

# 如何翻译和解释机器学习术语？请看 Google 官方答案 下

比特小组 机器学习与数学 2019-11-21

## 机器学习术语表

Google 官方出品机器学习中英文术语对照表，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。

A 到 K 部分术语请点击[这里](#)。

### 1 L

#### $L_1$ 损失函数 ( $L_1$ loss)

一种损失函数，基于模型预测的值与标签的实际值之差的绝对值。与  $L_2$  损失函数相比， $L_1$  损失函数对离群值的敏感性弱一些。

#### $L_1$ 正则化 ( $L_1$ regularization)

一种正则化，根据权重的绝对值的总和来惩罚权重。在依赖稀疏特征的模型中， $L_1$  正则化有助于使不相关或几乎不相关的特征的权重正好为 0，从而将这些特征从模型中移除。与  $L_2$  正则化相对。

#### $L_2$ 损失函数 ( $L_2$ loss)

请参阅平方损失函数。

#### $L_2$ 正则化 ( $L_2$ regularization)

一种正则化，根据权重的平方和来惩罚权重。 $L_2$  正则化有助于使离群值（具有较大正值或较小负值）权重接近于 0，但又不正好为 0。（与  $L_1$  正则化相对。）在线性模型中， $L_2$

正则化始终可以改进泛化。

## 标签 (label)

在监督式学习中，标签指**样本**的"答案"或"结果"部分。有标签数据集中的每个样本都包含一个或多个特征以及一个标签。例如，在房屋数据集中，特征可能包括卧室数、卫生间数以及房龄，而标签则可能是房价。在垃圾邮件检测数据集中，特征可能包括主题行、发件人以及电子邮件本身，而标签则可能是"垃圾邮件"或"非垃圾邮件"。

## 有标签样本 (labeled example)

包含**特征**和**标签**的样本。在监督式训练中，模型从有标签样本中学习规律。

## lambda

与**正则化率**的含义相同。

(多含义术语，我们在此关注的是该术语在**正则化**中的定义。)

## 层 (layer)

**神经网络**中的一组**神经元**，负责处理一组输入特征，或一组神经元的输出。

此外还指 TensorFlow 中的抽象层。层是 Python 函数，以**张量**和配置选项作为输入，然后生成其他张量作为输出。当必要的张量组合起来后，用户便可以通过**模型函数**将结果转换为**Estimator**。

## Layers API (tf.layers)

一种 TensorFlow API，用于以层组合的方式构建**深度**神经网络。通过 Layers API，您可以构建不同类型的**层**，例如：

- 通过 `tf.layers.Dense` 构建**全连接层**。
- 通过 `tf.layers.Conv2D` 构建**卷积层**。

在编写**自定义 Estimator**时，您可以编写"层"对象来定义所有**隐藏层**的特征。

Layers API 遵循 **Keras** layers API 规范。也就是说，除了前缀不同以外，Layers API 中的所有函数均与 Keras layers API 中的对应函数具有相同的名称和签名。

## 学习速率 (learning rate)

在训练模型时用于梯度下降的一个标量。在每次迭代期间，**梯度下降法**都会将学习速率与梯度相乘。得出的乘积称为**梯度步长**。

学习速率是一个重要的**超参数**。

## 最小二乘回归 (least squares regression)

一种通过最小化  $L_2$  损失训练出的线性回归模型。

## 线性回归 (linear regression)

一种**回归模型**，通过将输入特征进行线性组合输出连续值。

## 逻辑回归 (logistic regression)

一种模型，通过将 **S 型函数**应用于线性预测，生成分类问题中每个可能的离散标签值的概率。虽然逻辑回归经常用于**二元分类**问题，但也可用于**多类别**分类问题（其叫法变为**多类别逻辑回归**或**多项回归**）。

## 对数 (logits)

分类模型生成的原始（非标准化）预测向量，通常会传递给标准化函数。如果模型要解决多类别分类问题，则对数通常变成softmax 函数的输入。之后，softmax 函数会生成一个（标准化）概率向量，对应于每个可能的类别。

此外，对数有时也称为 **S 型函数**的元素级反函数。如需了解详细信息，请参阅**sigmoid cross entropy with logits<sup>[1]</sup>**。

## 对数损失函数 (Log Loss)

二元**逻辑回归**中使用的**损失函数**。

## 对数几率 (log-odds)

某个事件几率的对数。

如果事件涉及二元概率，则**几率**指的是成功概率 (p) 与失败概率 (1-p) 之比。例如，假设某个给定事件的成功概率为 90%，失败概率为 10%。在这种情况下，几率的计算公式如下：

$$\text{几率} = \frac{p}{(1-p)} = \frac{.9}{.1} = 9$$

简单来说，对数几率即几率的对数。按照惯例，"对数"指自然对数，但对数的基数其实可以是任何大于 1 的数。若遵循惯例，上述示例的对数几率应为：

$$\text{对数几率} = \ln(9) = 2.2$$

对数几率是 **S 型函数** 的反函数。

## 损失 (Loss)

一种衡量指标，用于衡量模型的**预测**偏离其**标签**的程度。或者更悲观地说是衡量模型有多差。要确定此值，模型必须定义损失函数。例如，线性回归模型通常将**均方误差**用作损失函数，而逻辑回归模型则使用**对数损失函数**。

## 2 M

## 机器学习 (machine learning)

一种程序或系统，用于根据输入数据构建（训练）预测模型。这种系统会利用学到的模型根据从分布（训练该模型时使用的同一分布）中提取的新数据（以前从未见过的数据）进行实用的预测。机器学习还指与这些程序或系统相关的研究领域。

## 均方误差 (MSE, Mean Squared Error)

每个样本的平均平方损失。MSE 的计算方法是**平方损失**除以**样本数**。**TensorFlow Playground**显示的"训练损失"值和"测试损失"值都是 MSE。

## 指标 (metric)

您关心的一个数值。可能可以也可能不可以在机器学习系统中得到优化。您的系统尝试优化的指标称为**目标**。

## Metrics API (tf.metrics)

一种用于评估模型的 TensorFlow API。例如，`tf.metrics.accuracy` 用于确定模型的预测与标签匹配的频率。在编写**自定义 Estimator** 时，您可以调用 Metrics API 函数来指定应如何评估您的模型。

## 小批次 (mini-batch)

从整批**样本**内随机选择并在训练或推断过程的一次迭代中一起运行的一小部分样本。小批次的**批次大小**通常介于 10 到 1000 之间。与基于完整的训练数据计算损失相比，基于小批次数据计算损失要高效得多。

## 小批次随机梯度下降法 (SGD, mini-batch stochastic gradient descent)

一种采用**小批次**样本的**梯度下降法**。也就是说，小批次 SGD 会根据一小部分训练数据来估算梯度。**Vanilla SGD**使用的小批次的大小为 1。

## ML

**机器学习**的缩写。

## 模型 (model)

机器学习系统从训练数据学到的内容的表示形式。多含义术语，可以理解为下列两种相关含义之一：

- 一种 **TensorFlow** 图，用于表示预测的计算结构。
- 该 TensorFlow 图的特定权重和偏差，通过**训练**决定。

## 模型函数 (model function)

**Estimator**中的函数，用于实现机器学习训练、评估和推断。例如，模型函数的训练部分

可以处理以下任务：定义深度神经网络的拓扑并确定其**优化器**函数。如果使用**预创建的 Estimator**，则有人已为您编写了模型函数。如果使用**自定义 Estimator**，则必须自行编写模型函数。

有关编写模型函数的详细信息，请参阅创建自定义 Estimator。

## 模型训练 (model training)

确定最佳**模型**的过程。

## 动量 (Momentum)

一种先进的梯度下降法，其中学习步长不仅取决于当前步长的导数，还取决于之前一步或多步的步长的导数。动量涉及计算梯度随时间而变化的指数级加权移动平均值，与物理学中的动量类似。动量有时可以防止学习过程被卡在局部最小的情况。

## 多类别分类 (multi-class classification)

区分两种以上类别的分类问题。例如，枫树大约有 128 种，因此，确定枫树种类的模型就属于多类别模型。反之，仅将电子邮件分为两类（"垃圾邮件"和"非垃圾邮件"）的模型属于**二元分类模型**。

## 多项分类 (multinomial classification)

与**多类别分类**的含义相同。

## 3 N

## NaN 陷阱 (NaN trap)

模型中的一个数字在训练期间变成**NaN<sup>[2]</sup>**，这会导致模型中的很多或所有其他数字最终也会变成 NaN。

NaN 是"非数字"的缩写。

## 负类别 (negative class)

在**二元分类**中，一种类别称为正类别，另一种类别称为负类别。正类别是我们要寻找的类别，负类别则是另一种可能性。例如，在医学检查中，负类别可以是"非肿瘤"。在电子邮件分类器中，负类别可以是"非垃圾邮件"。另请参阅**正类别**。

## 神经网络 (neural network)


一种模型，灵感来源于脑部结构，由多个层构成（至少有一个是**隐藏层**），每个层都包含简单相连的单元或**神经元**（具有非线性关系）。

## 神经元 (neuron)

**神经网络**中的节点，通常会接收多个输入值并生成一个输出值。神经元通过将**激活函数**（非线性转换）应用于输入值的加权和来计算输出值。

## 节点 (node)

多含义术语，可以理解为下列两种含义之一：

- **隐藏层**中的神经元。
- TensorFlow  图中的操作。

## 标准化 (normalization)

将实际的值区间转换为标准的值区间（通常为 -1 到 +1 或 0 到 1）的过程。例如，假设某个特征的自然区间是 800 到 6000。通过减法和除法运算，您可以将这些值标准化为位于 -1 到 +1 区间内。

另请参阅**缩放**。

## 数值数据 (numerical data)

用整数或实数表示的**特征**。例如，在房地产模型中，您可能会用数值数据表示房子大小（以平方英尺或平方米为单位）。如果用数值数据表示特征，则可以表明特征的值相互之间具有数学关系，并且与标签可能也有数学关系。例如，如果用数值数据表示房子大小，则可以表明面积为 200 平方米的房子是面积为 100 平方米的两倍。此外，房子面积的平方米数可能与房价存在一定的数学关系。

并非所有整数数据都应表示成数值数据。例如，世界上某些地区的邮政编码是整数，但在模型中，不应将整数邮政编码表示成数值数据。这是因为邮政编码 **20000** 在效力上并不是邮政编码 10000 的两倍（或一半）。此外，虽然不同的邮政编码确实与不同的房地产价值有关，但我们也不能假设邮政编码为 20000 的房地产在价值上是邮政编码为 10000 的房地产的两倍。邮政编码应表示成**分类数据**。

数值特征有时称为**连续特征**。

## Numpy

一个**开放源代码数学库**<sup>[3]</sup>，在 Python 中提供高效的数组操作。**Pandas** 建立在 Numpy 之上。

## 4 O

### 目标 (objective)

算法尝试优化的指标。

### 离线推断 (offline inference)

生成一组**预测**，存储这些预测，然后根据需求检索这些预测。与**在线推断**相对。

### 独热编码 (one-hot encoding)

一种稀疏向量，其中：

- 一个元素设为 1。
- 所有其他元素均设为 0。

独热编码常用于表示拥有有限个可能值的字符串或标识符。例如，假设某个指定的植物学数据集记录了 15000 个不同的物种，其中每个物种都用独一无二的字符串标识符来表示。在特征工程过程中，您可能需要将这些字符串标识符编码为独热向量，向量的大小为 15000。

### 单样本学习 (one-shot learning, 通常用于对象分类)



一种机器学习方法，通常用于对象分类，旨在通过单个训练样本学习有效的分类器。

另请参阅[少量样本学习](#)。

## 一对多 (one-vs.-all)

假设某个分类问题有  $N$  种可能的解决方案，一对多解决方案将包含  $N$  个单独的[二元分类器](#) - 一个二元分类器对应一种可能的结果。例如，假设某个模型用于区分样本属于动物、蔬菜还是矿物，一对多解决方案将提供下列三个单独的二元分类器：

- 动物和非动物
- 蔬菜和非蔬菜
- 矿物和非矿物

## 在线推断 (online inference)

根据需求生成[预测](#)。与[离线推断](#)相对。

## 操作 (op, Operation)

TensorFlow 图中的节点。在 TensorFlow 中，任何创建、操纵或销毁[张量](#)的过程都属于操作。例如，矩阵相乘就是一种操作，该操作以两个张量作为输入，并生成一个张量作为输出。

## 优化器 (optimizer)

[梯度下降法](#)的一种具体实现。TensorFlow 的优化器基类是[tf.train.Optimizer<sup>\[4\]</sup>](#)。不同的优化器可能会利用以下一个或多个概念来增强梯度下降法在指定[训练集](#)中的效果：

- [动量<sup>\[5\]</sup>](#) (Momentum)
- 更新频率 ( [AdaGrad<sup>\[6\]</sup>](#) = ADaptive GRADient descent; [Adam<sup>\[7\]</sup>](#) = ADaptive with Momentum; RMSProp)
- 稀疏性/正则化 ( [Ftrl<sup>\[8\]</sup>](#) )
- 更复杂的数学方法 ( [Proximal<sup>\[9\]</sup>](#) , 等等)

甚至还包括 [NN 驱动的优化器<sup>\[10\]</sup>](#)。

## 离群值 (outlier)

与大多数其他值差别很大的值。在机器学习中，下列所有值都是离群值。

- 绝对值很高的**权重**。
- 与实际值相差很大的预测值。
- 值比平均值高大约 3 个标准偏差的输入数据。

离群值常常会导致模型训练出现问题。

## 输出层 (output layer)

神经网络的"最后"一层，也是包含答案的层。

## 过拟合 (overfitting)

创建的模型与**训练数据**过于匹配，以致于模型无法根据新数据做出正确的预测。

## 5 P

### Pandas

面向列的数据分析 API。很多机器学习框架（包括 TensorFlow）都支持将 Pandas 数据结构作为输入。请参阅 [Pandas 文档<sup>\[11\]</sup>](#)。

## 参数 (parameter)

机器学习系统自行训练的模型的变量。例如，**权重**就是一种参数，它们的值是机器学习系统通过连续的训练迭代逐渐学习到的。与**超参数**相对。

## 参数服务器 (PS, Parameter Server)

一种作业，负责在分布式设置中跟踪模型**参数**。

## 参数更新 (parameter update)

在训练期间（通常是在**梯度下降法**的单次迭代中）调整模型**参数**的操作。

## 偏导数 (partial derivative)

一种导数，除一个变量之外的所有变量都被视为常量。例如， $f(x, y)$  对  $x$  的偏导数就是  $f(x)$  的导数（即，使  $y$  保持恒定）。 $f$  对  $x$  的偏导数仅关注  $x$  如何变化，而忽略公式中的所有其他变量。

## 划分策略 (partitioning strategy)

在**参数服务器**间分割变量的算法。

## 性能 (performance)

多含义术语，具有以下含义：

- 在软件工程中的传统含义。即：相应软件的运行速度有多快（或有多高效）？
- 在机器学习中的含义。在机器学习领域，性能旨在回答以下问题：相应**模型**的准确度有多高？即模型在预测方面的表现有多好？

## 困惑度 (perplexity)

一种衡量指标，用于衡量**模型**能够多好地完成任任务。例如，假设任务是读取用户使用智能手机键盘输入字词时输入的前几个字母，然后列出一组可能的完整字词。此任务的困惑度 (P) 是：为了使列出的字词中包含用户尝试输入的实际字词，您需要提供的猜测项的个数。

困惑度与**交叉熵**的关系如下：

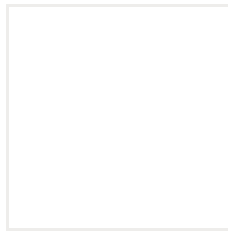
$$P = 2^{-\text{cross entropy}}$$

## 流水线 (pipeline)

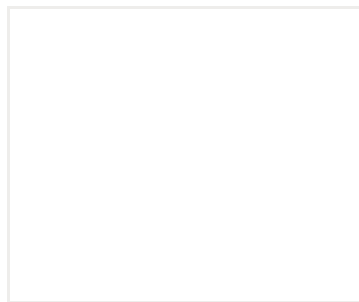
机器学习算法的基础架构。流水线包括收集数据、将数据放入训练数据文件、训练一个或多个模型，以及将模型导出到生产环境。

## 池化 (pooling)

将一个或多个由前趋的**卷积层**创建的矩阵压缩为较小的矩阵。池化通常是取整个池化区域的最大值或平均值。以下面的 3x3 矩阵为例：



池化运算与卷积运算类似：将矩阵分割为多个切片，然后按**步长**逐个运行卷积运算。例如，假设池化运算按 1x1 步长将卷积矩阵分割为 2x2 个切片。如下图所示，进行了四个池化运算。假设每个池化运算都选择该切片中四个值的最大值：



池化有助于在输入矩阵中实现**平移不变性**。

对于视觉应用来说，池化的更正式名称为**空间池化**。时间序列应用通常将池化称为**时序池化**。按照不太正式的说法，池化通常称为**下采样**或**降采样**。

## 正类别 (positive class)

在**二元分类**中，两种可能的类别分别被标记为正类别和负类别。正类别结果是我们要测试的对象。（不可否认的是，我们会同时测试这两种结果，但只关注正类别结果。）例如，在医学检查中，正类别可以是"肿瘤"。在电子邮件分类器中，正类别可以是"垃圾邮件"。

与**负类别**相对。

## 精确率 (precision)

一种**分类模型**指标。精确率指模型正确预测**正类别**的频率，即：

$$\text{精确率} = \frac{\text{正例数}}{\text{正例数} + \text{假正例数}}$$

## 预测 (prediction)

模型在收到输入 **样本** 后的输出。

## 预测偏差 (prediction bias)

一种值，用于表明 **预测** 平均值与数据集中 **标签** 的平均值相差有多大。

## 预创建的 Estimator (pre-made Estimator)

其他人已建好的 **Estimator**。TensorFlow 提供了一些预创建的 Estimator，包括 **DNNClassifier**、**DNNRegressor** 和 **LinearClassifier**。您可以按照 [这些说明](#)<sup>[12]</sup> 构建自己预创建的 Estimator。

## 预训练模型 (pre-trained model)

已经过训练的模型或模型组件（例如 **嵌套**）。有时，您需要将预训练的嵌套馈送到 **神经网络**。在其他时候，您的模型将自行训练嵌套，而不依赖于预训练的嵌套。

## 先验信念 (prior belief)

在开始采用相应数据进行训练之前，您对这些数据抱有的信念。例如， **$L_2$  正则化** 依赖的先验信念是 **权重** 应该很小且应以 0 为中心呈正态分布。

## 6 Q

## 队列 (queue)

一种 TensorFlow **操作**，用于实现队列数据结构。通常用于 I/O 中。

## 7 R

## 等级 (rank)

机器学习中的一个多含义术语，可以理解为下列含义之一：

- **张量**中的维数。例如，标量等级为 0，向量等级为 1，矩阵等级为 2。
- 在将类别从最高到最低进行排序的机器学习问题中，类别的顺序位置。例如，行为排序系统可以将狗狗的奖励从最高（牛排）到最低（枯萎的羽衣甘蓝）进行排序。

## 评分者 (rater)

为**样本**提供**标签**的人。有时称为"注释者"。

## 召回率 (recall)

一种**分类模型**指标，用于回答以下问题：在所有可能的正类别标签中，模型正确地识别出了多少个？即：

$$\text{召回率} = \frac{\text{正例数}}{\text{正例数} + \text{假负例数}}$$

## 修正线性单元 (ReLU, Rectified Linear Unit)

一种**激活函数**，其规则如下：

- 如果输入为负数或 0，则输出 0。
- 如果输入为正数，则输出等于输入。

## 回归模型 (regression model)

一种模型，能够输出连续的值（通常为浮点值）。请与**分类模型**进行比较，分类模型会输出离散值，例如"黄花菜"或"虎皮百合"。

## 正则化 (regularization)

对模型复杂度的惩罚。正则化有助于防止出现**过拟合**，包含以下类型：

- $L_1$  正则化
- $L_2$  正则化
- 丢弃正则化
- **早停法**（这不是正式的正则化方法，但可以有效限制过拟合）

## 正则化率 (regularization rate)

一种标量值，以  $\lambda$  表示，用于指定正则化函数的相对重要性。从下面简化的损失公式中可以看出正则化率的影响：

$$\text{最小化}(\text{损失方程} + \lambda(\text{正则化方程}))$$

提高正则化率可以减少过拟合，但可能会使模型的准确率降低。

## 表示法 (representation)

将数据映射到实用特征的过程。

## 受试者工作特征曲线 (receiver operating characteristic, 简称 ROC 曲线)

不同分类阈值下的正例率和假正例率构成的曲线。另请参阅曲线下面积。

## 根目录 (root directory)

您指定的目录，用于托管多个模型的 TensorFlow 检查点和事件文件的子目录。

## 均方根误差 (RMSE, Root Mean Squared Error)

均方误差的平方根。

## 旋转不变性 (rotational invariance)

在图像分类问题中，即使图像的方向发生变化，算法也能成功地对图像进行分类。例如，无论网球拍朝上、侧向还是朝下放置，该算法仍然可以识别它。请注意，并非总是希望旋转不变；例如，倒置的"9"不应分类为"9"。

另请参阅平移不变性和大小不变性。

## SavedModel

保存和恢复 TensorFlow 模型时建议使用的格式。SavedModel 是一种独立于语言且可恢复的序列化格式，使较高级别的系统和工具可以创建、使用和转换 TensorFlow 模型。

如需完整的详细信息，请参阅《TensorFlow 编程人员指南》中的[保存和恢复](#)<sup>[13]</sup>。

## Saver

一种 TensorFlow 对象，负责保存模型检查点。

## 缩放 (scaling)

**特征工程**中的一种常用做法，是指对某个特征的值区间进行调整，使之与数据集中其他特征的值区间一致。例如，假设您希望数据集中所有浮点特征的值都位于 0 到 1 区间内，如果某个特征的值位于 0 到 500 区间内，您就可以通过将每个值除以 500 来缩放该特征。

另请参阅[标准化](#)。

## scikit-learn

一个热门的开放源代码机器学习平台。请访问[www.scikit-learn.org](http://www.scikit-learn.org)<sup>[14]</sup>。

## 半监督式学习 (semi-supervised learning)

训练模型时采用的数据中，某些训练样本有标签，而其他样本则没有标签。半监督式学习采用的一种技术是推断无标签样本的标签，然后使用推断出的标签进行训练，以创建新模型。如果获得有标签样本需要高昂的成本，而无标签样本则有很多，那么半监督式学习将非常有用。

## 序列模型 (sequence model)

一种模型，其输入具有序列依赖性。例如，根据之前观看过的一系列视频对观看的下一个视频进行预测。

## 会话 (tf.session)



封装了 TensorFlow 运行时状态的对象，用于运行全部或部分图。在使用底层 TensorFlow API 时，您可以直接创建并管理一个或多个 `tf.session` 对象。在使用 Estimator API 时，Estimator 会为您创建会话对象。

## S 型函数 (sigmoid function)

一种函数，可将逻辑回归输出或多项回归输出（对数几率）映射到概率，以返回介于 0 到 1 之间的值。S 型函数的公式如下：

$$y = \frac{1}{1 + e^{-\sigma}}$$

在逻辑回归问题中， $\sigma$  非常简单：

$$\sigma = b + w_1x_1 + w_2x_2 + \dots + w_nx_n$$

换句话说，S 型函数可将  $\sigma$  转换为介于 0 到 1 之间的概率。

在某些神经网络中，S 型函数可作为激活函数使用。

## 大小不变性 (size invariance)

在图像分类问题中，即使图像的大小发生变化，算法也能成功地对图像进行分类。例如，无论一只猫以 200 万像素还是 20 万像素呈现，该算法仍然可以识别它。请注意，即使是最好的图像分类算法，在大小不变性方面仍然会存在切实的限制。例如，对于仅以 20 像素呈现的猫图像，算法（或人）不可能正确对其进行分类。

另请参阅[平移不变性](#)和[旋转不变性](#)。

## softmax

一种函数，可提供多类别分类模型中每个可能类别的概率。这些概率的总和正好为 1.0。例如，softmax 可能会得出某个图像是狗、猫和马的概率分别是 0.9、0.08 和 0.02。（也称为完整 softmax。）

与[候选采样](#)相对。

## 稀疏特征 (sparse feature)

一种**特征**向量，其中的大多数值都为 0 或为空。例如，某个向量包含一个为 1 的值和一百万个为 0 的值，则该向量就属于稀疏向量。再举一个例子，搜索查询中的单词也可能属于稀疏特征

- 在某种指定语言中有很多可能的单词，但在某个指定的查询中仅包含其中几个。

与**密集特征**相对。

## 稀疏表示法 (sparse representation)

一种张量**表示法**，仅存储非零元素。

例如，英语中包含约一百万个单词。表示一个英语句子中所用单词的数量，考虑以下两种方式：

- 要采用**密集表示法**来表示此句子，则必须为所有一百万个单元格设置一个整数，然后在大部分单元格中放入 0，在少数单元格中放入一个非常小的整数。
- 要采用稀疏表示法来表示此句子，则仅存储象征句子中实际存在的单词的单元格。因此，如果句子只包含 20 个独一无二的单词，那么该句子的稀疏表示法将仅在 20 个单元格中存储一个整数。

例如，假设以两种方式来表示句子"Dogs wag tails."。如下表所示，密集表示法将使用约一百万个单元格；稀疏表示法则只使用 3 个单元格：

## 稀疏性 (sparsity)

向量或矩阵中设置为 0（或空）的元素数除以该向量或矩阵中的条目总数。以一个 10x10 矩阵（其中 98 个单元格都包含 0）为例。稀疏性的计算方法如下：

$$\text{稀疏性} = \frac{98}{100} = 0.98$$

**特征稀疏性**是指特征向量的稀疏性；**模型稀疏性**是指模型权重的稀疏性。

## 空间池化 (spatial pooling)

请参阅**池化**。

## 平方合页损失函数 (squared hinge loss)

**合页损失函数**的平方。与常规合页损失函数相比，平方合页损失函数对离群值的惩罚更严厉。

## 平方损失函数 (squared loss)

在**线性回归**中使用的**损失函数**（也称为  $L_2$  **损失函数**）。该函数可计算模型为有标签**样本**预测的值和**标签**的实际值之差的平方。由于取平方值，因此该损失函数会放大不佳预测的影响。也就是说，与  $L_1$  **损失函数**相比，平方损失函数对离群值的反应更强烈。

## 静态模型 (static model)

离线训练的一种模型。

## 平稳性 (stationarity)

数据集中数据的一种属性，表示数据分布在一个或多个维度保持不变。这种维度最常见的是时间，即表明平稳性的数据不随时间而变化。例如，从 9 月到 12 月，表明平稳性的数据没有发生变化。

## 步 (step)

对一个**批次**的向前和向后评估。

## 步长 (step size)

与**学习速率**的含义相同。

## 随机梯度下降法 (SGD, stochastic gradient descent)

批次大小为 1 的一种**梯度下降法**。换句话说，SGD 依赖于从数据集中随机均匀选择的单个样本来计算每步的梯度估算值。

## 结构风险最小化 (SRM, structural risk minimization)

一种算法，用于平衡以下两个目标：

- 期望构建最具预测性的模型（例如损失最低）。
- 期望使模型尽可能简单（例如强大的正则化）。

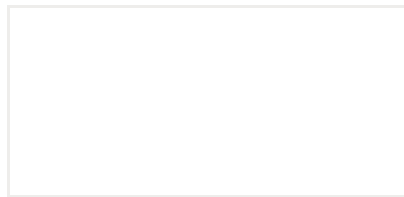
例如，旨在将基于训练集的损失和正则化降至最低的函数就是一种结构风险最小化算法。

如需更多信息，请参阅 <http://www.svms.org/srm/>。

与 **经验风险最小化** 相对。

## **步长 (stride)**

在卷积运算或池化中，下一个系列的输入切片的每个维度中的增量。例如，下面的动画演示了卷积运算过程中的一个 (1,1) 步长。因此，下一个输入切片是从上一个输入切片向右移动一个步长的位置开始。当运算到达右侧边缘时，下一个切片将回到最左边，但是下移一个位置。



前面的示例演示了一个二维步长。如果输入矩阵为三维，那么步长也将是三维。

## **下采样 (subsampling)**

请参阅 **池化**。

## **总结 (summary)**

在 TensorFlow 中的某一步计算出的一个值或一组值，通常用于在训练期间跟踪模型指标。

## **监督式机器学习 (supervised machine learning)**

根据输入数据及其对应的 **标签** 来训练 **模型**。监督式机器学习类似于学生通过研究一系列问题及其对应的答案来学习某个主题。在掌握了问题和答案之间的对应关系后，学生便可以回答关于同一主题的新问题（以前从未见过的问题）。请与 **非监督式机器学习** 进行比较。

## **合成特征 (synthetic feature)**

一种 **特征**，不在输入特征之列，而是从一个或多个输入特征衍生而来。合成特征包括以下类型：

- 对连续特征进行**分桶**，以分为多个区间分箱。
- 将一个特征值与其他特征值或其本身相乘（或相除）。
- 创建一个**特征组合**。

仅通过**标准化**或**缩放**创建的特征不属于合成特征。

## 9 T

---

### 目标 (target)

与**标签**的含义相同。

### 时态数据 (temporal data)

在不同时间点记录的数据。例如，记录的一年中每一天的冬外套销量就属于时态数据。

### 张量 (Tensor)

TensorFlow 程序中的主要数据结构。张量是 N 维（其中 N 可能非常大）数据结构，最常见的是标量、向量或矩阵。张量的元素可以包含整数值、浮点值或字符串值。

### 张量处理单元 (TPU, Tensor Processing Unit)

一种 ASIC（应用专用集成电路），用于优化 TensorFlow 程序的性能。

### 张量等级 (Tensor rank)

请参阅**等级**。

### 张量形状 (Tensor shape)

**张量**在各种维度中包含的元素数。例如，张量 [5, 10] 在一个维度中的形状为 5，在另一个维度中的形状为 10。

### 张量大小 (Tensor size)

**张量**包含的标量总数。例如，张量 [5, 10] 的大小为 50。

## **TensorBoard**

一个信息中心，用于显示在执行一个或多个 TensorFlow 程序期间保存的摘要信息。

## **TensorFlow**

一个大型的分布式机器学习平台。该术语还指 TensorFlow 堆栈中的基本 API 层，该层支持对数据流图进行一般计算。

虽然 TensorFlow 主要应用于机器学习领域，但也可用于需要使用数据流图进行数值计算的非机器学习任务。

## **TensorFlow Playground**

一款用于直观呈现不同的**超参数**对模型（主要是神经网络）训练的影响的程序。要试用 TensorFlow Playground，请前往<http://playground.tensorflow.org><sup>[15]</sup>。

## **TensorFlow Serving**

一个平台，用于将训练过的模型部署到生产环境。

## **测试集 (test set)**

数据集的子集，用于在**模型**经由验证集的初步验证之后测试模型。

与**训练集**和**验证集**相对。

## **tf.Example**

一种标准**协议缓冲区**<sup>[16]</sup>，旨在描述用于机器学习模型训练或推断的输入数据。

## **时间序列分析 (time series analysis)**

机器学习和统计学的一个子领域，旨在分析**时态数据**。很多类型的机器学习问题都需要时间序列分析，其中包括分类、聚类、预测和异常检测。例如，您可以利用时间序列分析根

据历史销量数据预测未来每月的冬外套销量。

## **训练 (training)**

确定构成模型的理想[参数](#)的过程。

## **训练集 (training set)**

数据集的子集，用于训练模型。

与[验证集](#)和[测试集](#)相对。

## **迁移学习 (transfer learning)**

将信息从一个机器学习任务迁移到另一个机器学习任务。例如，在多任务学习中，一个模型可以完成多项任务，例如针对不同任务具有不同输出节点的[深度模型](#)。迁移学习可能涉及将知识从较简单任务的解决方案迁移到较复杂的任务，或者将知识从数据较多的任务迁移到数据较少的任务。

大多数机器学习系统都只能完成一项任务。迁移学习是迈向人工智能的一小步；在人工智能中，单个程序可以完成多项任务。

## **平移不变性 (translational invariance)**

在图像分类问题中，即使图像中对象的位置发生变化，算法也能成功对图像进行分类。例如，无论一只狗位于画面正中央还是画面左侧，该算法仍然可以识别它。

另请参阅[大小不变性](#)和[旋转不变性](#)。

## **负例 (TN, true negative)**

被模型正确地预测为[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，而该电子邮件确实不是垃圾邮件。

## **正例 (TP, true positive)**

被模型正确地预测为[正类别](#)的样本。例如，模型推断出某封电子邮件是垃圾邮件，而该电



子邮件确实是垃圾邮件。

## 正例率 (true positive rate, 简称 TP 率)

与召回率的含义相同，即：

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

正例率是 ROC 曲线的 y 轴。

## 10 U

### 无标签样本 (unlabeled example)

包含特征但没有标签的样本。无标签样本是用于进行推断的输入内容。在半监督式和非监督式学习中，在训练期间会使用无标签样本。

### 非监督式机器学习 (unsupervised machine learning)

训练模型，以找出数据集（通常是无标签数据集）中的规律。

非监督式机器学习最常见的用途是将数据分为不同的聚类，使相似的样本位于同一组中。例如，非监督式机器学习算法可以根据音乐的各种属性将歌曲分为不同的聚类。所得聚类可以作为其他机器学习算法（例如音乐推荐服务）的输入。在很难获取真标签的领域，聚类可能会非常有用。例如，在反滥用和反欺诈等领域，聚类有助于人们更好地了解相关数据。

非监督式机器学习的另一个例子是主成分分析 (PCA)。例如，通过对包含数百万购物车中物品的数据集进行主成分分析，可能会发现有柠檬的购物车中往往也有抗酸药。

请与监督式机器学习进行比较。

## 11 V

## 验证集 (validation set)

数据集的一个子集，从训练集分离而来，用于调整超参数。

与训练集和测试集相对。

## 12 W

## 权重 (weight)

线性模型中特征的系数，或深度网络中的边。训练线性模型的目标是确定每个特征的理想权重。如果权重为 0，则相应的特征对模型来说没有任何贡献。

## 宽度模型 (wide model)

一种线性模型，通常有很多稀疏输入特征。我们之所以称之为"宽度模型"，是因为这是一种特殊类型的神经网络，其大量输入均直接与输出节点相连。与深度模型相比，宽度模型通常更易于调试和检查。虽然宽度模型无法通过隐藏层来表示非线性关系，但可以利用特征组合、分桶等转换以不同的方式为非线性关系建模。

与深度模型相对。

全文完，查看原文请前往[这里<sup>\[17\]</sup>](#)。

## 参考资料

- [1] sigmoid cross entropy: [https://www.tensorflow.org/api\\_docs/python/tf/nn/sigmoid\\_cross\\_entropy\\_with\\_logits?hl=zh-CN](https://www.tensorflow.org/api_docs/python/tf/nn/sigmoid_cross_entropy_with_logits?hl=zh-CN)
- [2] NaN: <https://en.wikipedia.org/wiki/NaN>
- [3] 开放源代码数学库: <http://www.numpy.org/>
- [4] tf.train.Optimizer: [https://www.tensorflow.org/api\\_docs/python/tf/train/Optimizer?hl=zh-CN](https://www.tensorflow.org/api_docs/python/tf/train/Optimizer?hl=zh-CN)
- [5] 动量: [https://www.tensorflow.org/api\\_docs/python/tf/train/MomentumOptimizer?hl=zh-CN](https://www.tensorflow.org/api_docs/python/tf/train/MomentumOptimizer?hl=zh-CN)

- [6] AdaGrad: [https://www.tensorflow.org/api\\_docs/python/tf/train/AdagradOptimizer?hl=zh-CN](https://www.tensorflow.org/api_docs/python/tf/train/AdagradOptimizer?hl=zh-CN)
- [7] Adam: [https://www.tensorflow.org/api\\_docs/python/tf/train/AdamOptimizer?hl=zh-CN](https://www.tensorflow.org/api_docs/python/tf/train/AdamOptimizer?hl=zh-CN)
- [8] Ftrl: [https://www.tensorflow.org/api\\_docs/python/tf/train/FtrlOptimizer?hl=zh-CN](https://www.tensorflow.org/api_docs/python/tf/train/FtrlOptimizer?hl=zh-CN)
- [9] Proximal: [https://www.tensorflow.org/api\\_docs/python/tf/train/ProximalGradientDescentOptimizer?hl=zh-CN](https://www.tensorflow.org/api_docs/python/tf/train/ProximalGradientDescentOptimizer?hl=zh-CN)
- [10] NN 驱动的优化器: <https://arxiv.org/abs/1606.04474>
- [11] Pandas 文档: <http://pandas.pydata.org/>
- [12] 自定义Estimator说明: <https://www.tensorflow.org/extend/estimators?hl=zh-CN>
- [13] 保存和恢复: [https://www.tensorflow.org/programmers\\_guide/saved\\_model?hl=zh-CN](https://www.tensorflow.org/programmers_guide/saved_model?hl=zh-CN)
- [14] [www.scikit-learn.org](http://www.scikit-learn.org/): <http://www.scikit-learn.org/>
- [15] [http://playground.tensorflow.org](http://playground.tensorflow.org/): <http://playground.tensorflow.org?hl=zh-CN>
- [16] 协议缓冲区: <https://developers.google.com/protocol-buffers/?hl=zh-CN>
- [17] 原文链接: <https://developers.google.com/machine-learning/glossary/?hl=zh-CN>