



机器学习/数据科学面试 备忘单

安华
版本: 0.1.0.3

本文档包含有关机器学习/数据科学面试期间提出的各种主题的备忘单。本文档会不断更新以包含更多主题。

[单击此处获取更新版本](#)

目录

机器学习主题。	2个
1. 偏差方差权衡	2个
2. 分类中的不平衡数据	3个
3. 主成分分析	4个
4. 贝叶斯定理和分类器	5
5. 回归分析	6个
6. 机器学习中的正则化	7
7. 卷积神经网络	8个
8. 著名的 CNN	9
9. 机器学习中的集成方法	10
10. 自编码器和变分自编码器	11
面试准备。	12
1. 数据结构	12
2. 准备编程面试	13
3. 如何准备行为面试?	14
4. 如何回答行为问题?	16

备忘单——偏差方差权衡

什么是偏见?

- 平均模型预测与地面实况之间的误差
- 估计函数的偏差告诉我们基础模型预测值的能力

$$bias = \mathbb{E}[f'(x)] - f(x)$$

什么是方差?

- 模型预测的平均变异性
- 估计函数的方差告诉您函数可以调整多少以适应数据集的变化

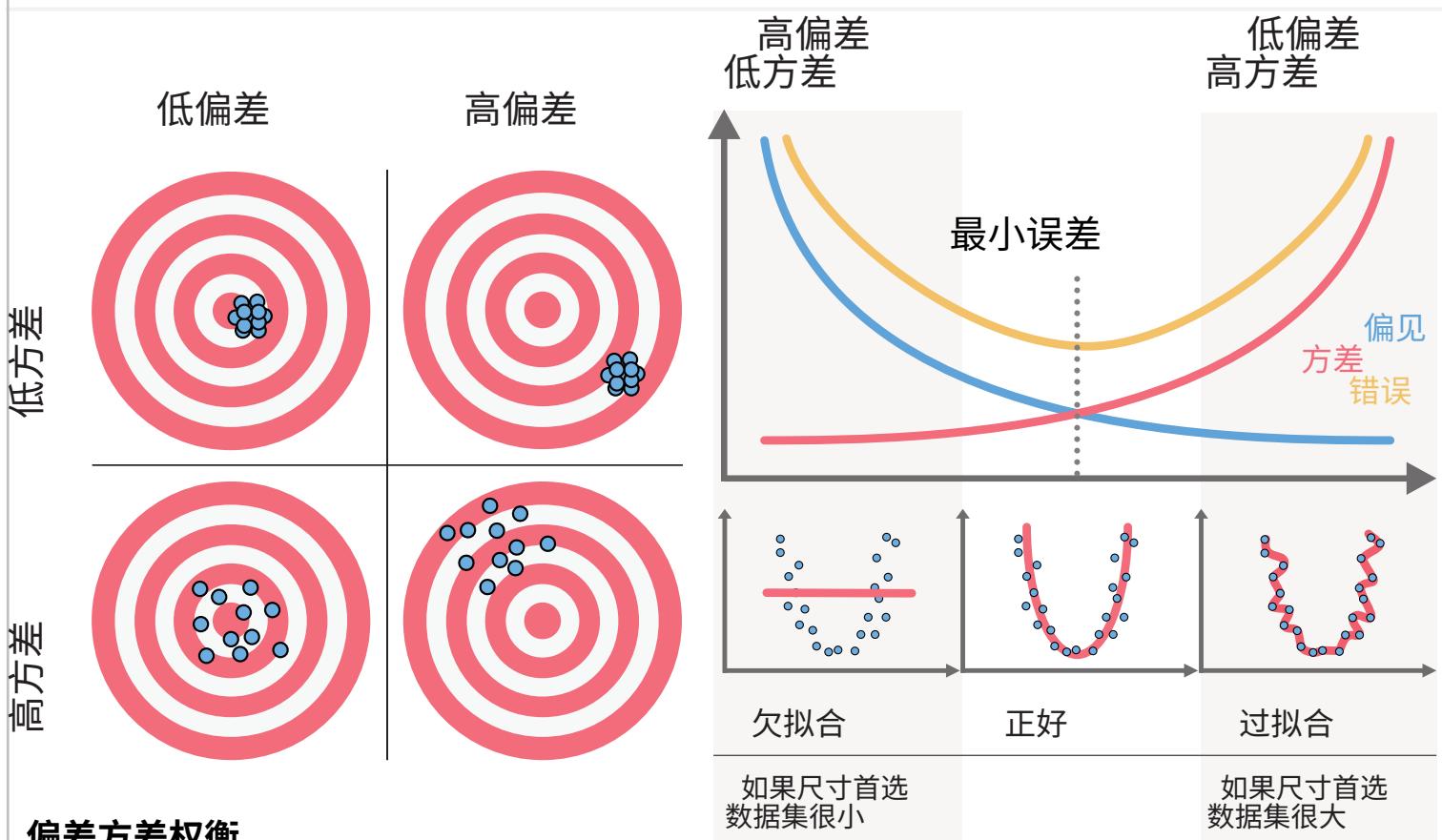
$$variance = \mathbb{E}[(f'(x) - \mathbb{E}[f'(x)])^2]$$

高偏差

- 过度简化的模型欠拟合
- 测试和训练数据的高误差

高方差

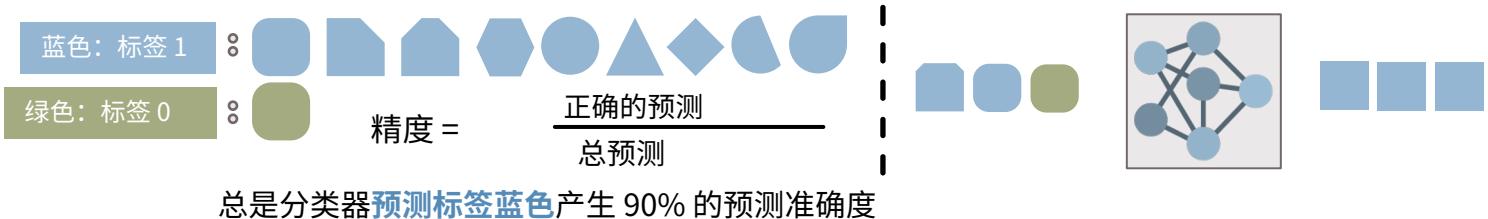
- 过于复杂的模型
- 过拟合
- 训练数据误差低，测试误差高 开始对输入中的噪声建模



偏差方差权衡

- 增加偏差（并不总是）会减少方差，反之亦然
- 误差 = 偏差²+方差+不可约误差
- 最好的模型是错误减少的地方。
- 偏差和方差之间的折衷

备忘单——分类中的不平衡数据



准确性并不总能提供有关训练模型的正确见解

准确性：百分比年龄正确预测
精确：精确模型的

对总预测的正确预测 从检测到的猫中，有多少实际上是猫

整个网络一个值每个类/标签都有一个值

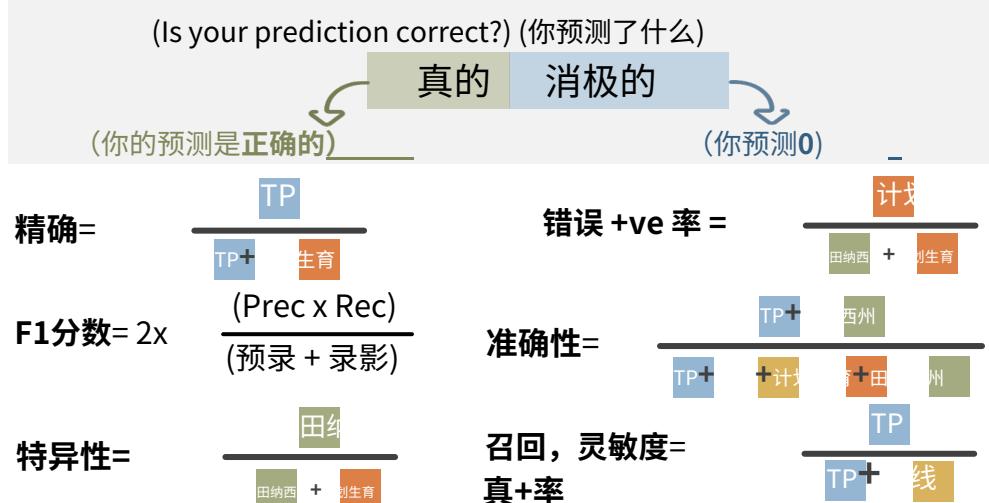
记起：完整性 模型的

正确检测到的猫数超过总猫数精度和召回率的调和平均值

每个类/标签都有一个值 每个类/标签都有一个值

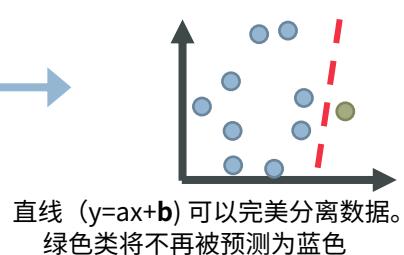
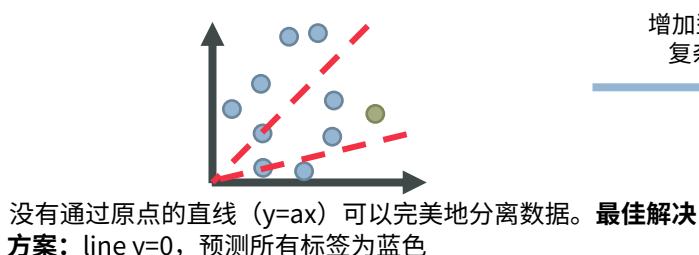
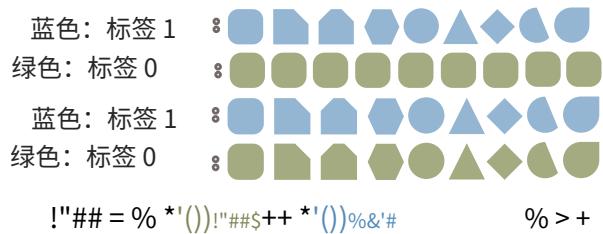
与 1 类相关的性能指标

		实际标签	
		1个	0
预测标签	1	真的积极的	错误的积极的
	0	错误的消极的	真的消极的



可能的解决方案

- 1. 数据复制：**复制可用数据，直到样本数量具有可比性
- 2. 综合数据：**图像：旋转、放大、裁剪、向现有输入图像添加噪声并创建新数据
- 3. 修正损失：**修改损失以反映错误分类较小样本集时的较大错误
- 4. 改变算法：**增加模型/算法的复杂性，使两个类完全可分离（缺点：过度拟合）



备忘单——PCA 降维

什么是主成分分析？

- 基于数据集找到一组新的正交特征向量，使得数据在特征向量（或维度）的方向上传播最大
- 按数据传播（或方差）的降序对特征向量进行评级
- 数据点在第一个特征向量中具有最大方差，在最后一个特征向量中具有最小方差
- 数据点在特征向量方向上的方差可以称为该方向上信息的度量。

脚步

1. 标准化数据点
2. 从给定的数据点中找到协方差矩阵
3. 对协方差矩阵进行特征值分解
4. 对特征值和特征向量进行排序

$$X_{new} = \frac{X - \text{mean}(X)}{\text{std}(X)}$$

$$C[i, j] = \text{cov}(x_i, x_j)$$

$$C = V \Sigma V^{-1}$$

$$\Sigma_{sort} = \text{sort}(\Sigma) \quad V_{sort} = \text{sort}(V, \Sigma_{sort})$$

使用 PCA 降维

- 保留 PCA 评定的 n 个特征向量中的前 m 个。这 m 个向量将是最好的 m 个向量，保留了 m 个可以保留的最大信息
- 给定数据集上的向量

脚步：

1. 执行上面的步骤 1-4
2. 保留排序后的特征向量矩阵中的前 m 个特征向量
3. 为新的基础（特征向量）转换数据
4. 特征向量的重要性与特征值的大小成正比

$$V_{reduced} = V[:, 0 : m]$$

$$X_{reduced} = X_{new} \times V_{reduced}$$

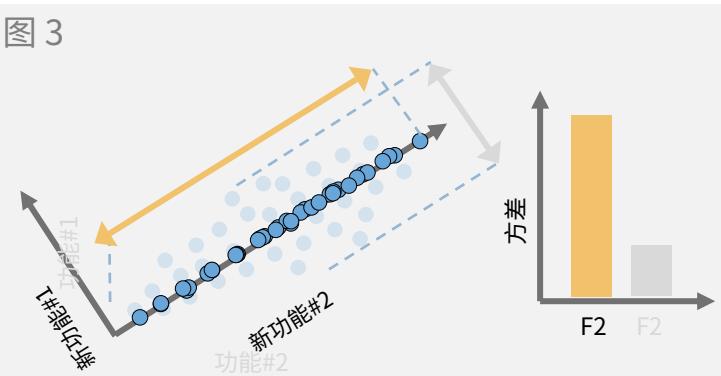
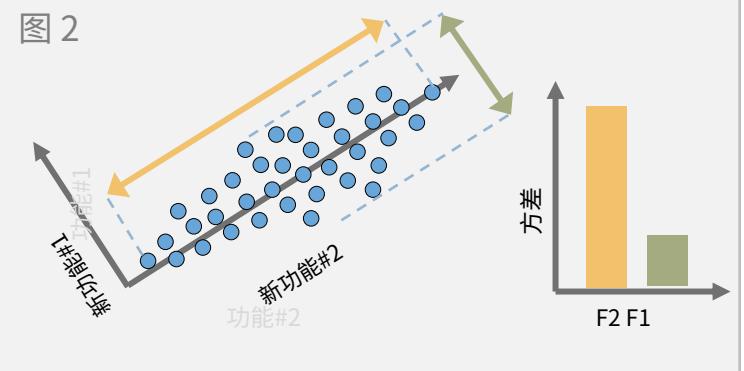
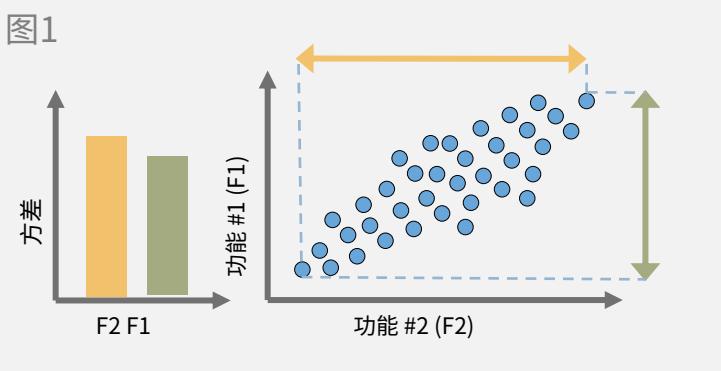


图1：以特征向量作为 x 轴和 y 轴的数据点

图2：旋转笛卡尔坐标系以最大化沿任何一个轴的标准偏差（新功能#2）

图3：删除具有最小数据点标准差的特征向量（新特征#1）并将数据投影到新特征#2

备忘单——贝叶斯定理和分类器

什么是贝叶斯定理？

- 根据可能重新发生的条件的先验知识描述事件发生的概率

$$P(A|B) = \frac{P(B|A)(\text{likelihood}) \times P(A)(\text{prior})}{P(B)(\text{evidence})}$$

- 当我们知道另一个事件时，一个事件的概率如何变化

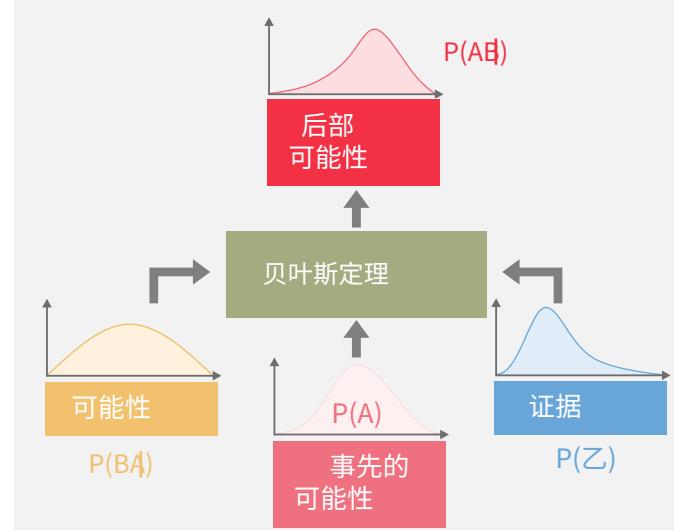
$$P(A) \longrightarrow P(AB)$$

通常，更好的
估计比 $P(A)$

例子

- 火灾概率 $P(F) = 1\%$
- 冒烟概率 $P(S) = 10\%$
- 发生火灾时冒烟的概率 $P(SF) = 90\%$
- 如果我们看到烟雾 $P(FS)$ ，发生火灾的概率是多少？

$$P(F|S) = \frac{P(S|F) \times P(F)}{P(S)} = \frac{0.9 \times 0.01}{0.1} = 9\%$$



最大后验概率 (MAP) 估计

随机变量 y 的 MAP 估计，假设我们已经观察到 iid ($x_1 \uparrow, x_2 \uparrow, x_3 \uparrow, \dots$)，是（谁）给的。我们在估算时尝试适应我们的先验知识。

$$\hat{y}_{\text{地图}} = \operatorname{argmax}_y P(y) \prod_i P(x_i|y)$$

y 最大化先验和似然的乘积

最大似然估计 (MLE)

随机变量 y 的 MAP 估计，假设我们已经观察到 iid ($x_1 \uparrow, x_2 \uparrow, x_3 \uparrow, \dots$)，是（谁）给的。我们假设我们对被估计的数量没有任何先验知识。

$$\hat{y} = \operatorname{argmax}_y \prod_i P(x_i|y)$$

仅最大化可能性的 y

MLE 是 MAP 的特例，其中我们的先验是统一的（所有值的可能性均等）

钠我ve Bayes 分类器（将 MAP 实例化为分类器）

假设我们有两个类， $y=y_1 \uparrow, y=y_2 \uparrow$. 假设我们有不止一个证据/特征 ($x_1 \uparrow, x_2 \uparrow, x_3 \uparrow, \dots$)，使用贝叶斯定理

$$P(y|x_1, x_2, x_3, \dots) = \frac{P(x_1, x_2, x_3, \dots | y) \times P(y)}{P(x_1, x_2, x_3, \dots)}$$

钠我ve 贝叶斯定理假定特征 ($x_1 \uparrow, x_2 \uparrow, \dots$) 是同龄人.IE

$$P(x_1, x_2, x_3, \dots | y) = \prod_i P(x_i|y)$$

$$P(y|x_1, x_2, x_3, \dots) = \prod_i P(x_i|y) \frac{P(y)}{P(x_1, x_2, x_3, \dots)}$$

$$\hat{y} = y_1 \text{ if } \frac{P(y_1|x_1, x_2, x_3, \dots)}{P(y_2|x_1, x_2, x_3, \dots)} > 1 \text{ else } \hat{y} = y_2$$

备忘单——回归分析

什么是回归分析?

将函数 $f(\cdot)$ 拟合到数据点 $y_i = f(x_i)$ 在一些误差函数下。基于估计的函数和误差，我们有以下类型的回归

1. 线性回归:

适合一个线最小化均方误差和对于每个数据点。

$$\min_{\beta} \sum_i \|y_i - f_{\beta}^{\text{linear}}(x_i)\|^2$$

$$f_{\beta}^{\text{linear}}(x_i) = \beta_0 + \beta_1 x_i$$

2. 多项式回归:

适合一个多项式 k 阶 ($k+1$ 个未知数) 最小化均方误差和对于每个数据点。

$$\min_{\beta} \sum_{i=0}^m \|y_i - f_{\beta}^{\text{poly}}(x_i)\|^2$$

$$f_{\beta}^{\text{poly}}(x_i) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \dots + \beta_k x_i^k$$

3. 贝叶斯回归:

对于每个数据点，适合高斯分布经过最小化均方误差。作为麻木的数据点 x 增加，收敛于点

估计即

$$n \rightarrow \infty, \sigma^2 \rightarrow 0$$

$$\mathcal{N}(\mu, \sigma^2) \rightarrow \text{Gaussian with mean } \mu \text{ and variance } \sigma^2$$

$$f_{\beta}(x_i) = f_{\beta}^{\text{poly}}(x_i) \text{ or } f_{\beta}^{\text{linear}}(x_i)$$

4. 岭回归:

可以适合任何一个直线或多项式最小化和均方误差对于每个数据点和加权 L2 范数函数参数 beta 的。

$$\min_{\beta} \sum_{i=0}^m \|y_i - f_{\beta}(x_i)\|^2 + \sum_{j=0}^k \beta_j^2$$

$$f_{\beta}(x_i) = f_{\beta}^{\text{poly}}(x_i) \text{ or } f_{\beta}^{\text{linear}}(x_i)$$

5. 套索回归:

可以适合任何一个直线或多项式最小化均方误差和对于每个数据点和加权 L1 范数函数参数 beta 的。

$$\min_{\beta} \sum_{i=0}^m \|y_i - f_{\beta}(x_i)\|^2 + \sum_{j=0}^k |\beta_j|$$

$$f_{\beta}(x_i) = f_{\beta}^{\text{poly}}(x_i) \text{ or } f_{\beta}^{\text{linear}}(x_i)$$

6. 逻辑回归:

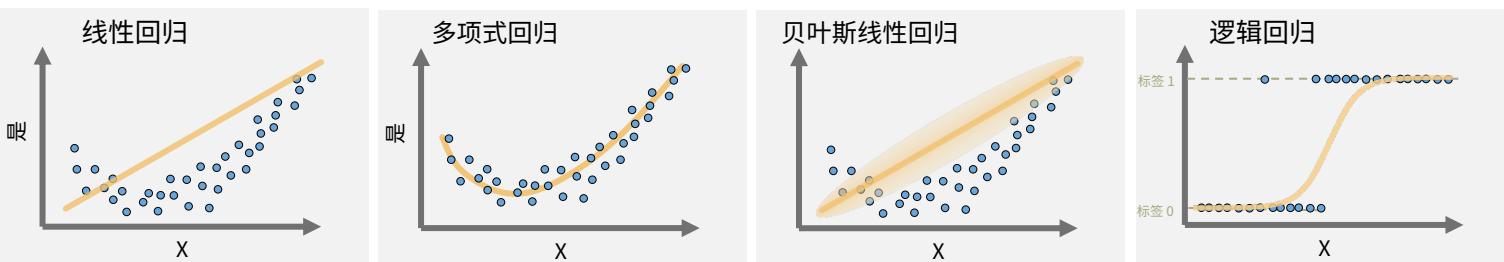
可以适合任何一个线，或具有 sigmoid 激活的多项式最小化二元交叉熵损失为了

每个数据点。标签 y 是二元类标签。视觉表现：

$$\min_{\beta} \sum_i -y_i \log(\sigma(f_{\beta}(x_i))) - (1 - y_i) \log(1 - \sigma(f_{\beta}(x_i)))$$

$$f_{\beta}(x_i) = f_{\beta}^{\text{poly}}(x_i) \text{ or } f_{\beta}^{\text{linear}}(x_i)$$

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$



概括：

	它适合什么？	估计函数	误差函数
线性的	n维的一条线	$f_{\beta}^{\text{linear}}(x_i) = \beta_0 + \beta_1 x_i$	$\sum_{i=0}^m \ y_i - f_{\beta}(x_i)\ ^2$
多项式	k 阶多项式	$f_{\beta}^{\text{poly}}(x_i) = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \dots$	$\sum_{i=0}^m \ y_i - f_{\beta}(x_i)\ ^2$
贝叶斯线性	每个点的高斯分布	$\mathcal{N}(f_{\beta}(x_i), \sigma^2)$	$\sum_i \ y_i - \mathcal{N}(f_{\beta}(x_i), \sigma^2)\ ^2$
岭	线性/多项式	$f_{\beta}^{\text{poly}}(x_i) \text{ or } f_{\beta}^{\text{linear}}(x_i)$	$\sum_{i=0}^m \ y_i - f_{\beta}(x_i)\ ^2 + \sum_{j=0}^n \beta_j^2$
套索	线性/多项式	$f_{\beta}^{\text{poly}}(x_i) \text{ or } f_{\beta}^{\text{linear}}(x_i)$	$\sum_{i=0}^m \ y_i - f_{\beta}(x_i)\ ^2 + \sum_{j=0}^n \beta_j $
物流	具有 sigmoid 的线性/多项式	$\sigma(f_{\beta}(x_i))$	$\min_{\beta} \sum_i -y_i \log(\sigma(f_{\beta}(x_i))) - (1 - y_i) \log(1 - \sigma(f_{\beta}(x_i)))$

备忘单——Regularizat 痞 米大号

ML 中的正则化是什么？

- 正则化是一种解决 ML 中过度拟合的方法。
- 过度拟合的模型无法概括对测试数据的估计
- 当要学习的底层模型是低偏差/高方差时，或者当我们有少量数据时，估计模型容易过拟合。
- 正则化减少了模型的方差

正则化的类型：

1. 修改损失函数：

- **L2 正则化：**防止权重变得太大（由 L2 范数定义）。权重越大，模型越复杂，过度拟合的可能性就越大。

$$\text{loss} = \text{error}(y, \hat{y}) + \lambda \sum_j \beta_j^2 \quad \lambda \geq 0, \lambda \propto \text{model bias}, \lambda \propto \frac{1}{\text{model variance}}$$

- **L1 正则化：**防止权重变得太大（由 L1 范数定义）。权重越大，模型越复杂，过度拟合的可能性就越大。L1 正则化在权重中引入了稀疏性。它迫使更多的权重为零，而不是减少所有权重的平均大小

$$\text{loss} = \text{error}(y, \hat{y}) + \lambda \sum_j |\beta_j| \quad \lambda \geq 0, \lambda \propto \text{model bias}, \lambda \propto \frac{1}{\text{model variance}}$$

- **熵：**用于输出概率的模型。强制概率分布朝向 un

$$\text{loss} = \text{error}(p, \hat{p}) - \lambda \sum_i \hat{p}_i \log(\hat{p}_i) \quad \lambda \geq 0, \lambda \propto \text{model bias}, \lambda \propto \frac{1}{\text{model variance}}$$

2. 修改数据采样：

- **数据扩充：**通过随机裁剪、扩大、旋转、添加少量噪声等，从可用数据中创建更多数据。

- **K 折交叉验证：**将数据分成 k 组。在 (k-1) 组上训练并在 1 组上测试。尝试所有 k 种可能的组合。

3. 改变训练方式：

- **注入噪声：**在学习权重时将随机噪声添加到权重中。它推动模型对权重的微小变化相对不敏感，因此正则化
- **辍学：**一般用于神经网络。连续层之间的连接根据丢弃率随机丢弃，其余网络在当前迭代中进行训练。在下一次迭代中，另一组随机连接被丢弃。

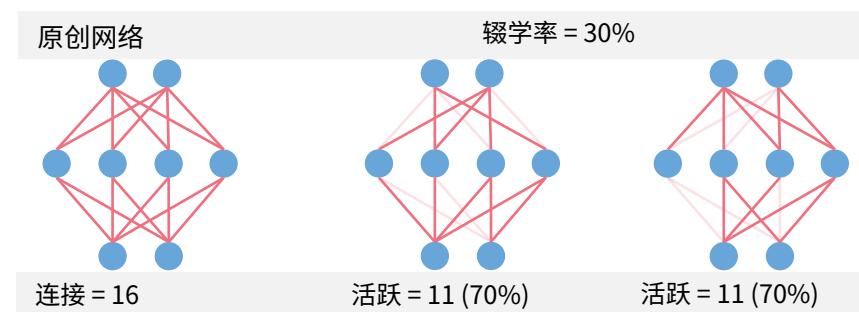


图 2. K 折 CV

来源：<https://www.cheatsheets.aqeel-anwar.com> 教程：[点击这里](#)

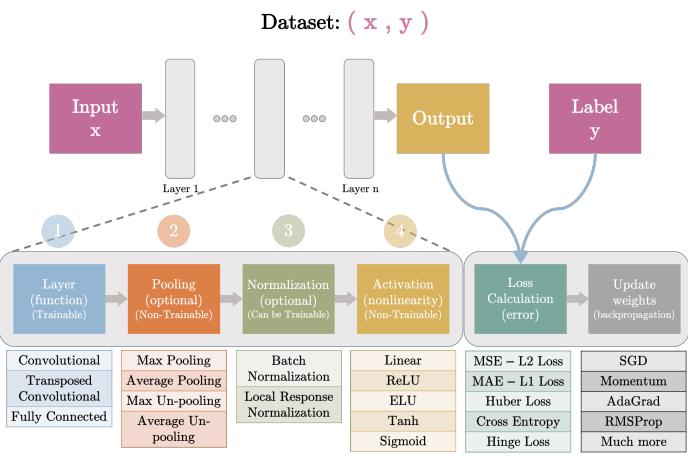
图 3. 辍学



备忘单——卷积神经网络

卷积神经网络：

数据通过输入层进入CNN，在到达输出层之前经过各个隐藏层。网络的输出在损失或错误方面与实际标签进行比较。计算此损失对可训练权重的偏导数，并通过使用反向传播的各种方法之一更新权重。



美国有线电视新闻网模板：

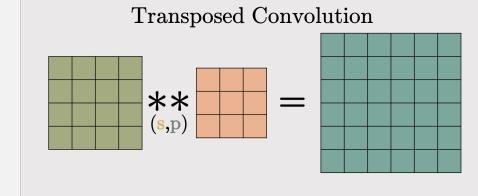
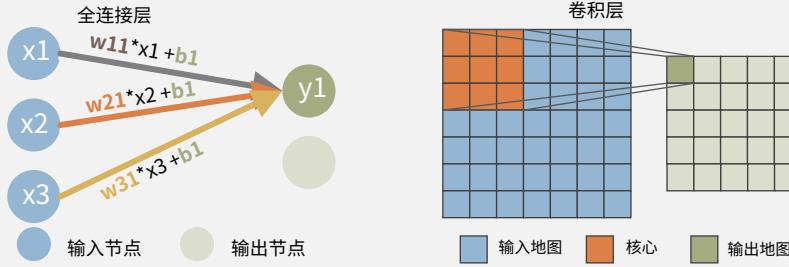
大多数常用的隐藏层（不是全部）都遵循一种模式

1.图层功能：基本的转换函数，例如卷积层或全连接层。

A. 完全连接：输入和之间的线性函数

B. 虚拟层：这些层应用于2D(3D)输入特征图。可训练权重是一个2D(3D)内核/过滤器，它在输入特征图上移动，生成与输入特征图重叠区域的点积。

b. 转置卷积 (DeConvolutional) 层：通常用于增加输出特征图的大小（上采样）。转置卷积层背后的想法是撤消（不完全是）卷积层



2. 池化：非可训练层改变特征图的大小

A. 最大/平均池化：基于选择内核定义的感受野中的最大值/平均值来减小输入层的空间大小

b. 取消池化：一个不可训练层，用于根据将输入像素放置在内核定义的输出的感受野中的某个索引处来增加输入层的空间大小。

3. 归一化：通常在激活函数之前使用，以限制无界激活将输出层值增加得太高

A. 局部响应归一化 LRN：一个不可训练层对局部邻域内特征图中的像素值进行平方归一化。

b. 批量归一化：一种可训练的方法，通过在训练期间学习尺度和移位变量来规范化数据。

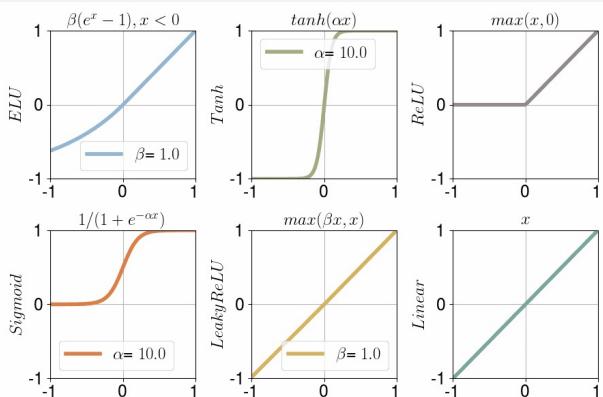
3. 激活：引入非线性，因此CNN可以

5. 损失函数：量化CNN预测有效映射非线性复杂映射的距离。
来自实际标签。

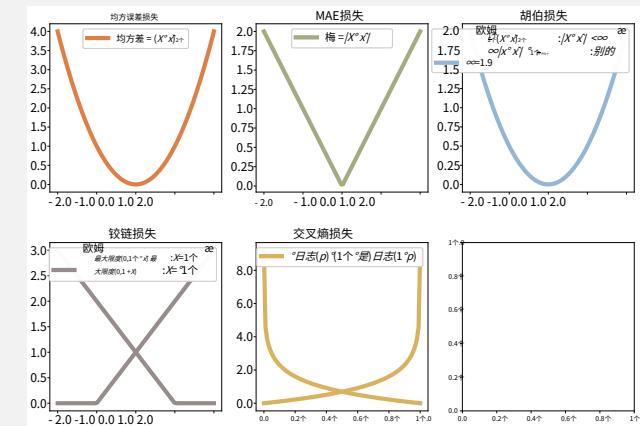
A. 非参数/静态函数：线性，ReLU

b. 参数函数：ELU、tanh、sigmoid、Leaky ReLU

C. 有界函数：tanh, 乙状结肠



Type: max'pool - Stride: 1 Padding: 1						
0	0	0	0	0	0	0
0	4.3	5	12	3.7	11	0
0	12	12	6	11	13	0
0	8.5	8.4	7.6	6	10	0
0	3.9	11	5.7	3.6	11	0
0	8.3	5.8	9.7	13	7.1	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
Input						
12	12	12	13	13		
0	0	0	0	0		
0	0	0	0	0		
0	0	0	0	0		
Output						



备忘单——机器学习中的集成学习

什么是集成学习？群众的智慧

将多个弱模型/学习器组合成一个预测模型，以减少偏差、方差和/或提高准确性。

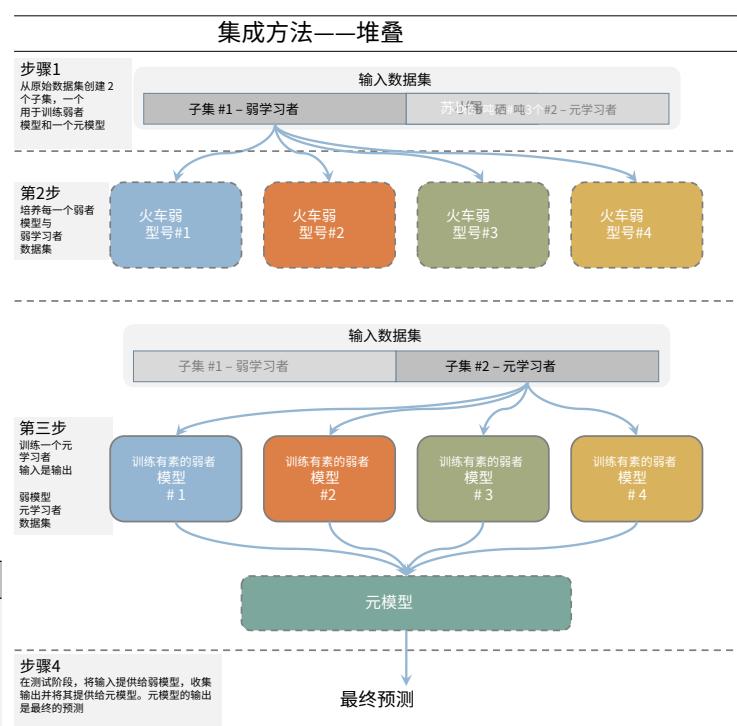
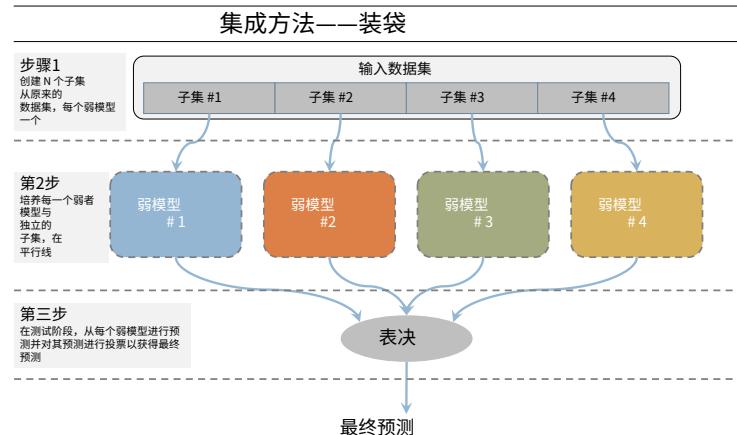
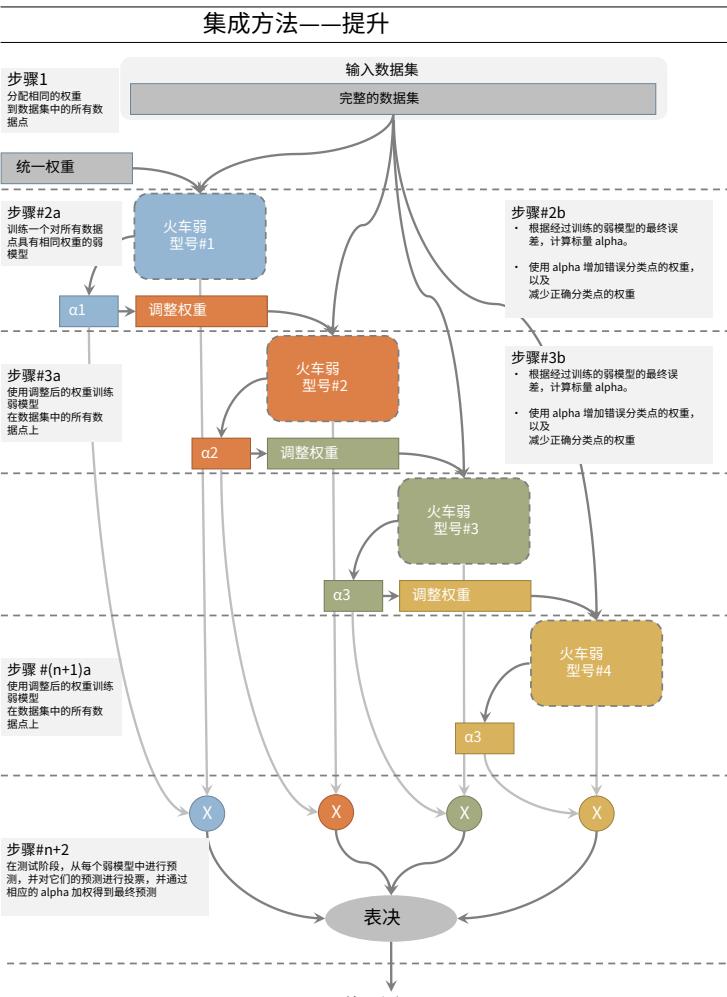
集成学习的类型：N个弱学习器

1.装袋：使用输入数据集的N个非重叠子集并行训练N个不同的弱模型（通常是相同类型 - 同质的）。在测试阶段，对每个模型进行评估。选择预测数量最多的标签作为预测。装袋方法减少了预测的方差

2.升压：按顺序使用完整数据集训练N个不同的弱模型（通常是相同类型 - 同质的）。为之前的弱模型错误分类的数据点提供了更多的权重，以便它们可以被下一个弱学习者正确分类。在测试阶段，对每个模型进行评估，并根据每个弱模型的测试误差，对预测进行加权投票。提升方法减少了预测的偏差。

3.堆叠：使用数据集的两个子集之一并行训练N个不同的弱模型（通常是不同类型的 - 异构的）。一旦训练了弱学习器，他们就会被用来训练一个元学习器来结合他们的预测并使用另一个子集进行最终预测。在测试阶段，每个模型预测其标签，这些标签集被提供给生成最终预测的元学习器。

这三种方法的框图和比较表如下所示。



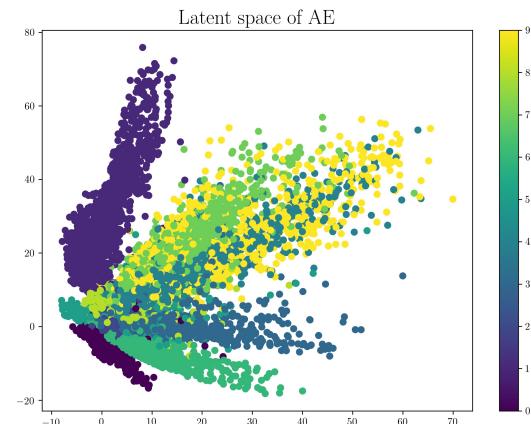
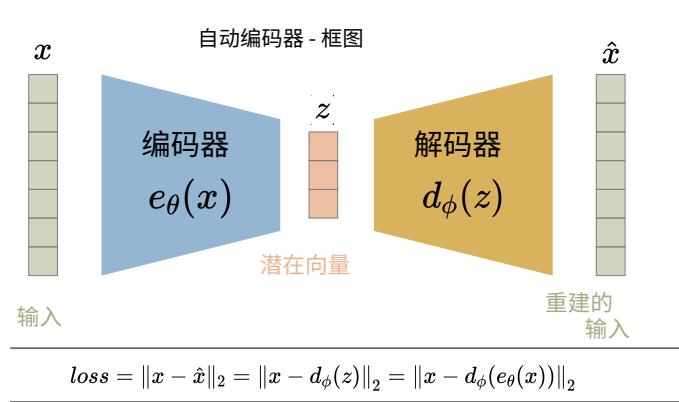
备忘单——自动编码器和变分自动编码器

上下文——数据压缩

- 数据压缩是训练网络的重要阶段。这个想法是压缩数据，这样相同数量的信息可以用更少的比特来表示。

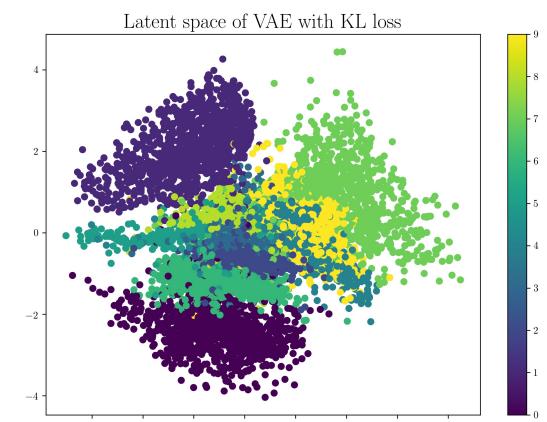
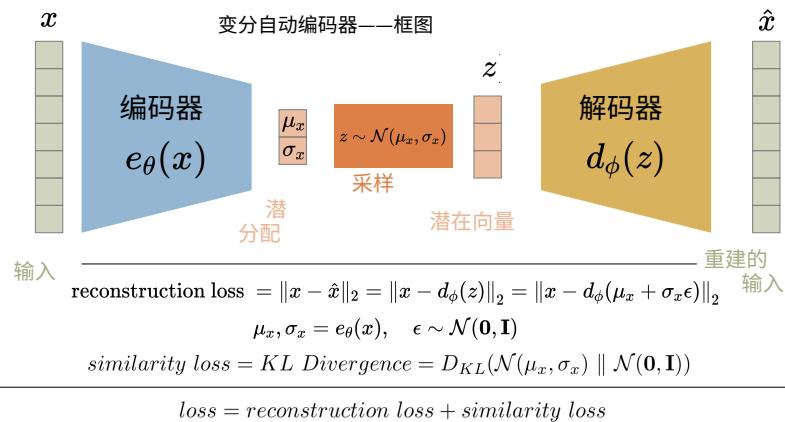
自动编码器 (AE)

- 自动编码器用于为给定的网络配置学习未标记数据的有效嵌入。它由两部分组成，一个**编码器**，和一个**解码器**。
- 编码器将数据从高维空间压缩到低维空间（也称为潜在空间），而解码器将潜在空间转换回高维空间。
- 整个编码器-解码器架构都在损失函数上进行了集体训练，这鼓励输入在输出端被重构。因此，损失函数是编码器输入和解码器输出之间的均方误差。
- 潜在变量未正则化。选择一个随机的潜在变量将产生垃圾输出。
- 潜在变量是确定性值，空间缺乏生成能力



变分自动编码器 (VAE)

- 变分自动编码器解决了自动编码器中非正则化潜在空间的问题，并为整个空间提供了生成能力。
- VAE 的编码器不是输出潜在空间中的向量，而是输出 μ 和 σ 的参数，每个输入在潜在空间中的预定义分布。
- 然后，VAE 对该潜在分布施加约束，迫使其成为正态分布。
- 压缩形式的潜在变量是均值和方差
- VAE 的训练损失定义为重建损失和相似性损失（单元高斯和解码器输出分布之间的 KL 散度）之和。
- 潜在变量是平滑连续的，即潜在变量的随机值在解码器处产生有意义的输出，因此潜在空间具有生成能力。
- 解码器的输入是从具有编码器输出的均值/方差的高斯样本中采样的。



备忘单——数据结构

1. 清单

- 已订购元素的集合
- 每个元素的位置由 **指数** 元素可以在 **任何订单**
-



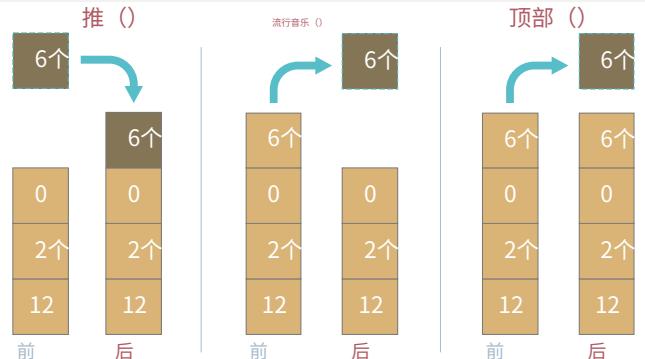
- 每个链表元素包含**值**和**地址（指针）**到下一个链表元素。
- 因此链表只能依次遍历**一次每个元素**

2. 链表

- 链表没有由它们定义的顺序 **内存中的物理位置**。
- 链表的连续元素在内存中并不相邻。

3. 堆栈

- 堆栈是一个**顺序数据结构**它保持元素插入时的顺序。
- 后进先出（后进先出法）**顺序，这意味着元素只能在**相反的顺序**当它们被插入堆栈时。
- 要成为的元素**最后插入**，将**第一个被删除**从堆栈。
- 推（）**在头堆栈的，而**流行音乐（）**从中删除一个元素头堆栈的
- 堆栈的一个真实示例是**一堆厨房盘子**

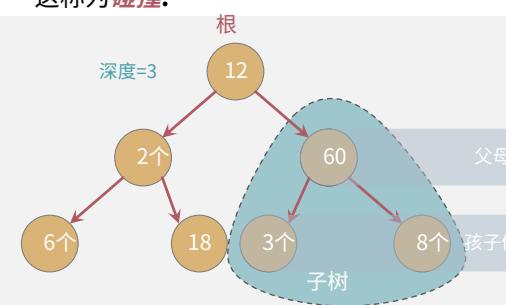


5. 哈希表

- 创造**配对作业**（键映射到值）所以**可以在恒定时间内访问这些对**
- 对于每个（核心价值）对，密钥通过哈希函数传递创建一个**唯一的物理地址**用于存储在内存中的值。
- 哈希函数最终可能会为不同的密钥生成相同的物理地址。这称为**碰撞**。

队列是一个**顺序数据结构**保持元素插入时的顺序

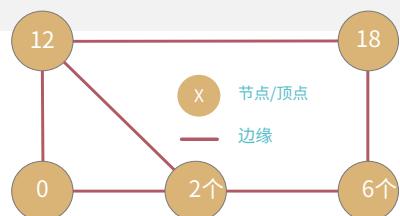
- 先进先出（先进先出）**, 元素是**先插入**, 将**第一个被删除**从队列中
- 每当添加一个元素（Enqueue()）时，它就被添加到**结尾**的队列。另一方面，元素移除（Dequeue()）是从**正面**的队列。
- 一个现实生活中的例子是**杂货店结账排队**



- 维护其元素之间的层次关系。
- 根节点**—树顶部的节点
- 父节点**—任何至少有一个孩子的节点
- 子节点**—父节点的后继者称为子节点。一个节点既可以是父节点也可以是子节点。根永远不是子节点。**叶节点**—没有任何子节点的节点。
- 遍历**—按一定顺序遍历节点，例如BFS、DFS

4. 图表

- 一个图是一个**一对套（五，五）**，在哪里 **V** 是所有顶点的集合，**E** 是所有边的集合。
- 一个节点的邻居是**通过边与该节点相连的所有顶点的集合**。
- 与树木相反，**图可以是循环的**，这意味着从一个节点开始，沿着边缘，你可以在同一个节点上结束



备忘单——为编程面试做准备

第 1 部分 – 如何准备编程面试? *

- **时间线:**

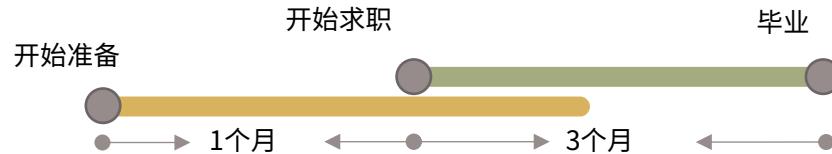


图 1 – 编码面试的准备时间表

- **审查数据结构和复杂性:**

以下7种数据结构是面试必备的，以及它们的时间/空间复杂度

- 列表/数组、链表、哈希表/字典、树、图、堆、队列
- 点击[这里](#)教程。

- **练习编码问题:**

- 多种在线资源，例如[LeetCode.com](#), [InterviewBit.com](#), [HackerRank.com](#)等。
- 选择一个在线资源并针对简单和中等编码问题（大约 100-150）。
- 初学者在面试前2-3个月开始准备，中级1个月左右。

- **笔记:**

- 根据我个人的经验，LeetCode.com 的付费订阅是值得的。
- LeetCode 的 Facebook、Uber、Google 和 Microsoft 标记问题几乎涵盖了所问问题的 90%

第 2 部分 – 如何回答编码问题? *

- **听问题**

面试官会举例说明问题。记下要点。

- **谈谈你对问题的理解**

重复问题并确认你的理解。提出澄清性问题，例如 1. 输入/输出数据类型限制

2. 输入大小/长度限制
3. 特殊/角落案例

- **讨论你的方法**

演练一下你将如何处理这个问题，并询问面试官他是否同意。谈谈你喜欢的数据结构和原因。从更大的角度讨论解决方案。

- **开始编码**

询问面试官您是否可以开始编码。定义有用的函数，边写边解释。大声思考，以便面试官评估您的思维过程。

- **讨论时间和空间复杂度**

根据大 O 为您的编码方法讨论时间和空间复杂性。

- **优化方法**

如果你的方法不是最优化的，面试官会提示你一些改进。注意提示并尝试优化您的代码。



图 2 – 如何回答编码问题?

*免责声明：这些建议是基于作者的个人经验。所提到的方法和资源可能对某些人非常有用，但对其他人则不是那么有用。



关键字

列出将填充您的个人故事的重要关键字。下表给出了最常见的关键字

冲突解决	谈判	妥协于达成目标	创造力	灵活性	令人信服
搬运危机	具有挑战性的情况	与困难的人	另一个团队优先事项不对齐	调整为同事风格	站出来
处理-ve反馈	同事对你的看法	与一个工作最后期限	你的实力	你的弱点	影响其他的
搬运失败	搬运意外情况	转换挑战机会	决定没有足够的数据	冲突解决	指导/领导

故事

- 列出你参与过的所有组织。例如
 - 学术界：学士、硕士、博士
 - 行业：工作，实习
 - 社团：文化、技术、体育
- 想一想第1步中可以归入其中一个关键字类别的故事。这故事越多越好。您应该至少有10-15个故事。
- 通过为每个故事分配多个关键字来创建汇总表。这将有助于你在面试中问问题时过滤掉故事。下面是一个例子

故事 1:	[说服] [采取立场] [影响他人] [指导] [领导]
故事 2:	
故事 3:	[冲突解决] [谈判] [没有足够数据的决定]
故事 4:	

星格式

按照本备忘单2/4部分中的说明，以STAR格式写下故事。这将帮助您以有意义的方式练习故事的组织。

2/4

如何准备行为面试?

直接的*, 有意义*, 个性化*, 逻辑*

* (示例中使用了各自的颜色来标识这些特征)



例子：“告诉我们你必须说服高级管理人员的时间”

情况

解释情况并提供必要的背景

你的故事。

况

“2019年暑假我在XYZ公司实习，提供给我的project details很详细。经过一些初步的头脑风暴和研究后，我意识到可以修改项目方法，使其在基本KPI方面更有效率。我决定和我的经理谈谈这件事。”

任务

解释任务和你的
责任在
情况

任

“我与我的经理通了一个小时的电话，向他详细解释了提议的方法以及它如何改进KPI。我能够说服他。他问我是否能够在高层管理人员面前展示我提议的方法以供批准。我同意了。我在ABC（城市）办公室工作，高管们需要从XYZ（城市）办公室飞过来。”

行动

浏览您为解决问题而采取的步
骤和行动

问题

行

“我对高管进行了快速背景调查，以便更好地了解他们的专业领域，以便我可以相应地说服他们。我精心准备了15张幻灯片演示从解释他们的方法开始，转向我提出的方法，最后比较他们的初步结果。”

结果

陈述结果
你行动的结果

结

“经过一些积极的讨论，我们能够确定提议的方法比最初的方法更好。高管们对我的方法提出了一些小改动，非常感谢我的立场。实习结束时，我从68名实习生中脱颖而出，与公司高级副总裁共进午餐。”



如何回答一个 行为问题？

理解、提取、映射、选择和应用



例子：“告诉我们你必须说服高级管理人员的时间”

理解

理解问题

例子：一个我能够说服我的前辈的故事。也许他们心里有数，
我有一个更好的方法并尝试
说服他们

提炼

提取关键字和标签

提取有用的关键字，封装
问题的要点

例子：
[说服力]、[创意]、[领导力]

地图

将关键字映射到您的故事

入围所有属于该类别的故事
从上一步中提取的关键字 **例子：**

故事 1、故事 2、故事 3、故事 4、……，故事 N

选择

选择最佳故事

从入围的故事中，选择最能描述问题且未被使用
的故事

到目前为止在采访中

例子：故事3

申请

应用 STAR 方法

将 STAR 方法应用于所选故事
回答问题

例子：有关详细信息，请参阅备忘单 2/3



4/4

行为面试 备忘单

总结行为面试



如何准备为了面试

1个

收集重要主题作为关键字

了解并收集所有重要主题
面试中经常被问到

2个

收集你的故事

根据您曾参与的所有组织，想一想属于上述关键字的所有故事

3个

以 STAR 格式练习故事

使用 STAR 格式练习每个故事。你将会拥有
按照这种格式回答问题。

4个

为故事分配关键字

为您的每个故事分配一个或多个关键字。这会
帮助您快速回忆起它们

5个

创建汇总表

创建一个将故事映射到其相关关键字的摘要表。这将在行为问题中使用

ü

理解问题

理解问题并澄清任何混淆
你有

乙

提取关键词

尝试从中提取一个或多个关键字
问题

米

将关键字映射到故事

根据提取的关键字，使用
准备期间创建的汇总表（步骤 4）

小是

选择一个故事

由于每个关键字可能分配给多个故事，因此请选择最相关且
尚未使用的一个。

A

应用 START 格式

故事入围后，应用 STAR 格式
回答问题的故事。

关注作者：

关注作者以获取更多机器学习/数据科学内容：

- ◆ 中等的：<https://aqeel-anwar.medium.com>
- ◆ 领英：<https://www.linkedin.com/in/aqeelanwarmalik/>

反馈：

如果您在备忘单中发现任何错误，请提供您的反馈[这里](#)

版本历史

- 版本 0.1.0.3 - 2021 年 12 月 25 日
 - 添加备忘单：[自编码器和变分自编码器](#)
- 版本 0.1.0.2 - 2021 年 5 月 19 日
 - 添加备忘单：[数据结构和准备编程面试](#)
 - 在每个备忘单的末尾添加了教程链接
- 版本 0.1.0.1 - 2021 年 4 月 5 日
 - 修复了贝叶斯定理、回归分析和分类器以及 PCA 降维备忘单中的小拼写错误。
- 版本 0.1.0.0 - 2021 年 3 月 30 日
 - 包含九个 ML 基础知识和两个行为面试备忘单的初稿。