Contents

安排

1) 深度学习基础
2) 神经网络基础类型
3) 计算机视觉与自然语言处理
4) 基于深度学习三维点云理解
5) 基于深度学习的蛋白质识别
6) 生成模型 (一)
7) 生成模型 (二)
8) 具身 (embodied) 人工智能简介

大作业 ppt 要求:

1) 任务与数据集介绍 (10 分); 任务难易程度 (5 分), 鼓励尝试具有挑战的前沿课题;
2) 输入, 神经网络结构, 输出, 损失函数, 训练过程介绍 (30 分);
3) 代码实现 (30 分) (完整的代码以及详细的 ReadMe 文件, 确保代码能够顺利跑通);
4) 成果, 性能展示 (性能高低不计入评分标准)(20 分);
5) 结果可视化, 新问题发现, 已有问题解决, 方法创新以及任何其他亮点 (5 分).

大作业代码要求:

1) dataset 数据集自己写
2) models 网络结构自己写
3) modules 可以调其他的包, 比如说 P4Transformer (点云处理)
4) train.py 就 train
5) utils.py 一些操作, 不属于上列的都塞到这里

I 深度学习基础

1. 发展历程

基于规则或基于学习
AlexNet(2012)

2. 神经网络基础知识

人工神经网络 + 训练学习策略
统计机器学习: 找规律, 有 pattern (拟合极小规模数据去验证网络的合理性)

$$\hat{y} \quad y = f(x;\theta)$$

2.1 网络类型

$f(\cdot;\theta)$:

1) 多层感知机 (MLP): >3 层, 单层就叫 fc 即可
2) 卷积神经网络 (CNN): 局部特征带有噪声
3) 循环神经网络 (RNN): 越后期的输入影响越大
4) Transformer: 自注意力

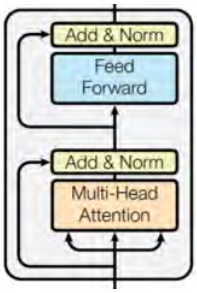

Figure 1: Transformer

5) 图网路 (GNN): 顶点与边都是特征

deep neural network (parameterized function): operation $\rightarrow$ architecture

2.2 数据

x:

1) 一维: 声音, 文本
2) 二维: 图像
3) 三维: 点云, 结构

2.3 任务

y:

1) 分类
2) 语义分割
3) 目标检测
4) 生成

2.4 训练

$\hat{y}$, e.g.

1) 监督学习: 数据有标签
2) 无监督学习: 数据无标签, 要模型寻找数据内在联系
3) 半监督学习: 部分数据有标签
4) 弱监督学习: 数据有标签, 但有一定噪声
5) 强化学习: 在线规划, 探索 (在未知之中) 与遵从 (现有知识) 寻找平衡, 有不可导的梯度时可以以此传递

3. 基础学习算法介绍

3.1 数据

3.1.1 图像 规则有序, 位置离散的二维网格

$$H \times W \times C \rightarrow 1 \times 1 \times C'$$

$C' \gg C$

3.1.2 文本 规则有序, 位置离散的列表

$$v^T W \to S$$

词典 $v \in \mathbb{R}^{5\times 1}$, $W \in \mathbb{R}^{5\times 128}$, $S \in \mathbb{R}^{5\times 128}$

3.1.3 点云 (point cloud or point set) 不规则无序的三维点集合 $P \in \mathbb{R}^{N\times 3}$, 点位置连续但点不连续

3.1.4 点云视频 点集合列表, 三维点 (不规则, 无序, 位置连续)+ 一维时间 (规则有序, 位置离散)

3.1.5 蛋白质结构 点列表, 三维结构 (不规则, 无序, 位置连续)+ 一维结构 (规则有序, 位置离散)

1) 一级
2) 二级
3) 三级
4) 四级

4. 大作业背景介绍

Transformer 找相关区域, 然后对其进行编码找特征. 使用自注意力追踪.

三维动作点云需要时空解耦

II 神经网络基础类型

1. 多层感知机 (MLP)

输入: vector $v \in R^{1\times C}$

$$\begin{aligned} o_1 &= ReLU(v \cdot W_1) \quad W_1 \in \mathbb{R}^{C\times C'} \\ o_2 &= ReLU(o_1 \cdot W_1) \quad W_2 \in \mathbb{R}^{C'\times C''} \\ o_3 &= ReLU(o_2 \cdot W_1) \quad W_3 \in \mathbb{R}^{C''\times C'''} \end{aligned}$$

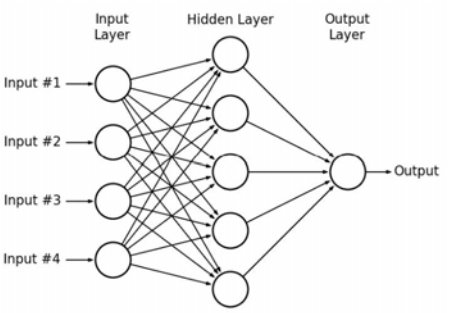

Figure 2: MLP

非线性函数 e.g.

1) sigmoid
2) tanh
3) ReLU
4) LeakyReLU

学习的根本是收集收益大的 activation, 将重要的信号汇聚到一起. 前两者限制了重要信号的大小, 产生了损失.

2. 卷积神经网络 (CNN)

三维的叫 tensor $\mathbf{I}$, 二维的叫矩阵 W, 一维的叫向量 $\mathbf{v}$?

以当前像素为中心, 在大小为 $[h, w]$ 的局部区域内, 每个方向和距离 (i, j) 安排一参数 $W^{(i,j)}$

$$\mathbf{I}'^{(x,y)} = \sum_{i=-\lfloor h/2 \rfloor}^{\lfloor h/2 \rfloor} \sum_{j=-\lfloor w/2 \rfloor}^{\lfloor w/2 \rfloor} W^{(i,j)} \cdot \mathbf{I}^{(x+i,y+j)} + b^{(i,j)}$$

$\mathbf{I} \in \mathbb{R}^{C\times H\times W}$, $\mathbf{I}^{(x,y)} \in R^C$, $W \in \mathbb{R}^{h\times w\times C'\times C}$, $W^{i,j} \in \mathbb{R}^{C'\times C}$.

CNN 可以感受到方向和距离.

1) Padding (填充), 一般用 0 填充.
2) Stride: 走的距离, 控制下采样的大小
3) Dilation (膨胀): 更大的卷积核, 更大的感受野, 但参数增多少.

4) Pooling (池化): Max or Average, 可以用卷积取代

e.g. AlexNet

3. 循环神经网络 (RNN)

$$h_t = \tanh(W_1 \cdot x_t + W_2 \cdot h_{t-1} + b)$$
$$o_t = h_t$$

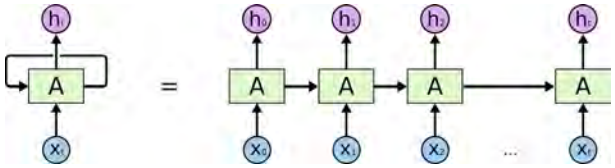

Figure 3: RNN

变体: LSTM, GRU

3.1 长短记忆网络 (LSTM)

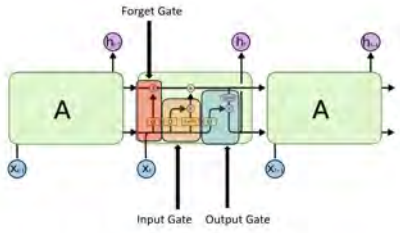

Figure 4: LSTM

3.2 Gated Recurrent Unit (GRU)

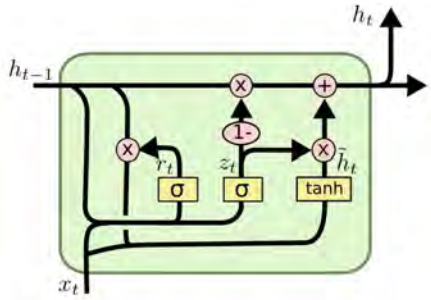

Figure 5: GRU

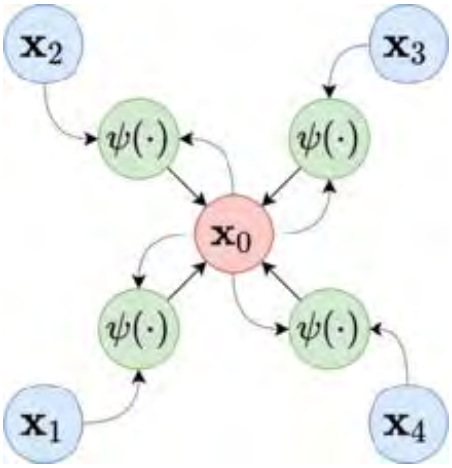

Figure 6: GNN

4. 图神经网络 (GNN)

$$h_u = \phi\left(x_u, \bigoplus_{v \in N_u} \psi(x_u, x_v, e_{uv})\right)$$

x_u 为中心点, x_v 为邻居点, e_{uv} 为边的信息, 是 vector.

5. Transformer

自注意力 (自适应计算感受野)

Image:	$I \in \mathbb{R}^{C \times N}$ $(N = H \times W)$
绝对位置编码:	$I^{(x,y)} = I^{(x,y)} + Emb(x, y)$
Query:	$Q = W_q \cdot I \in \mathbb{R}^{C' \times N}$, $W_q \in \mathbb{R}^{C' \times C}$
Key:	$K = W_k \cdot I \in \mathbb{R}^{C' \times N}$, $W_k \in \mathbb{R}^{C' \times C}$
Scaled Dot Product:	$P = \frac{Q^T \cdot K}{\sqrt{C'}} \in \mathbb{R}^{N \times N}$
Attention:	$A = softmax(P) \in \mathbb{R}^{N \times N}$
Value:	$V = W_v \cdot I \in \mathbb{R}^{C' \times N}$, $W_v \in \mathbb{R}^{C' \times C}$

Value 没有对方向, 距离信息的编码

$$F_a = \sum_{b=1}^{N} A_{ab} \times V_b$$

V_b 是 patch b 的表征, A_{ab} 是 patch a 与 patch b 直接的相关性. 最终求和得以 patch a 为中心的相关区域表征.

6. 总结

6.1 卷积与 Transformer 的统一

$$I_a = \sum_{b=1}^{N} A_{ab} \times W^{(b-a)} \times I_b$$

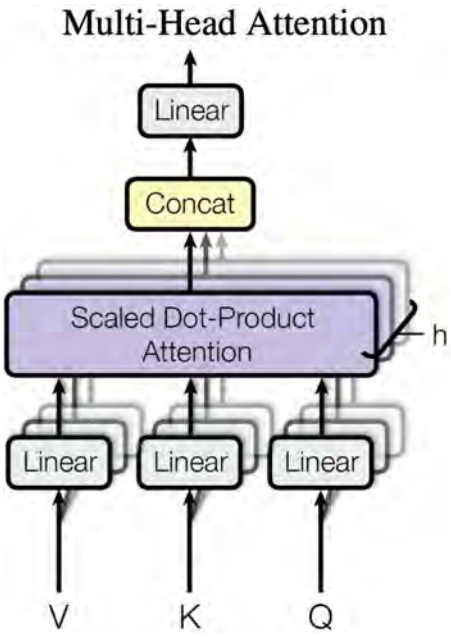

Figure 7: Multi-Head Attention

6.2 相对位置与绝对位置

绝对位置有效是因为数据足够多, 可以提供清晰的拟合对象

6.3 Transformer 模板

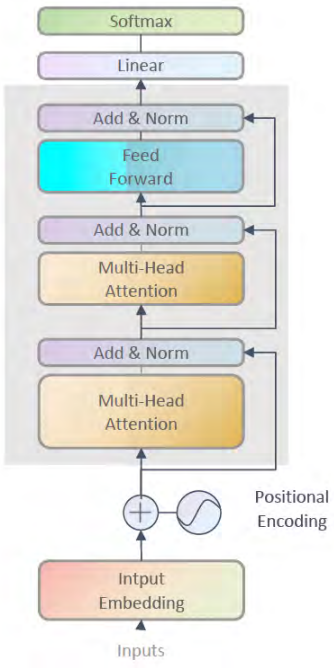

Figure 8: Transformer

III 计算机视觉与自然语言处理

1. 生成: 图像

1) Variational AutoEncoder, VAE (变分自动编码器)

2) Generative Adversarial Network, GAN (生成对抗网络)

3) Diffusion Model

1.1 数据分布

将一张 $H \times W \times 3$ 的图像看作 $3HW$ 维中的一点. 采样来自现实世界, 并不均匀分布到此空间中.

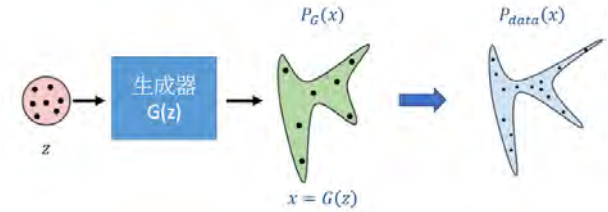

Figure 9: 数据分布看生成

从数据分布的角度看生成

$$G^* = \arg\min_G Div(P_G, P_{data})$$

1.2 VAE

AutoEncoder (自动编码器): 不具备生成功能, 仅作数据编解码

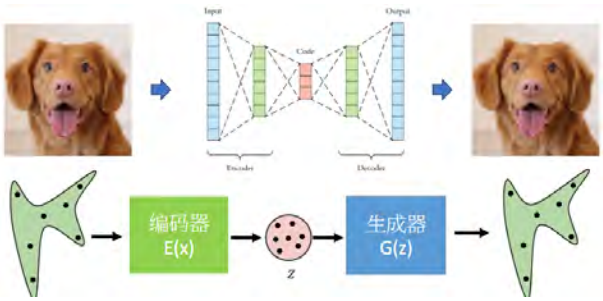

Figure 10: AutoEncoder

VAE: 加个噪声, 让一个点的特征能覆盖更多的编码空间, 这样采样编码空间时有更大机会采样出合理的特征.

1.3 GAN

两个网络, 对抗训练

1.4 Diffusion

"近似" 从高维空间采样

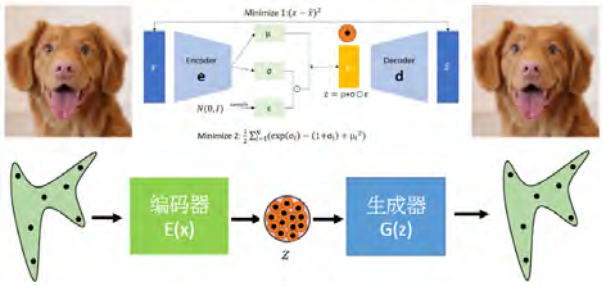

Figure 11: VAE

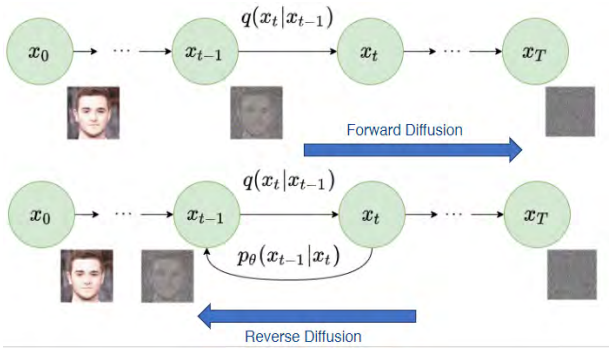

Figure 12: Diffusion

2. 其他任务

1) 机器翻译: 基于 LSTM 的机器翻译
2) 分类: Vision Transformer

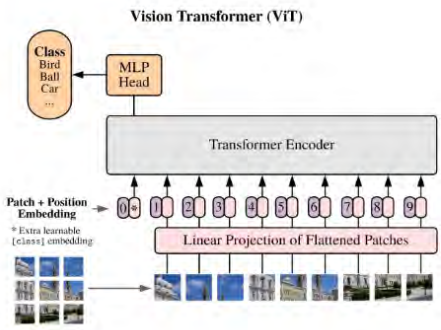

Figure 13: Vision Transformer

3) 语义分割: UNet
4) 无监督领域自适应: 行人重识别 (person re-ID) 从容易到复杂
5) 对比学习: CLIP (Contrastive Language-Image Pre-training)
6) 自监督学习: MAE(Masked Autoencoder)
7) 元学习: Meta-learning describes machine learning algorithms that acquire the knowledge and understanding from the outcome of other machine learning algorithms.

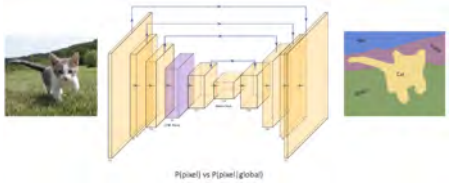

Figure 14: UNet

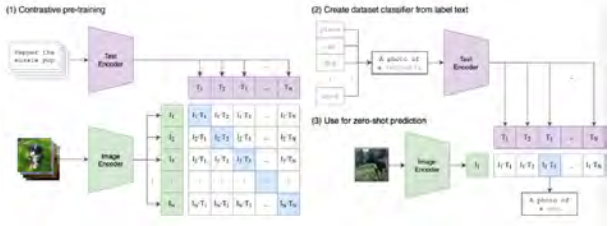

Figure 15: CLIP

3. 点云

特征越深, 等价于分得越细

$$F'^{(x,y,z)} = \max_{\|(\delta_x,\delta_y,\delta_z)\| \le r} MLP(F^{(x+\delta_x,y+\delta_y,z+\delta_z)}, (\delta_x, \delta_y, \delta_z))$$

IV 三维点云理解

1. 静态点云

1.1 任务

分类 + 分割

1.2 挑战

对无序的不变性

1.3 建模

1.3.1 PointNet 对坐标求特征后过 maxpooling.
有个变换矩阵, 但这个矩阵是全局的, 不可能有用.

1.3.2 PointNet++ (° ∀ 。)

1.3.3 DGCNN 动态图卷积神经网络—处理图数据

2. 点云视频

2.1 任务

动作识别 + 语义分割

2.2 建模

2.2.1 PSTNet++ 帧级 Attention, 隐含假设所有点在所有帧中都出现了

2.2.2 PST-Transformer 视频级 Attention

V 生成模型

输入: 一堆向量, 表示特征

输出: 图片

什么是好的生成模型? 给定训练集, 产生与训练集同分布的样本.

$$G^* = \arg\min_G Div(P_G, P_{data})$$

1. Variational AutoEncoder(VAE)

1.1 AutoEncoder(AE)

图像有冗余, 所以可以降维.

$$loss = \|x - \hat{x}\|_2$$

Why VAE: 填补训练中没有的数据.

1.2 Denoising AutoEncoder (DAE)

输入: 有噪音的图像
输出: 图像
但输入输出仍是一一对应的关系.

1.3 VAE

Encoder 输出 μ, σ, 即输出一个预测输入的正态分布.

重参数化, 从标准正态分布采样, 然后将标准正态分布变为目标正态分布, 以此就可以求得导数.

$$N(0,1) \to N(\mu, \sigma)$$

loss 为 KL 散度, 目的是想让 Encoder 输出的预测更接近标准正态分布 ($\sigma \to 1$), 因为纯 Encoder 会更倾向于 $\sigma \to 0$ (退化为 AE).

$$loss =$$

1.4 数学推导

KL 散度

$$D_{KL}(P||Q) = P \log\left(\frac{P}{Q}\right)$$

衡量两个分布之间的距离.

极大似然估计, $\int_z q(z|x)dz = 1$, $P(z,x) =$

$$P(z|x)P(x) = P(x|z)P(z)$$

$$P(x) = \int_z P(z)P(x|z)dz$$

$$L = \sum_x \log P(x) \text{ 尽量大}$$

$$\begin{aligned}
\log P(x) &= \int_z q(z|x) \log P(x) dz \\
&= \int_z q(z|x) \log \left(\frac{P(z,x)}{P(z|x)} \right) dz \\
&= \int_z q(z|x) \log \left(\frac{P(z,x)}{q(z|x)} \frac{q(z|x)}{P(z|x)} \right) dz \\
&= \int_z q(z|x) \log \left(\frac{P(z,x)}{q(z|x)} \right) dz \\
&+ \underbrace{\int_z q(z|x) \log \left(\frac{q(z|x)}{P(z|x)} \right) dz}_{D_{KL}(q(z|x)||P(z|x))} \\
&\geq \int_z q(z|x) \log \left(\frac{P(z|x)P(z)}{q(z|x)} \right) dz
\end{aligned}$$

$P(z)$ 为正态分布, $x|z \sim N(\mu(z), \sigma(z))$, $\mu(z), \sigma(z)$ 为待估计的参数.

$$\begin{aligned}
L_b &= \int_z q(z|x) \log \left(\frac{P(z|x)P(z)}{q(z|x)} \right) dz \\
&= \underbrace{\int_z q(z|x) \log \left(\frac{P(z)}{q(z|x)} \right) dz}_{-D_{KL}(q(z|x)||P(z))} + \underbrace{\int_z q(z|x) \log P(x|z) dz}_{\|x-\tilde{x}\|_2} \\
&\leq \int_z q(z|x) \log P(x|z) dz
\end{aligned}$$

当 $D_{KL}(q(z|x)||P(z)) = 0$ 时取等号. 最大化其下界, 近似最大化其最大值

1.5 总结

2. GAN

输入: 取自特定分布的随机噪声

输出: 采样自训练样本分布图像

即从特定分布到图像的映射

生成器与判别器 (生成器与 VAE 的 decoder 其实是一样的)

2.1 训练

使用 minmax 方式联合训练

$$\min_{\theta_g} \max_{\theta_d} [\mathbb{E}]$$

先优化 θ_d 再优化 θ_g.

交替完成

1) G
2) G

但这个函数不是很好

小寄巧: 取最大值

$$\max E - \log(D(G(z)))$$

2.2 证明

极大化似然估计等价于最小化 KL 散度

$$\theta^* =$$

$$JSD(P||Q) = \frac{1}{2} D_{KL}(P||M) + \frac{1}{2} D_{KL}(Q||M)$$

$$M = \frac{1}{2}(P+Q)$$

3. Diffusion model

VI 具身智能

1. 定义

1.1 学派

1.1.1 符号主义 专家系统, 定理证明机

1.1.2 链接主义 神经网络

1.2 离身智能局限

1) 被动
2) 消耗
3) 遗忘

1.3 具身智能

1) 感知
2) 推理
3) 行动

2. 应用