



# CPPT BERT 无监督训练 聚类等方法

估计一个给定文本序列在语言上的合理性或概率



$$P(w_j | w_1, w_2, \dots, w_{s-i}, w_j)$$

↑  
模型参数

Word2vec: 实现 Embedding 矩阵的训练

CBOW: 更新周围词向量

window  $m \times$  词库大小为  $m$

$\rightarrow$  向量维度为  $D$

window  $m \times$

$2m+1 \rightarrow V \times D \rightarrow \text{avg}() \Rightarrow 1 \times D \rightarrow D \times V \Rightarrow 1 \times V \rightarrow \text{softmax} \Rightarrow \text{Loss}$

one-hot hidden1 加和平均 向量 hidden2 初步结果 预测中心词

所需结果

Skip-gram: 更新中心词向量

$1 \times V \rightarrow V \times D \Rightarrow 1 \times D \rightarrow D \times V \Rightarrow 1 \times V \rightarrow \text{softmax} \Rightarrow \text{Loss}$

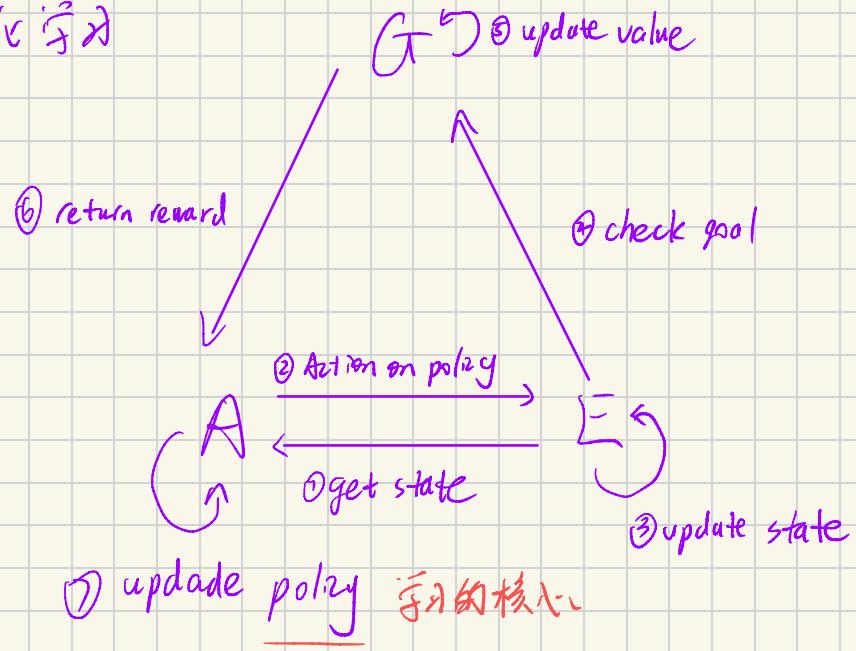
所需结果

一个中心词，一个周围词，但有  $2m$  个样本

FFNN：前馈神经网络 只有前向通路，不要求反馈

n-gram: 用n-1个词预测第n个

# 强化学习



```

graph LR
    A[学习方法] --- B[监督]
    A --- C[自监督]
    A --- D[无监督]
    A --- E[强化学习]

```

传统机器学习  
的数据处理流程

↓

数据预处理

↓

特征提取  
序列的东西

↓

特征转换

↓

预测

↓

结果

特征处理

特点：人工设计的

表子/特征  $\Rightarrow$  对数据编码或表示 特征空间的表子是一种形式

端到端

传统 NLP

分词  $\rightarrow$  词性标注  $\rightarrow$  句法分析  $\rightarrow$  语义分析  $\rightarrow$  语义推理

end2end

input  $\rightarrow$  black box  $\rightarrow$  output

深度学习的数据处理流程

原始数据



底层特征



中层特征



高层特征



预测

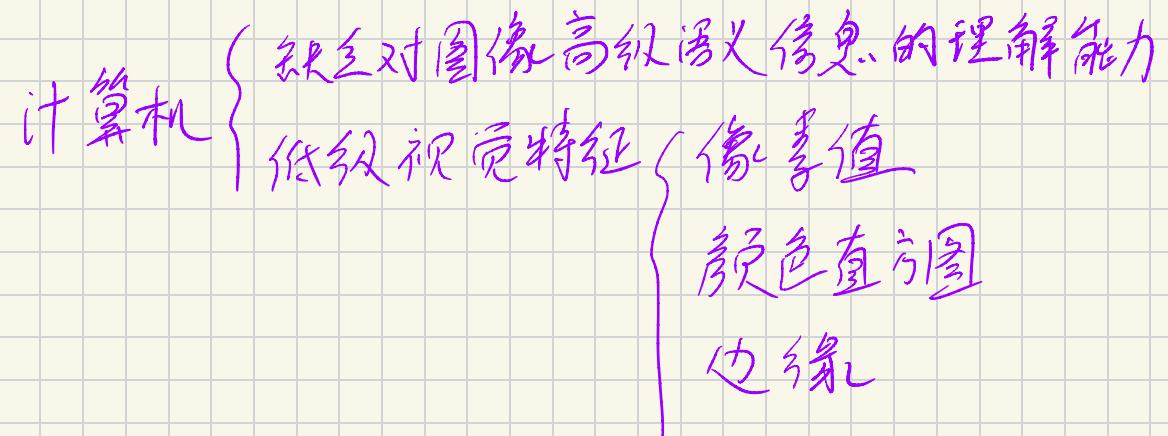
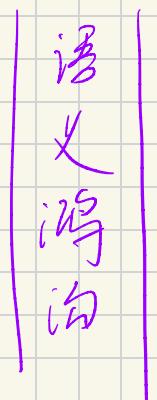
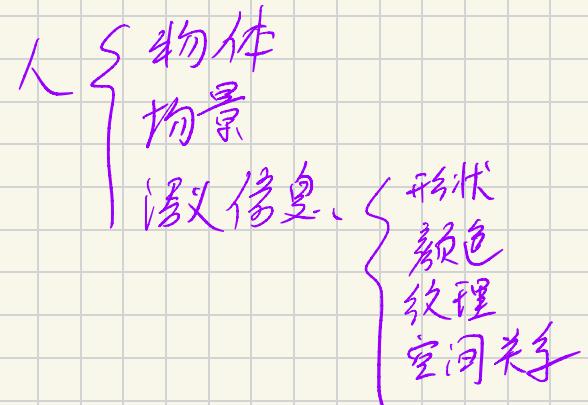


结果

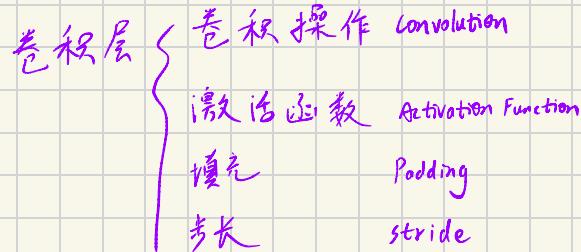
表子学习

深度学习

# MLP



# CNN

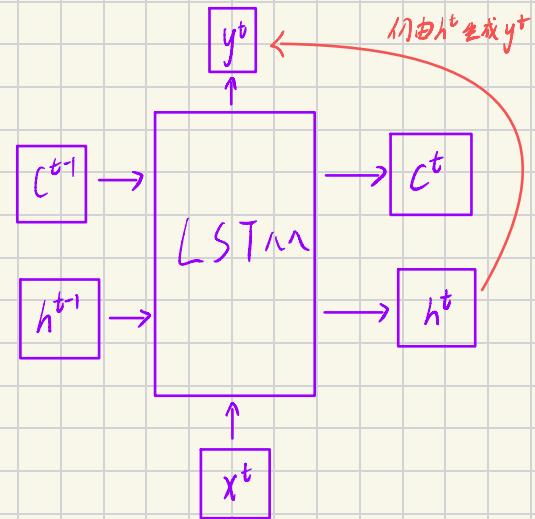
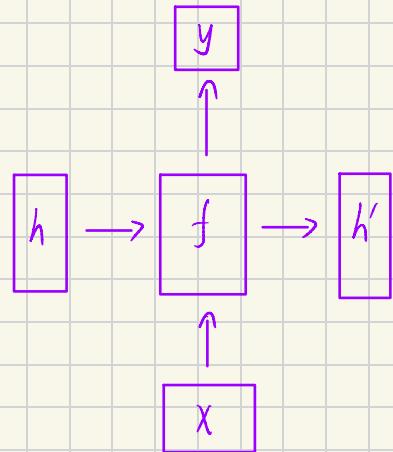


池化层：下采样

相比于MLP，如何减少参数？

- | 局部感受野
- | 权值共享
- | 池化层

# RNN



$$h' = \sigma(W^h h + W^i x + b_h)$$

$$y = \sigma(W^o h' + b_y)$$

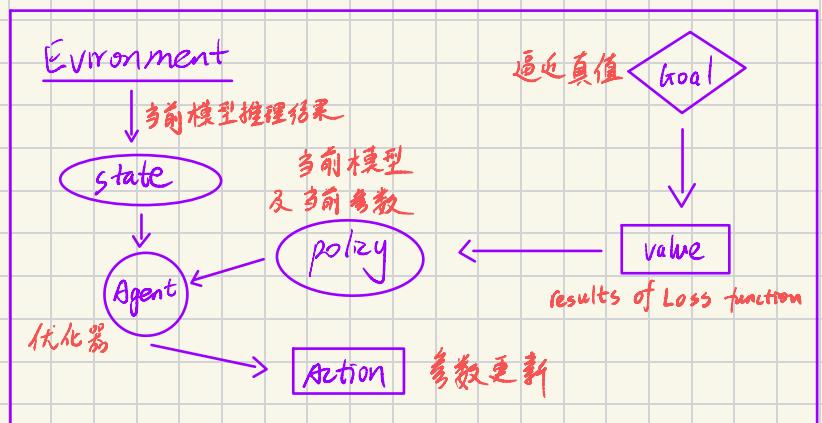
困难

- | 共用参数
  - | 记忆过强
  - | 幅度消失
  - | 爆炸
- | 难以并行

- ① 选择性遗忘 sigmoid
- ② 选择性记忆 sigmoid + tanh
- ③ 选择性输出 sigmoid

## Reinforcement Learning

在当前状态  $s_t$  下，制定一个最优策略  $\pi$ ，找到一个最佳动作  $a_t$ ，得到最大回报。如果可能，将策略集  $\{\pi\}$  和动作集  $\{a\}$  存储起来。



Value: { exploitation: 根據 value function 找 policy  
exploration: 优化 value function 本身

**最优行为价值函数**: 指不管政策如何, 当前的一步行为可以产生的回报的上限

$$Q_{\pi}^*(s_t, a_t) = \max_{\pi} Q_{\pi}(s_t, a_t)$$

抛开 policy, 对一个 action 做评估

The diagram illustrates the Reinforcement Learning (RL) framework. At the top left, the text "RL: environment" is followed by a purple arrow pointing right to "State". From "State", a purple arrow points down to "Goal". From "Goal", a purple arrow points right to "value". Below "value", the text "value function need to be learned" is written in red. From "value", a purple arrow points right to "Agent". From "Agent", a purple arrow points right to "Action". A blue arrow labeled "policy" points from "value" up towards "State". A blue arrow labeled "Reward + New state" points from the bottom right up towards "State". Above "value", the text "價值的 actions 也很重要" is written in orange.

# 在一个完整的价值学习过程中

$$\begin{aligned}
 & S_{t+1}, r_t \\
 & \downarrow \\
 & t \\
 & \quad \quad \quad q_t = Q(s_t, a_t; w_t) \longrightarrow \text{action} \\
 & t \\
 & \quad \quad \quad y_t = r_t + \gamma \max_a Q(S_{t+1}, a; w_t) \\
 & \quad \quad \quad L = \frac{1}{2} (q_t - y_t)^2 \\
 & \quad \quad \quad w_{t+1} = w_t - \alpha \frac{\partial L}{\partial w} \quad | \quad w = w_t
 \end{aligned}$$

## 状态价值函数

$$V_{\pi}(s_t) = \mathbb{E}_A [Q_{\pi}(s_t, A)] = \int \pi(a | s_t) * Q_{\pi}(s_t, a) da$$

$$\left\{ \begin{array}{l} \text{行为价值函数: } Q_{\pi}(s_t, a_t) = E[V_t | S_t = s_t, A_t = a_t] \\ Q_{\pi}^*(s_t, a_t) = \max_{\pi} Q_{\pi}(s_t, a_t) \\ \text{策略价值函数: } \max_a V_{\pi} \end{array} \right.$$

## Seq2seq

SMT

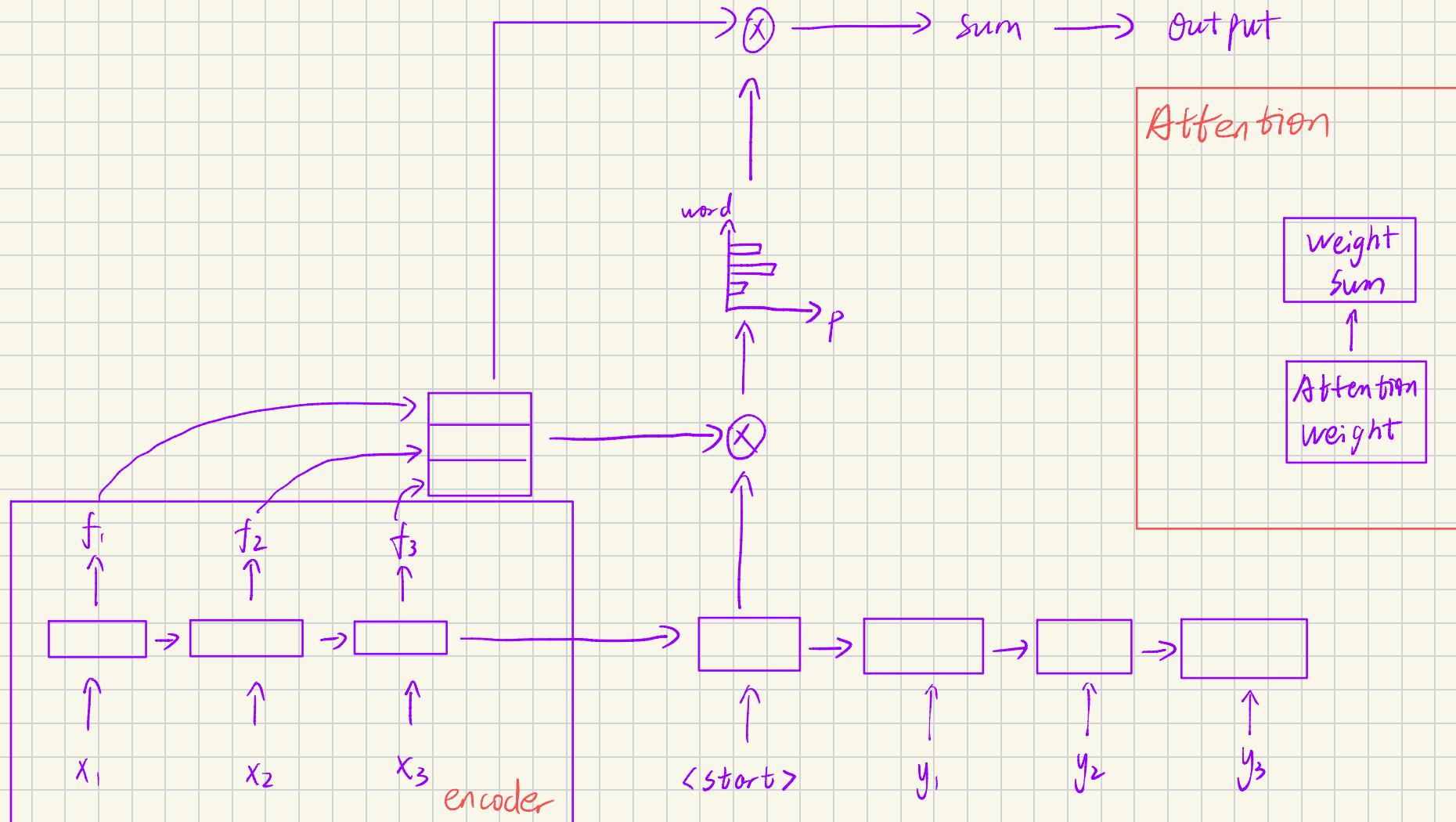
NMT

{ Greedy Decoding

Beam Search      解决 Greedy Decoding(贪心) 只保证局部最优, 不保证全局最优

## Attention

Seq2seq 只有最终的隐向量经 decoder, Attention 让之前的隐层向量都参与进 Decoder



# Pre-trained Model PTM

预训练不一定无监督，但通常是的

预训练是迁移学习的一种特定的形式

思想：① 模型参数不再随机初始化，而是通过一些任务（如语言模型）进行预训练  
② 将训练任务拆解成共性学习和特性学习两步

用于下游任务的策略

固定特征提取器：只取预训练模型的特征提取器部分，并冻结权重，用于特征提取  
基于微调：不冻结权重，有监督训练，更新参数

第一代 NLP 预训练

LM-LSTM  
Sequence Autoencoder

## 第二代 NLP PTM

wl context-aware 为核心特征

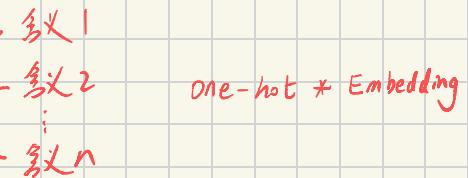
wl ELMO GPT BERT 为代表

Word2vec 和 Glove 的问题：一个词只有一个向量来表示

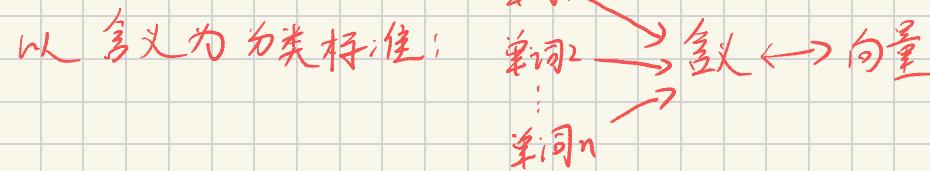
ELMO 用 特定任务上训练好的双向 LSTM 做词嵌入

Word2vec  
Glove

静态词嵌入：因为它们以 单词为分类标准，单词  $\leftrightarrow$  向量



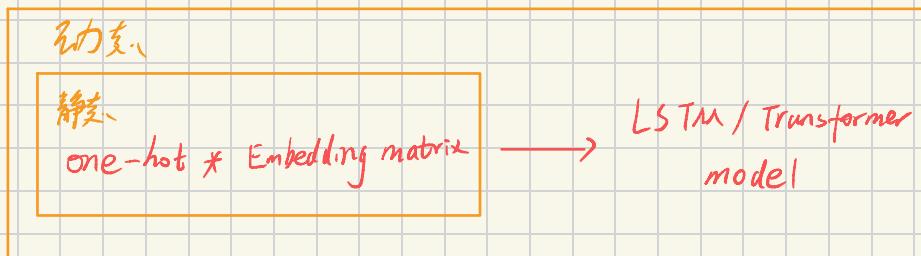
ELMO 动态词嵌入：



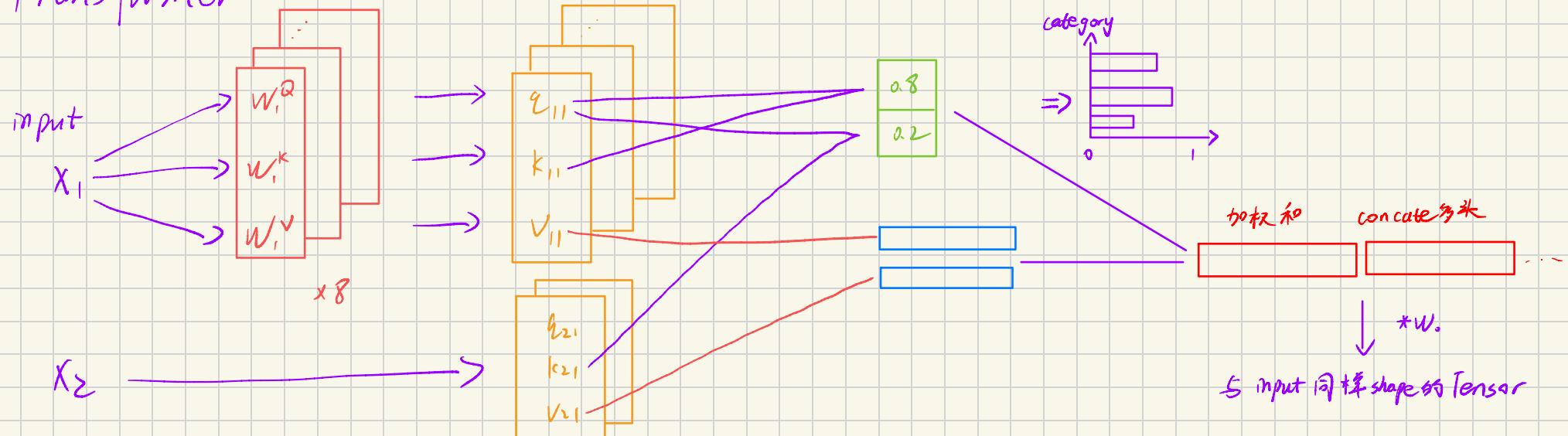
S Claude 聊一下

静态指：one-hot \* Embedding matrix = word embedding vector

ELMO 的动态指：在静态基础上，将 Bi-LSTM 也算做 Embedding 的一部分



# Transformer

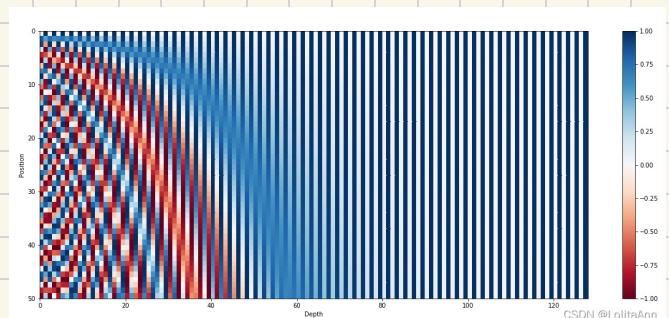


Masked Multi-Head Attention

$Q, K \longrightarrow \text{softmax}$

	A	B	C	D
A	0.11	-inf	-inf	-inf
B	0.19	0.50	-inf	-inf
C	0.33	0.98	0.95	-inf
D	0.81	0.86	0.88	0.90

	A	B	C	D
A	1	0	0	0
B	0.48	0.52	0	0
C	0.31	0.35	0.34	0
D	0.25	0.26	0.23	0.26



\* encoder-decoder attention \*

$Q$  来自 decoder 上一层的 self-attention 的结果

$K, V$  来自 encoder 的输出

position encoding

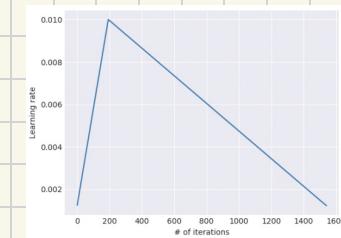
- ① 每个时间步都有唯一编码
- ② 在不同长度的句子中，两个时间步之间的距离应该一致
- ③ 模型不受句子长短的影响，并且编码范围是有界的（不会随着句子加长数字就无限增大）
- ④ 必须是确定性的

# ULM-FIT

## 利用CU的预训练模型到NLP中

### 三个新微调技术

- ① Discriminative fine-tuning : 微调过程中，每层学习率被设置为不同的值
- ② 斜三角学习率 (slanted triangular learning rates, STLR)
- ③ 渐进式解冻 (progressive unfreezing) 分段训练



# GPT

{ GPT decoder only 生成模型

| BERT encoder only 自然语言理解

{ one-shot: 例如, 只提供一张人脸的 fine-tune

| zero-shot: 对一个模型做它没微调过任务

| few-shot: 标注少量优质数据, 对模型进行微调

GPT 1 —— 模型统一, 根据任务需求, 重置输入, 添加标记符 { start  
Delim  
Extract

GPT 2 —— zero-shot (预训练后, 不做任何微调, 直接应用于任务)

多任务学习 (Multi-Task Learning, MTL)

NLP 中, 将所有任务统一成 QA 任务

难点: 不能有标记符 从 zero-shot 到 prompting

文章贡献度 = 新意度 \* 有效性 \* 问题的大小

GPT 3 —— 不做 fine-tune 参数量太大, 又难以梯度下降

重要概念的提出: 上下文学习 (in-context learning, ICL)

# Prompting

## 三个范式

{① 上下文学习: LLM

② 指令精调: Instruction-tuning

③ 翻译组: CoT

## 区分

{① Prompting: 给LLM一段文本，引导它输出我们想要的东西

② Prompt tuning: 会改变模型参数，但比 fine-tune 高效

③ Prompt learning: 清风提示词的机器学习范式也算是提示学习

## 进化

传统 pre-train + fine-tune: input  $x$  model predict  $y$

新范式 pre-train + prompt + predict: input  $x$ , according to template, get  $x'$

but  $x'$  not complete, model fills  $x'$  based on probability, get  $\hat{x}$

based on  $\hat{x}$ , model predict  $y$

## 发展历程

① 基于符号主义的全监督学习时代 ② 基于神经网络的全监督学习时代 ③ 预训练+微调阶段 ④ 预训练+提示词+预测阶段

① 特征工程

(手动提取特征)

② 结构工程

(搭建更好的神经网络结构  
以更好的提取特征)

③ 固标工程

(选取好的目标函数)

④ 提示工程

(精心设计 prompt, 让  
语言模型生成指定内容)

# 参数高效学习 Parameter-Efficient Transfer Learning

{ 全面微调：正常 Fine-tuning

参数高效微调：只更新全部参数

Parameter-Efficient Fine-tuning, PEFT

{ 选择法：Selective method 排选需要更新的参数

{ 附加法：Additive method Adapter pre-trained model + layers

基于重参数化的方法 Reparametrization-based PEFT 低秩适配 (Low-Rank Adaptation, LORA)

input +  $t_{\text{prompt}}$  + fill  $\Rightarrow$  output  
↑                   ↑  
Template          LLM

{ Manufacture

{ Automated

① { discrete prompts 文本构成 One-hot 自然语言  
continuous prompts Embedding vector 组合 Embedding 只有LM理解  
prompt tuning 只调模型参数

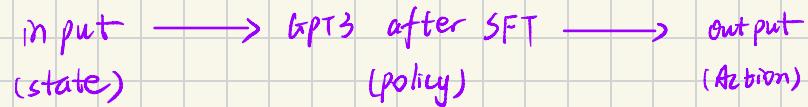
② { static prompt 基本固定模板

Dynamiz prompt 生成自定义模板

ChatGPT / InstructGPT / GPT3.5

基于人类反馈的强化学习(RLHF)

Step1 有监督微调(SFT) / 指令精调 (Instruct-Tuning)



Step2 训练一个奖励模型, Reward Model, RM 有监督学习

Step3 以大模型本身为策略函数, 以训练出的RM为奖励函数  
通过 PPO 算法去微调模型

近端策略优化, Proximal Policy Optimization

① 一个LLM如GPT3, 初步具备了一些zero-shot和Few-shot的能力, 这可能是因为语料库中本来就有类似的指令, 完成人类向他提出的下游任务要素可以帮助他更好的实现下一句预测

② 被Train on Code后具备了代码生成能力, 而且可能从中学会了Reasoning, 即逻辑推理

③ 被Instruct-Tuning后, 学会了如何遵循人的指令, 而且这件事具有很强的泛化能力

④ 被RLHF后, 付出了一些性能作为代价(Calignment tax), 进一步学会了遵循人的指令, 输出 Useful、relevant而且not-toxic 的内容, 实现了与人类的价值观对齐

Co T

Self-consistency

Emergent Abilities

大语言模型的涌现：如果一种能力在较小的模型中不存在，但在较大的模型中有存在，认为这种能力是涌现的

# BERT

词嵌入

词元嵌入

序位嵌入

位置嵌入

任务 Masked Language Modeling (MLM)

Next Sentence Prediction (NSP)