

# 人工智能导论

Introduction to Artificial Intelligence

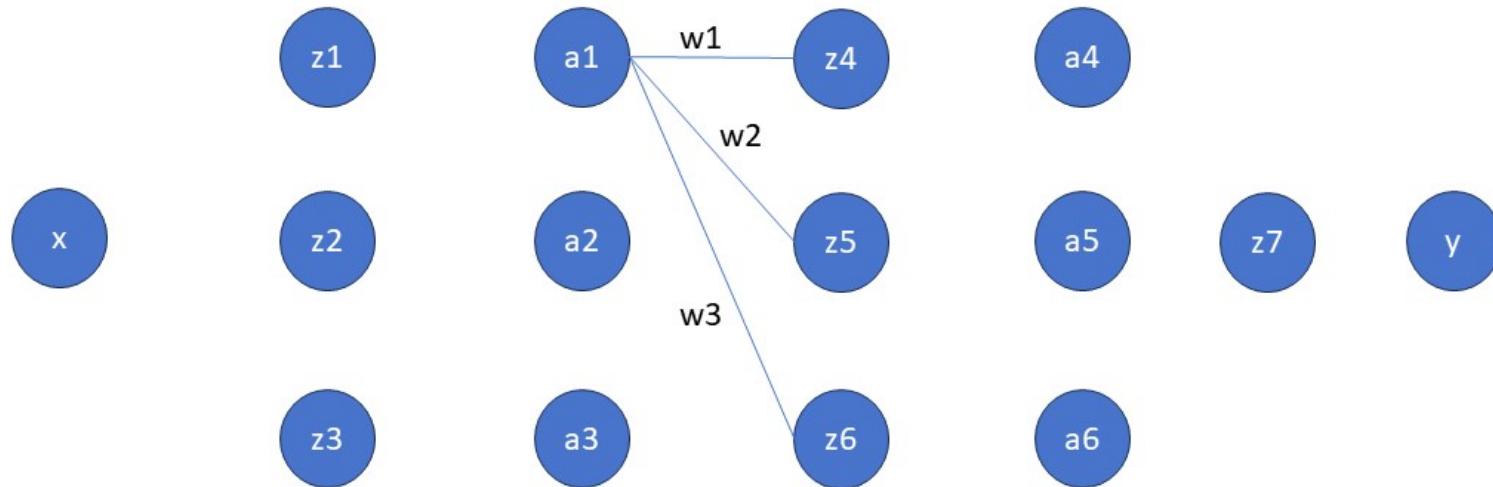
课前练习



1

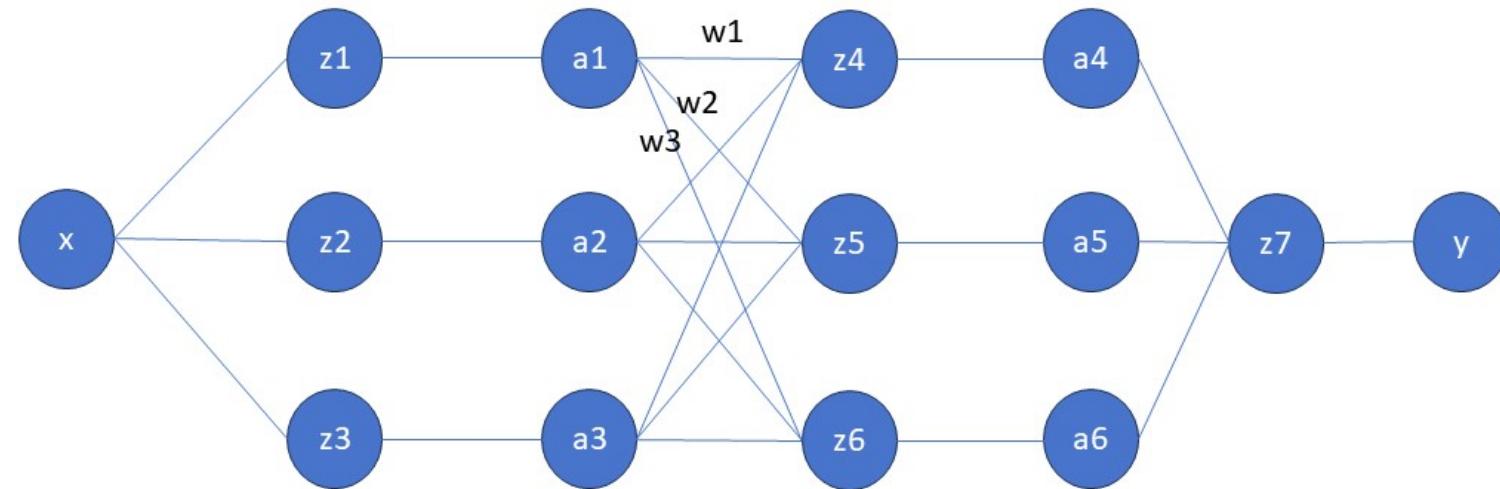
## 主观题 10分

1. 如果这是一个全连接网络，神经元之间该如何连接？（在图上连线）
2. 已知损失为L，且 $\frac{\partial L}{\partial z_4}$ ， $\frac{\partial L}{\partial z_5}$ ， $\frac{\partial L}{\partial z_6}$ 的值为 $\delta_4$ 、 $\delta_5$ 、 $\delta_6$ ，求 $\frac{\partial L}{\partial a_1}$ 。



2

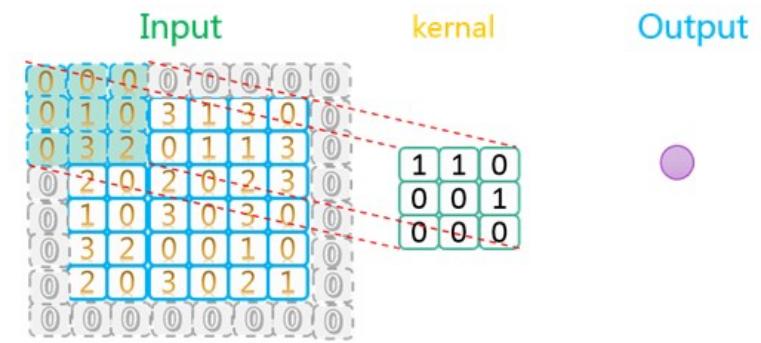
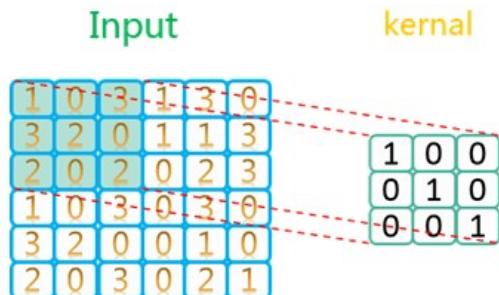
## 课程回顾



$$\frac{\partial L}{\partial a_1} = \frac{\partial L}{\partial z_4} \frac{\partial z_4}{\partial a_1} + \frac{\partial L}{\partial z_5} \frac{\partial z_5}{\partial a_1} + \frac{\partial L}{\partial z_6} \frac{\partial z_6}{\partial a_1} = \delta_4 w_1 + \delta_5 w_2 + \delta_6 w_3$$

## 填空题 3分

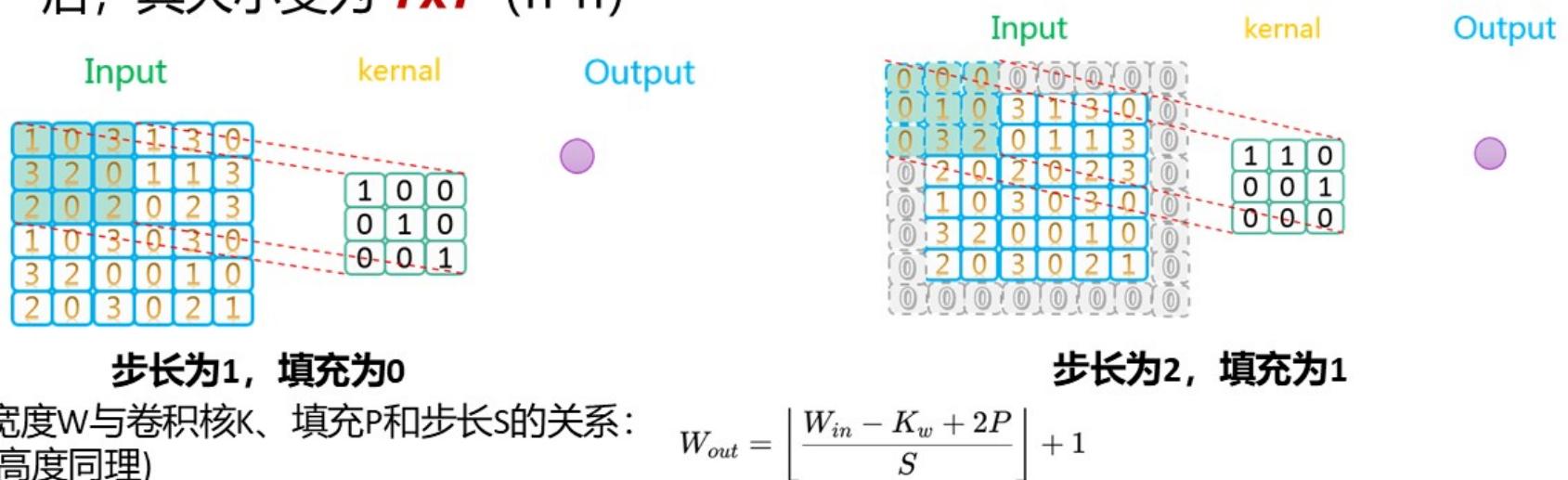
- 对于一个 $28 \times 28$ 的输入图像，经过 5 次步长为 1、填充为 0 的 $3 \times 3$ 卷积操作后，输出图像的大小为：[填空1] ( $n \times n$ )
- 对于一个 $28 \times 28$ 的输入图像，经过 5 次步长为 1、填充为 1 的 $3 \times 3$ 卷积操作后，输出图像的大小为：[填空2] ( $n \times n$ )
- 对于一个 $28 \times 28$ 的输入图像，经过 2 次步长为 2、填充为 0 的 $2 \times 2$ 池化后，其大小变为 [填空3] ( $n \times n$ )



宽度 $w$ 与卷积核 $K$ 、填充 $P$ 和步长 $s$ 的关系： $W_{out} = \left\lfloor \frac{W_{in} - K_w + 2P}{S} \right\rfloor + 1$   
(高度同理)

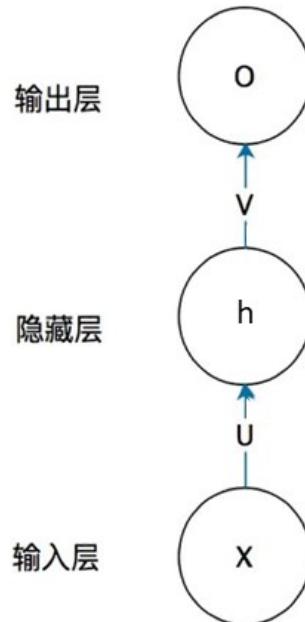
## 课程回顾

- 对于一个  $28 \times 28$  的输入图像，经过 5 次步长为 1、填充为 0 的  $3 \times 3$  卷积操作后，输出图像的大小为：**18x18** ( $n \times n$ )
- 对于一个  $28 \times 28$  的输入图像，经过 5 次步长为 1、填充为 1 的  $3 \times 3$  卷积操作后，输出图像的大小为：**28x28** ( $n \times n$ )
- 对于一个  $28 \times 28$  的输入图像，经过 2 次步长为 2、填充为 0 的  $2 \times 2$  池化后，其大小变为 **7x7** ( $n \times n$ )



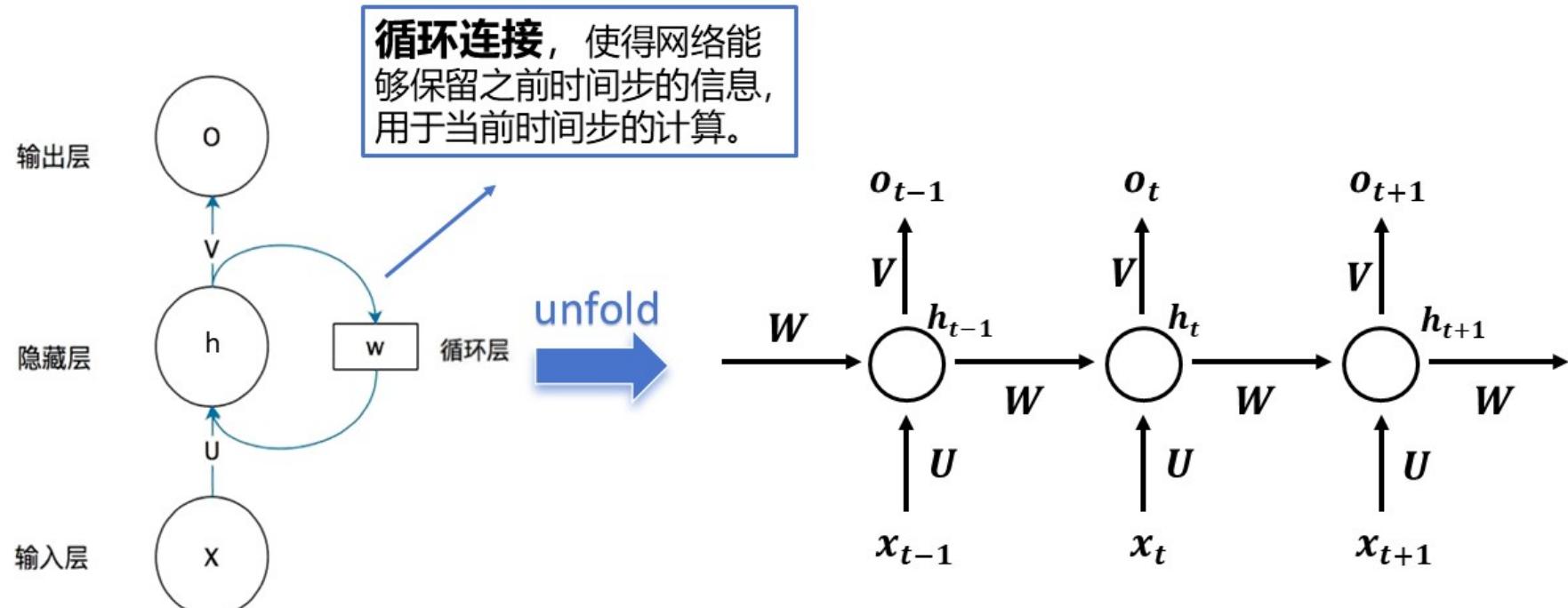
主观题 10分

这是一个简单神经网络的示意图，请你将其改造成循环神经网络  
(在图上加一条边)



6

## 课程回顾



## 循环神经网络

# 人工智能导论

Introduction to Artificial Intelligence

## 第四章 模型训练与优化



8

# 模型训练与优化

## 人工智能三要素

数据

算法

算力

①用什么训练

②模型如何学

③训练效果如何

④如何优化

⑤在什么上训练

# 模型训练与优化

## 人工智能三要素

数据

算法

算力

①用什么训练

**数据集准备与加载**

②模型如何学

③训练效果如何

④如何优化

⑤在什么上训练

10

# 数据集准备与加载

## 数据集作用



模型 = 汽车引擎



数据 = 汽油

- **没有数据，模型无法学习** → 引擎没油就无法启动
- **不同的任务依赖不同类型的数据** → 轿车加汽油，货车加柴油
- **数据质量影响模型性能** → 劣质燃油再好引擎也跑不快

基础

数据

模型训练

模型预测

11

# 数据集准备与加载

## 数据集介绍

数据集是AI模型学习的“教材”

- 包含一组有组织的**样本**（如图像、文本、音频等）
- 每个样本包含：**特征**（输入）和**标签**（输出/目标）

猫狗分类数据集



文本情感分类数据集

Sentence	Label
string - lengths	string - classes
20 ----- 828	13 values
Unfortunately later died from eating tainted meat NAME BBC documentary dynasties followed the marsh pride the lion episode was awesome	happiness
Last time I saw was loooong ago. Basically before LN announced they went for probabilistic routing.	neutral
You mean by number of military personnel? Because if you go by navy size or budget thats not even remotely true	neutral
Need to go middle of the road no NAME is going to vote for NAME	sadness
feel melty miserable enough imagine must	sadness
feel sense relief also sadness end colleagues anyway fab	happiness
think get feel weird ones use dryers time	surprise

# 数据集准备与加载

## 数据集介绍

数据集是AI模型学习的“教材”

- 包含一组有组织的**样本**（如图像、文本、音频等）
- 每个样本包含：**特征**（输入）和**标签**（输出/目标）

猫狗分类数据集



样本：一张“猫”图片

输入特征：图片

输出标签：猫（猫、狗）

文本情感分类数据集

Sentence	Label
string - lengths	string - classes
20 ----- 828	13 values
Unfortunately later died from eating tainted meat NAME BBC documentary dynasties followed the marsh pride the lion episode was awesome	happiness
Last time I saw was loooong ago. Basically before LN announced they went for probabilistic routing.	neutral
You mean by number of military personnel? Because if you go by navy size or budget thats not even remotely true	neutral
Need to go middle of the road no NAME is going to vote for NAME	sadness
feel meltly miserable enough imagine must	sadness
feel sense relief also sadness end colleagues anyway fab	happiness
think get feel weird ones use dryers time	surprise

样本：一段文本和其表达的情感类别

输入特征：文本

输出标签：happiness (surprise、happiness...)

13

# 数据集准备与加载

## 高质量数据集建设



国家数据局-高质量数据集建设指引

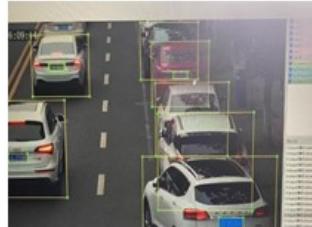
14

# 数据集准备与加载

## 高质量数据集建设



- 手工标注



- 传感器记录



数据采集是指通过软硬件手段从多种来源中获取原始数据的过程，为人工智能模型训练、大数据分析和业务决策提供基础数据支撑，对应于高质量数据集建设中的数据采集环节。

- 网络爬取



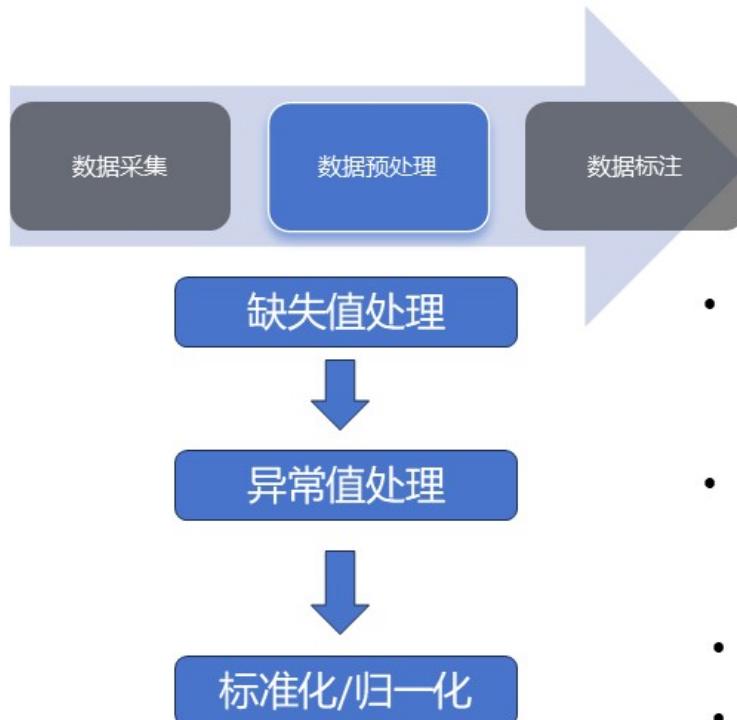
- 合成数据



15

# 数据集准备与加载

## 高质量数据集建设



收集得到的数据可能存在**数据质量差**（缺失值、异常值等）、**特征分布差异大**（例如，特征1取值范围为[0,1]，特征2取值范围为[0,1e6]）等问题

- 删除样本（缺失值数量少时）或填充均值/中位数/众数
- 检查极端值（常见方法有箱线图、Z-score等）
- **标准化 (Standardization)**：使特征均值为0，方差为1
- **归一化 (Normalization)**：将数据缩放到 [0,1] 区间

# 数据集准备与加载

## 高质量数据集建设



高质量数据集的数据标注环节主要是针对有监督机器学习的，其训练、验证和测试数据需要对单个或多个目标变量赋值。**标注质量往往直接影响人工智能模型的训练效果和性能表现。**

传统

**人工标注**

效率低下



趋势

**半自动化标注技术**: 通过引入人工智能辅助工具，减少人工劳动强度，提高标注效率与一致性。

**众包标注与分布式管理技术**: 搭建规模化协作平台，整合大量标注人员资源，解决大规模数据标注的人力瓶颈问题

**主动学习与模型辅助标注技术**: 利用模型预测指导标注优先级，提高标注资源的利用效率

17

# 数据集准备与加载

## 数据集划分

数据处理好后，我们如何组织这些数据来训练模型？

- **训练集**: 用于模型学习
- **验证集**: 调参、选择模型
- **测试集**: 评估泛化能力



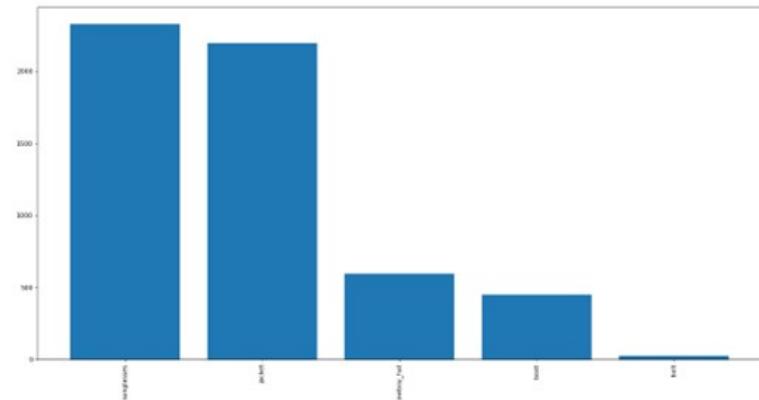
# 数据集准备与加载

## 常见划分方式

- **随机划分**: 将数据随机打乱后，按比例划分为训练集、验证集、测试集  
适用于分布均匀、样本间无明显依赖

当分布不均匀时？

- **分层划分**: 按照目标标签的分布进行**分层采样**，使训练集、验证集、测试集中的类别比例保持一致  
适用于分类任务中类别比例不均衡



19

# 数据集准备与加载

## 常见划分方式

当样本间有明显的依赖关系，比如时间依赖时？

- **时间顺序划分：**早期数据用于训练，后期数据用于验证和测试

适用于股票、气象等数据，必须时间先后划分，防止“未来信息泄露”



猜今天股价不能用明天的新闻

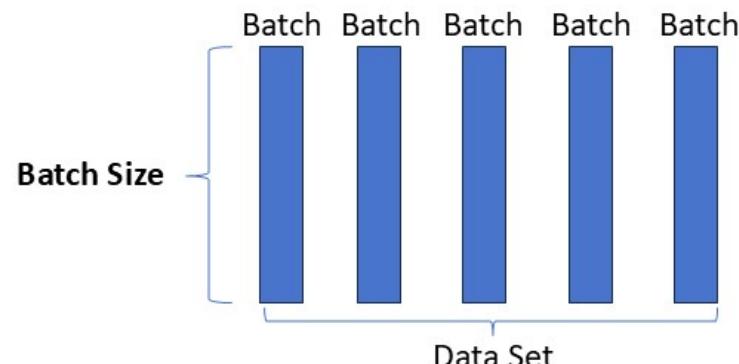
20

# 数据集准备与加载

## 数据批处理

当数据量级很大时，一次性加载入显存是不可能的  
怎么安排训练数据集？

- 模型每次不是一次性看完整个训练集，而是一小批一小批地读入数据来训练
- 每一批被称为一个 **batch** (批次)
- 一次完整遍历整个训练集称为一个 **epoch** (轮)



Batch Size 的常见设置

场景	Batch Size推荐范围
小模型 / 小数据集	16 ~ 64
中等规模训练	128 ~ 512
大模型 / 多GPU训练	1024以上

21

# 数据集准备与加载

## 数据加载

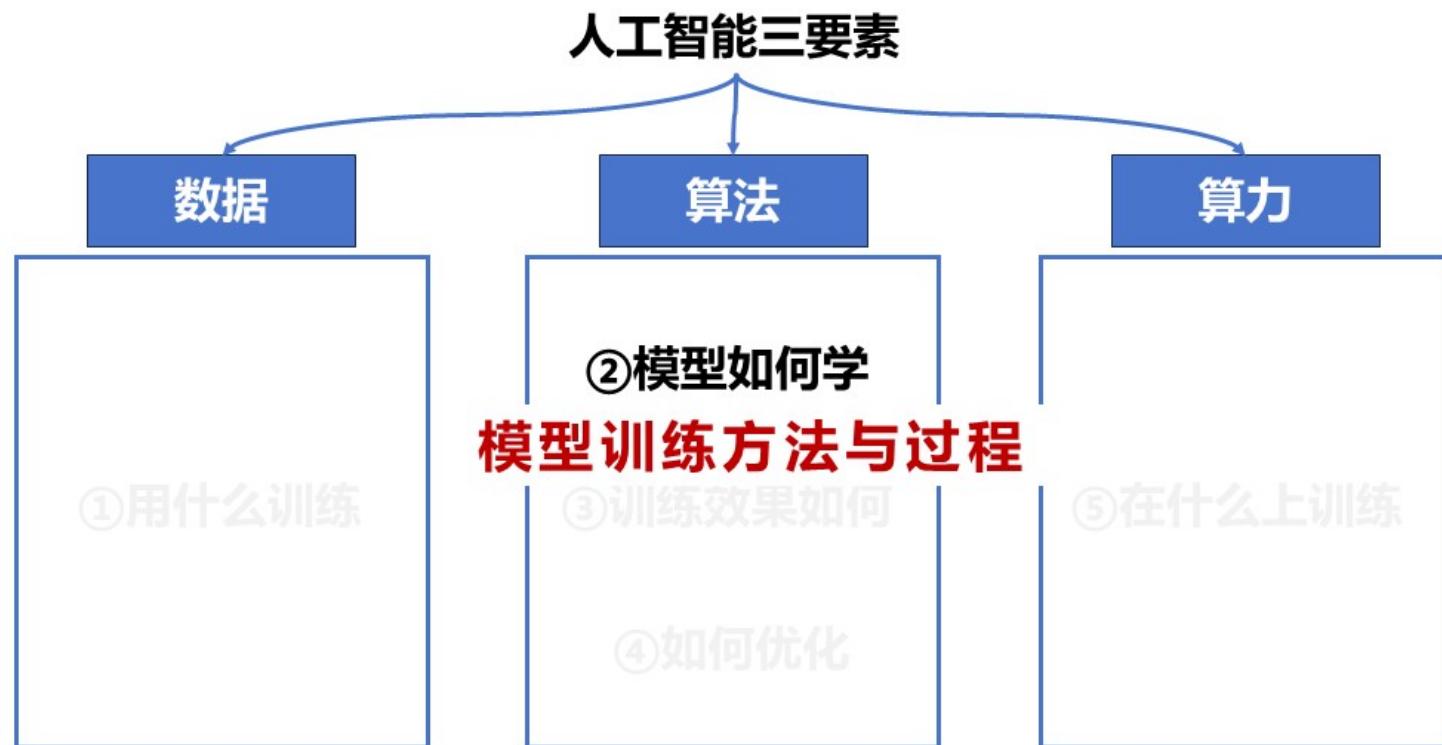
### Pytorch的数据加载



### 课后思考：

文本、图片、标签数据还需要进行什么处理么？[提示：模型只能处理数字。]

# 模型训练与优化



23

# 模型训练方法与过程

当数据准备就绪后，模型选择与训练成为实现任务的关键。

24

# 模型训练方法与过程

当数据准备就绪后，模型选择与训练成为实现任务的关键。

我们需要回答：

**选什么模型、如何初始化、用什么损失、怎么优化、怎样训练**

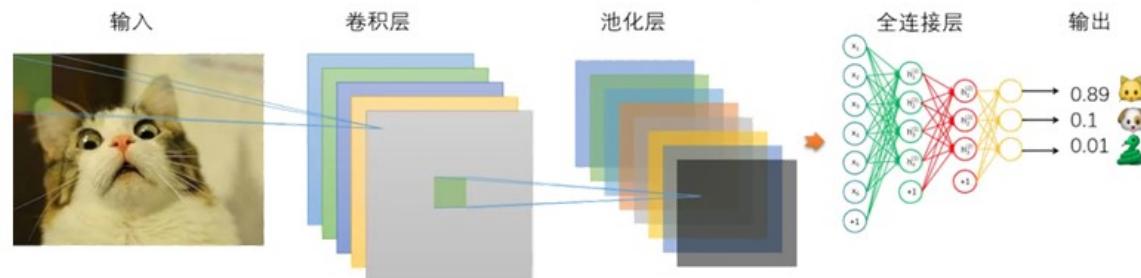
25

# 模型训练方法与过程

## 模型架构选取

不同模型适用于不同的任务，需要选择适合任务的模型架构

### 图像分类 → 卷积神经网络CNN



图像的相邻像素之间存在**空间局部相关性** → CNN通过**卷积操作**，就像一块可以在图像上滑动的小窗口（滤波器），能够捕捉到这些**局部特征**（如边缘、角点、纹理等）

**为什么不采用RNN？** 将图像逐像素线性展开后输入到 RNN，会丢失图像中天然的**空间布局信息**，导致模型难以捕捉像素间的**空间依赖关系**

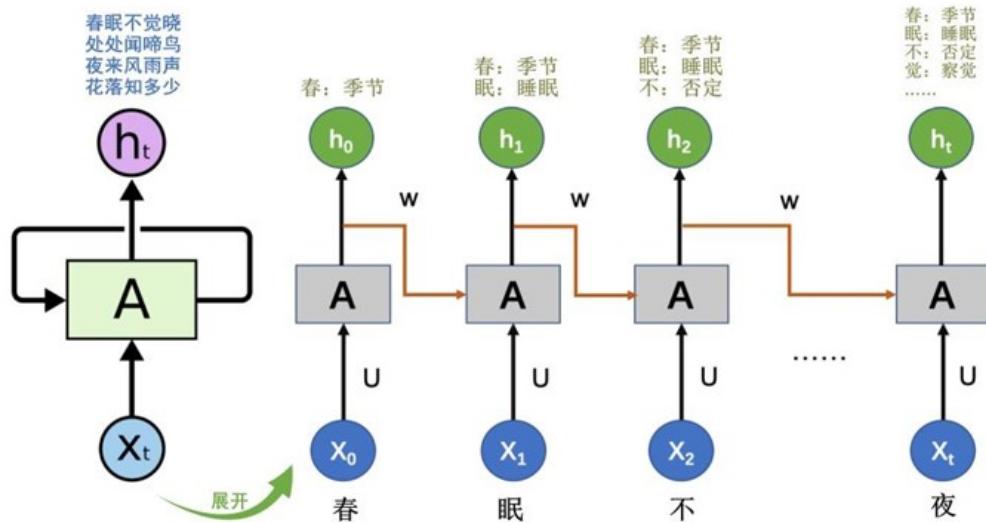
26

# 模型训练方法与过程

## 模型架构选取

不同模型适用于不同的任务，需要选择适合任务的模型架构

### 文本情感分类 → 循环神经网络RNN



**为什么不采用CNN？** 文本序列需要捕捉远距离的**上下文依赖**，卷积的感受野一般受限，因此CNN对长距离依赖的建模能力有限，难以建立全局的语义关联

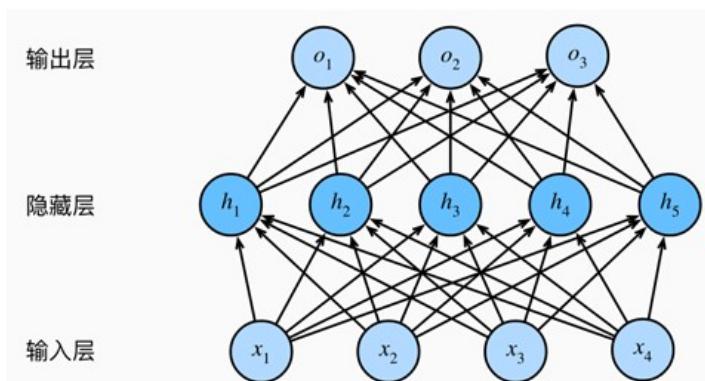
文本是一种典型的**序列数据**，单词的意义依赖于上下文 → RNN在设计上逐字/逐词读取输入，并将前面的信息通过隐藏状态传递到后续位置，能**捕捉语序上的依赖关系**

27

# 模型训练方法与过程

## 模型参数初始化

模型参数是什么？权重和偏置



模型的训练，实际上就是一个不断寻找最优参数的过程

$$O = \sigma(Hw_2 + b_2)$$

权重  
偏置

$$H = \sigma(Xw_1 + b_1)$$

模型参数决定了模型的性能

在训练之前需要先初始化模型参数

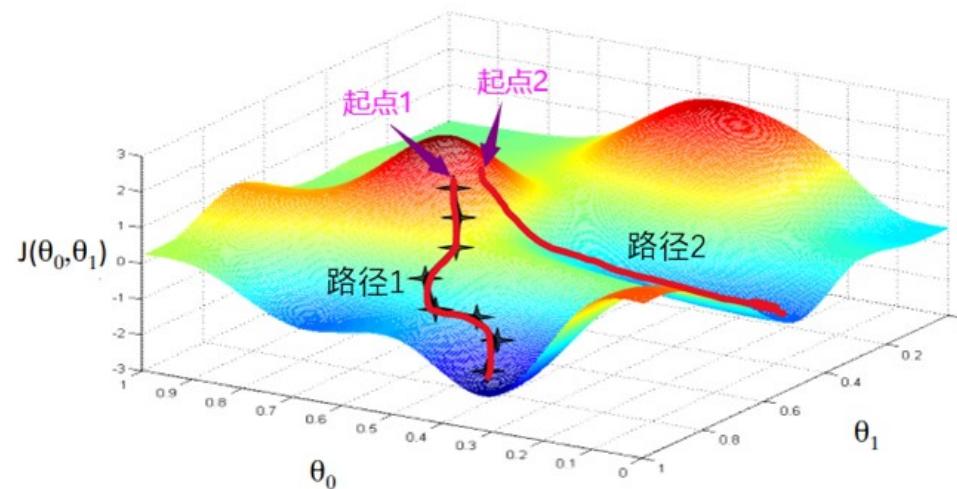
# 模型训练方法与过程

## 模型参数初始化

让模型从正确的起点出发

**合理初始化：**

- **提高训练效率：**良好的初始化可以帮助模型在训练中更快地收敛（**赢在起跑线**）



29

# 模型训练方法与过程

## 模型参数初始化

让模型从正确的起点出发

**合理初始化：**

- **提高训练效率：**良好的初始化可以帮助模型在训练中更快地收敛（**赢在起跑线**）
- **防止梯度消失/爆炸：**过小/过大的初始化值可能直接导致梯度过小/过大，使训练无法进行

梯度爆炸



梯度消失



30

# 模型训练方法与过程

## 模型参数初始化

让模型从正确的起点出发

**合理初始化：**

- **提高训练效率：**良好的初始化可以帮助模型在训练中更快地收敛（**赢在起跑线**）
- **防止梯度消失/爆炸：**过小/过大的初始化值可能直接导致梯度过小/过大，使训练无法进行

梯度爆炸



梯度消失



**课后思考：**

除了合理的参数初始化，梯度的消失  
/爆炸还会再什么情况发生呢？还能  
怎么避免？

31

# 模型训练方法与过程

## 模型参数初始化

让模型从正确的起点出发

**合理初始化：**

- **提高训练效率：**良好的初始化可以帮助模型在训练中更快地收敛（**赢在起跑线**）
- **防止梯度消失/爆炸：**过小/过大的初始化值可能直接导致梯度过小/过大，使训练无法进行



如果神经网络中所有神经元的权重都初始化为零，会有什么问题？

32

# 模型训练方法与过程

## 模型参数初始化

让模型从正确的起点出发

### 合理初始化：

- **提高训练效率**: 良好的初始化可以帮助模型在训练中更快地收敛 (**赢在起跑线**)
- **防止梯度消失/爆炸**: 过小/过大的初始化值可能直接导致梯度过小/过大，使训练无法进行
- **避免对称性问题**: 如果神经网络中所有神经元的权重都初始化为相同值 (如全为零)，则所有神经元梯度计算相同，学习相同的特征



如果神经网络中所有神经元的权重都初始化为零，会有什么问题？



不可以

33

# 模型训练方法与过程

## 模型参数初始化

让模型从正确的起点出发

### 合理初始化：

- **提高训练效率**: 良好的初始化可以帮助模型在训练中更快地收敛 (**赢在起跑线**)
- **防止梯度消失/爆炸**: 过小/过大的初始化值可能直接导致梯度过小/过大，使训练无法进行
- **避免对称性问题**: 如果神经网络中所有神经元的权重都初始化为相同值 (如全为零)，则所有神经元梯度计算相同，学习相同的特征

### 常用初始化方法：

- **随机初始化**: 将权重和偏置参数从某个随机分布中采样得到
- **加载预训练权重**: 不从头开始随机初始化，加载一个在大规模数据集上 (通常是相似任务) 训练好的模型的权重

34

# 模型训练方法与过程

## 损失函数选择

模型准备好之后，需要定义一个标准去告诉模型什么是‘好’，什么是‘坏’

- 均方误差(Mean Square Error, MSE)
  - 希望严惩大错误，误差越大，平方增长的越快

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2$$

- 平均绝对误差(Mean Absolute Error, MAE)
  - 希望对数据异常值更鲁棒，误差一视同仁

$$MAE(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i|$$

一般在回归任务中用于衡量预测值与真实值的距离

- 交叉熵损失 (Cross-Entropy Loss) – 二分类

$$L = -\frac{1}{m} \sum_{j=1}^m \left[ y^{(j)} \log(\hat{y}^{(j)}) + (1 - y^{(j)}) \log(1 - \hat{y}^{(j)}) \right]$$

一般在分类任务中用于衡量预测概率与真实标签分布差异

- 交叉熵损失 (Cross-Entropy Loss) – 多分类

$$L = -\frac{1}{m} \sum_{j=1}^m \sum_{i=1}^c y_{ji} \log(\hat{y}_{ji})$$

35

# 模型训练方法与过程

## 优化器选择

让模型一步步学得更好

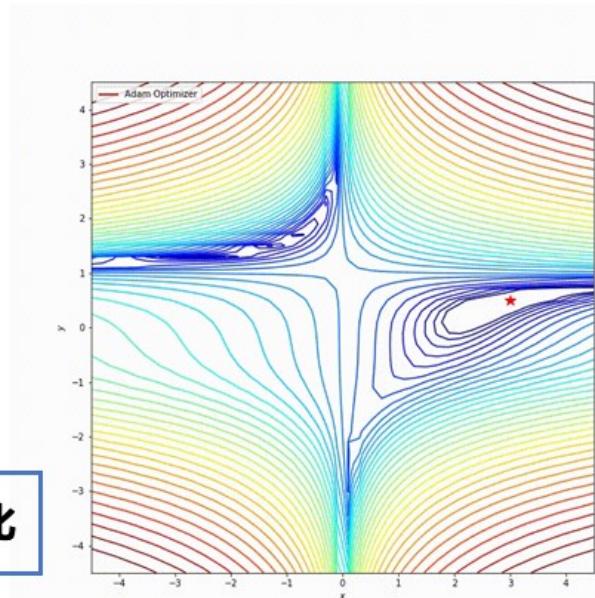
上节课神经网络基础中提到：

误差反向传播

- 1) 通过**损失函数**计算误差
- 2) 通过**反向传播**计算权重参数的梯度，用**优化算法**更新权重

优化器：通过梯度下降不断调整模型权重参数，使**损失函数最小化**

优化目标是指**最小化损失函数**，即通过调整模型参数，使模型预测结果尽可能接近真实标签或期望输出。



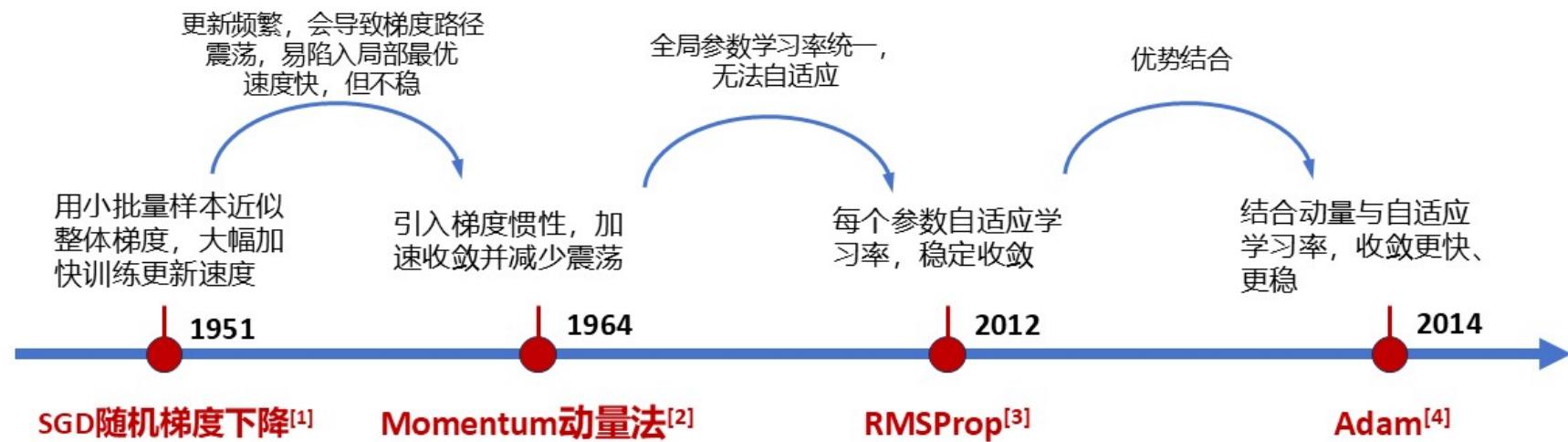
36

# 模型训练方法与过程

## 优化器选择

让模型一步步学得更好

通过梯度下降不断调整模型权重参数，使损失函数最小化



[1] Robbins H, Monroe S. A stochastic approximation method[J]. The annals of mathematical statistics, 1951: 400-407.

[2] Polyak B T. Some methods of speeding up the convergence of iteration methods[J]. Ussr computational mathematics and mathematical physics, 1964, 4(5): 1-17.

[3] Tieleman T. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude[J]. COURSERA: Neural networks for machine learning, 2012, 4(2): 26.

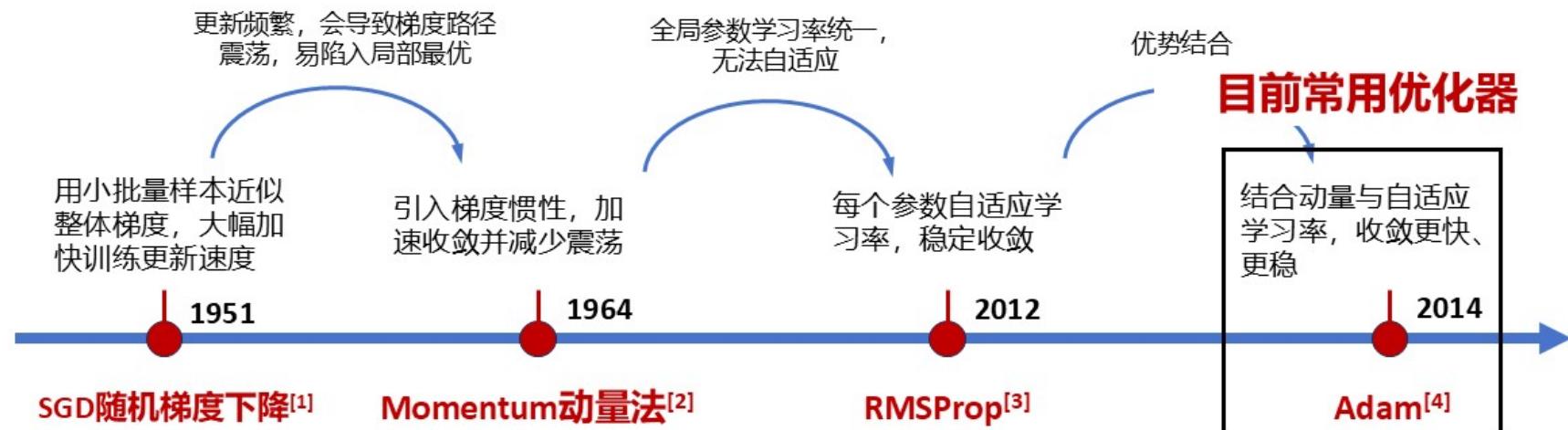
[4] Adam K D P B J. A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014, 1412(6).

# 模型训练方法与过程

## 优化器选择

让模型一步步学得更好

通过梯度下降不断调整模型权重参数，使损失函数最小化



[1] Robbins H, Monroe S. A stochastic approximation method[J]. The annals of mathematical statistics, 1951: 400-407.

[2] Polyak B T. Some methods of speeding up the convergence of iteration methods[J]. Ussr computational mathematics and mathematical physics, 1964, 4(5): 1-17.

[3] Tieleman T. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude[J]. COURSERA: Neural networks for machine learning, 2012, 4(2): 26.

[4] Adam K D P B J. A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014, 1412(6).

# 模型训练方法与过程

## 模型训练流程

```
for epoch in range(num_epochs):
    for batch in dataloader:
        # 1. 前向传播
        outputs = model(batch.inputs)
        loss = loss_function(outputs, batch.labels)

        # 2. 反向传播
        optimizer.zero_grad()
        loss.backward()

        # 3. 参数更新
        optimizer.step()
```

数据加载器

数据集取样



组织成批



动态预处理与增强



送入模型

39

# 模型训练方法与过程

## 模型训练流程

```
for epoch in range(num_epochs):
    for batch in dataloader:
        # 1. 前向传播
        outputs = model(batch.inputs)
        loss = loss_function(outputs, batch.labels)

        # 2. 反向传播 模型架构选取和参数初始化
        optimizer.zero_grad()
        loss.backward()

        # 3. 参数更新
        optimizer.step()
```

```
# 模型创建
model = MyModel(input_dim, hidden_dim, output_dim)

# 参数初始化
model.apply(init_weights)

def init_weights(m):
    if isinstance(m, nn.Linear):
        nn.init.xavier_uniform_(m.weight)
        nn.init.zeros_(m.bias)
```

# 模型训练方法与过程

## 模型训练流程

```
for epoch in range(num_epochs):
    for batch in dataloader:
        # 1. 前向传播
        outputs = model(batch.inputs)
        loss = loss_function(outputs, batch.labels)  # 模型预测值 真实标签

        # 2. 反向传播
        optimizer.zero_grad()
        loss.backward()

        # 3. 参数更新
        optimizer.step()
```

根据不同任务进行合理选择

# 均方误差 (MSE)  
loss\_function = nn.MSELoss()

# 平均绝对误差 (MAE)  
loss\_function = nn.L1Loss() # MAE的另一种叫法

# 二分类交叉熵  
loss\_function = nn.BCEWithLogitsLoss()

# 多分类交叉熵  
loss\_function = nn.CrossEntropyLoss()

# 模型训练方法与过程

## 模型训练流程

```
for epoch in range(num_epochs):
    for batch in dataloader:
        # 1. 前向传播
        outputs = model(batch.inputs)
        loss = loss_function(outputs, batch.labels)

        # 2. 反向传播
        optimizer.zero_grad()
        loss.backward()

        # 3. 参数更新
        optimizer.step()
```

优化器选择  
反向传播  
梯度更新

```
# SGD (随机梯度下降)
optimizer = optim.SGD(model.parameters(), lr=0.01)

# SGD + Momentum (带动量的随机梯度下降)
optimizer = optim.SGD(model.parameters(), lr=0.01,
momentum=0.9)

# RMSProp
optimizer = optim.RMSprop(model.parameters(), lr=0.001,
alpha=0.9)

# Adam
optimizer = optim.Adam(model.parameters(), lr=0.001,)
```

# 模型训练方法与过程

## 模型训练流程

```
for epoch in range(num_epochs):
    for batch in dataloader:
        # 1. 前向传播
        outputs = model(batch.inputs)
        loss = loss_function(outputs, batch.labels)

        # 2. 反向传播
        optimizer.zero_grad()
        loss.backward()

        # 3. 参数更新
        optimizer.step()
```

训练num\_epochs之后，loss收敛，模型训练完成

# 模型训练与优化



44

# 模型性能评估

## 评价指标



模型训练完成后，我们如何判断模型好坏呢？

45

# 模型性能评估

## 评价指标



模型训练完成后，我们如何判断模型好坏呢？

衡量模型在实际预测任务上的能力 → **评价指标**

预测是否准确？

错误是否严重？



46

# 模型性能评估

## 回归模型评价指标

回归任务输出是连续值，其误差loss衡量的是模型预测值与真实值之间的“距离”

因此，**损失函数本身即可作为评价指标**

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

上节课提到的常见损失函数

# 模型性能评估

## 回归模型评价指标

回归任务输出是连续值，其误差loss衡量的是模型预测值与真实值之间的“[距离](#)”

因此，**损失函数本身即可作为评价指标**

平均绝对误差

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad \downarrow$$

均方误差

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad \downarrow$$

上节课提到的常见损失函数

# 模型性能评估

## 回归模型评价指标

回归任务输出是连续值，其误差loss衡量的是模型预测值与真实值之间的“距离”

因此，**损失函数本身即可作为评价指标**

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$
$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

上节课提到的常见损失函数

均方根误差

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2}$$

平均绝对百分比误

$$MAPE = \frac{100\%}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

决定系数

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

可选

那对于分类模型能直接用loss进行评估吗？

# 模型性能评估

## 分类模型评价指标

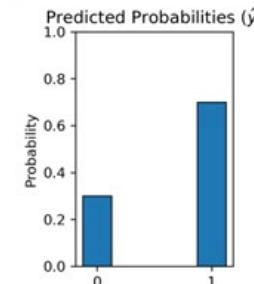
分类任务中，交叉熵损失loss衡量的是模型预测的概率分布与数据真实分布之间的距离

$$L = -\frac{1}{m} \sum_{j=1}^m \left[ y^{(j)} \log(\hat{y}^{(j)}) + (1 - y^{(j)}) \log(1 - \hat{y}^{(j)}) \right]$$

交叉熵损失 (Cross-Entropy Loss) – 二分类

真实标签, 0或1

预测0或1的概率分布



# 模型性能评估

## 分类模型评价指标

分类任务中，交叉熵损失loss衡量的是模型预测的概率分布与数据真实分布之间的距离。

$$L = -\frac{1}{m} \sum_{j=1}^m \left[ y^{(j)} \log(\hat{y}^{(j)}) + (1 - y^{(j)}) \log(1 - \hat{y}^{(j)}) \right]$$

交叉熵损失 (Cross-Entropy Loss) – 二分类

交叉熵损失函数衡量模型“有多确定自己是对的”，而不是“是否对”



针对分类任务，衡量模型结果好不好，有什么指标呢？

不仅看你预测对不对，还看你有多自信地预测成那个类别

- 模型A输出：猫 0.9, 狗 0.1 ✓
- 模型B输出：猫 0.51, 狗 0.49 ✓

两个模型都预测对了，但交叉熵会认为：  
模型A更好，因为它更“确定”自己是对的。

只关心最终预测的类别是不是正确

猫图预测成猫 ✓  
预测成狗 ✗

只看最大概率类别

51

# 模型性能评估

## 分类模型评价指标

分类模型通常使用以下四个评价指标：

- 准确率 (Accuracy)
- 精确率 (Precision)
- 召回率 (Recall)
- F1分数 F1-Score



如何计算？

# 模型性能评估

## 分类模型评价指标

### 混淆矩阵（用于总结分类预测结果）

四个核心量：

- **True Positive/TP 真正类**

样本的真实类别是`<猫>`，并且模型预测的结果也是`<猫>`。

- **False Positive/FP 假负类**

样本的真实类别是`<猫>`，但是模型将其识别为`<不是猫>`。

- **True Negative/TN 假正类 - 误报**

样本的真实类别是`<不是猫>`，但是模型将其识别为`<猫>`。

- **False Negative/FN 真负类 - 漏报**

样本的真实类别是`<不是猫>`，并且模型将其识别为`<不是猫>`。

混淆矩阵		真实值	
		Positive	Negative
预测值	Positive	TP	FP
	Negative	FN	TN

图像二分类任务上：

混淆矩阵		真实值	
		猫	不是猫
预测值	猫	10	3
	不是猫	8	45

## 填空题 2分

### 分类模型评价指标

#### ➤ 准确率 (Accuracy)

- 公式: Accuracy = 正确分类数 / 样本总数 =  $(TP+TN)/(TP+FN+FP+TN)$
- 指所有分类 (无论是正类还是负类) 正确分类的比例, 理想场景1.0

#### ➤ 精确率 (Precision)

- 公式: Precision =  $TP/(TP+FP)$
- 在模型识别为正类的样本中, 真正为正类的样本所占的比例。  
精确率越高, 模型对负样本的区分能力越强

混淆矩阵		真实值	
		Positive	Negative
预测值	Positive	TP	FP
	Negative	FN	TN

混淆矩阵		真实值	
		猫	不是猫
预测值	猫	10	3
	不是猫	8	45

准确率= [填空1] (示例: m/n)

精确率= [填空2]

# 模型性能评估

## 分类模型评价指标

### ➤ 准确率 (Accuracy)

- 公式: Accuracy = 正确分类数 / 样本总数 =  $(TP+TN)/(TP+FN+FP+TN)$
- 指所有分类 (无论是正类还是负类) 正确分类的比例, 理想场景1.0

### ➤ 精确率 (Precision)

- 公式: Precision =  $TP/(TP+FP)$
- 在模型识别为正类的样本中, 真正为正类的样本所占的比例。  
精确率越高, 模型对负样本的区分能力越强

混淆矩阵		真实值	
		Positive	Negative
预测值	Positive	TP	FP
	Negative	FN	TN

混淆矩阵		真实值	
		猫	不是猫
预测值	猫	10	3
	不是猫	8	45

准确率=5/6

精确率=10/13

55

## 填空题 2分

### 分类模型评价指标

#### ➤ 召回率 (Recall)

- 公式:  $Recall = TP / (TP + FN)$
- 模型正确认别出为正类的样本的数量占总的正类样本数量的比值。

召回率越高，模型对正样本的识别能力越强

#### ➤ F1分数 F1-Score

- 公式:  $F_1 = 2 * (\text{精确率} * \text{召回率}) / (\text{精确率} + \text{召回率})$
- 精确率和召回率的调和平均数。

F1 score越高，说明模型越稳健

混淆矩阵		真实值	
预测值	Positive	Positive	Negative
	Positive	TP	FP
预测值	Negative	FN	TN

混淆矩阵		真实值	
预测值	猫	猫	不是猫
	猫	10	3
预测值	不是猫	8	45

召回率= [填空1]

F1 分数= [填空2]

# 模型性能评估

## 分类模型评价指标

### ➤ 召回率 (Recall)

- 公式:  $Recall = TP / (TP + FN)$
- 模型正确认别出为正类的样本的数量占总的正类样本数量的比值。

召回率越高，模型对正样本的识别能力越强

### ➤ F1分数 F1-Score

- 公式:  $F_1 = 2 * (\text{精确率} * \text{召回率}) / (\text{精确率} + \text{召回率})$
- 精确率和召回率的调和平均数。

F1 score越高，说明模型越稳健

混淆矩阵		真实值	
预测值	Positive	Positive	Negative
	Positive	TP	FP
预测值	Negative	FN	TN

混淆矩阵		真实值	
预测值	猫	猫	不是猫
	猫	10	3
预测值	不是猫	8	45

召回率=5/9

F1 分数=20/31

课后思考:

多分类任务中的macro-F1和  
micro-F1的计算

57

# 模型性能评估

## 模型评估效果不理想

确定评估指标后，即可衡量模型的预测能力

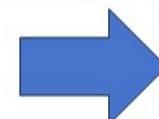
表现

模型在测试集  
上的评估效果  
不理想/很差



模型在训练过程  
中出现问题

怎么办?



看loss曲线

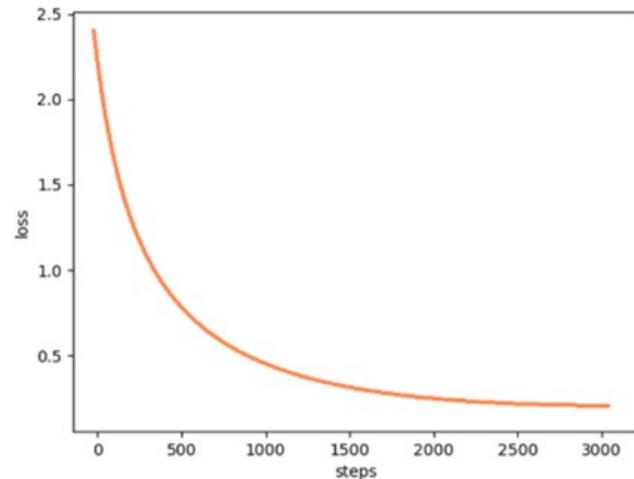


58

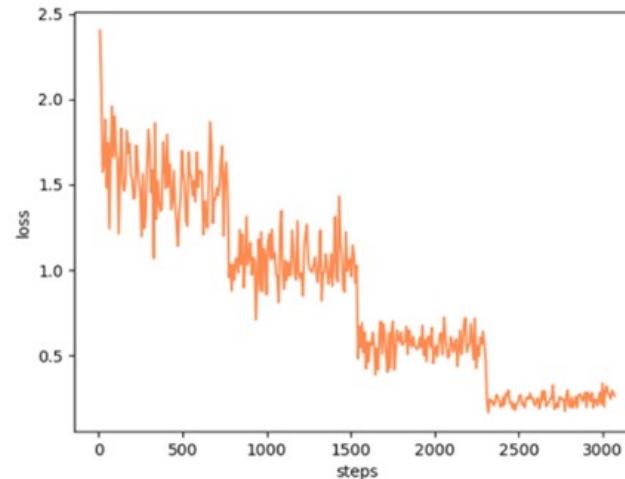
# 模型性能评估

## Train-loss分析

我们通常先观察模型在**训练集**的表现来判断是否正常学习



平滑下降



阶梯下降

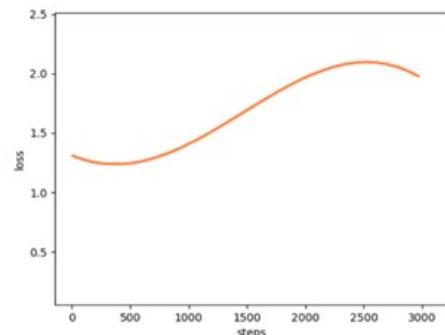
**正常的训练曲线**

59

# 模型性能评估

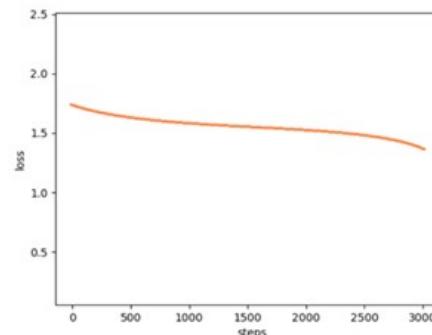
## Train-loss分析

我们通常先观察模型在训练集的表现来判断是否正常学习



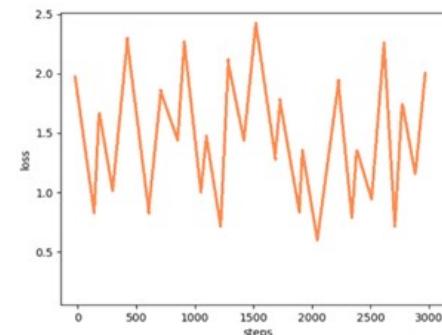
上升

数据质量差，或者学习率设置得过大，使模型在最优解附近震荡，甚至跳过最优解，导致无法收敛



平缓，保持高位

目标任务的难度较大，或者模型的学习率设置得过小，导致收敛速度太慢，无法达到最优解



异常抖动

训练数据质量差（存在噪声、分布不均衡）导致训练过程不稳定

## 异常的训练曲线

60

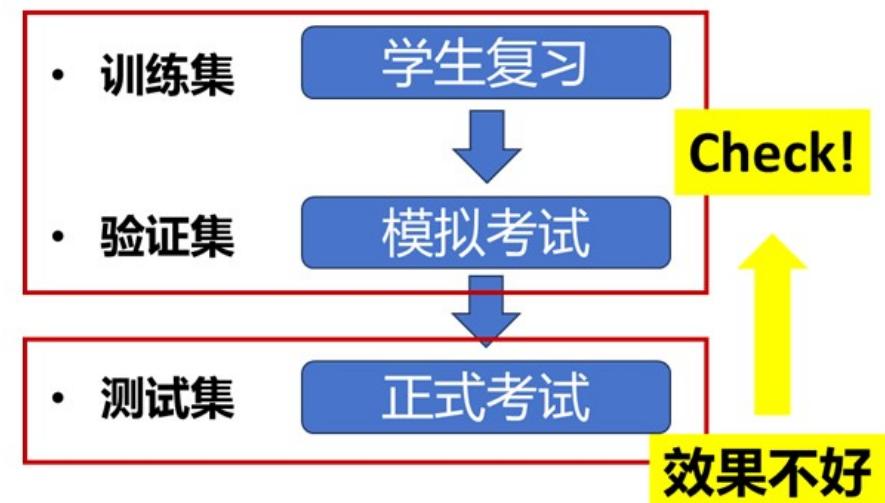
# 模型性能评估

## 引入valid-loss

仅关注训练集的表现难以发现**模型的泛化问题**，因此需要同时关注模型在验证集的表现；通过比较**train-loss**与 **valid-loss**的变化趋势，可以更全面地评估模型是否真正学到了可推广的规律

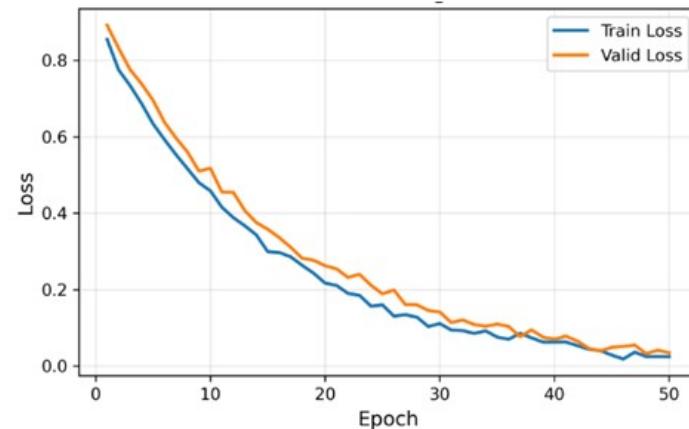
Train-loss下降→模型有效的学习到训练数据

Valid-loss下降→模型的泛化能力较强



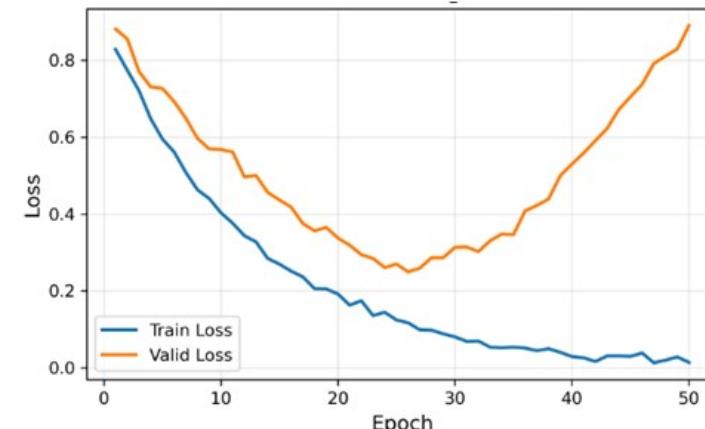
# 模型性能评估

## Train-loss & Valid-loss分析



① train和valid loss均正常下降

理想状态，表明模型有效的学习到训练数据，并有较强的泛化能力

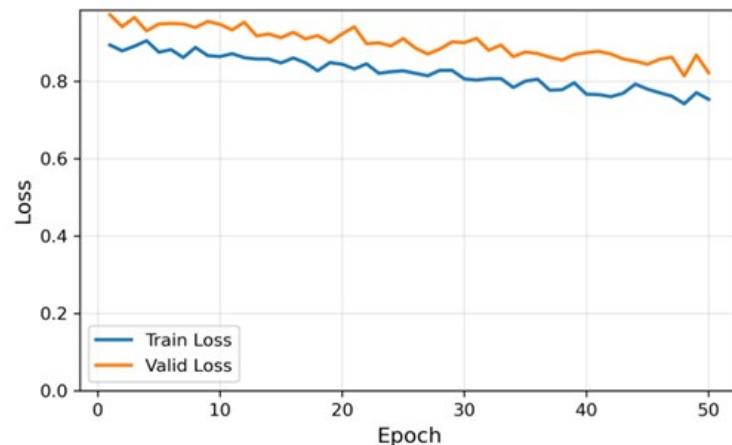


② train下降，valid先下降后上升

训练数据量有限，或过度训练，模型“把训练数据背下来了”，而不是“学会了规律”

# 模型性能评估

## Train-loss & Valid-loss分析



③ train和valid loss均居高不下/下降缓慢

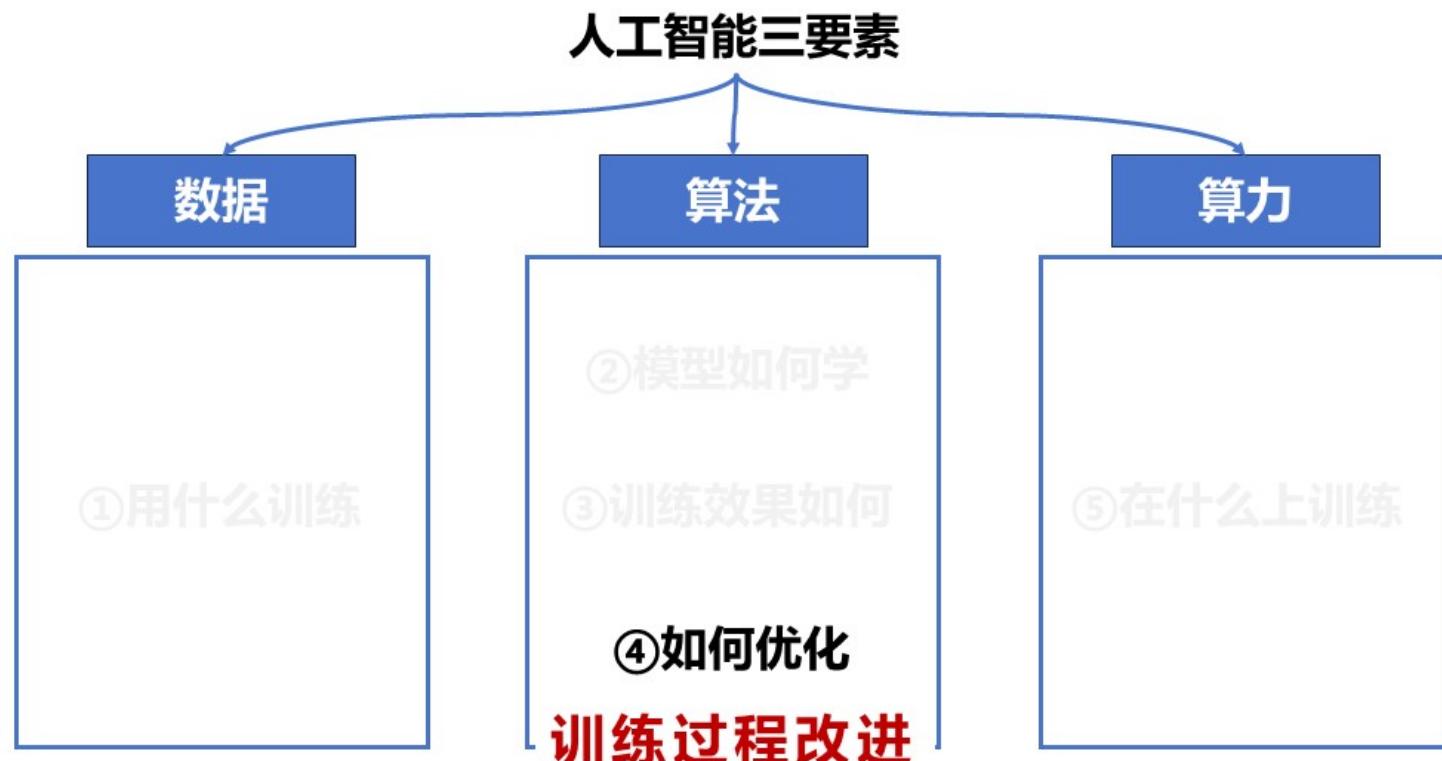
模型还未收敛，可能存在模型能力有限，  
或者学习率过低、训练轮数不够等，模  
型“脑子不够”或者“学得太慢”

讨论思考其他情况：

- ④ train下降, valid先下降后持平
  - ⑤ train持平, valid下降
  - ⑥ train下降, valid持平
- .....

63

# 模型训练与优化



64

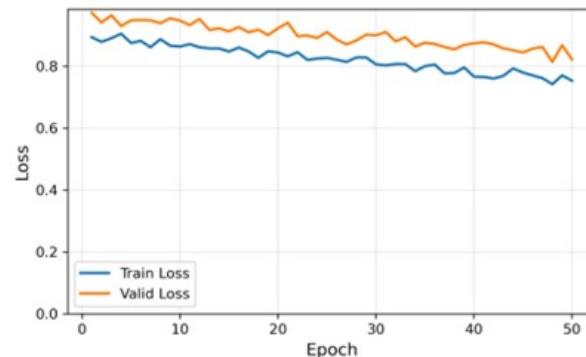
# 训练过程改进

## 欠拟合、过拟合分析

### ➤ 欠拟合问题

模型过于简单，无法捕捉训练数据中的关键特征或复杂关系

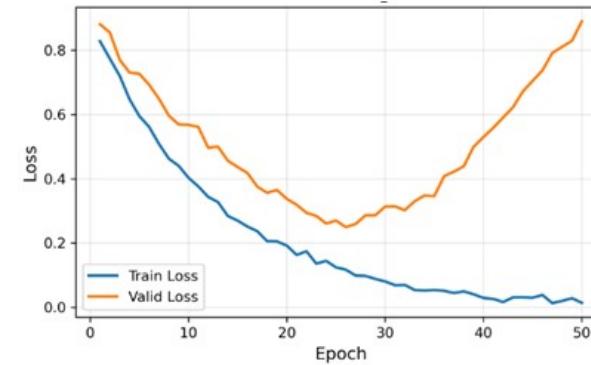
表现：训练集和测试集上都很差



### ➤ 过拟合问题

模型过于复杂，记住了训练数据中的噪声或细节，导致泛化能力不足

表现：训练集上好，测试集上表现较差



65

# 训练过程改进

## 欠拟合问题

### 成因

- 模型容量小
- 训练轮数不足
- 学习率太小

66

# 训练过程改进

## 欠拟合问题

### 成因

- 模型容量小
- 训练轮数不足
- 学习率太小

### 解决方法

- 使用更复杂模型
- 增加训练轮数
- 调大学习率，改用更快优化器（如Adam）

67

# 训练过程改进

## 欠拟合问题

### 成因

- 模型容量小
- 训练轮数不足
- 学习率太小

### 解决方法

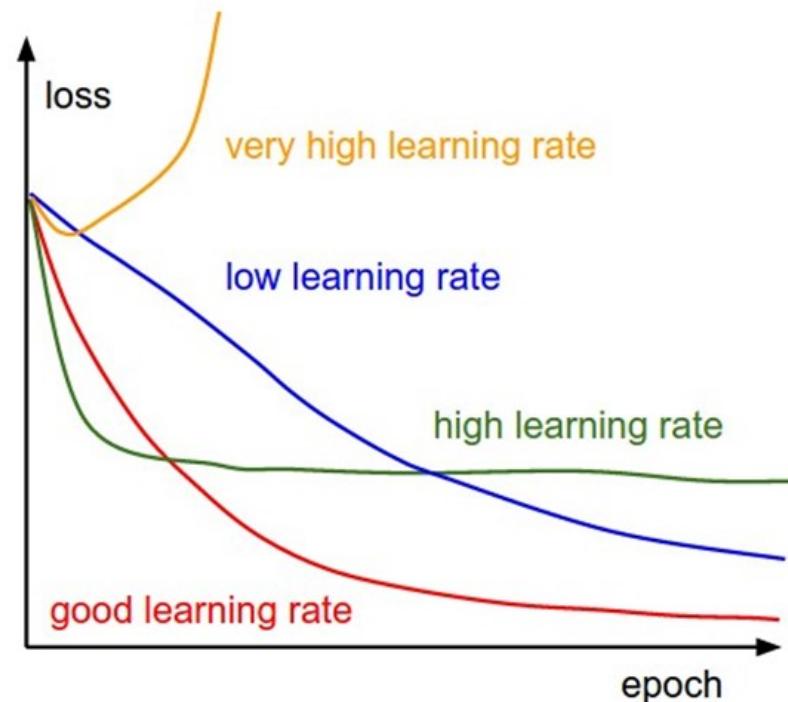
- 使用更复杂模型
- 增加训练轮数
- 调大学习率，改用更快优化器（如Adam）

比较直接

68

# 训练过程改进

## 欠拟合问题



### 解决方法

- 使用更复杂模型
- 增加训练轮数

比较直接

- 调大学习率，改用更快优化器（如Adam）

学习率问题比较复杂，  
简单的调整容易导致训练出现问题

69

# 训练过程改进

## 欠拟合问题 - 学习率

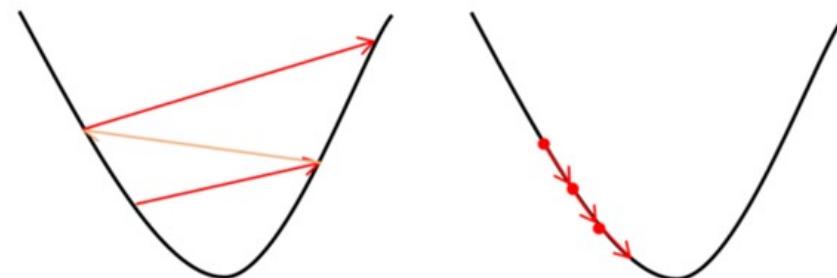
- 学习率控制每次更新的步长

Big learning rate (大学习率) Small learning rate (小学习率)

$$\theta := \theta - \alpha \nabla_{\theta} L(\theta)$$

学习率

- 一开始学习率过小 → 收敛慢 → 浪费大量时间
- 一开始学习率太大 → 来回震荡 → 无法收敛
- 后期学习率不变 → 在最低点附近跳动 → 达不到最优



通常我们会尝试性的将初始学习率设为: 0.1, 0.01, 0.001, 0.0001

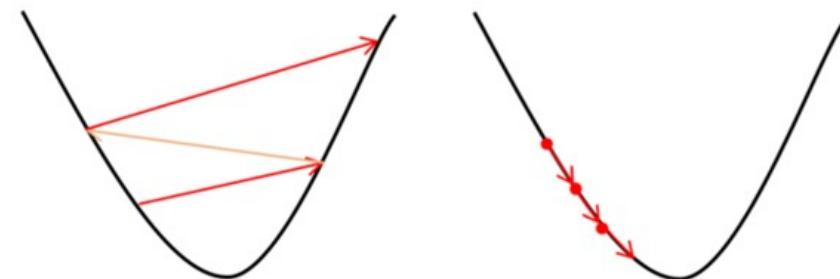
# 训练过程改进

## 欠拟合问题 - 学习率

### 实际需要的训练策略

- 前期：需要大步探索方向
- 后期：需要细步微调

Big learning rate (大学习率) Small learning rate (小学习率)



如何自适应调整学习率？

71

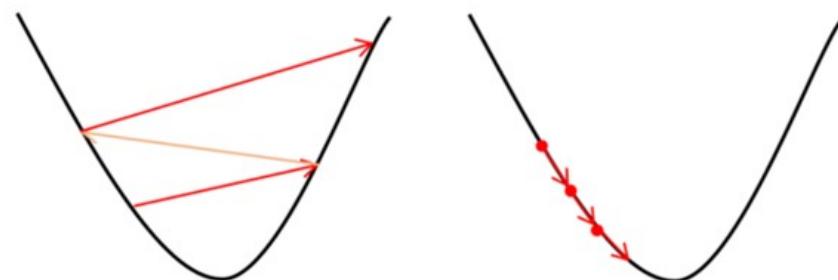
# 训练过程改进

## 欠拟合问题 - 学习率

### 实际需要的训练策略

- 前期：需要大步探索方向
- 后期：需要细步微调

Big learning rate (大学习率) Small learning rate (小学习率)



### 如何自适应调整学习率？

在模型训练阶段选择能自适应调整学习率的优化器，比如Adam、 RMSProp

# 训练过程改进

## 过拟合问题

### 成因

- 模型容量过大
- 训练轮数太多
- 数据太少

73

# 训练过程改进

## 过拟合问题

### 成因

- 模型容量过大
- 训练轮数太多
- 数据太少

### 解决方法

- 减少模型复杂度、正则化 (L1/L2)
- Dropout
- Early Stopping早停
- 数据增广

74

## 训练过程改进

### 过拟合问题 – L1/L2正则化

在损失函数中加入“权重（模型参数）大小”的惩罚项，约束模型复杂度

$$L_1 \text{ 正则化: } Total\_Loss = Loss + \lambda \sum |\omega_i|$$

$$L_2 \text{ 正则化: } Total\_Loss = Loss + \lambda \sum \omega_i^2$$

其中， $\omega_i$ : 模型的每一个权重； $\lambda$ : 正则化系数（控制惩罚强度）

# 训练过程改进

## 过拟合问题 – Dropout丢弃

在神经网络的**训练过程中**，对于一次迭代中的某一层神经网络，

先**随机选择中的一些神经元并将其临时隐藏(丢弃)**，然后再进行本次训练和优化。



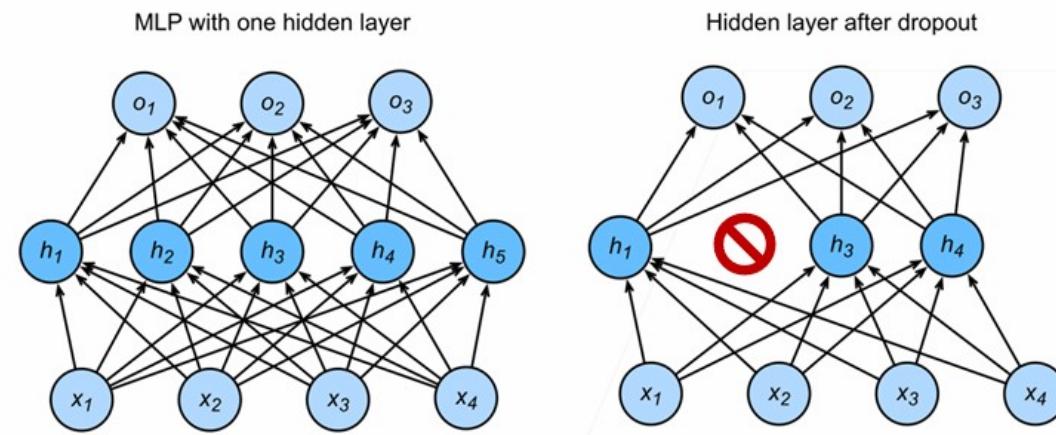
模型泛化性更强，因为它不会太依赖某些局部的特征

$$\mathbf{h} = \sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1)$$

$$\mathbf{h}' = \text{dropout}(\mathbf{h})$$

$$\mathbf{o} = \mathbf{W}_2 \mathbf{h}' + \mathbf{b}_2$$

$$\mathbf{y} = \text{softmax}(\mathbf{o})$$



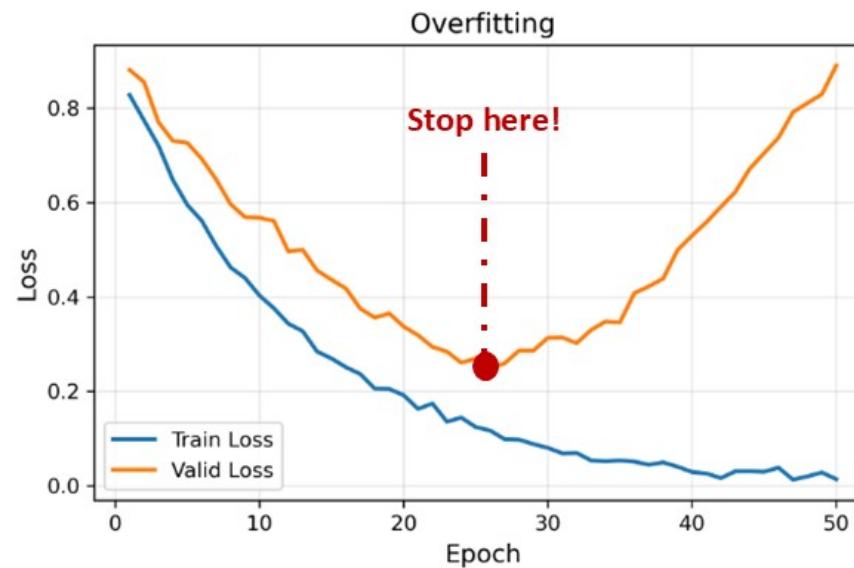
76

# 训练过程改进

## 过拟合问题 – Early Stopping早停

Valid loss不再下降 → 停止训练，防止过拟合

一般设置“**patience**”参数，**连续N轮valid loss无提升则停止**



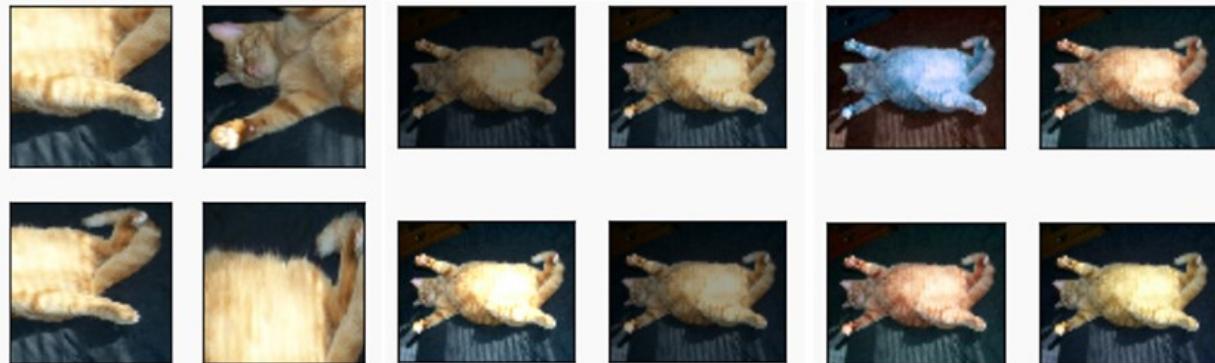
77

# 训练过程改进

## 过拟合问题 - 数据增广

丰富数据集**多样性**，针对图像数据集：

- 在语言里面加入各种不同的**背景噪音**
- 改变图片的**颜色和形状**



78

# 训练过程改进

## 过拟合问题 - 数据增广

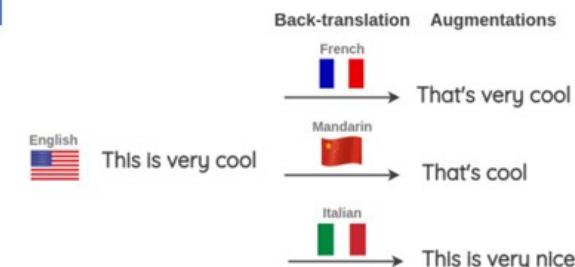
丰富数据集**多样性**, 针对文本数据集:

- 基于规则的增广: 同义词替换、随机插入/删除/交换

The movie is great → The film is fantastic

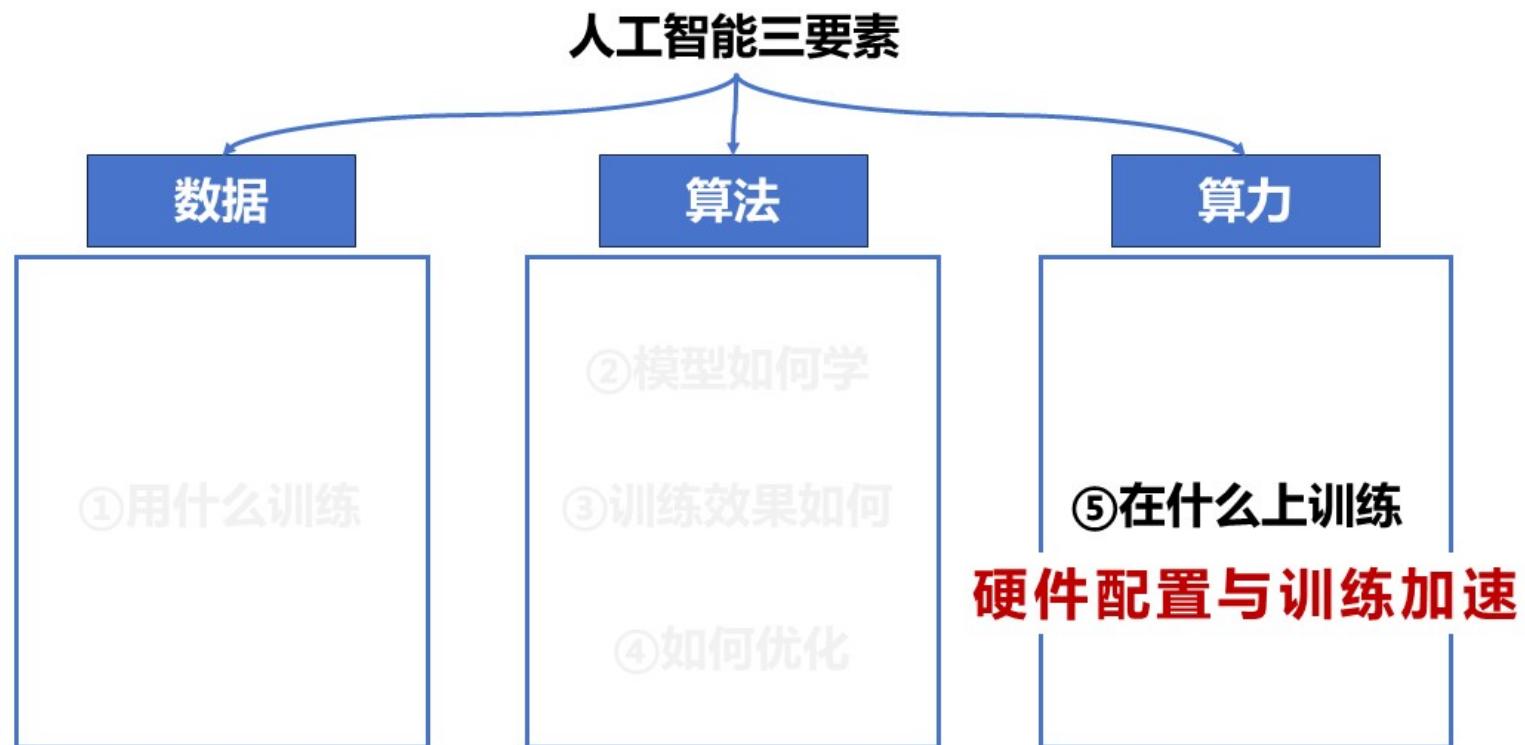


- 回译: 将文本翻译成另外一种语言, 然后再翻译回来



- 生成式改写: 利用T5、ChatGPT、GPT4.....生成不同表述

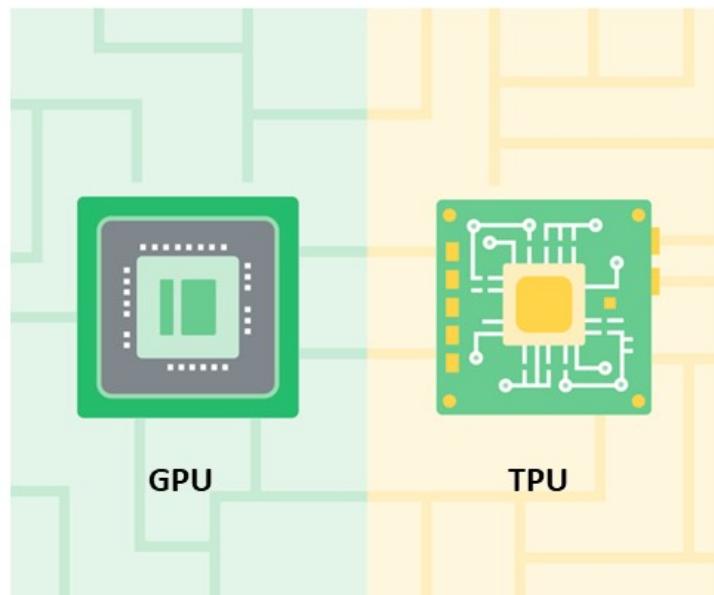
# 模型训练与优化



80

# 硬件配置与训练加速

## 主流硬件—本地部署



## 常用云服务平台

kaggle

colab

AutoDL

GPU类型/ 资源	Tesla T4 / P100 / V100 (自动分配)	T4 / P100 / A100 (Pro 系列)	RTX 4090 / A100 / H100 (可选)
价格与限制	免费 每次最长 9 小时	基础版免费 Pro \$9.99/月	按小时计费 如 4090 ¥ 1.98 / h
优点	完全免费、适合新手	简单易用，与 Google Drive 结合，持久存储	高显存、高带宽、支持 SSH/Jupyter/多 GPU 并行
适合场景	课程作业、Kaggle 竞赛	深度学习实验、课程演示	科研项目、模型训练

81

## 课程第一次作业

### 房价预测模型实践

82

# 课程第一次作业

## 房价预测模型实践

**任务目标** 基于Kaggle房价预测数据集，完成从数据预处理 → 模型训练 → 提交结果的完整流程

**数据来源** <https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/overview>

**提交要求**

- 代码文件 (.ipynb 或 .py)
- 实验报告 (包含思路、kaggle提交截图、收获与反思)

83

# 课程第一次作业

## 数据准备与预处理

### 数据集

- train.csv – 训练集，包含特征和标签
- test.csv – 测试集，**只包含特征**
- data\_description.txt – 数据特征的完整描述
- sample\_submission.csv – 结果提交文件样例

### 特征预处理

- 缺失值处理、类别变量处理（one-hot编码）、数值归一化（标准化）

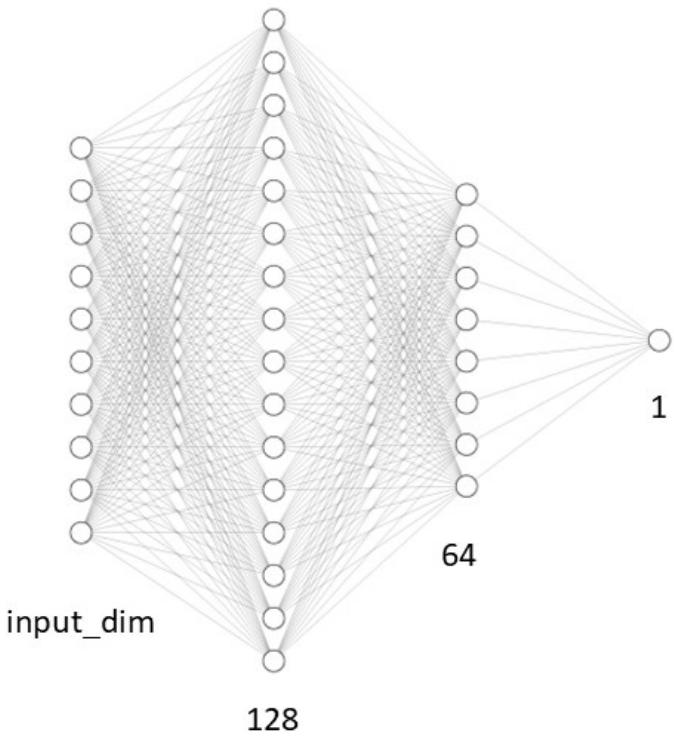
84

# 课程第一次作业

## 模型构建

### 示例——全连接神经网络

- 输入层：经过处理的 `input_dim` 个特征
- 隐藏层：
  - 第一层：128个神经元 + ReLU激活 + Dropout
  - 第二层：64个神经元 + ReLU激活 + Dropout
- 输出层：**1个实数**，表示预测的房价



85

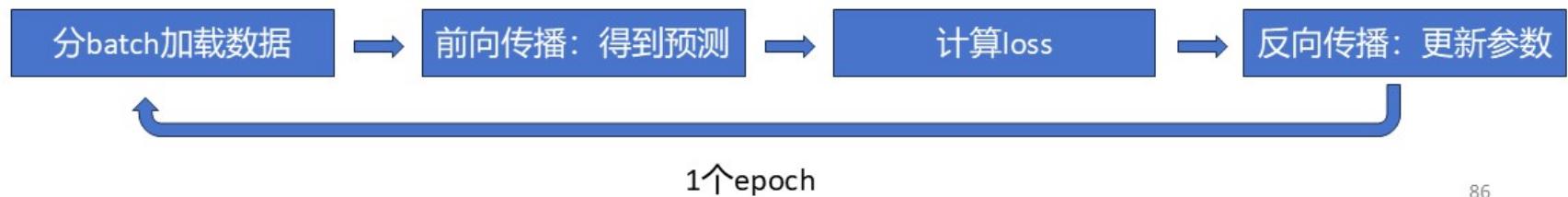
# 课程第一次作业

## 模型训练

### 示例——超参数设置

TEST_SIZE	0.2
BATCH_SIZE	32
EPOCHS	10
LEARNING_RATE	0.001
DROPOUT_RATE	0.1

### 构建训练循环



86

# 课程第一次作业

## 后续步骤

### 调参与过拟合控制

- 调整学习率、batch size、模型结构
- 加入L2正则化、EarlyStopping

### 模型保存与使用

- 训练完成后**保存**模型参数
- 加载模型并使用test.csv进行**预测**
- 获得预测结果，保存为submission.csv
- 到kaggle上**提交**，获取分数和排名

87

# 人工智能导论

Introduction to Artificial Intelligence

谢谢！



88