

# 1

单模态的表示学习负责将信息表示为计算机可以处理的数值向量或者进一步抽象为更高层的特征向量。

# 2

数据集一般由标签/目标值 (label) 和特征 (feature) 构成。利用特征值预测, 利用目标值确定预测正确程度。

# 3

简单分类器: 最近邻分类器

训练集包含  $N$  个样本  $\{x_1, x_2, \dots, x_N\}$ , 分别属于  $c$  个类别

定义距离为  $D(x, x_i) = \|x - x_i\|$  (也可以选择其他距离函数)

若有  $D(x, x_k) = \min_{i=1,2,\dots,N} \{D(x, x_i)\}$

且  $x_k \in \omega_j$ , 则  $x \in \omega_j$

# 4

对于模型性能的评估, 我们通常分为三步:

- 1.对数据集进行划分, 分为训练集和测试集两部分;
  - 2.对模型在测试集上面的泛化性能进行度量;
  - 3.基于测试集上面的泛化性能, 依据假设检验来推广到全部数据集上面的泛化性能
- 一般有

## 1. 留出法:

留出法是直接将数据集 $D$ 划分为两个互斥的集合, 其中一个集合作为训练集 $S$ , 另一个作为测试集 $T$ , 我们需要注意的是在划分的时候要尽可能保证数据分布的一致性, 即避免因数据划分过程引入额外的偏差而对最终结果产生影响。

## 2. 交叉验证法:

$k$ 折交叉验证通常把数据集 $D$ 分为 $k$ 份, 其中的 $k-1$ 份作为训练集, 剩余的那一份作为测试集, 这样就可以获得 $k$ 组训练/测试集, 可以进行 $k$ 次训练与测试, 最终返回的是 $k$ 个测试结果的均值。

## 3. 自助法

自助法使用有放回重复采样的方式进行数据采样, 即我们每次从数据集 $D$ 中取一个样本作为训练集中的元素, 然后把该样本放回, 重复该行为 $m$ 次, 这样我们就可以得到大小为 $m$ 的训练集, 在这里面有的样本重复出现, 有的样本则没有出现过, 我们把那些没有出现过的样本作为测试集。

# 5

评分函数将图像像素值集合最终映射为各个分类类别的得分，得分高低代表图像属于该类别的可能性高低。例如，有训练集 $x_i \in R^D$ ，对应的标签为 $y_i$ ，这里 $i = 1, 2, \dots, N$ ， $y_i \in 1, 2, \dots, k$ 。也就是说，我们有 $N$ 个图像样本，每个样本的维度是 $D$ ，共有 $K$ 个不同的类别。

## 6

线性分类器使用线性的函数表达式对样本进行分类，即划分边界为一个超平面。

$$y = f(\vec{w} \cdot \vec{x}) = f\left(\sum_j w_j x_j\right)$$

## 7

损失函数 (loss function) 就是用来度量模型的预测值 $f(x)$ 与真实值 $Y$ 的差异程度的运算函数，它是一个非负实值函数，通常使用 $L(Y, f(x))$ 来表示，损失函数越小，模型的鲁棒性就越好。损失函数：

### 1. 均方误差损失函数 (MSE)

$$L(Y | f(x)) = \frac{1}{n} \sum_{i=1}^N (Y_i - f(x_i))^2$$

### 2. L1损失函数

$$L(Y | f(x)) = \sum_{i=1}^N |Y_i - f(x_i)|$$

### 3. L2损失函数

$$L(Y | f(x)) = \sqrt{\frac{1}{n} \sum_{i=1}^N (Y_i - f(x_i))^2}$$

### 4. Smooth L1损失函数

$$L(Y | f(x)) = \begin{cases} \frac{1}{2}(Y - f(x))^2 & |Y - f(x)| < 1 \\ |Y - f(x)| - \frac{1}{2} & |Y - f(x)| \geq 1 \end{cases}$$

### 5. huber损失函数

$$L(Y | f(x)) = \begin{cases} \frac{1}{2}(Y - f(x))^2 & |Y - f(x)| \leq \delta \\ \delta|Y - f(x)| - \frac{1}{2}\delta^2 & |Y - f(x)| > \delta \end{cases}$$

### 6. KL三度损失函数

$$L(Y | f(x)) = \sum_{i=1}^n Y_i \times \log\left(\frac{Y_i}{f(x_i)}\right)$$

### 7. 交叉熵损失函数

$$L(Y | f(x)) = - \sum_{i=1}^N Y_i \log f(x_i)$$

## 8. softmax损失函数

$$L(Y | f(x)) = - \frac{1}{n} \sum_{i=1}^n \log \frac{e^{f_{Y_i}}}{\sum_{j=1}^c e^{f_j}}$$

## 9. Focal loss

$$FE = \begin{cases} -\alpha(1-p)^\gamma \log(p) & y = 1 \\ -(1-\alpha)p^\gamma \log(1-p) & y = 0 \end{cases}$$

# 8

1. 我们在神经网络中随机初始化权重
2. 我们将第一组输入值发送到神经网络，并通过它传播值以获得输出值。
3. 我们将输出值与预期输出值进行比较，并使用成本函数计算误差。
4. 我们将错误反向传播回网络并根据该信息设置权重。
5. 对于训练集中的每个输入值，重复步骤2到4。
6. 当整个训练集通过神经网络发送时，我们已经完成了一个epoch。在那之后，我们重复更多的epoch。