

# 动手学深度学习

---

[toc]

## 前言

### 基础了解

深度学习与一门学科结合：

- 以特定的方式提出问题的动机
- 给定建模方法的数学
- 将模型拟合数据的优化算法
- 有效训练模型、克服数值计算并最大限度地利用现有硬件的工程方法

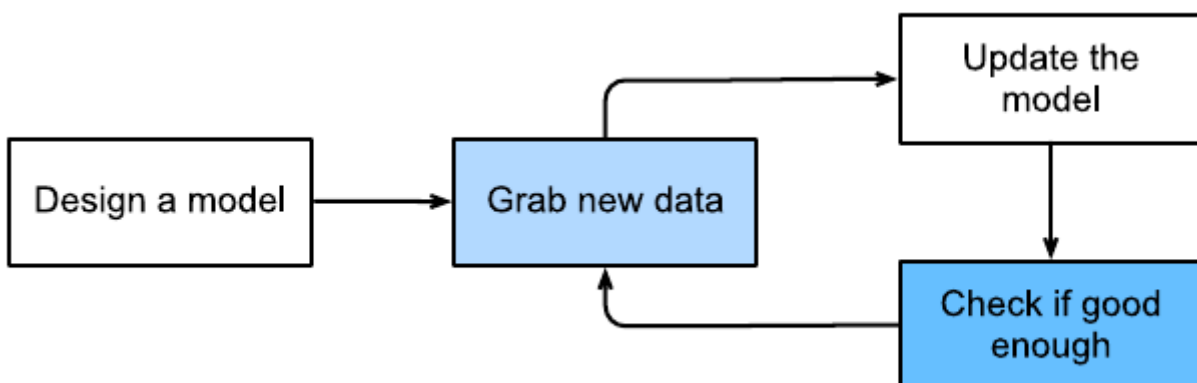
个人需要的能力：

- 批判性思维技能
- 解决问题所需要的数学知识
- 解决方案所需的软件工具

### 生活中的机器学习

#### 基本词汇

- 参数：通过调整参数，来调节目程序的行为。
- 模型族：通过调整参数而生成不同程序的集合(程序：输入输出的映射)。
- 学习算法：使用数据集选择参数的元程序。
- 训练流程：



在使用及其学习解决问题之前，必须精确定义问题，确定输入输出的性质，选择合适的模型族。

1. 确定一个模型结构，并初始化参数
2. 获取数据样本
3. 调整参数
4. 重复获取2、3两个步骤

## 关键组件

机器学习的关键组件包括四个部分：

- 我们可以学习的数据
- 如何转换数据的模型
- 一个目标函数，用来量化模型的有效性
- 调整模型参数以优化目标函数的算法

### 1. 数据：

每个数据集由样本构成，每个样本由特征构成。每个样本遵循iid的条件。

每个样本的特征数量都一致，有相同的长度，长度被成为数据的维度。

数据量越大，越有助于训练更强大的模型，从而减少对冗余设想的依赖。

Garbage in, garbage out。错误的数据会造成效果差。比如：数据分布不均匀（某种情况数据量少）；数据本身有偏见（数据本身就存在偏见，低质量）。

数据分为训练集和测试集。

数据集被分为三个部分：

1. 训练集(用于拟合模型参数)
2. 验证集(用于评估拟合效果)
3. 测试集(通常测试集是不可以见的，验证集用于代替测试集)

### 2. 模型

机器学习从数据中学习，学习是指自主提高模型完成某些任务的效能。  
深度学习注重强大的模型，将神经网络进行堆叠交织。

### 3. 目标函数

目标函数定义为模型的优劣程度，根据模型的参数（受决定于数据集）的具体情况，定义一个可以优化的目标函数，寻找到最佳的模型。

目标函数：

预测数值：平方误差。

分类问题：错误率

### 4. 优化算法

梯度下降方法：梯度的方向是函数在给定点上升最快的方向，那么梯度的反方向就是函数在给定点下降最快的方向

## 分类

### 监督学习

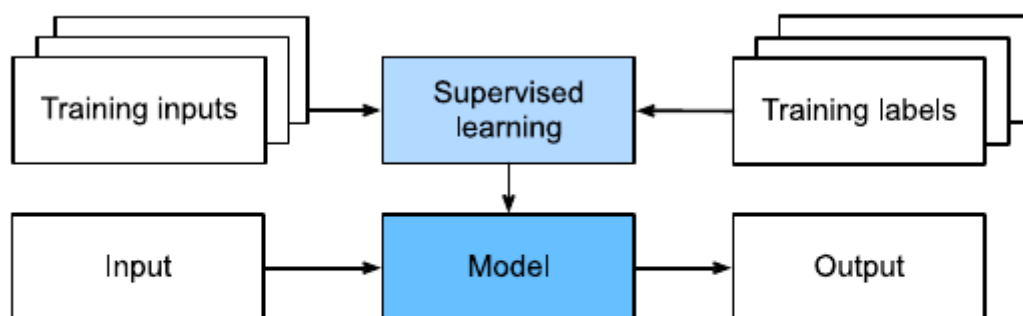


图1.3.1: 监督学习

- 回归
- 分类
- 标记
- 搜索
- 推荐系统
- 序列学习

### 无监督学习

- 聚类
- 主成分分析
- 因果关系和概率图模型问题
- 生成对抗性网络

### 与环境互动

主要应对以下几种问题：

1. 环境对模型是否重要
2. 环境是否有助于建模（语音识别）
3. 环境是否想打败模型（垃圾邮件过滤）
4. 环境是否变化（distribution shift）

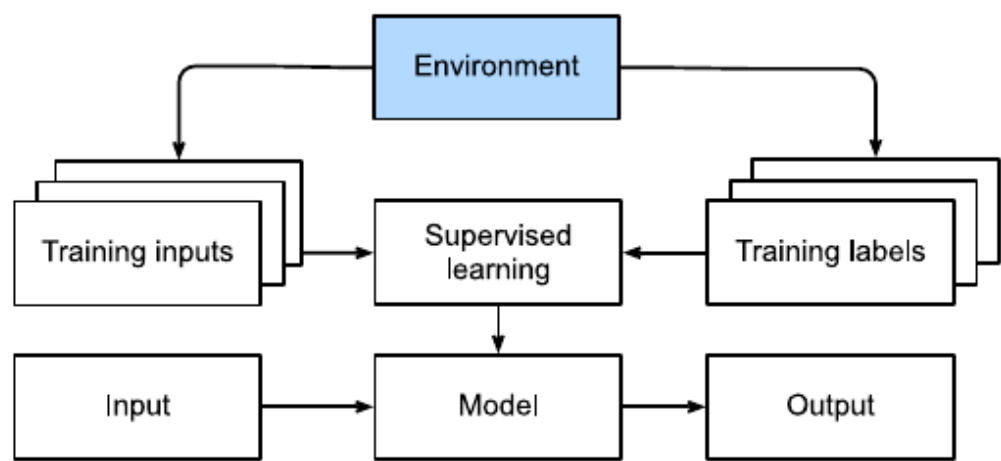


图1.3.6: 从环境中为监督学习收集数据。

强化学习

强化学习的过程是：agent持续接受environment的一些observation。

agent接收到environment的observation  
agent接收，并选择一个action  
action通过某种机制传输到environment  
agent从environment获得reward  
循环第一步

目标是产生一个好的策略

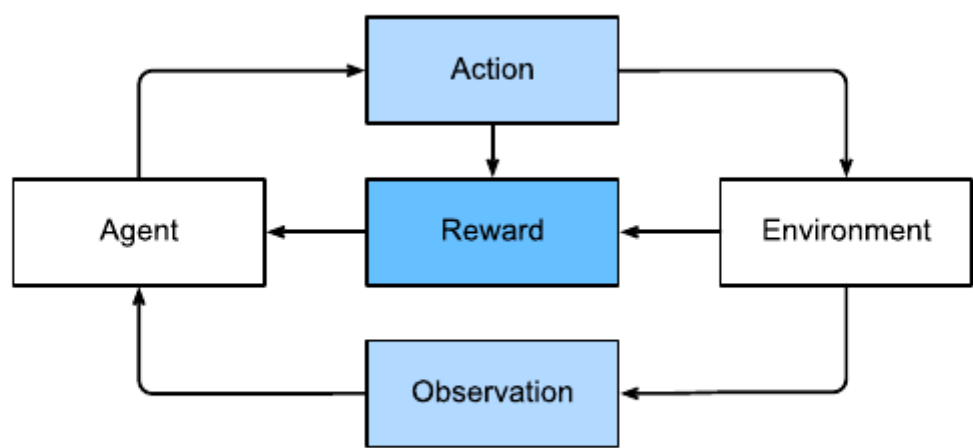


图1.3.7: 强化学习和环境之间的相互作用

近年的进展

dropout attention机制 多阶段设计 生成对抗网络 并行计算能力提升 可伸缩算法 深度学习框架

## 基础操作

### 基础运算符

按照元素进行计算(+ - \* \*\* / ==)

### 基本运算

```
torch.cat
torch.dot
torch.mv
torch.mm
torch.exp
torch.cumsum
X.sum(dim) //sum默认会降维，可以通过keepdims属性设置
X.mean()
X.numel()
```

广播机制：维度不同的张量会扩展成相同类型，再进行基本操作。

切片：

```
X[1:4:2, 1:-1:4]
```

节省内存：

切片法，避免重新开辟空间(Memo[:] = <expression>)  
避免轻易对变量转换类型，可以通过调用变量本身的方法

## 数据预处理

```
1. 读取数据集 //pd.read_csv
2. NAN值处理 //data.fillna()
3. onehot编码 //data.getdummies()
4. 转换成张量 //torch.tensor(data.values)
```

## 微分

目标函数反映模型的效果。拟合的主要体现在两个方面：优化和泛化。都需要通过微分来寻找

- 自动求导(反向传播、Y.sum().backward()、Y.detach())
- 反向传播

神经网络沿着输出层到输入层的顺序计算梯度的方法，依据的是链式法则

- 正向传播

神经网络沿着输入层到输出层的顺序，依次进行计算并存储模型的中间变量。比如：输入层、隐藏层、输出层、loss。

## MLE

$Y = P(X|\theta)$ ，在已知一部分X、Y数据的基础上，求取 $\theta$ 。采用的MLE，需要Y关于 $\theta$ 的函数达到最大值即可。

## 线性回归