人工智能各领域知识脉络和学习指南

学术部　　竞赛部

浙江大学学生人工智能协会

项目地址:https://github.com/zjuai/AI_Learning_Guidance

日期：2020 年 11 月 22 日

1　代码入门

1.1　Python 基础

- 了解基本的 python 语法。推荐书籍：**Python 编程：从入门到实践, 流畅的 Python**。
- 熟悉使用 pip 包管理，尝试使用 venv、virtualenv、conda 等进行环境管理
  - https://pypi.org/
  - https://virtualenv.pypa.io/en/latest/
  - https://docs.conda.io/en/latest/
  - https://www.anaconda.com/

1.2　基础数据知识，基本的 python 包和其他工具

几乎所有的库/工具都在文档里有 tutorial，跟着走一遍就可以了

- 数据格式: csv, json, xml, sql
- 数据采集
  - requests: https://requests.readthedocs.io/en/master/
  - scrapy: https://scrapy.org/
  - bs4: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- 数据处理
  - pandas: https://pandas.pydata.org/
  - numpy: https://numpy.org/
- 数据可视化
  - matplotlib: https://matplotlib.org/
  - plotly: https://plotly.com/
- 数据存储
  - sqlite: https://docs.python.org/3/library/sqlite3.html
  - mysql: https://dev.mysql.com/doc/
  - MongoDB: https://www.mongodb.com/

1.3 基础 ML 库

- sklearn: https://scikit-learn.org/。传统机器学习库。
- tensorflow: https://www.tensorflow.org/。谷歌的文档一贯质量很高，直接看官方文档就好。
  tensorflow2.0 tutorial: https://github.com/dragen1860/TensorFlow-2.x-Tutorials
- Pytorch: https://pytorch.org/。目前主流的趋势是转向 torch。torch 有一个推荐的教程，d2l，可以选择使用里面的 torch 版代码进行学习。
  Pytorch 的一个入门教程：https://github.com/ShusenTang/Dive-into-DL-PyTorch

1.4 ML 封装库

本质是对于上面提到的库的封装，但是使用起来非常高效

- PyCaret:https://pycaret.org/。封装了 sklearn 和一些其他的传统机器学习库，并且整合了 AutoML 自动调参。
- pytorch-lightning: https://pytorchlightning.ai。封装了 pytorch，使得代码的可读性更好。个人感觉称为 best practice 更好。
- keras: https://keras.io/。主要封装了 tensorflow，使用起来比较简单，易于上手。
- ML.NET: https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet。可以一行代码不写进炼丹，全程 GUI。底层是 tf，不过还不是很成熟，熟悉 C# 可以玩玩。

2 机器学习

2.1 知识脉络图

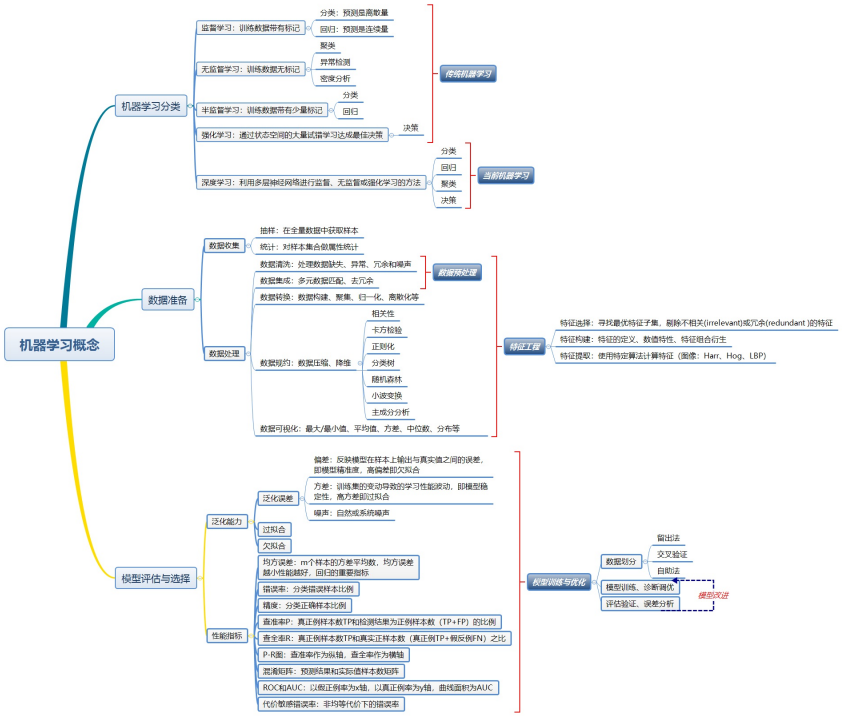

图 1: 机器学习总知识脉络（原图）

针对机器学习分类中各个类别所常用的模型如下图所示：

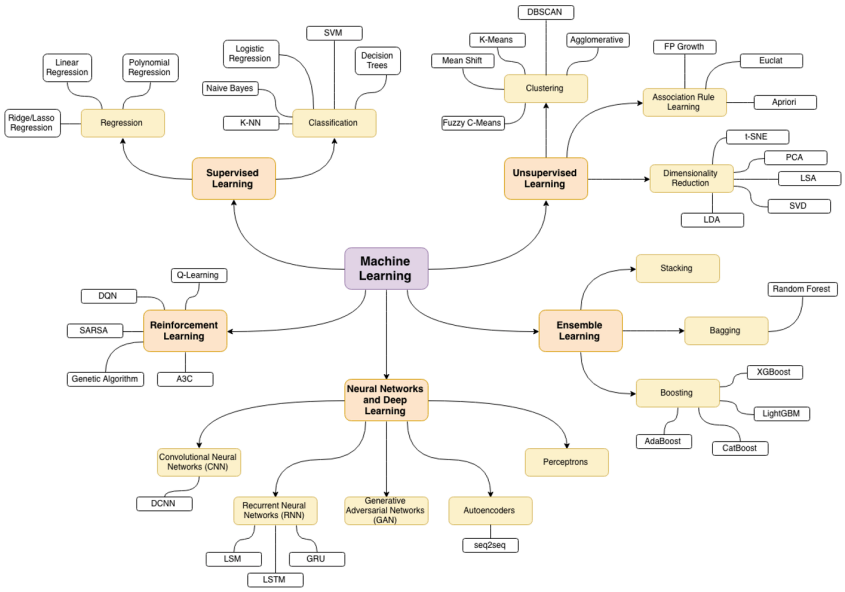

图 2: 各个类别常用模型（原图）

而如何分析问题并且选择模型具体如下图所示:

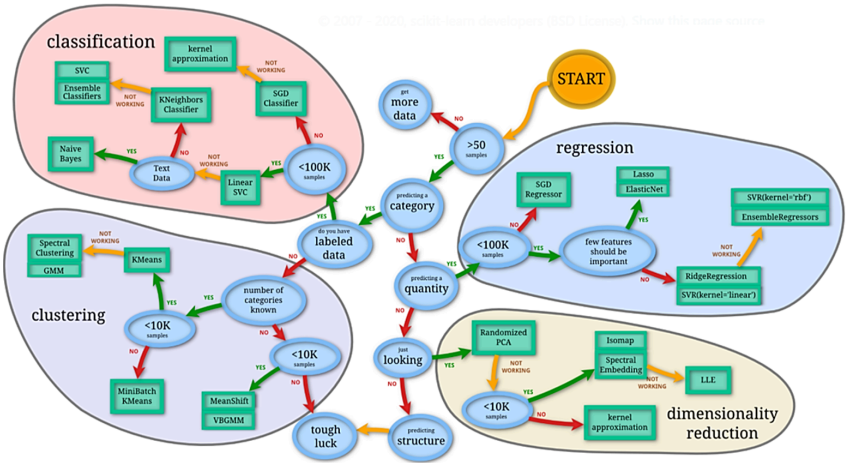

图 3: 各类别模型选择（原图）

最后推荐https://whimsical.com/CA7f3ykvXpnJ9Az32vYXva，里面对于机器学习脉络有着非常详细的叙述

2.2 学习资料

书籍

- 入门：周志华的《机器学习》
- 公式和原理：李航的《统计学习方法》
- 实战：《机器学习实战》，作者是 Peter Harrington；《Python 机器学习及实践从零开始通往 Kaggle 竞赛之路

网站

- 机器学习简介：https://blog.csdn.net/hohaizx/article/details/80584307
- 机器学习十大算法：https://blog.csdn.net/qq_42379006/article/details/80741808
- 英文版的入门介绍，可以当课外阅读看：https://machinelearningmastery.com/machine-learning-roadmap-your-self-study-guide-to-machine-learning/
- 人工智能、机器学习和深度学习的区别：https://www.zhihu.com/question/57770020

学习课程和 ppt

- 基础版：http://www.cs.cmu.edu/ tom/10701_sp11/lectures.shtml
  课程内容：Decision Trees；Probability and Estimation；Naive Bayes；Logistic Regression；Linear Regression； Practical Issues: Feature selection etc.
- 进阶版：http://work.caltech.edu/lectures.html
  课程内容：Theory of Generalization；The VC Dimension；Bias-Variance Tradeoff；The Linear Model II； Neural Networks； Overfitting； Regularization； Validation etc.
- 吴恩达机器学习：
  https://www.bilibili.com/video/BV164411b7dx
- 李宏毅机器学习：
  https://www.bilibili.com/video/BV1JE411g7XF

2.3 相关论文

- 线性回归：Linear Regression Analysis: Theory and Computing by Xin Yan, Xiao Gang Su
- 逻辑斯蒂回归：Applied logistic regression
- VC 维：Learnability and the Vapnik–Chervonenkis dimension
- 支持向量机 (SVM)：
  - A training algorithm for optimal margin classifiers
  - Support vector networks
  - Extracting support data for a given task
  - Advances in Kernel Methods
- K 近邻算法 (KNN)：An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression
- 决策树与随机森林：
  - Classification and Regression Trees
  - Induction of decision trees
  - Random Forests
- Boosting：
  - Boosting a weak learning algorithm by majority
  - A decision-theoretic generalization of on-line learning and an application to boosting.
  - An adaptive version of the boost by majority algorithm
  - LightGBM: A Highly Efficient Gradient Boosting Decision Tree.

- 主成分分析 (PCA)：Kernel PCA and de-noising in feature spaces
- K-means：A K-Means Clustering Algorithm
- 神经网络：
  - The Perceptron: A Model for Brain Functioning
  - Learning representations by back-propagating errors
  - Gradient-based learning applied to document recognition
  - Backpropagation through time: what it does and how to do it
- 优化理论：
  - On the momentum term in gradient descent learning algorithms
  - Stochastic Gradient Descent Tricks.
  - Adam: a method for stochastic optimization

3 计算机视觉

3.1 传统算法的识别和跟踪

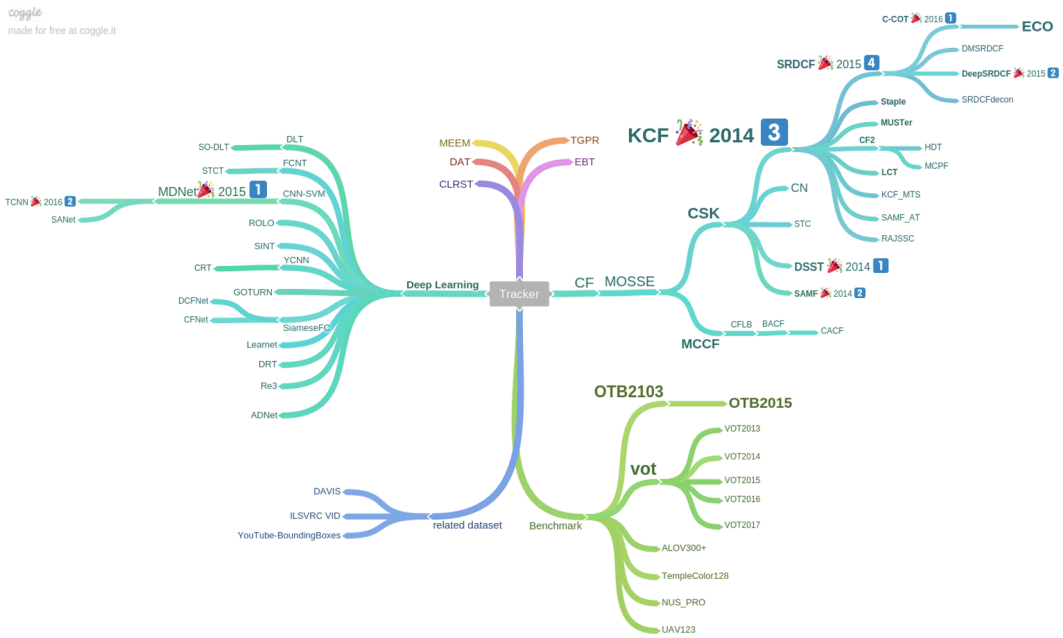

图 4: 传统识别跟踪算法的知识脉络图

对图片或者视频内容的识别和跟踪是一个相当普遍的任务，发展时间很长，历经了从传统方法到机器学习方法的转变，但是在小型无人机或者无人车等计算资源有限的环境下，一般还是考虑采取传统的方法。传统方法里应该就是两个最有代表性的算法，**TLD** 和 **KCF**。

TLD TLD 叫 Tracking-Learning-Detection，主要由追踪器（Tracker）、检测器（Detector）和学习（Learning）三个模块构成。TLD 是对视频中未知物体的长时间跟踪的算法，未知物体指的是任意的物体，在开始追踪之前不知道哪个物体是目标。物体始终可见时使用追踪器连续跟踪，检测器对图像进行全面扫描更新追踪器的初始位置，学习器迭代训练分类器，改善检测器的精度。

KCF　第二个最有代表的算法就是 KCF，kernel correlation filter，核相关滤波算法，其核心思想是设计一个滤波的模板，使得这个模板作用在整张图片上时，作用到跟踪目标上得到的响应最大，在背景上响应较小。而这个滤波模板则能够通过正负样本的循环采取和岭回归进行不断训练，从而对目标进行长时间的跟踪。

总的来说，传统方法是从对图像的理解上出发，基于严格的数学推导去识别和跟踪视频中移动的物体，涉及图像的傅立叶变换，线性与非线性回归，和相当多的机器学习（不是深度学习）知识。在后人不断改进下，应用范围也不断增广，如 DS-KCF，能用于在深度图上进行跟踪，鲁棒性与实时性也不断提高，在机器人领域有着广泛的应用价值。

之后深度学习开始火起来，使用卷积神经网络的图像检测与跟踪算法层出不穷，像大名鼎鼎、不断迭代更新的 yolo 系列，但是基于学习的方法一是需要大量样本训练，二是泛化性不高，三是对设备的要求较高，因此基于自监督的方法越来越受到人们的关注，个人没有很积极关注，有兴趣的同学可以好好研究。

上面提到的经典论文：

- TLD:http://vision.stanford.edu/teaching/cs231b_spring1415/papers/Kalal-PAMI.pdf
- KCF:https://arxiv.org/pdf/1404.7584
- DS-KCF:https://pdfs.semanticscholar.org/3aa4/573f8735e984c25db8b2a805235bf22dc042.pdf

相关的网课与书籍

- 数字图像处理（冈萨雷斯版）大部头书籍，很难啃，一般当百科全书用。
- 斯坦福大学 CS231n: Convolutional Neural Networks for Visual Recognition。经典 CS231n，主要涉及深度学习方法。
- 普林斯顿大学的 tracking 的 benchmark: http://tracking.cs.princeton.edu/

3.2　目标检测

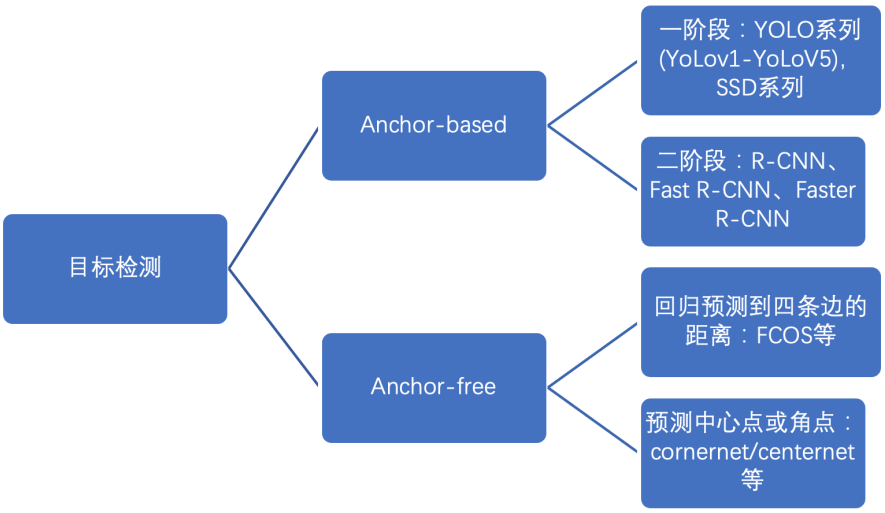

图 5: 目标检测知识脉络图

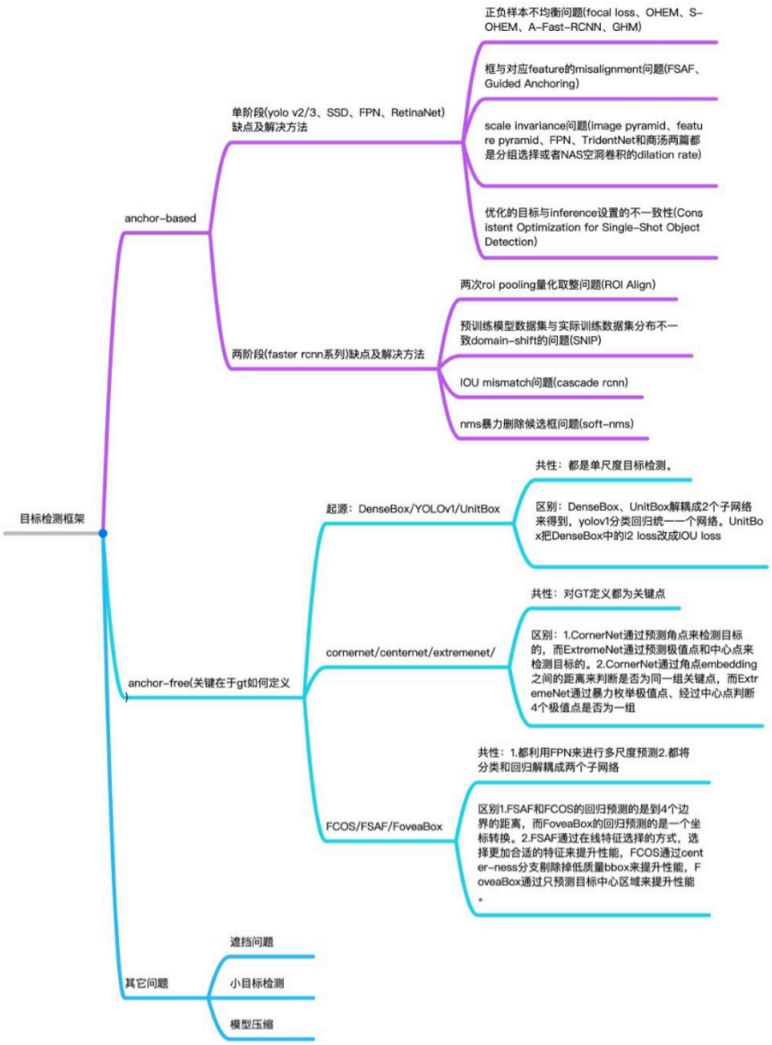

图 6: 目标检测详细知识脉络图

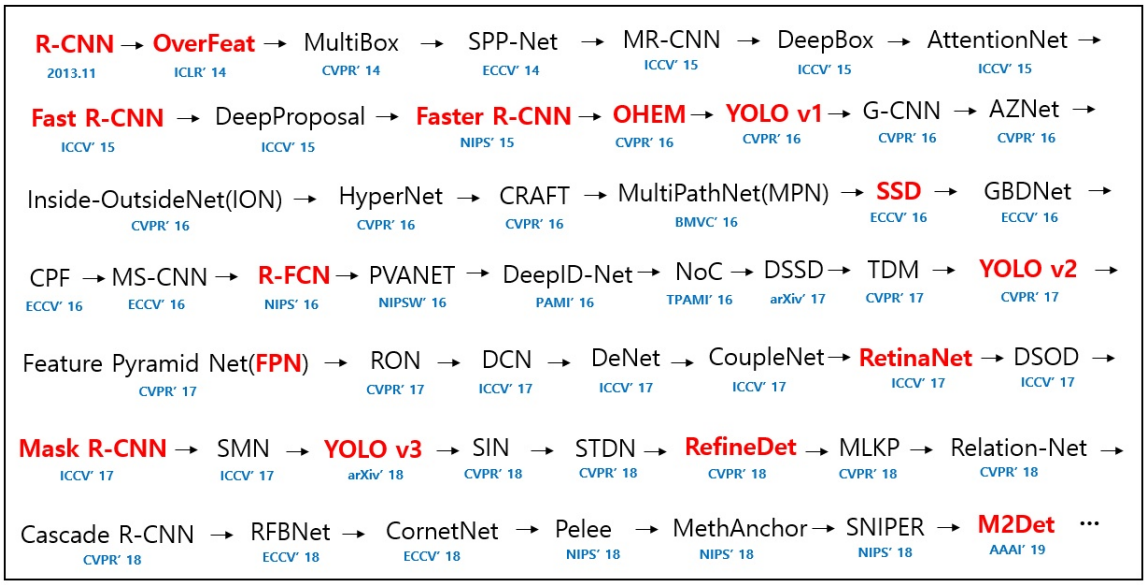

图 7: 目标检测算法发展时间线

性能对比汇总表（评价指标 mAP）：

Performance table

FPS(Speed) index is related to the hardware spec(e.g. CPU, GPU, RAM, etc), so it is hard to make an equal comparison. The solution is to measure the performance of all models on hardware with equivalent specifications, but it is very difficult and time consuming.

Detector	VOC07 (mAP@IoU=0.5)	VOC12 (mAP@IoU=0.5)	COCO (mAP@IoU=0.5:0.95)	Published In
R-CNN	58.5	-	-	CVPR'14
SPP-Net	59.2	-	-	ECCV'14
MR-CNN	78.2 (07+12)	73.9 (07+12)	-	ICCV'15
Fast R-CNN	70.0 (07+12)	68.4 (07++12)	19.7	ICCV'15
Faster R-CNN	73.2 (07+12)	70.4 (07++12)	21.9	NIPS'15
YOLO v1	66.4 (07+12)	57.9 (07++12)	-	CVPR'16
G-CNN	66.8	66.4 (07+12)	-	CVPR'16
AZNet	70.4	-	22.3	CVPR'16
ION	80.1	77.9	33.1	CVPR'16
HyperNet	76.3 (07+12)	71.4 (07++12)	-	CVPR'16
OHEM	78.9 (07+12)	76.3 (07++12)	22.4	CVPR'16
MPN	-	-	33.2	BMVC'16
SSD	76.8 (07+12)	74.9 (07++12)	31.2	ECCV'16
GBDNet	77.2 (07+12)	-	27.0	ECCV'16
CPF	76.4 (07+12)	72.6 (07++12)	-	ECCV'16
R-FCN	79.5 (07+12)	77.6 (07++12)	29.9	NIPS'16
DeepID-Net	69.0	-	-	PAMI'16
NoC	71.6 (07+12)	68.8 (07+12)	27.2	TPAMI'16
DSSD	81.5 (07+12)	80.0 (07++12)	33.2	arXiv'17
TDM	-	-	37.3	CVPR'17
FPN	-	-	36.2	CVPR'17
YOLO v2	78.6 (07+12)	73.4 (07++12)	-	CVPR'17

图 8: 检测算法 benchmark

经典论文：https://github.com/hoya012/deep_learning_object_detection 看这个链接就够了，从 2014-2020 按照时间线，列举了有重大贡献的论文和其开源代码。

其他网课，书籍，资料（经典的 Review/Survey）推荐：

1. 公开课：斯坦福大学 CS231n: Convolutional Neural Networks for Visual Recognition
链接：http://cs231n.stanford.edu/syllabus.html
2. 吴恩达深度学习第 4 篇计算机视觉 (3.9 目标检测)
链接：https://www.bilibili.com/video/BV1gb411j7Bs?from=search&seid=722731644245107858
3. 天池-目标检测 https://tianchi.aliyun.com/course/video?liveId=41141

3.3 目标追踪

现有的目标追踪主要分为两个任务方向，即单目标追踪（single object tracking，SOT）与多目标追踪（multi object tracking）。主要针对的对象是行人、车辆。p.s. MOT 部分，自 DeepSort

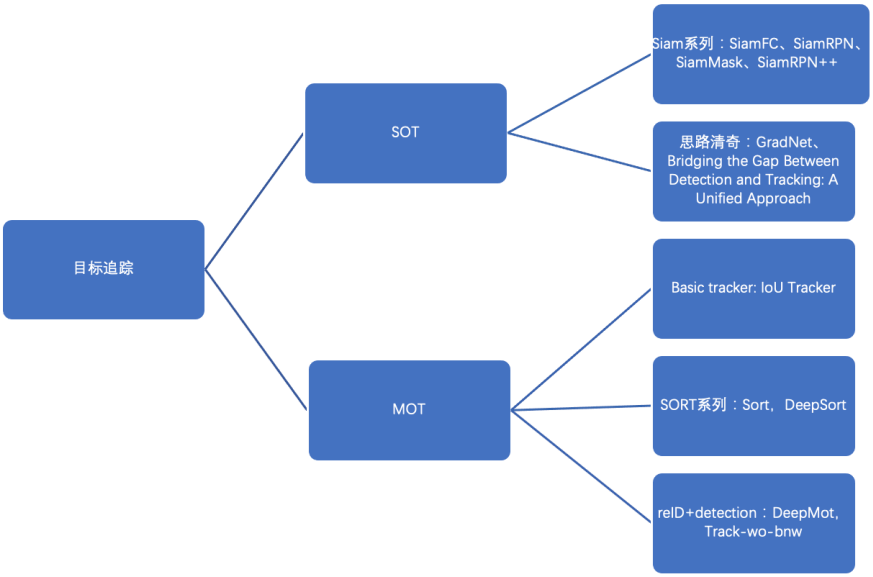

图 9: 目标追踪知识脉络

之后，大家慢慢把多目标追踪改造成一个 reid 和 detection 的联合任务（用于解决工程情况下容易出现的长时间遮挡情况）

经典论文

1. SiamFC：

   论文地址：https://arxiv.org/pdf/1606.09549.pdf（ECCV 2016）

   代码：https://github.com/torrvision/siamfc-tf

   主要贡献：单目标追踪 siamese 系列的开山之作，目前大半的 SOT 任务都使用 siamese 网络完成，主要的思路是将跟踪当作匹配问题，帮助模型高效精确定位

2. GradNe：

   论文地址：https://arxiv.org/pdf/1909.06800.pdf（ICCV 2019）

   代码：https://github.com/LPXTT/GradNet-Tensorflow

   主要贡献：针对 Siam 系列网络不实时更新背景信息的问题，使用在线 BP 熏蒸 template，帮助模型学习背景信息，提升模型的精度。

3. IoU Tracker：

   论文地址：http://elvera.nue.tu-berlin.de/files/1517Bochinski2017.pdf（AVSS 2017）

   代码：https://github.com/bochinski/iou-tracker（非官方复现）

   主要贡献：提出了一种简单高效的跟踪模型——IOU Tracker。速度快、基于 TBD(tracking-by-detection)、不需要图像信息。

4. Sort：

   论文地址：https://arxiv.org/pdf/1602.00763.pdf（ICIP 2017）

   代码：https://github.com/abewley/sort

   主要贡献：引入 kalman 滤波预测目标状态，结合检测框位置和 IOU 的匈牙利算法进行检测框的匹配，提升算法精度，且检测较为高效。

5. DeepSort：论文地址：https://arxiv.org/pdf/1703.07402.pdf（WACV 2018）

   代码：https://github.com/nwojke/deep_sort

   主要贡献：在匹配阶段引入马氏距离和表观匹配（reid），结合运动和图像信息减少了预测时出现的 ID switch。

6. Tracking without bells and whistles：

   论文地址：https://arxiv.org/pdf/1903.05625.pdf（ICCV 2019）

   代码：https://github.com/phil-bergmann/tracking_wo_bnw

   主要贡献：作提出将目标检测器转换为追踪器，不特意去对当前存在的遮挡，重识别和运动预测进行优化而完成跟踪任务。且不需要进行训练和优化。利用对象检测器的边界框回归来预测对象在下一帧中的新位置，通过简单的重新识别和相机运动补偿对其进行扩展，展示了跟踪器的可扩展。

目标追踪数据集/比赛/benchmark

- PETS2009 : An old dataset.
- MOT dataset : A dataset for multi-person detection and tracking, mostly used.
- UA-DETRAC : A dataset for multi-car detection and tracking.
- AVSS2018 Challenge : AVSS2018 Challenge based on UA-DETRAC is opened!
- DukeMTMC : A dataset for multi-camera multi-person tracking.
- PoseTrack: A dataset for multi-person pose tracking.
- NVIDIA AI CITY Challenge: Challenges including "Traffic Flow Analysis", "Anomaly Detection" and "Multi-sensor Vehicle Detection and Reidentification", you may find some insteresting codes on their Github repos
- Vis Drone: Tracking videos captured by drone-mounted cameras.
- JTA Dataset: A huge dataset for pedestrian pose estimation and tracking in urban scenarios created by exploiting the highly photorealistic video game Grand Theft Auto V developed by Rockstar North.
- Path Track: A new dataset with many scenes.
- MOTS MOTS: Multi-Object Tracking and Segmentation. In CVPR 2019

Reference

[1] https://github.com/SpyderXu/multi-object-tracking-paper-list

[2] https://zhuanlan.zhihu.com/p/65177442

[3] https://link.zhihu.com/?target=https%3A//github.com/huanglianghua/mot-papers/blob/master/README.md

3.4 图像分割

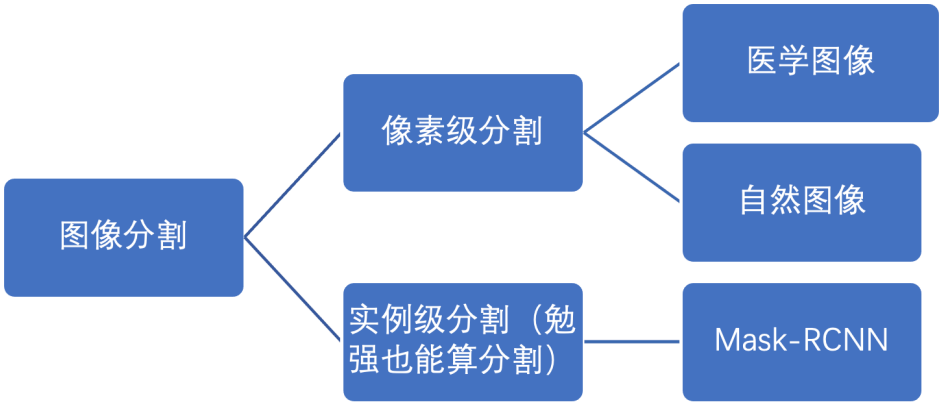

图 10: 图像分割主要知识脉络（其实主要就是像素级分割）

分割是个很普遍存在的任务，相关的东西很多很杂乱，所以没有发展脉络图。下面主要说一下几个里程碑。

深度学习火起来之前像素级分割是很难的任务，但是后来卷积在 AlexNet 上大放异彩后迅速被用到了分割上，于是现在分割可以算是一个比较简单的东西了。实际上，像素级的分割无非就是对每个像素做分类，所以在很多时候损失函数依然可以沿用分类任务里常见的交叉熵损失函数。至今，分割依然保持着和分类任务极为相近的思路，都是先用卷积提取特征，只不过分类任务是后面接全连接，而分割后面是接卷积和上采样等等操作罢了。

这种把全连接换成卷积的网络就是全卷积网络，这是第一个里程碑，也是语义分割的开山之作。

如果硬要说第二个里程碑的话，那可能就是空洞卷积和特征金字塔，这些都是 Deeplab 系列网络提出来的新概念。其中空洞卷积就是在普通卷积核中间插入孔洞，使其感受野（就是它一次计算能覆盖的像素范围）增大同时参数量不变；金字塔池化就是多个卷积层叠呈金字塔形，并且允许一层存在多个不同大小和形式的卷积核，只要在最后将卷积核的输出结果合并到一起就好了。不过这个东西实际用起来效果并不是那么稳定的，我个人感觉算不上里程碑吧，但现在只要是相关的课程都会讲。

我个人认为的一个非常重要的里程碑是 UNet 的诞生。该网络本来是用于医学图像分割的，但是经过千锤百炼证明了自己在自然图像上照样管用，超分辨也经常用它。这个网络极为鲁棒、通用性高、结构也相对简单，强烈推荐大家干什么都可以用它试试。Unet 本身是 Encoder-Decoder 的设计，但是在 Encoder 和 Decoder 之间采用了跨层连接的设计，这样网络能够很好地融合低维特征和高维特征，非常适合医学图像里的一些类似肿瘤分割的任务。

图 11: 图像分割算法发展时间线

上面提到的经典论文

- FCN

  论文地址：https://arxiv.org/abs/1411.4038

- Deeplab 系列以最新的 Deeplab v3+ 为例（实际上也是好几年前了）

  论文地址：https://arxiv.org/pdf/1802.02611.pdf

  代码地址：https://github.com/tensorflow/models/tree/master/research/deeplab

- UNet

  论文地址：https://arxiv.org/pdf/1505.04597.pdf

  代码地址：https://github.com/jakeret/tf_unet

另外，可以用一些如 VOC，Cityscapes 之类的数据集跑一下就知道是什么个情况了。

4 自然语言处理

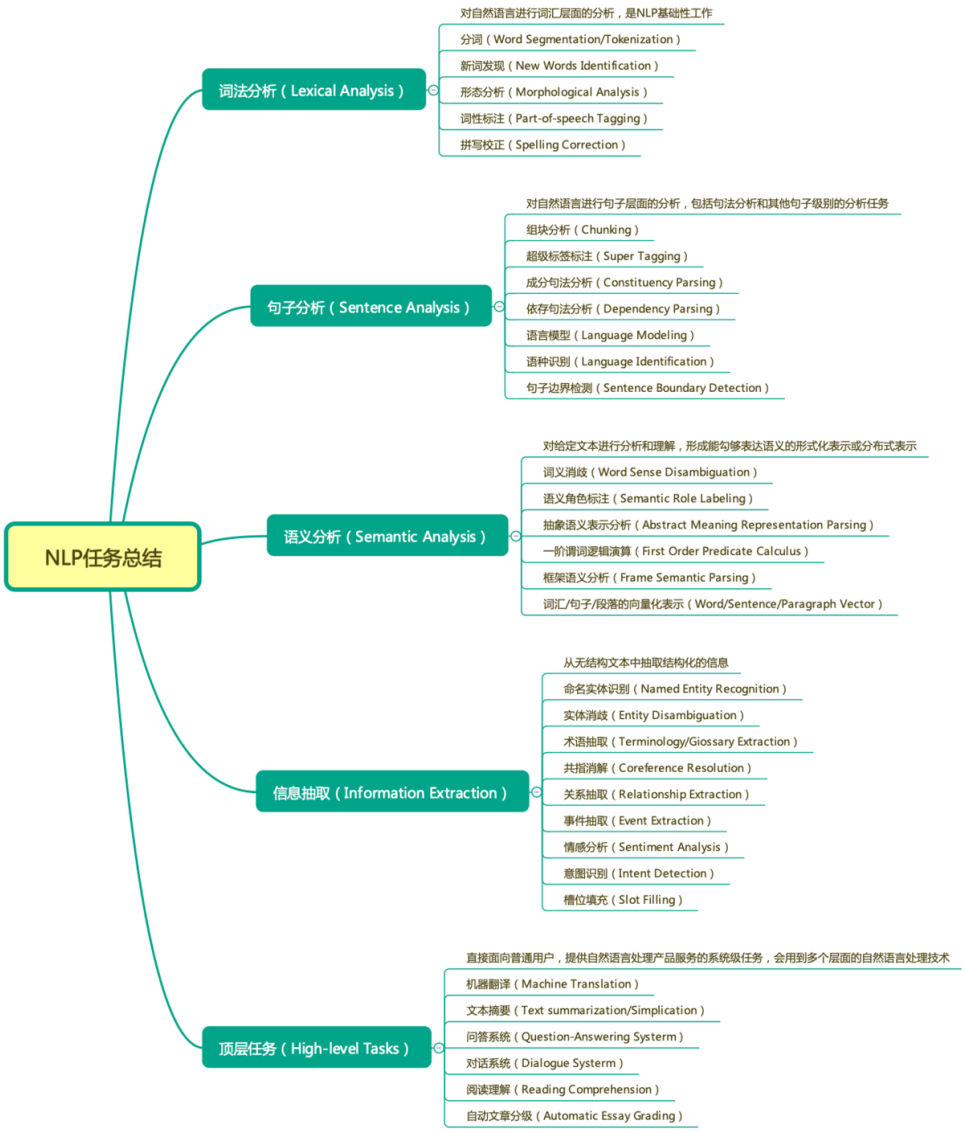

图 12: 自然语言处理任务分类以及子领域（原图）

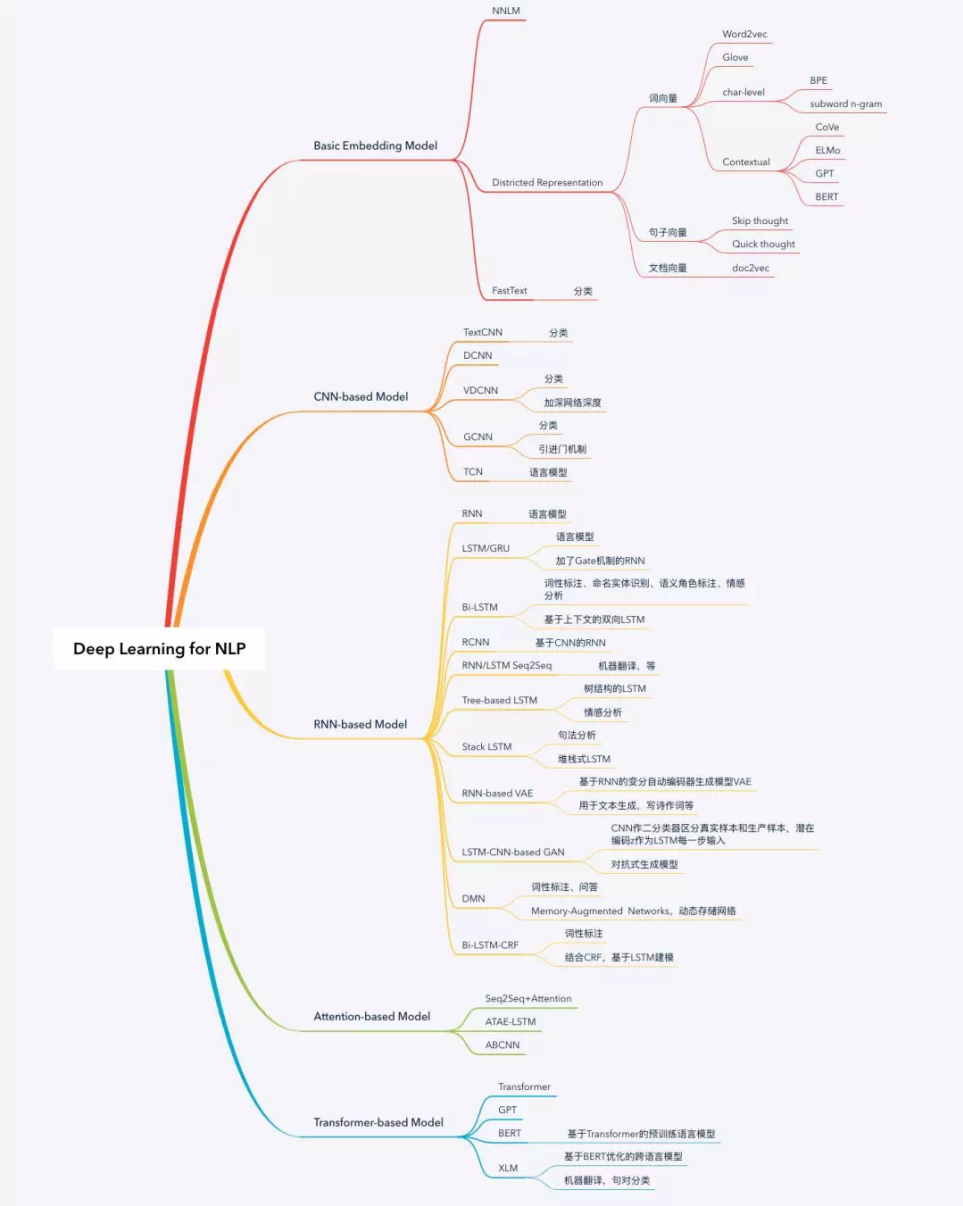

图 13: 自然语言处理知识脉络图（原图）

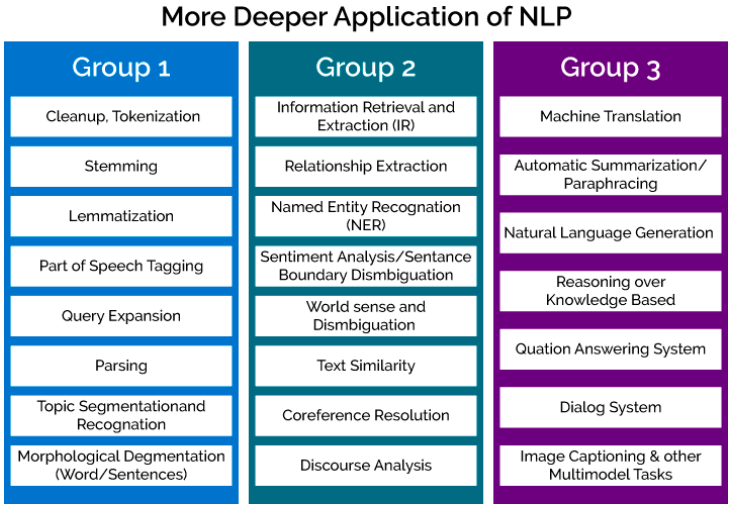

图 14: Group1 到 Group3：底层/基础任务到顶层/具体应用（原图）

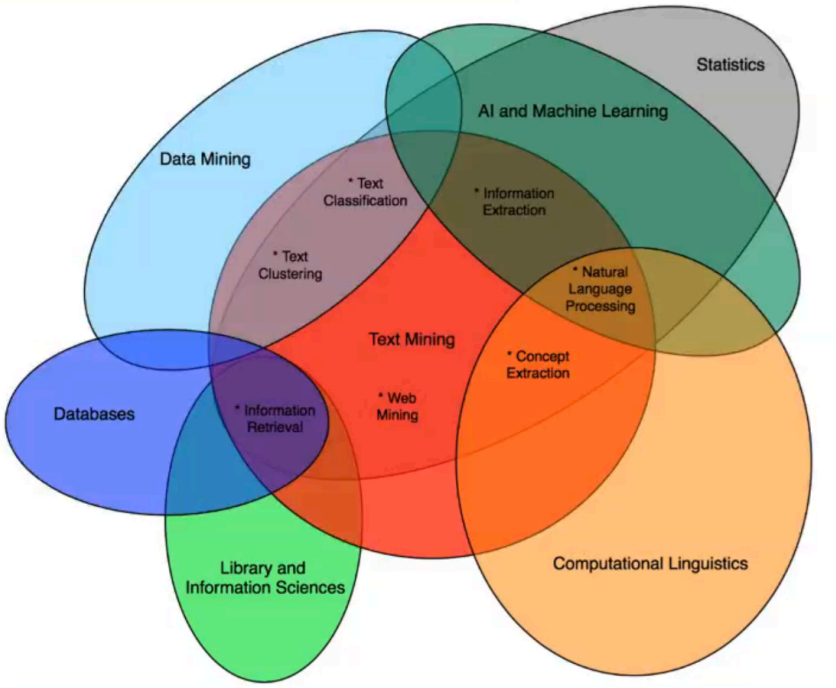

图 15: 自然语言处理任务子领域之间的交叉关系。原图

4.1 经典论文

1. Seq2Seq + Attention

   论文：https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

   实现：https://github.com/google/seq2seq

   贡献：Seq2Seq 模型与注意力机制的结合。

2. Transformer

   论文：https://arxiv.org/pdf/1706.03762.pdf

   tf 实现：https://github.com/Kyubyong/transformer

   pytorch 实现：https://github.com/jadore801120/attention-is-all-you-need-pytorch

   贡献：用全 attention 的结构代替了 lstm，在翻译任务上取得了更好的成绩。Transformer 改进了 RNN 最被人诟病的训练慢的缺点，利用 self-attention 机制实现快速并行。并且 Transformer 可以增加到非常深的深度。

3. ELMo

   论文：https://arxiv.org/pdf/1802.05365.pdf

   贡献：根据当前上下文对 Word Embedding 动态调整，而非过去的静态词嵌入。

4. BERT

   论文：https://arxiv.org/abs/1810.04805

   tensorflow 实现：https://github.com/google-research/bert

   pytorch 实现：https://github.com/huggingface/transformers

   贡献：1) 使用了 Transformer 作为算法的主要框架，Transformer 能更彻底的捕捉语句中的双向关系；

2) 使用了 Mask Language Model(MLM) 和 Next Sentence Prediction(NSP) 的多任务训练目标；

3) 使用更强大的机器训练更大规模的数据，使 BERT 的结果达到了全新的高度，并且 Google 开源了 BERT 模型，用户可以直接使用 BERT 作为 Word2Vec 的转换矩阵并高效的将其应用到自己的任务中。

5. XLM

论文：https://arxiv.org/abs/1901.07291

实现：https://github.com/facebookresearch/XLM

贡献：1、提出了一种新的无监督方法。使用跨语言语言建模来学习跨语言表示，并研究了两种单语预训练的目标函数。

2、提出一个新的监督学习目标。当有平行语料时，该目标可以改进跨语言的预训练。

3、本文的模型在跨语言分类、无监督机器翻译和有监督机器翻译方面都显著优于以往的最优结果。

4、本文实验表明跨语言模型对于 low-resource 语种数据集，也能够显著改善其他语种的困惑度 (perplexity)。

6. XLNet

论文：https://arxiv.org/abs/1906.08237

实现：https://github.com/zihangdai/xlnet

贡献：1）与 Bert 采取 De-noising Autoencoder 方式不同的新的预训练目标：Permutation Language Model(简称 PLM)；这个可以理解为在自回归 LM 模式下，如何采取具体手段，来融入双向语言模型。这个是 XLNet 在模型角度比较大的贡献，确实也打开了 NLP 中两阶段模式潮流的一个新思路。

2）引入了 Transformer-XL 的主要思路：相对位置编码以及分段 RNN 机制。实践已经证明这两点对于长文档任务是很有帮助的；

3）加大增加了预训练阶段使用的数据规模；Bert 使用的预训练数据是 BooksCorpus 和英文 Wiki 数据，大小 13G。XLNet 除了使用这些数据外，另外引入了 Giga5，ClueWeb 以及 Common Crawl 数据，并排掉了其中的一些低质量数据，大小分别是 16G,19G 和 78G。可以看出，在预训练阶段极大扩充了数据规模，并对质量进行了筛选过滤。这个明显走的是 GPT2.0 的路线。

7. ERNIE

论文：https://arxiv.org/pdf/1905.07129.pdf

实现：https://github.com/thunlp/ERNIE

贡献：将知识图谱的信息加入到 BERT 模型的训练中，这样模型就可以从大规模的文本语料和先验知识丰富的知识图谱中学习到字、词、句以及知识表示等内容，让 BERT 掌握更多的人类先验知识。

8. NNLM：https://jmlr.org/papers/volume3/bengio03a/bengio03a.pdf
9. CNN 在 NLP：https://arxiv.org/abs/1408.5882

4.2 公开课

1. 汤斯亮-自然语言处理

   内网访问：https://10.214.143.253/

2. CS224n

   主页：http://web.stanford.edu/class/cs224n/index.html

   B 站链接：https://www.bilibili.com/video/av46216519/

   介绍：斯坦福相当不错的 NLP 入门课程，包括词嵌入、依存关系解析、神经机器翻译、语音识别、Transformer 、预训练表征和语义消歧等。课程笔记、PPT 资料以及作业都可以在主页上获取。

3. CS 11-747 主页：http://phontron.com/class/nn4nlp2019/

   B 站链接：https://www.bilibili.com/video/av40929856/

   介绍：伯克利的，包括建模不同长度和结构的句子、处理大型自然语言数据、半监督和无监督学习、结构化预测和多语言建模、ELMo/BERT 上下文词表示、模型可解释性等等。

4.3 书籍

- 《Python 自然语言处理》（python NLTK 库配套）: 极好的 NLP 入门图书，涉及到自然语言处理的方方面面，包括分词、词典、词性标注、NER、语法分析、文本分类、语料库等等。缺点：未涉及中文语言处理。
- 《自然语言处理入门》:HanLP 作者何晗，一个从零开始学 NLP 的大佬的历程与经验。
- 《NLP 汉语自然语言处理》: 从语言学讲到自然语言处理，比较全面。

4.4 其他资料

1. HanLP

   链接：https://github.com/hankcs/HanLP

   介绍：基于 TensorFlow 2.x 中文分词词性标注命名实体识别依存句法分析语义依存分析新词发现关键词短语提取自动摘要文本分类聚类拼音简繁转换自然语言处理等。

2. NLP tutorial

   链接：https://github.com/graykode/nlp-tutorial

   介绍：NLP 的 pytorch 小教程，其实似乎没啥用。

3. thunlp

   链接：https://github.com/thunlp/

   介绍：清华的 NLP 项目组开源。

4. 中文 BERT 预训练模型

   链接：https://github.com/ymcui/Chinese-BERT-wwm

   介绍：哈工大的中文 NLP 预训练模型。

5. huggingface transformer

   链接：https://huggingface.co/transformers/index.html

介绍：非常全的 NLP 预训练模型代码库。huggingface 还有一个 model zoo，上面有非常多的可供使用的预训练模型。

6. https://github.com/fastnlp/fastNLP
7. https://github.com/FudanNLP/nlp-beginner

4.5 国际顶级会议

- ACL - Meeting of the Association for Computational Linguistics
- EMNLP - Conference on Empirical Methods in Natural Language Processing
- NAACL –Conference of the North American Chapter of the Association for Computational Linguistics

5 图网络

5.1 斯坦福 CS224W:Machine Learning with Graphs

斯坦福大学公开课，前身是 Analysis of Networks，2019 年课改，所以课程中留有不少传统网络的东西。教授发音清晰（俄国口音真的很清晰，音节和音节之间分得明明白白）。
课程网站：http://web.stanford.edu/class/cs224w/
官方笔记：https://snap-stanford.github.io/cs224w-notes/
深度学习相关部分用加粗标出。

序号	名称	内容
1	Introduction Structure of Graphs	图的基本概念：度、表示方法、连通度.
2	Measuring Networks and Random Graphs	图的统计量：度分布、路径长度、聚类系数、连通度.
3	Snap.py and Google Cloud Tutorial	介绍斯坦福大学的图分析框架 snap.py（类似 networkx 的工具）以及谷歌云的使用教程.
4	Motifs and Structral Roles in Network	图的局部结构.
5	Community Structure in Networks	社区发现算法.
6	Review of Linear Algebra Probability and Proof techniques	顾名思义，可以跳过.
7	Spectral Clustering	谱聚类!
8	Message Passing and Node classification	传统的节点分类方法，主要关注 BP（信念传播）算法.
9	**Graph Representation Learning**	**图嵌入方法：Shallow, DeepWalk, TransE, node2vec.**
10	**Graph Neural Networks**	**GNN 终于来了：GCN, GraphSage, GAT.**
11	**Graph Neural Networks Hands-on Session**	**动手实践，使用 Pytorch 的图学习框架 Pytorch_geometric.**

12	**Deep Generative Models for Graphs**	**GraphRNN：用 RNN 做图生成（教授自己团队的成果），另外末尾还有一个非常有意思的东西：有目标导向的图生成（用于生成药物），结合了强化学习和对抗训练，同样是教授自己团队的成果.**
13	Link Analysis PageRank	大名鼎鼎的 PageRank 算法.
14	Networks Effects and Cascading Behavior	图的传播效应和级联行为，该节主要是基于博弈论的方法（如疾病传播、假新闻传播、故障传播）.
15	Probabilistic Contagion and Models of Influence	图传播的动力学模型：概率传染模型（传染病模型、谣言传播模型
16	Influence Maximization in Networks	影响最大化：线性阈值模型和独立级联模型
17	Outbreak Detection in Networks	爆发点检测：爬山算法和 CELF 算法
18	Network Evolution	动态网络：图统计量的变化、森林火灾模型、时序网络（时序 PageRank）
19	**Reasoning over Knowledge Graphs**	**向量空间中知识图谱推理（Query2box）**
20	**Limitations on Graph Neural Networks**	**讲了 GNN 的 2 个缺陷：1. 对于不同结构的图可能会产生相同的结果（即不是单射），由此提出了 GIN；2. 和其他 DNN 一样，GNN 也会被对抗攻击. 另外提出了 GNN 一些未来的工作.**
21	**Applications of Graph Neural Networks**	**讲了 GNN 的三个应用：主要篇幅在讲用于推荐系统的 PinSage，另外还有用于异质网络的 Decagon，以及第 12 节讲过的药物生成 GCPN.**

5.2 其他课程和 tutorial

1. 宾夕法尼亚大学，ESE680: Graph Neural Networks
   宾大的新课程，仍在更新当中。相较斯坦福大学的课程更为 machine learning 一些。老师长得像小罗伯特 · 唐尼。
   课程网站：https://gnn.seas.upenn.edu/
2. NLPCC Tutorial
   NLPCC2020 的图神经网络教程，内容很新，并且与其他 AI 领域（CV、NLP）联系紧密。
   slides 地址：http://tcci.ccf.org.cn/conference/2020/dldoc/tutorial_3.pdf

5.3 一些综述

1. Deep Learning on Graphs: A Survey
2. Graph Neural Networks: A Review of Methods and Applications
3. A Comprehensive Survey on Graph Neural Networks

5.4 图网络知识脉络

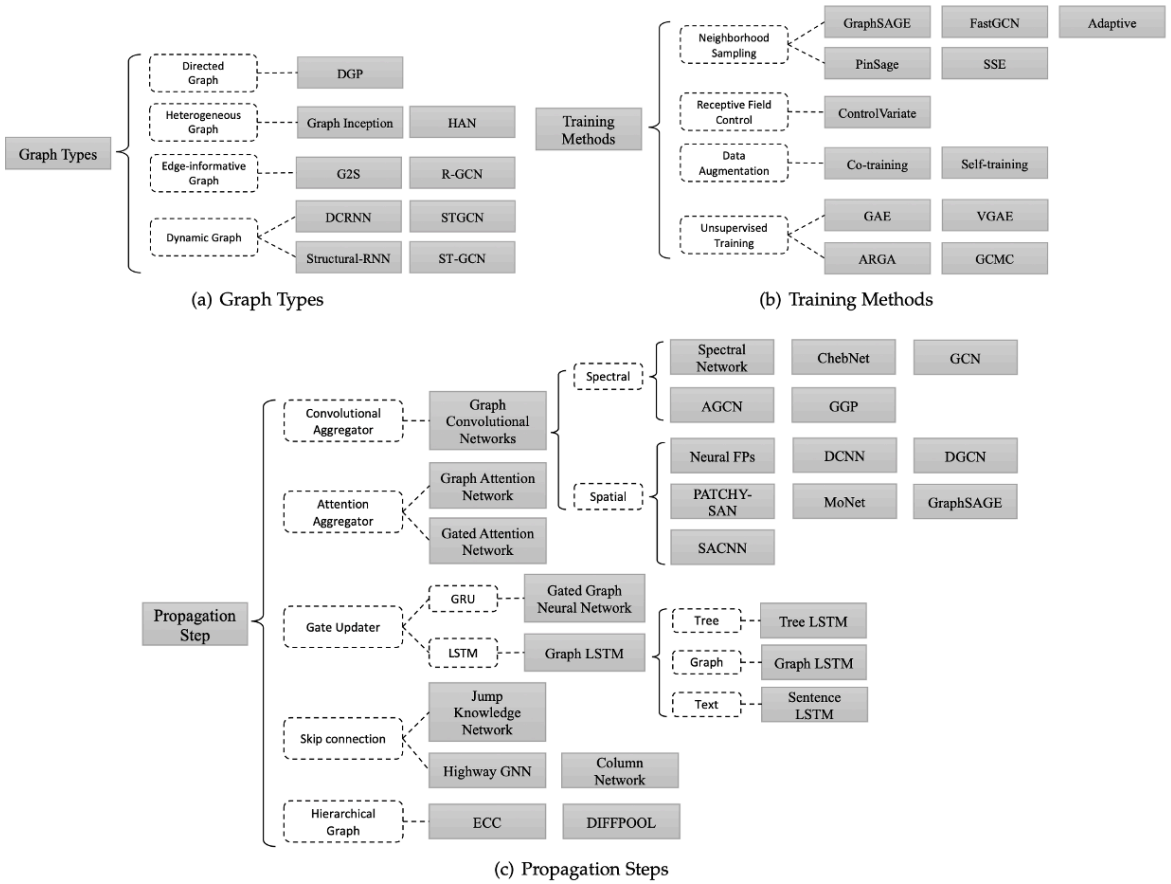

图 16: 图网络知识脉络图

5.5 必读论文

- Ng, A. Y., Jordan, M. I., & Weiss, Y. (2002). On spectral clustering: Analysis and an algorithm. In Advances in neural information processing systems (pp. 849-856).
  吴恩达老师引用量第二的文章，谱聚类的代表作。
- Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., & Vandergheynst, P. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE signal processing magazine, 30(3), 83-98.
  图信号处理的基本方法，算是基础。
- Defferrard, M., Bresson, X., & Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In Advances in neural information processing systems (pp. 3844-3852).
  ChebNet，图网络中频域卷积的代表作。
- Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.
  GCN 的原始论文，对 ChebNet 做了简化和修改。

- Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. In Advances in neural information processing systems (pp. 1024-1034).
  GraphSage，针对 GCN 只能处理固定图的缺陷，提出了一种归纳（inductive）学习的方法，使模型可以处理没有见过的节点。
- Velikovi, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.
  GAT，图上的注意力机制。
- Chami, I., Ying, Z., Ré, C., & Leskovec, J. (2019). Hyperbolic graph convolutional neural networks. In Advances in neural information processing systems (pp. 4868-4879).
  在双曲空间中做图嵌入。

6 强化学习

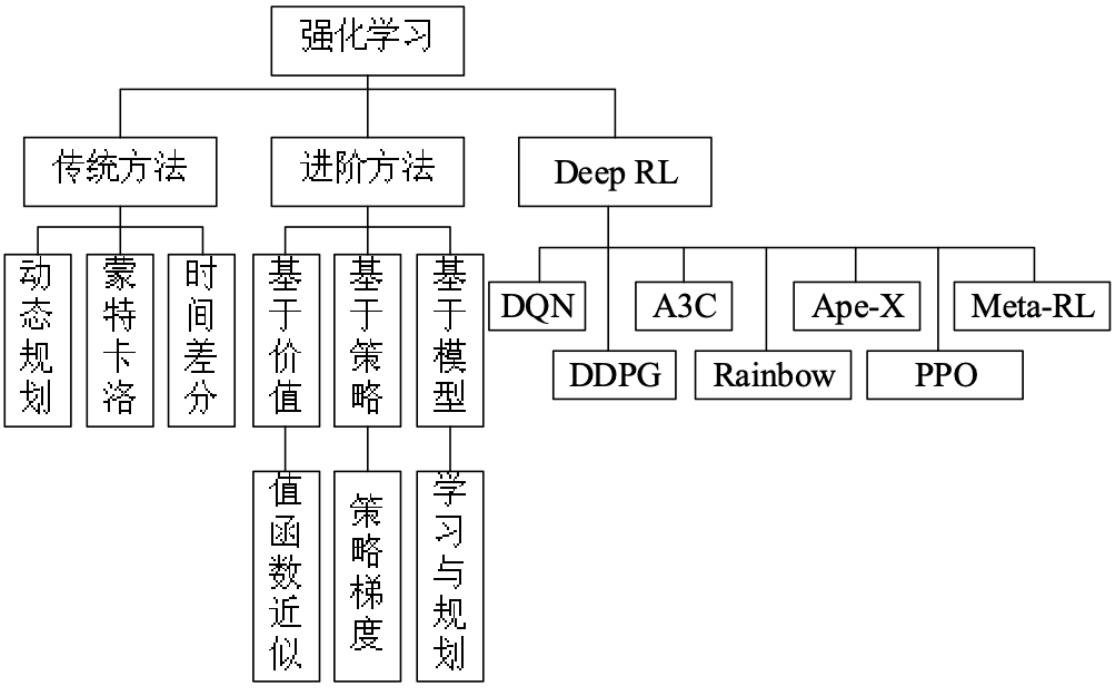

图 17: 强化学习知识脉络图

学习路径：马尔科夫过程-> 贝尔曼方程-> 动态规划法-> 蒙特卡洛法-> 时间差分法->Sarsa 算法->Q-learning 算法-> 值函数近似法-> 策略梯度法-> 基于模型的强化学习-> 蒙特卡洛树搜索->DQN->DDPG->A3C->Rainbow->Ape-X->PPO->Meta-RL

6.1 网课

1. 强化学习论文解读-10 mins paper，上海交大伯禹讲论文
   B 站伯禹账号：https://space.bilibili.com/447303411
   DQN：https://www.bilibili.com/video/BV1H4411d7Mh

DDPG：https://www.bilibili.com/video/BV18E411f7zd

TD3：https://www.bilibili.com/video/BV1nE41117CR

PPO：https://www.bilibili.com/video/BV1pE41117zu

Soft Actor-Critic：https://www.bilibili.com/video/BV1pE41117q1

QMIX：https://www.bilibili.com/video/BV1pE411176N

VDRL：https://www.bilibili.com/video/BV1nE41117qu

Nash Q-Learning：https://www.bilibili.com/video/BV1mE411b7fW

RL with Deep Energy-Based Policies：https://www.bilibili.com/video/BV1pE41117C2

Interactive Teaching Algorithm for IRL：https://www.bilibili.com/video/BV1mE411b7eu

Least-Squares Temporal Difference Learning：https://www.bilibili.com/video/BV1nE41117C8

2. 强化学习 Reinforcement Learning (莫烦 Python 教程)

   链接：https://www.bilibili.com/video/av16921335?from=search&seid=7037144790835305588

   代码：https://github.com/AndyYue1893/Reinforcement-learning-with-tensorflow

3. DeepMind - David Silver - UCL 深度强化学习课程 (2015)

   链接：https://www.bilibili.com/video/av45357759?from=search&seid=7037144790835305588

   PPT 地址：https://blog.csdn.net/u_say2what/article/details/89216190

   笔记及代码：https://zhuanlan.zhihu.com/p/37690204

4. 台大李宏毅深度强化学习 (国语) 课程 (2018)

   链接：https://www.bilibili.com/video/av24724071?from=search&seid=7037144790835305588

   PPT 地址：https://www.bilibili.com/video/av24724071?from=search&seid=7037144790835305588

   学习笔记：https://blog.csdn.net/cindy_1102/article/details/87904928

5. UC Berkeley - Sergey Levine - CS285(294) 深度强化学习 (2019)

   链接：https://www.bilibili.com/video/av69455099?from=search&seid=7037144790835305588

   PPT 地址：http://rail.eecs.berkeley.edu/deeprlcourse/

   代码：https://github.com/berkeleydeeprlcourse/homework

6.2 书籍

1. 强化学习（第 2 版）（Rich Sutton）

   链接：https://item.jd.com/12696004.html

   介绍：基础必读，有助于理解强化学习精髓

   英文电子版：http://incompleteideas.net/book/the-book-2nd.html

   代码：https://github.com/AndyYue1893/reinforcement-learning-an-introduction

2. Python 强化学习实战：应用 OpenAI Gym 和 TensorFlow 精通强化学习和深度强化学习（Sudharsan.Ravichandiran）

   链接：https://item.jd.com/12506442.html

   代码：https://github.com/AndyYue1893/Hands-On-Reinforcement-Learning-With-Python

3. 强化学习精要：核心算法与 TensorFlow 实现（冯超）

   链接：https://item.jd.com/12344157.html

代码：https://pan.baidu.com/share/init?surl=nQpNbhkI-3WucSD0Mk7Qcg 提取码: av5p

4. 深度强化学习原理与实践（陈仲铭）
   链接：https://item.jd.com/12524381.html
   代码：https://github.com/chenzomi12/Deep-Reinforcement-Learning

6.3 经典论文

6.3.1 典型应用

1. AlphaGo：https://www.nature.com/articles/nature16961，战胜人类棋手，David Silver 等人发表于 Nature
2. 玩 Atari：https://deepmind.com/research/publications/playing-atari-deep-reinforcement-learning，游戏领域超过普通玩家
3. 玩 Pong：http://karpathy.github.io/2016/05/31/rl/
4. 教小人走路：https://arxiv.org/abs/1707.02286，Deepmind 出品，入门必玩项目
5. 控制机械臂：https://ai.googleblog.com/2016/03/deep-learningfor-robots-learningfrom.html

6.3.2 传统方法

1. 蒙特卡洛估计：
   https://link.springer.com/chapter/10.1007/11871842_29
2. 时间差分学习：
   Off-Policy Temporal-Difference Learning with Function Approximation

6.3.3 强化学习理论

1. Policy Gradient Methods for Reinforcement Learning with Function Approximation
   链接：https://papers.nips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-function-approximation.pdf, Sutton et al, 2000.
   主要贡献：建立 policy gradient 理论，并展示出对任意 policy classes，算法都可以收敛。
2. An Analysis of Temporal-Difference Learning with Function Approximation
   链接：http://web.mit.edu/jnt/www/Papers/J063-97-bvr-td.pdf, Tsitsiklis and Van Roy, 1997.
   主要贡献：RL 中价值学习方法的各种收敛结果和反例。
3. Reinforcement Learning of Motor Skills with Policy Gradients
   链接：http://www.kyb.mpg.de/fileadmin/user_upload/files/publications/attachments/Neural-Netw-2008-21-682_48670.pdf, Peters and Schaal, 2008.
   主要贡献：彻底地回顾了当时 policy gradient 各种方法，其中有很多现在依然是非常有用的 DRL 方法。
4. Approximately Optimal Approximate Reinforcement Learning
   链接：https://people.eecs.berkeley.edu/ pabbeel/cs287-fa09/readings/KakadeLangford-icml2002.pdf,

Kakade and Langford, 2002.

主要贡献：单调优化理论的早期来源，是后来 TRPO 和其他算法的理论依据。

5. A Natural Policy Gradient

   链接：https://papers.nips.cc/paper/2073-a-natural-policy-gradient.pdf, Kakade, 2002.

   主要贡献：在 RL 中引入了 natural gradients，后来在 deep RL 中引入了 TRPO、ACKTR 和其他一些方法。

6. Algorithms for Reinforcement Learning

   链接：https://sites.ualberta.ca/ szepesva/papers/RLAlgsInMDPs.pdf, Szepesvari, 2009.

   主要贡献：在 DRL 之前对 RL 的基础和理论背景进行了综合讲解

6.3.4 Model-Free RL

a. Deep Q-Learning

1. Playing Atari with Deep Reinforcement Learning:
   https://www.cs.toronto.edu/ vmnih/docs/dqn.pdf, Mnih et al, 2013. **Algorithm: DQN.**
2. Deep Recurrent Q-Learning for Partially Observable MDPs:
   https://arxiv.org/abs/1507.06527, Hausknecht and Stone, 2015. **Algorithm: Deep Recurrent Q-Learning.**
3. Dueling Network Architectures for Deep Reinforcement Learning:
   https://arxiv.org/abs/1511.06581, Wang et al, 2015. **Algorithm: Dueling DQN.**
4. Deep Reinforcement Learning with Double Q-learning:
   https://arxiv.org/abs/1509.06461, Hasselt et al 2015. **Algorithm: Double DQN.**
5. Prioritized Experience Replay:
   https://arxiv.org/abs/1511.05952, Schaul et al, 2015. **Algorithm: Prioritized Experience Replay (PER).**
6. Rainbow: Combining Improvements in Deep Reinforcement Learning:
   https://arxiv.org/abs/1710.02298, Hessel et al, 2017. **Algorithm: Rainbow DQN.**

b. Policy Gradients

1. Asynchronous Methods for Deep Reinforcement Learning:
   https://arxiv.org/abs/1602.01783, Mnih et al, 2016. **Algorithm: A3C.**
2. Trust Region Policy Optimization:
   https://arxiv.org/abs/1502.05477, Schulman et al, 2015. **Algorithm: TRPO.**
3. High-Dimensional Continuous Control Using Generalized Advantage Estimation:
   https://arxiv.org/abs/1506.02438, Schulman et al, 2015. **Algorithm: GAE.**
4. Proximal Policy Optimization Algorithms:
   https://arxiv.org/abs/1707.06347, Schulman et al, 2017. **Algorithm: PPO-Clip, PPO-Penalty.**
5. Emergence of Locomotion Behaviours in Rich Environments:
   https://arxiv.org/abs/1707.02286, Heess et al, 2017. **Algorithm: PPO-Penalty.**

6. Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation:
   https://arxiv.org/abs/1708.05144, Wu et al, 2017. **Algorithm: ACKTR.**
7. Sample Efficient Actor-Critic with Experience Replay:
   https://arxiv.org/abs/1611.01224, Wang et al, 2016. **Algorithm: ACER.**
8. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor:
   https://arxiv.org/abs/1801.01290, Haarnoja et al, 2018. **Algorithm: SAC.**

c. Deterministic Policy Gradients

1. Deterministic Policy Gradient Algorithms:
   http://proceedings.mlr.press/v32/silver14.pdf, Silver et al, 2014. **Algorithm: DPG.**
2. Continuous Control With Deep Reinforcement Learning:
   https://arxiv.org/abs/1509.02971, Lillicrap et al, 2015. **Algorithm: DDPG.**
3. Addressing Function Approximation Error in Actor-Critic Methods:
   https://arxiv.org/abs/1802.09477, Fujimoto et al, 2018. **Algorithm: TD3.**

d. Distributional RL

1. A Distributional Perspective on Reinforcement Learning:
   https://arxiv.org/abs/1707.06887, Bellemare et al, 2017. **Algorithm: C51.**
2. Distributional Reinforcement Learning with Quantile Regression:
   https://arxiv.org/abs/1710.10044, Dabney et al, 2017. **Algorithm: QR-DQN.**
3. Implicit Quantile Networks for Distributional Reinforcement Learning:
   https://arxiv.org/abs/1806.06923, Dabney et al, 2018. **Algorithm: IQN.**
4. Dopamine: A Research Framework for Deep Reinforcement Learning:
   https://openreview.net/forum?id=ByG_3s09KX, Anonymous, 2018. **Contribution:** 介绍了 Dopamine, 一个实现了 DQN, C51, IQN, 和 Rainbow 等算法的代码仓. [Code link.

e. Policy Gradients with Action-Dependent Baselines

1. Q-Prop: Sample-Efficient Policy Gradient with An Off-Policy Critic:
   https://arxiv.org/abs/1611.02247, Gu et al, 2016. **Algorithm: Q-Prop.**
2. Action-depedent Control Variates for Policy Optimization via Stein' s Identity:
   https://arxiv.org/abs/1710.11198, Liu et al, 2017. **Algorithm: Stein Control Variates.**
3. The Mirage of Action-Dependent Baselines in Reinforcement Learning:
   https://arxiv.org/abs/1802.10031, Tucker et al, 2018. **Contribution:** 这篇论文中，对以前的论文中的说法（包括 Q-Prop 和 Stein Control Variates）进行了批判和重新评价，并发现其中存在重要的方法论错误。

f. Path-Consistency Learning

1. Bridging the Gap Between Value and Policy Based Reinforcement Learning: https://arxiv.org/abs/1702.08892, Nachum et al, 2017. **Algorithm: PCL.**
2. Trust-PCL: An Off-Policy Trust Region Method for Continuous Control: https://arxiv.org/abs/1707.01891, Nachum et al, 2017. **Algorithm: Trust-PCL.**

g. Other Directions for Combining Policy-Learning and Q-Learning

1. Combining Policy Gradient and Q-learning: https://arxiv.org/abs/1611.01626, O' Donoghue et al, 2016. **Algorithm: PGQL.**
2. The Reactor: A Fast and Sample-Efficient Actor-Critic Agent for Reinforcement Learning: https://arxiv.org/abs/1704.04651, Gruslys et al, 2017. **Algorithm: Reactor.**
3. Interpolated Policy Gradient: Merging On-Policy and Off-Policy Gradient Estimation for Deep Reinforcement Learning: http://papers.nips.cc/paper/6974-interpolated-policy-gradient-merging-on-policy-and-off-policy-gradient-estimation-for-deep-reinforcement-learning, Gu et al, 2017. **Algorithm: IPG.**
4. Equivalence Between Policy Gradients and Soft Q-Learning: https://arxiv.org/abs/1704.06440, Schulman et al, 2017. **Contribution:** 揭示了这两类 RL 算法之间的理论联系。

h. Evolutionary Algorithms

1. Evolution Strategies as a Scalable Alternative to Reinforcement Learning: https://arxiv.org/abs/1703.03864, Salimans et al, 2017. **Algorithm: ES.**

6.3.5 Model-Based RL

a. Model is Learned

1. Imagination-Augmented Agents for Deep Reinforcement Learning: https://arxiv.org/abs/1707.06203, Weber et al, 2017. **Algorithm: I2A.**
2. Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning: https://arxiv.org/abs/1708.02596, Nagabandi et al, 2017. **Algorithm: MBMF.**
3. Model-Based Value Expansion for Efficient Model-Free Reinforcement Learning: https://arxiv.org/abs/1803.00101, Feinberg et al, 2018. **Algorithm: MVE.**
4. Sample-Efficient Reinforcement Learning with Stochastic Ensemble Value Expansion: https://arxiv.org/abs/1807.01675, Buckman et al, 2018. **Algorithm: STEVE.**
5. Model-Ensemble Trust-Region Policy Optimization: https://openreview.net/forum?id=SJJinbWRZ&noteId=SJJinbWRZ, Kurutach et al, 2018. **Algorithm: ME-TRPO.**
6. Model-Based Reinforcement Learning via Meta-Policy Optimization: https://arxiv.org/abs/1809.05214, Clavera et al, 2018. **Algorithm: MB-MPO.**

7. Recurrent World Models Facilitate Policy Evolution:
   https://arxiv.org/abs/1809.01999, Ha and Schmidhuber, 2018.

b. Model is Given

1. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm:
   https://arxiv.org/abs/1712.01815, Silver et al, 2017. **Algorithm: AlphaZero.**
2. Thinking Fast and Slow with Deep Learning and Tree Search:
   https://arxiv.org/abs/1705.08439, Anthony et al, 2017. **Algorithm: ExIt.**

6.3.6 Imitation Learning & Inverse Reinforcement Learning

1. Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization:
   https://arxiv.org/abs/1603.00448, Finn et al, 2016. **Algorithm: GCL.**
2. Maximum Entropy Deep Inverse Reinforcement Learning:
   http://arxiv.org/abs/1507.04888, M. Wulfmeier et al., *arXiv*, 2015.
3. Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy:
   http://www.cs.cmu.edu/ bziebart/publications/thesis-bziebart.pdf, Ziebart 2010. **Contributions:** 最大熵 IRL 的公式提出和推导.
4. Generative Adversarial Imitation Learning:
   https://arxiv.org/abs/1606.03476, Ho and Ermon, 2016. **Algorithm: GAIL.**
5. DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills:
   https://xbpeng.github.io/projects/DeepMimic/2018_TOG_DeepMimic.pdf, Peng et al, 2018. **Algorithm: DeepMimic.**
6. Variational Discriminator Bottleneck: Improving Imitation Learning, Inverse RL, and GANs by Constraining Information Flow:
   https://arxiv.org/abs/1810.00821, Peng et al, 2018. **Algorithm: VAIL.**
7. One-Shot High-Fidelity Imitation: Training Large-Scale Deep Nets with RL:
   https://arxiv.org/abs/1810.05017, Le Paine et al, 2018. **Algorithm: MetaMimic.**

6.3.7 Multi-Task and Transfer Learning

1. Actor-Mimic: Deep Multitask and Transfer Reinforcement Learning:
   http://arxiv.org/abs/1511.06342, E. Parisotto, et al., *ICLR*, 2016.
2. Policy Distillation:
   http://arxiv.org/abs/1511.06295, A. A. Rusu et at., *ICLR*, 2016.
3. ADAAPT: A Deep Architecture for Adaptive Policy Transfer from Multiple Sources:
   http://arxiv.org/abs/1510.02879, J. Rajendran et al., *arXiv*, 2015.
4. Universal Value Function Approximators:
   http://schaul.site44.com/publications/uvfa.pdf, T. Schaul et al., *ICML*, 2015.**Algorithm: UVFA.**

5. Progressive Neural Networks:
   https://arxiv.org/abs/1606.04671, Rusu et al, 2016. **Algorithm: Progressive Networks.**
6. Reinforcement Learning with Unsupervised Auxiliary Tasks:
   https://arxiv.org/abs/1611.05397, Jaderberg et al, 2016. **Algorithm: UNREAL.**
7. The Intentional Unintentional Agent: Learning to Solve Many Continuous Control Tasks Simultaneously:
   https://arxiv.org/abs/1707.03300, Cabi et al, 2017. **Algorithm: IU Agent.**
8. PathNet: Evolution Channels Gradient Descent in Super Neural Networks:
   https://arxiv.org/abs/1701.08734, Fernando et al, 2017. **Algorithm: PathNet.**
9. Mutual Alignment Transfer Learning:
   https://arxiv.org/abs/1707.07907, Wulfmeier et al, 2017. **Algorithm: MATL.**
10. Learning an Embedding Space for Transferable Robot Skills:
    https://openreview.net/forum?id=rk07ZXZRb&noteId=rk07ZXZRb, Hausman et al, 2018.
11. Hindsight Experience Replay:
    https://arxiv.org/abs/1707.01495, Andrychowicz et al, 2017. **Algorithm: Hindsight Experience Replay (HER).**

6.3.8 Meta-RL

1. RL$\hat{2}$: Fast Reinforcement Learning via Slow Reinforcement Learning:
   https://arxiv.org/abs/1611.02779, Duan et al, 2016. **Algorithm: RL$\hat{2}$.**
2. Learning to Reinforcement Learn:
   https://arxiv.org/abs/1611.05763, Wang et al, 2016.
3. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks:
   https://arxiv.org/abs/1703.03400, Finn et al, 2017. **Algorithm: MAML.**
4. A Simple Neural Attentive Meta-Learner:
   https://openreview.net/forum?id=B1DmUzWAW&noteId=B1DmUzWAW, Mishra et al, 2018. **Algorithm: SNAIL.**

6.3.9 Improving Exploration

a. Intrinsic Motivation

1. VIME: Variational Information Maximizing Exploration:
   https://arxiv.org/abs/1605.09674, Houthooft et al, 2016. **Algorithm: VIME.**
2. Unifying Count-Based Exploration and Intrinsic Motivation:
   https://arxiv.org/abs/1606.01868, Bellemare et al, 2016. **Algorithm: CTS-based Pseudocounts.**
3. Count-Based Exploration with Neural Density Models:
   https://arxiv.org/abs/1703.01310, Ostrovski et al, 2017. **Algorithm: PixelCNN-based Pseudocounts.**

4. #Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning: https://arxiv.org/abs/1611.04717, Tang et al, 2016. **Algorithm: Hash-based Counts.**
5. EX2: Exploration with Exemplar Models for Deep Reinforcement Learning: https://arxiv.org/abs/1703.01260, Fu et al, 2017. **Algorithm: EX2.**
6. Curiosity-driven Exploration by Self-supervised Prediction: https://arxiv.org/abs/1705.05363, Pathak et al, 2017. **Algorithm: Intrinsic Curiosity Module (ICM).**
7. Large-Scale Study of Curiosity-Driven Learning: https://arxiv.org/abs/1808.04355, Burda et al, 2018. **Contribution:** 系统分析基于好奇心的内在动机算法在各种环境中的表现
8. Exploration by Random Network Distillation: https://arxiv.org/abs/1810.12894, Burda et al, 2018. **Algorithm: RND.**

b. Unsupervised RL

1. Variational Intrinsic Control: https://arxiv.org/abs/1611.07507, Gregor et al, 2016. **Algorithm: VIC.**
2. Diversity is All You Need: Learning Skills without a Reward Function: https://arxiv.org/abs/1802.06070, Eysenbach et al, 2018. **Algorithm: DIAYN.**
3. Variational Option Discovery Algorithms: https://arxiv.org/abs/1807.10299, Achiam et al, 2018. **Algorithm: VALOR.**

6.3.10 Multi-Agent

1. Learning to Communicate to Solve Riddles with Deep Distributed Recurrent Q-Networks: http://arxiv.org/abs/1602.02672, J. N. Foerster et al., *arXiv*, 2016.
2. Multiagent Cooperation and Competition with Deep Reinforcement Learning: http://arxiv.org/abs/1511.08779, A. Tampuu et al., *arXiv*, 2015.

6.3.11 Hierarchical Learning

1. Deep Successor Reinforcement Learning: http://arxiv.org/abs/1606.02396, T. D. Kulkarni et al., *arXiv*, 2016.
2. Hierarchical Reinforcement Learning using Spatio-Temporal Abstractions and Deep Neural Networks: https://arxiv.org/abs/1605.05359, R. Krishnamurthy et al., *arXiv*, 2016.
3. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation: https://arxiv.org/abs/1604.06057, T. D. Kulkarni et al., *arXiv*, 2016.
4. Strategic Attentive Writer for Learning Macro-Actions: https://arxiv.org/abs/1606.04695, Vezhnevets et al, 2016. **Algorithm: STRAW.**

5. FeUdal Networks for Hierarchical Reinforcement Learning:
   https://arxiv.org/abs/1703.01161, Vezhnevets et al, 2017. **Algorithm: Feudal Networks**
6. Data-Efficient Hierarchical Reinforcement Learning:
   https://arxiv.org/abs/1805.08296, Nachum et al, 2018. **Algorithm: HIRO.**

6.3.12 Safety

1. Concrete Problems in AI Safety:
   https://arxiv.org/abs/1606.06565, Amodei et al, 2016. **Contribution:** 建立安全问题的分类，这个方向需要解决的问题。
2. Deep Reinforcement Learning From Human Preferences:
   https://arxiv.org/abs/1706.03741, Christiano et al, 2017. **Algorithm: LFP.**
3. Constrained Policy Optimization:
   https://arxiv.org/abs/1705.10528, Achiam et al, 2017. **Algorithm: CPO.**
4. Safe Exploration in Continuous Action Spaces:
   https://arxiv.org/abs/1801.08757, Dalal et al, 2018. **Algorithm: DDPG+Safety Layer.**
5. Trial without Error: Towards Safe Reinforcement Learning via Human Intervention:
   https://arxiv.org/abs/1707.05173, Saunders et al, 2017. **Algorithm: HIRL.**
6. Leave No Trace: Learning to Reset for Safe and Autonomous Reinforcement Learning:
   https://arxiv.org/abs/1711.06782, Eysenbach et al, 2017. **Algorithm: Leave No Trace.**

6.3.13 Memory

1. Model-Free Episodic Control:
   https://arxiv.org/abs/1606.04460, Blundell et al, 2016. **Algorithm: MFEC.**
2. Neural Episodic Control:
   https://arxiv.org/abs/1703.01988, Pritzel et al, 2017. **Algorithm: NEC.**
3. Neural Map: Structured Memory for Deep Reinforcement Learning:
   https://arxiv.org/abs/1702.08360, Parisotto and Salakhutdinov, 2017. **Algorithm: Neural Map.**
4. Unsupervised Predictive Memory in a Goal-Directed Agent:
   https://arxiv.org/abs/1803.10760,
   Wayne et al, 2018. **Algorithm: MERLIN.**
5. Relational Recurrent Neural Networks:
   https://arxiv.org/abs/1806.01822, Santoro et al, 2018. **Algorithm: RMC.**

6.3.14 Scaling RL（分布式 RL）

1. Accelerated Methods for Deep Reinforcement Learning:
   https://arxiv.org/abs/1803.02811, Stooke and Abbeel, 2018. **Contribution:** 跨算法的深度 RL 中的并行化系统分析。

2. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures: https://arxiv.org/abs/1802.01561, Espeholt et al, 2018. **Algorithm: IMPALA.**
3. Distributed Prioritized Experience Replay: https://openreview.net/forum?id=H1Dy—0Z, Horgan et al, 2018. **Algorithm: Ape-X.**
4. Recurrent Experience Replay in Distributed Reinforcement Learning: https://openreview.net/forum?id=r1lyTjAqYX, Anonymous, 2018. **Algorithm: R2D2.**
5. RLlib: Abstractions for Distributed Reinforcement Learning: https://arxiv.org/abs/1712.09381, Liang et al, 2017. **Contribution:** 提出一个实现了 17 年前大部分 RL 算法的可扩展库。

6.3.15 RL in the Real World

1. Benchmarking Reinforcement Learning Algorithms on Real-World Robots: https://arxiv.org/abs/1809.07731, Mahmood et al, 2018.
2. Learning Dexterous In-Hand Manipulation: https://arxiv.org/abs/1808.00177, OpenAI, 2018.
3. QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation: https://arxiv.org/abs/1806.10293, Kalashnikov et al, 2018. **Algorithm: QT-Opt.**
4. Horizon: Facebook’s Open Source Applied Reinforcement Learning Platform: https://arxiv.org/abs/1811.00260, Gauci et al, 2018.

6.3.16 RL 的可复现性等方面的分析

1. Benchmarking Deep Reinforcement Learning for Continuous Control: https://arxiv.org/abs/1604.06778, Duan et al, 2016. **Contribution: rllab 库.**
2. Reproducibility of Benchmarked Deep Reinforcement Learning Tasks for Continuous Control: https://arxiv.org/abs/1708.04133, Islam et al, 2017.
3. Deep Reinforcement Learning that Matters: https://arxiv.org/abs/1709.06560, Henderson et al, 2017.
4. Where Did My Optimum Go?: An Empirical Analysis of Gradient Descent Optimization in Policy Gradient Methods: https://arxiv.org/abs/1810.02525, Henderson et al, 2018.
5. Are Deep Policy Gradient Algorithms Truly Policy Gradient Algorithms?: https://arxiv.org/abs/1811.02553, Ilyas et al, 2018.
6. Simple Random Search Provides a Competitive Approach to Reinforcement Learning: https://arxiv.org/abs/1803.07055, Mania et al, 2018.
7. Benchmarking Model-Based Reinforcement Learning: https://arxiv.org/abs/1907.02057, Wang et al, 2019.