

# Week1 - 机器学习的基本概念简介

## 几种机器学习任务

回归(Regression) : 函数返回一个数值

分类(Classification) : 给定多个选项, 函数返回正确的一个选项

结构化学习(Structured Learning) : 产生结构化的东西(图片,文档)

## Machine Learning Framework

以视频播放数为例

### 1. Function with Unknown Parameters

可以假设观看人数函数  $y = f(x)$  为  $y = b + wx_1$ , 其中

- $x_1$ : 2/25的观看人数
- $y$ : 2/26的观看人数
- $b, w$ : 未知的参数(从数据中学习)

其中, 通常将

- $y = b + wx_1$  称为 model
- $b, w$  称为 feature
- $w$  称为 weight,  $b$  称为 bias

### 2. Define Loss from Training Data

Loss : 一个未知参数的函数  $L(b, w)$ , 输出的值代表这组参数是否合适

#### Example

已知过去一年每天视频的观看人数

| Date       | Viewer - ( $\hat{y}$ ) |
|------------|------------------------|
| 2017/01/01 | 4.8k                   |
| 2017/01/02 | 4.9k                   |
| 2017/01/03 | 7.5k                   |
| ...        | ...                    |

| Date       | Viewer - ( $\hat{y}$ ) |
|------------|------------------------|
| 2017/12/30 | 3.4k                   |
| 2017/12/31 | 9.8k                   |

令  $b = 0.5k$ ,  $w = 1$ , 可以根据数据  $\hat{y}$  得出下一天的观看人数  $y = 0.5k + 1x_1$ , 和每一天的误差  $e_i = |y - \hat{y}|$

| Date       | Viewer - $\hat{y}$ | Predict - $y$ | Error - $e_i$ |
|------------|--------------------|---------------|---------------|
| 2017/01/01 | 4.8k               | -             | -             |
| 2017/01/02 | 4.9k               | 5.3k          | 0.4k          |
| 2017/01/03 | 7.5k               | 5.4k          | 1.9k          |
| ...        | ...                | ...           | ...           |
| 2017/12/30 | 3.4k               |               |               |
| 2017/12/31 | 9.8k               | 10.3k         | 0.5k          |

Loss:

$$L = \frac{1}{N} \sum_n e_n$$

### Loss的不同表示方法

- MAE(平均绝对误差) : 若  $e = |y - \hat{y}|$ ,  $L$  表示 Absolute error(绝对误差)
- MSE(平均均方误差) : 若  $e = (y - \hat{y})^2$ ,  $L$  表示 Square error(均方误差)

若  $y$  和  $\hat{y}$  都是概论数值的话, 选用 Cross-entropy(交叉熵)

## 3. Optimization

$$w^*, b^* = \arg \min_{w, b} L$$

### Gradient Descent (梯度下降)

- (Randomly) 选择一个初始值  $w^0, b^0$
- 计算微分  $\frac{\partial L}{\partial w} |_{w=w^0, b=b^0}$ ,  $\frac{\partial L}{\partial b} |_{w=w^0, b=b^0}$ .
  - 若计算结果为正则增加  $w$ , 反之减少
- $w^1 \leftarrow w^0 - \eta \frac{\partial L}{\partial w} |_{w=w^0, b=b^0}$ ,  $b^1 \leftarrow b^0 - \eta \frac{\partial L}{\partial b} |_{w=w^0, b=b^0}$ 
  - $\eta$  为 Learning rate (学习速率), 影响  $w, b$  的步长.
  - 类似  $\eta$  需要自己设定的值, 称之为 hyperparameters
- 迭代更新  $w, b$

当  $\frac{\partial L}{\partial w}|_{w=w^T, b=b^T} = 0, \frac{\partial L}{\partial b}|_{w=w^T, b=b^T} = 0$  时, 梯度下降算法结束, 此时的  $W^T, B^T$  可能并非最优解, 再该状况下, 称该点为 Local minima (局部最优解), 而真正的最优解称之为 Global minima (全局最优解). 在实际机器学习中, 局部最优解并不是真正的问题

## Linear Model

根据 Gradient Descent, 得到  $w^* = 0.97, b^* = 0.1k$ . 此时, 训练集的Loss为  $L = 0.48k$ , 预测集的Loss  $L' = 0.58k$

根据规律, 一般而言观看人数呈7天一循环, 此时将模型

$$y = b + wx_1$$

更新为

$$y = b + \sum_{j=1}^7 w_j x_j$$

此时, 训练集的Loss为  $L = 0.38k$ , 预测集的Loss  $L' = 0.49k$

在上述的例子中, 模型  $y = b + wx_1$  和  $y = b + \sum_{j=1}^7 w_j x_j$ , 称之为 Linear Model (线性模型)

Linear Model 存在很大的限制, 存在 Model Bias, 无法表示出复杂的情况.

## Use Sigmoid Function to create New Model

### Sigmoid Function

$$\begin{aligned} y &= c \frac{1}{1 + e^{-(b+wx_1)}} \\ &= c \text{sigmoid}(b + wx_1) \end{aligned}$$

- Different  $w$ : change slopes
- Different  $b$ : Change shift
- Different  $c$ : Change height

通过多个 Sigmoid Function 组合, 可以得到新的模型函数:

$$y = b + \sum_i c_i \text{sigmoid}(b_i + w_i x_1)$$

可以得到 Sigmoid Function 组合而成的 Model 拥有更多的 Features,

$$\begin{aligned} y &= b + wx_1 \\ \Rightarrow y &= b + \sum_i c_i \text{sigmoid}(b_i + w_i x_1) \end{aligned}$$

$$y = b + \sum_j w_j x_j$$

$$\Rightarrow y = b + \sum_i c_i \operatorname{sigmoid}(b_i + \sum_j w_{ij} x_j)$$

## Example

以  $i = 1, 2, 3; j = 1, 2, 3$  为例

对于

$$y = b + \sum_i c_i \operatorname{sigmoid}(b_i + \sum_j w_{ij} x_j)$$

$$r_1 = b_1 + w_{11}x_1 + w_{12}x_2 + w_{13}x_3$$

$$r_2 = b_2 + w_{21}x_1 + w_{22}x_2 + w_{23}x_3$$

$$r_3 = b_3 + w_{31}x_1 + w_{32}x_2 + w_{33}x_3$$

以矩阵方式表示

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

简写为

$$r = b + Wx$$

从而得到  $a$

$$a_1 = \operatorname{sigmoid}(r_1) = \frac{1}{1 + e^{-r_1}}$$

$$a_2 = \operatorname{sigmoid}(r_2) = \frac{1}{1 + e^{-r_2}}$$

$$a_3 = \operatorname{sigmoid}(r_3) = \frac{1}{1 + e^{-r_3}}$$

简写为

$$a = \sigma(r)$$

得到

$$y = b + c^T a$$

综上

$$y = b_1 + c^T \sigma(b_2 + Wx)$$

其中  $b_1, b_2$  为两个不同的 Bias 向量

在该方程中,

- Feature:  $x$
- Unknown parameters:  $W, b_1, c^T, b_2$

## Optimization of New Model

将所有 Unknown parameters 的行或列拆分组成一个向量  $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \end{bmatrix}$

$$\theta^* = \arg \min_{\theta} L$$

如同 Linear Model, 矩阵  $\theta$  每项求微分可写成

$$g = \begin{bmatrix} \frac{\partial L}{\partial \theta_1} \big|_{\theta=\theta^0} \\ \frac{\partial L}{\partial \theta_2} \big|_{\theta=\theta^0} \\ \vdots \end{bmatrix} = \nabla L(\theta^0)$$

其中称  $g$  为 gradient

同理,

$$\begin{bmatrix} \theta_1^1 \\ \theta_2^1 \\ \vdots \end{bmatrix} \leftarrow \begin{bmatrix} \theta_1^0 \\ \theta_2^0 \\ \vdots \end{bmatrix} - \begin{bmatrix} \eta \frac{\partial L}{\partial \theta_1} \big|_{\theta=\theta^0} \\ \eta \frac{\partial L}{\partial \theta_2} \big|_{\theta=\theta^0} \\ \vdots \end{bmatrix}$$

可以简写为

$$\theta^1 \leftarrow \theta^0 - \eta g$$

计算  $\theta^*$  步骤如下:

- (Randomly) 选择一个初始值  $\theta^0$
- 计算  $g = \nabla L(\theta^0)$
- $\theta^1 \leftarrow \theta^0 - \eta g$
- 迭代更新  $\theta$

通常来说, 我们会将一段长度为  $N$  的数据分成  $n$  段 batch, 每段 batch 有  $B$  个数据, 第  $i$  个 batch 的 Loss 记作  $L^i$ , 而并非计算整一段数据

步骤如下:

- 选择一个初始值  $\theta^0$
- 计算第一段 batch 的 gradient  $g^1 = \nabla L^1(\theta^0)$ ,  $\text{update} \theta^1 \leftarrow \theta^0 - \eta g^1$
- 计算第二段 batch 的 gradient  $g^2 = \nabla L^2(\theta^1)$ ,  $\text{update} \theta^2 \leftarrow \theta^1 - \eta g^2$
- ...
- 迭代计算  $\theta$

epoch: 所有 batch 都被遍历过一遍, 称之为 1 epoch

update:  $\theta$  被更新一边称之为 1 次 update

Example

- 数据长度  $N = 10000$ , Batch 大小  $B = 10$ , 一个 epoch 经过了 1000 次 update
- 数据长度  $N = 1000$ , Batch 大小  $B = 100$ , 一个 epoch 经过了 10 次 update

## Rectified Linear Unit (ReLU)

$$y = c \max(0, b + wx_1)$$

通常将 Sigmoid function 和 ReLU function 等类似的函数称之为 Activation function(激活函数)

## Newral Network (神经网络)

通常, 一个 Sigmoid 就是一个 neuron, 很多个 neuron 就组成了 neural network(神经网络). neuron 层叫 layer, 输入输出之外的层叫 hidden layer. 有很多层就叫 Deep learning