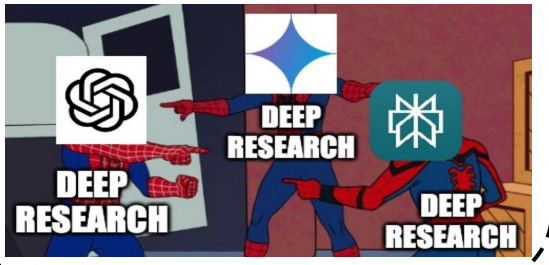
一周AI大事

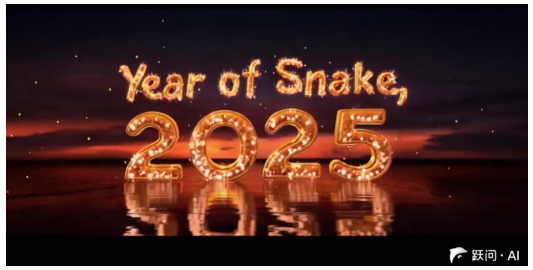
自Google和OpenAI后，Perplexity推出Deep Research



微信、知乎等平台接入Deepseek R1大模型

知乎直答 R1

阶跃星辰发布全球最大开源视频模型Step-Video-T2V



马斯克xAI的Grok 3亮相,lmarena首个突破1400分的模型，出道即巅峰

Elon Musk @elonmusk 订阅

Grok 3 release with live demo on Monday night at 8pm PT.

Smartest AI on Earth.

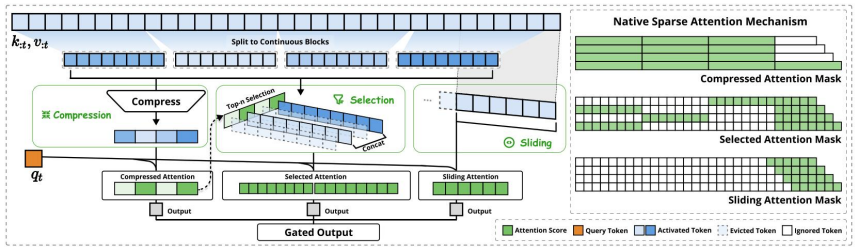
lmarena.ai

xAI Grok-3 #1 in Arena

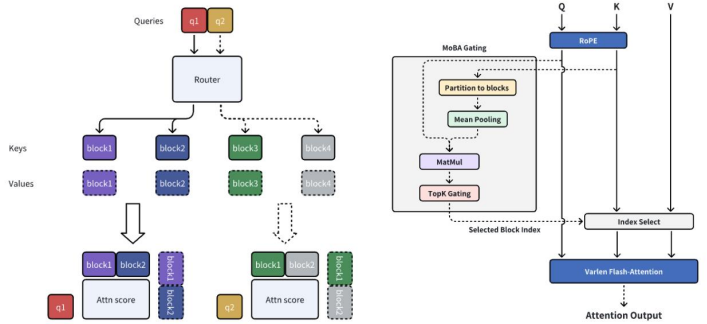
First model with >1400 score

Rank* (UB)	Model	Arena Score	95% CI	Votes	Organization
1	chocolate (Early Grok-3)	1402	+7/-6	7829	xAI
2	Gemini-2.0-Flash-Thinking-Exp-01-21	1385	+5/-5	13336	Google
2	Gemini-2.0-Pro-Exp-02-05	1379	+5/-6	11197	Google
2	ChatGPT-4o-latest (2025-01-29)	1377	+5/-6	10529	OpenAI
5	DeepSeek-R1	1361	+8/-7	5079	DeepSeek
5	Gemini-2.0-Flash-001	1356	+6/-5	9092	Google
5	o1-2024-12-17	1353	+6/-5	15437	OpenAI
8	o1-preview	1335	+4/-4	33169	OpenAI

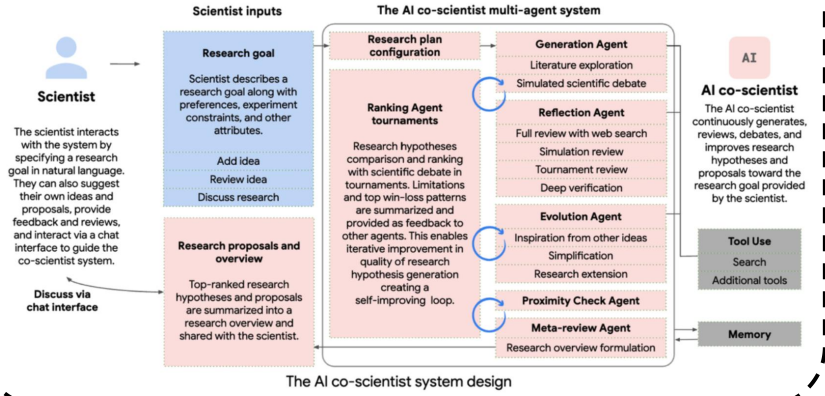
Deepseek提出新型注意力机制NSA，梁文峰亲自提交arxiv预印本



Kimi提出MoBA的新型注意力机制，能将处理1M长文本的速度一下子提升6.5倍



Google AI co-scientists利用多智能体系统，携手人类科学家加速科学创新研究



2025-02-15 ~ 2025.02.20

机器学习基础

CS2916 大语言模型

饮水思源 爱国荣校

https://plms.ai/teaching/index.html

该章节内容主要参参照神经网络与深度学习一书

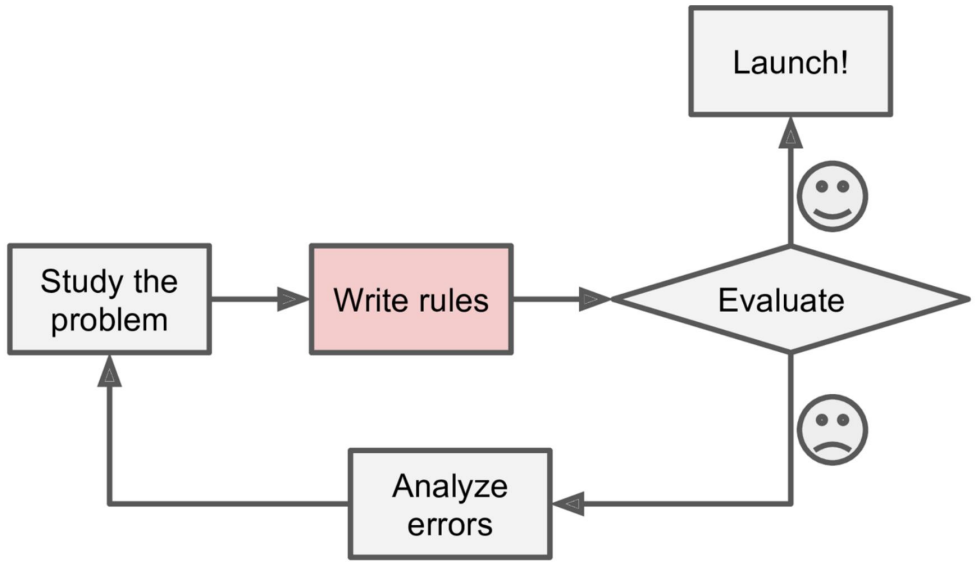
如何设计一个邮件过滤系统?

基于规则

- 包含“五折优惠”
- 包含“免费访问”,”交易”

垃圾邮件

主题：**五折优惠**！**免费访问**我们的**交易**平台，享受限时优惠！
亲爱的，
我们很高兴地通知您，我们的交易平台现正进行五折优惠活动！现在，您可以免费访问我们的平台，并享受各种交易的优惠。
这是一个难得的机会，您可以以更低的价格购买您所需的商品，或者通过我们的平台卖出您不需要的物品。我们的平台提供了安全、快捷、方便的交易体验，让您可以轻松地进行买卖操作。
请注意，这是一个限时的优惠活动，机会难得，不要错过！现在就点击下面的链接，免费访问我们的交易平台，并开始享受您的优惠吧！
如果您有任何疑问或需要帮助，请随时联系我们的客服团队。我们期待与您的合作，并为您提供最好的服务。
谢谢！



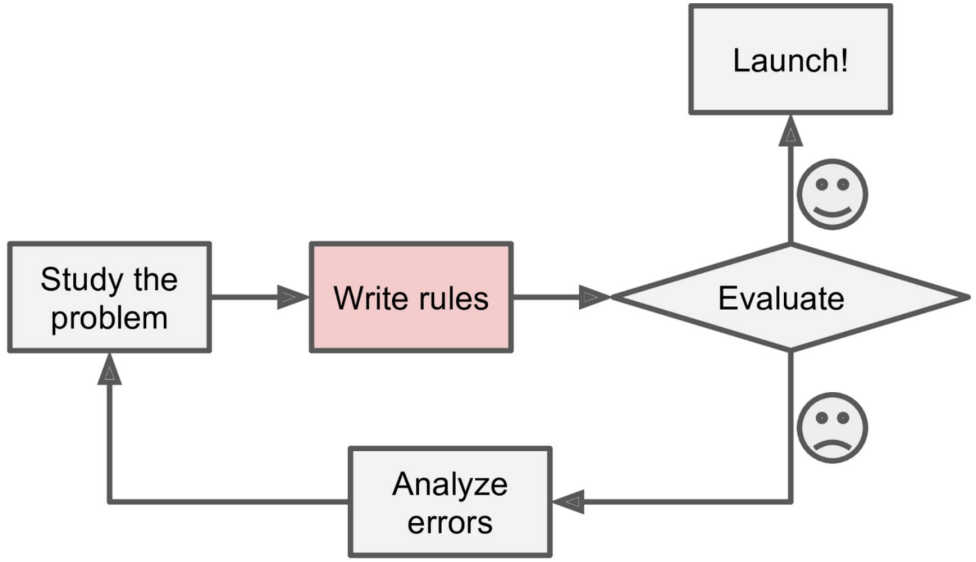
如何设计一个邮件过滤系统?

基于规则

- 包含“五折优惠”
- 包含“免费访问”,”交易”

垃圾邮件

亲爱的，
我们很高兴地通知您，我们正在进行一项特别活动！现在，您可以访问我们的平台，并享受各种优惠。
这是一个难得的机会，您可以以更低的价格购买您所需的商品，或者卖出您不需要的物品。我们的平台提供了安全、快捷、方便的体验，让您可以轻松地进行买卖操作。
请注意，这是一个限时活动，机会难得，不要错过！现在就点击下面的链接，访问我们的平台，并开始享受您的优惠吧！
如果您有任何疑问或需要帮助，请随时联系我们的客服团队。我们期待与您的合作，并为您提供最好的服务。
谢谢！



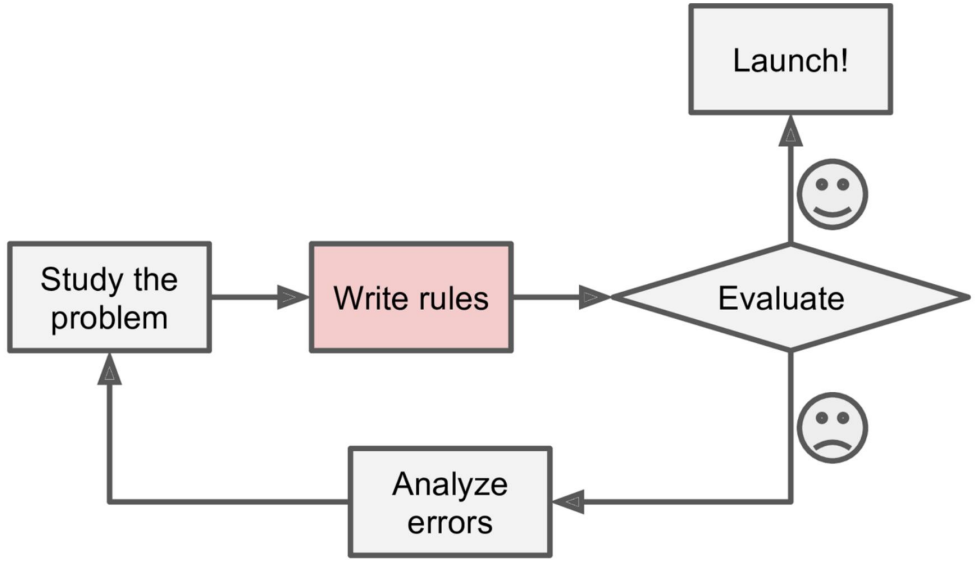
如何设计一个邮件过滤系统?

基于规则

- 包含“五折优惠”
- 包含“免费访问”,”交易”

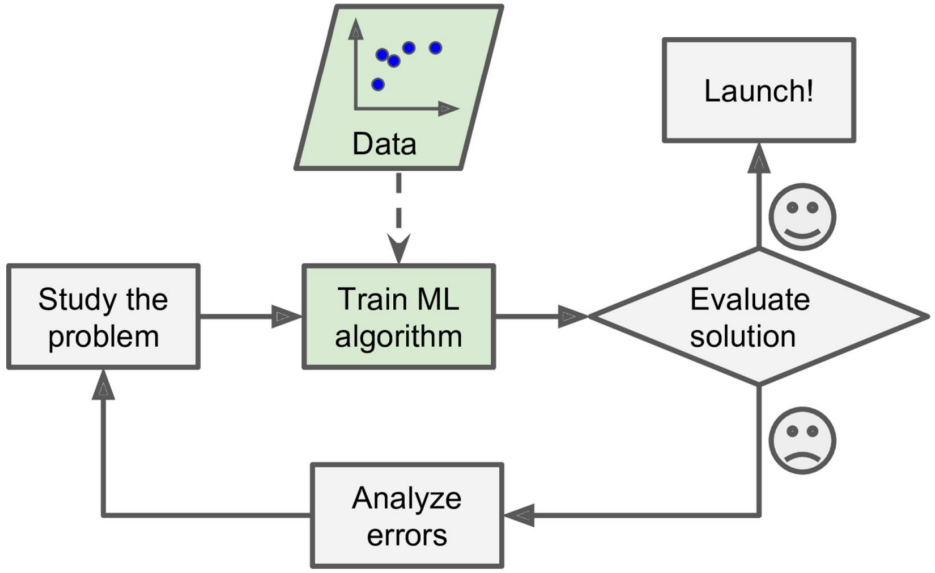
正常邮件

主题：关于校园内新服务的**免费访问**与**交易**信息
亲爱的师生们，
我们很高兴地通知大家，学校最近引进了一项新的服务，现在开始向全校师生提供免费访问。这项服务旨在丰富我们的教育资源，帮助大家更有效地学习和教学。
同时，我们也将开展一系列与此服务相关的交易活动，包括资源共享、教材交换等。我们鼓励大家积极参与，共同打造更加丰富多彩的校园文化。
请大家通过以下链接了解更多详情并开始享受这项免费服务：[链接]
感谢大家的关注与支持!
祝学习进步，



如何设计一个邮件过滤系统?

- **基于规则**
  - 包含“五折优惠”
  - 包含“免费访问”,”交易”
- **机器学习**
  - 收集标注为“垃圾”和“正常”的邮件
  - 设计可以描述一组邮件的特征
  - 学习一个分类器
    - f(“邮件”) = “垃圾邮件” or “正常邮件”

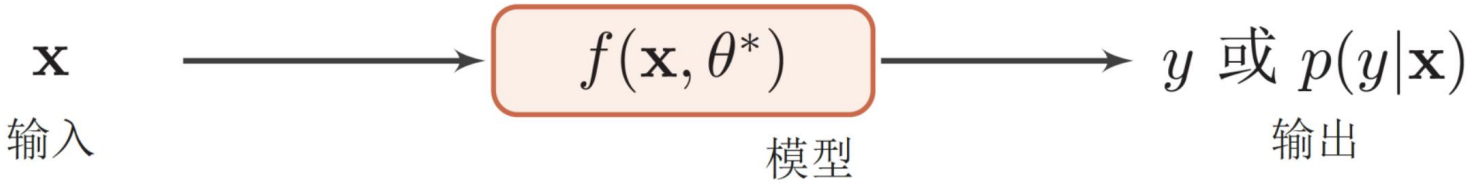


机器学习≈构建一个映射函数

- **邮件过滤**
  - f（“尊敬的...交易..”）= “垃圾邮件”
- **图片识别**
  - f（  ）= “猫”
- **语音识别**
  - f（  ）= “你好”

机器学习≈构建一个映射函数

- **邮件过滤**
  - f（“尊敬的...交易..”）= “垃圾邮件”
- **图片识别**
  - f（  ）= “猫”
- **语音识别**
  - f（  ）= “你好”



机器学习三要素

- **模型**：模型是对现实世界数据关系的数学表达式
  - 线性方法：$f(\mathbf{x}, \theta) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$
- **学习准则**：损失函数或目标函数，计算模型预测值与实际值之间的差异
  - 期望风险 $\mathcal{R}(f) = \mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[\mathcal{L}(f(\mathbf{x}), y)]$
- **优化方法**：用于调整模型参数，以最小化学习准则
  - 梯度下降

常见机器学习类型

	监督学习	无监督学习	强化学习
训练样本	训练集 $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$	训练集 $\{\mathbf{x}^{n}\}_{n=1}^{N}$	智能体和环境交互的轨迹 τ 和累积奖励 G_τ
优化目标	$y = f(\mathbf{x})$ 或 $p(y\|\mathbf{x})$	$p(\mathbf{x})$ 或带隐变量 $\mathbf{z}$ 的 $p(\mathbf{x}\|\mathbf{z})$	期望总回报 $\mathbb{E}_\tau[G_\tau]$
学习准则	期望风险最小化 最大似然估计	最大似然估计 最小重构错误	策略评估 策略改进

常见机器学习类型

半监督学习：利用少量的有标签数据和大量的无标签数据来提高学习精度

	监督学习	无监督学习	强化学习
训练样本	训练集 $\{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$	训练集 $\{\mathbf{x}^{n}\}_{n=1}^{N}$	智能体和环境交互的轨迹 τ 和累积奖励 G_τ
优化目标	$y = f(\mathbf{x})$ 或 $p(y\|\mathbf{x})$	$p(\mathbf{x})$ 或带隐变量 $\mathbf{z}$ 的 $p(\mathbf{x}\|\mathbf{z})$	期望总回报 $\mathbb{E}_\tau[G_\tau]$
学习准则	期望风险最小化 最大似然估计	最大似然估计 最小重构错误	策略评估 策略改进

自监督训练：从未标记的数据中自动生成标签或目标，然后使用这些生成的标签来训练模型

参数学习

- **期望风险未知，通过经验风险近似** $\mathcal{R}(f) = \mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[\mathcal{L}(f(\mathbf{x}), y)]$
  - **训练数据：** $\mathcal{D} = \{\mathrm{x}^{(i)}, y^{(i)}\}, i \in [1, N]$

$$\mathcal{R}_{\mathcal{D}}^{emp}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(y^{(n)}, f(x^{(n)}, \theta))$$

- **经验风险最小化**
  - 寻找参数，使得经验风险最小化

$$\theta^* = \arg\min_{\theta} \mathcal{R}_{\mathcal{D}}^{emp}(\theta)$$

- **机器学习问题转化成为一个最优化问题**

优化：梯度下降法

梯度下降(Gradient Decent, GD)

$$w \leftarrow w - \eta \nabla_w \underbrace{\sum_{i=1}^{n} L(x^{(i)}, y^{(i)}, f_w)}_{\text{training loss}}$$

优化：梯度下降法

梯度下降(Gradient Decent, GD)

$$w \leftarrow w - \eta \nabla_w \underbrace{\sum_{i=1}^{n} L(x^{(i)}, y^{(i)}, f_w)}_{\text{training loss}}$$

随机梯度下降 (Stochastic GD)

For each $(x, y) \in D_{\text{train}}$:

$$w \leftarrow w - \eta \nabla_w \underbrace{L(x, y, f_w)}_{\text{example loss}}$$

输入: 训练集 $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}, n = 1, \cdots, N$，验证集 $\mathcal{V}$，学习率 α

随机初始化 θ;

repeat

  对训练集 $\mathcal{D}$ 中的样本随机重排序;

  for $n = 1 \cdots N$ **do**

    从训练集 $\mathcal{D}$ 中选取样本 $(\mathbf{x}^{(n)}, y^{(n)})$;

    // 更新参数

    $\theta \leftarrow \theta - \alpha \dfrac{\partial \mathcal{L}\left(\theta; x^{(n)}, y^{(n)}\right)}{\partial \theta}$;

  end

until 模型 $f(\mathbf{x}, \theta)$ 在验证集 $\mathcal{V}$ 上的错误率不再下降;

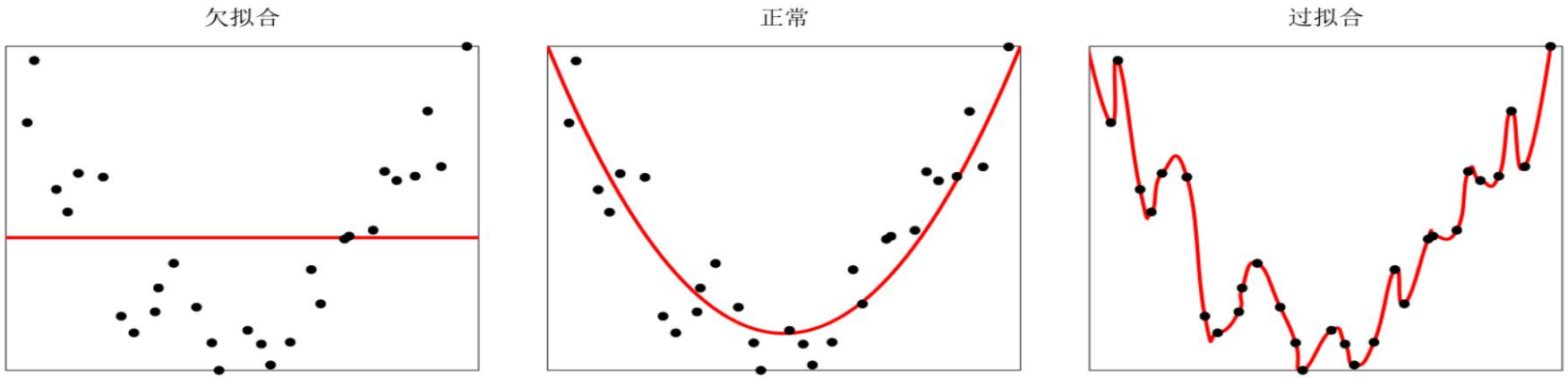
输出: θ

一般有训练、验证、测试集合

优化：GD vs SGD

特性/算法	梯度下降 (GD)	随机梯度下降 (SGD)	批量梯度下降 (batch GD)
计算开销	高（需要所有样本）	低（单个样本）	中等（小批量样本）
收敛速度	慢	快	快于GD，慢于SGD
稳定性	高	低	中等
内存开销	高	低	中等
参数调整难度	较低	较高	中等
适用场景	数据量较小，计算资源充足时	大规模数据集，需要快速迭代时	大规模数据集，平衡计算效率和模型性能时

机器学习=优化?



过拟合：经验风险最小化原则很容易导致模型在训练集上错误率很低，但是在未知数据上错误率很高。

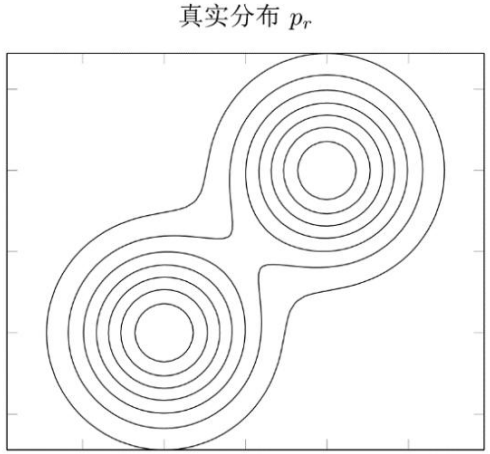
泛化错误

期望风险

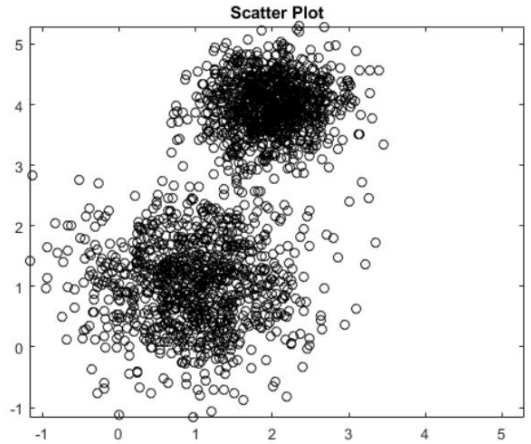
$$\mathcal{R}(f) = \mathbb{E}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[\mathcal{L}(f(\mathbf{x}), y)],$$

经验风险

$$\mathcal{R}_{\mathcal{D}}^{emp}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\mathcal{L}(y^{(n)}, f(x^{(n)}, \theta))$$



$\neq$



$$\mathcal{G}_{\mathcal{D}}(f) = \mathcal{R}(f) - \mathcal{R}_{\mathcal{D}}^{emp}(f)$$

泛化错误

如何减少泛化错误?

优化

经验风险最小

正则化

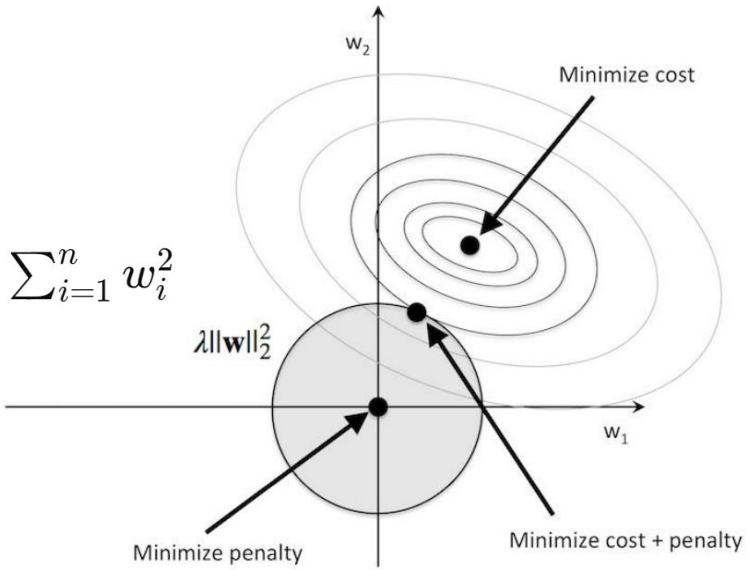
降低模型复杂度

正则化

增加优化约束

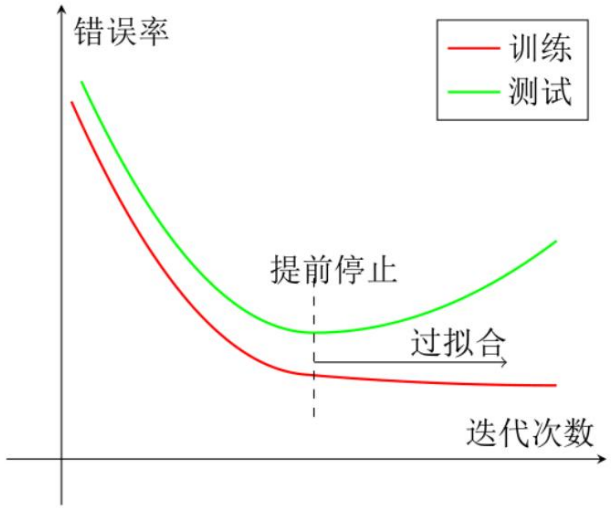
L1/L2约束、数据增强

$$L = L_0 + \lambda \sum_{i=1}^{n} w_i^2$$



干扰优化过程

权重衰减、随机梯度下降、提前停止



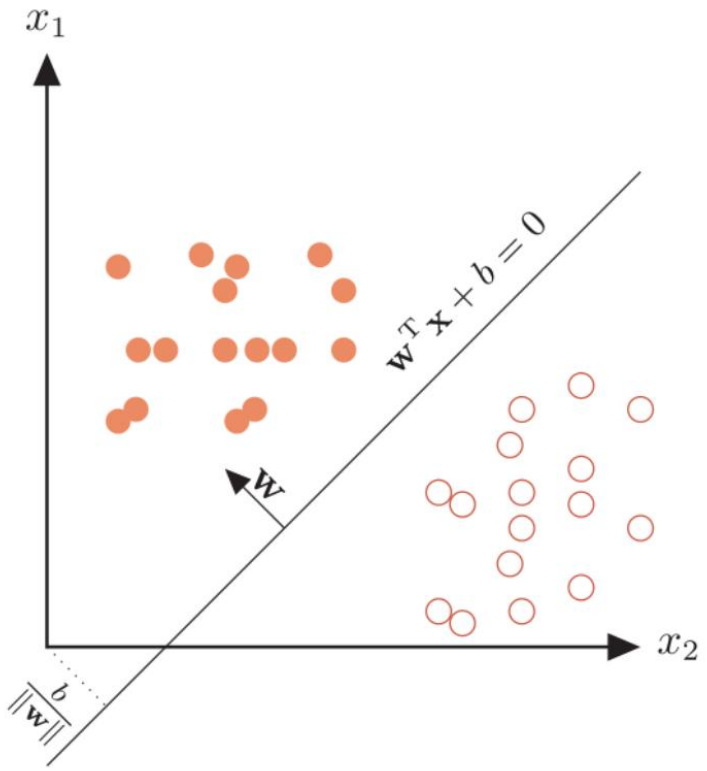
优化过程超参数设置例子：“Alpaca”

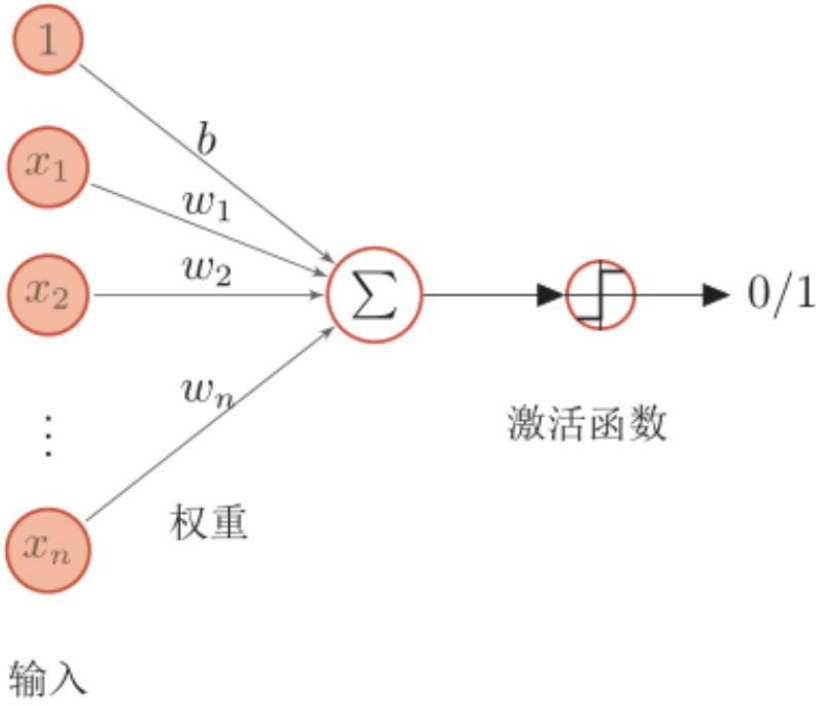
```
torchrun --nproc_per_node=4 --master_port=<your_random_port> train.py \
    --model_name_or_path <your_path_to_hf_converted_llama_ckpt_and_tokenizer> \
    --data_path ./alpaca_data.json \
    --bf16 True \
    --output_dir <your_output_dir> \
    --num_train_epochs 3 \
    --per_device_train_batch_size 4 \
    --per_device_eval_batch_size 4 \
    --gradient_accumulation_steps 8 \
    --evaluation_strategy "no" \
    --save_strategy "steps" \
    --save_steps 2000 \
    --save_total_limit 1 \
    --learning_rate 2e-5 \
    --weight_decay 0. \
    --warmup_ratio 0.03 \
    --lr_scheduler_type "cosine" \
    --logging_steps 1 \
    --fsdp "full_shard auto_wrap" \
    --fsdp_transformer_layer_cls_to_wrap 'LlamaDecoderLayer' \
    --tf32 True
```

https://github.com/tatsu-lab/stanford_alpaca

线性模型

它假设输入变量（自变量）和输出变量（因变量）之间存在线性关系，即模型预测的**输出是输入变量的加权和**



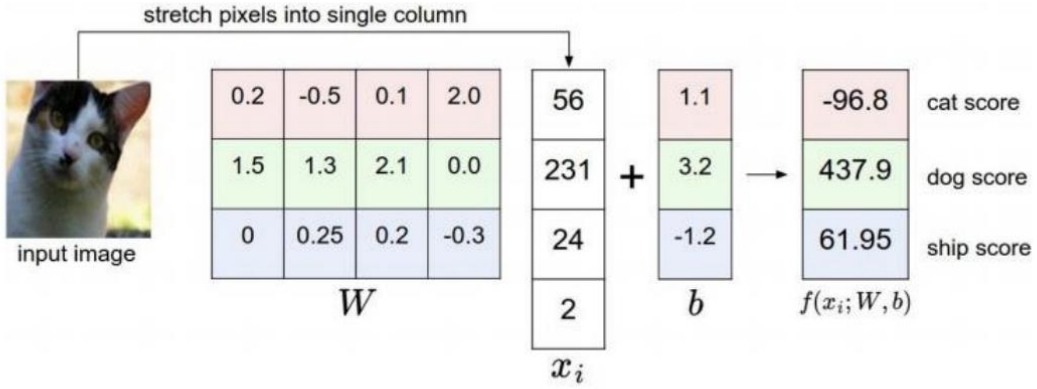


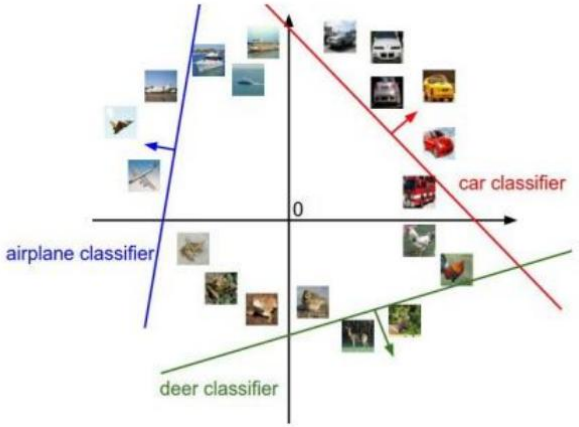
应用：图像分类

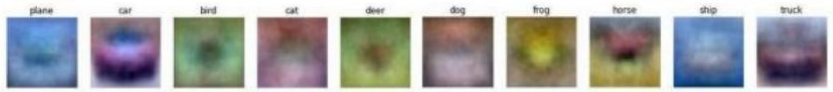
[32x32x3]
array of numbers 0...1
(3072 numbers total)

image parameters
f(x,W)

10 numbers, indicating
class scores







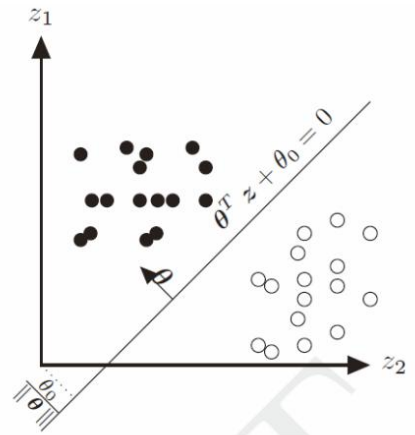
应用：文本分类

根据文本内容来判断文本的相应类别

D_1：“我喜欢读书”

D_2：“我讨厌读书”

	我	喜欢	讨厌	读书
D_1	1	1	0	1
D_2	1	0	1	1



\+

\-

感知器

模型

$$g(\mathbf{x}, \mathbf{w}) = \begin{cases} +1 & \text{当} \quad \mathbf{w}^{\mathrm{T}}\mathbf{x} > 0, \\ -1 & \text{当} \quad \mathbf{w}^{\mathrm{T}}\mathbf{x} < 0. \end{cases}$$

学习准则

$$\mathcal{L}(\mathbf{w}; \mathbf{x}, y) = \max(0, -y\mathbf{w}^{\mathrm{T}}\mathbf{x}).$$

优化：随机梯度下降

$$\frac{\partial \mathcal{L}(\mathbf{w}; \mathbf{x}, y)}{\partial \mathbf{w}} = \begin{cases} 0 & \text{当} \quad y\mathbf{w}^{\mathrm{T}}\mathbf{x} > 0, \\ -y\mathbf{x} & \text{当} \quad y\mathbf{w}^{\mathrm{T}}\mathbf{x} < 0. \end{cases}$$

两类感知器算法

输入: 训练集 $\{(\mathbf{x}^{(n)}, y^{(n)})\}, n = 1, \cdots, N$，迭代次数 T

初始化：$\mathbf{w}_0 \leftarrow 0, k \leftarrow 0$;
for $t = 1 \cdots T$ **do**
  随机对训练样本进行随机排序;
  for $n = 1 \cdots N$ **do**
    选取一个样本 $(\mathbf{x}^{(n)}, y^{(n)})$;
    if $\mathbf{w}_k^{\mathrm{T}}(y^{(n)}\mathbf{x}^{(n)}) \leq 0$ **then**
      $\mathbf{w}_{k+1} \leftarrow \mathbf{w}_k + y^{(n)}\mathbf{x}^{(n)}$;
      $k \leftarrow k + 1$;
    end
  end
end

输出: $\mathbf{w}_k$

Logistic回归

模型

$$p(y=1|\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x}) \triangleq \frac{1}{1+\exp(-\mathbf{w}^{\mathrm{T}}\mathbf{x})}$$

学习准则：交叉熵

$$\mathcal{R}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\left(y^{(n)}\log\hat{y}^{(n)} + (1-y^{(n)})\log(1-\hat{y}^{(n)})\right)$$

交叉熵是一种常用于衡量两个概率分布之间差异的度量方法.

优化：随机梯度下降

$$\frac{\partial\mathcal{R}(\mathbf{w})}{\partial\mathbf{w}} = -\frac{1}{N}\sum_{n=1}^{N}\mathbf{x}^{(n)}\left(y^{(n)}-\hat{y}^{(n)}\right)$$

Softmax回归

模型

$$P(y=c|\mathbf{x}) = \text{softmax}(\mathbf{w}_c^{\mathrm{T}}\mathbf{x})$$
$$= \frac{\exp(\mathbf{w}_c^{\top}\mathbf{x})}{\sum_{i=1}^{C}\exp(\mathbf{w}_i^{\top}\mathbf{x})}.$$

学习准则：交叉熵

$$\mathcal{R}(W) = -\frac{1}{N}\sum_{n=1}^{N}(\mathbf{y}^{(n)})^{\mathrm{T}}\log\hat{\mathbf{y}}^{(n)}$$

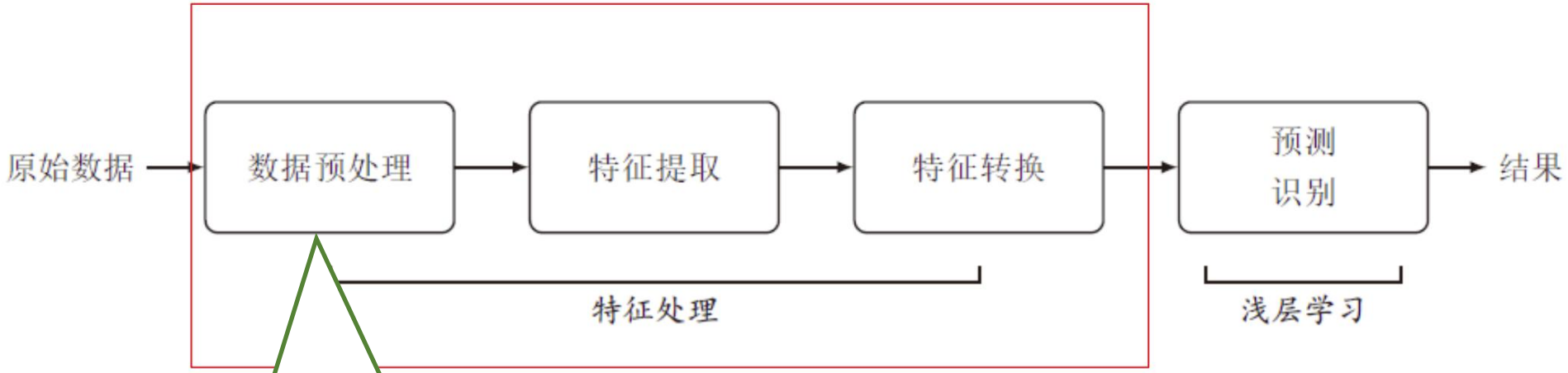
优化：随机梯度下降

$$\frac{\partial\mathcal{R}(W)}{\partial W} = -\frac{1}{N}\sum_{n=1}^{N}\mathbf{x}^{(n)}\left(\mathbf{y}^{(n)}-\hat{\mathbf{y}}^{(n)}\right)^{\mathrm{T}}$$

推导过程

特征工程问题

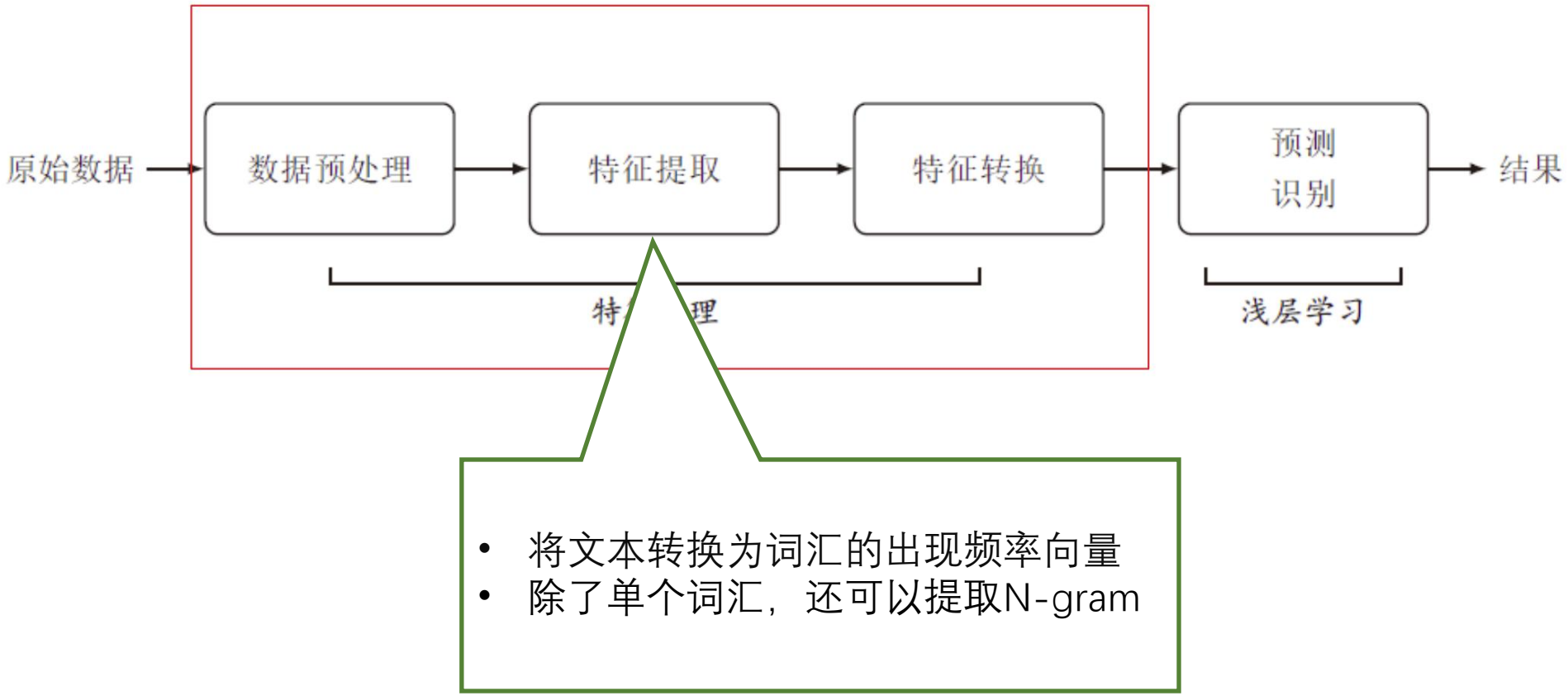
- 模型
  - 在实际应用中，特征往往比分类器更重要



- 移除电子邮件中的HTML标签
- 将所有文本转换为统一的大小写
- 分词
- 去除停用词

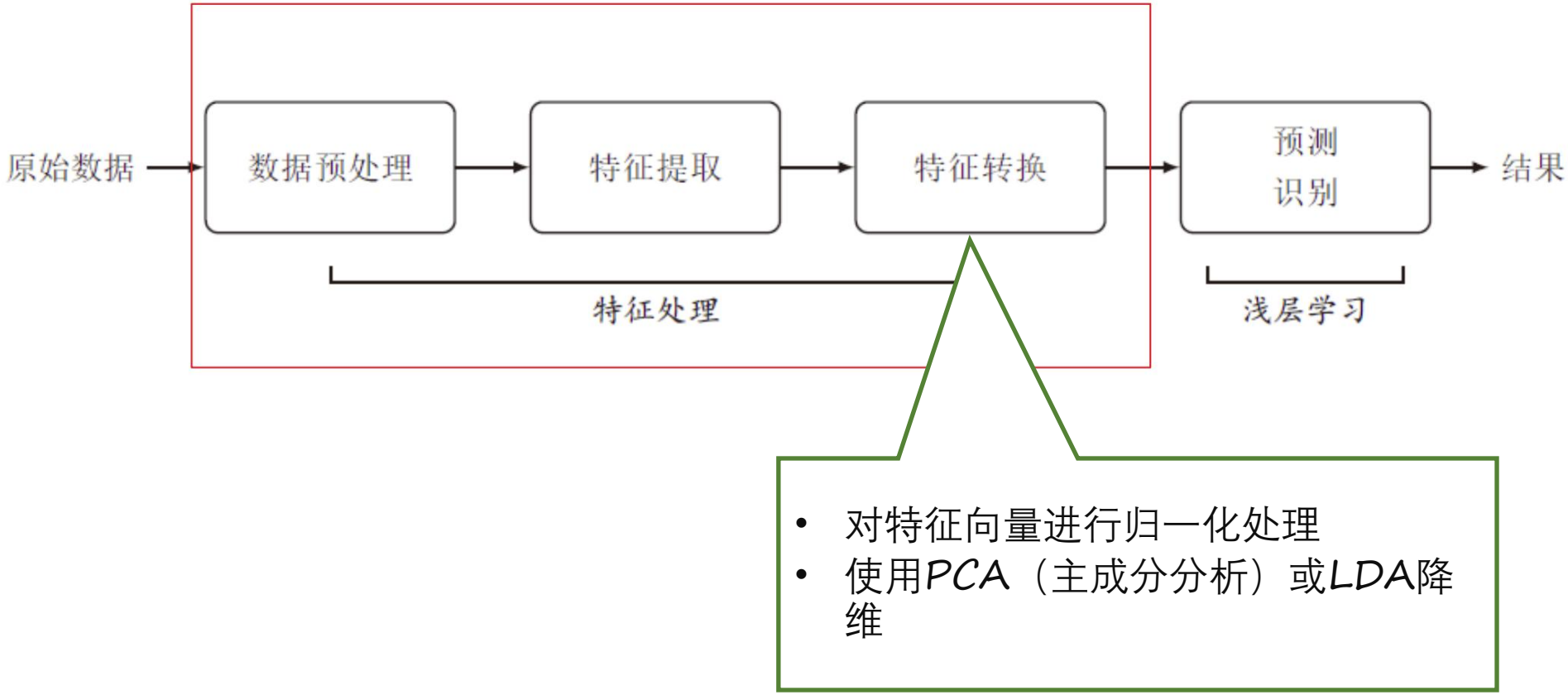
特征工程问题

- 模型
  - 在实际应用中，特征往往比分类器更重要



特征工程问题

- 模型
  - 在实际应用中，特征往往比分类器更重要



特征工程问题

- 模型
  - 在实际应用中，特征往往比分类器更重要