

NLP

<https://github.com/huggingface/transformers>

transformers

transformer 提供了数千种预训练模型来执行文本相关任务，如分类、信息提取、问答、摘要、翻译、文本生成等。github 中提供了相应的 api，可以在给定文本上快速下载和使用这些预训练模型，在自己的数据集上对它们进行微调。

人体识别

<https://github.com/CMU-Perceptual-Computing-Lab/openpose>

OpenPose

OpenPose 人体姿态识别项目是美国卡耐基梅隆大学（CMU）基于卷积神经网络和监督学习并以 caffe 为框架开发的开源库。可以实现人体动作、面部表情、手指运动等姿态估计，在单个图像上联合检测人体，手，面部和脚步等共 135 个关键点。适用于单人和多人，具有极好的鲁棒性。

<https://github.com/MVIG-SJTU/AlphaPose>

AlphaPose

AlphaPose 是一款精准的多人姿态评估工具。AlphaPose 既可以在图片、视频中进行姿态估计，也能在画面中对动作进行追踪。它的输出形式非常广泛，包括 PNG、JPG 和 AVI 等具有关键点的图片形式，也有 JSON 格式的输出，这一特点也使其成为众多应用受欢迎的工具。目前，这一工具支持 TensorFlow 和 PyTorch 两种实现。

https://github.com/akanazawa/human_dynamics

视频三维人体姿态的动态学习

利用行动中的人的图片，很容易地猜测这个人在过去和未来的 3D 运动。这是因为我们有一个有效的 3D 人体动力学的理论模型，这个模型是通过观察运动中人类的视觉序列所获得的。这项工作提出了一个框架，通过简单但有效的图像特征时间编码，可以类似地从视频中学习人类 3D 动态的表示。在测试时，通过视频，学习的时间表示可以恢复平滑的 3D 网格预测。我们的模型可以从单个图像中恢复当前的 3D 网格以及 3D 过去和未来的运动。

https://github.com/ageitgey/face_recognition

人脸识别：

本项目是世界上最简洁的人脸识别库，你可以使用 Python 和命令行工具提取、识别、操作人脸。使用世界上最简单的面部识别库从 Python 或命令行识别和操作面部。使用 dlib 的最先进的面部识别技术和深度学习技术构建而成。该模型在 Wild 基准数据集中的 Labeled Faces 上的准确性为 99.38%。

<https://github.com/google-research/planet>

PlaNet

在未知环境中利用规划(例如仅将像素作为输入来控制机器人)，智能体必须从经验中学习规则或动态。由于这种动态模型原则上允许更高的效率和自然的多任务学习，因此创建足够精确的模型以成功地进行规划是 RL 的长期目标。

PlaNet 学习了给定图像输入的动态模型 (dynamics model), 并有效地利用该模型进行规划, 以收集新的经验。与以前的图像规划方法不同, 我们依赖于隐藏状态或潜在状态的紧凑序列。这被称为 latent dynamics model: 我们不是直接从一个图像到下一个图像地预测, 而是预测未来的潜在状态。然后从相应的潜在状态生成每一步的图像和奖励。通过这种方式压缩图像, agent 可以自动学习更抽象的表示, 例如对象的位置和速度, 这样就可以更容易地向前预测, 而不需要沿途生成图像。

量化金融

<https://github.com/thuquant/awesome-quant>

一份中文量化相关资源索引。其中有数据源, 数据库, 量化交易平台, 策略和回测方式等等。

如数据源包括以下部分:

- [TuShare](#) - 中文财经数据接口包
- [Quandl](#) - 国际金融和经济数据
- [Wind 资讯-经济数据库](#) - 收费
- [锐思数据 - 首页](#) - 收费
- [国泰安数据服务中心](#) - 收费
- [恒生 API](#) - 收费
- [Bloomberg API](#) - 收费
- [数库金融数据和深度分析 API 服务](#) - 收费
- [Historical Data Sources](#) - 一个数据源索引

<https://github.com/wilsonfreitas/awesome-quant>

一份量化资源合集。其包含如下部分:

- Numerical Libraries & Data Structures
- Financial Instruments and Pricing
- Indicators
- Trading & Backtesting
- Risk Analysis
- Factor Analysis
- Time Series
- Calendars
- Data Sources
- Excel Integration
- Visualization

生物基因/细胞/医学

<https://github.com/topics/genome-sequencing>

一共有 36 个公共库与基因序列相关。均位于该话题下, 如:

jcvi:Python 库, 以方便基因组组装、注释和比较基因。

pyGeno:个性化基因和蛋白质。

Viral-ngs:病毒基因分析方法

<https://github.com/tariqdaouda/pyGeno>

pyGeno:

pyGeno 是一个 python 软件包，主要用于基因学与蛋白质学相关的精密医学应用。它将来自 Ensembl 的参考序列和注释，来自 dbSNP 数据库的基因组多态性以及来自下一代测序的数据整合到一个易于使用，内存高效且快速的框架中，从而使用户可以轻松地探索受试者的基因组和蛋白质组。

pyGeno 通过操作诸如基因组，染色体，基因，转录本，蛋白质和外显子之类的熟悉对象，使用户可以使用 Python 的全部功能来探索参考和特定于患者的基因组和蛋白质组。

<https://github.com/InsightSoftwareConsortium/ITKColorNormalization>

ITK

ITK 包用 C++ 编写，提供 python 接口。在对细胞部分进行染色时，一般会将细胞核染成蓝紫色，细胞外基质和细胞质染成粉红色，但是染色的确切颜色却会因图像而异。该算法包通过使用参考图像的配色方案对细胞图像进行重新着色。

具体使用算法请参考：Fast GPU-Enabled Color Normalization for Digital Pathology 2019 Goutham Ramakrishnan, Deepak Anand, Amit Sethi

CellProfiler:

如今，生物图像数据越来越庞大，利用眼睛观察是乏味且主观的。为此产生了用软件对生物图像进行分析的需求。

CellProfiler 是免费的开源软件，旨在让生物学家无需计算机视觉或编程方面的培训，即可自动定量测量数千张图像的表型。研究人员可以下载在线示例的工作流程或从头开始创建自己的工作流。这些流程易于保存，重复使用和共享，有助于提高科学的可重复性。

CellProfiler 应用例子 <https://cellprofiler.org/examples>

CellProfiler 链接: <https://cellprofiler.org/>

CellProfiler 开源链接: <https://github.com/CellProfiler>

CellProfiler Analyst 允许交互式探索和分析数据，特别是来自基于图像的高通量实验。其中包括一个受监督的机器学习系统，可以对其进行训练以识别复杂和微妙的表型，从而自动计数检查百万个细胞。

CellProfiler Analyst 开源链接: <https://github.com/CellProfiler/CellProfiler-Analyst>

SCANPY (large-scale single-cell gene expression data analysis) :

Scanpy 是一个基于 Python 分析单细胞数据的软件包，用于分析与 [anndata](#) 联合构建的单细胞基因表达数据，内容包括预处理，可视化，聚类，拟时序分析和差异表达分析等

Scanpy 开源链接: <https://github.com/theislab/scanpy>

Scanpy 论文链接: <https://genomebiology.biomedcentral.com/articles/10.1186/s13059-017-1382-0>

语音:

https://github.com/Uber/speech_recognition

用于执行语音识别的库，支持在线和离线多个引擎和 API。

其可完成的功能有：

识别来自麦克风的语音输入；转录音频文件；校准识别器能量阈值以获取环境噪声水平等等

<https://github.com/mozilla/DeepSpeech>

DeepSpeech:

DeepSpeech 是一个开源语音转文本引擎。它是一个基于百度深度语音技术训练出来的模型。DeepSpeech 使用 Google 的 TensorFlow 来简化实施过程。

<https://github.com/ibab/tensorflow-wavenet>

WaveNet

Wavenet 模型是一种序列生成模型，可以用于语音生成建模。在语音合成的声学模型建模中，Wavenet 可以直接学习到采样值序列的映射，因此具有很好的合成效果。目前 wavenet 在语音合成声学模型建模，vocoder 方面都有应用，在语音合成领域有很大的潜力。

虚拟助理

<https://github.com/topics/virtual-assistant>

Leon:

Leon 是一个开源可以部署在服务器上的个人助理，你可以和他说话，他可以和你说话。您也可以发短信给他，他也可以发短信给您。如果您愿意，Leon 可以通过离线与您联系以保护您的隐私

如果你是一名开发人员（或者不是），你可能希望构建许多在日常生活中可能会有帮助的东西。Leon 可以帮助你构建它的包 / 模块（技能）结构，而不是为每个想法构建专门的项目。使用这种通用结构，每个人都可以创建自己的模块并与其他人共享。因此，只有一个核心模块来管理它们。

Leon 链接：<https://github.com/leon-ai/leon>

音乐播放器

<https://github.com/kannan-anbu/python-music-player>

python-music-player

python-music-player 是一个开源的 python 音乐播放软件，实现了经典的音乐播放软件具有的常用功能。结合开源代码可以很方便的学习使用 python 语言进行软件开发的能力。

网络爬虫

<https://github.com/facert/awesome-spider>

此项目是一个 python 网络爬虫的整合，整合了 github 上大量的常用中文、英文网站爬虫项目。但是需要注意的是，爬虫具有时效性。网站运维人员在发现有大量非人类流量影响网络带宽时，会更新反爬虫机制，使得原有爬虫失效，这也要求如果遇到爬虫没法直接运行时，要适当更改爬虫逻辑。

https://github.com/RecluseXU/learning_spider

对于想要系统的学习 python 网络爬虫、反爬虫与反反爬虫的技术, learning_spider 是个很好的开源学习项目。

此项目主要分为三个部分

① 爬虫案例

对一些网站的数据进行爬取, 标记出 难度和要点

② 基础练习网站 <http://learnsnider.evilrecluse.top/>

根据反反爬经验, 尝试去接入、编写各类爬虫障碍, 顺便研究前端/后端/服务器知识

③ 辅助工具/脚本制造

尝试制作各种工具/脚本

物理学:

<https://github.com/barbagroup/CFDPython>

CFD Python:

Navier-Stokes 方程 (N-S 方程) 是流体力学中的基本方程, 然而 N-S 方程没有解析解, 实际运用中需要用计算机模拟的方法得出数值解, 由此发展出计算流体力学这门学科。

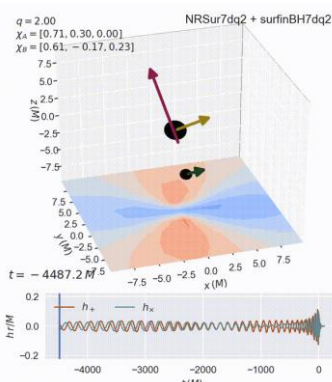
CFD Python, 又称 Navier-Stokes 的 12 个步骤, 是一个实用的模块, 用于通过对描述流体流动物理原理的基本偏微分方程的解进行编码来学习计算流体力学 (CFD) 的基础。该项目是 Lorena Barba 教授于 2009 年至 2013 年在波士顿大学机械工程系教授的课程的一部分。

<https://github.com/achael/eht-imaging>

eht-imaging

这是一个应用在射电干涉测量方面的成像、分析、仿真软件, 可以仿真和操纵 VLBI 数据并使用正则化的最大似然方法生成图像的一个 Python 模块。eht-imaging 提供了以下工具: 加载图像和数据, 从真实的 uv 轨迹生成模拟数据, 校准, 检查和绘制数据以及从数据集中生成图像 使用各种数据项和正则化函数进行各种极化处理。

<https://vijayvarma392.github.io/binaryBHexp/>



该算法包模拟两个黑洞在旋转过程中如何对外辐射引力波, 以及它们合并的全过程。这是迄今位置最精确的模拟黑洞合并模型, 而且模拟的时间很短。我们可以将这项技术用于模拟黑洞合并过程, 帮助引力波干涉天文台能发现更多的引力波, 或是验证广义相对论, 或者找到它的缺陷。

游戏应用

<https://pypi.org/project/pymunk/>

Pymunk:

Pymunk 是一个易于使用的 2D 物理库，可以在需要 Python 的 2D 刚体物理时使用。利用 Pymunk 可以很方便的模拟一些复杂的物理运动，比如碰撞、旋转、滑轮、绳子、轮子等。为一些简单的 2D 小游戏的开发提供了便捷的工具。

<https://github.com/grantjenks/free-python-games>

Free Python Games:

Free Python Games 一个包含了很多小游戏的开源平台，这些游戏是用简单的 Python 代码编写的，旨在进行实验和更改。包含的游戏有吃豆人，井字棋、Flappy 与一些简化的街机游戏等。

中国科学技术大学 面向交叉学科的python程序设计与跨学科实践 计算机学院

最近几年，深度学习各个领域技术发展：

1.自然语言处理：

1.1 Bert：

BERT 是 2018 年 10 月由 Google AI 研究院提出的一种预训练模型。

BERT 的全称是 Bidirectional Encoder Representation from Transformers。BERT 在机器阅读理解顶级水平测试 SQuAD1.1 中表现出惊人的成绩：全部两个衡量指标上全面超越人类，并且在 11 种不同 NLP 测试中创出 SOTA 表现，包括将 GLUE 基准推高至 80.4% (绝对改进 7.6%)，MultiNLI 准确度达到 86.7% (绝对改进 5.6%)，成为 NLP 发展史上的里程碑式的模型成就。

Bert 架构：

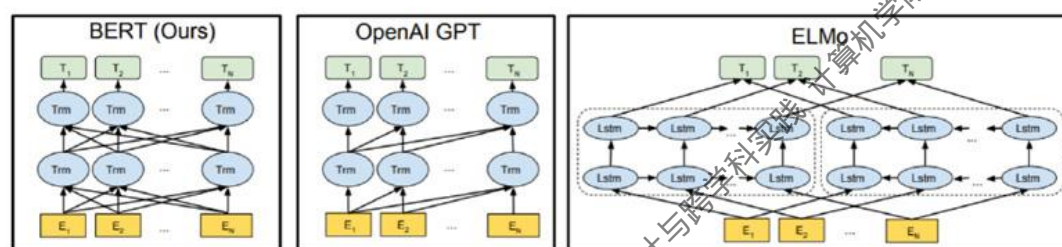


Figure 3: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTMs to generate features for downstream tasks. Among the three, only BERT representations are jointly conditioned on both left and right context in all layers. In addition to the architecture differences, BERT and OpenAI GPT are fine-tuning approaches, while ELMo is a feature-based approach.

如图所示，最左边的就是 BERT 的架构图，可以很清楚的看到 BERT 采用了 Transformer Encoder block 进行连接，因为是一个典型的双向编码模型。

Github：

<https://github.com/google-research/bert>

<https://github.com/kyone/bert4keras>

1.2 XLNet：

BERT 这样基于去噪自编码器的预训练模型可以很好地建模双向语境信息，性能优于基于自回归语言模型的预训练方法。然而，由于需要 mask 一部分输入，BERT 忽略了 mask 位置之间的依赖关系，因此出现预训练和微调效果的差异。

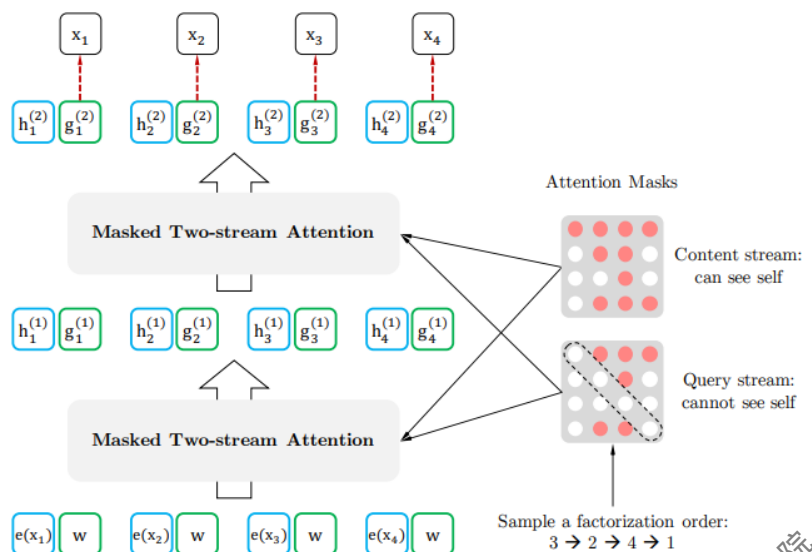
基于这些优缺点，该研究提出了一种泛化的自回归预训练模型 XLNet。

XLNet 可以：

- 1) 通过最大化所有可能的因式分解顺序的对数似然，学习双向语境信息；
- 2) 用自回归本身的特点克服 BERT 的缺点。

此外，XLNet 还融合了当前最优自回归模型 Transformer-XL 的思路。

最终，XLNet 在 20 个任务上超过了 BERT 的表现，并在 18 个任务上取得了当前最佳效果 (state-of-the-art)，包括机器问答、自然语言推断、情感分析和文档排序。



Github:

<https://github.com/zihangdai/xlnet>

<https://github.com/ymcui/Chinese-XLNet>

1.3 ALBert:

Bert 参数过大，训练过慢。

ALBert 主要通过两个参数削减技术克服预训练模型扩展的障碍：

1、Factorized embedding parameterization (embedding 参数因式分解)

将大的词嵌入矩阵分解为两个小的矩阵，从而将隐藏层与词典的大小关系分割开来，两者不再有直接关系，使得隐藏层的节点数扩展不再受限制。

2、Cross-layer parameter sharing (跨层参数共享)

这样就可以避免参数量随着网络的深度增加而增加。

结果：

1. 比 BERT-large 模型缩小了 18x 的参数

2. 训练加速 1.7x

3. 在 GLUE, RACE 和 SQUAD 得到 SOTA 结果：

RACE: 89.4% [提升 45.3%]

GLUE Benchmark: 89.4

SQUAD2.0 f1 score: 92.2

Github: <https://github.com/google-research/albert>

1.4 FastQAExt

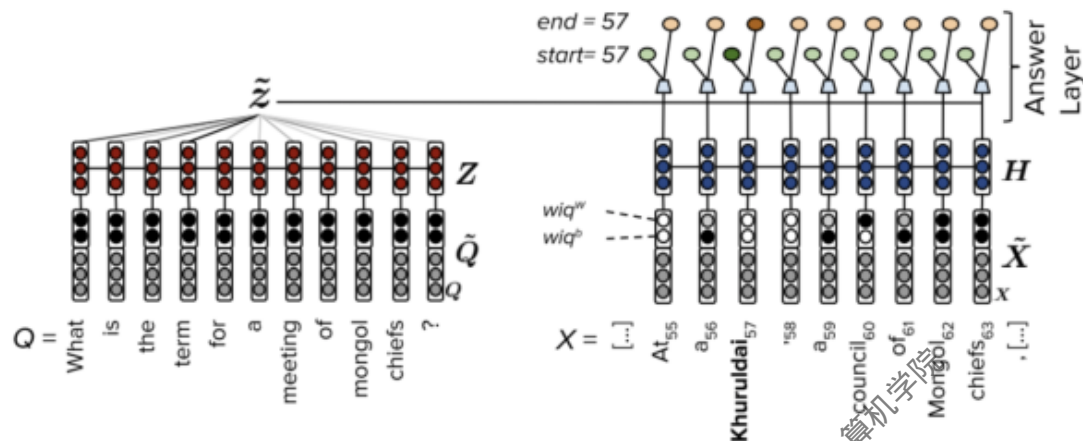
FastQAExt 使用了一种轻量级的架构，其 Embed 层除了 word 和 char 两种 embedding 作为输入以外，还额外使用了两个特征：

1. binary 特征表示原文词是否出现在问题中；

2. weighted 特征表示原文词对于问题中所有词的相似度。并且这两个特征同样用在了问题词上。

在 Interaction 层，FastQAExt 使用了两种轻量级的信息 fusion 策略：1. Intra-Fusion，即每个原文词和其他原文词计算相似度，并汇总得到原文总表示，接着将该原文词和对应原文总表示输入 Highway Networks 进行聚合，聚合后原文词表示进一步和上下文词表示进行类似的聚合；2. Inter-Fusion，即对每个原文词计算和问题词的相似度，并汇总得到问题总

表示, 接着将该原文词和对应问题总表示输入 Highway Networks 进行聚合, 得到 query-aware 原文表示。此外, 在 Answer 层, FastQAExt 首先计算了一个问题的总表示, 接着将 query-aware 原文表示和问题总表示共同输入两个前馈网络产生答案开始和结束位置概率。在确定答案范围时, FastQAExt 使用了 Beam-search。

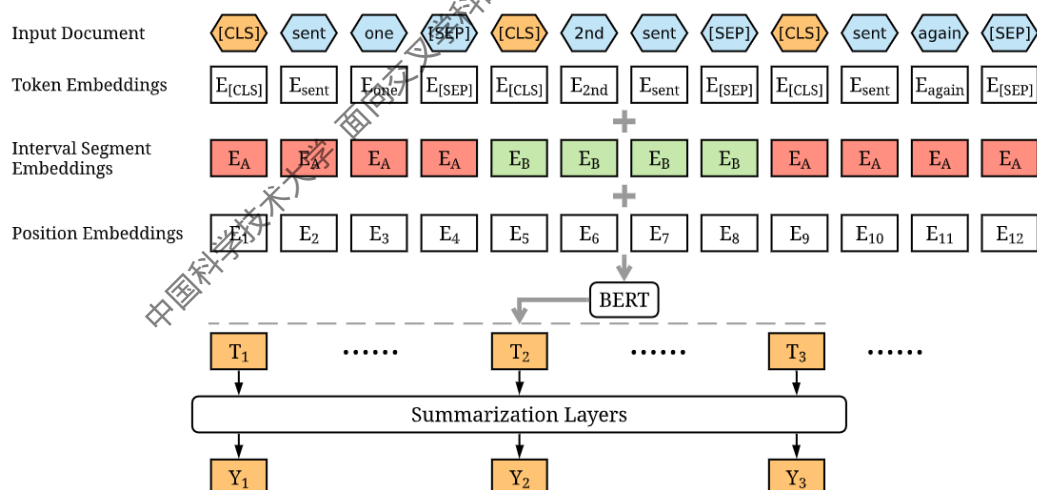


Github: https://github.com/BrambleXu/keras_fastqa

1.5 Fine-tune BERT for Extractive Summarization

首先介绍一下模型的结构, 原始的 BERT 的输出是针对 Token 而不是句子的, 而且原始 BERT 的输入只有两个句子, 并不适合文本摘要。

因此首先作者对 BERT 的结构做了一些更改, 让他变得更适合文本摘要这个任务, 作者的更改可以在下图体现到:



作者使用[CLS]和[SEP]区分每一个句子, 在原始的 BERT 中[CLS]表示一整个句子或者句子对的内容, 在这里作者修改了模型结构, 使用[CLS]来区分每一个句子作者对每一个句子增加了 segment embedding, segment embedding 由句子的奇偶顺序决定, 例如对于句子[sen1, sen2, sen3, sen4, sen5]他们的 segment embedding 就是[EA, EB, EA, EB, EA]。

Summarization layer 得到了句子的向量之后, 接下来要做的工作就是判断这句话是否要组成文本的摘要。这就是一个二分类的工作了, 作者尝试了三种 summarization layer, 分别是:

1. 传统的全连接层
2. Inter-sentence Transdormer, 初始位置的句子向量为 position embedding, 之后每一个位置的输入都是由上一个位置的输入经过多头 Attention 层, layer norm 和全连接层之后的输出结果。最后的输出依然是一个二分类。
3. RNN 层, 这里是在 BERT 之后接了 LSTM 层, LSTM 是非常适合 NLP 任务的一种结构, 当然最后输出也是一个二分类的结果。

Github : <https://github.com/nlpyang/BertSum>

1.6 multilingual Random Aligned Substitution Pre-training

核心理想就是打造“机器翻译界的 BERT”, 通过预训练技术再在具体语种上微调即可达到领先的翻译效果, 其在 32 个语种上预训练出的统一模型在 47 个翻译测试集上取得了全面显著的提升。

mRASP 提出了全新的思路, 利用多个语言已经积累的大量双语平行语料, 合并起来联合训练一个统一的模型, 之后再基于此微调, 让预训练和微调目标尽可能接近, 这样才能更大发挥预训练模型作用。

Github: <https://github.com/linzehui/mRASP>

2 自动驾驶:

如今自动驾驶有很多开源的框架, 方便我们使用学习。

2.1 AutoWare:

其最早是由名古屋大学研究小组在加藤伸平教授的领导下于 2015 年 8 月正式发布。2015 年 12 月下旬, 加藤伸平教授创立了 Tier IV, 以维护 Autoware 并将其应用于真正的自动驾驶汽车。随着时间的流逝, Autoware 已成为公认的开源项目。Autoware 也是世界上第一个用于自动驾驶技术的“多合一”开源软件。

它包含如下模块:

1. 定位 (Localization): 通过结合 GNSS 和 IMU 传感器的 3D 地图和 SLAM 算法来实现定位。
2. 检测 (Detection): 使用具有传感器融合算法和深度神经网络的摄像机以及 LiDAR。
3. 预测和规划 (Prediction and Planning): 基于概率机器人技术和基于规则的系统, 部分还使用深度神经网络。
4. 控制 (Control): Autoware 向车辆输出的是速度和角速度的扭曲量。尽管控制量的主要部分通常位于车辆的线控控制器中, 但这些都是 Control 的一部分。

Gitlab: <https://gitlab.com/autowarefoundation/autoware.ai/autoware/wikis/Source-Build>

2.2 Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving

3D 物体检测是自动驾驶中的基本任务。如果从精确但昂贵的 LiDAR 技术获得 3D 输入数据, 则最近的技术具有高度准确的检测率。迄今为止, 基于较便宜的单目或立体图像数据的方法导致精度显著降低 - 这种差距通常归因于基于图像的深度估计不良。然而, 在本文中, 我们认为数据表示 (而不是其质量) 占据了差异的大部分。考虑到卷积神经网络的内部工作原理, 我们建议将基于图像的深度图转换为伪 LiDAR 表示 - 基本上模仿 LiDAR 信号。通过这种表示, 我们可以应用不同的现有基于 LiDAR 的检测算法。在流行的 KITTI 基准测试中, 我们的方法在现有的基于图像的性能方面取得了令人印象深刻的改进 - 提高了 30 米

范围内物体的检测精度，从先前的 22% 到现在的前所未有的 74%。在提交时，我们的算法在 KITTI 3D 对象检测排行榜上保持最高条目，用于基于立体图像的方法。

Github: https://github.com/mileyan/pseudo_lidar

2.3 DeepMapping: Unsupervised Map Estimation From Multiple Point Clouds

我们提出 DeepMapping，一种新颖的注册框架，使用深度神经网络（DNN）作为辅助功能，将多点云从头开始对齐到全局一致的帧。我们使用 DNN 来模拟高度非凸映射过程，该过程传统上涉及手工制作的数据关联，传感器姿态初始化和全局细化。我们的关键新颖之处在于，正确定义无监督损失以通过反向传播来“训练”这些 DNN 等同于解决基础注册问题，但是对 ICP 的要求实现良好初始化的依赖性更小。我们的框架包含两个 DNN：一个估计输入点云姿态的本地化网络，以及一个通过估计全局坐标的占用状态来模拟场景结构的地图网络。这允许我们将配准问题转换为二进制占用分类，这可以使用基于梯度的优化来有效地解决。我们进一步表明，通过在连续点云之间施加几何约束，可以很容易地扩展 DeepMapping 以解决激光雷达 SLAM 的问题。在模拟和真实数据集上进行实验。定性和定量比较表明，与现有技术相比，DeepMapping 通常能够实现更加稳健和准确的多点云全局注册。在模拟和真实数据集上进行实验。定性和定量比较表明，与现有技术相比，DeepMapping 通常能够实现更加稳健和准确的多点云全局注册。在模拟和真实数据集上进行实验。定性和定量比较表明，与现有技术相比，DeepMapping 通常能够实现更加稳健和准确的多点云全局注册。

Github: <https://ai4ce.github.io/DeepMapping/>

3 深度学习与游戏竞技：

3.1 Pluribus (Superhuman AI for multiplayer poker) :

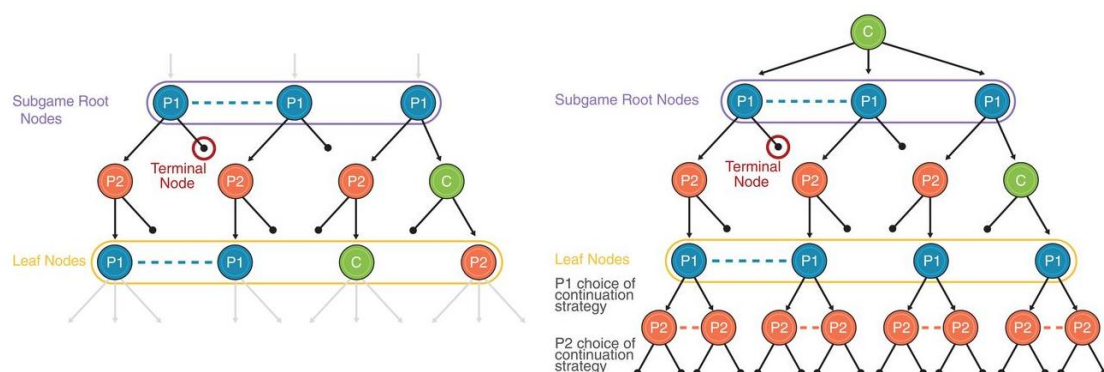
Pluribus 是一个德州扑克 AI。德州不同于围棋，是不完全信息博弈。不完全信息博弈模型的策略设置中存在隐藏的信息，在不完全信息博弈中有以下几个难点：

1. 对手策略的不确定性，即对手的出牌是对手的策略所决定的。

解决方式：对于这种不确定性，既不是采用单一策略，也不是将所有的策略都考虑进去，而是针对每个玩家，采用 k 种不同的策略，文章 $k=4$ 。

2. 一个玩家针对特定环境的最佳策略取决于从他对手的角度观察他采取的策略所对应的情况。

解决方式：Pluribus 跟踪当前情况下他的策略每一手的可能性



Pluribus 算法中的实时在线搜索过程图

此外，根据子对局和部分对局的大小，Pluribus 从两种不同形式的 CFR (counterfactual

regret minimization) 中选取一种进行计算策略。如果子对局较复杂或者是对局的早期, 则采用蒙特卡洛线性反事实遗憾最小化作为蓝图策略; 否则采用线性反事实遗憾最小化的基于向量的最优化形式进行采样可能事件。

3.2 MuZero (Mastering Go, chess, shogi and Atari without rules)

AlphaZero 和 MuZero 都利用称为蒙特卡洛树搜索 (MCTS) 的技术来选择下一个最佳动作。这个想法是, 为了选择下一个最佳动作, 从当前位置“播出”可能的未来情景, 使用神经网络评估其价值并选择使未来预期价值最大化的行动是有意义的。但是, MuZero 有一个问题。由于它不了解游戏规则, 因此不知道给定的动作将如何影响游戏状态, 因此无法想象 MCTS 中的未来情况。它甚至不知道如何从给定位置算出哪些举动是合法的, 或者一方是否获胜。

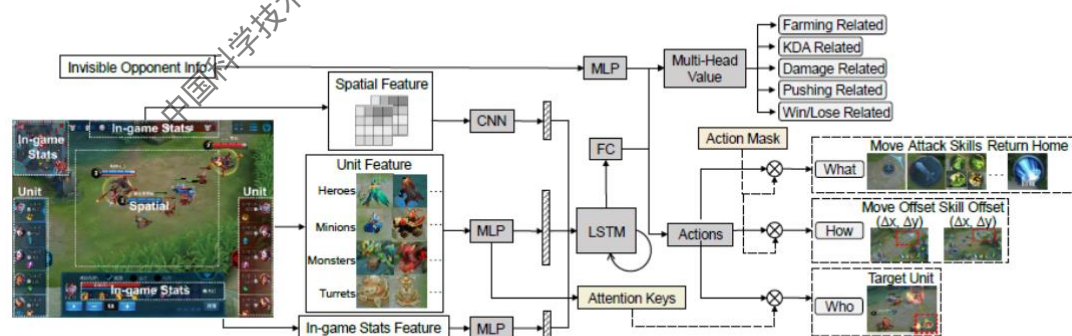
MuZero 算法主要的思想是预测未来, 主要是预测那些可以直接用来规划未来的数据, 比如预测未来的值函数。之后就可以基于预测数据进行规划。

具体做法是: 模型接收一个观测数据(比如游戏的一帧像素), 然后将其编码到隐藏状态空间。在隐藏状态空间可以通过给定动作和自回归的方式进行学习和训练。在每一步, 模型需要产生(或者说去预测)一个策略, 一个值函数(预测的累计奖励), 和一个即时奖励的预测(当前步的奖励预测)。模型端到端直接训练好, 并且作者这里并不采用 dreamer 和 planet 学习环境模型的方法, 认为模型没有必要具有从隐状态复原到原像素这种能力。隐状态只要能够去正确地估计策略、值函数和即时奖励即可。

GitHub: <https://github.com/werner-duvaud/muzero-general>

3.3 Towards Playing Full MOBA Games with Deep Reinforcement Learning

在 MOBA 游戏中, 若想扩展英雄库, 现有的工作在处理由智能体组合 (即阵容) 爆炸性增长所导致的游戏复杂性方面的问题存在难度。此文中腾讯的团队在《王者荣耀》游戏上将新的和已有的学习技术结合起来, 对系统设计和算法都做了大量的优化, 包括课程自我博弈学习、策略蒸馏 (policy distillation), 离策略自适应 (off-policy adaption)、多头值估计 (multi-head value estimation)、蒙特卡洛树搜索等, 在训练和用大量英雄玩游戏, 同时巧妙地解决了可扩展性的问题。网络结构图如下图所示:



3.4 深度学习与游戏经济开源平台:

在深度学习与游戏经济领域, 目前有很多开源平台, 其中比较经典有的 DeepMind 开源 OpenSpiel 平台和 OpenAI 的 spinning up 平台。研究者可在线部署游戏环境, 快速开始实验。

OpenSpiel 是一个综合性的强化学习游戏测试平台, 包括了多种游戏环境和算法, 用于

强化学习研究或搜索策略的研究。

OpenSpiel 可以帮助研究者解决很多强化学习研究中需要设置实验的问题，它支持：

- 单人或多人博弈；
- 完美信息或不完美信息博弈；
- 带有随机性的博弈；
- 普通的多玩家「一步」或二人玩家的多步博弈；
- 交替行动（如下棋）或同时行动的游戏；
- 零和博弈和非零和博弈（如需要合作的博弈等）。

目前 OpenSpiel 支持 45 款游戏，包括国际象棋、围棋、双陆棋、翻转棋等游戏。OpenSpiel 实现的算法有：

| Algorithms | Category | Reference | Status |
|---|--------------|---|--------|
| Information Set Monte Carlo Tree Search (IS-MCTS) | Search | Cowley et al. '12 | ~ |
| Minimax (and Alpha-Beta) Search | Search | Wikipedia1, Wikipedia2, Knuth and Moore '75 | ■ |
| Monte Carlo Tree Search | Search | Wikipedia, UCT paper, Coulom '06, Cowling et al. survey | ■ |
| Lemke-Howson (via <code>nashpy</code>) | Opt. | Wikipedia, Shoham & Leyton-Brown '09 | ■ |
| Sequence-form linear programming | Opt. | Koller, Megiddo, and von Stengel '94, Shoham & Leyton-Brown '09 | ■ |
| Counterfactual Regret Minimization (CFR) | Tabular | Zinkevich et al. '08, Neller & Lanctot '13 | ■ |
| CFR against a best responder (CFR-BR) | Tabular | Johanson et al. '12 | ■ |
| Exploitability / Best response | Tabular | Shoham & Leyton-Brown '09 | ■ |
| External sampling Monte Carlo CFR | Tabular | Lanctot et al. '09, Lanctot '13 | ■ |
| Outcome sampling Monte Carlo CFR | Tabular | Lanctot et al. '09, Lanctot '13 | ■ |
| Q-learning | Tabular | Sutton & Barto '18 | ■ |
| SARSA | Tabular | Sutton & Barto '18 | ■ |
| Policy Iteration | Tabular | Sutton & Barto '18 | ■ |
| Value Iteration | Tabular | Sutton & Barto '18 | ■ |
| Advantage Actor-Critic (A2C) | RL | Mnih et al. '16 | ■ |
| Deep Q-networks (DQN) | RL | Mnih et al. '15 | ■ |
| Ephemeral Value Adjustments (EVA) | RL | Hansen et al. '18 | ~ |
| Deep CFR | MARL | Brown et al. '18 | ~ |
| Exploitability Descent (ED) | MARL | Lockhart et al. '19 | ■ |
| (Extensive-form) Fictitious Play (XFP) | MARL | Heinrich, Lanctot, & Silver '15 | ■ |
| Neural Fictitious Self-Play (NFSP) | MARL | Heinrich & Silver '16 | ■ |
| Neural Replicator Dynamics (NeuRD) | MARL | Omidshafiei, Hennes, Morrill, et al. '19 | X |
| Regret Policy Gradients (RPG, RMPG) | MARL | Srinivasan, Lanctot, et al. '18 | ■ |
| Policy-Space Response Oracles (PSRO) | MARL | Lanctot et al. '17 | ■ |
| Q-based ("all-actions") Policy Gradient (QPG) | MARL | Srinivasan, Lanctot, et al. '18 | ■ |
| Regression CFR (RCFR) | MARL | Waugh et al. '15, Morrill '16 | ■ |
| Rectified Nash Response (PSRO_m) | MARL | Balduzzi et al. '19 | ~ |
| α -Rank | Eval. / Viz. | Omidshafiei et al. '19, arXiv | ■ |
| Replicator / Evolutionary Dynamics | Eval. / Viz. | Hofbauer & Sigmund '98, Sandholm '10 | ■ |

OpenSpiel : https://github.com/deepmind/open_spiel

spinning up 平台: <https://github.com/openai/spinningup>

4. 无监督学习

无监督学习是一种机器学习，其中模型必须在没有标签的情况下并且在最少的人工

监督下查找数据集中的模式。这与监督学习技术（例如分类或回归）相反，在监督学习技术中，模型被赋予一组训练的输入和一组观察值，并且必须学习从输入到观察值的映射。在无监督学习中，只有输入可用，并且模型必须在数据中寻找有趣的模式。无监督学习的另一个名称是知识发现。常见的无监督学习技术包括聚类和降维。

4.1 传统无监督学习：

4.1.1 K-means 聚类

k 均值聚类算法 (k-means clustering algorithm) 是一种迭代求解的聚类分析算法，其步骤是随机选取 K 个对象作为初始的聚类中心，然后计算每个对象与各个种子聚类中心之间的距离，把每个对象分配给距离它最近的聚类中心。聚类中心以及分配给它们的对象就代表一个聚类。每分配一个样本，聚类的聚类中心会根据聚类中现有的对象被重新计算。这个过程将不断重复直到满足某个终止条件。终止条件可以是没有（或最小数目）对象被重新分配给不同的聚类，没有（或最小数目）聚类中心再发生变化，误差平方和局部最小。

https://github.com/kjahan/k_means

4.1.2 层次聚类：

层次聚类 (Hierarchical clustering) 先计算样本之间的距离。每次将距离最近的点合并到同一个类。然后，再计算类与类之间的距离，将距离最近的类合并为一个类。不停的合并，直到合成了一个类。其中类与类的距离的计算方法有：最短距离法，最长距离法，中间距离法，类平均法等。比如最短距离法，将类与类的距离定义为类与类之间样本的最短距离。

层次聚类算法根据层次分解的顺序分为：自下底向上和自上向下，即凝聚的层次聚类算法和分裂的层次聚类算法，也可以理解为自下而上法 (bottom-up) 和自上而下法 (top-down)。自下而上法就是一开始每个个体 (object) 都是一个类，然后根据 linkage 寻找同类，最后形成一个“类”。自上而下法就是反过来，一开始所有个体都属于一个“类”，然后根据 linkage 排除异己，最后每个个体都成为一个“类”。这两种方法没有孰优孰劣之分，只是在实际应用的时候要根据数据特点以及你想要的“类”的个数，来考虑是自上而下更快还是自下而上更快。

<https://github.com/shubhamjha97/hierarchical-clustering>

4.1.3 PCA 降维

PCA 通过线性变换将原始数据变换为一组各维度线性无关的表示，保留部分主成分，实现数据降维

[https://github.com/scikit-learn/scikit-](https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/decomposition/pca.py)

[learn/blob/main/sklearn/decomposition/pca.py](https://github.com/scikit-learn/scikit-learn/blob/main/sklearn/decomposition/pca.py)

<https://gist.github.com/LukeHong/31a6e6cb192626424d3a9efb585759a1>

4.2 深度学习中的无监督学习：

深度学习中的无监督学习主要为基于 generative 和 contrastive。

4.2.1 基于 generative 的无监督学习：

在自编码器中对数据样本编码成特征再解码重构，这里认为重构的效果比较好则说明模型学到了比较好的特征表达。

4.2.1.1 自编码器 Autoencoder (AE)

自编码器模型主要由编码器 (Encoder) 和解码器 (Decoder) 组成，其主要目的是

将输入 x 转换成中间变量 y ，然后再将 y 转换成 x' ，然后对比输入 x 和输出 x' 使得他们两个无限接近。由于编码器将原始输入转换为中间特征，该特征的维度小于原始输入，因此 AE 可以用来数据降维。由于神经网络能够学习非线性关系，AE 可以以低维表示复杂的数据，并可以解码为原始数据。瓶颈层维度可以控制 AE 编码能力。1986 年 Rumelhart 提出自动编码器的概念，并将其用于高维复杂数据处理，Hinton 对原型自动编码器结构进行改进，相对于原始的 Auto Encoder 加大了深度产生了 DAE，先用无监督逐层贪心训练算法完成对隐含层的预训练，然后用 BP 算法对整个神经网络进行系统性参数优化调整。

<https://github.com/nathanhubens/Autoencoders>

4.2.1.2 稀疏自动编码器:

稀疏自动编码器为我们提供了一种信息瓶颈的替代方法，而无需减少隐藏层的节点数量。通过引入正则化，将网络的编码解码过程使用的瓶颈层激活神经元数量引入损失函数，实现稀疏编码

<https://github.com/vivekamin/sparse-autoencoder>

4.2.1.3 去噪自动编码器

在 AE 基础上，具备抗干扰能力，进一步提高编码器通用性，要求能够通过微量干扰的数据还原出原始数据

<https://github.com/alexvmlis/denoising-autoencoder>

4.2.1.4 Variational Autoencoder (VAE)

对于一个生成模型而言，解码器部分应该是单独能够提取出来的，并且对于在规定维度下任意采样的一个编码，都应该能够通过解码器产生一张清晰且真实的图片，因为编码和解码的过程使用了深度神经网络，这是一个非线性的变换过程，所以在 code 空间上点与点之间的迁移是非常没有规律的，前述去噪自编码器是一个合理思路，可以引入噪声，使得图片的编码区域得到扩大，从而掩盖掉失真的空白编码点。不过这还并不充分，可以试图把噪音无限拉长，使得对于每一个样本，它的编码会覆盖整个编码空间。不过需要保证，在原编码附近编码的概率最高，离原编码点越远，编码概率越低。在这种情况下，图像的编码就由原先离散的编码点变成了一条连续的编码分布曲线，这就是变分自编码的核心思想。

在 auto-encoder 中，编码器是直接产生一个编码的，但是在 VAE 中，为了给编码添加合适的噪音，编码器会输出两个编码，一个是原有编码 m ，另外一个控制噪音干扰程度的编码 α ，第二个编码其实很好理解，就是为随机噪音码 α 分配权重，然后加上 $\exp()$ 的目的是为了保证这个分配的权重是个正值，最后将原编码与噪音编码相加，就得到了 VAE 在 code 层的输出结果，其它网络架构都与 Deep Auto-encoder 无异。

<https://github.com/altosaar/variational-autoencoder>

4.2.1.5 Generative Adversarial Network (GAN)

以图像生成模型举例。假设我们有一个图片生成模型 (generator)，它的目标是生成一张真实的图片。与此同时我们有一个图像判别模型 (discriminator)，它的目标是能够正确判别一张图片是生成出来的还是真实存在的。那么如果我们把刚才的场景映射成图片生成模型和判别模型之间的博弈，就变成了如下模式：生成模型生成一些图片→判别模型学习区分生成的图片和真实图片→生成模型根据判别模型改进自己，生成新的

图片→·····, 上述这种博弈式的训练过程就是生成式对抗网络训练过程。

GAN 的主要结构包括一个生成器 G (Generator) 和一个判别器 D (Discriminator), 对于生成器, 输入需要一个 n 维度向量, 输出为图片像素大小的图片, 对于判别器, 输入为图片, 输出为图片的真伪标签。假设我们的生成模型是 $g(z)$, 其中 z 是一个随机噪声, 而 g 将这个随机噪声转化为数据类型 x , 仍拿图片问题举例, 这里 g 的输出就是一张图片。D 是一个判别模型, 对任何输入 x , $D(x)$ 的输出是 0-1 范围内的一个实数, 用来判断这个图片是一个真实图片的概率是多大。

<https://github.com/eriklindernoren/PyTorch-GAN>

4.2.1.6 WGAN:

训练 GAN 有很多头疼的问题。例如: GAN 的训练对超参数特别敏感, 需要精心设计。该文章提出了一种解决方案: 使用 Wasserstein 距离代替 Jensen-Shannon 距离。并依据 Wasserstein 距离设计了相应的算法, 即 WGAN。新的算法与原始 GAN 相比, 参数更加不敏感, 训练过程更加平滑。

<https://github.com/martinarjovsky/WassersteinGAN>

4.2.2 基于 contrastive 的无监督学习:

通过将数据分别与正例样本和负例样本在特征空间进行对比, 来学习样本的特征表示。

4.2.2.1 Deep InfoMax

通过对比学习任务, 显式地建模互信息的模型, 它使用互信息神经估计 (MINE) 来明确地最大化输入数据和学习的低级表示之间的互信息。DIM 认为, 重构误差小, 不能说明学习出来的特征好, 好特征应该是能够提取出样本的最独特, 具体的信息。那如何衡量学习出来的信息是该样本独特的呢? 这里就是用“互信息” (Mutual Information, MI) 来衡量。因此, DIM 的一个核心 idea 是, 训练一个编码器, 它的目标函数, 不是最小化输入与输出的 MSE, 而是最大化输入与输出的互信息。

互信息是概率论和信息论中重要的内容, 它表示的是一个随机变量中包含另一个随机变量的信息量, 可以理解成两个随机变量之间的相关程度。

<https://github.com/rdevon/DIM>

4.2.2.2 contrastive multiview coding:

基于实例区分思想, CMC 建议采用图像的多个不同视图作为正样本, 而将另一个图像作为负样本。在特征空间中, CMC 拉近图像的多个视图, 并远离其他样本。但是, 它在某种程度上受到 Deep InfoMax 概念的约束, 为每个正样本采样一个负样本。

<https://github.com/HobbitLong/CMC>

4.2.2.3 Momentum Contrast for Unsupervised

在 MoCo 中, 研究人员进一步发展了通过动量对比进行实例区分的想法, 它放弃了传统的端到端训练框架, 并设计了带有两个编码器 (查询和键) 的动量对比学习, 这可以防止在开始阶段损失收敛的波动为了扩大负样本的容量, MoCo 使用队列 (K 最多为 65536) 将最近编码的批次保存为负样本, 这显着提高了负采样效率。

<https://github.com/facebookresearch/moco>

5. 人体姿态识别与估计

5.1 OpenPose(CVPR2017)

OpenPose 人体姿态识别项目是美国卡耐基梅隆大学 (CMU) 基于卷积神经网络和监督学习并以 caffe 为框架开发的开源库。可以实现人体动作、面部表情、手指运动等姿态估计。适用于单人和多人, 具有极好的鲁棒性。是世界上首个基于深度学习的实时多人二维姿态估计应用, 基于它的实例如雨后春笋般涌现。

特点:

关键点亲和度向量场: 解决关键点的聚类

自下而上的 parse 解析: 效率更高, 与人的数量无关

两个分支: 关键点置信度预测网络 & 关键点亲和度向量场预测网络

挑战:

人的数量不定人出现的位置不定

人的尺寸大小不定

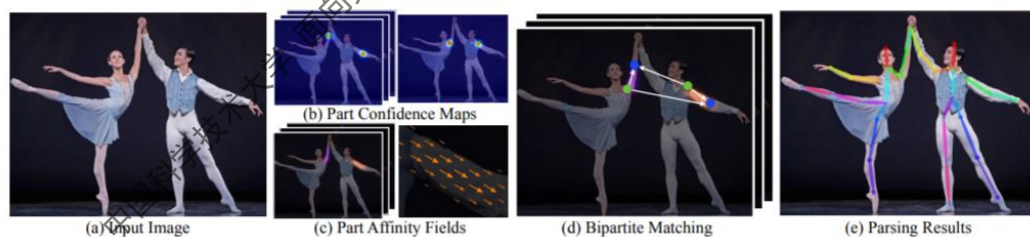
人的社交互动造成空间干扰

- 接触
- 遮挡

总体思路:

openpose 是基于 CVPR 2016 Convolutional Pose Machine (CPM) 和 CVPR2017 realtime multi-person pose estimation 以及 CVPR2017 Hand Keypoint Detection in Single Images using Multiview Bootstrapping 这 3 篇 paper 的模型做出来的。

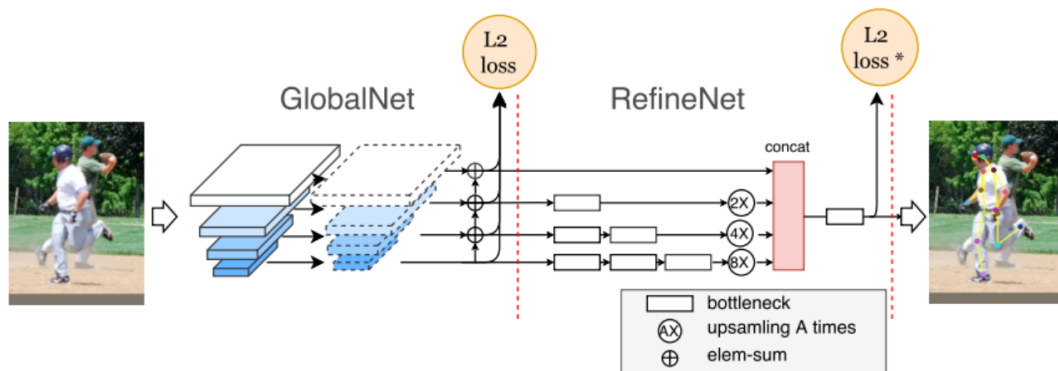
(a)输入图像 --> (b)预测关键点置信度 & (c)关键点亲和度向量 --> (d)关键点聚类 --> (e)骨架组装



5.2 CPN(Cascaded Pyramid Network, CVPR2018)

2017 年 coco keypoint benchmark 数据集冠军。

总体结构:



网络整体结构采用 CPN(Cascaded Pyramid Network)结构, Cascaded 指的是级联的意思, 代表了网络级联了 2 个类似的模块 (GlobalNet 和 RefineNet), Pyramid 指的是类似于 FPN 的金字塔网络结构。

其中, GlobalNet 负责网络所有关键点的检测, 重点是对比较容易检测的眼睛, 胳膊等部位的关键点预测效果较好, 采用的损失函数为 L2 loss。其中在每一个 elem-sum 操作之前, 都对 featuremap 使用了 1*1 的卷积操作。

RefineNet 指的是对 GlobalNet 预测的结果进行修正的网络。GlobalNet 对身体部位的那些遮挡, 看不见, 或者有复杂背景的关键点预测误差较大, RefineNet 则专门修正这些点。主要还是基于 shortcut 的思想。在该阶段的训练中, 还使用了类似 OHEM 的 online hard keypoints mining 难例挖掘策略。

5.3 MSPN(CVPR2019)

原来 CPN 的两个 stage 可能并没有把 context 信息利用好, 单个 stage 的模型能力可能已经比较饱和了, 增加更多 stage 来做 refinement 可能是一个解决当前问题, 提升人体姿态估计算法 upper-bound 的途径。在 CPN 的 globalNet 基础上面, 做了多个 stage 的堆叠, 类似于 Hourglass 的结构。

总体思路:

改进 1: 优化单个 stage 结构

在现有的多 stage 网络中, 每个 stage 在进行特征降采样和升采样时, 其通道是保持不变的, 而不是像 Resnet 一样随着特征图的减小会相应的增加通道。这种现象就会导致特征丢失。本文每个 stage 就是一个 CPN 的 globalNet。

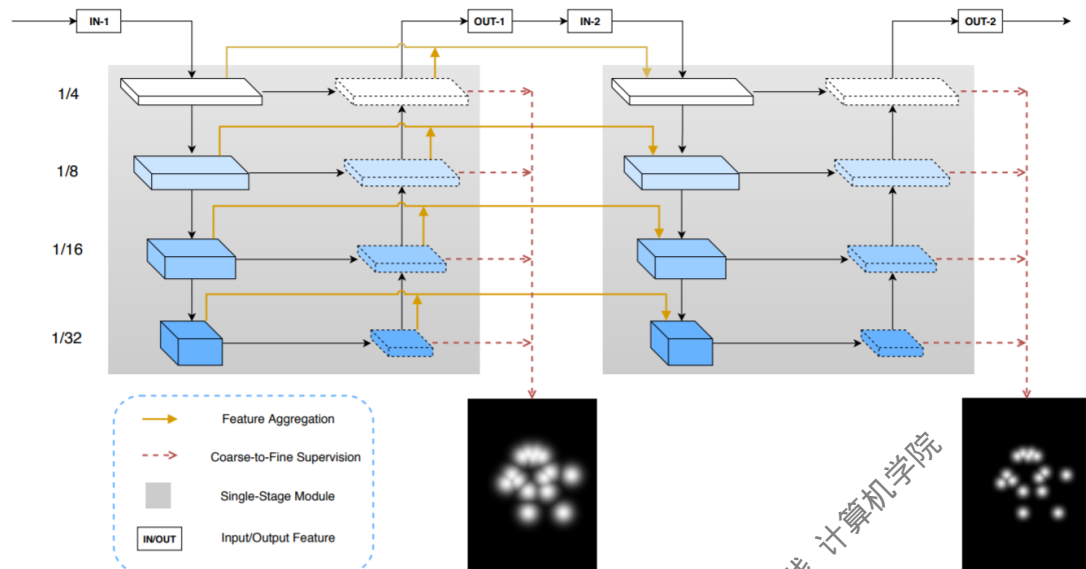
改进 2: 相邻 stage 特征聚合

每个 stage 都会对特征图进行降采样和升采样, 多个 stage 就会反复的进行降采样和升采样, 这种反复的操作使得特征信息丢失明显, 最终导致网络优化困难。作者将相邻阶段的特征进行聚合, 以增强特征信息传播能力并降低训练难度。具体的特征聚合方法为: 上个阶段中相同 size 的降采样特征经过 11 卷积编码后的特征, 上个阶段中相同 size 的升采样特征经过 11 卷积编码后的特征, 以及当前 stage 的降采样特征。

改进 3: 多 stage 由粗到细监督

多 stage 网络的特点是, 每个 stage 的输出都能作为最终的关键点检测结果。而且随着 stage 的增多, 关键点定位会越来越准。为了使得在前端的 stage 能够获得更好的知道, 作者提出了由粗到细的多分支监督的方式来优化多 stage 的能力。对于每个 stage 的

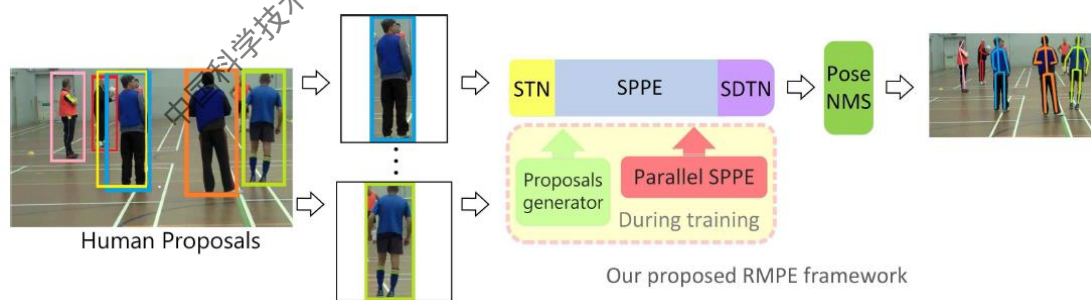
特点，采用不同 kernel-size 的高斯核制作标签，越靠近输入的 stage kernel-size 越大。



5.4 AlphaPose(CVPR2019)

AlphaPose 是一个精准的多人姿态估计系统，是首个在 COCO 数据集上可达到 70+ mAP (72.3 mAP, 高于 Mask-RCNN 8.2 个百分点)，在 MPII 数据集上可达到 80+ mAP (82.1 mAP) 的开源系统。为了能将同一个人的所有姿态关联起来，AlphaPose 还提供了一个称为 Pose Flow 的在线姿态跟踪器，这也是首个在 PoseTrack 挑战数据集上达到 60+ mAP (66.5 mAP) 和 50+ MOTA (58.3 MOTA) 的开源在线姿态跟踪器，精准度高于现有技术的最好结果。

总体结构：



Symmetric STN=STN+SPPE+SDTN

STN：空间变换网络，对于不准确的输入，得到准确的人的框。输入候选区域，用于获取高质量的候选区域。

SPPE：得到估计的姿态。

SDTN：空间逆变换网络，将估计的姿态映射回原始的图像坐标。

Pose-NMS：消除额外的估计到的姿态

Parallel SPPE：训练阶段作为额外的正则项，避免陷入局部最优，并进一步提升 SSTN

的效果。包含相同的 STN 及 SPPE（所有参数均被冻结），无 SDTN。测试阶段无此模块。

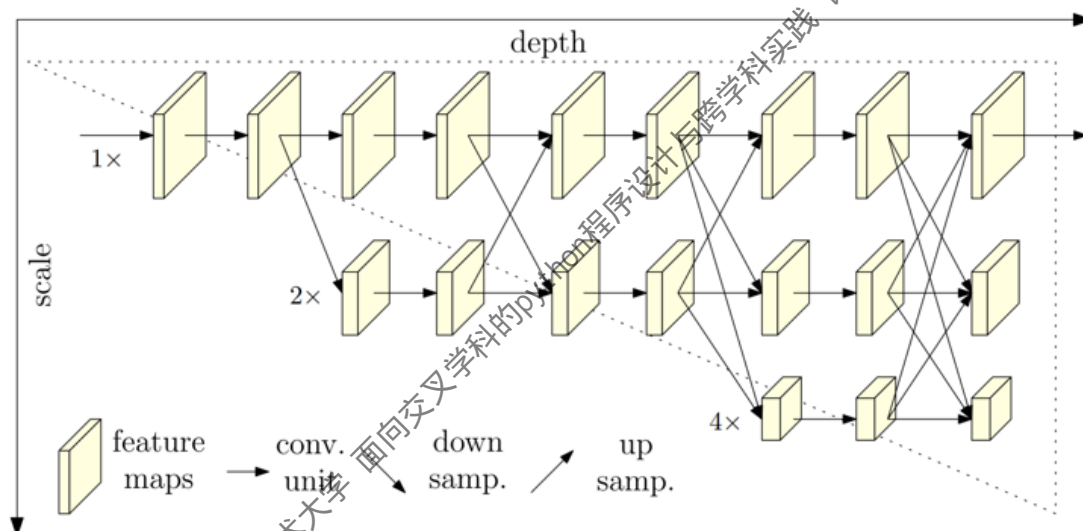
PGPG（Pose-guided Proposals Generator）：通过 PGPG 网络得到训练图像，用来训练 SSTN+SPPE 模块。

相比 Openpose，Alpha 的准确率更高，能够处理 human bbox 不太精确的时候以及多余的检测

5.5 HRNet(CVPR2019)

高分辨率网络（HRNet）是用于人体姿势估计的先进神经网络-一种图像处理任务，可在图像中找到对象的关节和身体部位的配置。网络中的新颖之处在于保持输入数据的高分辨率表示，并将其与高分辨率到低分辨率子网并行组合，同时保持有效的计算复杂性和参数计数。

整体结构：



优点：

并行连接高低分辨率子网，而不是像大多数现有解决方案那样串联连接。因此，我们的方法能够保持高分辨率而不是通过从低到高的过程恢复分辨率，因此预测的热图可能在空间上更精确

大多数现有的融合方案汇总了低级别和高级别的表示。相反，我们在相同深度和相似水平的低分辨率表示的帮助下执行重复的多尺度融合以提升高分辨率表示，反之亦然，导致高分辨率表示对于姿势估计也是丰富的。因此，我们预测的热图可能更准确。个人感觉增加多尺度信息之间的融合是正确的，例如原图像和模糊图像进行联合双边滤波可以得到介于两者之间的模糊程度的图像，而 RGF 滤波就是重复将联合双边滤波的结果作为那张模糊的引导图，这样得到的结果会越来越趋近于原图。此处同样的道理，不同分辨率的图像采样到相同的尺度反复的融合，加之网络的学习能力，会使得多次融合后的结果更加趋近于正确的表示。

5.6 HigherHRNet(CVPR 2020)

HigherHRNet, 是微软在 HRNet 之后延续的一个工作。HRNet 在 top-down 的方法中表现的很好, 是因为这种并行的结构使得最后的 feature map 能够包含各个分辨率的信息, 尤其是对高分辨率信息保留的效果较之前提升尤为明显。在 bottom-up 的方法中, 作者认为有两个问题需要解决: ①scale variation, 即图片中有各种 scale 的人, 如果仅仅在一个分辨率的 feature map 下进行预测, 难以保证每个 scale 的人的关键点都能被精确预测; ②精确定位 small person 的关键点。之前一些网络在推理时使用 multiscale evaluation, 能够缓解问题①, 但仍然无法解决问题②, 对 small person 的预测不够精确。

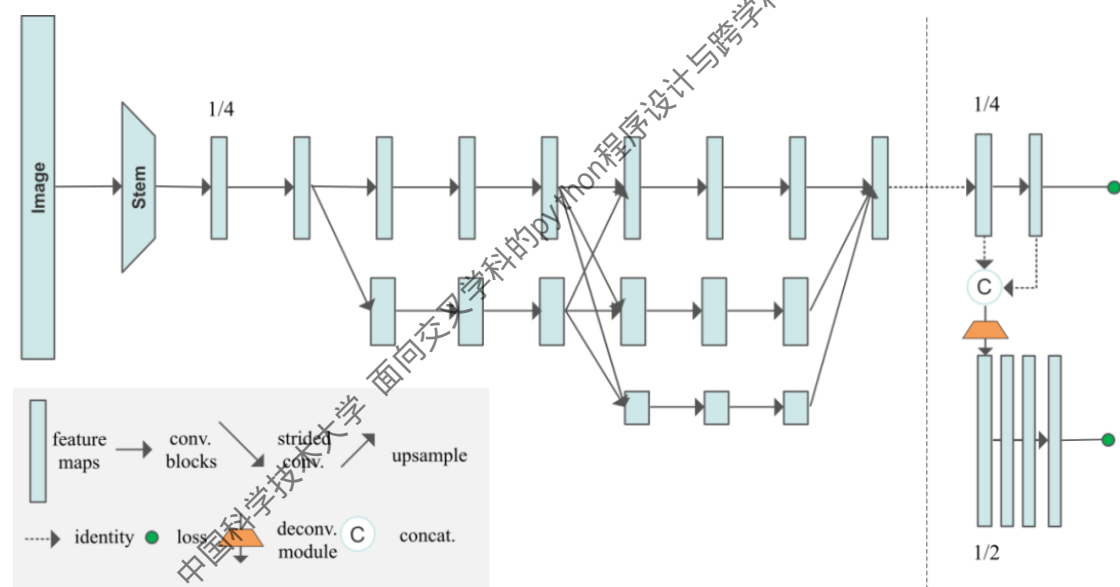
特点:

解决尺度变化的挑战。

可以在训练阶段生成具有多分辨率监督的高分辨率特征金字塔, 并在推理阶段生成多分辨率热图聚合, 以预测有利于小型人体的尺度感知的高分辨率热图。

优于 COCO 数据集上的所有其他自下而上的方法

总体思路:



该网络的骨干是常规的 HRNet, 但最后增加了一部分, 用于输出更高分辨率的热图。架构的右侧部分输出两个热图-一个用于低分辨率, 一个用于高分辨率-分辨率为 128×128 和 256×256 。在推论期间, 两个热图均被汇总为更高的分辨率, 并且为关键点选择了最高值的点。检测。梯形是一个反卷积层, 其输出分辨率高 2 倍, 其后跟随 4 个残差块。而且, 对于每个关键点, 计算输出标量标签, 接近标签值形成属于特定人物的一组关键点, 而遥远标签值指示属于不同人物的关键点组。标记根据在所述的“融合嵌入”方法计算本文。标签值仅针对最低分辨率的热图进行训练和预测, 因为作者发现, 经验上更高分辨率的热图的标签值无法学会预测, 甚至无法收敛。

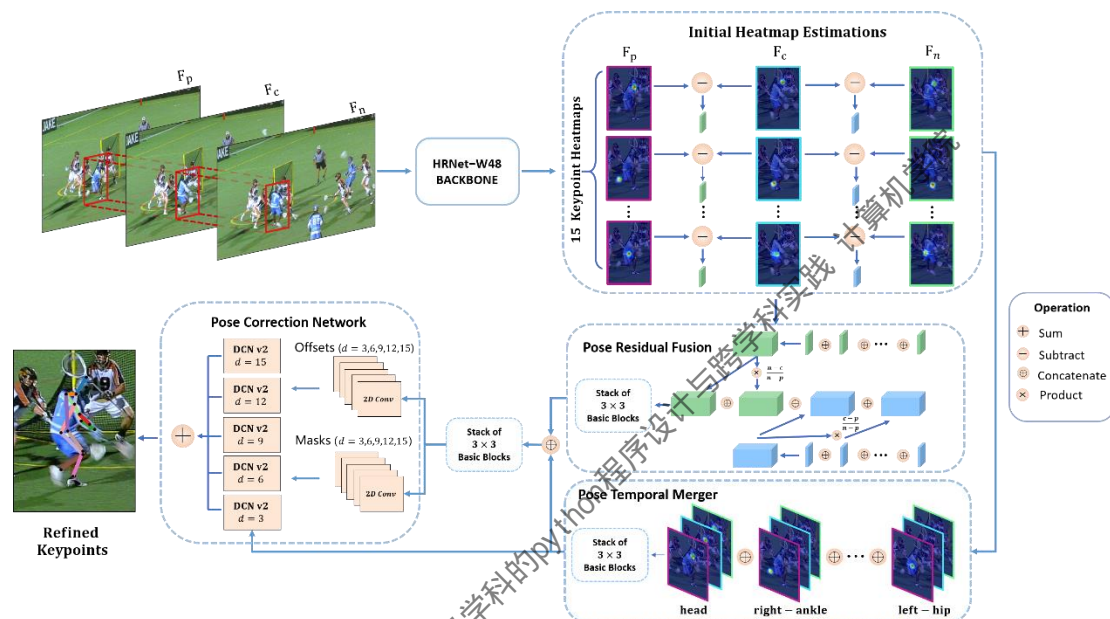
在训练过程中, 损失函数是热图预测损失和标签值损失的加权平均值 (根据关联嵌入方法, 同一组标签之间的距离越小, 损失就越小, 不同组标签之间的距离就越远)。每个热图分辨率损失均根据基本情况独立计算, 并进行总和运算。

5.7 DCPose (CVPR2021)

一种新的多帧人体姿态估计框架，利用视频帧之间丰富的时间线索来促进关键点检测。在大规模基准数据集 PoseTrack2017 和 PoseTrack2018 的多帧人姿态估计挑战中排名第一。

整体思路：

框架中设计了三个模块化组件。位姿时间融合模块对关键的时空背景进行编码，生成有效的搜索范围；位姿残差融合模块在两个方向上计算加权位姿残差。然后通过位姿校正网络处理这些，以有效地细化位姿估计。



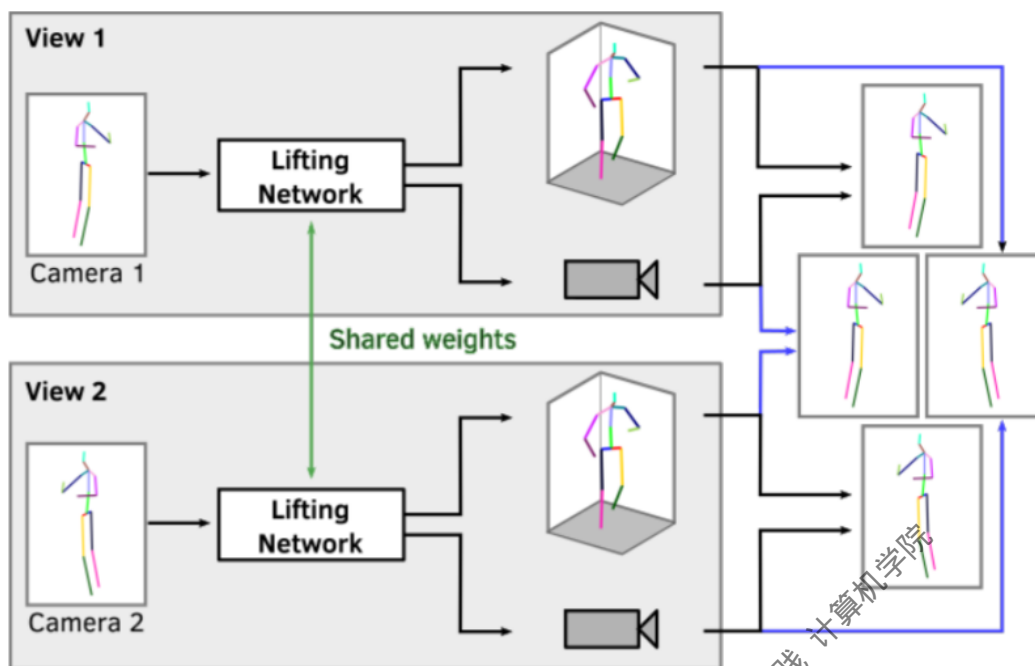
5.8 CanonPose(CVPR2021)

基于自监督的 3D 人体姿态估计方法，利用多视角 2D 图像估计 3D 人体姿态。在两个基准数据集 (Human3.6M 和 MPI-INF-3DHP) 和 in-the-wild 数据集 SkiPose 上进行了评估

总体思路：

首先将同一姿态不同视角下的图像分别输入两个共享权重的 Lifting 网络，这部分网络输出为两个分支，一个分支输出标准坐标下的 3D 姿态 (outputs a 3D pose in a canonical rotation)，另一个分支输出为相机方向 (camera orientation)，预测的相机方向用来将预测的 3D 姿态旋转回相机坐标系下的 2D 姿态。

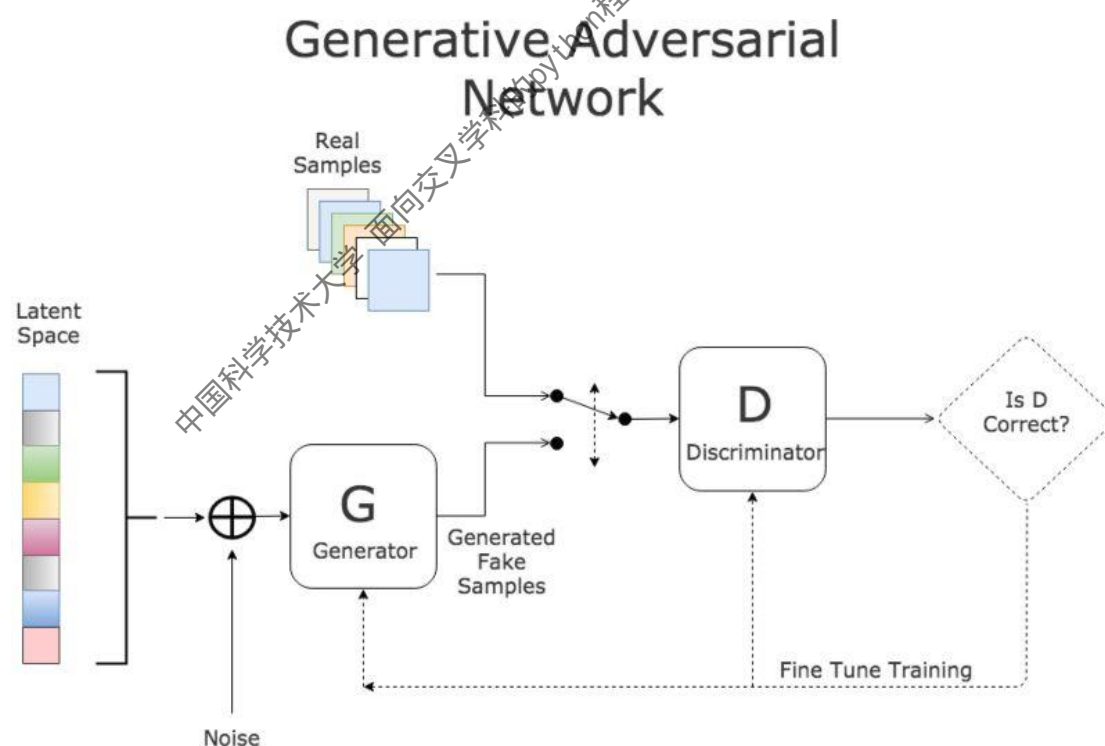
如果两个网络对 3D 姿态和相机方向预测比较准的话，第一个网络预测得到的 3D 姿态结合第二个网络预测的相机方向结合应该能获得第二个网络输入图的 2D 姿态，这样就可以根据 2D 的姿态标签来对网络进行训练获取 3D 姿态了。



6.生成对抗网络 gan

6.1 Generative Adversarial Network

<https://github.com/goodfeli/adversarial>

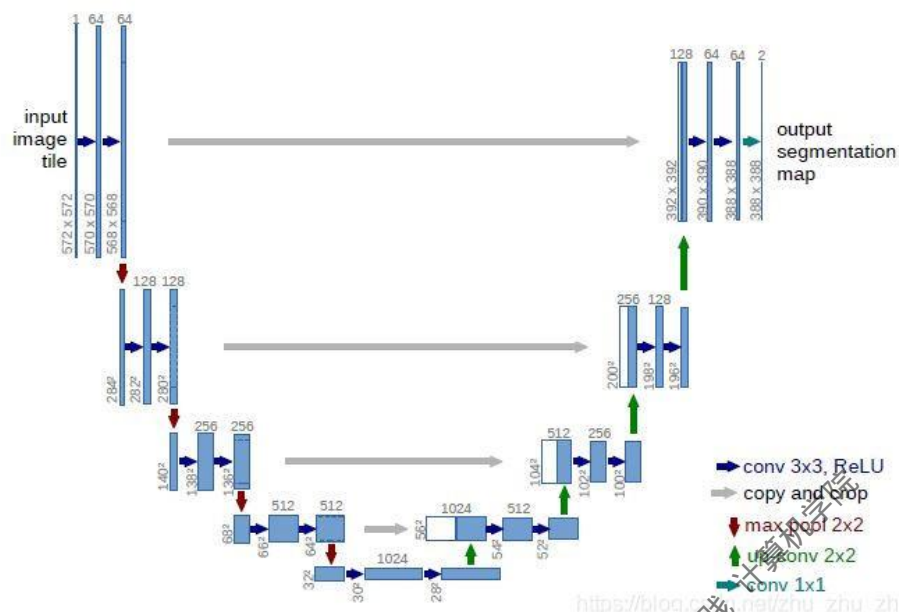


6.2 Image-to-Image Translation with Conditional Adversarial Networks

<https://github.com/williamFalcon/pix2pix-keras>

模型在训练时候需要采用成对的训练数据, 并对 GAN 模型采用了不同的配置, 它应用到了 PatchGAN 这个模型, PatchGAN 对图片的一块 70*70 大小的区域进行观察来判断

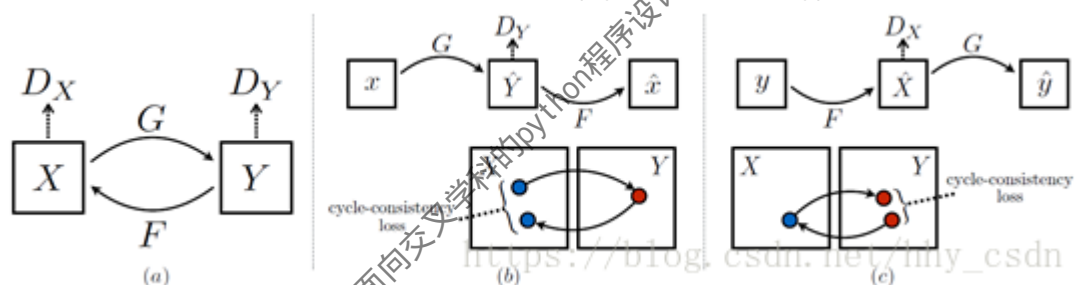
该图片是真是假，而不需要观察整张图片。



6.3 Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

<https://github.com/hanyoseob/pytorch-CycleGAN>

论文提出了一个非常好的方法—循环一致性(Cycle-Consistency)损失函数，



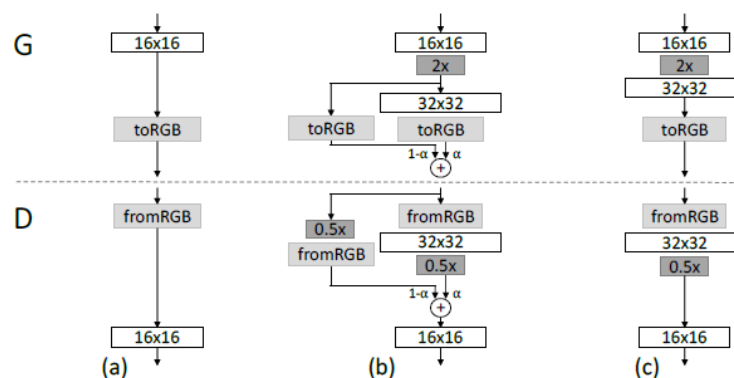
6.4 Progressive Growing of GANs for Improved Quality, Stability, and Variation

https://github.com/karras/progressive_growing_of_gans

提出了一种新的训练对抗神经网络的方法。核心思想是逐步训练生成器和判别器：从低分辨率开始，随着训练进程推进，逐步增加新的层来提炼细节。这种方法不仅加快了训练速度并且更加稳定，可以产生高质量的图像

提出了一些实施的细节对于消除生成器和判别器的不好的竞争

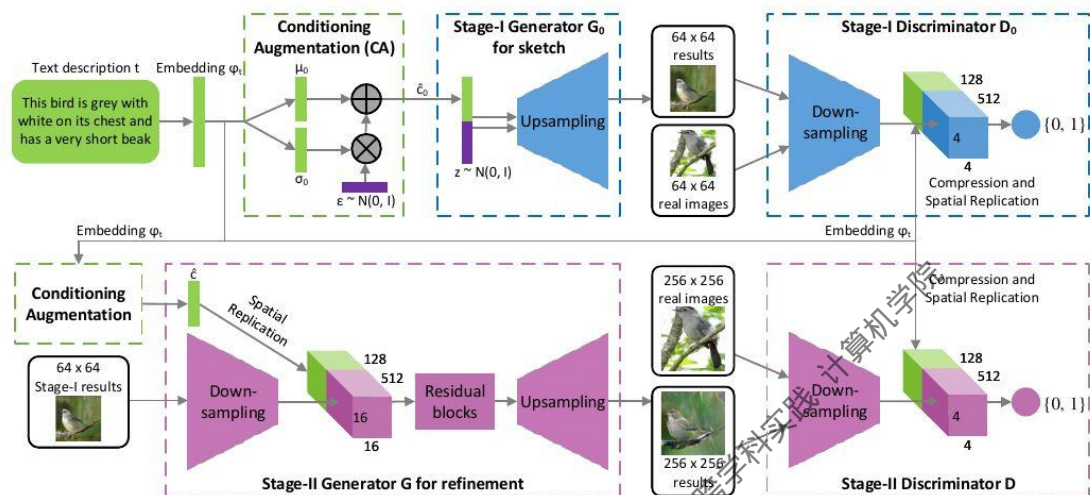
提出了新的方法来评估 GAN 产生的结果



6.5 StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks

<https://github.com/suryar510/StackGAN>

着重要解决的问题是提高图片的分辨率同时为图片添加必要的细节。采用了先验知识，以及多尺度方法，并且通过一个文本向量来引入文本信息，并提取一些视觉特征



6.6 Large Scale GAN Training for High Fidelity Natural Image Synthesis

<https://github.com/sxhxtiang/BigGAN-pytorch>

论文同时结合了很多结构和技术，包括自注意机制(Self-Attention)、谱归一化(Spectral Normalization)等

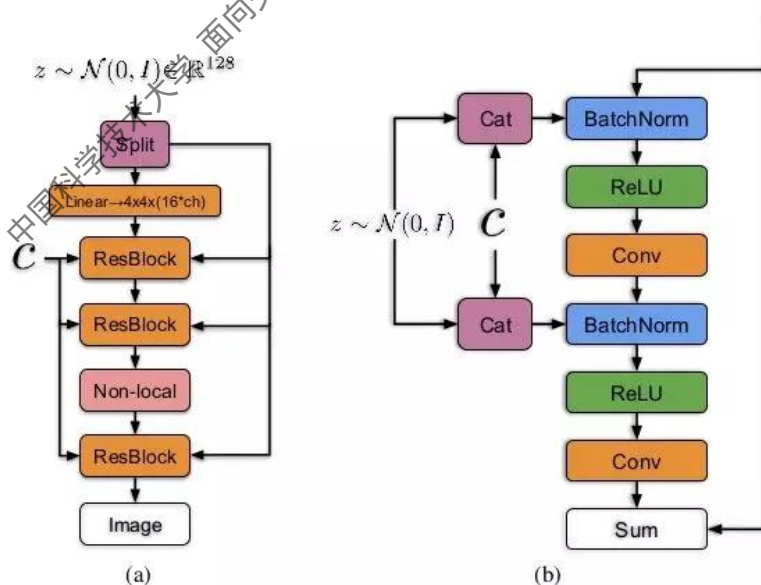
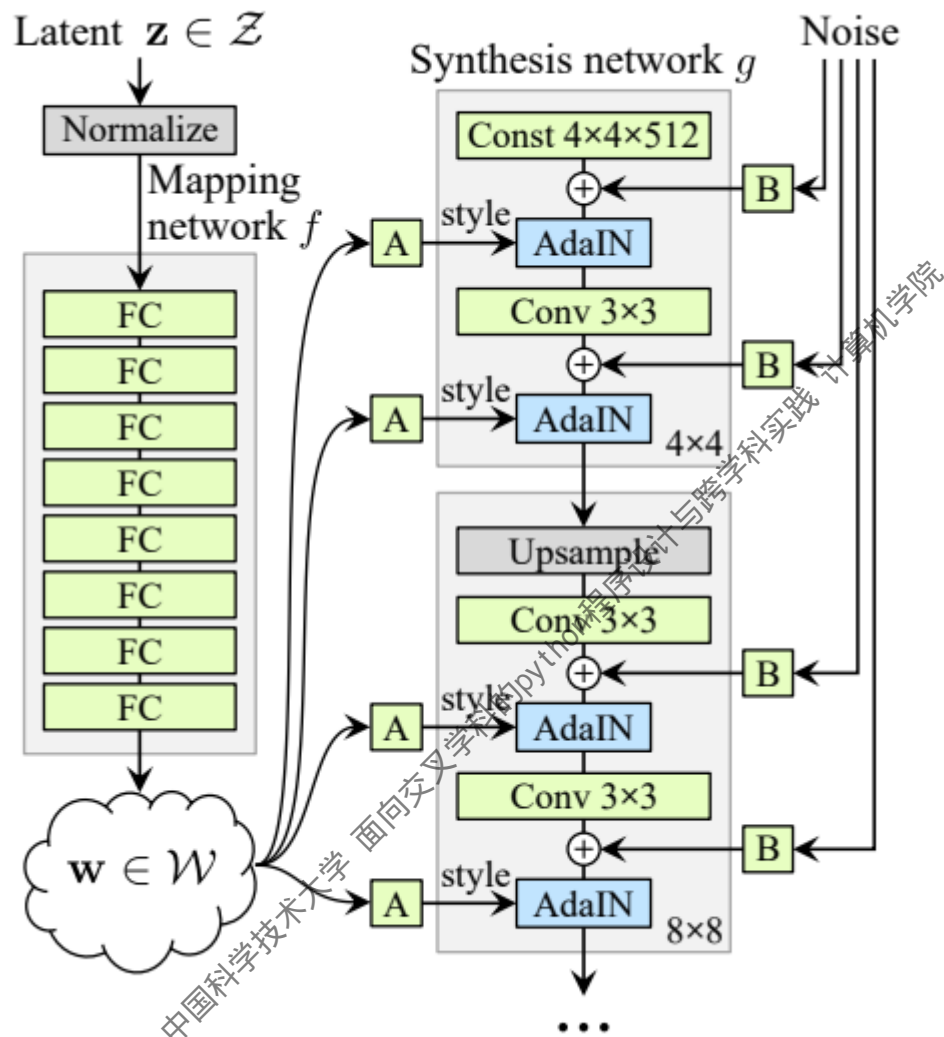


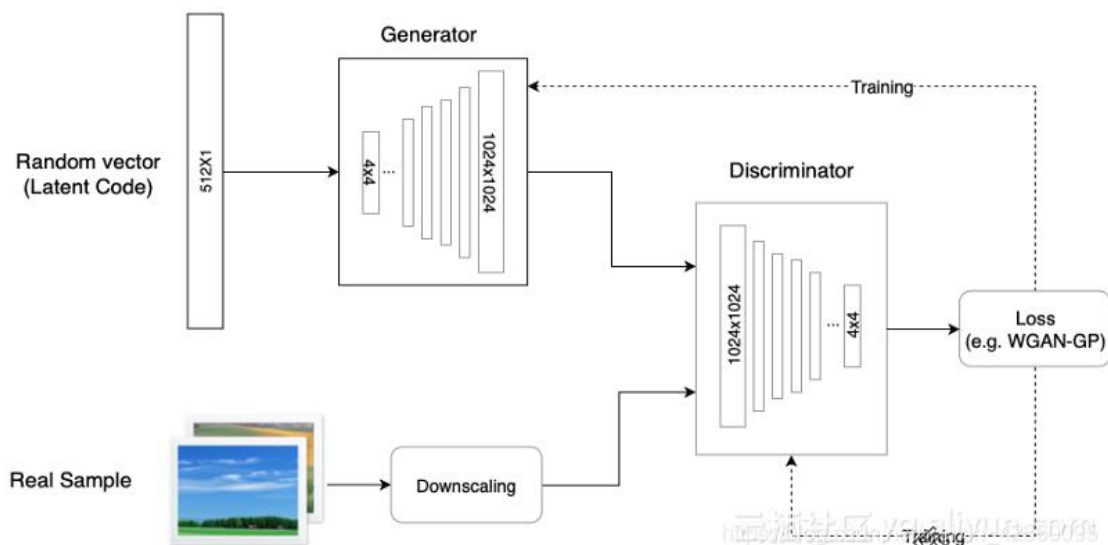
Figure 15: (a) A typical architectural layout for G ; details are in the following tables. (b) Details of the Residual Block in G . c is concatenated with a chunk of z and projected to the BatchNorm gains and biases.

6.7 A Style-Based Generator Architecture for Generative Adversarial Networks

<https://github.com/rosinality/style-based-gan-pytorch>

借鉴了如 Adaptive Instance Normalization (AdaIN)的自然风格转换技术，来控制隐空间变量 \mathbf{z} ，在生产模型中结合了一个映射网络以及 AdaIN 条件分布的做法

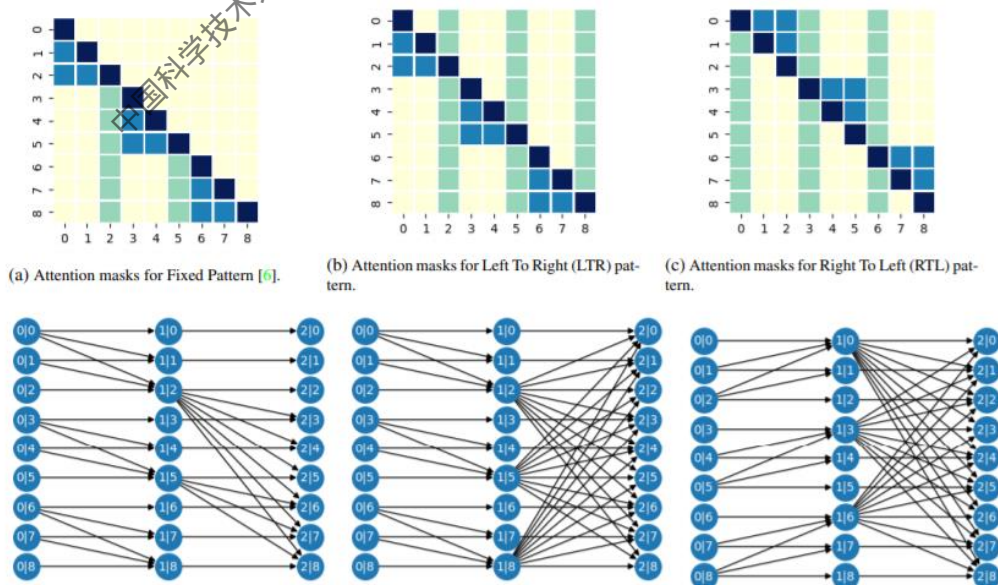




6.8 Your Local GAN: Designing Two Dimensional Local Attention Mechanisms for Generative Models

<https://github.com/giannisdaras/ylg>

为了解决密集注意力层 (Dense attention) 的计算问题, 本文提出一种稀疏的注意力层. 它利用 attention heads [25] 并将注意力特征图稀疏化, 从而既能够并行运算, 加快运算速度, 同时利用全信息有向图生成的稀疏注意力特征图, 能够保留密集注意力层所需的特征.



6.9 A Characteristic Function Approach to Deep Implicit Generative Modeling

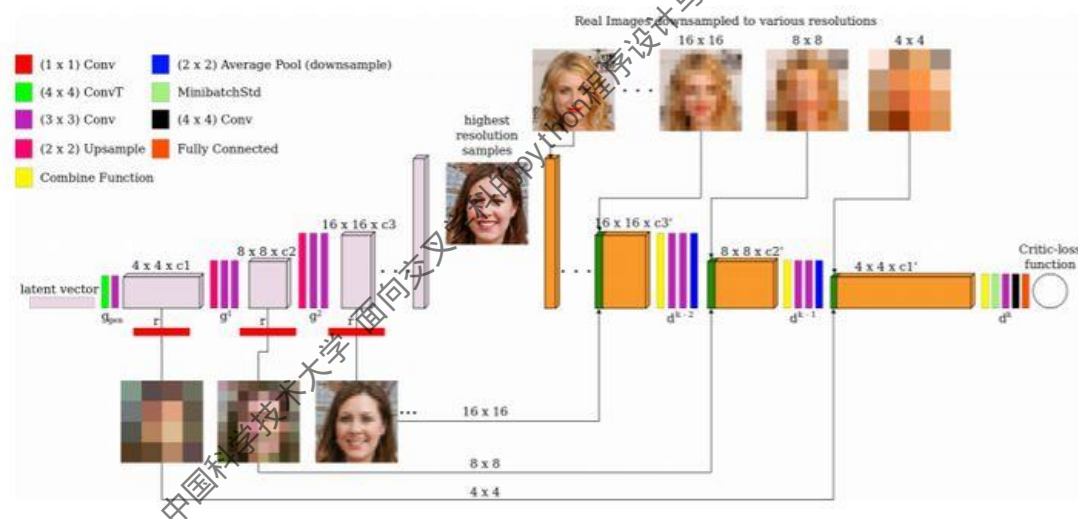
<https://github.com/crslab/OCFGAN>

将学习隐式生成模型（IGM）的问题公式化为最小化 characteristic functions 之间的期望距离。在适当选择的加权分布下，匹配实际数据分布和生成数据分布的 characteristic functions。通过用 characteristic function distance (CFD) 改进 GAN，可获得一个易于实现、训练的模式。

6.10 MSG-GAN: Multi-Scale Gradients for Generative Adversarial Networks

<https://github.com/akanimax/msg-gan-v1>

尽管 GAN 在图像合成任务中取得了巨大成功，但很难适应不同数据集，部分原因是训练期间不稳定及对超参数敏感。这项工作提出多尺度梯度生成对抗性网络（MSG-GAN），简单但有效，从鉴别器到生成器的多个尺度的梯度流来解决此问题



6.11 AdversarialNAS: Adversarial Neural Architecture Search for GANs

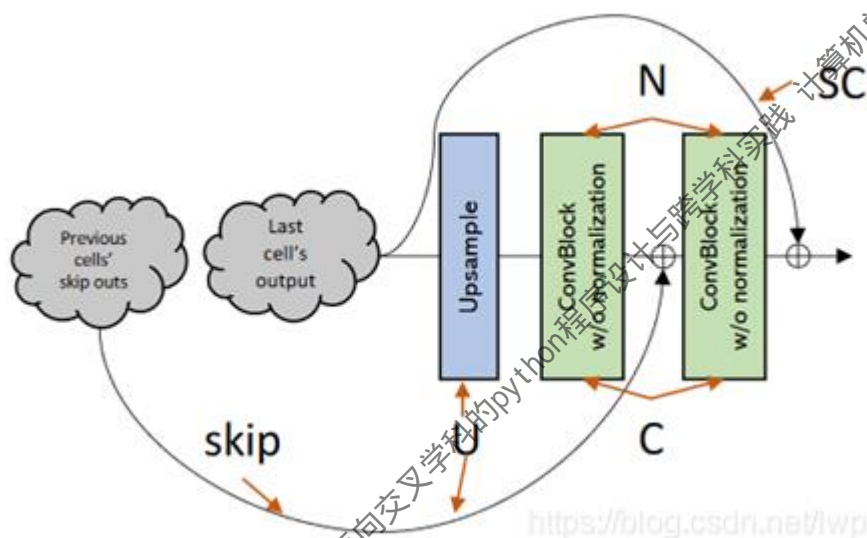
<https://github.com/chengaopro/AdversarialNAS>

本文提出一种专门针对生成对抗网络（GAN）量身定制的 Adversarial NAS 方法，以搜索无条件图像生成任务中的高级生成模型



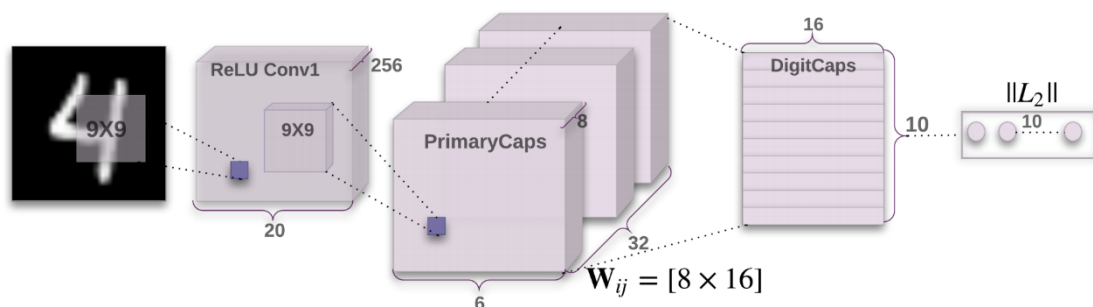
搜索空间如下

- skip: 二进制值跳跃, 指示当前单元是否从前一个单元获取附加跳跃连接;
- C: 基本卷积块, 决定是否包括前激活或后激活;
- N: 归一化类型的选择: 批归一化、实例归一化、无归一化;
- U: 要使用的上采样类型: 双线性上采样、最近邻上采样或 2 步幅反卷积;
- SC: 是否使用单元内的附加跳跃连接。



7. 胶囊网络

7.1 2017 年: Dynamic Routing Between Capsules



对象的实例化参数表示 (即特定物体、概念实体等出现的概率与某些属性): 向量的方向
对象存在概率: 向量的长度

层间路由方式：动态路由，本质为 K-means 聚类

论文开源代码：<https://github.com/Sarasra/models/tree/master/research/capsules>

网络项目：

Keras:

<https://github.com/XifengGuo/CapsNet-Keras>

Tensorflow:

<https://github.com/naturomics/CapsNet-Tensorflow>

PyTorch:

<https://github.com/XifengGuo/CapsNet-Pytorch>

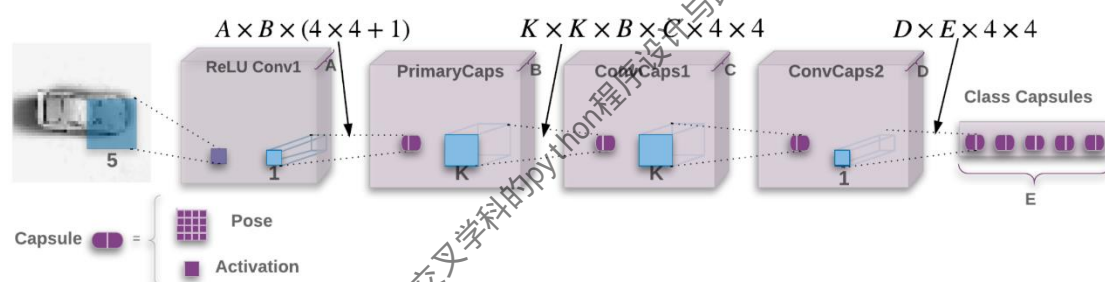
<https://github.com/gram-ai/capsule-networks>

<https://github.com/timomernick/pytorch-capsule>

Hinton's talk:

<https://www.youtube.com/watch?v=rTawFwUvnLE>

7.2 2018 年：Matrix Capsules With EM Routing



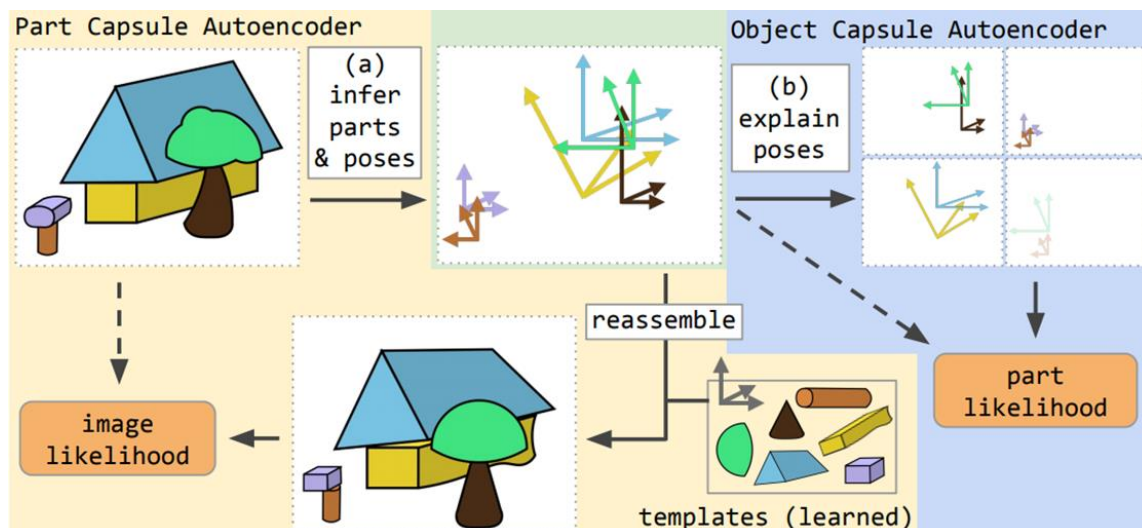
改变了实例化对象的表示和层间路由方式，将自身的姿态矩阵与视角不变矩阵的变换矩阵相乘，为上一层胶囊的姿态矩阵投票。

实例化对象表示：由向量改为矩阵

对象存在概率：由向量模长改为一个单独的激活值维度

层间路由方式：改为高斯混合模型聚类（GMM），用 EM 算法来迭代更新权重

7.3 2019 年：Stacked Capsule Autoencoders



推翻了前述工作中胶囊的设计，采用自编码，将图像中的实例的部件及属性从像素二维空间中以像素重建的方式抽取出来，再用重构的方式解释部件与整体的关系。

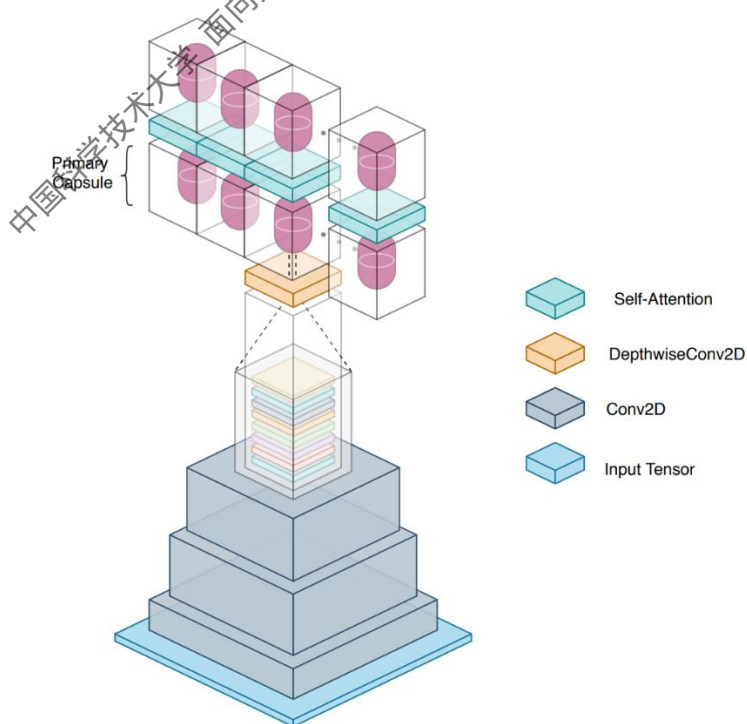
编码器：将像素图像编码为胶囊的集合，每个胶囊表示图像中的一个部件，包括姿态矩阵，存在概率和特殊特征。

解码器：通过训练学习部件模板，用姿态矩阵对模板进行仿射变换，重建图像。

Hinton's talk:

<https://www.bilibili.com/video/av88128940>

7.4 2021 年：Efficient-CapsNet: Capsule Network With Self-Attention Routing



建立在 2017 年 CapsNet 的基础上，采用自注意力机制实现层间路由。

与原始 CapsNet 相比，参数量降低到 2%，同时在 MNIST 和 smallNORB 的分类任务上降低了错误率。

中国科学技术大学 面向交叉学科的python程序设计与跨学科实践 计算机学院