

# 1 数据分析与处理

## 数据清洗方法包括哪些？

数据清洗方法有：

1 数据清洗是指发现并纠正数据文件中可识别的错误的最后一道程序，包括检查数据一致性，处理无效值和缺失值等。

2 数据清洗的主要包括：纠正错误、删除重复项、统一规格、修正逻辑、转换构造、数据压缩、补足残缺/空值、丢弃数据/变量。①查看数据描述：检查数据量和数据属性；属性的类型，值域区间，关联，数据的可获得性；从业务角度理解数据属性和属性值的含义；计算每个属性的统计信息（如最大值，最小值，均值和方差）②检测数据质量问题并修复：检查数据值是否有错误；有无缺失值；有无重复属性；检查数据值是否有异常值；值与属性本身的含义是否符合

数据清洗的主要任务：将文本拆成不同的属性，解决分隔符问题 补充缺失的数据 格式转换问题 异常值检测 同一实体不同表示的识别 如何清洗：查看数据描述 检测数据质量问题并修复 清洗什么数据：重复值 缺失值 异常值

## L1 范数和 L2 范数是什么？计算公式是什么？

**L1 范数是指向量中各个元素绝对值之和。**其作用也是可以提高模型参数的稀疏性，效果没有 L0 范数好，但是更容易求解，更常用。它也被称为曼哈顿距离，因为它类似于在网格状街道中行走的距离。

对于一个向量  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ ，L1范数定义为：

$$\|\mathbf{x}\|_1 = |x_1| + |x_2| + \dots + |x_n| = \sum_{i=1}^n |x_i|$$

**L2 范数是指向量各元素的平方和然后求平方根。**其作用是减小模型所有参数大小，可以防止模型过拟合，也很常用。它也被称为欧几里得范数，因为它表示了向量在欧几里得空间中的长度。

对于向量  $\mathbf{x} = (x_1, x_2, \dots, x_n)$ ，L2范数定义为：

$$\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} = \sqrt{\sum_{i=1}^n x_i^2}$$

## 数据归一化是什么？有什么意义？

归一化是数据预处理中的一种常用技术，旨在将数据按比例缩放，使之落入一个小的特定区间，通常是[0, 1]或[-1, 1]。这个过程对于许多机器学习算法来说是非常重要的，因为它可以帮助改善算法的收敛速度和性能，特别是在处理不同量纲或量级的特征时。

不同评价指标（即特征向量中的不同特征就是所述的不同评价指标）往往具有不同的量纲和量纲单位，这样的情况会影响到数据分析的结果，

为了消除指标之间的量纲影响，需要进行数据标准化处理，以解决数据指标之间的可比性。原始数据经过数据标准化处理后，各指标处于同一数量级，适合进行综合对比评价。其中，最典型的的就是数据的归一化处理。

### 归一化的作用？

**加速算法收敛：**许多机器学习算法，特别是基于梯度的优化算法（如梯度下降），在特征处于相似尺度时表现更好。归一化通过将所有特征缩放到相同的尺度（如[0, 1]或[-1, 1]），可以减少不同特征之间的尺度差异，从而加速算法的收敛速度。

**提高模型精度：**对于某些算法，如 K 近邻算法（KNN）和神经网络，特征的尺度对模型的性能有显著影响。在 KNN 中，距离的计算对特征的尺度敏感，而神经网络中的权重更新也受到特征尺度的影响。归一化可以帮助这些算法更准确地捕捉特征之间的关系，从而提高模型的精度。

**防止数值问题：**在某些计算过程中，如使用梯度下降算法时，如果特征的尺度差异很大，可能会导致数值不稳定或梯度消失/爆炸的问题。归一化有助于避免这类数值问题。

**提高算法稳定性：**对于某些算法，如支持向量机（SVM），数据的尺度可能会对其性能产生较大影响。归一化可以提高算法的稳定性，使得算法对于不同的数据集或数据子集具有更一致的性能

## 变量（如 SSS 和 SST）相关性衡量方法？（相关系数 R）

相关系数（皮尔森相关系数协方差/标准差相乘，越接近 1 越好）

皮尔逊相关系数：

$$R_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} = \frac{\sum_{i=1}^n (x_i - E(X))(y_i - E(Y))}{\sqrt{\sum_{i=1}^n (x_i - E(X))^2} \sqrt{\sum_{i=1}^n (y_i - E(Y))^2}}$$

式中， $x_i$  和  $y_i$  分别为 SSS 和 SST 的观测值； $E(X)$  和  $E(Y)$  分别为 SSS 和 SST 的均值； $n$  为观测点

数量。

$R \in [-1, 1]$ ;  $R=1$  代表完全正相关（线性增长）;  $R=-1$  代表完全负相关（线性下降）;  $R=0$  代表无线性相关。

图表相关分析（折线图及散点图），将数据进行可视化处理，简单的说就是绘制图表。单纯从数据的角度很难发现其中的趋势和联系，而将数据点绘制成图表后趋势和联系就会变的清晰起来。

协方差及协方差矩阵，协方差用来衡量两个变量的总体误差，如果两个变量的变化趋势一致，协方差就是正值，说明两个变量正相关。如果两个变量的变化趋势相反，协方差就是负值，说明两个变量负相关。如果两个变量相互独立，那么协方差就是 0，说明两个变量不相关。

相关系数，相关系数(Correlation coefficient)是反应变量之间关系密切程度的统计指标，相关系数的取值区间在 1 到-1 之间。1 表示两个变量完全线性相关，-1 表示两个变量完全负相关，0 表示两个变量不相关。数据越趋近于 0 表示相关关系越弱。以下是相关系数的计算公式。

$$r_{xy} = \frac{S_{xy}}{S_x S_y}$$

其中  $r_{xy}$  表示样本相关系数， $S_{xy}$  表示样本协方差， $S_x$  表示 X 的样本标准差， $S_y$  表示 y 的样本标准差。下面分别是  $S_{xy}$  协方差和  $S_x$  和  $S_y$  标准差的计算公式。由于是样本协方差和样本标准差，因此分母使用的是  $n-1$ 。

$S_{xy}$  样本协方差计算公式：

$$S_{xy} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1}$$

$S_x$  样本标准差计算公式：

$$S_x = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n - 1}}$$

$S_y$  样本标准差计算公式：

$$s_y = \sqrt{\frac{\sum (y_i - \bar{y})^2}{n - 1}}$$

一元回归及多元回归，回归分析（regression analysis）是确定两组或两组以上变量间关系的统计方法。回归分析按照变量的数量分为一元回归和多元回归。两个变量使用一元回归，两个以上变量使用多元回归。进行回归分析之前有两个准备工作，第一确定变量的数量。第二确定自变量和因变量。

信息熵及互信息，实际工作中影响最终效果的因素可能有很多，并且不一定是数值形式。比如我们站在更高的维度来看之前的数据。

### 了解数据增强：意义及方法。

在深度学习中，一般要求样本的数量要充足，样本数量越多，训练出来的模型效果越好，模型的泛化能力越强。但是实际中，**样本数量不足或者样本质量不够好，这就要对样本做数据增强，来提高样本质量。**

关于数据增强的作用总结如下：

- 1，增加训练的数据量，提高模型的泛化能力
- 2，增加噪声数据，提升模型的鲁棒性

1）**避免过拟合**。当数据集具有某种明显的特征，例如数据集中图片基本在同一个场景中拍摄，使用 Cutout 方法和风格迁移变化等相关方法可避免模型学到跟目标无关的信息。

2）**提升模型鲁棒性**，降低模型对图像的敏感度。当训练数据都属于比较理想的状态，碰到一些特殊情况，如遮挡，亮度，模糊等情况容易识别错误，对训练数据加上噪声，掩码等方法可提升模型鲁棒性。

- 3）**增加训练数据，提高模型泛化能力。**

4）**避免样本不均衡**。在工业缺陷检测方面，医疗疾病识别方面，容易出现正负样本极度不平衡的情况，通过对少样本进行一些数据增强方法，降低样本不均衡比例。

### 几何变换类

随机裁剪（Random Cropping）（从原始图像中随机裁剪出一部分作为新的训练样本），翻转（Flip），旋转（Rotation），缩放（Scaling），平移（Translation），仿射变换（Affine Transformation）（包括缩放、平移、旋转和剪切组合在一起的变换，保持图像的直线性），裁剪与填充（Crop and Pad）

### 颜色空间变换类

亮度调整（Brightness Adjustment），对比度调整（Contrast Adjustment），饱和度调整（Saturation Adjustment），色调变化（Hue Adjustment），色彩抖动（Color Jitter），灰度转换（Grayscale Conversion）（将彩色图像转换为灰度图像，减少色彩干扰），通道扰动（Channel Shuffle）（随机改变图像 RGB 通道的顺序（如交换 R 和 G 通道））

### 噪声与模糊类

添加噪声（Add Noise）（添加高斯噪声、椒盐噪声或泊松噪声，增强模型对噪声的鲁棒性），高斯模糊（Gaussian Blur）（使用高斯核对图像进行模糊处理，模拟焦点偏移），随机遮挡（Random Erasing）（随机在图像上擦除一块区域，增加模型对缺失数据的鲁棒性）Cutout（在图像上随机裁剪出小块区域并填充为 0 或其他固定值）Mixup（将两张图像按照权重进行线性组合，生成新的训练样本）Mosaic（将四张图像拼接成一张，扩充数据量并丰富背景信息）

### 高级数据增强方法

随机增强策略（AutoAugment）

通过搜索算法自动选择最优的增强策略，组合多种变换方法。

随机深度增强（RandAugment）

随机选择增强操作并控制其强度，减少搜索空间。

CutMix

在一张图像中裁剪出一个区域，并将另一张图像的对应区域填充进去。

Style Transfer（风格迁移）

使用生成对抗网络（GAN）改变图像风格，例如将夏季海洋影像转换为冬季风格。

## 2 算法及优化

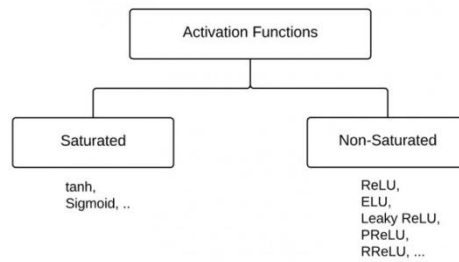
### 激活函数：类型及意义。

激活函数（Activation Function），就是在人工神经网络的神经元上运行的函数，负责将神经元的**输入映射到输出端**，旨在帮助网络学习数据中的复杂模式。

如果不用激活函数，每一层输出都是上层输入的线性函数，无论神经网络有多少层，输出都是输入的线性组合，这种情况就是最原始的感知机（Perceptron）。

使用激活函数能够给神经元引入非线性因素，使得神经网络可以任意逼近任何非线性函数，使深层神经网络表达能力更加强大，这样神经网络就可以应用到众多的非线性模型中。

激活函数可以分为两大类：



**饱和激活函数： sigmoid、 tanh...**

**非饱和激活函数： ReLU 、 Leaky Relu 、 ELU、 PReLU、 RReLU...**

首先，我们先了解一下什么是饱和？

假设  $h(x)$  是一个激活函数。

当我们的  $x$  趋近于正无穷，激活函数的导数趋近于 0，那么我们称之为右饱和。

$$\lim_{x \rightarrow +\infty} h'(x) = 0$$

当我们的  $x$  趋近于负无穷，激活函数的导数趋近于 0，那么我们称之为左饱和。

$$\lim_{x \rightarrow -\infty} h'(x) = 0$$

当一个函数既满足左饱和又满足右饱和的时候我们就称之为饱和，典型的函数有 Sigmoid, Tanh 函数。

反之，不满足以上条件的函数则称为非饱和激活函数。

Sigmoid 函数需要一个实值输入压缩至[0,1]的范围；tanh 函数需要讲一个实值输入压缩至 [-1, 1]的范围

相对于饱和激活函数，使用非饱和激活函数的优势在于两点：

- 1.非饱和激活函数能解决深度神经网络（层数非常多）带来的梯度消失问题
- 2.使用非饱和激活函数能加快收敛速度。

**Sigmoid**激活函数的数学表达式为：

$$f(x) = \frac{1}{1 + e^{-x}}$$

导数表达式为：

$$f'(x) = f(x)(1 - f(x))$$

容易造成梯度消失。我们从导函数图像中了解到 sigmoid 的导数都是小于 0.25 的，那么

在进行反向传播的时候，梯度相乘结果会慢慢的趋向于 0。这样几乎就没有梯度信号通过神经元传递到前面层的梯度更新中，因此这时前面层的权值几乎没有更新，这就叫梯度消失。除此之外，为了防止饱和，必须对于权重矩阵的初始化特别留意。如果初始化权重过大，可能很多神经元得到一个比较小的梯度，致使神经元不能很好的更新权重提前饱和，神经网络就几乎不学习。

函数输出不是以 0 为中心的，梯度可能就会向特定方向移动，从而降低权重更新的效率。

Sigmoid 函数执行指数运算，计算机运行得较慢，比较消耗计算资源。

tanh 激活函数的数学表达式为：

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$\tanh(x) = 2\text{sigmoid}(2x) - 1$$

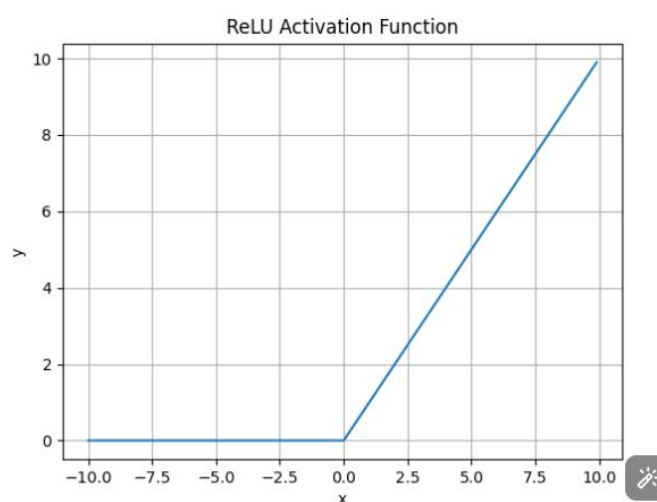
仍然存在梯度饱和的问题

依然进行的是指数运算

ReLU 激活函数的数学表达式为：

$$f(x) = \max(0, x)$$

函数图像如下：



ReLU 解决了梯度消失的问题，当输入值为正时，神经元不会饱和

由于 ReLU 线性、非饱和的性质，在 SGD 中能够快速收敛

计算复杂度低，不需要进行指数运算

与 Sigmoid 一样，其输出不是以 0 为中心的

Dead ReLU 问题。当输入为负时，梯度为 0。这个神经元及之后的神经元梯度永远为 0，不再对任何数据有所响应，导致相应参数永远不会被更新

训练神经网络的时候，一旦学习率没有设置好，第一次更新权重的时候，输入是负值，那么这个含有 ReLU 的神经节点就会死亡，再也不会被激活。所以，要设置一个合适的较小的学习率，来降低这种情况的发生

**Leaky Relu**激活函数的数学表达式为：

$$f(x) = \max(\alpha x, x)$$

**PRelu**激活函数的数学表达式为：

$$f(\alpha, x) = \begin{cases} \alpha x, & \text{for } x < 0 \\ x, & \text{for } x \geq 0 \end{cases}$$

**ELU**激活函数的数学表达式为：

$$f(\alpha, x) = \begin{cases} \alpha(e^x - 1), & \text{for } x \leq 0 \\ x, & \text{for } x > 0 \end{cases}$$

**SELU**激活函数的数学表达式为：

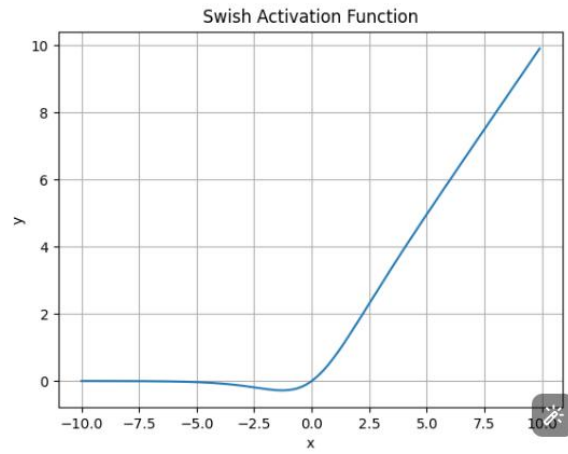
$$f(\alpha, x) = \lambda \begin{cases} \alpha(e^x - 1), & \text{for } x \leq 0 \\ x, & \text{for } x > 0 \end{cases}$$



Swish激活函数的数学表达式为：

$$f(x) = x * \text{sigmoid}(x)$$

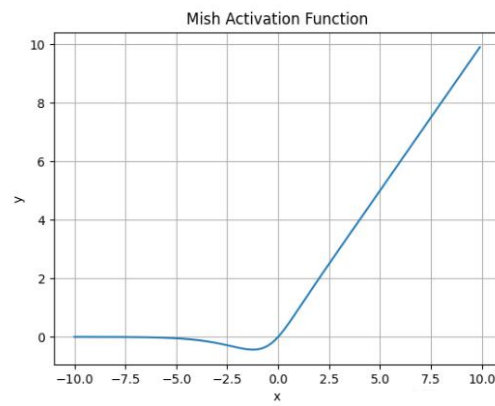
函数图像如下：



Mish激活函数的数学表达式为：

$$f(x) = x * \tanh(\ln(1 + e^x))$$

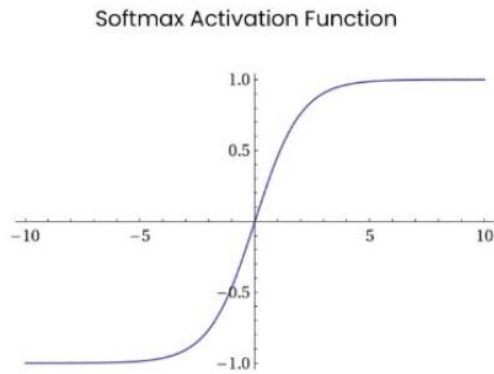
函数图像如下：



Softmax激活函数的数学表达式为：

$$\text{Softmax}(x) = \frac{e^{x_i}}{\sum_i e^{x_i}}$$

函数图像如下：



将神经元输出构造成概率分布，用于多分类问题中，Softmax 激活函数映射值越大，则真实类别可能性越大

[深度学习中常见的 10 种激活函数 \(Activation Function\) 总结 激活函数有哪些-CSDN 博客](#)

**池化层：类型及意义。**

池化层：池化（Pooling）：也称为欠采样或下采样。主要用于**特征降维**，压缩数据和参数的数量，减小过拟合，同时提高模型的容错性。主要有：最大池化和平均池化

1）一般池化（General Pooling）：

其中最常见的池化操作有平均池化、最大池化：

平均池化（average pooling）：计算图像区域的平均值作为该区域池化后的值。

最大池化（max pooling）：选图像区域的最大值作为该区域池化后的值。

（2）重叠池化（Overlapping Pooling）：

重叠池化就是，相邻池化窗口之间有重叠区域，此时一般  $\text{sizeX} > \text{stride}$ 。

（3）金字塔池化（Spatial Pyramid Pooling）

空间金字塔池化的思想源自 Spatial Pyramid Model，它将一个 pooling 变成了多个 scale 的 pooling。用不同大小池化窗口作用于上层的卷积特征。也就是说 spatial pyramid pooling layer 就是把前一卷积层的 feature maps 的每一个图片上进行了 3 个卷积操作，并把结果输出给全连接层。其中每一个 pool 操作可以看成是一个空间金字塔的一层。

池化层实际上是一种形式的降采样。有多种不同形式的非线性池化函数，而其中“最大池化（Max pooling）”是最为常见的。池化层会不断地减小数据的空间大小，因此参数的数量和计算量也会下降，这在一定程度上也控制了过拟合。通常来说，CNN 的卷积层之间都会周期性地插入池化层。

### 过拟合：成因及解决方法。

过拟合：是指学习时选择的模型所包含的参数过多，以至于出现这一模型对已知数据预测的很好，但对未知数据预测得很差的现象。这种情况下模型可能只是记住了训练集数据，而不是学习到了数据特征。可以通过**增加训练数据**，**降低模型复杂性**，对神经网络使用 **dropout** 等来应对过拟合问题。

欠拟合：模型描述能力太弱，以至于不能很好地学习到数据中的规律。产生欠拟合的原因通常是模型过于简单，可以采用调节模型大小，约束模型权重，即权重正则化（在机器学习中一般使用 L2 正则化），随机失活（Dropout）等来应对欠拟合问题

### 过拟合的解决方法：

1. **增加训练数据**：更多的数据有助于模型更好地泛化。
2. **正则化（Regularization）**：
  1. L1 正则化（Lasso）：通过惩罚模型参数，使不重要的特征权重趋于 0。
  2. L2 正则化（Ridge）：通过惩罚参数的平方，限制参数值的大小。
3. **使用 Dropout（神经网络）**：随机丢弃部分神经元，防止模型过度依赖某些特征。
4. **简化模型**：降低模型复杂度，减少参数数量。
5. **提前停止（Early Stopping）**：在验证误差不再下降时停止训练。
6. **数据增强**：增加数据的多样性，例如图像数据可以进行旋转、缩放等操作。

### 反向传播算法：意义

反向传播是神经网络的核心算法之一，用于通过误差反传调整网络参数，从而最小化损失函数。它是一种基于链式法则的高效梯度计算方法，是训练神经网络的关键步骤。

反向传播算法利用链式法则，通过从输出层向输入层逐层**计算误差梯度**，高效求解神经网络参数的**偏导数**，以实现网络参数的优化和损失函数的最小化。

利用链式法则：

反向传播算法基于微积分中的链式法则，通过逐层计算梯度来求解神经网络中参数的偏导数。

从输出层向输入层传播：

算法从输出层开始，根据损失函数计算输出层的误差，然后将误差信息反向传播到隐藏层，逐层计算每个神经元的误差梯度。

计算权重和偏置的梯度：

利用计算得到的误差梯度，可以进一步计算每个权重和偏置参数对于损失函数的梯度。

参数更新：

根据计算得到的梯度信息，使用梯度下降或其他优化算法来更新网络中的权重和偏置参数，以最小化损失函数。

优化算法有哪些？

优化算法：使用反向传播提供的梯度来更新神经网络的参数，进而最小化损失函数。

优化算法就是一种能够帮我们最小化或者最大化目标函数  $Q$ （有时候也叫损失函数）的一类算法。而目标函数往往是模型参数和数据的数学组合。例如给定数据  $X$  和其对应的标签  $Y$ ，我们构建的模型是一个线性模型  $f(x) = Wx + b$ ，有了模型后，根据输入  $a$  就可以得到预测输出  $f(x)$ ，并且可以计算出预测值和真实值之间的差距  $(f(x) - Y)^2$ ，这个就是损失函数。我们的目的是找到合适的  $W, b$  使损失函数的值达到最小，损失值越小，则说明我们的模型越接近于真实情况。

通过上面的描述可以知道，模型的内参  $(W, b)$  在模型中扮演若非常重要的角色，而这些内参的更新和优化就用到我们所说的优化算法，所以优化算法也层出不穷，而一个好的优化算法往往能够更加高效、更加准确的训练模型的内参。

梯度下降法（Gradient Descent），随机梯度下降法（SGD），带动量的梯度下降法（Momentum），Adam 优化算法，Adagrad、RMSprop 等

算法	特点	适用场景
SGD	基本梯度下降，计算简单，但收敛慢	适用于小规模数据
Momentum	引入动量，加速收敛，减少震荡	适用于深度网络训练
AdaGrad	自适应学习率，但学习率持续减小	稀疏数据和文本数据
RMSProp	平滑自适应学习率，适合非平稳目标函数	神经网络训练
Adam	结合 Momentum 和 RMSProp，收敛快，效果好	常用于大部分深度学习任务
AdamW	在 Adam 基础上加入权重衰减，改善正则化性能	防止过拟合，适用于大规模任务
NAG	使用 Nesterov 动量法，更新更平滑	需要更快收敛的任务

## 深度学习网络深度的意义？

深度学习中的“深度”指的是神经网络的层数，也就是从输入层到输出层所经历的层次数量。传统的机器学习算法一般采用浅层模型，例如线性回归、支持向量机（SVM）等，这些模型的表达能力有限，无法处理复杂的非线性问题。而深度学习的优势在于其能够通过增加神经网络的层数来**增强模型的表达能力**，从而更好地处理复杂的非线性问题。

深度学习的“深度”意义主要体现在以下几个方面：

### 增加模型的表达能力

增加神经网络的层数可以使得模型具备更强的表达能力。从数学角度来看，一个浅层模型只能拟合一个线性函数或者一个简单的非线性函数，而深度神经网络可以通过多个非线性函数的复合来逼近任意的非线性函数。因此，深度神经网络可以更好地处理复杂的非线性问题。

### 减少模型过拟合的风险

在机器学习中，过拟合是一个常见的问题。过拟合指的是模型在训练数据上表现很好，但是在测试数据上表现很差的现象。增加神经网络的层数可以使得模型更加复杂，从而降低模型在训练数据上过拟合的风险。同时，深度学习中的正则化技术和 dropout 技术等也可以有效地防止过拟合问题的出现。

### 更好地处理高维数据

随着数据量的不断增加，数据的维度也越来越高。高维数据给机器学习算法带来了很大的挑战。传统的机器学习算法往往只适用于低维数据，而在高维数据上表现很差。深度学习可以通过自动提取特征来处理高维数据，从而避免了手工设计特征的麻烦和困难。

### 提高模型的鲁棒性

在实际应用中，数据往往存在各种各样的噪声和异常情况。这些噪声和异常情况会对模型的性能产生很大的影响。深度学习中的一些技术，例如卷积神经网络（CNN）、循环神经网络（RNN）等，可以有效地抑制噪声和异常情况对模型性能的影响，从而提高模型的鲁棒性。

### 高效的学习能力

深度学习模型的学习能力非常强大。在很多情况下，深度学习模型可以在成千上万甚至更多的参数中进行学习，从而得到更加精确的模型预测结果。同时，深度学习还可以处理大规模的数据集，从而提高了模型的学习效率。

总之，深度学习的“深度”意义主要体现在**增加模型的表达能力、减少模型过拟合的风险、更好地处理高维数据、提高模型的鲁棒性以及高效的学习能力**等方面。正是由于这些

优势，深度学习在各个领域都得到了广泛的应用，例如语音识别、图像分类、自然语言处理、推荐系统等。然而，深度学习也面临着一些挑战和难点，例如模型复杂度高、训练时间长、数据量大等问题，需要我们不断地研究和探索新的技术和方法来克服这些困难，进一步发挥其应用价值。

## 迁移学习的意义？

迁移学习（Transfer Learning）是一种机器学习方法，是把一个领域（即源领域）的知识，迁移到另外一个领域（即目标领域），使得目标领域能够取得更好的学习效果。通常，**源领域数据量充足，而目标领域数据量较小**，这种场景就很适合做迁移学习，例如我们要对一个任务进行分类，但是此任务中数据不充足（目标域），然而却又大量的相关的训练数据（源域），但是此训练数据与所需进行的分类任务中的测试数据特征分布不同（例如语音情感识别中，一种语言的语音数据充足，然而所需进行分类任务的情感数据却极度缺乏），在这种情况下如果可以采用合适的迁移学习方法则可以大大提高样本不充足任务的分类识别结果。

迁移学习在人工智能领域有着广泛的应用，包括但不限于以下几个方面：

**数据稀缺情况下的学习：**在目标领域数据较少或标注困难的情况下，通过迁移学习可以利用源领域中的大量数据和知识，来辅助目标领域的学习，提高模型的泛化能力。

**领域自适应：**当源领域和目标领域的分布不一致时，通过迁移学习可以在不同领域之间进行知识迁移，使得模型更适应目标领域的分布，提高模型的适应性和泛化能力。

**模型初始化和预训练：**通过在大规模数据集上进行预训练，学习到的模型参数或特征表示可以作为目标任务的初始化参数，从而加速模型的训练和提高性能。

**跨模态学习：**在涉及多种数据类型的任务中，如图像和文本的跨模态学习，通过迁移学习可以将不同数据类型之间的知识进行有效整合和利用，提高模型的表现。

**增量学习：**在动态环境下，通过迁移学习可以在新任务或新数据到来时，利用已有模型的知识来快速适应新情况，实现增量学习和持续改进。

## 梯度下降法的实现原理。

比如常见的均方误差（Mean Squared Error）损失函数：

$$L(w, b) = \frac{1}{N} \sum_{i=1}^N (y_i - f(wx_i + b))^2$$

其中， $y_i$  表示样本数据的实际目标值， $f(wx_i + b)$  表示预测函数， $f$  根据样本数据  $x_i$  计算出的预测值。从几何意义上来说，它可以看成预测值和实际值的平均距离的平方。

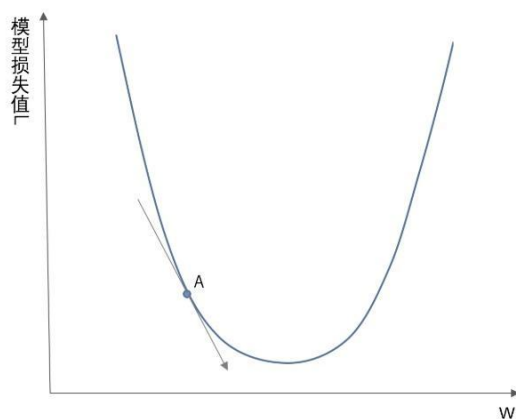
损失函数用来衡量机器学习模型的精确度。一般来说，损失函数的值越小，模型的精确度就越高。如果要提高机器学习模型的精确度，就需要尽可能降低损失函数的值。而降低损失函数的值，我们一般采用梯度下降这个方法。所以，**梯度下降的目的，就是为了最小化损失函数。**

### 梯度下降的原理

使用微积分里导数寻找损失函数的最低点，从而找到函数下降的方向或者是最低点（极值点）。

损失函数里一般有两种参数，一种是控制输入信号量的权重(Weight, 简称  $w$ )，另一种是调整函数与真实值距离的偏差 (Bias, 简称  $b$ )。我们所要做的工作，就是通过梯度下降方法，不断地调整权重  $w$  和偏差  $b$ ，使得损失函数的值变得越来越小。

假设某个损失函数里，模型损失值  $L$  与权重  $w$  有下图这样的关系。实际模型里，可能会有多个权重  $w$ ，这里为了简单起见，举只有一个权重  $w$  的例子。权重  $w$  目前的位置是在 A 点。此时如果求出 A 点的梯度  $\frac{dL}{dw}$ ，便可以知道如果我们向右移动，可以使损失函数的值变得更小。



通过计算梯度，我们就可以知道  $w$  的移动方向，应该让  $w$  向右走而不是向左走，也可以知道什么时候会到达最低点（梯度为 0 的地方）。

上面的例子里只出现了一个权重  $w$ ，实际的项目里样本数据会有很多个。对于每一个样本数据，我们都可以求出一个权重的梯度。这个时候，我们需要把各个样本数据的权重梯度加起来，并求出它们的平均值，用这个平均值来作为样本整体的权重梯度。

现在知道了  $w$  需要前进的方向，接下来需要知道应该前进多少。这里我们用到学习率

(Learning Rate)这个概念。通过学习率，可以计算前进的距离（步长）。

我们用  $w_i$  表示权重的初始值， $w_{i+1}$  表示更新后的权重值，用  $\alpha$  表示学习率，则有：

$w_{i+1} = w_i - \alpha * \frac{dL}{dw_i}$  (2) 在梯度下降中，我们会重复式子(2)多次，直至损失函数值收敛不变。

如果学习率  $\alpha$  设置得过大，有可能我们会错过损失函数的最小值；如果设置得过小，可能我们要迭代式子(2)非常多次才能找到最小值，会耗费较多的时间。因此，在实际应用中，我们需要为学习率  $\alpha$  设置一个合适的值。

## Dropout 避免过拟合的原理。

利用神经网络的分布式特征表达（只要能**保留核心特征**），既可以实现成功完成任务（例如成功识别图片为猫），还可以用来阻止过拟合的发生，分布式特征表达可称为 Dropout 的来源。“丢弃学习（Dropout，也有人称之为“随机失活”）”是指在深度学习网络的训练过程中，对于神经网络单元，按照一定的概率将其暂时从网络中丢弃。

Dropout 如何防止过拟合

### （1）数据层面

对于每一个 dropout 后的网络，进行训练时，相当于做了 **Data Augmentation**。比如，对于某一层，dropout 一些单元后，形成的结果是(1.5, 0, 2.5, 0, 1, 2, 0)，其中 0 是被 drop 的单元，那么总能找到一个样本，使得结果也是如此。这样每一次 dropout 其实都相当于增加了样本。

### （2）模型层面

2.1 在较大程度上**减小了网络的大小**：在这个“残缺”的网络中，让神经网络学习数据中的局部特征（即部分分布式特征），但这些特征也足以进行输出正确的结果。

2.2 Dropout 思想类似于集成学习中的 Bagging 思想：由学习阶段可知，每一次训练都会按  $keep\_probability=p$  来保留每个神经元，这意味着每次迭代过程中，随机删除一些神经元，这就意味着在多个“残缺”的神经网络中，每次都进行随机的特征选择，这要比仅在单个健全网络上进行特征学习，其**泛化能力**来得更加健壮。

由此思想可知如下两个作用：

取平均的作用：先回到正常的模型（没有 dropout），我们用相同的训练数据去训练 5 个不同的神经网络，一般会得到 5 个不同的结果，此时我们可以采用“5 个结果取均值”或者“多数取胜的投票策略”去决定最终结果。（例如 3 个网络判断结果为数字 9,那么很



有可能真正的结果就是数字 9，其它两个网络给出了错误结果）。这种“综合起来取平均”的策略通常可以有效防止过拟合问题。因为不同的网络可能产生不同的过拟合，取平均则有可能让一些“相反的”拟合互相抵消。**每次训练随机 dropout 掉不同的隐藏神经元，网络结构已经不同，这就类似在训练不同的网络，整个 dropout 过程就相当于对很多个不同的神经网络取平均。**而不同的网络产生不同的过拟合，一些互为“反向”的拟合相互抵消就可以达到整体上减少过拟合。

**减少神经元之间共适应关系：** 因为 dropout 导致两个神经元不一定每次都在一个网络中出现，这样权值的更新不再依赖于有固定关系的隐含节点的共同作用，**阻止了某些特征仅仅在其它特定特征下才有效果的情况，迫使网络去学习更加鲁棒的特征。**换句话说，假如神经网络是在做出某种预测，它不应该对一些特定的线索片段太过敏感，即使丢失特定的线索，它也应该可以从众多其它线索中学习一些共同的模式（鲁棒性）。（这个角度看 dropout 有点像 L1, L2 正则，减少权重，使得网络对丢失特定神经元连接的鲁棒性提高）

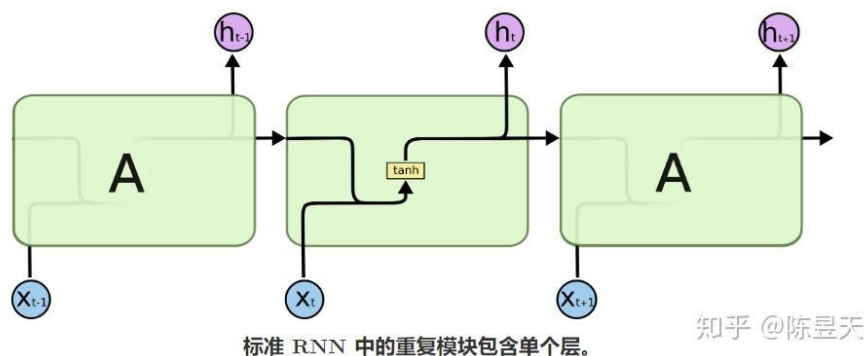
## 前馈神经网络、循环神经网络和卷积神经网络是什么？

**前馈神经网络**（Feedforward Neural Network, FNN）是最早发明的简单人工神经网络。在 FNN 中，不同的神经元属于不同的层，每一层的神经元可以接受到前一层的神元信号，并产生信号输出到下一层。因此，前馈神经网络也成为多层感知器（Mutlti-Layer Perceptron, MLP）。

FNN 的工作原理相对简单。它**从输入层接收输入信号，并通过多个隐藏层的处理将信号传递到输出层。**在每个隐藏层中，神经元会对输入信号进行线性组合和激活函数处理，以产生输出信号。最后，输出层将多个隐藏层的输出信号组合起来，产生最终的输出结果。

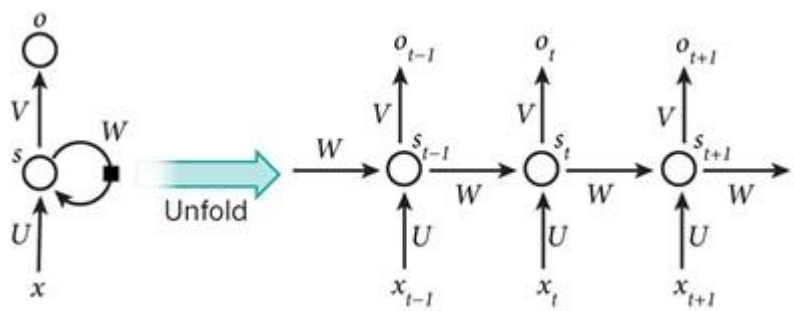
FNN 在许多领域都有广泛的应用，例如分类、回归和聚类等。由于其结构简单，易于训练和实现，FNN 成为深度学习的基石之一。

**循环神经网络**（Recurrent Neural Network, RNN）是一种用于处理序列数据的神经网络。RNN 通过在时间维度上展开神经网络，使得同一层神经元之间的连接具有时序依赖性，从而能够捕捉序列数据中的时间依赖关系。



循环神经网络(Recurrent Neural Network, RNN)一般是指时间递归神经网络而非结构递归神经网络 (Recursive Neural Network)，其主要用于对序列数据进行建模。

RNN 之所以称为循环神经网络，即一个序列当前的输出与前面的输出也有关。具体的表现形式为网络会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点不再无连接而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出。理论上，RNN 能够对任何长度的序列数据进行处理。但是在实践中，为了降低复杂性往往假设当前的状态只与前面的几个状态相关。



$x$  是输入层的值。 $s$  表示隐藏层的值， $U$  是输入层到隐藏层的权重矩阵， $O$  是输出层的值。 $V$  是隐藏层到输出层的权重矩阵。循环神经网络的隐藏层的值  $s$  不仅仅取决于当前这次的输入  $x$ ，还取决于上一次隐藏层的值  $s$ 。权重矩阵  $W$  就是隐藏层上一次的值作为这一次的输入的权重。

RNN 由输入层、隐藏层和输出层组成。在隐藏层中，每个神经元接收前一时刻的输出作为输入，并产生当前时刻的输出。通过这种方式，RNN 能够将之前的信息传递到后续的时刻，从而实现序列数据的处理。

RNN 在自然语言处理领域有着广泛的应用，例如文本生成、语音识别和机器翻译等。此外，RNN 也被应用于时间序列分析、音乐生成和情感分析等其他领域。

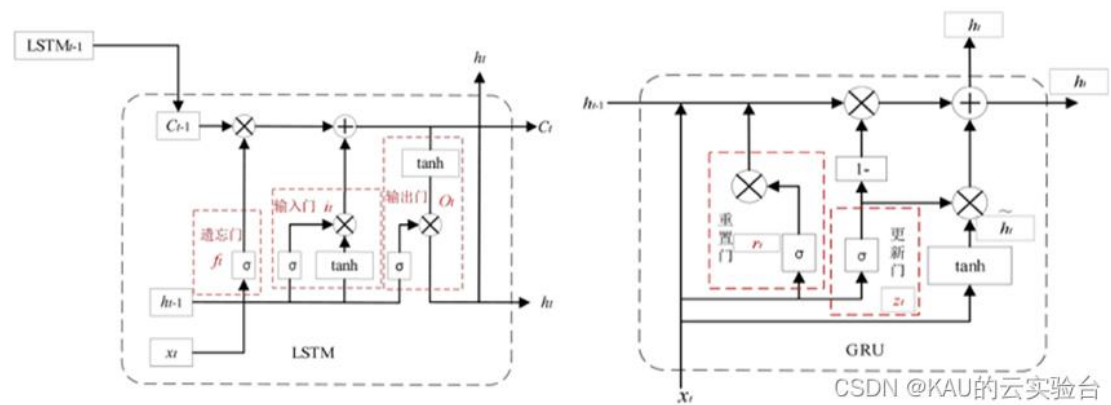
**卷积神经网络** (Convolutional Neural Network, CNN) 是一种专门用于处理具有类似网

格结构数据的神经网络。CNN 通过模拟人脑中视觉皮层的神经元之间的连接方式，实现对图像的识别和处理。

CNN 主要由**输入层、卷积层、池化层和全连接层**组成。在卷积层中，神经元会对输入图像进行**局部感知和权重共享**，以捕捉图像中的局部特征。池化层则对卷积层的输出进行下采样，以减少计算量和过拟合。全连接层则将前面层的输出作为输入，产生最终的输出结果。

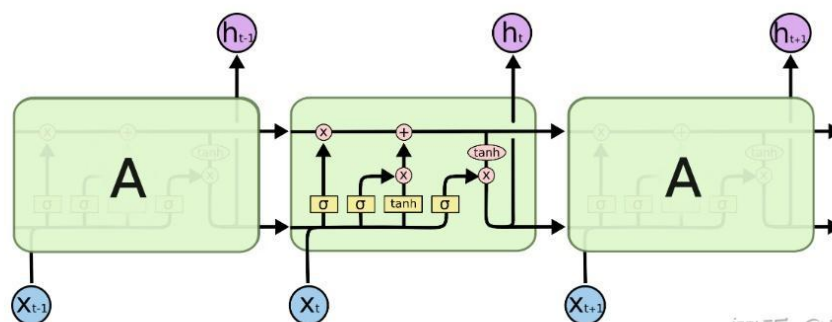
CNN 在计算机视觉领域有着广泛的应用，例如图像分类、目标检测和人脸识别等。此外，CNN 也被应用于自然语言处理领域，例如文本分类和机器翻译等。

**LSTM 和 GRU 解决长程依赖的原理**



深度学习中的长时依赖性问题一直是一个备受关注的话题。在处理序列数据时，**长时依赖性**指的是隔着多个时间步的信息对当前预测结果产生影响。相比之下，短时依赖性只涉及相邻时间步的信息传递。**长时依赖性对于序列数据的处理至关重要，然而梯度消失和梯度爆炸问题却给深度学习模型的训练带来了挑战。**这两个问题主要是由于深层网络中的反向传播过程中梯度不稳定导致的，影响了信息在网络中的传播。因此，为了解决这一问题，研究人员提出了许多改进方法，其中包括门控循环单元（GRU）和长短时记忆网络（LSTM）。

LSTM，全称为 Long Short-Term Memory，即长短时记忆网络，是一种特殊的 RNN 循环神经网络。与原始的 RNN 相比，**LSTM 通过特殊的结构设计解决了 RNN 存在的长期依赖问题，使其能够处理长序列数据，成为当前最流行的 RNN 变体。**



LSTM 中的重复模块包含四个交互层。

知乎 @陈昱天

在解决 RNN 的长期依赖问题之前，我们先来了解一下 RNN 的局限性。RNN 是一种用于处理序列数据的神经网络，通过共享权重的方式进行参数更新。然而，当序列长度增加时，RNN 会面临一个严重的问题：**梯度消失或梯度爆炸**。这意味着在训练过程中，随着时间的推移，前一时刻的信息会逐渐消失或变得不重要，导致模型无法有效地利用历史信息。

为了解决这个问题，LSTM 被设计出来。与标准 RNN 中的重复模块的单层神经网络不同，LSTM 有四层以特殊方式进行交互。LSTM 结构中图的顶部水平线表示细胞状态，类似于传送带，细胞的状态在整个链上传递。细胞状态可以被 LSTM 改变，如删除或添加。这个能力是由 Gate 门结构实现的。LSTM 有三个门，用于保护和控制细胞的状态。

**遗忘门：**用于控制记忆信息的选择和丢弃。遗忘门通过 sigmoid 函数将两个向量拼接起来，得到一个 0~1 之间的数作为控制信号。当值为 0 时，表示丢弃该信息；当值为 1 时，表示保留该信息。

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

**输入门：**用于控制当前时刻输入信息的选择和添加。输入门通过一个 tanh 层和一个 sigmoid 层实现，将当前时刻的输入信息和上一个时刻的输出值拼接  $[h_{t-1}, x_t]$  起来，经过 sigmoid 函数得到控制信号。控制信号用于确定哪些信息被添加到记忆单元中。

$$\begin{aligned} 1. C'_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\ 2. i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \end{aligned}$$

**输出门：**用于控制当前时刻输出信息的选择和生成。输出门通过一个 tanh 层和一个 sigmoid 层实现，将当前时刻的记忆信息和上一个时刻的输出信息拼接起来，经过 sigmoid 函数得到控制信号。控制信号用于确定哪些信息被作为当前时刻的输出。

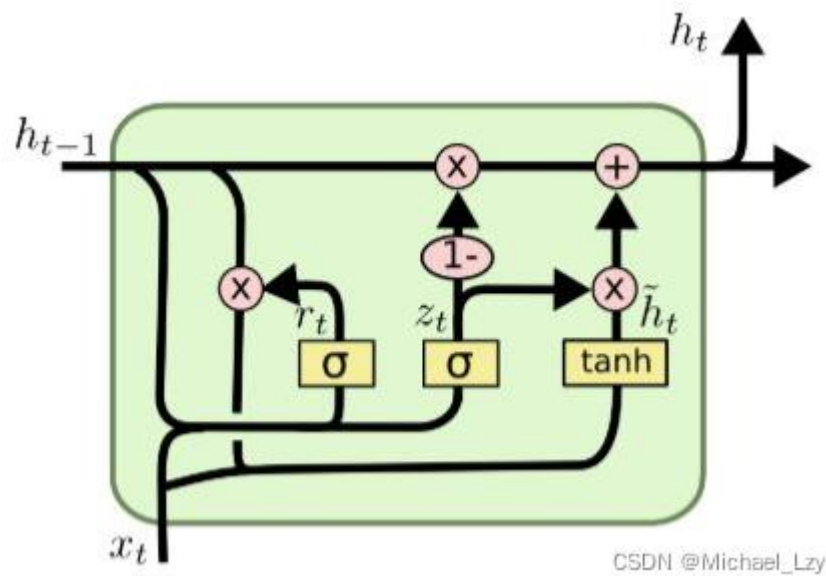
通过上述三个门控机制的协作，LSTM 能够有效地处理长序列数据并解决 RNN 的长期依赖问题。相比普通的神经网络，LSTM 的参数量是它们的 4 倍，因此在实际应用中需要小心地调整超参数和优化模型结构以获得更好的性能。

在实际应用中，LSTM 已经在自然语言处理、语音识别、机器翻译等领域取得了显著成果。例如，在机器翻译中，LSTM 可以捕捉句子中的长期依赖关系，使得翻译更加准确；在语音识别中，LSTM 可以有效地处理语音信号中的时间依赖性，提高识别准确率。

总结起来，LSTM 是一种特殊的 RNN 循环神经网络，通过**特殊的结构和门控机制**解决了 RNN 存在的长期依赖问题。相比普通的神经网络，LSTM 的参数量是它们的 4 倍，因此在实际应用中需要小心地调整超参数和优化模型结构以获得更好的性能。LSTM 在自然语言处理、语音识别、机器翻译等领域已经取得了显著成果。在未来的工作中，我们可以通过进一步研究和改进 LSTM 的结构和算法，提高其在各种任务中的性能表现。

在深度学习和人工智能领域中，循环神经网络（RNN）和 GRU（门控循环单元）是一种重要的算法，它们被广泛应用于处理序列数据，如语音、文本和时间序列等。本文将介绍 GRU 循环神经网络和循环神经网络 RNN 的异同以及重要应用。

GRU 循环神经网络是一种特殊的循环神经网络，它通过**引入门机制来控制信息的流动**，从而有效地解决了长期依赖问题。**GRU 网络由三个门控单元组成：更新门、重置门和输出门**。在每个时刻，**更新门决定是否更新当前状态**，**重置门决定是否保留前一时刻的信息**，**而输出门则控制当前状态是否被输出**。通过这三个门控单元的协同作用，GRU 网络能够捕捉到序列数据中的长距离依赖关系。



循环神经网络 RNN 是一种经典的深度学习模型，它通过将神经网络连接成环状来处理序列数据。RNN 具有记忆能力，可以在处理序列数据时将先前的信息存储在隐藏状态中，以便在后续处理中使用。然而，传统的 RNN 存在长期依赖问题，即难以捕捉到序列数据中的长距离依赖关系。为了解决这一问题，人们提出了许多改进方法，如使用更复杂的结构、增加隐藏层数等。

GRU 循环神经网络在处理序列数据方面具有显著优势。首先，GRU 网络的结构相对简单，相比于 LSTM（长短时记忆网络）等其他 RNN 变体，它不需要额外的记忆单元和复杂的训练过程。其次，GRU 网络的参数数量较少，这有利于减少模型的过拟合风险，同时降低计算成本。此外，GRU 网络的训练过程中，参数更新速度较快，能够更快地收敛到最优解。

循环神经网络 RNN 在语音识别、自然语言处理等领域有着广泛应用。在语音识别领域，RNN 被用于建模声音信号的时间依赖关系，从而提高了识别准确率。在自然语言处理领域，RNN 被广泛应用于文本分类、机器翻译和文本生成等任务。通过对文本中的单词或字符序列进行建模，RNN 能够捕捉到文本中的语义和语法信息。

在应用方面，GRU 循环神经网络也被广泛应用于各个领域。在语音识别领域，GRU 网络能够有效地建模声音信号的时间依赖关系，提高识别准确率。在自然语言处理领域，GRU 网络被用于文本分类、机器翻译和文本生成等任务，表现出了强大的能力。此外，GRU 网络还被广泛应用于图像处理、时间序列预测等任务，取得了良好的效果。

总结 GRU 循环神经网络和循环神经网络 RNN 的异同点：

GRU 网络通过引入门控机制来控制信息的流动，从而解决了长期依赖问题，而 RNN 没

有这种机制。

GRU 网络的结构相对简单，参数数量较少，有利于减少过拟合风险和降低计算成本。

GRU 网络的训练过程中，参数更新速度较快，能够更快地收敛到最优解。

RNN 在语音识别、自然语言处理等领域有着广泛应用，而 GRU 网络在这些领域也具有很强竞争力。

调参中主要超参数及作用。

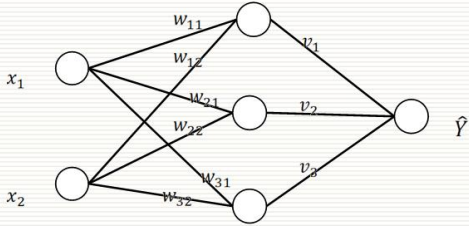
通常可以将超参数分为三类：**网络参数、优化参数、正则化参数**。

**网络参数：**可指网络层与层之间的交互方式（相加、相乘或者串接等）、卷积核数量和卷积核尺寸、网络层数（也称深度）和激活函数等。

**优化参数：**一般指学习率（learning rate）、批样本数量（batch size）、不同优化器的参数以及部分损失函数的可调参数。

**正则化：**权重衰减系数，丢弃法比率（dropout）

BP 算法实现原理推导



神经网络输入:  $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ , 隐含层权重:  $W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \end{bmatrix}$ ,

输出层权重:  $V = \begin{bmatrix} v_1 & v_2 & v_3 \end{bmatrix}$ ,

隐含层线性加和:  $Z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = WX = \begin{bmatrix} w_{11}x_1 + w_{12}x_2 \\ w_{21}x_1 + w_{22}x_2 \\ w_{31}x_1 + w_{32}x_2 \end{bmatrix}$ ,

隐含层激活后输出:  $A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} Relu(z_1) \\ Relu(z_2) \\ Relu(z_3) \end{bmatrix}$ , 其中  $Relu(z) = \begin{cases} z & z > 0 \\ 0 & z \leq 0 \end{cases}$

输出层预测值:  $\hat{Y} = VA = \sum_{i=1}^3 v_i a_i$

损失函数  $E = \frac{1}{2}(Y - \hat{Y})^2$ , 其中,  $Y$  为真实值

$w_{11} \leftarrow w_{11} + \Delta w_{11}$

$$\Delta w_{11} = -\eta \frac{\partial E}{\partial w_{11}} = -\eta \frac{\partial E}{\partial \hat{Y}} \cdot \frac{\partial \hat{Y}}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_{11}}$$
$$= -\eta(Y - \hat{Y}) \cdot v_1 \cdot \frac{\partial a_1}{\partial z_1} \cdot x_1 = \begin{cases} -\eta(Y - \hat{Y}) \cdot v_1 \cdot x_1, & z_1 > 0 \\ 0, & z_1 \leq 0 \end{cases}$$

卷积实现方法及计算方法

卷积是一种数学操作，常用于图像处理和信号处理。在深度学习中，卷积操作可以理解为将一个小的矩阵（**卷积核或滤波器**）与输入数据进行逐元素相乘，并对结果求和，以生成输出数据。

**卷积操作：**把卷积核扣在图像的点阵上，然后对应的两个格子点对点相乘，后将相乘的结果进行一个相加的操作，**实质上是对信号进行滤波**。进行卷积的目的是从输入中提取有用的特征。在图像处理中，可以选择各种各样的 filters。每种类型的 filter 都有助于从输入图像中



提取不同的特征，例如水平/垂直/对角线边缘等特征。在卷积神经网络中，通过使用 filters 提取不同的特征，这些 filters 的权重是在训练期间自动学习的，然后将所有这些提取的特征“组合”以做出决策。

### 数学公式：

假设输入数据  $I$  大小为  $m \times m$ ，卷积核  $K$  大小为  $n \times n$ ，输出数据  $O$  大小为  $p \times p$ ，卷积操作定义如下：

$$O(i, j) = \sum_{r=0}^{n-1} \sum_{c=0}^{n-1} I(i+r, j+c) \cdot K(r, c)$$

其中：

- $I(i+r, j+c)$ ：输入数据上与卷积核重叠的区域值。
- $K(r, c)$ ：卷积核的值。
- $O(i, j)$ ：输出矩阵中第  $i, j$  个位置的值。

### 输出结果尺寸计算公式

卷积操作后，输出矩阵的尺寸为：

$$p = \frac{m - n + 2P}{S} + 1$$

其中：

- $m$ ：输入数据的尺寸（高度或宽度）。
- $n$ ：卷积核的尺寸。
- $P$ ：填充（Padding）的大小。
- $S$ ：步长（Stride）的大小。

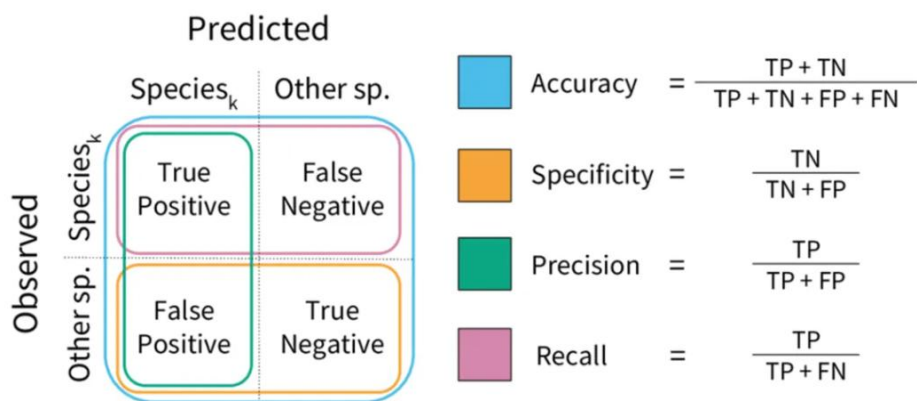
## 3 训练及评估方法

评估模型性能的指标包括哪些？

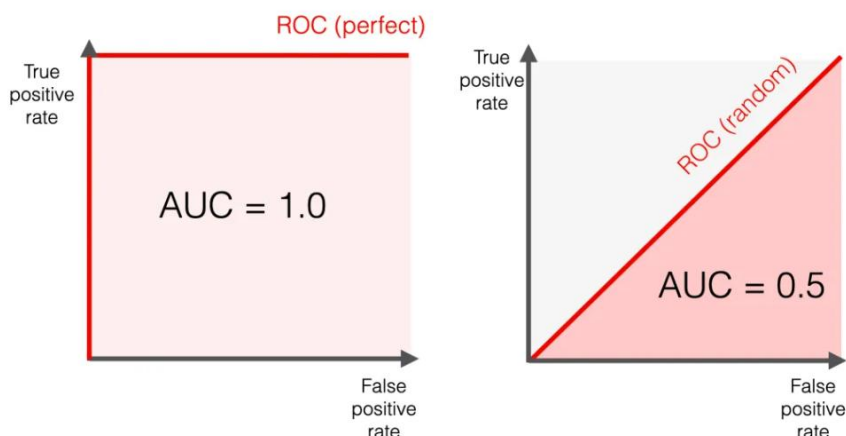
分类任务的评估指标包括**准确率（Accuracy）**、**精确率（Precision）**、**召回率（Recall）**和**F1 分数（F1 Score）**等。

准确率示的是预测为正的样本中有多少是真正的正样本。 召回率是针对我们原来的样本而言的，它表示的是样本中的正例有多少被预测正确了。 精确率和召回率是相互矛盾的一组指标，即精确率提高就会导致召回率降低。





ROC 曲线是展示模型在不同阈值下真正例率与假正例率关系的曲线，越靠近左上角性能越好。AUC 值是 ROC 曲线下方的面积，量化模型性能，取值 0.5 到 1，越接近 1 性能越好。

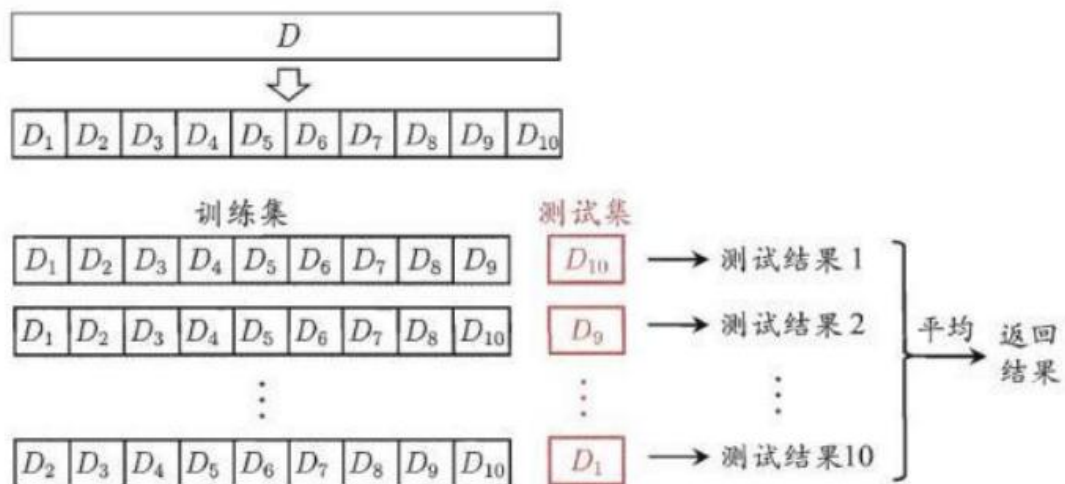


回归问题中评估指标包括均方误差（Mean Squared Error, MSE）和平均绝对误差（Mean Absolute Error, MAE）等。

除了 MSE 和 MAE 之外，还有其他一些回归问题的评估指标，如均方根误差（Root Mean Squared Error, RMSE）、 $R^2$ （决定系数）等。

### k 折交叉验证：意义及优化（K 值选择）

指的是把训练数据  $D$  分为  $K$  份，用其中的  $(K-1)$  份训练模型，把剩余的 1 份数据用于评估模型的质量。将这个过程在  $K$  份数据上依次循环，并对得到的  $K$  个评估结果进行合并，如求平均或投票。如下图所示的 10 折交叉验证，训练数据  $D$  被分为了  $D_1, D_2, D_3, \dots, D_{10}$ ，每次取其中 9 份数据作为训练集，1 份作为测试集，最终将循环后所有的评估结果取平均。



## K 折交叉验证的意义

- 评估模型在未见数据上的表现（泛化能力）。
- 避免模型过拟合或欠拟合。
- 在数据有限的情况下，充分利用数据进行训练和验证。

## K 值的选择

选择合适的 K 值非常重要，通常需要在性能评估与计算代价之间找到平衡。小的 K 值，方差较大，评估结果不够稳定。大的 K 值，计算代价大，因为需要训练 K 个模型。

## K 折交叉验证的优化方法

### 分层 K 折交叉验证（Stratified K-Fold）：

适用于分类问题，确保每个 Fold 中的类分布与原始数据集保持一致。

避免某些类别在某个 Fold 中缺失，提高评估的稳定性。

### 重复 K 折交叉验证（Repeated K-Fold Cross-Validation）：

重复多次 K 折交叉验证，每次重新随机划分数据集，最终取平均性能。

可以进一步减少方差，提供更稳定的评估结果。

### 时间序列交叉验证：

适用于时间序列数据，确保训练数据发生在验证数据之前，避免数据泄漏。

### 嵌套交叉验证（Nested Cross-Validation）：

在模型选择和超参数调优时，使用双重交叉验证。

外层 K 折用于评估模型性能，内层 K 折用于选择最佳参数。

## 拟合和过拟合现象及判断标准。

拟合是指模型对训练数据的学习情况，分为以下三种情况：

正常拟合：模型恰当地学习了数据中的规律，且在训练集和测试集上表现良好。

欠拟合（Underfitting）：模型学习能力不足，未能捕捉数据的真实规律，导致训练集和测试集上的表现都较差。

过拟合（Overfitting）：模型学习能力过强，过于贴合训练数据，甚至将噪声和随机性也当作规律，导致训练集表现良好但测试集表现较差。

## 判断标准

### 训练误差和测试误差对比

正常拟合：训练误差和测试误差较低且接近。

欠拟合：训练误差较高，测试误差与训练误差接近。

过拟合：训练误差很低，而测试误差较高，且两者差异大。

## 交叉验证性能

通过 K 折交叉验证检查模型在不同数据划分下的表现，稳定性差且误差较高的模型可能存在过拟合或欠拟合问题。

## 过拟合的解决方法：

**增加训练数据：**更多的数据有助于模型更好地泛化。

**正则化（Regularization）：**

1. L1 正则化（Lasso）：通过惩罚模型参数，使不重要的特征权重趋于 0。
2. L2 正则化（Ridge）：通过惩罚参数的平方，限制参数值的大小。

**使用 Dropout（神经网络）：**随机丢弃部分神经元，防止模型过度依赖某些特征。

**简化模型：**降低模型复杂度，减少参数数量。

**提前停止（Early Stopping）：**在验证误差不再下降时停止训练。

**数据增强：**增加数据的多样性，例如图像数据可以进行旋转、缩放等操作。

## 残差计算方法：平方误差和 SSE。

均方误差 (Mean Square Error)

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$$

平均绝对误差 (Mean Absolute Error)

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$$

4 AI 海洋学

常用的网络架构及对应海洋任务：举例（遥感任务、要素场分析）

- （1）MLP：海洋参数反演，包括遥感光谱超分、浮游植物色素浓度反演、多源数据的海洋参数反演。
- （2）U-Net：基于 CNN 的语义分割网络，用于海岸线提取、海冰分类、绿潮检测等。
- （3）ConvLSTM：CNN+LSTM，集合了空间信息提取和时序预报的能力，用于海洋环境场预报，包括温度、盐度、波高等。
- （4）YOLO：遥感图像的目标检测，包括船舶检测、涡旋识别等。
- （5）Swin-Transformer：视觉 Transformer 模型，用于多要素、长时序、多海区的海洋要素场分析。

网络架构	对应任务	应用示例
CNN	遥感影像分类/目标检测/分割	船只检测、海冰分割、油污分类
3D-CNN/ConvLSTM	时空数据分析/变化检测	海表温度时空变化、海冰漂移跟踪
RNN/LSTM/GRU	时间序列预测/要素场分析	海洋温度、盐度、洋流预测
GAN	数据增强/超分辨率重建	低分辨率遥感影像重建、风格转换
Transformer	大尺度遥感影像检测/分割	高分辨率SAR影像船只检测、油污分割
CNN+RNN（混合）	时空数据分析与预测	海洋要素时空建模，海浪高度时空预测

海洋数据平台：

Ocean Observing Systems, World Ocean Database, Digital twin ocean, World Ocean Atlas, ocean explorer, Coupled Model Intercomparison Project (CMIP), PANGEA

（1）Ocean Observing Systems：全球海洋观测系统，提供全球的海洋观测数据，包括 Argo 浮标观测计划在内的多个国际项目。

（<https://www.aoml.noaa.gov/global-ocean-observing-system/>）

（2）World Ocean Database：世界海洋数据库，包含自 1772 年库克船长航行时期至 Argo 时期的全球海洋剖面数据。（<https://www.ncei.noaa.gov/products/world-ocean-database>）

（3）Digital twin ocean：数字孪生海洋，结合了实际观测、人工智能和数值模式的交互平台。（<https://digitaltwinoccean.mercator-ocean.eu/>）

（4）World Ocean Atlas：世界海洋地图集，基于 WOD 的海洋剖面数据进行客观分析、

质量控制的温度、盐度、氧、磷酸盐、硅酸盐和硝酸盐的集合。它可用于为各种海洋模型创建边界和/或初始条件，验证海洋的数值模拟，并证实卫星数据。

(<https://www.ncei.noaa.gov/products/world-ocean-atlas>)

(5) **ocean explorer**: 海洋探索者项目，提供海洋勘探中的海洋学和地球物理参数、视频、图像、文档和其他信息。

(<https://oceanexplorer.noaa.gov/data/access/access.html>)

(6) **Coupled Model Intercomparison Project (CMIP)**: 耦合模式比较计划，提供了全球气候模式输出的标准化数据集，用于研究气候变化的多模式比较。

(<https://www.wcrp-climate.org/wgcm-cmip>)

## 海洋大数据的特征？（参考[1]）

信息技术的快速发展，带动海洋数据快速积累，海洋已经进入大数据时代。海洋大数据即是在当前大数据时代背景下，大数据技术在海洋领域的科学实践，具有**大体量 (Volume)**、**多样性 (Variety)**、**快速流转 (Velocity)** 和**高价值 (Value)** 的“4V”特征，是在大数据的理论指导和技术支撑下的价值实现，也是实施海洋强国战略、开发海洋资源、拉动海洋经济、维护国家海洋权益的重要基础；

海洋大数据分为两大类：海洋自然科学类大数据和海洋社会科学类大数据；

海洋自然科学类大数据：海洋实测数据、海洋遥感数据、海洋模式数据、海洋再分析产品数据。

## 数据集选择——如何构建合适的数据集：多样性、均衡性、大规模、低噪声……

在海洋学中，不同类型的任务需要的数据集不同，在构建数据集之前首先需要明确研究的任务类型。常见的任务包括：分类任务、回归任务、目标检测、语义分割和时序预测等。

(1) **多样性**: 数据集的样本应覆盖尽可能多的场景、模式和特征，避免数据集偏向于单一类型或条件。

(2) **均衡性**: 数据集中的样本应在类别、区域或时间上保持均衡，避免数据分布不均引起模型偏差。

(3) **大规模**: 数据集规模足够大，能够涵盖任务所需的多样性和复杂度，避免模型过拟合。

(4) **低噪声**: 数据集应尽量减少错误、异常值或不相关噪声，保证数据的准确性和一致性。

**(5) 代表性：**数据应能反映真实的海洋物理、化学、生物过程，具备科学和实际代表性。

**(6) 时空关联性：**对于时空数据集，样本间需保持合理的时间和空间连续性。

在机器学习和深度学习中，构建一个合适且高质量的数据集是成功的关键。数据集的质量直接影响到模型的训练效果、泛化能力和最终性能。为了确保数据集的有效性，我们需要考虑以下几个关键因素：

## 1. 数据集的多样性

**重要性：**数据集的多样性意味着它包含了问题空间中尽可能多的不同特征和场景。如果训练数据不足以覆盖实际应用中的各种情况，模型可能无法有效处理实际任务中的各种输入，导致欠拟合或泛化能力差。

**如何确保多样性：**

**多来源收集：**数据集应从多个来源收集，以确保能够捕捉到不同的环境或条件。例如，图像分类任务可以从不同的光照、天气、视角等条件下获取图像。

**增加变异性：**可以通过数据增强技术（如旋转、翻转、缩放、平移、裁剪等）来增强训练数据的多样性，尤其在图像和语音任务中非常有效。

**跨领域数据：**如果可能，从不同领域或应用场景中获取数据，帮助模型适应更多的实际情形。

## 2. 数据集的均衡性

**重要性：**均衡性是指不同类别的数据样本在数据集中应尽量保持相对的数量平衡。类别不平衡问题通常会导致模型对某些类别的预测能力差，倾向于偏向样本更多的类别，造成不准确或不公平的结果。

**如何确保均衡性：**

**重采样：**

**过采样：**增加少数类别的样本，例如通过复制少数类样本或使用合成少数类样本（如 SMOTE 算法）。

**欠采样：**减少多数类别的样本数量，确保各类别的样本数量平衡。

**类别加权：**在训练过程中对不同类别给予不同的权重。对于少数类别，可以增加权重，使模型对这些类别更加关注。

**数据增强：**通过合成数据或应用生成对抗网络（GANs）等技术，生成少数类别的更多

样本。

### 3. 数据集的规模

**重要性：**大规模的数据集通常可以帮助模型学到更多的特征和模式，从而提高模型的泛化能力。特别是在深度学习中，训练深度神经网络时，往往需要大量数据才能避免过拟合并确保模型的鲁棒性。

**如何确保大规模：**

**收集更多数据：**通过不同的途径（如爬虫、公开数据集、众包等）收集尽可能多的数据。

**数据增强：**通过数据增强技术有效增加训练数据的数量，特别是在数据量较小的任务中尤为重要。

**合成数据：**使用生成对抗网络（GANs）、数据模拟等技术生成人工数据，尤其在现实世界中收集困难的场景下，可以用合成数据扩充训练集。

### 4. 数据集的低噪声

**重要性：**噪声指的是数据中的不相关或错误的信息，它可能来自数据标注错误、传感器问题、数据处理错误等。高噪声的数据集会严重影响模型的训练，导致过拟合或模型无法有效地学习到数据中的重要模式。

**如何确保低噪声：**

**数据清洗：**在构建数据集时，首先要进行数据清洗，包括处理缺失值、去除重复样本、纠正标注错误等。

**异常值检测：**使用统计方法、可视化工具等检测数据中的异常值，并决定是否将其去除或处理。

**数据标注的准确性：**确保数据标注的质量，特别是在监督学习中，错误的标签可能导致模型学习错误的模式。

**去除冗余特征：**使用特征选择方法（如 L1 正则化、PCA 等）减少数据集中的冗余特征，避免噪声干扰。

### 5. 数据集的代表性和可扩展性

**重要性：**数据集应该能够代表实际应用中的数据分布。一个不具有代表性的数据集可能导致训练出来的模型在实际应用中的表现不佳。此外，随着时间的推移和新数据的到来，数据集应该具备良好的可扩展性，便于随着新的数据补充更新。

**如何确保代表性和可扩展性：**

**跨时间、空间和环境的多样化收集，确保数据能够适应不同的应用场景和变化。**

持续更新：定期收集新数据，更新数据集，尤其对于需要处理动态环境（如金融、气候等）的任务尤为重要。

分层采样：确保数据采样能够涵盖各个重要类别和变化情况，以增加数据集的代表性。

**什么是卷积？简述 CNN 的原理？如何确定 CNN 的卷积核通道数和卷积输出层的通道数？什么是 CNN 的池化层其作用是什么？（本题 10 分）**

卷积操作：把卷积核扣在图像的点阵上，然后对应的两个格子点对点相乘，后将相乘的结果进行一个相加的操作，**实质上是对信号进行滤波**。进行卷积的目的是从输入中提取有用的特征。在图像处理中，可以选择各种各样的 filters。每种类型的 filter 都有助于从输入图像中提取不同的特征，例如水平/垂直/对角线边缘等特征。在卷积神经网络中，通过使用 filters 提取不同的特征，这些 filters 的权重是在训练期间自动学习的，然后将所有这些提取的特征“组合”以做出决策。

CNN 原理：

输入层：与传统神经网络/机器学习一样，模型需要输入的进行预处理操作，常见的 3 中预处理方式有：去均值，归一化和 PCA/SVD 降维

卷积层：卷积层使用“卷积核”进行局部感知，大大减少了模型的计算参数

激励层：所谓激励，实际上是对卷积层的输出结果做一次非线性映射。

池化层：池化（Pooling）：也称为欠采样或下采样。主要用于特征降维，压缩数据和参数的数量，减小过拟合，同时提高模型的容错性。主要有：最大池化 和平均池化

输出层：经过前面若干次卷积+激励+池化后，就来到最后一层输出层。

CNN 的卷积核通道数 = 卷积输入层的通道数

CNN 的卷积输出层通道数(深度)= 卷积核的个数

在卷积层的计算中，假设输入是  $H \times W \times C$ ,  $C$  是输入的深度(即通道数)，那么卷积核(滤波器)的通道数需要和输入的通道数相同，所以也为  $C$ ，假设卷积核的大小为  $K \times K$ ，一个卷积核就为  $K \times K \times C$ ，计算时卷积核的对应通道应用于输入的对应该通道，这样一个卷积核应用于输入就得到输出的一个通道。假设有  $P$  个  $K \times K \times C$  的卷积核，这样每个卷积核应用于输入都会得到一个通道，所以输出有  $P$  个通道

池化层实际上是一种形式的降采样。有多种不同形式的非线性池化函数，而其中“最大池化（Max pooling）”是最为常见的。池化层会不断地减小数据的空间大小，因此参数的数量和计算量也会下降，这在一定程度上也控制了过拟合。通常来说，CNN 的卷积层之间都会周期性地插入池化层。



