人工智能基础A复习

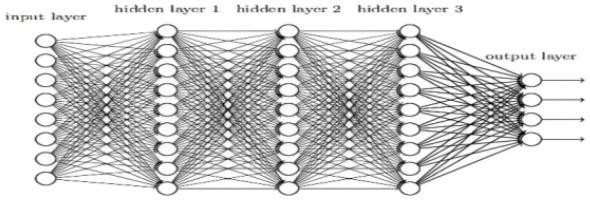
介绍

- 主要的考试范围应该就是围绕深度学习的部分

深度学习概览

- 深度学习(Deep Learning ,DL)是机器学习的一个子集，它使用多层人工神经网络来精准完成诸如物体检测、语音识别、语言翻译等任务。
  - 基本思想：通过构建多层网络，对目标进行多层表示，以期通过**多层**的**高层次特征**来表示数据的抽象语义信息，从而获得更好的**特征鲁棒性**

特征

- **多层网络结构**：由多层神经元组成，包括输入层、隐藏层和输出层。这些层之间通过权重和偏置进行连接，形成复杂的网络结构
- 

- **自动特征提取**：自动从原始数据中提取有用的特征，无需人工设计特征工程。
- **非线性激活函数**：深度学习模型中通常使用非线性激活函数（如ReLU,sigmoid等），使得模型能够学习复杂的非线性关系
- **大规模数据处理能力**：借助现代计算机书和大数据资源，深度学习模型能够处理大规模数据集，提高模型的准确性和泛化能力

细分子集

1. 多层感知机（MLP）
2. 卷积神经网络（CNN）
3. 循环神经网络（RNN）
4. Transfomer

5. 扩散模型（Diffusion）

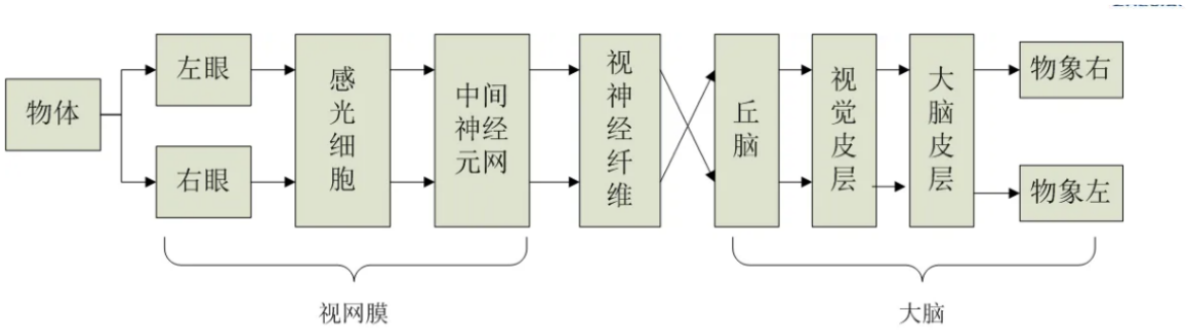
神经网络的分类

---

1. 前馈神经网络
    1. 多层感知机 MLP
    2. 卷积神经网络 CNN
2. 反馈神经网络
    1. 循环神经网络 RNN
    2. 长短程记忆网络 LSTM
    3. Hopfield 网络
    4. 玻尔兹曼机
3. 图网络
    1. 知识图谱
    2. 社交网络
    3. 城市交通

人眼识别物体的基本过程

- 流程图

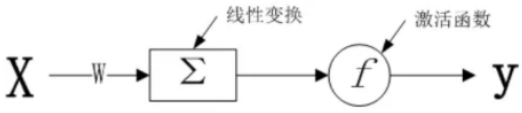


- 神经元的“全或无”现象
    - 只有当外来刺激有足够大的强度，才能引起神经元细胞的兴奋并产生动作电位，但在增加刺激强度并不会导致动作电位幅度发生变化
    - 动作电位的传播范围和距离不会随刺激强度的不同而不同

多层感知机（感知机模型）

- 流程图

$$X \overset{weigh}{\rightarrow} \Sigma \rightarrow activate(f) \rightarrow y$$

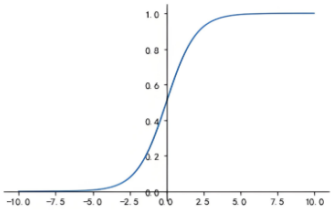


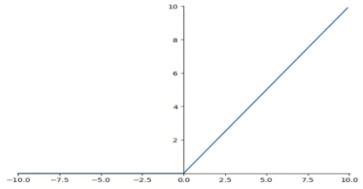
- y公式的形式化表达
  - $if\ X = [x_1, x_2, x_3], W = [w_1, w_2, w_3]$
  - $y = f(w_1x_1, w_2x_2, w_3x_3 + b)$
  - 其中f为特定的激活函数，b为偏置量

激活函数

- 感觉直接接在感知机模型后面比较顺畅，可以与前面的计算联系起来

主要函数

1. Sigmold
   - $f(x) = \frac{1}{1+e^{-x}}$
   - 示意图
   - 
   - 特性
     - 非线性
     - 在（-3,3）之间，优化比较明显
     - 在（-3,3）之外，优化不明显
     - 值域在0~1之间，是非对称算法，这意味着下一个神经元只能接受正值的输入
2. ReLU
   - $f(x) = max(0, x)$
   - 示意图



- 特性
  - 非线性
  - 不会同时激活所有神经元
  - 计算速度快
  - 有趋于0的梯度
  - ReLU是目前隐含层中最为常用的损失函数

3. Softmax
   - $S_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$
   - **归一化**:
     - Softmax是将一个包含任意实数的K维向量压缩成另一个K维的实数向量，可以被解释为概率分布，每个概率对应每一个输入元素的属于某个类别的概率
   - 用途：
     - 应用最为广泛的一个激活函数分类器，常用语最后一层，输出分类的概率结果
     - 手写数字识别、物体分类、蛋白质结构分类、气象预测等应用
   - 计算（简单理解就是每个都带进去求和）

例 8-5：对于输入$X = [2, 0.7, -1.5, -0.9]$，计算 Softmax 输出。

解：

$$sum = \sum_j e^{x_j} = e^2 + e^{0.7} + e^{-1.5} + e^{-0.9} = 10.03$$

$$S_1 = e^{x_1}/sum = e^2/10.03 = 0.74$$

$$S_2 = e^{x_2}/sum = e^{0.7}/10.03 = 0.20$$

$$S_3 = e^{x_3}/sum = e^{-1.5}/10.03 = 0.02$$

$$S_4 = e^{x_4}/sum = e^{-0.9}/10.03 = 0.04$$

常用损失函数

- 回归损失函数（用于衡量回归系统的误差）

均方误差（mean_squared_error）

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(T_i - Y_i)^2$$

平均绝对误差（mean_absolute_error）

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|T_i - Y_i|$$

平均绝对百分比误差（mean_absolute_percentage_error）

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}|\frac{Y_i - T_i}{Y_i}| \times 100$$

均方根对数误差（mean_squared_logarithmic_error）

$$MSLE = \frac{1}{N}\sum_{i=1}^{N}[log(T_i + 1) - log(Y_i + 1)]^2$$

- 分类损失函数（用于衡量分类系统的误差）

（2） 分类损失函数，用于衡量分类系统的误差

二进制交叉熵（binary_crossentropy）：用于二分类问题，对应的激活函数是sigmoid

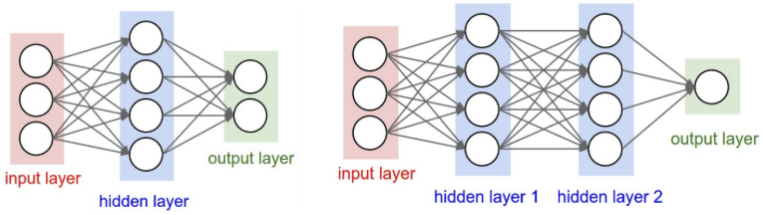
$$Loss = -\frac{1}{N}\sum_{i=1}^{N}[Y_i \times logT_i + (1 - Y_i) \times (1 - T_i)]$$

多分类交叉熵（categorical_crossentropy）：用于多分类问题，对应的激活函数是softmax

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}Y_i \times logT_i$$

多层感知机（MLP）

- 示意图



- 多层感知机是罗布森拉特标准感知器的扩展。如果以神经元来计算层数，一个多层感知器至少包含三层：一个输入层，一个隐藏层和一个输出层
- 数据通过输入层进入网络，乘以连接的权重后输入到隐藏层节点。隐藏层节点将这些加权后的输入求和并经过一个非线性变换（激活函数）后送往输出层
- 只有一个隐含层的叫浅层学习网络，隐含层大于一层的叫深度学习网络

误差反向传播算法-BP

- 反向传播算法是一种常见的人工神经网络学习算法，特别实用于多层前馈神经网络的训练
- 该算法由学习过程中的信号正向传播与误差的反向传播两个过程组成
  - 有点像是模电里面的反馈
- 流程图

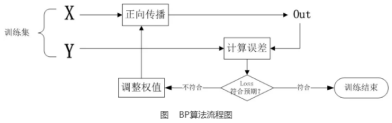


图 BP算法流程图

- BP算法的特点
  - 自适应、自主学习
    - BP能够根据设定的参数更新规则，不断地调整神经网络中的参数，来达到最符合期望的输出
  - 较强的非线性映射能力
    - 由于神经网络的激活函数都是非线性的，所以BP算法能够处理复杂的非线性问题
  - 严谨的推导过程
    - 误差的反向传播采用的是已经非常成熟的链式法则，推导过程严谨且科学
  - 较强的泛化能力
    - 训练技术之后，BP算法可以利用从训练数据中学到的知识解决新的问题
- BP算法的局限性
  - 容易陷入局部最小值
    - 由于BP算法采用的是梯度下降法，容易陷入局部极小值而得不到全局最优解
  - 收敛速度慢

        - 由于神经网络的参数众多，每次迭代都需要更新大量的权值和阈值，故收敛速度较慢
    - 隐节点选取缺乏理论指导
        - 传统方法都是不断试凑来做的
    - 学习新样本的时候可能会遗忘之前学习过的旧的样本

梯度下降法

- 以$f(x,y)$为例,假设学习率为η
    - $grad(f)=\left[\frac{\partial f}{\partial x},\frac{\partial f}{\partial y}\right]$
    - 从梯度反方向开始下降寻找
    - 满足：$x_i=x_{i-1}-\eta\frac{\partial f}{\partial x};y_i=y_{i-1}-\eta\frac{\partial f}{\partial y}$
    - 不断迭代，直到到达规定终点（终止条件）
- 梯度消失
    - 层数太多导致
- 梯度爆炸
    - 学习率过大或者初始权重太大
- 局部最优
    - 只能获得局部最优解

卷积神经网络（CNN）

图像的数字化表示、感受野

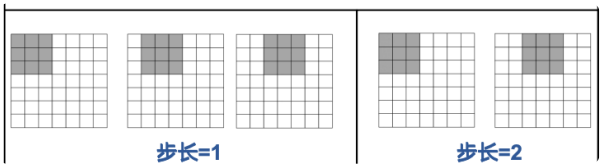
- 现实世界中的图像不能用直接用于计算处理，必须先经过数字化处理，比如扫描仪、数字相机等。
- 为了统一规范，采用RGB三原色通道来表示，每种颜色的强度用一个字节来表示，取值范围0-255
- 颜色数：256x256x256=16777216

CNN拓扑结构

- 三要件：卷积、池化、展平

卷积运算

- 卷积运算就是通过设计一系列大小适中的卷积核（感受野），对数字图像的各通道分量进行卷积，提取特征值得过程
- 示意图
- 

    - 对于图的解释：相当于就是找了一个系数矩阵（卷积核），对输入部分某一个点的周边进行计算（矩阵乘法）
- 常用卷积核大小（形状）为3x3,5x5,7x7
- 步长：决定运算的速度
- 

- 卷积核的数量：决定每一层能够提取的特征数

池化（pooling）

- 也称下采样
- 作用时缩小特征图的尺寸，减少计算量。
- 原理：用某一成像区域子块的统计信息包含了该子块的全局信息（代表）
- 分类
    - 最大池化
    - 平均池化
    - 随机池化
    - L2范数赤化
    - K-max池化
    - 全局平均池化

- CNN常用2x2区域进行池化
- 下列图进行2x2平均池化，步长为2

例9-2：对如下的特征图进行平均池化计算，池化窗口2×2，

23	34	55	32	66	43
15	43	27	30	39	54
33	67	47	28	36	49
56	89	90	35	77	65
54	32	37	48	65	32
63	67	38	43	29	54

步长=2的池化

  - 结果：变成3x3的矩阵，值就是原来每个对应2x2方格的平均值

展平与独热码

- 独热码(One-Hot code)：在分类问题中，我们不能将不同的分类用1,2,3...等有序的数字来表示，而是用独热码这样的向量矩阵来进行编码，来工计算机识别计算，并用索引所在位置的概率值来预测最终的分类结果

CNN本质-层次特征学习

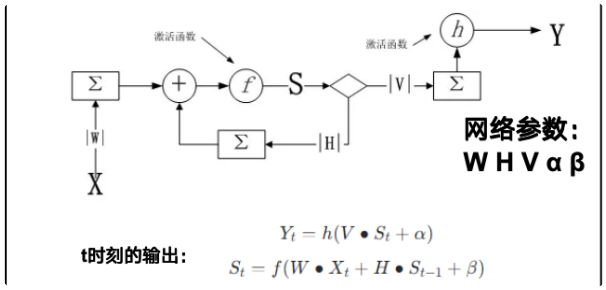
- 层次特征学习：深度学习网络之所以强大，一个很重要原因就在于它可以层次性处理并逐渐提取抽象特征

循环神经网络（RNN）

- Recurrent Neural Network

RNN的结构

- 是一种特殊类型的反馈神经网络，专门用于处理序列数据
- 核心思想
  - 循环结构使得网络能够捕捉和利用序列中的顺序依赖性信息
- 基本单元

    - 一个具有循环连接的神经网络层
    - 这个循环连接允许网络在处理每个时间步的数据时，能够利用时间步的信息
- 流程：RNN在每个时间步都会接收到一个输入产生一个输出，同时其内部状态（隐藏状态）会被更新和传递到下一个时间步
- 典型示意图
- 

    - 简直就是反馈电路，连符号都差不多

RNN计算

例10-1：已知拓扑结构为同步多对多RNN（基本结构见图 RNN基本逻辑结构），输入层、隐含层（一层）、输出层的神经元均为一个，激活函数均为 ReLU，$W=[0.5,0.1,0.2]$，$H=[1]$，$V=[3]$，$S_0=0$，$\alpha=0$，$\beta=0$，对 X =[[1,1,1]，[2,2,2]，[3,3,3]] 的输入序列，计算其输出序列 Y 。

解：$X_1=[1,1,1]$　$X_2=[2,2,2]$　$X_3=[3,3,3]$

$$S_1 = f(W \bullet X_1^T + H \bullet S_0) = f(0.8) = 0.8$$
$$Y_1 = h(V \bullet S_1) = h(2.4) = 2.4$$
$$S_2 = f(W \bullet X_2^T + H \bullet S_1) = f(1.6 + 0.8) = 2.4$$
$$Y_2 = h(V \bullet S_2) = h(7.2) = 7.2$$
$$S_3 = f(W \bullet X_3^T + H \bullet S_2) = f(2.4 + 2.4) = 4.8$$
$$Y_3 = h(V \bullet S_3) = h(7.2) = 14.4$$
$$\therefore Y = [2.4, 7.2, 14.4]$$

-
    - 激活函数$f(x), h(x)$均为$ReLU$函数
- 公式
    - $Y_t = h(V \cdot S_t + \alpha)$
    - $S_t = f(W \cdot X_t + H \cdot S_{t-1} + \beta)$

RNN的问题

- 无学习太长的序列，会“很快忘记前面说过的话”
- 所以引入了一种新的结构---*LSTM*

LSTM

- Long Short-Term Memory 长短期记忆网络
- 专门设计用于解决长期依赖的问题
- LSTM结构
  - 遗忘门
    - 决定什么时候把以前的状态遗忘
  - 输入门
    - 决定什么时候加入新的状态
  - 输出门
    - 决定什么时候把状态和输入叠加输出
  - 记忆状态
    - 累计历史信息，调控h_f输出内容
  - 隐式编码
    - 与下一次输入一起参加运算
- 局限性
  - 计算复杂
  - 顺序处理，无法进行并行化计算
  - 计算慢，实时响应难

自然语言处理（NLP）

NLP的重要性

- 自然语言处理的准确性是机器具备智能的必要条件

NLP任务分类

---

1. 文本转换
   - 翻译
   - 编程代码
   - 加密解密
   - 格式转换

    - 文本转语音
2. 文本生成
    - 聊天机器人
    - 写作
    - 问答
3. 语音识别
    - 特征提取
    - 转文本
    - 执行指令
4. 文本分析
    - 语义关系
    - 句法分析
    - 命名实体识别
    - 情感分类
    - 关键词提取
    - 搜索引擎

NLP技术演变

- 基于规则算法
- 统计语言模型（也是一种规则）
    - 大规模文本数据
- 序列生成模型
    - 输入序列->编码器->处理器->解码器->输出序列
- 预训练-微调模型（两种模式）
    - 全量微调
        - 更好适应新任务
        - 数据量要求大
    - 参数高效微调
        - 一般成本较低
        - 只调一部分参数

NLP语言基础

分词

- Token：令牌，可以是一个字，也可以是一个词...
- 一个"Token"就是通过分词技术（工具）将一句话分割成最小的单位，是一个特定的自然语言处理模型能处理的最基本元素
- 分词需在词汇表和语料大小之间取得平衡，也是模型成败的最关键一步
- 开源模型都自带分词工具和词汇表

词向量和词嵌入

- 词向量就是词的特征分布，是NLP模型层与层之间进行信息传递的数据形式
- 词向量和词嵌入是指的同一个东西，区别在于词向量是指数字编码技术，词语嵌入是指NLP网络之间的数据存在形式

文本相似度

- 有了词向量的表示，就可以非常容易的计算文本相似度
- 一般的，语义相近的词在向量空间上具有相近的位置
- **余弦相似度**
  - 假设两个文本的向量为$\vec{A}, \vec{B}$，那么$\vec{A}$和$\vec{B}$的余弦相似度如下：
  - $Cosine\ Similarity = \cos(\theta) = \frac{\vec{A}\cdot\vec{B}}{||A||\cdot||B||}$
- 欧氏距离
  - $d = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$
  - 距离越短，两个文本越相似
- **Jaccard相似度**
  - $EJ(A,B) = \frac{\vec{A}\cdot\vec{B}}{||A||^2+||B||^2-\vec{A}\cdot\vec{B}}$
  - 狭义定义
    - $J(A,B) = \frac{|A\cap B|}{|A\cup B|}$
  - 值越大，两个文本越相似

- 例题(注意多维向量的运算)

例11-1：根据上节的表2，计算华为和苹果的余弦相似度 和广义Jaccard相似度。

解：根据表2可知，令华为的词向量 A =[0.02, 0.93, 0.95, 0.01]，苹果的词向量 B =[0.96, 0.77, 0.85, 0.15]

$A \cdot B$ =0.02×0.96+0.93×0.77+0.95×0.85+0.01×0.15=1.5443

$\|A\| = \sqrt{0.02^2 + 0.93^2 + 0.95^2 + 0.01^2} = \sqrt{1.7679}$ =1.3296

$\|B\| = \sqrt{0.96^2 + 0.77^2 + 0.85^2 + 0.15^2} = \sqrt{2.2595}$ =1.5032

余弦相似度： $\cos\theta = \frac{A \cdot B}{\|A\| \|B\|}$ =0.7727

广义Jaccard相似度： $\mathrm{EJ}(A,B) = \frac{A \cdot B}{\|A\|^2 + \|B\|^2 - A \cdot B} = \frac{1.5443}{1.7679 + 2.2595 - 1.5443}$ =0.6219

例11-2：计算以下两个文本的Jaccard相似度，文本1 “我爱北京天安门”，文本2 “天安门雄伟壮阔让人不得不爱” （不考虑词频）。

解：文本1的集合 A ={我，爱，北，京，天，安，门}

文本2的集合 B ={天，安，门，雄，伟，壮，阔，让，人，不，得，爱}

$A \cap B$ ={爱，天，安，门}

$A \cup B$ ={我，爱，北，京，天，安，门，雄，伟，壮，阔，让，人，不，得}

Jaccard相似度： $\mathrm{J(A,B)} = \frac{|A \cap B|}{|A \cup B|} = \frac{4}{15}$ =0.2667

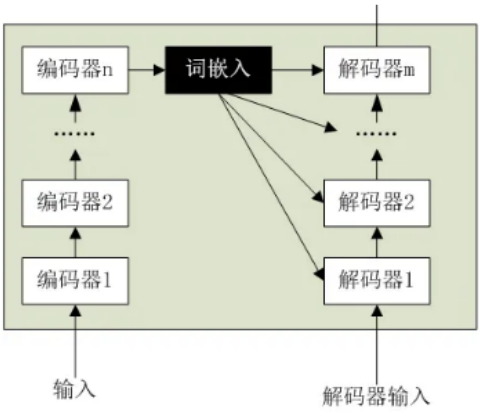
经典NLP模型

- 词袋模型(BoW)
  - 是自然语言处理和信息检索中的一种常用文本表示方法，忽略词语上下文的关系，只计算词语的出现频率和其他统计值
  - 不考虑次的次序（问题）
- Word2Vec
  - 是一种用于生成词嵌入的模型，它通过神经网络将词语映射到一个低维的连续向量空间中
  - 两种模型
    - **CBOW（Continuous Bag of Words）**：这种模式下，模型通过上下文词汇来预测中心词。它适合处理小数据集，并且训练速度较快
    - **Skip-Gram**：在这种模式下，模型通过中心词来预测其上下文词汇。这适合处理大数据集，并且能够生成更高质量的词向量

Transformer

- Transformer 是一种基于注意力机制的序列模型
- 与传统的RNN和CNN不同，Transformer仅使用自注意力机制来处理输入序列和输出序列，可以并行计算，极大提高了计算效率
- 两个技术特点
  - 序列到序列的编码器-解码器，自回归语言生成技术
  - 注意力机制

- 整体结构



- GPT是基于Transformer结构的预训练模型

编码器

编码器由多个相同的层（通常称为编码器层）堆叠而成，每层包含两个主要的子层：

1. **自注意力层（Self-Attention Layer）**：这一层允许模型在编码输入序列时考虑到序列中所有位置的依赖关系。自注意力机制通过计算输入序列中每个词与序列中其他词的相关性（注意力分数），然后根据这些分数对词进行加权求和，从而得到每个词的表示。
2. **前馈神经网络（Feed-Forward Neural Network）**：这是一个简单的全连接网络，它对自注意力层的输出进行进一步的非线性变换。

每个子层后面都跟着一个残差连接（Residual Connection），然后进行层归一化（Layer Normalization）

解码器

解码器同样由多个相同的层堆叠而成，每层包含三个主要的子层：

1. **掩码自注意力层（Masked Self-Attention Layer）**：这一层与编码器中的自注意力层类似，但增加了掩码（Masking）以防止模型在预测下一个词时看到未来的信息。这是通过在自注意力计算中对非法位置（即序列中尚未生成的部分）的注意力分数设置为负无穷大来实现的。
2. **编码器-解码器注意力层（Encoder-Decoder Attention Layer）**：这一层允许解码器在生成输出序列时考虑到编码器的输出。它通过计算解码器当前状态与编码器所有状态之间的注意力分数，然后将这些分数用于加权编码器的输出。

3. **前馈神经网络（Feed-Forward Neural Network）**：与编码器中的前馈网络相同，用于对注意力层的输出进行非线性变换。

解码器的每个子层后面同样跟着一个残差连接和层归一化。

多头自注意力机制

- Transformer 模型中的自注意力和编码器-解码器注意力都可以是多头的，这意味着它们可以并行地执行多个注意力机制，每个机制学习序列的不同表示子空间。

三种模型

- BERT
  - 仅采用编码器结而无解码器
  - 强项：语义理解，通过注意力机制，可同时考虑单词上下文信息
  - 弱项：自然语言生成能力。需要搭配其他语言模型或进行微调
  - 应用：阅读理解、完型题空...
- GPT
  - 仅采用解码器结构而无编码器
  - 强项：自然语言生成能力
  - 弱项：语义理解。需要搭配其他语言模型或进行微调
  - 应用：内容生成、问答系统、机器翻译、代码补全
- T5
  - 同时采用解码器和编码器
  - 强项：将NLP任务统一到一个框架下进行训练，能力比较平衡
  - 弱项：参数规模小，模型结构复杂，计算资源需求高，特定人物表现差
  - 应用：翻译、摘要等

AIGC

- AIGC 人工智能生成内容
- 是指利用AI技术，计算机自动生成的各种形式的内容
- 搜索引擎只能获取已经存在的内容或者知识

AIGC的局限与挑战

- AIGC的原创性：几乎为0
- AIGC的可解释性：缺乏，说的比想的快
- 语义理解：长度有限，包括时间跨度和文本长度
- 挑战
    1. AIGC生成速度远超人类，应该听谁的
    2. 如何识别AI生成的内容
    3. AI造假了怎么办
    4. AI错了怎么办
    5. 陷入困局怎么办

大语言模型（LLM）

- Large Language Model ，是特指用于执行NLP任务的语言模型

LLM中大的含义

- 训练数据大
- 参数规模大
- 耗资巨大

LLM的问题

- 模型不是越大越好，尤其是在专业领域
- 数据偏差与偏见依然存在
- 隐私和数据保护依然是关键
- 训练数据的伦理要求

生成式人工智能(GAI)

- 是特指生成全新内容的AI，其生成内容就是前面提到的AIGC
- GAI更侧重AI系统的功能特点

通用人工智能（AGI）

- 是指机器能够完成人类能够完成的任何智力任务的能力
- 旨在实现一般的认知能力，能够适应任何情况或目标，是人工智能研究的最终目标之一

训练要求

---

1. 领域无关
2. 任务无关

AI造假

- 不可预知性
- 隐蔽性
- 以假乱真

AI绘画

CLIP 模型

- CILP-视觉语言预训练模型
- 文本信息通过文本编码器进行编码，图像信息通过图像编码器进行编码
- 二者编码信息存入多模态的隐空间（数据的一种表示和存储方式）
- 将现实世界的实体编码为计算机算法可运算的数据格式

扩散模型

- Diffusion Models
- 前向过程（扩散过程），不断模糊
- 反向过程（去噪过程），不断恢复图像原始状态

多模型大语言模型(MLLM)

- 能感知不同模态的输入
- 能完成多模态任务
  - 图文跨模态检索
  - 视觉问答
  - 文生图
  - 图像描述及指称表达