（例如 ReLU 或 S 型函数），用于对上一层的所有输入求加权和，然后生成一个输出值（通常为非线性值），并将其传递给下一层。

AdaGrad

一种先进的梯度下降法，用于重新调整每个参数的梯度，以便有效地为每个参数指定独立的学习速率。如需查看完整的解释，请参阅这篇论文[1]。

ROC 曲线下面积 (AUC, Area under the ROC Curve)

一种会考虑所有可能分类阈值的评估指标。

ROC 曲线下面积是，对于随机选择的正类别样本确实为正类别，以及随机选择的负类别样本为正类别，分类器更确信前者的概率。

2 B

反向传播算法 (backpropagation)

在神经网络上执行梯度下降法的主要算法。该算法会先按前向传播方式计算（并缓存）每个节点的输出值，然后再按反向传播遍历图的方式计算损失函数值相对于每个参数的偏导数[2]。

基准 (baseline)

一种简单的模型或启发法，用作比较模型效果时的参考点。基准有助于模型开发者针对特定问题量化最低预期效果。

批次 (batch)

模型训练的一次迭代（即一次梯度更新）中使用的样本集。

另请参阅**批次大小**。

批次大小 (batch size)

一个**批次**中的样本数。例如，**SGD** 的批次大小为 1，而**小批次**的大小通常介于 10 到 1000 之间。批次大小在训练和推断期间通常是固定的；不过，TensorFlow 允许使用动态批次大小。

偏差 (bias)

距离原点的截距或偏移。偏差（也称为**偏差项**）在机器学习模型中用 b 或 w~0~ 表示。例如，在下面的公式中，偏差为 b：

$$y' = b + w_1x_1 + w_2x_2 + \ldots + w_nx_n$$

请勿与**预测偏差**混淆。

二元分类 (binary classification)

一种分类任务，可输出两种互斥类别之一。例如，对电子邮件进行评估并输出"垃圾邮件"或"非垃圾邮件"的机器学习模型就是一个二元分类器。

分箱 (binning)

请参阅**分桶**。

分桶 (bucketing)

将一个特征（通常是**连续**特征）转换成多个二元特征（称为桶或箱），通常根据值区间进行转换。例如，您可以将温度区间分割为离散分箱，而不是将温度表示成单个连续的浮点特征。假设温度数据可精确到小数点后一位，则可以将介于 0.0 到 15.0 度之间的所有温度都归入一个分箱，将介于 15.1 到 30.0 度之间的所有温度归入第二个分箱，并将介于 30.1 到 50.0 度之间的所有温度归入第三个分箱。

3 C

校准层 (calibration layer)

一种预测后调整，通常是为了降低**预测偏差**的影响。调整后的预测和概率应与观察到的标签集的分布一致。

候选采样 (candidate sampling)

一种训练时进行的优化，会使用某种函数（例如 softmax）针对所有正类别标签计算概率，但对于负类别标签，则仅针对其随机样本计算概率。例如，如果某个样本的标签为"小猎犬"和"狗"，则候选采样将针对"小猎犬"和"狗"类别输出以及其他类别（猫、棒棒糖、栅栏）的随机子集计算预测概率和相应的损失项。这种采样基于的想法是，只要**正类别**始终得到适当的正增强，**负类别**就可以从频率较低的负增强中进行学习，这确实是在实际中观察到的情况。候选采样的目的是，通过不针对所有负类别计算预测结果来提高计算效率。

分类数据 (categorical data)

一种**特征**，拥有一组离散的可能值。以某个名为 `house style` 的分类特征为例，该特征拥有一组离散的可能值（共三个），即 `Tudor, ranch, colonial` 。通过将 `house style` 表示成分类数据，相应模型可以学习 `Tudor` 、 `ranch` 和 `colonial` 分别对房价的影响。

有时，离散集中的值是互斥的，只能将其中一个值应用于指定样本。例如， `car maker` 分类特征可能只允许一个样本有一个值（ `Toyota` ）。在其他情况下，则可以应用多个值。一辆车可能会被喷涂多种不同的颜色，因此， `car color` 分类特征可能会允许单个样本具有多个值（例如 `red` 和 `white` ）。

分类特征有时称为**离散特征**。

与**数值数据**相对。

形心 (centroid)

聚类的中心，由 **k-means** 或 **k-median**算法决定。例如，如果 k 为 3，则 k-means 或 k-median 算法会找出 3 个形心。

检查点 (checkpoint)

一种数据，用于捕获模型变量在特定时间的状态。借助检查点，可以导出模型**权重**，跨多个会话

执行训练，以及使训练在发生错误之后得以继续（例如作业抢占）。请注意，图本身不包含在检查点中。

类别 (class)

为标签枚举的一组目标值中的一个。例如，在检测垃圾邮件的二元分类模型中，两种类别分别是"垃圾邮件"和"非垃圾邮件"。在识别狗品种的多类别分类模型中，类别可以是"贵宾犬"、"小猎犬"、"哈巴犬"等等。

分类不平衡的数据集 (class-imbalanced data set)

一种二元分类问题，在此类问题中，两种类别的标签在出现频率方面具有很大的差距。例如，在某个疾病数据集中，0.0001 的样本具有正类别标签，0.9999 的样本具有负类别标签，这就属于分类不平衡问题；但在某个足球比赛预测器中，0.51 的样本的标签为其中一个球队赢，0.49 的样本的标签为另一个球队赢，这就不属于分类不平衡问题。

分类模型 (classification model)

一种机器学习模型，用于区分两种或多种离散类别。例如，某个自然语言处理分类模型可以确定输入的句子是法语、西班牙语还是意大利语。请与回归模型进行比较。

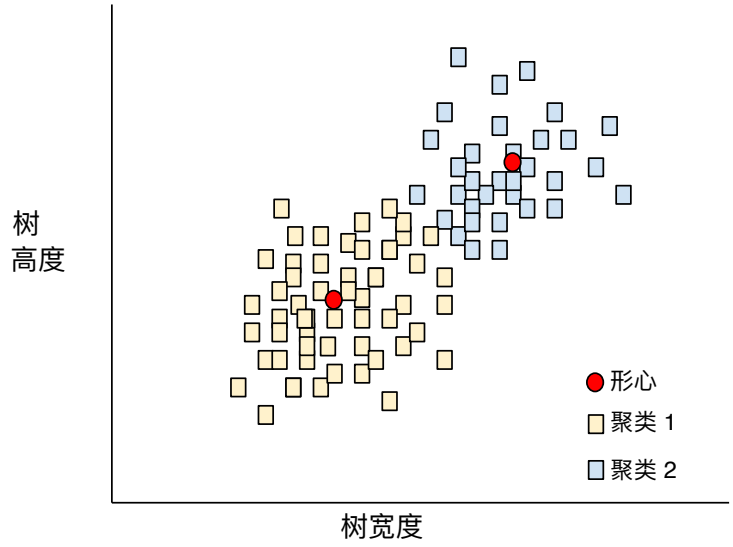
分类阈值 (classification threshold)

一种标量值条件，应用于模型预测的得分，旨在将正类别与负类别区分开。将逻辑回归结果映射到二元分类时使用。以某个逻辑回归模型为例，该模型用于确定指定电子邮件是垃圾邮件的概率。如果分类阈值为 0.9，那么逻辑回归值高于 0.9 的电子邮件将被归类为"垃圾邮件"，低于 0.9 的则被归类为"非垃圾邮件"。

聚类 (clustering)

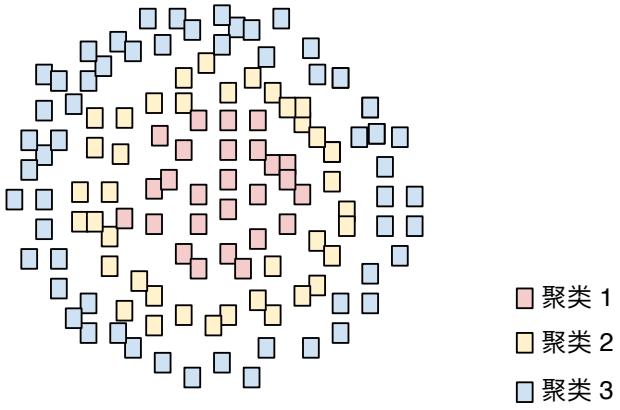
将关联的样本分成一组，一般用于非监督式学习。在所有样本均分组完毕后，相关人员便可选择性地为每个聚类赋予含义。

聚类算法有很多。例如，**k-means**算法会基于样本与形心的接近程度聚类样本，如下图所示：



之后，研究人员便可查看这些聚类并进行其他操作，例如，将聚类 1 标记为"矮型树"，将聚类 2 标记为"全尺寸树"。

再举一个例子，例如基于样本与中心点距离的聚类算法，如下所示：



协同过滤 (collaborative filtering)

根据很多其他用户的兴趣来预测某位用户的兴趣。协同过滤通常用在推荐系统中。

混淆矩阵 (confusion matrix)

一种 NxN 表格，用于总结分类模型的预测效果；即标签和模型预测的分类之间的关联。在混淆矩阵中，一个轴表示模型预测的标签，另一个轴表示实际标签。N 表示类别个数。在二元分类问题中，N=2。例如，下面显示了一个二元分类问题的混淆矩阵示例：

	肿瘤预测标签	非肿瘤预测标签

肿瘤实际标签	18	1
非肿瘤实际标签	6	452

上面的混淆矩阵显示，在 19 个实际有肿瘤的样本中，该模型正确地将 18 个归类为有肿瘤（18 个正例），错误地将 1 个归类为没有肿瘤（1 个假负例）。同样，在 458 个实际没有肿瘤的样本中，模型归类正确的有 452 个（452 个负例），归类错误的有 6 个（6 个假正例）。

多类别分类问题的混淆矩阵有助于确定出错模式。例如，某个混淆矩阵可以揭示，某个经过训练以识别手写数字的模型往往会将 4 错误地预测为 9，将 7 错误地预测为 1。

混淆矩阵包含计算各种效果指标（包括**精确率**和**召回率**）所需的充足信息。

连续特征 (continuous feature)

一种浮点特征，可能值的区间不受限制。与**离散特征**相对。

收敛 (convergence)

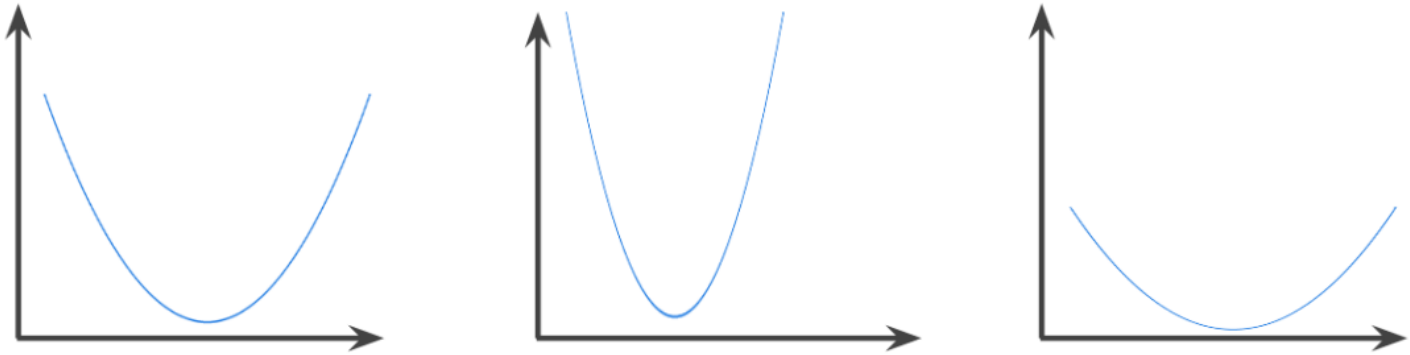
通俗来说，收敛通常是指在训练期间达到的一种状态，即经过一定次数的迭代之后，训练**损失**和验证损失在每次迭代中的变化都非常小或根本没有变化。也就是说，如果采用当前数据进行额外的训练将无法改进模型，模型即达到收敛状态。在深度学习中，损失值有时会在最终下降之前的多次迭代中保持不变或几乎保持不变，暂时形成收敛的假象。

另请参阅**早停法**。

另请参阅 Boyd 和 Vandenberghe 合著的 Convex Optimization（《凸优化》）。

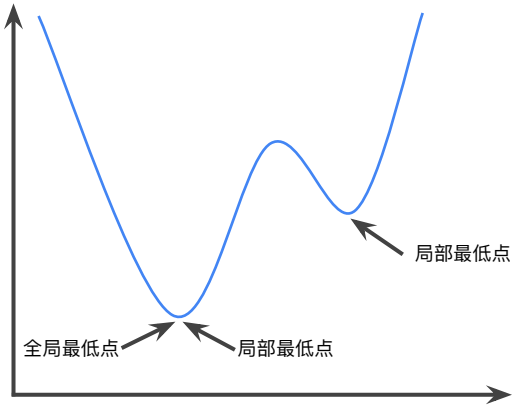
凸函数 (convex function)

一种函数，函数图像以上的区域为**凸集**。典型凸函数的形状类似于字母**U**。例如，以下都是凸函数：



典型凸函数的形状类似于字母 U。

相反，以下函数则不是凸函数。请注意图像上方的区域如何不是凸集：



严格凸函数只有一个局部最低点，该点也是全局最低点。经典的 U 形函数都是严格凸函数。不过，有些凸函数（例如直线）则不是这样。

很多常见的**损失函数**（包括下列函数）都是凸函数：

- L_2 损失函数
- 对数损失函数
- L_1 正则化
- L_2 正则化

梯度下降法的很多变体都一定能找到一个接近严格凸函数最小值的点。同样，**随机梯度下降法**的很多变体都有很高的可能性能够找到接近严格凸函数最小值的点（但并非一定能找到）。

两个凸函数的和（例如 L_2 损失函数 + L_1 正则化）也是凸函数。

深度模型绝不会是凸函数。值得注意的是，专门针对**凸优化**设计的算法往往总能在深度网络上找到非常好的解决方案，虽然这些解决方案并不一定对应于全局最小值。

凸优化 (convex optimization)

使用数学方法（例如**梯度下降法**）寻找**凸函数**最小值的过程。机器学习方面的大量研究都是专注于如何通过公式将各种问题表示成凸优化问题，以及如何更高效地解决这些问题。

如需完整的详细信息，请参阅 Boyd 和 Vandenberghe 合著的 Convex Optimization（《凸优化》）。

凸集 (convex set)

欧几里得空间的一个子集，其中任意两点之间的连线仍完全落在该子集内。例如，下面的两个图形都是凸集：

矩形和半椭圆形都是凸集。

相反，下面的两个图形都不是凸集：

缺少一块的饼图以及烟花图都是非凸集。

卷积 (convolution)

简单来说，卷积在数学中指两个函数的组合。在机器学习中，卷积结合使用卷积过滤器和输入矩阵来训练权重。

机器学习中的"卷积"一词通常是**卷积运算**或**卷积层**的简称。

如果没有卷积，机器学习算法就需要学习大张量中每个单元格各自的权重。例如，用 2K x 2K 图像训练的机器学习算法将被迫找出 400 万个单独的权重。而使用卷积，机器学习算法只需在**卷积过滤器**中找出每个单元格的权重，大大减少了训练模型所需的内存。在应用卷积过滤器后，它只需跨单元格进行复制，每个单元格都会与过滤器相乘。

卷积过滤器 (convolutional filter)

卷积运算中的两个参与方之一。（另一个参与方是输入矩阵切片。）卷积过滤器是一种矩阵，其**等级**与输入矩阵相同，但形状小一些。以 28×28 的输入矩阵为例，过滤器可以是小于 28×28 的任何二维矩阵。

在图形操作中，卷积过滤器中的所有单元格通常按照固定模式设置为 1 和 0。在机器学习中，卷积过滤器通常先选择随机数字，然后由网络训练出理想值。

卷积层 (convolutional layer)

深度神经网络的一个层，**卷积过滤器**会在其中传递输入矩阵。以下面的 3x3 **卷积过滤器**为例：

0	1	0
1	0	1
0	1	0

下面的动画显示了一个由 9 个卷积运算（涉及 5x5 输入矩阵）组成的卷积层。请注意，每个卷积运算都涉及一个不同的 3x3 输入矩阵切片。由此产生的 3×3 矩阵（右侧）就包含 9 个卷积运算的结果：

128	97	53	201	198
35	22	25	200	195
37	24	28	197	182
33	28	92	195	179
31	40	100	192	177

181	303	618
115		

卷积神经网络 (convolutional neural network)

一种神经网络，其中至少有一层为**卷积层**。典型的卷积神经网络包含以下几层的组合：

- 卷积层
- 池化层
- 密集层

卷积神经网络在解决某些类型的问题（如图像识别）上取得了巨大成功。

卷积运算 (convolutional operation)

如下所示的两步数学运算：

1. 对**卷积过滤器**和输入矩阵切片执行元素级乘法。（输入矩阵切片与卷积过滤器具有相同的等级和大小。）
2. 对生成的积矩阵中的所有值求和。

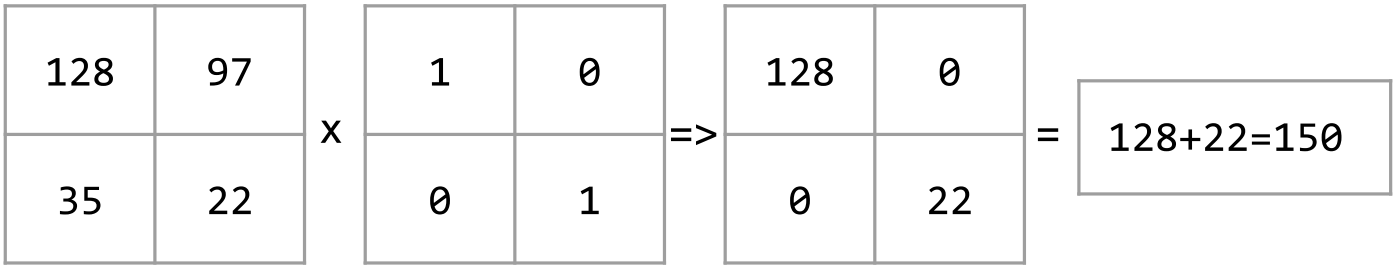
以下面的 5x5 输入矩阵为例：

128	97	53	201	198
35	22	25	200	195
37	24	28	197	182
33	28	92	195	179
31	40	100	192	177

现在，以下面这个 2x2 卷积过滤器为例：

1	0
0	1

每个卷积运算都涉及一个 2x2 输入矩阵切片。例如，假设我们使用输入矩阵左上角的 2x2 切片。这样一来，对此切片进行卷积运算将如下所示：



卷积层由一系列卷积运算组成，每个卷积运算都针对不同的输入矩阵切片。

成本 (cost)

与损失的含义相同。

交叉熵 (cross-entropy)

对数损失函数向多类别分类问题的一种泛化。交叉熵可以量化两种概率分布之间的差异。另请参阅困惑度。

自定义 Estimator (custom Estimator)

您按照这些说明[3]自行编写的**Estimator**。

与预创建的 **Estimator** 相对。

4 D

数据分析 (data analysis)

根据样本、测量结果和可视化内容来理解数据。数据分析在首次收到数据集、构建第一个模型之前特别有用。此外，数据分析在理解实验和调试系统问题方面也至关重要。

DataFrame

一种热门的数据类型，用于表示 Pandas 中的数据集。DataFrame 类似于表格。DataFrame 的每一列都有一个名称（标题），每一行都由一个数字标识。

数据集 (data set)

一组样本的集合。

Dataset API (tf.data)

一种高级别的 TensorFlow API，用于读取数据并将其转换为机器学习算法所需的格式。`tf.data.Dataset` 对象表示一系列元素，其中每个元素都包含一个或多个张量。`tf.data.Iterator` 对象可获取 `Dataset` 中的元素。

如需详细了解 Dataset API，请参阅《TensorFlow 编程人员指南》中的导入数据[4]。

决策边界 (decision boundary)

在二元分类或多类别分类问题中，模型学到的类别之间的分界线。例如，在以下表示某个二元分类问题的图片中，决策边界是橙色类别和蓝色类别之间的分界线：

两种类别之间明确定义的边界。

密集层 (dense layer)

与全连接层的含义相同。

深度模型 (deep model)

一种神经网络，其中包含多个隐藏层。深度模型依赖于可训练的非线性关系。

与宽度模型相对。

密集特征 (dense feature)

一种大部分值是非零值的**特征**，通常是浮点值**张量**。与**稀疏特征**相对。

设备 (device)

一类可运行 TensorFlow 会话的硬件，包括 CPU、GPU 和 TPU。

离散特征 (discrete feature)

一种**特征**，包含有限个可能值。例如，某个值只能是"动物"、"蔬菜"或"矿物"的特征便是一个离散特征（或分类特征）。与**连续特征**相对。

丢弃正则化 (dropout regularization)

正则化的一种形式，在训练**神经网络**方面非常有用。丢弃正则化的运作机制是，在一个梯度步长中移除从神经网络层中随机选择的固定数量的单元。丢弃的单元越多，正则化效果就越强。这类似于训练神经网络以模拟较小网络的指数级规模集成学习。如需完整的详细信息，请参阅 Dropout: A Simple Way to Prevent Neural Networks from Overfitting（《丢弃：一种防止神经网络过拟合的简单方法》）。

动态模型 (dynamic model)

一种**模型**，以持续更新的方式在线接受训练。也就是说，数据会源源不断地进入这种模型。

5 E

早停法 (early stopping)

一种**正则化**方法，是指在训练损失仍可以继续降低之前结束模型训练。使用早停法时，您会在**验证数据集**的损失开始增大（也就是**泛化**效果变差）时结束模型训练。

嵌套 (embeddings)

一种分类特征，以连续值特征表示。通常，嵌套是指将高维度向量映射到低维度的空间。例如，

您可以采用以下两种方式之一来表示英文句子中的单词：

- 表示成包含百万个元素（高维度）的**稀疏向量**，其中所有元素都是整数。向量中的每个单元格都表示一个单独的英文单词，单元格中的值表示相应单词在句子中出现的次数。由于单个英文句子包含的单词不太可能超过 50 个，因此向量中几乎每个单元格都包含 0。少数非 0 的单元格中将包含一个非常小的整数（通常为 1），该整数表示相应单词在句子中出现的次数。
- 表示成包含数百个元素（低维度）的**密集向量**，其中每个元素都存储一个介于 0 到 1 之间的浮点值。这就是一种嵌套。

在 TensorFlow 中，会按**反向传播损失**训练嵌套，和训练**神经网络**中的任何其他参数一样。

经验风险最小化 (ERM, empirical risk minimization)

用于选择可以将基于训练集的损失降至最低的函数。与**结构风险最小化**相对。

集成学习 (ensemble)

多个**模型**的预测结果的并集。您可以通过以下一项或多项来创建集成学习：

- 不同的初始化
- 不同的**超参数**
- 不同的整体结构

深度模型和宽度模型[5]属于一种集成学习。

周期 (epoch)

在训练时，整个数据集的一次完整遍历，以便不漏掉任何一个样本。因此，一个周期表示（ N / **批次大小**）次训练**迭代**，其中 N 是样本总数。

Estimator

`tf.Estimator` 类的一个实例，用于封装负责构建 TensorFlow 图并运行 TensorFlow 会话的逻辑。您可以创建**自定义 Estimator**（如需相关介绍，请**点击此处**[6]），也可以实例化其他人**预创建的 Estimator**。

样本 (example)

数据集的一行。一个样本包含一个或多个**特征**，此外还可能包含一个**标签**。另请参阅**有标签样本**和**无标签样本**。

6 F

假负例 (FN, false negative)

被模型错误地预测为**负类别**的样本。例如，模型推断出某封电子邮件不是垃圾邮件（负类别），但该电子邮件其实是垃圾邮件。

假正例 (FP, false positive)

被模型错误地预测为**正类别**的样本。例如，模型推断出某封电子邮件是垃圾邮件（正类别），但该电子邮件其实不是垃圾邮件。

假正例率（false positive rate, 简称 FP 率）

ROC 曲线中的 x 轴。FP 率的定义如下：

$$\text{假正例率} = \frac{\text{假正例数}}{\text{假正例数} + \text{负例数}}$$

特征 (feature)

在进行**预测**时使用的输入变量。

特征列 (tf.feature_column)

指定模型应该如何解读特定特征的一种函数。此类函数的输出结果是所有**Estimators** 构造函数的必需参数。

借助 `tf.feature_column` 函数，模型可对输入特征的不同表示法轻松进行实验。有关详情，请参阅《TensorFlow 编程人员指南》中的**特征列**[7]一章。

"特征列"是 Google 专用的术语。特征列在 Yahoo/Microsoft 使用的**VW**[8]系统中称为"命名空间"，也称为**场**[9]。

特征组合 (feature cross)

通过将单独的特征进行组合（求笛卡尔积）而形成的**合成特征**。特征组合有助于表达非线性关系。

特征工程 (feature engineering)

指以下过程：确定哪些**特征**可能在训练模型方面非常有用，然后将日志文件及其他来源的原始数据转换为所需的特征。在 TensorFlow 中，特征工程通常是指将原始日志文件条目转换为**tf.Example** 协议缓冲区。另请参阅**tf.Transform**[10]。

特征工程有时称为**特征提取**。

特征集 (feature set)

训练机器学习模型时采用的一组**特征**。例如，对于某个用于预测房价的模型，邮政编码、房屋面积以及房屋状况可以组成一个简单的特征集。

特征规范 (feature spec)

用于描述如何从 **tf.Example**协议缓冲区提取**特征**数据。由于 tf.Example 协议缓冲区只是一个数据容器，因此您必须指定以下内容：

- 要提取的数据（即特征的键）
- 数据类型（例如 float 或 int）
- 长度（固定或可变）

Estimator API 提供了一些可用来根据给定**FeatureColumns** 列表生成特征规范的工具。

少量样本学习 (few-shot learning)

一种机器学习方法（通常用于对象分类），旨在仅通过少量训练样本学习有效的分类器。

另请参阅**单样本学习**。

完整 softmax (full softmax)

请参阅**softmax**。与**候选采样**相对。

全连接层 (fully connected layer)

一种**隐藏层**，其中的每个**节点**均与下一个隐藏层中的每个节点相连。

全连接层又称为**密集层**。

7 G

泛化 (generalization)

指的是模型依据训练时采用的数据，针对以前未见过的新数据做出正确预测的能力。

广义线性模型 (generalized linear model)

最小二乘回归模型（基于**高斯噪声**[11]）向其他类型的模型（基于其他类型的噪声，例如**泊松噪声**[12]或分类噪声）进行的一种泛化。广义线性模型的示例包括：

- **逻辑回归**
- 多类别回归
- 最小二乘回归

可以通过**凸优化**[13]找到广义线性模型的参数。

广义线性模型具有以下特性：

- 最优的最小二乘回归模型的平均预测结果等于训练数据的平均标签。
- 最优的逻辑回归模型预测的平均概率等于训练数据的平均标签。

广义线性模型的功能受其特征的限制。与深度模型不同，广义线性模型无法"学习新特征"。

梯度 (gradient)

偏导数相对于所有自变量的向量。在机器学习中，梯度是模型函数偏导数的向量。梯度指向最高速上升的方向。

梯度裁剪 (gradient clipping)

在应用梯度值之前先设置其上限。梯度裁剪有助于确保数值稳定性以及防止梯度爆炸[14]。

梯度下降法 (gradient descent)

一种通过计算并且减小梯度将损失降至最低的技术，它以训练数据为条件，来计算损失相对于模型参数的梯度。通俗来说，梯度下降法以迭代方式调整参数，逐渐找到权重和偏差的最佳组合，从而将损失降至最低。

图 (graph)

TensorFlow 中的一种计算规范。图中的节点表示操作。边缘具有方向，表示将某项操作的结果（一个张量[15]）作为一个操作数传递给另一项操作。可以使用TensorBoard 直观呈现图。

8 H

启发法 (heuristic)

一种非最优但实用的问题解决方案，足以用于进行改进或从中学习。

隐藏层 (hidden layer)

神经网络中的合成层，介于输入层（即特征）和输出层（即预测）之间。神经网络包含一个或多个隐藏层。

合页损失函数 (hinge loss)

一系列用于分类的损失函数，旨在找到距离每个训练样本都尽可能远的决策边界，从而使样本和边界之间的裕度最大化。KSVM使用合页损失函数（或相关函数，例如平方合页损失函数）。对于二元分类，合页损失函数的定义如下：

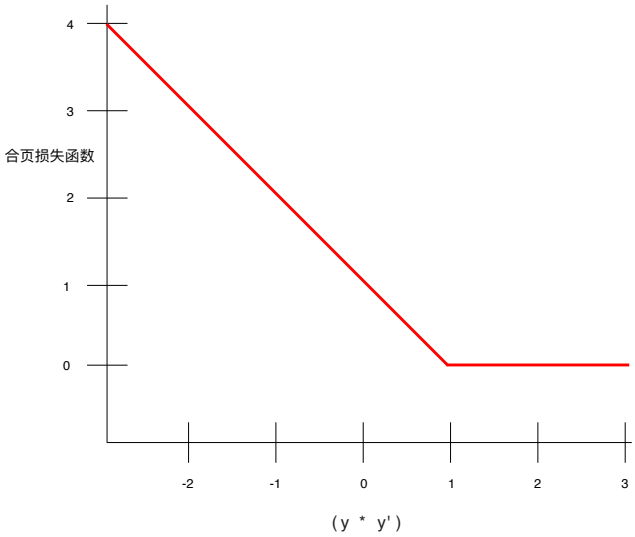
$$\text{loss} = \max(0, 1 - (y' * y))$$

其中"y'"表示分类器模型的原始输出：

$$y' = b + w_1x_1 + w_2x_2 + \ldots w_nx_n$$

"y"表示真标签，值为 -1 或 +1。

因此，合页损失与 (y * y') 的关系图如下所示：



维持数据 (holdout data)

训练期间故意不使用（"维持"）的**样本**。**验证数据集**和**测试数据集**都属于维持数据。维持数据有助于评估模型向训练时所用数据之外的数据进行泛化的能力。与基于训练数据集的损失相比，基于维持数据集的损失有助于更好地估算基于未见过的数据集的损失。

超参数 (hyperparameter)

在模型训练的连续过程中，您调节的"旋钮"。例如，**学习速率**就是一种超参数。

与**参数**相对。

超平面 (hyperplane)

将一个空间划分为两个子空间的边界。例如，在二维空间中，直线就是一个超平面，在三维空间中，平面则是一个超平面。在机器学习中更典型的是：超平面是分隔高维度空间的边界。**核支持**

向量机利用超平面将正类别和负类别区分开来（通常是在极高维度空间中）。

9

独立同等分布 (i.i.d, independently and identically distributed)

从不会改变的分布中提取的数据，其中提取的每个值都不依赖于之前提取的值。i.i.d. 是机器学习的理想气体[16] - 一种实用的数学结构，但在现实世界中几乎从未发现过。例如，某个网页的访问者在短时间内的分布可能为 i.i.d.，即分布在该短时间内没有变化，且一位用户的访问行为通常与另一位用户的访问行为无关。不过，如果将时间窗口扩大，网页访问者的分布可能呈现出季节性变化。

推断 (inference)

在机器学习中，推断通常指以下过程：通过将训练过的模型应用于无标签样本来做出预测。在统计学中，推断是指在某些观测数据条件下拟合分布参数的过程。（请参阅维基百科中有关统计学推断的文章[17]。）

输入函数 (input function)

在 TensorFlow 中，用于将输入数据返回到 **Estimator**的训练、评估或预测方法的函数。例如，训练输入函数会返回训练集中的一批特征和标签。

输入层 (input layer)

神经网络中的第一层（接收输入数据的层）。

实例 (instance)

与样本的含义相同。

可解释性 (interpretability)

模型的预测可解释的难易程度。深度模型通常不可解释，也就是说，很难对深度模型的不同层进行解释。相比之下，线性回归模型和宽度模型的可解释性通常要好得多。

评分者间一致性信度 (inter-rater agreement)

一种衡量指标，用于衡量在执行某项任务时评分者达成一致的频率。如果评分者未达成一致，则可能需要改进任务说明。有时也称为**注释者间一致性信度**或**评分者间可靠性信度**。另请参阅Cohen's kappa（最热门的评分者间一致性信度衡量指标之一）。

迭代 (iteration)

模型的权重在训练期间的一次更新。迭代包含计算参数在单**批次**数据上的梯度损失。

10 K

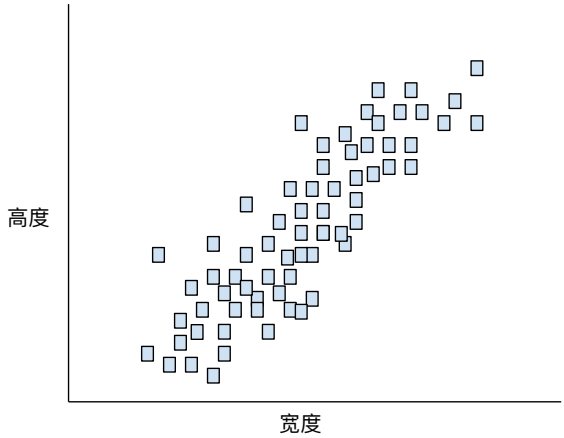
k-means

一种热门的**聚类**算法，用于对非监督式学习中的样本进行分组。k-means 算法基本上会执行以下操作：

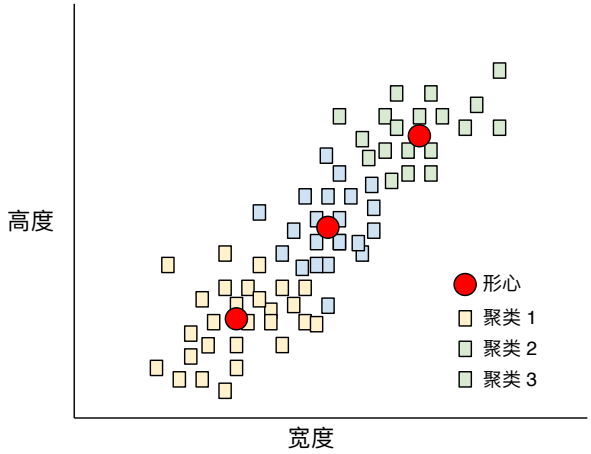
- 以迭代方式确定最佳的 k 中心点（称为**形心**）。
- 将每个样本分配到最近的形心。与同一个形心距离最近的样本属于同一个组。

k-means 算法会挑选形心位置，以最大限度地减小每个样本与其最接近形心之间的距离的累积平方。

以下面的小狗高度与小狗宽度的关系图为例：



如果 k=3，则 k-means 算法会确定三个形心。每个样本都被分配到与其最接近的形心，最终产生三个组：



假设制造商想要确定小、中和大号狗毛衣的理想尺寸。在该聚类中，三个形心用于标识每只狗的平均高度和平均宽度。因此，制造商可能应该根据这三个形心确定毛衣尺寸。请注意，聚类的形心通常不是聚类中的样本。

上图显示了 k-means 应用于仅具有两个特征（高度和宽度）的样本。请注意，k-means 可以跨多个特征为样本分组。

k-median

与 **k-means** 紧密相关的聚类算法。两者的实际区别如下：

- 对于 k-means，确定形心的方法是，最大限度地减小候选形心与它的每个样本之间的距离平方和。
- 对于 k-median，确定形心的方法是，最大限度地减小候选形心与它的每个样本之间的距离总和。

请注意，距离的定义也有所不同：

- k-means 采用从形心到样本的**欧几里得距离**[18]。（在二维空间中，欧几里得距离即使用勾股定理来计算斜边。）例如，(2,2) 与 (5,-2) 之间的 k-means 距离为：

$$\text{欧几里德距离} = \sqrt{(2-5)^2 + (2--2)^2} = 5$$

- k-median 采用从形心到样本的**曼哈顿距离**[19]。这个距离是每个维度中绝对差异值的总和。例如，(2,2) 与 (5,-2) 之间的 k-median 距离为：

$$\text{曼哈顿距离} = |2-5| + |2--2| = 7$$

Keras

一种热门的 Python 机器学习 API。**Keras**[20]能够在多种深度学习框架上运行，其中包括 TensorFlow（在该框架上，Keras 作为**tf.keras**[21]提供）。

核支持向量机 (KSVM, Kernel Support Vector Machines)

一种分类算法，旨在通过将输入数据向量映射到更高维度的空间，来最大化**正类别**和**负类别**之间的裕度。以某个输入数据集包含一百个特征的分类问题为例。为了最大化正类别和负类别之间的裕度，KSVM 可以在内部将这些特征映射到百万维度的空间。KSVM 使用合页损失函数。

上篇完

本文版权归谷歌，本公众号精心编译制作，以方便大家手机上浏览、查阅和学习。

参考资料

[1] 学习速率: http://www.jmlr.org/papers/volume12/duchi11a/duchi11a.pdf

[2] 偏导数: https://en.wikipedia.org/wiki/Partial_derivative

[3] 自定义 estimator 说明: https://www.tensorflow.org/extend/estimators?hl=zh-CN

[4] 导入数据: https://www.tensorflow.org/programmers_guide/datasets?hl=zh-CN

[5] 深度模型和宽度模型: https://www.tensorflow.org/tutorials/wide_and_deep?hl=zh-CN

[6] 自定义 estimator: https://www.tensorflow.org/extend/estimators?hl=zh-CN

[7] 特征列: https://www.tensorflow.org/get_started/feature_columns?hl=zh-CN

[8] VW: https://en.wikipedia.org/wiki/Vowpal_Wabbit

[9] 场: https://www.csie.ntu.edu.tw/~cjlin/libffm/

[10] tf.Transform: https://github.com/tensorflow/transform

[11] 高斯噪声: https://en.wikipedia.org/wiki/Gaussian_noise

[12] 泊松噪声: https://en.wikipedia.org/wiki/Shot_noise

[13] 凸优化: https://en.wikipedia.org/wiki/Convex_optimization

[14] 梯度爆炸：http://www.cs.toronto.edu/~rgrosse/courses/csc321_2017/readings/L15%20Exploding%20and%20Vanishing%20Gradients.pdf

[15] 张量: https://www.tensorflow.org/api_docs/python/tf/Tensor?hl=zh-CN

[16] 理想气体: https://en.wikipedia.org/wiki/Ideal_gas

[17] 统计学推断: https://en.wikipedia.org/wiki/Statistical_inference

[18] 欧几里得距离: https://en.wikipedia.org/wiki/Euclidean_distance

[19] 曼哈顿距离: https://en.wikipedia.org/wiki/Taxicab_geometry

[20] Keras: https://keras.io

[21] tf.keras: https://www.tensorflow.org/api_docs/python/tf/keras?hl=zh-CN