第一章 机器学习基础

1.1 机器学习简介

人工智能是我们想要达到的目标,机器学习则是实现人工智能的手段,深度学习则是机器学习的其中一种。

那么机器学习是什么?机器学习可以看做是从数据中学习一个函数 (function),对于给定输入得到输出结果。如在语音辨识、图像识别等领域的应用。

机器学习框架如图 1.1 所示,首先包含一系列函数 model 的集合,利用训练数据评价函数的品质,并挑选出最优函数模型。

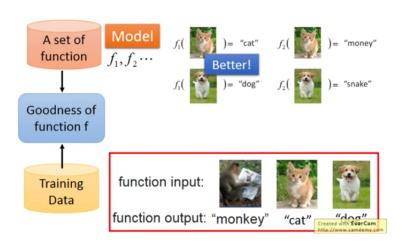


图 1.1: 机器学习框架

详细步骤如图 1.2 所示,可以总结为:

- 1. 挑选模型
- 2. 评价函数品质 goodness
- 3. 挑选最优函数 f^*

机器学习的学习图谱如图 1.3 所示, 具体描述如下:

1. 监督学习

回归

线性模型

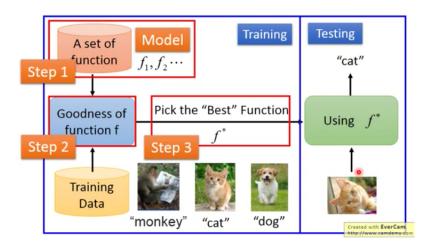


图 1.2: 机器学习三步骤

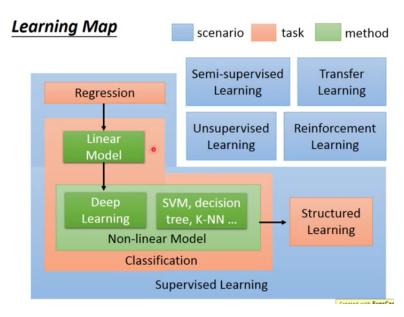


图 1.3: 机器学习图谱

深度学习,非线性模型 其它非线性模型,如 SVM、决策树、knn。 structure learning

- 2. 无监督学习
- 3. 半监督学习
- 4. 迁移学习
- 5. 强化学习

1.2 回归问题 3

1.2 回归问题

线性模型: $y = b + \sum w_i x_i$, 其中 x_i 为输入数据的特征, w_i 为权重, b 为偏置。使用损失函数评价选定模型的好坏。如对于模型 f,样本 x^n ,对应的输出真值为 \hat{y} :

$$L(f) = \sum_{n=1}^{N} (\hat{y}^{n} - f(x_{cp}^{n}))^{2}$$

对于线性模型:

$$L(w,b) = \sum_{n=1}^{N} (\hat{y}^{n} - (b + w \cdot x_{xp}^{n}))^{2}$$

最优化模型:

$$w^*, b^* = \arg\min_{w,b} L(w,b) = \arg\min_{w,b} \sum_{n=1}^{N} (\hat{y}^n - (b + w \cdot x_{cp}^n))^2$$

为求得最优解,使用梯度下降法进行优化求解。若将b看做权重w的一部分,优化模型:

$$w^* = \arg\min_w L(w)$$

权重通过梯度进行迭代:

$$w^1 \leftarrow w^0 - \eta \left. \frac{dL}{dw} \right|_{w=w^0}$$

其中, η 为学习率。梯度下降实例如图 1.4所示: 对参数的偏导:

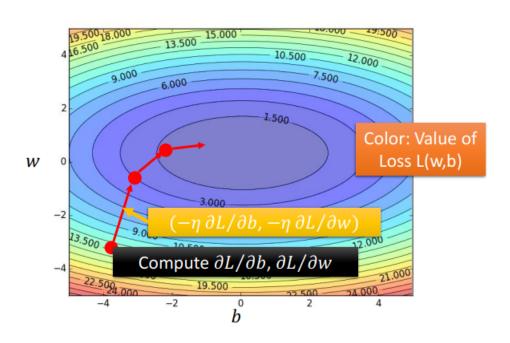


图 1.4: gradient descent

$$\frac{\partial L}{\partial w} = \sum_{n=1}^{N} 2 \left(\hat{y}^n - (b + w \cdot x^n) \right) (-x^n)$$

$$\frac{\partial L}{\partial b} = -\sum_{n=1}^{N} 2 \left(\hat{y}^n - (b + w \cdot x^n) \right)$$



A more complex model does not always lead to better performance on *testing data*.

This is **Overfitting**. Select suitable model

图 1.5: 过拟合现象

对于更为复杂的模型,如:

$$y = b + w_1 \cdot x + w_2 \cdot x^2$$

$$y = b + w_1 \cdot x + w_2 \cdot x^2 + w_3 \cdot x^3$$

$$y = b + w_1 \cdot x + w_2 \cdot x^2 + w_3 \cdot x^3 + \dots + w_5 \cdot x^5$$

过于复杂的模型在训练集上能够取得小的误差,但在测试集的误差会异常大,即发生了过拟合 (overfitting)。同时对于复杂模型使用简单的模型会出现欠拟合现象,不同模型在训练集和测试集上的误差如图 1.5所示:为缓解过拟合,可以通过正则化实现,对权重 w 加以约束。

$$L = \sum_{n=1}^{N} \left(\hat{y}_n - (b + \sum w_i x_{ni}) \right)^2 + \lambda \sum w_i^2$$

正则化参数的选择不宜过大或过小,以宝可梦 cp 值回归模型为例,参数 λ 影响如图 1.6所示:

1.3 误差来自何处

简单的模型有着较大的偏差 (bias),使用相同模型,不同数据得到的最优函数 f^* 的取值区间可能不包括理论最优解 \hat{f} ,因此偏差较大。而对于过于复杂的模型,又容易产生对数据的过拟合,出现大的方差。对于复杂模型,多次实验求取的模型均值能够更加接近理论解 \hat{f} ,但同时一个模型对于新的测试集数据容易产生大的误差,此时误差主要来自方差 (variance)。偏差与误差的形象解释如图 1.7所示:

针对方差和偏差导致的过拟合和欠拟合问题,分别通过以下方法改善:

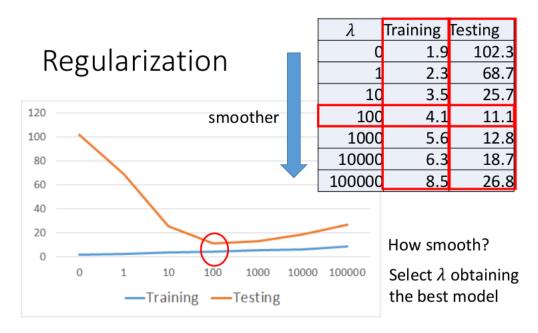
偏差

重新修正模型,使用更加复杂的模型 使用更多的样本特征

方差

增加样本数据量 模型正则化

1.3 误差来自何处 5



- \triangleright Training error: larger λ , considering the training error less
- ➤ We prefer smooth function, but don't be too smooth.

图 1.6: 正则化参数对模型误差的影响

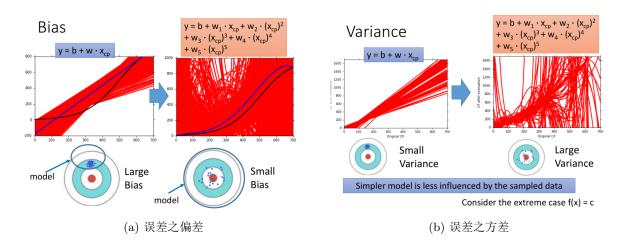


图 1.7: bias VS variance

N-fold Cross Validation Cross Validation **Training Set** Model 1 Model 2 Model 3 public private Err = 0.2 Err = 0.4 Err = 0.4 Training Set **Testing Set Testing Set** Err = 0.4 Err = 0.5 Err = 0.5 Using the results of public testing Err = 0.3 Err = 0.6 Err = 0.3 data to tune your model Avg Err Avg Err Avg Err You are making public set = 0.5 better than private set. Model 1 • Err = 0.9 Not recommend Model 2 + Err = 0.7 Testing Set Model 3 +Err = 0.5 Err > 0.5 ----- Err > 0.5 public private (a) 划分验证集 (b) N-折交叉验证

图 1.8: 交叉验证

对于在实际训练模型过程中,训练集上得到的模型,通过 public 测试集得到的错误率与在 private 测试集上的错误率不一定是一致的。为了在没有 private 测试集的前提下能够较好的预知错误率,可以通过在训练集上划分训练集和验证集加以实现。如图 1.8subfig:validation 所示:同时为了充分利用训练集数据可以采用 N-折交叉验证。模型的正确率由 N 个正确率的均值决定。如图 1.8b所示。

1.4 梯度下降

损失函数的优化求解可以通过梯度下降法实现,对于以下参数优化:

$$\theta^* = \arg\min L(\theta) \tag{1.1}$$

其中,L 为损失函数, θ 为优化参数。使用梯度下降迭代求解:

$$\theta_n = \theta_{n-1} - \eta \frac{\partial L}{\partial \theta} \bigg|_{\theta_{n-1}} \tag{1.2}$$

- 1.4.1 学习率微调
- 1.4.2 随机梯度下降法 Stochastic Gradient Descent
- 1.4.3 特征缩放
- 1.4.4 梯度下降数学理论

标题 12425 dfadf

自定义数据