# Contents


**1 一些回顾性的经典总结** **8**
- 1.1 2019 年盘点 . . . 8
- 1.2 2019nlp . . . 9
- 1.3 2021 年盘点 . . . 9

**2 传统 ML** **9**
- 2.1 PRML . . . 9
- 2.2 传统 ML 库 . . . 9
- 2.3 数学相关 . . . 9
- 2.4 数据降维 . . . 9
- 2.5 孪生网络 . . . 9
- 2.6 聚类 . . . 9
- 2.7 树相关 . . . 10
- 2.8 xgb . . . 10
- 2.9 深度森林 . . . 10
- 2.10 主题模型 . . . 10
- 2.11 CRF . . . 10

**3 DL 基础研究** **10**
- 3.1 DL 背后的原理 . . . 10
- 3.2 NTK . . . 11
- 3.3 优化算法 . . . 11
- 3.4 优化算法综述 . . . 11
- 3.5 复合函数最优化 . . . 11
- 3.6 adabound . . . 11
- 3.7 batchsize 与学习率相关 . . . 11
- 3.8 backpack 框架 . . . 12
- 3.9 Ranger . . . 12
- 3.10 Shampoo . . . 12
- 3.11 激活函数 . . . 12
- 3.12 激活函数汇总 . . . 12
- 3.13 GELU . . . 12
- 3.14 Relu 神经网络的记忆能力 . . . 12
- 3.15 梯度泄露 . . . 12
- 3.16 彩票假设 . . . 13
- 3.17 知识蒸馏 . . . 13
- 3.18 生成模型 . . . 13
- 3.19 双下降问题 . . . 13
- 3.20 一些疑难杂症 . . . 13
- 3.21 度量学习 . . . 13
- 3.22 损失函数 . . . 13
- 3.23 mse vs cross-entropy . . . 13
- 3.24 cross-entroy vs nllloss . . . 14
- 3.25 L_DMI . . . 14
- 3.26 损失函数的 pattern . . . 14
- 3.27 softmax 优化 . . . 14
- 3.28 Mixtape . . . 14
- 3.29 自监督 . . . 15
- 3.30 机器学习 + 博弈论 . . . 15
- 3.31 normalization 相关 . . . 15
- 3.32 BN . . . 15
- 3.33 FRN . . . 15

3.34 VAE . . . 15
3.35 优化方法 . . . 15
3.36 调参 . . . 15
3.37 表示学习 . . . 15
3.38 新的结构 . . . 15
3.39 NALU . . . 15
3.40 权重无关 . . . 16
3.41 on-lstm . . . 16
3.42 其他相关理论 . . . 16
3.43 因果关系 . . . 16
3.44 能量模型 . . . 16
3.45 贝叶斯深度学习 . . . 16
3.46 可解释性 . . . 17
3.47 子集选择 . . . 17

**4 计算机视觉 17**
4.1 cv 数据集 . . . 17
4.2 cv 基础 . . . 17
4.3 cv 历史 . . . 18
4.4 cnn 相关 . . . 18
4.5 图像分割 . . . 18
4.6 图像分割综述 . . . 18
4.7 MoCo . . . 18
4.8 PointRend . . . 18
4.9 Graph-FCN . . . 18
4.10 目标检测 . . . 18
4.11 自然场景文字定位 . . . 18
4.12 EfficientDet . . . 19
4.13 YoloVxxx . . . 19
4.14 图像识别 . . . 19
4.15 图像补全 . . . 19
4.16 文字检测与识别 . . . 19
4.17 图像合成 . . . 19
4.18 人脸识别 . . . 20
4.19 CV 相关比赛 . . . 20
4.20 3D 模型相关 . . . 20
4.21 GNN+CV . . . 20
4.22 CV 最新进展 . . . 20
4.23 半弱监督 . . . 20
4.24 advprop . . . 20
4.25 稀疏性 +cv . . . 21
4.26 自监督 + 半监督 EnAET . . . 21
4.27 无监督 SimCLR . . . 21
4.28 图像对抗攻击 . . . 21

**5 自然语言处理 22**
5.1 nlp 综述 . . . 22
5.2 LSTM 相关 . . . 22
5.3 fasttext&word2vec . . . 22
5.4 分词 . . . 22
5.5 语法解析 . . . 22
5.6 self-attention . . . 22
5.7 文本匹配 . . . 22
5.8 机器翻译 . . . 23

5.9 nlp 标准 & 数据集 . . . 23
5.10 中文 glue . . . 23
5.11 中文阅读理解数据集 . . . 23
5.12 物理常识推理任务数据集 . . . 23
5.13 常识推理数据集 WinoGrande . . . 24
5.14 阅读理解 . . . 24
5.15 DCMN+ . . . 24
5.16 相关性模型 . . . 24
5.17 ULMFiT . . . 24
5.18 bert/transformer 相关总结 . . . 24
5.19 huggingface 的 nlp 预训练模型库 . . . 24
5.20 bert . . . 24
5.20.1 multi-head att 实现 . . . 25
5.20.2 masked-language-model 的实现 . . . 28
5.21 gpt-2 . . . 29
5.22 gpt-2 8b . . . 29
5.23 distill gpt-2 . . . 29
5.24 albert . . . 29
5.25 XLNet . . . 29
5.26 ELECTRA . . . 30
5.27 BART . . . 30
5.28 gated transformer-xl . . . 30
5.29 bert/transformer 加速 . . . 30
5.29.1 bert 蒸馏、量化、剪枝 . . . 30
5.29.2 reformer . . . 30
5.29.3 LTD-bert . . . 30
5.29.4 Q-bert . . . 31
5.29.5 Adabert . . . 31
5.30 t5 . . . 31
5.31 哪吒 +tinybert . . . 31
5.32 XLM-R . . . 31
5.33 transformer+ 生成模型 . . . 31
5.34 经典文本生成模型 . . . 31
5.35 UniLM . . . 31
5.36 LaserTagger . . . 32
5.37 pegasus . . . 32
5.38 T-NLG/DeepSpeed . . . 32
5.39 transformer 应用于检索召回 . . . 32
5.40 一些 bert/transformer 的应用 . . . 32
5.41 poly-encoder . . . 32
5.42 bert/transformer 其他 . . . 33
5.43 对话 . . . 33
5.44 对话数据集 . . . 33
5.45 对话领域的传统模型 . . . 33
5.46 convai . . . 33
5.47 微软小冰 . . . 33
5.48 RASA . . . 33
5.49 生成式对话 . . . 34
5.50 开放领域聊天机器人 . . . 34
5.51 问答系统 . . . 34
5.52 NER . . . 34
5.53 知识图谱 . . . 34
5.54 关系提取 . . . 34
5.55 常识知识与常识推理 . . . 34

**6 语音算法** **34**
6.1 语音数据集 . . . 34
6.2 中文音乐数据集 . . . 34
6.3 时域音频分离模型 . . . 34
6.4 中文语音识别 . . . 34
6.5 顺滑度 . . . 35
6.6 语音识别加速 . . . 35
6.7 唇读 . . . 35
6.8 Live caption . . . 35
6.9 demucs . . . 35
6.10 语音版 bert . . . 35
6.11 搜狗录音笔 . . . 35
6.12 音乐推荐相关 . . . 36

**7 视频算法** **36**
7.1 视频数据集 . . . 36
7.2 VTAB . . . 36
7.3 视频检索 . . . 36
7.4 视频编码相关 . . . 36
7.5 视频显著区域检测 . . . 36
7.6 视频理解 . . . 36
7.7 pyslowfast . . . 36
7.8 视频插帧相关 . . . 36
7.9 DAIN . . . 36
7.10 Quadratic Video Interpolation . . . 37
7.11 TVN . . . 37
7.12 MvsGCN：多视频摘要 . . . 37
7.13 A-GANet . . . 37
7.14 VRD-GCN . . . 37
7.15 video caption . . . 38
7.16 小视频推荐 . . . 38
7.17 MMGCN . . . 38
7.18 ALPINE . . . 38
7.19 长视频剪辑 . . . 38
7.20 AutoFlip . . . 38
7.21 快手视频相关工作 . . . 39
7.22 EIUM：讲究根源的快手短视频推荐 . . . 39
7.23 Comyco：基于质量感知的码率自适应策略 . . . 39
7.24 Livesmart：智能 CDN 调度 . . . 39
7.25 抖音视频相关工作 . . . 39
7.26 google 视频相关工作 . . . 39
7.27 阿里短视频推荐相关工作 . . . 39

**8 GAN** **39**
8.1 GAN 综述 . . . 39
8.2 LOGAN . . . 39
8.3 ShapeMatchingGAN . . . 39
8.4 图像生成 +GAN . . . 39
8.5 模式崩塌问题 . . . 40
8.6 imagestylegan++ . . . 40
8.7 stylegan2 . . . 40
8.8 starganv2 . . . 40
8.9 nlp+gan . . . 40
8.10 gan 压缩 . . . 40

**9 推荐系统** **40**
9.1 推荐系统整体梳理 . . . 41
9.2 推荐中的采样 . . . 41
9.3 序列建模 . . . 41
9.4 bias v.s. debias . . . 41
9.5 position bias . . . 41
9.6 用户模型 . . . 42
9.7 PeterRec . . . 42
9.8 召回 . . . 42
9.9 JTM . . . 43
9.10 DR . . . 43
9.11 transformer+ 推荐 . . . 43
9.12 多目标 . . . 43
9.13 工业界的一些推荐应用 . . . 43
9.14 dlrm . . . 43
9.15 混合推荐架构 . . . 43
9.16 instagram 推荐系统 . . . 43
9.17 微信读书推荐系统 . . . 43
9.18 youtube 推荐梳理 . . . 43
9.19 认知推荐 . . . 43
9.20 自监督 . . . 44
9.21 GNN+ 推荐 . . . 44
9.22 DCNv2 . . . 44
9.23 item 冷启 . . . 44
9.24 暴力召回 ANN 加速 . . . 44
9.25 多目标 . . . 45
9.25.1 多目标 + 推荐综述 . . . 45
9.25.2 阿里多目标 . . . 45
9.25.3 Youtube 多目标——MMoE . . . 45
9.26 特征工程 . . . 45
9.27 CTR 预估 . . . 45
9.27.1 传统 ctr . . . 45
9.27.2 lr for ctr . . . 45
9.27.3 gbdt for ctr . . . 45
9.27.4 深度学习 ctr . . . 46
9.27.5 ctr 特征 . . . 46
9.27.6 HugeCTR . . . 46
9.27.7 阿里妈妈 CTR . . . 46
9.27.8 凤巢 . . . 47
9.27.9 cvr . . . 47
9.27.10 时长预估 . . . 47
9.27.11 APG . . . 47
9.27.12 保序回归 . . . 47

**10 图神经网络** **48**
10.1 GNN 数据集 . . . 48
10.2 GNN 综述 . . . 48
10.3 GNN 理论研究 . . . 48
10.4 图翻译 . . . 48
10.5 异构图 GNN . . . 48
10.6 HAN-GNN . . . 48
10.7 GTN . . . 49
10.8 HetGNN . . . 49
10.9 HGAT . . . 49

10.10MEIRec . . . 49
10.11GAS . . . 49
10.12AAAI2020 GNN . . . 49
10.13cluster-GCN . . . 49
10.14深层 GCN . . . 49
10.15GNN 或者图模型的一些应用场景 . . . 49
10.16风控关系 . . . 49
10.17社区发现相关 . . . 50
10.18transformer+gnn . . . 50

**11 强化学习 50**
11.1 RL 历史 . . . 50
11.2 RL 概述 . . . 50
11.3 MAB 相关 . . . 50
11.4 multitask+mab . . . 50
11.5 RL 基础 . . . 50
11.6 srl+drl . . . 50
11.7 模仿学习 . . . 50
11.8 推荐 + 强化学习 . . . 50
11.9 2019 强化学习论文 . . . 50
11.10ICLR2020 强化学习相关 . . . 50
11.11HER . . . 51
11.12多智能体 RL . . . 51
11.13LIIR . . . 51
11.14AlphaStar . . . 51
11.15TVT . . . 51
11.16upside-down rl . . . 51
11.17游戏 +RL . . . 52
11.18游戏 AI 历史 (alphago 系列) . . . 52
11.18.1 alphago . . . 52
11.18.2 alphago zero . . . 52
11.18.3 alpha zero . . . 52
11.18.4 muzero . . . 52
11.19绝悟 . . . 53
11.20RL+ 因果 . . . 53
11.21RL+Active learning . . . 53
11.22ES . . . 53

**12 Auto-ML 53**
12.1 automl 综述 . . . 53
12.2 HM-NAS . . . 53
12.3 FGNAS . . . 53
12.4 NAT . . . 54
12.5 NASP . . . 54
12.6 NASP+ 推荐系统 . . . 54
12.7 automl+nlp . . . 54
12.8 nni . . . 54
12.9 视频 NAS . . . 54

**13 few-shot & meta-learning 54**
13.1 meta-learning . . . 54
13.2 few-shot 数据集 . . . 55
13.3 incremental few-shot . . . 55
13.4 few-shot 无监督 img2img . . . 55

13.5 TADAM . . . 55
13.6 AutoGRD . . . 55
13.7 few-shot 的一些应用 . . . 55
13.8 nips 2019 few-shot . . . 55

**14 压缩与部署** **55**
14.1 压缩综述 . . . 55
14.2 layer dropout . . . 56
14.3 剪枝相关 . . . 56
14.4 slimmable networks . . . 56
14.5 TAS(NAS+ 剪枝) . . . 56
14.6 GDP . . . 56
14.7 metapruning . . . 56
14.8 data-free student . . . 56
14.9 样本相关性用于蒸馏 . . . 57
14.10 pu learning+ 压缩 . . . 57
14.11 对抗训练 + 压缩 . . . 57
14.12 GSM . . . 57
14.13 无监督量化 . . . 57
14.14 Autocompress . . . 57

**15 硬件** **57**
15.1 硬件综述 . . . 57
15.2 TPU . . . 57
15.3 pixel 4 . . . 58
15.4 MNNKit . . . 58
15.5 deepshift . . . 58
15.6 传感器相关 . . . 58

**16 架构** **58**
16.1 分布式机器学习 . . . 58
16.2 JAX . . . 58
16.3 trax . . . 58
16.4 spark . . . 59
16.5 kaggle 相关工具 . . . 59
16.6 blink . . . 59
16.7 cortex . . . 59
16.8 optuna . . . 59
16.9 DALI . . . 59
16.10 tf . . . 59
16.11 tf-lite . . . 59
16.12 tensorboard.dev . . . 59
16.13 pytorch . . . 59
16.13.1 torch 里的 categorical 分布 (类别分布) . . . 59
16.13.2 pytorch 设计思路 . . . 60
16.13.3 pytorch 分布式训练 . . . 60
16.13.4 Texar-PyTorch . . . 60
16.14 Streamlit . . . 60
16.15 paddle . . . 60
16.16 TensorRT . . . 60
16.17 开源 gnn 平台 . . . 60
16.18 dgl . . . 60
16.19 plato . . . 60
16.20 angel(java) . . . 60

16.21bytegraph . . . 61
16.22tensorlayer . . . 61
16.23MAX . . . 61
16.24CogDL . . . 61
16.25auto-ml 架构 . . . 61
16.26megengine . . . 61
16.27异构硬件加速 . . . 61
16.28GPU 基础知识 . . . 61

**17 课程资源 62**
17.1 分布式系统课程 . . . 62
17.2 微软 ml 基础课程 (win10 版) . . . 62
17.3 无监督课程 . . . 62
17.4 tf2.0 课程 . . . 62
17.5 tf 2.0 分布式课程 . . . 62
17.6 深度学习 + 强化学习课程 . . . 62
17.7 统计学习方法课程 . . . 63
17.8 Colab 相关课程 . . . 63
17.9 李宏毅机器学习课程 . . . 63
17.10nlp+ 社交课程 . . . 63
17.11 图机器学习课程 . . . 63
17.12deeplearning.ai 课程 . . . 63
17.13多任务与元学习课程 . . . 63

**18 一些网址和应用 63**
18.1 一些笔记 . . . 63
18.2 各类数据集 . . . 63
18.3 数据集搜索 . . . 63
18.4 烂番茄影评数据集 . . . 64
18.5 numpy 实现 ml 算法 . . . 64
18.6 pytorch 实现 rl . . . 64
18.7 一些有趣应用 . . . 64

**19 基础知识 64**
19.1 C++ . . . 64
19.2 python . . . 64
19.3 core 相关 . . . 64

**20 其他领域 65**
20.1 信息安全 . . . 65
20.2 量子计算 . . . 65
20.3 机器人 . . . 65
20.4 落地的一些思考 . . . 66

下载本文 pdf: https://github.com/daiwk/collections/blob/master/pdfs/collections.pdf

# 1 一些回顾性的经典总结

## 1.1 2019 年盘点

告别 2019：属于深度学习的十年，那些我们必须知道的经典

“深度学习”这十年：52 篇大神级论文再现 AI 荣与光

Jeff Dean 谈 2020ML：专用芯片、多模态多任务学习，社区不用痴迷 SOTA

NLPer 复工了！先看看这份 2019 机器学习与 NLP 年度盘点吧

三巨头共聚 AAAI：Capsule 没有错，LeCun 看好自监督，Bengio 谈注意力

Geoffrey Hinton 介绍了Stacked Capsule Autoencoders，即一种无监督版本的 Capsule 网络，这种神经编码器能查看所有的组成部分，并用于推断跟细节的特征；Yann LeCun 在《Self-Supervised Learning》中再次强调了自监督学习的重要性；(nlp 那章里讲到了) Yoshua Bengio 在Deep Learning for System 2 Processing中回顾了深度学习，并讨论了当前的局限性以及前瞻性研究方向。

### 1.2 2019nlp

2019 NLP 大全：论文、博客、教程、工程进展全梳理（长文预警）

### 1.3 2021 年盘点

google research 总结

https://ai.googleblog.com/2022/01/google-research-themes-from-2021-and.html

谷歌大神 Jeff Dean 领衔，万字展望 5 大 AI 趋势播报文章

## 2 传统 ML

### 2.1 PRML

GitHub 标星 6000+！Python 带你实践机器学习圣经 PRML

### 2.2 传统 ML 库

Scikit-learn 新版本发布，一行代码秒升级

### 2.3 数学相关

这本机器学习的数学“百科全书”，可以免费获取 | 宾大计算机教授出品

https://www.cis.upenn.edu/~jean/math-deep.pdf

一图胜千言，这本交互式线代教科书让你分分钟理解复杂概念，佐治亚理工出品

https://textbooks.math.gatech.edu/ila/

https://textbooks.math.gatech.edu/ila/ila.pdf

https://github.com/QBobWatson/gt-linalg

### 2.4 数据降维

哈工大硕士生用 Python 实现了 11 种经典数据降维算法，源代码库已开放

https://github.com/heucoder/dimensionality_reduction_alo_codes

### 2.5 孪生网络

Siamese network 孪生神经网络---一个简单神奇的结构

### 2.6 聚类

快速且不需要超参的无监督聚类方法

Efficient Parameter-free Clustering Using First Neighbor Relations

https://github.com/ssarfraz/FINCH-Clustering

## 2.7 树相关

## 2.8 xgb

珍藏版 | 20 道 XGBoost 面试题

参考7 papers | Quoc V. Le、何恺明等新论文；用进化算法设计炉石

A Comparative Analysis of XGBoost

XGBoost 是一项基于梯度提升可扩展集合技术，在解决机器学习难题方面是可靠和有效的。在本文中，研究者对这项新颖的技术如何在训练速度、泛化性能和参数设置方面发挥作用进行了实证分析。此外，通过精心调整模型和默认设置，研究者还对 XGBoost、随机森林和梯度提升展开了综合比较。结果表明，XGBoost 在所有情况下并不总是最佳选择。最后，他们还对 XGBoost 的参数调整过程进行了扩展分析。

推荐：通过对随机森林、梯度提升和 XGBoost 的综合比较，来自法国波尔多大学、匈牙利帕兹曼尼·彼得天主教大学以及马德里自治大学的三位研究者得出结论：从调查问题的数量看，梯度提升是最好的分类器，但默认参数设置下 XGBoost 和随机森林在平均排名（average rank）方面的差异不具备统计显著性。

## 2.9 深度森林

周志华团队：深度森林挑战多标签学习，9 大数据集超越传统方法

## 2.10 主题模型

如何找到好的主题模型量化评价指标？这是一份热门方法总结

## 2.11 CRF

【机器学习】条件随机场

# 3 DL 基础研究

## 3.1 DL 背后的原理

从 2019 AI 顶会最佳论文，看深度学习的理论基础

MIT 教授 Tomaso Poggio 曾在他的系列研究中 [1] 表示深度学习理论研究可以分为三大类：

- 表征问题（Representation）：为什么深层网络比浅层网络的表达能力更好？
- 最优化问题（Optimization）：为什么梯度下降能找到很好的极小值解，好的极小值有什么特点？
- 泛化问题（Generalization）：为什么过参数化仍然能拥有比较好的泛化性，不过拟合？

对于表征问题，我们想要知道深度神经网络这种「复合函数」，它的表达能力到底怎么确定，它的复合机制又是什么样的。我们不再满足于「能拟合任意函数」这样的定性描述，我们希望知道是不是有一种方法能描述 50 层 ResNet、12 层 Transformer 的拟合能力，能不能清楚地了解它们的理论性质与过程。

有了表征能力，那也只是具备了拟合潜力，深度学习还需要找到一组足够好的极值点，这就是模型的最优解。不同神经网络的「最优化 Landscape」是什么样的、怎样才能找到这种高维复杂函数的优秀极值点、极值点的各种属性都需要完善的理论支持。

最后就是泛化了，深度模型泛化到未知样本的能力直接决定了它的价值。那么深度模型的泛化边界该怎样确定、什么样的极值点又有更好的泛化性能，很多重要的特性都等我们确定一套理论基准。

英伟达工程师解读 NeurIPS 2019 最热趋势：贝叶斯深度学习、图神经网络、凸优化

neurips2019 杰出新方向论文奖颁给了 Vaishnavh Nagarajan 和 J. Zico Kolter 的《一致收敛理论可能无法解释深度学习中的泛化现象》(Uniform convergence may be unable to explain generalization in deep learning)，其论点是一致收敛理论本身并不能解释深度学习泛化的能力。随着数据集大小的增加，泛化差距 (模型对可见和不可见数据的性能差距) 的理论界限也会增加，而经验泛化差距则会减小。

Shirin Jalali 等人的论文《高斯混合模型的高效深度学习》(Efficient Deep Learning of Gaussian mix Models) 从这个问题引入：“通用逼近定理指出，任何正则函数都可以使用单个隐藏层神经网络进行逼近。深度是否能让它更具效率?”他们指出，在高斯混合模型的最佳贝叶斯分类的情况下，这样的函数可以用具有一个隐藏层的神经网络中的 O(exp(n)) 节点来近似，而在两层网络中只有 O(n) 节点。

## 3.2 NTK

神经正切核 (neural tangent kernel, NTK) 是近年来研究神经网络优化与泛化的一个新方向。它出现在数个 spotlight 报告和我在 NeuIPS 与许多人的对话中。Arthur Jacot 等人基于完全连通神经网络在无限宽度限制下等同于高斯过程这一众所周知的概念，在函数空间而非参数空间中研究了其训练动力学。他们证明了“在神经网络参数的梯度下降过程中，网络函数 (将输入向量映射到输出向量) 遵循函数的核函数梯度成本，关于一个新的核：NTK。”他们还表明，当有限层版本的 NTK 经过梯度下降训练时，其性能会收敛到无限宽度限制 NTK，然后在训练期间保持不变。

NeurIPS 上关于 NTK 的论文有：

Learning and Generalization in Overparameterized Neural Networks, Going Beyond Two Layers

On the Inductive Bias of Neural Tangent Kernels

但是，许多人认为 NTK 不能完全解释深度学习。神经网络接近 NTK 状态所需要的超参数设置——低学习率、大的初始化、无权值衰减——在实践中通常不用于训练神经网络。NTK 的观点还指出，神经网络只会像 kernel 方法一样泛化，但从经验上看，它们可以更好地泛化。

Colin Wei 等人的论文 “Regularization Matters: Generalization and Optimization of Neural Nets v.s. their Induced Kernel” 从理论上证明了具有权值衰减的神经网络泛化效果要比 NTK 好得多，这表明研究 L2-regularized 神经网络可以更好的理解泛化。NeurIPS 的以下论文也表明，传统的神经网络可以超越 NTK：

What Can ResNet Learn Efficiently, Going Beyond Kernels?

Limitations of Lazy Training of Two-layers Neural Network

## 3.3 优化算法

## 3.4 优化算法综述

理论、算法两手抓，UIUC 助理教授孙若愚 60 页长文综述深度学习优化问题

Optimization for deep learning: theory and algorithms

一个框架看懂优化算法之异同 SGD/AdaGrad/Adam

## 3.5 复合函数最优化

Efficient Smooth Non-Convex Stochastic Compositional Optimization via Stochastic Recursive Gradient Descent

nips2019 快手，针对复合函数的最优化方法。这里复合指的是一个数学期望函数中复合了另一个数学期望，而常规的 ML 目标函数就只有最外面一个数学期望。这种最优化方法在风险管理或 RL 中非常有用，例如在 RL 中解贝尔曼方程，它本质上就是复合函数最优化问题。

## 3.6 adabound

AdaBound 是一种优化程序，旨在提高不可见的数据的训练速度和性能，可用 PyTorch 实现。

AdaBound：一种基于 PyTorch 实现的优化器，训练速度堪比 Adam，质量堪比 SGD（ICLR 2019）

Adaptive Gradient Methods with Dynamic Bound of Learning Rate

https://github.com/Luolc/AdaBound

## 3.7 batchsize 与学习率相关

《控制批大小和学习率以很好地泛化：理论和实证证据》(Control Batch Size and Learning Rate to Generalize Well: Theoretical and Empirical Evidence) 中，Fengxiang He 的团队在 CIFAR 数据集上使用 SGD 训练了 1600 个 ResNet-110 和 VGG-19 模型，发现这些模型的泛化能力与 batch size 负相关，与学习率正相关，与批大小/学习率之比负相关。

## 3.8 backpack 框架

BackPACK: Packing more into backprop

自动微分框架只在计算平均小批量（mini-batch）梯度时进行优化。但在理论上，小批量梯度方差或 Hessian 矩阵近似值等其他数量可以作为梯度实现高效的计算。研究人员对这些数量抱有极大的兴趣，但目前的深度学习软件不支持自动计算。此外，手动执行这些数量非常麻烦，效率低，生成代码的共享性也不高。这种情况阻碍了深度学习的进展，并且导致梯度下降及其变体的研究范围变窄。与此同时，这种情况还使得复现研究以及新提出需要这些数量的方法之间的比较更为复杂。因此，为了解决这个问题，来自图宾根大学的研究者在本文中提出一种基于 PyTorch 的高效框架 BackPACK，该框架可以扩展反向传播算法，进而从一阶和二阶导数中提取额外信息。研究者对深度神经网络上额外数量的计算进行了基准测试，并提供了一个测试最近几种曲率估算优化的示例应用，最终证实了 BackPACK 的性能。

## 3.9 Ranger

可以丢掉 SGD 和 Adam 了，新的深度学习优化器 Ranger：RAdam + LookAhead 强强结合

## 3.10 Shampoo

二阶梯度优化新崛起，超越 Adam，Transformer 只需一半迭代量

Second Order Optimization Made Practical

摘要：目前，无论是从理论还是应用层面来说，机器学习中的优化都是以随机梯度下降等一阶梯度方法为主。囊括二阶梯度和/或二阶数据统计的二阶优化方法虽然理论基础更强，但受限于计算量、内存和通信花销等因素，二阶梯度优化方法的普及度不高。然而在谷歌大脑与普林斯顿大学等研究者的努力下，二阶梯度优化终于在实战大模型上展现出独特的优势。

研究者表示，为了缩短理论和实际优化效果之间的差距，该论文提出了一种二阶优化的概念性验证，并通过一系列重要的算法与数值计算提升，证明它在实际深度模型中能有非常大的提升。具体而言，在训练深度模型过程中，二阶梯度优化 Shampoo 能高效利用由多核 CPU 和多加速器单元组成的异构硬件架构。并且在大规模机器翻译、图像识别等领域实现了非常优越的性能，要比现有的顶尖一阶梯度下降方法还要好。

推荐：本文的亮点在于研究者提出了真正应用的二阶梯度最优化器，在实战大模型上展现出独特的优势。

## 3.11 激活函数

## 3.12 激活函数汇总

从 ReLU 到 GELU，一文概览神经网络的激活函数

## 3.13 GELU

超越 ReLU 却鲜为人知，3 年后被挖掘：BERT、GPT-2 等都在用的激活函数

Gaussian Error Linear Units (GELUs)

## 3.14 Relu 神经网络的记忆能力

Small ReLU networks are powerful memorizers: a tight analysis of memorization capacity

Chulhee Yun 等人发表“小型 ReLU 网络是强大的记忆器：对记忆能力的严格分析”，表明“具有 Omega(sqrt(N)) 隐藏节点的 3 层 ReLU 网络可以完美地记忆具有 N 个点的大多数数据集。

## 3.15 梯度泄露

梯度会泄漏训练数据？MIT 新方法从梯度窃取训练数据只需几步

Deep Leakage from Gradients

### 3.16 彩票假设

原始论文：The lottery ticket hypothesis: Finding sparse, trainable neural networks

田渊栋从数学上证明 ICLR 最佳论文“彩票假设”，强化学习和 NLP 也适用

fb 的博客：https://ai.facebook.com/blog/understanding-the-generalization-of-lottery-tickets-in-neural-networks

One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers

NLP 中彩票假设的应用：

Playing the lottery with rewards and multiple languages: lottery tickets in RL and NLP

Proving the Lottery Ticket Hypothesis: Pruning is All You Need

Frankle 和 Carbin 在 2018 年提出的彩票假说表明，一个随机初始化的网络包含一个小的子网络，这个子网络在进行单独地训练时，其性能能够与原始网络匹敌。在本文中，研究者证明了一个更有力的假说（正如 Ramanujan 等人在 2019 年所猜想的那样），即对于每个有界分布和每个带有有界权重的目标网络来说，一个具有随机权重的充分过参数化神经网络包含一个具有与目标网络几乎相同准确率的子网络，并且无需任何进一步的训练。

===> 从根本上来说，剪枝随机初始化的神经网络与优化权重值一样重要。

剪枝与学习权重同等重要，Lottery Ticket Hypothesis 第一次被理论证明

### 3.17 知识蒸馏

一文总览知识蒸馏概述

### 3.18 生成模型

AAAI 2020 论文解读：关于生成模型的那些事

Probabilistic Graph Neural Network (PGNN): Deep Generative Probabilistic Graph Neural Networks for Scene Graph Generation

Reinforcement Learning（RL）: Sequence Generation with Optimal-Transport-Enhanced Reinforcement Learning

Action Learning: MALA: Cross-Domain Dialogue Generation with Action Learning

### 3.19 双下降问题

深度学习模型陷阱：哈佛大学与 OpenAI 首次发现“双下降现象”

Deep Double Descent: Where Bigger Models and More Data Hurt

### 3.20 一些疑难杂症

如何发现「将死」的 ReLu? 可视化工具 TensorBoard 助你一臂之力

### 3.21 度量学习

深度度量学习中的损失函数

### 3.22 损失函数

### 3.23 mse vs cross-entropy

不一定回归就用 mse，分类才用交叉熵

https://zhuanlan.zhihu.com/p/362496849

https://www.zhihu.com/question/415245797/answer/1791746717

https://zhuanlan.zhihu.com/p/304462034

https://blog.csdn.net/u011508640/article/details/72815981

各种分布：

https://github.com/graykode/distribution-is-all-you-need

### 3.24 cross-entroy vs nllloss

https://blog.csdn.net/geter_CS/article/details/84857220

https://blog.csdn.net/qq_22210253/article/details/85229988

CrossEntropyLoss 就是把 Softmax-Log-NLLLoss 合并成一步

可以看出nn.LogSoftmax()的对输入的操作就是：

$$log(\frac{exp(x)}{\sum_i exp(x[i])})$$

x是输入向量。

而nn.NLLLoss()的操作是：

$$loss_n = -w_n x_{n,y_n}$$

这里没有设置权重，也就是权重默认为1，x_{n,y_n}表示目标类所对应输入x中值，则loss就为

$$loss = -1 * x[0] = 1.3447$$

### 3.25 L_DMI

NeurIPS 2019 | 一种对噪音标注鲁棒的基于信息论的损失函数

L_DMI: An Information-theoretic Noise-robust Loss Function

https://github.com/Newbeeer/L_DMI

### 3.26 损失函数的 pattern

Loss Landscape Sightseeing with Multi-Point Optimization

https://github.com/universome/loss-patterns

### 3.27 softmax 优化

### 3.28 Mixtape

CMU 杨植麟等人再次瞄准 softmax 瓶颈，新方法 Mixtape 兼顾表达性和高效性

Mixtape: Breaking the Softmax Bottleneck Efficiently

### 3.29 自监督

OpenAI 科学家一文详解自监督学习

https://lilianweng.github.io/lil-log/2019/11/10/self-supervised-learning.html

A Cookbook of Self-Supervised Learning

### 3.30 机器学习 + 博弈论

当博弈论遇上机器学习：一文读懂相关理论

### 3.31 normalization 相关

### 3.32 BN

批归一化到底做了什么？DeepMind 研究者进行了拆解

### 3.33 FRN

超越 BN 和 GN！谷歌提出新的归一化层：FRN

谷歌力作：神经网络训练中的 Batch 依赖性很烦？那就消了它！

Filter Response Normalization Layer: Eliminating Batch Dependence in the Training of Deep Neural Networks

### 3.34 VAE

变分推断（Variational Inference）最新进展简述

### 3.35 优化方法

On Empirical Comparisons of Optimizers for Deep Learning

摘要：优化器选择是当前深度学习管道的重要步骤。在本文中，研究者展示了优化器比较对元参数调优协议的灵敏度。研究结果表明，在解释文献中由最近实证比较得出的排名时，元参数搜索空间可能是唯一最重要的因素。但是，当元参数搜索空间改变时，这些结果会相互矛盾。随着调优工作的不断增加，更一般的优化器性能表现不会比近似于它们的那些优化器差，但最近比较优化器的尝试要么假设这些包含关系没有实际相关性，要么通过破坏包含的方式限制元参数。研究者在实验中发现，优化器之间的包含关系实际上很重要，并且通常可以对优化器比较做出预测。具体来说，流行的自适应梯度方法的性能表现绝不会差于动量或梯度下降法。

推荐：如何选择优化器？本文从数学角度论证了不同优化器的特性，可作为模型构建中的参考资料。

### 3.36 调参

你有哪些 deep learning（rnn、cnn）调参的经验?

### 3.37 表示学习

窥一斑而知全豹，三篇论文遍历 ICLR 2020 新型表征方式

### 3.38 新的结构

### 3.39 NALU

Measuring Arithmetic Extrapolation Performance

摘要：神经算术逻辑单元（NALU）是一种神经网络层，可以学习精确的算术运算。NALU 的目标是能够进行完美的运算，这需要学习到精确的未知算术问题背后的底层逻辑。评价 NALU 性能是非常困难的，因为一个算术问题可能有许多种类的解法。因此，单实例的 MSE 被用于评价和比较模型之间的表现。然而，MSE 的大小并不能说明是否是一个正确的方法，也不能解释模型对初始化的敏感性。因此，研究

者推出了一种「成功标准」，用来评价模型是否收敛。使用这种方法时，可以从很多初始化种子上总结成功率，并计算置信区间。通过使用这种方法总结 4800 个实验，研究者发现持续性的学习算术推导是具有挑战性的，特别是乘法。

推荐：尽管神经算术逻辑单元的出现说明了使用神经网络进行复杂运算推导是可行的，但是至今没有一种合适的评价神经网络是否能够成功收敛的标准。本文填补了这一遗憾，可供对本领域感兴趣的读者参考。

## 3.40 权重无关

https://weightagnostic.github.io/

Weight Agnostic Neural Networks

探索权重无关神经网络

## 3.41 on-lstm

Ordered Neurons: Integrating Tree Structures Into Recurrent Neural Networks

https://zhuanlan.zhihu.com/p/65609763

## 3.42 其他相关理论

## 3.43 因果关系

贝叶斯网络之父 Judea Pearl 力荐、LeCun 点赞，这篇长论文全面解读机器学习中的因果关系

Causality for Machine Learning

由 Judea Pearl 倡导的图形因果推理（graphical causal inference）源于 AI 研究，并且在很长一段时间内，它与机器学习领域几乎没有任何联系。在本文中，研究者探讨了图形因果推理与机器学习之间已建立以及应该建立哪些联系，并介绍了一些关键概念。本文认为，机器学习和 AI 领域的未解决难题在本质上与因果关系有关，并解释了因果关系领域如何理解这些难题。

https://www.cnblogs.com/caoyusang/p/13518354.html

## 3.44 能量模型

ICLR 2020 | 分类器其实是基于能量的模型？判别式分类器设计新思路

Your Classifier is Secretly an Energy Based Model and You Should Treat it Like One

## 3.45 贝叶斯深度学习

正如 Emtiyaz Khan 在他的受邀演讲《基于贝叶斯原理的深度学习》中所强调的那样，贝叶斯学习和深度学习是非常不同的。根据 Khan 的说法，深度学习使用“试错”(trial and error) 的方法——看实验会把我们带向何方——而贝叶斯原理迫使你事先思考假设 (先验)。

与常规的深度学习相比，贝叶斯深度学习主要有两个吸引人的点：不确定性估计和对小数据集的更好的泛化。在实际应用中，仅凭系统做出预测是不够的。知道每个预测的确定性很重要。在贝叶斯学习中，不确定性估计是一个内置特性。

传统的神经网络给出单点估计——使用一组权值在数据点上输出预测。另一方面，贝叶斯神经网络使用网络权值上的概率分布，并输出该分布中所有权值集的平均预测，其效果与许多神经网络上的平均预测相同。因此，贝叶斯神经网络是自然的集合体，它的作用类似于正则化，可以防止过拟合。

拥有数百万个参数的贝叶斯神经网络的训练在计算上仍然很昂贵。收敛到一个后验值可能需要数周时间，因此诸如变分推理之类的近似方法已经变得流行起来。Probabilistic Methods -Variational Inference 类发表了 10 篇关于这种变分贝叶斯方法的论文。

Importance Weighted Hierarchical Variational Inference

A Simple Baseline for Bayesian Uncertainty in Deep Learning

Practical Deep Learning with Bayesian Principles

### 3.46 可解释性

相信你的模型：初探机器学习可解释性研究进展

NeurIPS 2019：两种视角带你了解网络可解释性的研究和进展

Intrinsic dimension of data representations in deep neural networks

对于一个深度网络，网络通过多层神经层渐进的转换输入，这其中的几何解释应该是什么样的呢？本文的作者通过实验发现，以固有维度（ID：intrinsic dimensionality）为切入点，可以发现训练好的网络相比较未训练网络而言，其每层的固有维度数量级均小于每层单元数，而且 ID 的存在可以来衡量网络的泛化性能。

This Looks Like That: Deep Learning for Interpretable Image Recognition

当人遇到图像判断的时候，总是会分解图片并解释分类的理由，而机器在判断的时候总是跟人的判断会有些差距。本文旨在缩小机器分类和人分类之间的差距，提出了一个 ProtoPNet，根据人判断的机理来分类图像。本文网络通过分解图像，得到不同的原型部分，通过组成这些信息最终得到正确的分类。

challenge sets，大部分是英文，中文比较少。构造方法：

- 从已有数据集泛化：改下位词、同义词、反义词
- 从已有数据集只抽出可用的部分
- 使用模板建模具体语言特征
- 对抗样本

### 3.47 子集选择

AAAI 2020 线上分享 | 南京大学：一般约束下子集选择问题的高效演化算法

An Efficient Evolutionary Algorithm for Subset Selection with General Cost Constraints

子集选择问题旨在从 n 个元素中，选择满足约束 c 的一个子集，以最大化目标函数 f。它有很多应用，包括影响力最大化，传感器放置等等。针对这类问题，现有的代表性算法有广义贪心算法和 POMC。广义贪心算法耗时较短，但是受限于它的贪心行为，其找到的解质量往往一般；POMC 作为随机优化算法，可以使用更多的时间来找到质量更好的解，但是其缺乏多项式的运行时间保证。因此，我们提出一个高效的演化算法 EAMC。通过优化一个整合了 f 和 c 的代理函数，它可以在多项式时间内找到目前已知最好的近似解，并且其在多类问题上的试验也显示出比广义贪心算法更好的性能。

AAAI 2020 | 南京大学提出高效演化算法 EAMC：可更好解决子集选择问题

## 4 计算机视觉

### 4.1 cv 数据集

ResNet 图像识别准确率暴降 40 个点！这个 ObjectNet 让世界最强视觉模型秒变水货

实测超轻量中文 OCR 开源项目，总模型仅 17M

https://github.com/ouyanghuiyu/chineseocr_lite

### 4.2 cv 基础

计算机视觉入门大全：基础概念、运行原理、应用案例详解

Pytorch 中的数据增强方式最全解释

传统计算机视觉技术落伍了吗？不，它们是深度学习的「新动能」

Deep Learning vs. Traditional Computer Vision

## 4.3 cv 历史

历史需要重写？AlexNet 之前，早有算法完成计算机视觉四大挑战

图像分类最新技术综述论文：21 种半监督、自监督和无监督学习方法一较高低

## 4.4 cnn 相关

67 页综述深度卷积神经网络架构：从基本组件到结构创新

A Survey of the Recent Architectures of Deep Convolutional Neural Networks

卷积神经网络性能优化

解析卷积的高速计算中的细节，一步步代码带你飞

## 4.5 图像分割

## 4.6 图像分割综述

最全综述 | 图像分割算法

100 个深度图像分割算法，纽约大学 UCLA 等最新综述论文

## 4.7 MoCo

何恺明一作，刷新 7 项检测分割任务，无监督预训练完胜有监督

Momentum Contrast for Unsupervised Visual Representation Learning

## 4.8 PointRend

何恺明团队又出神作：将图像分割视作渲染问题，性能显著提升！

Ross、何恺明等人提出 PointRend：渲染思路做图像分割，显著提升 Mask R-CNN 性能

PointRend: Image Segmentation as Rendering

## 4.9 Graph-FCN

另辟蹊径，中科院自动化所等首次用图卷积网络解决语义分割难题

Graph-FCN for image semantic segmentation

使用深度学习执行语义分割在图像像素分类方面取得了巨大进步。但是，深度学习提取高级特征时往往忽略了局部位置信息（local location information），而这对于图像语义分割而言非常重要。

为了避免上述问题，来自中科院自动化所、北京中医药大学的研究者们提出一个执行图像语义分割任务的图模型 Graph-FCN，该模型由全卷积网络（FCN）进行初始化。首先，通过卷积网络将图像网格数据扩展至图结构数据，这样就把语义分割问题转换成了图节点分类问题；然后，使用图卷积网络解决图节点分类问题。研究者称，这是首次将图卷积网络用于图像语义分割的尝试。该方法在 VOC 数据集上获得了有竞争力的 mIOU 性能，相比原始 FCN 模型有 1.34% 的性能提升。

## 4.10 目标检测

## 4.11 自然场景文字定位

ICDAR 2019 论文：自然场景文字定位技术详解

## 4.12 EfficientDet

比当前 SOTA 小 4 倍、计算量少 9 倍，谷歌最新目标检测器 EfficientDet

EfficientDet: Scalable and Efficient Object Detection

计算机视觉领域，模型效率已经变得越来越重要。在本文中，研究者系统地研究了用于目标检测的各种神经网络架构设计选择，并提出了一些关键的优化措施来提升效率。首先，他们提出了一种加权双向特征金字塔网络（weighted bi-directional feature pyramid network，BiFPN），该网络可以轻松快速地进行多尺度特征融合；其次，他们提出了一种复合缩放方法，该方法可以同时对所有骨干、特征网络和框/类预测网络的分辨率、深度和宽度进行统一缩放。基于这些优化，研究者开发了一类新的目标检测器，他们称之为 EfficientDet。在广泛的资源限制条件下，该检测器始终比现有技术获得更高数量级的效率。具体而言，在没有附属条件的情况下，EfficientDet-D7 在 52M 参数和 326B FLOPS1 的 COCO 数据集上实现了 51.0 mAP 的 SOTA 水平，体积缩小了 4 倍，使用的 FLOPS 减少了 9.3 倍，但仍比先前最佳的检测器还要准确（+0.3% mAP）。

推荐：本文探讨了计算机视觉领域的模型效率问题，分别提出了加权双向特征金字塔网络和复合缩放方法，进而开发了一种新的 EfficientDet 目标检测器，实现了新的 SOTA 水平。

## 4.13 YoloVxxx

超全！YOLO 目标检测从 V1 到 V3 结构详解

【目标检测】YOLOv4 特征提取网络——CSPDarkNet 结构解析及 PyTorch 实现

## 4.14 图像识别

显著提升图像识别网络效率，Facebook 提出 IdleBlock 混合组成方法

Hybrid Composition with IdleBlock: More Efficient Networks for Image Recognition

近年来，卷积神经网络（CNN）已经主宰了计算机视觉领域。自 AlexNet 诞生以来，计算机视觉社区已经找到了一些能够改进 CNN 的设计，让这种骨干网络变得更加强大和高效，其中比较出色的单个分支网络包括 Network in Network、VGGNet、ResNet、DenseNet、ResNext、MobileNet v1/v2/v3 和 ShuffleNet v1/v2。近年来同样吸引了研究社区关注的还有多分辨率骨干网络。作者认为目前实现高效卷积网络的工作流程可以分成两步：1）设计一种网络架构；2）对该网络中的连接进行剪枝。在第一步，作者研究了人类专家设计的架构与搜索得到的架构之间的共同模式：对于每种骨干网络，其架构都是由其普通模块和归约模块（reduction block）的设计所确定的。第二步会将某些连接剪枝去掉，这样就不能保证每个模块都有完整的信息交换了。Facebook AI 的研究者在这篇论文中通过在网络设计步骤中考虑剪枝，为图像识别任务设计了一种更高效的网络。他们创造了一种新的模块设计方法：Idle。

## 4.15 图像补全

拍照总被路人甲抢镜？那就用这个项目消 Ta

## 4.16 文字检测与识别

AAAI 2020 | 旷视研究院：深度解读文字检测与识别新突破

## 4.17 图像合成

SEAN: Image Synthesis with Semantic Region-Adaptive Normalization

本论文要解决的问题是使用条件生成对抗网络（cGAN）生成合成图像。具体来说，本文要完成的具体任务是使用一个分割掩码控制所生成的图像的布局，该分割掩码的每个语义区域都具有标签，而网络可以根据这些标签为每个区域「添加」具有真实感的风格。尽管之前已经有一些针对该任务的框架了，但当前最佳的架构是 SPADE（也称为 GauGAN）。因此，本论文的研究也是以 SPADE 为起点的。具体来说，本文针对原始 SPADE 的两个缺陷提出了新的改进方案。本文在几个高难度的数据集（CelebAMaskHQ、CityScapes、ADE20K 和作者新建的 Facades 数据集）上对新提出的方法进行了广泛的实验评估。定量实验方面，作者基于 FID、PSNR、RMSE 和分割性能等多种指标对新方法进行了评估；定性实验方面，作者展示了可通过视觉观察进行评估的样本。

推荐：图像合成是近来非常热门的研究领域，世界各地的研究者为这一任务提出了许多不同的框架和算法，只为能合成出更具真实感的图像。阿卜杜拉国王科技大学和卡迪夫大学近日提出了一种新改进方案 SEAN，能够分区域对合成图像的内容进行控制和编辑（比如只更换眼睛或嘴），同时还能得到更灵活更具真实感的合成结果。有了这个技术，修图换眼睛时不用再担心风格不搭了。

CVPR 2020 | 让合成图像更真实，上交大提出基于域验证的图像和谐化

### 4.18 人脸识别

面部识别必看！5 篇顶级论文了解如何实现人脸反欺诈、跨姿势识别等

### 4.19 CV 相关比赛

ICCV 2019 COCO & Mapillary 挑战赛冠军团队技术分享

### 4.20 3D 模型相关

图像转换 3D 模型只需 5 行代码，英伟达推出 3D 深度学习工具 Kaolin

内存计算显著降低，平均 7 倍实测加速，MIT 提出高效、硬件友好的三维深度学习方法

Point-Voxel CNN for Efficient 3D Deep Learning

FaceBook 开源 PyTorch3D：基于 PyTorch 的新 3D 计算机视觉库

https://github.com/facebookresearch/pytorch3d

PolyGen: An Autoregressive Generative Model of 3D Meshes

摘要：在本文中，来自 DeepMind 的研究者提出了一种直接建模网格的方法 PolyGen，该方法利用 Transformer 架构来循序地预测网格的顶点和表面。文中提出的 3D 网格深度生成模型 PolyGen 以对象类、三维像素和图像等一系列输入为条件，同时由于该模型是概率性的，因此它可以生成捕获模糊场景中不确定性的样本。

实验表明，该模型能够生成高质量、可用的网格，并为网格建模任务创建对数似然基准。研究者表示，PolyGen 模型能够生成连贯的、多样化的 3D 网格，并且相信可以扩展该模型在计算机视觉、机器人学和 3D 内容创建中的应用。

推荐：本文的亮点在于，研究者将网格生成问题作为自回归序列建模来处理，同时结合了 Transformers 和指针网络的优势，从而能够灵活地建模长度可变的网格序列。

### 4.21 GNN+CV

一文读懂：图卷积在基于骨架的动作识别中的应用

NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding

### 4.22 CV 最新进展

### 4.23 半弱监督

10 亿照片训练，Facebook 半弱监督训练方法刷新 ResNet-50 ImageNet 基准测试

https://github.com/facebookresearch/semi-supervised-ImageNet1K-models

https://ai.facebook.com/blog/billion-scale-semi-supervised-learning

Facebook 将该方法称为“半弱监督”(semi-weak supervision)，是结合了半监督学习和弱监督学习者两种不同训练方法的有点的一种新方法。通过使用 teacher-student 模型训练范式和十亿规模的弱监督数据集，它为创建更准确、更有效的分类模型打开了一扇门。如果弱监督数据集 (例如与公开可用的照片相关联的 hashtags) 不能用于目标分类任务，该方法还可以利用未标记的数据集来生成高度准确的半监督模型。

### 4.24 advprop

Quoc Le 推新论文：打破常规，巧用对抗性样本改进图像识别性能

Adversarial Examples Improve Image Recognition

https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet

对抗样本经常被认为是卷积神经网络的一个威胁。而研究者在这篇论文中提出了相反的论点：对抗网络可以被用来提升图像识别模型的准确率，只要使用正确的方式。研究者在这里提出了 AdvProp，这是一个增强对抗训练方法，能够将对抗样本视为额外样本，以方式过拟合。这一方法的关键在于对对抗样本使用了分离的辅助批归一化，因为它们和正常样本的隐藏分布不同。

研究说明，AdvProp 在很多图像识别任务上提升了一系列模型的性能，而且当模型变得更大的时候，性能也会更好。例如，通过将 AdvProp 用在最新的 EfficientNet-B7 模型上，使用 ImageNet 进行训练，研究者可以取得性能点的提升，如 ImageNet (+0.7%)、ImageNet-C (+6.5%)、ImageNet-A (+7.0%)、Stylized- ImageNet (+4.8%)。而在 增强的 EfficientNet-B8 上，这一方法在没有额外数据的情况下达到了 SOTA——85.5% 的 ImageNet top-1 精确度。这一结果超越了使用 3.5B Instagram 数据和 9.4 倍参数量的最佳模型。

## 4.25 稀疏性 +cv

Fast Sparse ConvNets

从历史发展的角度来看，对有效推理（efficient inference）的追求已经成为研究新的深度学习架构和构建块背后的驱动力之一。近来的一些示例包括：压缩和激发模块（squeeze-and-excitation module）、Xception 中的深度级可分离卷积（depthwise seperable convolution）和 MobileNet v2 中的倒置瓶颈（inverted bottleneck）。在所有这些示例中，生成的构建块不仅实现了更高的有效性和准确率，而且在领域内得到广泛采用。在本文中，来自 DeepMind 和 Google 的研究者们进一步扩展了神经网络架构的有效构建块，并且在没有结合标准基本体（standard primitive）的情况下，他们主张用稀疏对应（sparse counterpart）来替换这些密集基本体（dense primitive）。利用稀疏性来减少参数数量的想法并不新鲜，传统观点也认为理论浮点运算次数的减少不能转化为现实世界的效率增益。

研究者通过提出一类用于 ARM 和 WebAssembly 的有效稀疏核来纠正这种错误观点，并且进行开源作为 XNNPACK 库的组成部分。借助于稀疏标准体（sparse primitive）的有效实现，研究者表明，MobileNet v1、MobileNet v2 和 EfficientNet 架构的稀疏版本在有效性和准确率曲线（efficiency-accuracy curve）上显著优于强大的密集基线（dense baseline）。在骁龙 835 芯片上，他们提出的稀疏网络比同等的密集网络性能增强 1.3-2.4 倍，这几乎相当于 MobileNet-family 一整代的性能提升。研究者希望他们的研究成果可以促进稀疏性更广泛地用作创建有效和准确深度学习架构的工具。

## 4.26 自监督 + 半监督 EnAET

华为美研所推出 EnAET：首次用自监督学习方法加强半监督学习

EnAET: Self-Trained Ensemble AutoEncoding Transformations for Semi-Supervised Learning

https://github.com/wang3702/EnAET

## 4.27 无监督 SimCLR

Hinton 组力作：ImageNet 无监督学习最佳性能一次提升 7%，媲美监督学习

A Simple Framework for Contrastive Learning of Visual Representations

在这篇论文中，研究者发现：

- 多个数据增强方法组合对于对比预测任务产生有效表示非常重要。此外，与有监督学习相比，数据增强对于无监督学习更加有用；
- 在表示和对比损失之间引入一个可学习的非线性变换可以大幅提高模型学到的表示的质量；
- 与监督学习相比，对比学习得益于更大的批量和更多的训练步骤。

基于这些发现，他们在 ImageNet ILSVRC-2012 数据集上实现了一种新的半监督、自监督学习 SOTA 方法——SimCLR。在线性评估方面，SimCLR 实现了 76.5% 的 top-1 准确率，比之前的 SOTA 提升了 7%。在仅使用 1% 的 ImageNet 标签进行微调时，SimCLR 实现了 85.8% 的 top-5 准确率，比之前的 SOTA 方法提升了 10%。在 12 个其他自然图像分类数据集上进行微调时，SimCLR 在 10 个数据集上表现出了与强监督学习基线相当或更好的性能。

## 4.28 图像对抗攻击

胶囊网络显神威：Google AI 和 Hinton 团队检测到针对图像分类器的对抗攻击

# 5 自然语言处理

## 5.1 nlp 综述

https://github.com/PengboLiu/NLP-Papers

## 5.2 LSTM 相关

超生动图解 LSTM 和 GRU，一文读懂循环神经网络!

## 5.3 fasttext&word2vec

注：w2v 训练时的内积不是 2 个 emb-in 的内积，而是 emb-in 和 emb-out 的内积

fasttext 源码解析

- `Dictionary::readWord`：空格分割，一次读出来一个 word
- `Dictionary::add`：每个 word 求个 hash，加进词典时，id 就是从 0 开始的序号，同时记录一下词频
- `Dictionary::threshold`：按词频排序，扔掉低频词
- `Dictionary::initNgrams`：每个词，加上前缀 BOW（<）和后缀（>），然后先扔进这个词的 subwords 里，然后再调用 `Dictionary::computeSubwords` 把这个词的 ngrams 也扔进它的 subwords 里

整个词表，是 word 数 +bucket 这么大，其中 bucket 表示可容纳的 subwords 和 wordNgrams 的数量，默认 200w

## 5.4 分词

【Subword】深入理解 NLP Subword 算法：BPE、WordPiece、ULM

## 5.5 语法解析

EMNLP 2019 最佳论文

Specializing Word Embeddings（for Parsing）by Information Bottleneck

预训练词向量，如 ELMo 和 BERT 包括了丰富的句法和语义信息，使这些模型能够在各种任务上达到 SOTA 表现。在本文中，研究者则提出了一个非常快速的变分信息瓶颈方法，能够用非线性的方式压缩这些嵌入，仅保留能够帮助句法解析器的信息。研究者将每个词嵌入压缩成一个离散标签，或者一个连续向量。在离散的模式下，压缩的离散标签可以组成一种替代标签集。通过实验可以说明，这种标签集能够捕捉大部分传统 POS 标签标注的信息，而且这种标签序列在语法解析的过程中更为精确（在标签质量相似的情况下）。而在连续模式中，研究者通过实验说明，适当地压缩词嵌入可以在 8 种语言中产生更精确的语法解析器。这比简单的降维方法要好。

## 5.6 self-attention

从三大顶会论文看百变 Self-Attention

包学包会，这些动图和代码让你一次读懂「自注意力」

http://jalammar.github.io/illustrated-transformer/

从熵不变性看 Attention 的 Scale 操作

## 5.7 文本匹配

谈谈文本匹配和多轮检索

搜索中的深度匹配模型

## 5.8 机器翻译

102 个模型、40 个数据集，这是你需要了解的机器翻译 SOTA 论文

1. Transformer Big + BT：回译

通过单语数据提升 NMT 模型最高效的方法之一是回译（back-translation）。如果我们的目标是训练一个英语到德语的翻译模型，那么可以首先训练一个从德语到英语的翻译模型，并利用该模型翻译所有的单语德语数据。然后基于原始的英语到德语数据，再加上新生成的数据，我们就能训练一个英语到德语的最终模型。

Understanding Back-Translation at Scale

2. MASS：预训练

MASS: Masked Sequence to Sequence Pre-training for Language Generation

MASS 采用了编码器-解码器框架，并尝试在给定部分句子的情况下修复整个句子。如下所示为 MASS 的框架图，其输入句子包含了一些连续的 Token，并且中间会带有一些连续的 Mask，模型的任务是预测出被 Mask 掉的词是什么。相比 BERT 只有编码器，MASS 联合训练编码器与解码器，能获得更适合机器翻译的表征能力。

这里有：https://daiwk.github.io/posts/nlp-paddle-lark.html#massmicrosoft

## 5.9 nlp 标准 & 数据集

## 5.10 中文 glue

ChineseGLUE：为中文 NLP 模型定制的自然语言理解基准

超 30 亿中文数据首发！首个专为中文 NLP 打造的 GLUE 基准发布

https://github.com/CLUEbenchmark/CLUE

https://www.cluebenchmarks.com/

https://github.com/brightmart/nlp_chinese_corpus

http://thuctc.thunlp.org/#%E4%B8%AD%E6%96%87%E6%96%87%E6%9C%AC%E5%88%86%E7%B1%BB%E6%95%B0%E6%8D%AE%E9%9B%86THUCNews

## 5.11 中文阅读理解数据集

首个中文多项选择阅读理解数据集：BERT 最好成绩只有 68%，86% 问题需要先验知识

Investigating Prior Knowledge for Challenging Chinese Machine Reading Comprehension

https://github.com/nlpdata/c3

## 5.12 物理常识推理任务数据集

PIQA: Reasoning about Physical Commonsense in Natural Language

「在不使用刷子涂眼影的情况下，我应该用棉签还是牙签?」类似这种需要物理世界常识的问题对现今的自然语言理解系统提出了挑战。虽然最近的预训练模型 (如 BERT) 在更抽象的如新闻文章和百科词条这种具有丰富文本信息的领域问答方面取得了进展，但在更现实的领域，由于报导的偏差，文本本质上是有限的，类似于「用牙签涂眼影是一个坏主意」这样的事实很少得到直接报道。人工智能系统能够在不经历物理世界的情况下可靠地回答物理常识问题吗？是否能够捕获有关日常物品的常识知识，包括它们的物理特性、承受能力以及如何操纵它们。

在本文中，研究者介绍了一个关于物理常识推理任务和相应的基准数据集 PIQA（Physical Interaction：Question Answering）进行评估。虽然人类应对这一数据集很容易 (95% 的准确率)，但是大型的预训模型很难 (77%)。作者分析了现有模型所缺乏的知识为未来的研究提供了重要的机遇。

## 5.13 常识推理数据集 WinoGrande

WinoGrande: An Adversarial Winograd Schema Challenge at Scale

研究者提出了 WINOGRANDE，一个有着 44k 个问题的大规模数据集。该数据集在规模和难度上较之前的数据集更大。该数据集的构建包括两个步骤：首先使用众包的方式设计问题，然后使用一个新的 AFLITE 算法缩减系统偏见（systematic bias），使得人类可以察觉到的词汇联想转换成机器可以检测到的嵌入联想（embedding association）。现在最好的 SOTA 模型可以达到的性能是 59.4 -79.1%，比人脸性能水平（94%）低 15-35%（绝对值）。这种性能波动取决于训练数据量（2% 到 100%）。

本论文荣获了 AAAI 2020 最佳论文奖，文中提出的 WINOGRANDE 是一个很好的迁移学习资源；但同时也说明我们现在高估了模型的常识推理的能力。研究者希望通过这项研究能够让学界重视减少算法的偏见。

## 5.14 阅读理解

## 5.15 DCMN+

AAAI 2020 | 云从科技 & 上交大提出 DCMN+ 模型，在多项阅读理解数据集上成绩领先

DCMN+: Dual Co-Matching Network for Multi-choice Reading Comprehension

## 5.16 相关性模型

Yahoo 相关性模型总结

## 5.17 ULMFiT

ULMFiT 面向文本分类的通用语言模型微调

Universal Language Model Fine-tuning for Text Classification

中文 ulmfit：https://github.com/bigboNed3/chinese_ulmfit

归纳迁移学习（Inductive Transfer Learning）对计算机视觉（Compute Vision，CV）产生了巨大影响，但对自然语言处理（Natural Language Processing，NLP）一直没有突破性进展，现有的 NLP 方法，仍然需要根据特定任务进行修改，并且从零开始训练。我们提出了通用语言模型微调（Universal Language Model Fine-tuning for Text Classification，ULMFiT），这是一种有效的迁移学习方法，可以适用于任何 NLP 任务，另外，我们引入了语言模型微调的关键技术。我们的方法在 6 个文本分类任务上显著优于现有的最先进方法，在大部分数据集上将错误率降低了 18-24%。此外，ULMFiT 仅用 100 个标记样本训练出来的性能，可以媲美从零开始训练（Training From Scratch）使用 100 倍以上数据训练出来的性能。

## 5.18 bert/transformer 相关总结

## 5.19 huggingface 的 nlp 预训练模型库

用于 NLP 的预训练 Transformer 模型的开源库。它具有六种架构，分别是：

- Google 的 BERT
- OpenAI 的 GPT 和 GPT-2
- Google / CMU 的 Transformer-XL 和 XLNet
- Facebook 的 XLM

https://github.com/huggingface/transformers

## 5.20 bert

BERT 小学生级上手教程，从原理到上手全有图示，还能直接在线运行

BERT 源码分析（PART I）

BERT 源码分析（PART II）

Dive into BERT：语言模型与知识

关于 BERT，面试官们都怎么问

主要讲了下面 3 篇：

Language Models as Knowledge Bases?

Linguistic Knowledge and Transferability of Contextual Representations

What does BERT learn about the structure of language?

bert 的一些细节：

https://github.com/google-research/bert/blob/master/modeling.py

### 5.20.1 multi-head att 实现

输入原始的 query(即 from_tensor) 之后，把 `[batch, from_seq, emb]` 变成 `[?, emb]`，其中?=batch*from_seq

```
from_tensor_2d = reshape_to_matrix(from_tensor)

def reshape_to_matrix(input_tensor):
    """Reshapes a >= rank 2 tensor to a rank 2 tensor (i.e., a matrix)."""
    ndims = input_tensor.shape.ndims
    if ndims < 2:
        raise ValueError("Input tensor must have at least rank 2. Shape = %s" %
            (input_tensor.shape))
    if ndims == 2:
        return input_tensor
    width = input_tensor.shape[-1]
output_tensor = tf.reshape(input_tensor, [-1, width])
    return output_tensor
```

然后再接一个 fc，把 `[?, emb]` 变成 `[?, head_num * per_head]`，一般 `head_num * per_head=emb`。

```
query_layer = tf.layers.dense(
        from_tensor_2d,
        num_attention_heads * size_per_head,
        activation=query_act,
        name="query",
        kernel_initializer=create_initializer(initializer_range))
```

因为?=batch*from_seq，所以可以直接做如下变换

```
query_layer = transpose_for_scores(query_layer, batch_size,
        num_attention_heads, from_seq_length,
        size_per_head)
```

实际就是把? 拆开成 `batch, from_seq`，整个变成 `[batch, from_seq, head_num, per_head]`，然后做了个 transpose，把 1 和 2 互换了下，得到 `[batch, head_num, from_seq, per_head]`

```
def transpose_for_scores(input_tensor, batch_size, num_attention_heads,
        seq_length, width):
    output_tensor = tf.reshape(
            input_tensor, [batch_size, seq_length, num_attention_heads, width])

    output_tensor = tf.transpose(output_tensor, [0, 2, 1, 3])
    return output_tensor
```

然后 key 也做完全一样的操作 (不过处理的是 to_tensor，如果是 self-attention，那 to_tensor=from_tensor)，得到 `[batch, head_num, to_seq, per_head]`：

```
to_tensor_2d = reshape_to_matrix(to_tensor)
key_layer = tf.layers.dense(
        to_tensor_2d,
        num_attention_heads * size_per_head,
        activation=key_act,
        name="key",
        kernel_initializer=create_initializer(initializer_range))

key_layer = transpose_for_scores(key_layer, batch_size, num_attention_heads,
        to_seq_length, size_per_head)
```

然后就算 $QK^T$ 了，注意这里对 key 取了转置，也就是 [batch, head_num, from_seq, per_head] 乘以 [batch, head_num, per_head, to_seq]，得到的结果是 [batch, head_num, from_seq, to_seq]：

```
attention_scores = tf.matmul(query_layer, key_layer, transpose_b=True)
attention_scores = tf.multiply(attention_scores,
    1.0 / math.sqrt(float(size_per_head)))

if attention_mask is not None:
    # `attention_mask` = [B, 1, F, T]
    attention_mask = tf.expand_dims(attention_mask, axis=[1])

    # Since attention_mask is 1.0 for positions we want to attend and 0.0 for
    # masked positions, this operation will create a tensor which is 0.0 for
    # positions we want to attend and -10000.0 for masked positions.
    adder = (1.0 - tf.cast(attention_mask, tf.float32)) * -10000.0

    # Since we are adding it to the raw scores before the softmax, this is
    # effectively the same as removing these entirely.
    attention_scores += adder
attention_probs = tf.nn.softmax(attention_scores)
attention_probs = dropout(attention_probs, attention_probs_dropout_prob)
```

然后看下 value 的操作：

```
value_layer = tf.layers.dense(
        to_tensor_2d,
        num_attention_heads * size_per_head,
        activation=value_act,
        name="value",
        kernel_initializer=create_initializer(initializer_range))

# `value_layer` = [batch, to_seq, head_num, per_head]
value_layer = tf.reshape(
            value_layer,
            [batch_size, to_seq_length, num_attention_heads, size_per_head])

# `value_layer` = [batch, head_num, to_seq, per_head]
value_layer = tf.transpose(value_layer, [0, 2, 1, 3])

# `context_layer` = [batch, head_num, from_seq, per_head]
context_layer = tf.matmul(attention_probs, value_layer)

# `context_layer` = [batch, from_seq, head_num, per_head]
context_layer = tf.transpose(context_layer, [0, 2, 1, 3])
```

再确认一点，$softmax(QK^T)$ 是 [batch, head_num, from_seq, to_seq]，而 $V$ 是 [batch, head_num, to_seq, per_head]，所以 context_layer 是 [batch, head_num, from_seq, per_head]

最后，再搞一下，变回 [batch, from_seq, head_num * per_head]：

```
if do_return_2d_tensor:
# `context_layer` = [B*F, N*H]
    context_layer = tf.reshape(
        context_layer,
        [batch_size * from_seq_length, num_attention_heads * size_per_head])
else:
# `context_layer` = [B, F, N*H]
    context_layer = tf.reshape(
        context_layer,
        [batch_size, from_seq_length, num_attention_heads * size_per_head])
```

如上过程是 $Concat(head_1, ..., head_h)$，其中 $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$。包装在了函数 attention_layer 之中，我们注意到原文还有一个大小为 $hd_v \times d_{model}$ 的 $W^O$，也就是大小为 $d_{model} \times d_{model}$，再看看源码。。也就是说，正常的 bert 里，attention_heads 就只有一个元素，然后接了个 hidden_size 的 fc，而前面的代码里也提到了 hidden_size 正好就是 $d_{model}$，所以这就是 $W^O$。

```
attention_heads = []
with tf.variable_scope("self"):
    attention_head = attention_layer(xxxxx)
    attention_heads.append(attention_head)
    attention_output = None
    if len(attention_heads) == 1:
        attention_output = attention_heads[0]
    else:
        # In the case where we have other sequences, we just concatenate
        # them to the self-attention head before the projection.
        attention_output = tf.concat(attention_heads, axis=-1)
    # Run a linear projection of `hidden_size` then add a residual
    # with `layer_input`.
    with tf.variable_scope("output"):
        attention_output = tf.layers.dense(
            attention_output,
            hidden_size,
            kernel_initializer=create_initializer(initializer_range))
        attention_output = dropout(attention_output, hidden_dropout_prob)
        attention_output = layer_norm(attention_output + layer_input)
```

关于 mask，可以看看这个https://juejin.im/post/5b9f1af0e51d450e425eb32d

摘抄一下：

什么是 padding mask 呢？回想一下，我们的每个批次输入序列长度是不一样的！也就是说，我们要对输入序列进行对齐！具体来说，就是给在较短的序列后面填充 0。因为这些填充的位置，其实是没什么意义的，所以我们的 attention 机制不应该把注意力放在这些位置上，所以我们需要进行一些处理。具体的做法是，把这些位置的值加上一个非常大的负数 (可以是负无穷)，这样的话，经过 softmax，这些位置的概率就会接近 0！

而 sequence mask 是为了使得 decoder 不能看见未来的信息。也就是对于一个序列，在 time_step 为 t 的时刻，我们的解码输出应该只能依赖于 t 时刻之前的输出，而不能依赖 t 之后的输出。因此我们需要想一个办法，把 t 之后的信息给隐藏起来。那么具体怎么做呢？也很简单：产生一个上三角矩阵，上三角的值全为 1，下三角的值权威 0，对角线也是 0。把这个矩阵作用在每一个序列上，就可以达到我们的目的啦。

### 5.20.2 masked-language-model 的实现

https://github.com/google-research/bert/blob/eedf5716ce1268e56f0a50264a88cafad334ac61/run_pretraining.py#L240

如下，其中 hidden_size 就是是 $d_{model}$:

```
def get_masked_lm_output(bert_config, input_tensor, output_weights, positions,
                         label_ids, label_weights):
  """Get loss and log probs for the masked LM."""
  input_tensor = gather_indexes(input_tensor, positions)

  with tf.variable_scope("cls/predictions"):
    # We apply one more non-linear transformation before the output layer.
    # This matrix is not used after pre-training.
    with tf.variable_scope("transform"):
      input_tensor = tf.layers.dense(
          input_tensor,
          units=bert_config.hidden_size,
          activation=modeling.get_activation(bert_config.hidden_act),
          kernel_initializer=modeling.create_initializer(
              bert_config.initializer_range))
      input_tensor = modeling.layer_norm(input_tensor)

    # The output weights are the same as the input embeddings, but there is
    # an output-only bias for each token.
    output_bias = tf.get_variable(
        "output_bias",
        shape=[bert_config.vocab_size],
        initializer=tf.zeros_initializer())
    logits = tf.matmul(input_tensor, output_weights, transpose_b=True)
    logits = tf.nn.bias_add(logits, output_bias)
    log_probs = tf.nn.log_softmax(logits, axis=-1)

    label_ids = tf.reshape(label_ids, [-1])
    label_weights = tf.reshape(label_weights, [-1])

    one_hot_labels = tf.one_hot(
        label_ids, depth=bert_config.vocab_size, dtype=tf.float32)

    # The `positions` tensor might be zero-padded (if the sequence is too
    # short to have the maximum number of predictions). The `label_weights`
    # tensor has a value of 1.0 for every real prediction and 0.0 for the
    # padding predictions.
    per_example_loss = -tf.reduce_sum(log_probs * one_hot_labels, axis=[-1])
    numerator = tf.reduce_sum(label_weights * per_example_loss)
    denominator = tf.reduce_sum(label_weights) + 1e-5
    loss = numerator / denominator

  return (loss, per_example_loss, log_probs)
```

其中的 gather 如下:

```
def gather_indexes(sequence_tensor, positions):
  """Gathers the vectors at the specific positions over a minibatch."""
  sequence_shape = modeling.get_shape_list(sequence_tensor, expected_rank=3)
```

```
  batch_size = sequence_shape[0]
  seq_length = sequence_shape[1]
  width = sequence_shape[2]

  flat_offsets = tf.reshape(
      tf.range(0, batch_size, dtype=tf.int32) * seq_length, [-1, 1])
  flat_positions = tf.reshape(positions + flat_offsets, [-1])
  flat_sequence_tensor = tf.reshape(sequence_tensor,
                                    [batch_size * seq_length, width])
  output_tensor = tf.gather(flat_sequence_tensor, flat_positions)
  return output_tensor
```

注意调用时传的是如下参数

```
    (masked_lm_loss,
     masked_lm_example_loss, masked_lm_log_probs) = get_masked_lm_output(
         bert_config, model.get_sequence_output(), model.get_embedding_table(),
         masked_lm_positions, masked_lm_ids, masked_lm_weights)
```

## 5.21 gpt-2

15 亿参数最强通用 NLP 模型面世！Open AI GPT-2 可信度高于所有小模型

中文 GPT2

只需单击三次，让中文 GPT-2 为你生成定制故事

https://github.com/imcaspar/gpt2-ml

https://colab.research.google.com/github/imcaspar/gpt2-ml/blob/master/pretrained_model_demo.ipynb

## 5.22 gpt-2 8b

47 分钟，BERT 训练又破全新纪录！英伟达 512 个 GPU 训练 83 亿参数 GPT-2 8B

## 5.23 distill gpt-2

语言模型秒变 API，一文了解如何部署 DistilGPT-2

huggingface 的 distill gpt-2：https://github.com/huggingface/transformers

## 5.24 albert

刚刚，Google 发布 24 个小型 BERT 模型，直接通过 MLM 损失进行预训练

ALBERT：用于语言表征自监督学习的轻量级 BERT

谷歌 ALBERT 模型 V2+ 中文版来了：之前刷新 NLP 各大基准，现在 GitHub 热榜第二

## 5.25 XLNet

XLNet：运行机制及和 Bert 的异同比较

Transformer-XL 与 XLNet 笔记

什么是 XLNet 中的双流自注意力

## 5.26 ELECTRA

2019 最佳预训练模型：非暴力美学，1/4 算力超越 RoBERTa

ELECTRA: 超越 BERT, 19 年最佳 NLP 预训练模型

ELECTRA: pre-training text encoders as discriminators rather than generators

## 5.27 BART

多项 NLP 任务新 SOTA，Facebook 提出预训练模型 BART

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

自监督方法在大量 NLP 任务中取得了卓越的成绩。近期研究通过改进 masked token 的分布（即 masked token 被预测的顺序）和替换 masked token 的可用语境，性能获得提升。然而，这些方法通常聚焦于特定类型和任务（如 span prediction、生成等），应用较为有限。

Facebook 的这项研究提出了新架构 BART，它结合双向和自回归 Transformer 对模型进行预训练。BART 是一个适用于序列到序列模型的去噪自编码器，可应用于大量终端任务。预训练包括两个阶段：1）使用任意噪声函数破坏文本；2）学得序列到序列模型来重建原始文本。BART 使用基于 Tranformer 的标准神经机器翻译架构，可泛化 BERT、GPT 等近期提出的预训练模型。

## 5.28 gated transformer-xl

Stabilizing Transformers for Reinforcement Learning

摘要：得益于预训练语言模型强大的能力，这些模型近来在 NLP 任务上取得了一系列的成功。这需要归功于使用了 transformer 架构。但是在强化学习领域，transformer 并没有表现出同样的能力。本文说明了为什么标准的 transformer 架构很难在强化学习中优化。研究者同时提出了一种架构，可以很好地提升 transformer 架构和变体的稳定性，并加速学习。研究者将提出的架构命名为 Gated Transformer-XL(GTrXL)，该架构可以超过 LSTM，在多任务学习 DMLab-30 基准上达到 SOTA 的水平。

推荐：本文是 DeepMind 的一篇论文，将强化学习和 Transformer 结合是一种新颖的方法，也许可以催生很多相关的交叉研究。

## 5.29 bert/transformer 加速

### 5.29.1 bert 蒸馏、量化、剪枝

BERT 瘦身之路：Distillation，Quantization，Pruning

### 5.29.2 reformer

哈希革新 Transformer：这篇 ICLR 高分论文让一块 GPU 处理 64K 长度序列

Reformer: The Efficient Transformer

https://github.com/google/trax/blob/master/trax/models/research/reformer.py

大型的 Transformer 往往可以在许多任务上实现 sota，但训练这些模型的成本很高，尤其是在序列较长的时候。在 ICLR 的入选论文中，我们发现了一篇由谷歌和伯克利研究者发表的优质论文。文章介绍了两种提高 Transformer 效率的技术，最终的 Reformer 模型和 Transformer 模型在性能上表现相似，并且在长序列中拥有更高的存储效率和更快的速度。论文最终获得了「8，8，6」的高分。在最开始，文章提出了将点乘注意力（dot-product attention）替换为一个使用局部敏感哈希（locality-sensitive hashing）的点乘注意力，将复杂度从 O(L2 ) 变为 O(L log L)，此处 L 指序列的长度。此外，研究者使用可逆残差（reversible residual layers）代替标准残差（standard residuals），这使得存储在训练过程中仅激活一次，而不是 n 次（此处 n 指层数）。最终的 Reformer 模型和 Transformer 模型在性能上表现相同，同时在长序列中拥有更高的存储效率和更快的速度。

大幅减少 GPU 显存占用：可逆残差网络 (The Reversible Residual Network)

### 5.29.3 LTD-bert

内存用量 1/20，速度加快 80 倍，腾讯 QQ 提出全新 BERT 蒸馏框架，未来将开源

#### 5.29.4 Q-bert

AAAI 2020 | 超低精度量化 BERT，UC 伯克利提出用二阶信息压缩神经网络

Q-BERT: Hessian Based Ultra Low Precision Quantization of BERT

#### 5.29.5 Adabert

推理速度提升 29 倍，参数少 1/10，阿里提出 AdaBERT 压缩方法

AdaBERT: Task-Adaptive BERT Compression with Differentiable Neural Architecture Search

### 5.30 t5

谷歌 T5 模型刷新 GLUE 榜单，110 亿参数量，17 项 NLP 任务新 SOTA

谷歌最新 T5 模型 17 项 NLP 任务霸榜 SuperGLUE，110 亿参数量！

### 5.31 哪吒 +tinybert

哪吒”出世！华为开源中文版 BERT 模型

NEZHA: Neural Contextualized Representation for Chinese Language Understanding

https://github.com/huawei-noah/Pretrained-Language-Model

华为诺亚方舟开源哪吒、TinyBERT 模型，可直接下载使用

### 5.32 XLM-R

Facebook 最新语言模型 XLM-R：多项任务刷新 SOTA，超越单语 BERT

Unsupervised Cross-lingual Representation Learning at Scale

来自 facebook。针对多种跨语言的传输任务，大规模地对多语言语言模型进行预训练可以显著提高性能。在使用超过 2TB 的已过滤 CommonCrawl 数据的基础上，研究者在 100 种语言上训练了基于 Transformer 的掩模语言模型。该模型被称为 XLM-R，在各种跨语言基准测试中，其性能显著优于多语言 BERT（mBERT），其中 XNLI 的平均准确度为 + 13.8%，MLQA 的平均 F1 得分为 + 12.3%，而 FQ 的平均 F1 得分为 + 2.1%NER。XLM-R 在低资源语言上表现特别出色，与以前的 XLM 模型相比，斯瓦希里语（Swahili）的 XNLI 准确性提升了 11.8%，乌尔都语（Urdu）的准确性提升了 9.2%。研究者还对获得这些提升所需的关键因素进行了详细的实证评估，包括（1）积极转移和能力稀释；（2）大规模资源资源的高低性能之间的权衡。最后，他们首次展示了在不牺牲每种语言性能的情况下进行多语言建模的可能性。XLM-Ris 在 GLUE 和 XNLI 基准测试中具有强大的单语言模型，因此非常具有竞争力。

### 5.33 transformer+ 生成模型

### 5.34 经典文本生成模型

AI 也能精彩表达：几种经典文本生成模型一览

### 5.35 UniLM

NeurIPS 2019 | 既能理解又能生成自然语言，微软提出统一预训练新模型 UniLM

Unified Language Model Pre-training for Natural Language Understanding and Generation

https://github.com/microsoft/unilm

## 5.36 LaserTagger

谷歌开源文本生成新方法 LaserTagger，直击 seq2seq 效率低、推理慢、控制差三大缺陷！

推断速度达 seq2seq 模型的 100 倍，谷歌开源文本生成新方法 LaserTagger

序列到序列（seq2seq）模型给机器翻译领域带来了巨大变革，并成为多种文本生成任务的首选工具，如文本摘要、句子融合和语法纠错。模型架构改进（如 Transformer）以及通过无监督训练方法利用大型无标注文本数据库的能力，使得近年来神经网络方法获得了质量上的提升。

但是，使用 seq2seq 模型解决文本生成任务伴随着一些重大缺陷，如生成的输出不受输入文本支持（即「幻觉」，hallucination）、需要大量训练数据才能实现优秀性能。此外，由于 seq2seq 模型通常逐词生成输出，因此其推断速度较慢。

谷歌研究人员在近期论文《Encode, Tag, Realize: High-Precision Text Editing》中提出一种新型文本生成方法，旨在解决上述三种缺陷。该方法速度快、精确度高，因而得名 LaserTagger。

Encode, Tag, Realize: High-Precision Text Editing

http://lasertagger.page.link/code

## 5.37 pegasus

华人博士一作：自动生成摘要超越 BERT！帝国理工 & 谷歌提出新模型 Pegasus

PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization

来自帝国理工学院和谷歌大脑团队的研究者提出了大规模文本语料库上具有新的自监督目的的大型 Transformer 预训练编码器-解码器模型 PEGASUS（Pre-training with Extracted Gap-sentences for Abstractive Summarization）。与抽取式文本摘要（extractive summary）相似，在 PEGASUS 模型中，输入文档中删除或 mask 重要句子，并与剩余句子一起作为输出序列来生成。研究者在新闻、科学、故事、说明书、邮件、专利以及立法议案等 12 项文本摘要下游任务上测试了 PEGASUS 模型，结果表明该模型在全部 12 项下游任务数据集上取得了 SOTA 结果（以 ROUGE score 衡量）。此外，该模型在低资源（low-resource）文本摘要中也有非常良好的表现，在仅包含 1000 个示例的 6 个数据集上超越了以往的 SOTA 结果。

## 5.38 T-NLG/DeepSpeed

搞定千亿参数，训练时间只用 1/3，微软全新工具催生超级 NLP 模型

https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

https://github.com/microsoft/DeepSpeed

## 5.39 transformer 应用于检索召回

ICLR2020 cmu+google:

Pre-training Tasks for Embedding-based Large-scale Retrieval

## 5.40 一些 bert/transformer 的应用

美团 BERT 的探索和实践

Bert 时代的创新（应用篇）：Bert 在 NLP 各领域的应用进展

## 5.41 poly-encoder

https://zhuanlan.zhihu.com/p/119444637

## 5.42 bert/transformer 其他

BERT 系列文章汇总导读

ALBERT、XLNet，NLP 技术发展太快，如何才能跟得上节奏?

绝对干货！NLP 预训练模型：从 transformer 到 albert

ALBERT 一作蓝振忠：预训练模型应用已成熟，ChineseGLUE 要对标 GLUE 基准

有哪些令你印象深刻的魔改 Transformer?

BERT 模型超酷炫，上手又太难？请查收这份 BERT 快速入门指南！

https://github.com/jalammar/jalammar.github.io/blob/master/notebooks/bert/A_Visual_Notebook_to_Using_BERT_for_the_First_Time.ipynb

Transformers Assemble（PART I） 讲了 3 篇

Transformers Assemble（PART II） 又讲了三篇

站在 BERT 肩膀上的 NLP 新秀们（PART III）

BERT 时代与后时代的 NLP

新预训练模型 CodeBERT 出世，编程语言和自然语言都不在话下，哈工大、中山大学、MSRA 出品

AAAI 2020 | BERT 稳吗？亚马逊、MIT 等提出针对 NLP 模型的对抗攻击框架 TextFooler

A Primer in BERTology: What we know about how BERT works

摘要：目前，基于 Transformer 的模型已经广泛应用于自然语言处理中，但我们依然对这些模型的内部工作机制知之甚少。在本文中，来自麻省大学洛威尔分校的研究者对流行的 BERT 模型进行综述，并综合分析了 40 多项分析研究。他们还概览了对模型和训练机制提出的改进，然后描画了未来的研究方向。

Pre-trained Models for Natural Language Processing: A Survey

## 5.43 对话

## 5.44 对话数据集

谷歌发布世界最大任务型对话数据集 SGD，让虚拟助手更智能

Towards Scalable Multi-domain Conversational Agents: The Schema-Guided Dialogue Dataset

## 5.45 对话领域的传统模型

HMM 模型在贝壳对话系统中的应用

## 5.46 convai

GitHub 超 1.5 万星 NLP 团队热播教程：使用迁移学习构建顶尖会话 AI

https://convai.huggingface.co/

https://github.com/huggingface/transfer-learning-conv-ai

## 5.47 微软小冰

微软小冰是怎样学会对话、唱歌和比喻？我们听三位首席科学家讲了讲背后的原理

## 5.48 RASA

【RASA 系列】语义理解（上）

### 5.49 生成式对话

生成式对话 seq2seq：从 rnn 到 transformer

### 5.50 开放领域聊天机器人

Towards a Human-like Open-Domain Chatbot

不再鹦鹉学舌：26 亿参数量，谷歌开放领域聊天机器人近似人类水平

### 5.51 问答系统

AAAI 2020 提前看 | 三篇论文解读问答系统最新研究进展

Improving Question Generation with Sentence-level Semantic Matching and Answer Position Inferring

TANDA: Transfer and Adapt Pre-Trained Transformer Models for Answer Sentence Selection

On the Generation of Medical Question-Answer Pairs

### 5.52 NER

OpenNRE 2.0：可一键运行的开源关系抽取工具包

https://github.com/thunlp/OpenNRE

### 5.53 知识图谱

NAACL 2019 开源论文：基于胶囊网络的知识图谱完善和个性化搜索

知识图谱从哪里来：实体关系抽取的现状与未来

### 5.54 关系提取

关系提取简述

### 5.55 常识知识与常识推理

AAAI 2020 学术会议提前看：常识知识与常识推理

## 6 语音算法

### 6.1 语音数据集

### 6.2 中文音乐数据集

中文歌词生成，缺不缺语料？这里有一个开源项目值得推荐

https://github.com/yangjianxin1/QQMusicSpider

数据集链接: https://pan.baidu.com/s/1WNYLcOrd3hOiATu44gotBg 提取码: cy6f

### 6.3 时域音频分离模型

时域音频分离模型登 GitHub 热榜，效果超传统频域方法，Facebook 官方出品

### 6.4 中文语音识别

实战：基于 tensorflow 的中文语音识别模型 | CSDN 博文精选

## 6.5 顺滑度

赛尔原创 | AAAI20 基于多任务自监督学习的文本顺滑研究

Multi-Task Self-Supervised Learning for Disfluency Detection

自动语音识别（ASR）得到的文本中，往往含有大量的不流畅现象。这些不流畅现象会对后面的自然语言理解系统（如句法分析，机器翻译等）造成严重的干扰，因为这些系统往往是在比较流畅的文本上训练的。不流畅现象主要分为两部分，一部分是 ASR 系统本身识别错误造成的，另一部分是 speaker 话中自带的。NLP 领域主要关注的是 speaker 话中自带的不流畅现象，ASR 识别错误则属于语音识别研究的范畴。顺滑 (Disfluency Detection) 任务的目的就是要识别出 speaker 话中自带的不流畅现象。

## 6.6 语音识别加速

GPU 解码提升 40 倍，英伟达推进边缘设备部署语音识别，代码已开源

## 6.7 唇读

Hearing Lips: Improving Lip Reading by Distilling Speech Recognizers

年来，得益于深度学习和大型数据集的可用性，唇读（lip reading）已经出现了前所未有的发展。尽管取得了鼓舞人心的结果，但唇读的性能表现依然弱于类似的语音识别，这是因为唇读刺激因素的不确定性导致很难从嘴唇运动视频中提取判别式特征（discriminant feature）。

在本文中，来自浙江大学、斯蒂文斯理工学院和阿里巴巴的研究者提出了一种名为 LIBS（Lip by Speech）的方法，其目的是通过学习语音识别器来增强唇读效果。方法背后的基本原理是：提取自语音识别器的特征可能提供辅助性和判别式线索，而这些线索从嘴唇的微妙运动中很难获得，并且这些线索会因此促进唇阅读器的训练。具体而言，这是通过将语音识别器中的多粒度知识蒸馏到唇阅读器实现的。为了进行这种跨模式的知识蒸馏，研究者不仅利用有效的对齐方案来处理音频和视频之间长度不一致的问题，而且采用一种创造性的过滤策略来重新定义语音识别器的预测结果。研究者提出的方法在 CMLR 和 LRS2 数据集上取得了新的 SOTA 结果，在字符误差率（Character Error Rate，CER）方面分别超出基准方法 7.66% 和 2.75%。

## 6.8 Live caption

借助 Live Caption 在设备上生成字幕

## 6.9 demucs

Demucs: Deep Extractor for Music Sources with extra unlabeled data remixed

https://github.com/facebookresearch/demucs

在录制某些歌曲时，每种乐器都分别录制到单独的音轨或 stem 中。之后在混音和母带阶段，这些词干被合并在一起，生成歌曲。本文的目的是找到这一过程的逆向过程的方法，也就是说要从完成的歌曲中提取每个单独的 stem。这个问题的灵感源自所谓“鸡尾酒会效应”，是说人脑可以从一个嘈杂的聊天室的环境中将单独对话分离出来，并专注于这个特定的对话，自带降噪效果。

本文提出的体系架构是 SING 神经网络体系结构和 Wave-U-Net 的思想的结合。前者用于符号到乐器的音乐合成，而后者是从混音中提取 stem 的方法之一。本质上是 LSTM、卷积层与 U-Net 架构的结合。其中卷积层负责体系结构的编码，LSTM 层用于解码。为了提高模型性能，本文中的架构不使用批量归一化层。

## 6.10 语音版 bert

语音版 BERT？滴滴提出无监督预训练模型，中文识别性能提升 10% 以上

Improving transformer-based speech recognition using unsupervised pre-training

## 6.11 搜狗录音笔

投喂 4 万种噪声，20 种语言方言实时转录，搜狗「开挂」录音笔这样炼成

非常时期，搜狗新一代“AI 笔皇”问世！支持同声传译，转写准确率 98%

### 6.12 音乐推荐相关

https://paperswithcode.com/search?q_meta=&q=music+recommend

spotify 的 paper:

https://research.atspotify.com/?s=playlist&type=publications

歌单生成:

A Comparison of Methods for Treatment Assignment with an Application to Playlist Generation

探索利用:

Explore, Exploit, and Explain: Personalizing Explainable Recommendations with Bandits

## 7 视频算法

### 7.1 视频数据集

### 7.2 VTAB

The Visual Task Adaptation Benchmark

谷歌 AI 推出了「视觉任务适应性基准」(Visual Task Adaptation Benchmark，VTAB)。这是一个多样性的、真实的和具有挑战性的表征基准。这一基准基于以下原则：在所需领域内数据有限的情况下，更好的表征应当能够在未见任务上实现更佳的性能。受启发于推动其他机器学习领域进展的一些基准，如用于自然图像分类的 ImageNet、自然语言处理的 GLUE 和强化学习的 Atari，VTAB 遵循相似的准则：(i) 对解决方案施加最小约束，以鼓励创造性；(ii) 注重实际；(iii) 借助挑战性任务进行评估。

### 7.3 视频检索

用语言直接检索百万视频，这是阿里 TRECVID 视频检索冠军算法

### 7.4 视频编码相关

TIP 2019 开源论文：基于深度学习的 HEVC 多帧环路滤波方法

### 7.5 视频显著区域检测

AAAI 2020 | 速度提升 200 倍，爱奇艺 & 北航等提出基于耦合知识蒸馏的视频显著区域检测算法

Ultrafast Video Attention Prediction with Coupled Knowledge Distillation

### 7.6 视频理解

### 7.7 pyslowfast

视频识别 SOTA 模型都在这了—PySlowFast! Facebook AI Research 开源视频理解前沿算法代码库

https://github.com/facebookresearch/SlowFast

### 7.8 视频插帧相关

### 7.9 DAIN

Depth-Aware Video Frame Interpolation

https://github.com/baowenbo/DAIN

视频帧合成是信号处理领域的一个有趣的分支。通常，这都是关于在现有视频中合成视频帧的。如果在视频帧之间完成操作，则称为内插(interpolation)；而在视频帧之后进行此操作，则称为外推（extrapolation)。视频帧内插是一个长期存在的课题，并且已经在文献中进行

了广泛的研究。这是一篇利用了深度学习技术的有趣论文。通常，由于较大的物体运动或遮挡，插值的质量会降低。在本文中，作者使用深度学习通过探索深度信息来检测遮挡。

他们创建了称为“深度感知视频帧内插”（Depth-Aware video frame INterpolation，DAIN）的架构。该模型利用深度图、局部插值核和上下文特征来生成视频帧。本质上，DAIN 是基于光流和局部插值核，通过融合输入帧、深度图和上下文特征来构造输出帧。在这些文章中，我们有机会看到一些有趣的论文和在深度学习领域取得的进步。这一领域在不断发展，我们预计 2020 年会更有趣。

## 7.10 Quadratic Video Interpolation

NeurIPS 2019 Spotlight | 超清还不够，商汤插帧算法让视频顺滑如丝

这个方法的论文被 NeurIPS 2019 接收为 Spotlight 论文，该方法还在 ICCV AIM 2019 VideoTemporal Super-Resolution Challenge 比赛中获得了冠军。

Quadratic Video Interpolation

## 7.11 TVN

单 CPU 处理 1s 视频仅需 37ms、GPU 仅需 10ms，谷歌提出 TVN 视频架构

Tiny Video Networks

## 7.12 MvsGCN：多视频摘要

MvsGCN: A Novel Graph Convolutional Network for Multi-video Summarization

试图为视频集合生成单个摘要的多视频摘要，是处理不断增长的视频数据的重要任务。在本文中，我们第一个提出用于多视频摘要的图卷积网络。这个新颖的网络衡量了每个视频在其自己的视频以及整个视频集中的重要性和相关性。提出了一种重要的节点采样方法，以强调有效的特征，这些特征更有可能被选择作为最终的视频摘要。为了解决视频摘要任务中固有的类不平衡问题，提出了两种策略集成到网络中。针对多样性的损失正则化用于鼓励生成多样化的摘要。通过大量的实验，与传统的和最新的图模型以及最新的视频摘要方法进行了比较，我们提出的模型可有效地生成具有良好多样性的多个视频的代表性摘要。它还在两个标准视频摘要数据集上达到了最先进的性能。

## 7.13 A-GANet

Deep Adversarial Graph Attention Convolution Network for Text-Based Person Search

新出现的基于文本的行人搜索任务旨在通过对自然语言的查询以及对行人的详细描述来检索目标行人。与基于图像/视频的人搜索（即人重新识别）相比，它实际上更适用，而不需要对行人进行图像/视频查询。在这项工作中，我们提出了一种新颖的深度对抗图注意力卷积网络（A-GANet），用于基于文本的行人搜索。A-GANet 利用文本和视觉场景图，包括对象属性和关系，从文本查询和行人画廊图像到学习信息丰富的文本和视觉表示。它以对抗性学习的方式学习有效的文本-视觉联合潜在特征空间，弥合模态差距并促进行人匹配。具体来说，A-GANet 由图像图注意力网络，文本图注意力网络和对抗学习模块组成。图像和文本图形注意网络设计了一个新的图注意卷积层，可以在学习文本和视觉特征时有效利用图形结构，从而实现精确而有区别的表示。开发了具有特征转换器和模态鉴别器的对抗学习模块，以学习用于跨模态匹配的联合文本-视觉特征空间。在两个具有挑战性的基准（即 CUHK-PEDES 和 Flickr30k 数据集）上的大量实验结果证明了该方法的有效性。

## 7.14 VRD-GCN

Video Relation Detection with Spatio-Temporal Graph

我们从视觉内容中看到的不仅是对象的集合，还包括它们之间的相互作用。用三元组 <subject，predicate，object> 表示的视觉关系可以传达大量信息，以供视觉理解。与静态图像不同，由于附加的时间通道，视频中的动态关系通常在空间和时间维度上都相关，这使得视频中的关系检测变得更加复杂和具有挑战性。在本文中，我们将视频抽象为完全连接的时空图。我们使用图卷积网络使用新颖的 VidVRD 模型在这些 3D 图中传递消息并进行推理。我们的模型可以利用时空上下文提示来更好地预测对象及其动态关系。此外，提出了一种使用暹罗网络的在线关联方法来进行精确的关系实例关联。通过将我们的模型（VRD-GCN）与所提出的关联方法相结合，我们的视频关系检测框架在最新基准测试中获得了最佳性能。我们在基准 ImageNet-VidVRD 数据集上验证了我们的方法。实验结果表明，我们的框架在很大程度上领先于最新技术，一系列的消去研究证明了我们方法的有效性。

## 7.15 video caption

AAAI 2020 | 北理工 & 阿里文娱：结合常识与推理，更好地理解视频并生成描述

Joint Commonsense and Relation Reasoning for Image and Video Captioning

北京理工大学和阿里合作的一篇关于利用对象之间的关系进行图像和视频描述 (image caption/video caption) 的论文。大多数现有方法严重依赖于预训练的对象及其关系的检测器，因此在面临诸如遮挡，微小物体和长尾类别等检测挑战时可能效果不佳。

在本文中，研究者提出了一种联合常识和关系推理的方法 (C-R Reasoning)，该方法利用先验知识进行图像和视频描述，而无需依赖任何目标检测器。先验知识提供对象之间的语义关系和约束，作为指导以建立概括对象关系的语义图，其中一些对象之间的关系是不能直接从图像或视频中获得。特别是，本文的方法是通过常识推理和关系推理的迭代学习算法交替实现的，常识推理将视觉区域嵌入语义空间以构建语义图，关系推理用于编码语义图以生成句子。作者在几个基准数据集上的实验验证了该方法的有效性。

这篇论文并不是聚焦于常识知识和常识推理本身，而是联合常识和关系推理使得图像和视频描述中那些「难以捉摸」,「并非直接可见」的物体或关系现形，使得描述更加精准。

## 7.16 小视频推荐

## 7.17 MMGCN

MMGCN: Multi-modal Graph Convolution Network for Personalized Recommendation of Micro-video

个性化推荐在许多在线内容共享平台中起着核心作用。为了提供优质的微视频推荐服务，重要的是考虑用户与项目（即短视频）之间的交互以及来自各种模态（例如视觉，听觉和文本）的项目内容。现有的多媒体推荐作品在很大程度上利用多模态内容来丰富项目表示，而为利用用户和项目之间的信息交换来增强用户表示并进一步捕获用户对不同模式的细粒度偏好所做的工作却较少。在本文中，我们建议利用用户-项目交互来指导每种模式中的表示学习，并进一步个性化微视频推荐。我们基于图神经网络的消息传递思想设计了一个多模态图卷积网络（MMGCN）框架，该框架可以生成用户和微视频的特定模态表示，以更好地捕获用户的偏好。具体来说，我们在每个模态中构造一个 user-item 二部图，并用其邻居的拓扑结构和特征丰富每个节点的表示。通过在三个公开可用的数据集 Tiktok，Kwai 和 MovieLens 上进行的大量实验，我们证明了我们提出的模型能够明显优于目前最新的多模态推荐方法。

## 7.18 ALPINE

Routing Micro-videos via A Temporal Graph-guided Recommendation System

在过去的几年中，短视频已成为社交媒体时代的主流趋势。同时，随着短视频数量的增加，用户经常被他们不感兴趣的视频所淹没。尽管现有的针对各种社区的推荐系统已经取得了成功，但由于短视频平台中的用户具有其独特的特征：多样化的动态兴趣，多层次的兴趣以及负样本，因此它们无法应用于短视频的一种好的方式。为了解决这些问题，我们提出了一个时间图指导的推荐系统。特别是，我们首先设计了一个新颖的基于图的顺序网络，以同时对用户的动态兴趣和多样化兴趣进行建模。同样，可以从用户的真实负样本中捕获不感兴趣的信息。除此之外，我们通过用户矩阵将用户的多层次兴趣引入推荐模型，该矩阵能够学习用户兴趣的增强表示。最后，系统可以通过考虑上述特征做出准确的推荐。在两个公共数据集上的实验结果证明了我们提出的模型的有效性。

## 7.19 长视频剪辑

让 UP 主不再为剪视频发愁，百度等提出用 AI 自动截取故事片段

TruNet: Short Videos Generation from Long Videos via Story-Preserving Truncation

## 7.20 AutoFlip

不想横屏看视频？谷歌开源框架 AutoFlip 一键截出最精彩竖版视频

在使用过程中，只需要将一段视频和目标维度（如截取的长宽比类型）作为输入，AutoFlip 会分析视频内容并提出一个优化路径和裁剪策略，最后输出一段视频。

https://github.com/google/mediapipe

https://github.com/google/mediapipe/blob/master/mediapipe/docs/autoflip.md

### 7.21 快手视频相关工作

同为工业界最大的推荐业务场景，快手短视频推荐与淘宝推荐有何不同?

AI 碰撞短视频，从推荐到直播，快手探索了这些 ML 新思路

视频推荐、内容分发优化、视频码率优化这三方面探索提升快手视频体验的新方案。

### 7.22 EIUM：讲究根源的快手短视频推荐

Explainable Interaction-driven User Modeling over Knowledge Graph for Sequential Recommendation

### 7.23 Comyco：基于质量感知的码率自适应策略

Comyco: Quality-aware Adaptive Video Streaming via Imitation Learning

### 7.24 Livesmart：智能 CDN 调度

Livesmart: a QoS-Guaranteed Cost-Minimum Framework of Viewer Scheduling for Crowdsourced Live Streaming

### 7.25 抖音视频相关工作

图解抖音推荐算法

### 7.26 google 视频相关工作

通过未标记视频进行跨模态时间表征学习

两篇:

VideoBERT: A Joint Model for Video and Language Representation Learning，VideoBert 模型。

Contrastive Bidirectional Transformer for Temporal Representation Learning，CBT 模型。

### 7.27 阿里短视频推荐相关工作

淘宝如何拥抱短视频时代？视频推荐算法实战

## 8 GAN

### 8.1 GAN 综述

密歇根大学最新 28 页综述论文《GANs 生成式对抗网络综述：算法、理论与应用》，带你全面了解 GAN 技术趋势

### 8.2 LOGAN

BigGAN 被干了！DeepMind 发布 LOGAN：FID 提升 32%，华人一作领衔

### 8.3 ShapeMatchingGAN

ICCV 2019 开源论文 | ShapeMatchingGAN：打造炫酷动态的艺术字

### 8.4 图像生成 +GAN

在图像生成领域里，GAN 这一大家族是如何生根发芽的

## 8.5 模式崩塌问题

GAN:「太难的部分我就不生成了，在下告退」

Seeing What a GAN Cannot Generate

## 8.6 imagestylegan++

Image2StyleGAN++: How to Edit the Embedded Images?

研究者提出了一个名为 Image2StyleGAN++ 的网络，是一种多应用的图像编辑框架。这一框架从三个方面扩展了近来提出的 Image2StyleGAN。首先，研究者引入了噪声优化机制，用来弥补 W+ 隐空间嵌入。这一噪声优化机制可以重置图像中的高频特征，并显著提升重建图像的质量。其次，研究者扩展了全局 W+ 印控机嵌入，以便局部嵌入。第三，研究者将嵌入和激活张量（activation tensor）操纵结合，让局部编辑像全局情感编辑那样有着很高的图像质量。这种编辑方法能够推动很多高质量图像编辑应用，如图像重建、重着色、图像混合、局部风格迁移等。

## 8.7 stylegan2

英伟达发布最强图像生成器 StyleGAN2，生成图像逼真到吓人

如果没有 StyleGAN2，真以为初代就是巅峰了：英伟达人脸生成器高能进化，弥补重大缺陷

https://github.com/NVlabs/stylegan2

Analyzing and Improving the Image Quality of StyleGAN

## 8.8 starganv2

StarGAN v2: Diverse Image Synthesis for Multiple Domains

https://github.com/clovaai/stargan-v2

特别是在图像创建和处理方面。这个领域中一个非常有趣的问题就是所谓的“图像到图像转换问题”，我们希望将特征从一个图像域转移到另一个图像域（这里的“图像域”代表可以归类为视觉上独特的类别的一组图像）。我们喜欢 CycleGAN 和 StarGAN 等旨在解决此问题的解决方案，因此您可以想象几天前看到 StarGAN v2 论文时我们有多么兴奋。

本文还讨论了另一个问题——域的可伸缩性。这意味着它可以同时解决多个图像域的问题。本质上，这个架构依赖于 StarGAN 早期版本的成功，并为其添加了样式层。它由四个模块组成：第一个模块是生成器，它负责将输入图像转换为反映域特定样式的输出图像；接下来是映射网络转换器（Mapping Network Transformer），它将潜在代码转换为多个域的样式代码；第三个是样式编码器，它提取图像的样式并将其提供给生成器；最后，判别器可以从多个域中区分真实图像和伪图像。

## 8.9 nlp+gan

AAAI 2020 线上分享 | Bert 稳吗？解读 NLP 对抗模型新进展

Is BERT Really Robust? A Strong Baseline for Natural Language Attack on Text Classification and Entailment

众所周知，CV 领域的 adversarial attack 被非常广泛的研究，但是在 NLP 领域的对抗攻击却因为文本的离散的特性而难以推进。对于 NLP 的模型来说，那些在人们眼里几乎没变的文本却会被模型非常不同地对待，甚至错判。这些是特别致命的、且急需研究的方向。这是一篇与 MIT 合作的 AAAI 2020 Oral 文章，自然语言对抗样本生成，我们将详细解读如何简单高效地生成自然语言对抗样本，并且高度 attack 文本分类和文本推测的 7 个数据集。

## 8.10 gan 压缩

韩松、朱俊彦等人提出 GAN 压缩法：算力消耗不到 1/9，现已开源

# 9 推荐系统

https://daiwk.github.io/posts/links-navigation-recommender-system.html

## 9.1 推荐系统整体梳理

https://github.com/Doragd/Algorithm-Practice-in-Industry

王喆的机器学习笔记系列：

https://github.com/wzhe06/Reco-papers

https://github.com/wzhe06/Ad-papers

深度学习传送门系列：

https://github.com/imsheridan/DeepRec

推荐系统遇上深度学习系列：

链接: https://pan.baidu.com/s/1jZkJ2d9WckbZL48aGFudOA 密码:kme3

推荐系统技术演进趋势：召回-> 排序-> 重排

推荐系统的发展与 2019 最新论文回顾

深度推荐系统 2019 年度阅读收藏清单

推荐工业界实战角度详解 TensorFlow 中 Wide & Deep 源码（三）

## 9.2 推荐中的采样

batch 内 shuffle 采样（有放回）

On Sampling Strategies for Neural Network-based Collaborative Filtering

浅谈个性化推荐系统中的非采样学习

Sampling-Bias-Corrected Neural Modeling for Large Corpus Item Recommendations

https://www.tensorflow.org/extras/candidate_sampling.pdf

推荐系统遇上深度学习 (七十二)-[谷歌] 采样修正的双塔模型

## 9.3 序列建模

一文看懂序列推荐建模的最新进展与挑战

从 MLP 到 Self-Attention，一文总览用户行为序列推荐模型

## 9.4 bias v.s. debias

推荐系统炼丹笔记：推荐系统 Bias 大全 | Debias 方法综述

## 9.5 position bias

搜索、推荐业务中 - position bias 的工业界、学术界 发展历程 - 系列 1(共计 2)

推荐系统遇上深度学习 (七十一)-[华为] 一种消除 CTR 预估中位置偏置的框架

PAL: A Position-bias Aware Learning Framework for CTR Prediction in Live Recommender Systems

推荐系统之 Position-Bias 建模

## 9.6 用户模型

## 9.7 PeterRec

仅需少量视频观看数据，即可精准推断用户习惯：腾讯、谷歌、中科大团队提出迁移学习架构 PeterRec

Parameter-Efficient Transfer from Sequential Behaviors for User Modeling and Recommendation

https://github.com/fajieyuan/sigir2020_peterrec

搞一个 pretrain-finetune 的架构，学好一套用户的表示，可以给各种下游任务用。

采用如下方式：

- **无监督**地学习用户表示：使用**序列模型**，**预测**用户的**下一次点击**。为了能建模**超长**的 u-i 交互序列，使用类似 NextItNet（A Simple Convolutional Generative Network for Next Item Recommendation）的模型
- 使用预训练好的模型去**有监督**地 finetune 下游任务
- 在各个下游任务间，想要尽可能共享更多的网络参数：参考 learning to learn，即一个网络的大部分参数可以其他参数来预测（一层里 95% 的参数可以通过剩下的 5% 的参数来预测）。文章提出了 model patch(模型补丁)，每个模型补丁的参数量不到原始预训练模型里的卷积层参数的 10%。通过加入模型补丁，不仅可以保留原来的预训练参数，还可以更好地适应下游任务。模型补丁有串行和并行两种加入方式。

序列推荐模型：

- RNN：强序列依赖
- CNN：可并行，能比 RNN 叠更多层，所以准确率更高。难以建模长序列是因为卷积核一般都比较小（如 3x3），但可以通过空洞 (dilated) 卷积来解决，可以使用不变的卷积核，指数级地扩充表示域。
- 纯 attention：可并行，例如 SASRec（Self-attentive sequential recommendation）。但因为时间和存储消耗是序列长度的平方的复杂度。

考虑到用户的点击序列往往成百上千，所以使用类似 NextItNet 的 casual 卷积，以及类似 GRec（Future Data Helps Training: Modeling Future Contexts for Session-based Recommendation）的双向 encoder 的这种 non-casual 卷积。

与推荐系统现有的 transfer learning 对比：

- DUPN：
  - 训练的时候就有多个 loss。如果没有相应的 loss 和 data，学好的用户表示效果就会很差。而本文只有一个 loss，却能用在多个 task 上，所以算是一种 multi-domain learning(Efficient parametrization of multi-domain deep neural networks)
  - DUPN 在用户和 item 特征上需要很多特征工程，并没有显式地对用户的行为序列建模
  - DUPN 要么 finetune 所有参数，要么只 finetune 最后一个分类层。PeterRec 则是对网络的一小部分进行 finetune，效果并不比全 finetune 差，比只 finetune 最后一个分类层要好很多
- CoNet：杨强提出的Conet: Collaborative cross networks for cross-domain recommendation
  - cross-domain 用于推荐的一个网络。同时训练 2 个目标函数，一个表示 source 网络，一个表示 target 网络。
  - pretrain+finetune 效果不一定好，取决于预训练的方式、用户表示的表达能力、预训练的数据质量等

预训练时没有 [TCL]，fintune 时加上。

- 原 domain$S$：有大量用户交互行为的图文或视频推荐。一条样本包括 $(u, x^u) \in \mathcal{S}$，其中，$x^u = \{x_1^u, ..., x_n^u\}$ $(x_i^u \in X)$ 表示用户的点击历史
- 目标 domain$T$：可以是用户 label 很少的一些预测任务。例如用户可能喜欢的 item、用户性别、用户年龄分桶等。一条样本包括 $(u, y) \in \mathcal{T}$，其中 $y \in \mathcal{Y}$ 是一个有监督的标签。

## 9.8 召回

360 展示广告召回系统的演进

推荐场景中深度召回模型的演化过程

https://github.com/imsheridan/DeepRec/tree/master/Match

精准推荐的秘术：阿里解耦域适应无偏召回模型详解对应Co-training Disentangled Domain Adaptation Network for Leveraging Popularity Bias in Recommenders

## 9.9 JTM

下一代深度召回与索引联合优化算法 JTM

## 9.10 DR

字节最新复杂召回模型，提出深度检索 DR 框架解决超大规模推荐系统中的匹配问题

Deep Retrieval: An End-to-End Learnable Structure Model for Large-Scale Recommendations

## 9.11 transformer+ 推荐

Transformer 在推荐模型中的应用总结

## 9.12 多目标

cgc 参考 paddle 代码：cgc_demo.py

## 9.13 工业界的一些推荐应用

## 9.14 dlrm

Facebook 深度个性化推荐系统经验总结 (阿里内部分享 PPT))

## 9.15 混合推荐架构

混合推荐系统就是多个推荐系统“大杂烩”吗?

## 9.16 instagram 推荐系统

Facebook 首次揭秘：超过 10 亿用户使用的 Instagram 推荐算法是怎样炼成的?

https://venturebeat.com/2019/11/25/facebook-details-the-ai-technology-behind-instagram-explore/

Instagram 个性化推荐工程中三个关键技术是什么?

## 9.17 微信读书推荐系统

微信读书怎么给你做推荐的?

## 9.18 youtube 推荐梳理

一文总览近年来 YouTube 推荐系统算法梳理

## 9.19 认知推荐

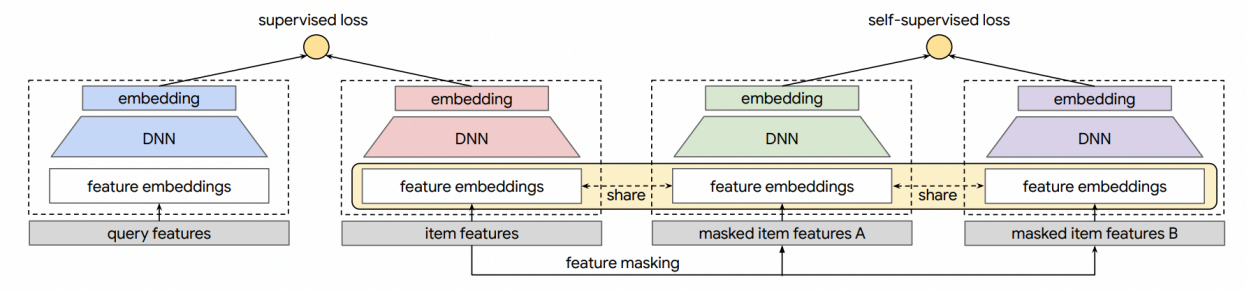
NeurIPS 2019 | 从感知跃升到认知，这是阿里在认知智能推荐领域的探索与应用

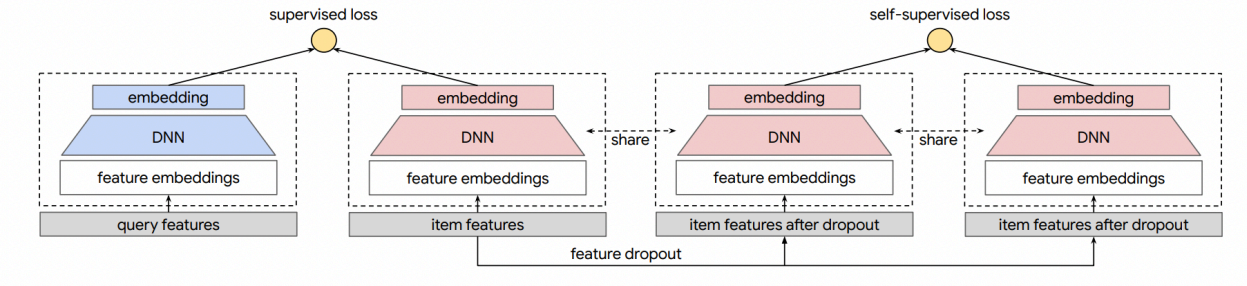
Learning Disentangled Representations for Recommendation

## 9.20 自监督

Self-supervised Learning for Large-scale Item Recommendations

v3 有两个图：https://arxiv.org/pdf/2007.12865v3.pdf




## 9.21 GNN+ 推荐

https://zhuanlan.zhihu.com/p/323302898

Graph Neural Networks in Recommender Systems: A Survey

Graph Neural Networks for Recommender Systems: Challenges, Methods, and Directions

## 9.22 DCNv2

DCN V2：Google 提出改进版 DCN，用于大规模排序系统中的特征交叉学习 (附代码)

## 9.23 item 冷启

poso

Personalized Cold Start Modules for Large-scale Recommender Systems

https://zhuanlan.zhihu.com/p/534056942

## 9.24 暴力召回 ANN 加速

https://kexue.fm/archives/9336

大致思想，CUR 分解：query 和 item 的 MxN 打分矩阵，分解成 F(Mxk1), G(k1xk2), H(k2xN) 三个矩阵

- Mxk1 矩阵：原矩阵里搞 k1 列出来，即选出 k1 个种子 item，得到 F
- k2xN 矩阵：原矩阵里搞 k2 列出来，即选出 k2 个种子 query，得到 H
- k1xk2 矩阵：即矩阵 1 和矩阵 2 求交集，比如矩阵 1 是抽的第 1,23,54 列出来，矩阵 2 是抽的第 4,80 行出来，那交集元素就是 (1,4),(1,80),(23,4),(23,80),(54,4),(54,80) 这 6 个点，构成 k1xk2 矩阵，然后算一下伪逆得到 G

建索引：+ 挑出种子 query，和所有 item 两两计算相似度，得到 H 矩阵 + 挑出种子 item，和种子 query 两两计算相似度，再算伪逆，得到 G 矩阵 + 计算 G*H，存起来

检索：＋输入的 query 和 k1 个种子 item 算一下相似度，得到 1xk1 的矩阵 q＋q 和 G*H* 相乘，就能得到 *q* 和每个 *item* 的相似度了 ＋【这步可以 *ann* 化】：*GH* 就是 *k1*N，按列来看，就是 N 个 k1 维向量，相当于 N 个 item 向量，扔到 annlib 里去就行了，而输入的 q 也是一个 k1 维向量，就可以 ann 了

## 9.25 多目标

### 9.25.1 多目标＋推荐综述

Multi-task 多任务模型在推荐算法中应用总结 1

Multi-task 多任务学习在推荐算法中应用 (2)

多任务学习在推荐算法中的应用

### 9.25.2 阿里多目标

阿里提出多目标优化全新算法框架，同时提升电商 GMV 和 CTR

### 9.25.3 Youtube 多目标——MMoE

YouTube 多目标排序系统：如何推荐接下来收看的视频

https://daiwk.github.io/posts/dl-youtube-multitask.html

## 9.26 特征工程

浅谈微视推荐系统中的特征工程

推荐系统之数据与特征工程

## 9.27 CTR 预估

### 9.27.1 传统 ctr

https://daiwk.github.io/posts/dl-traditional-ctr-models.html

### 9.27.2 lr for ctr

Simple and scalable response prediction for display advertising

Online Models for Content Optimization

### 9.27.3 gbdt for ctr

gbdt 基础知识：

https://zhuanlan.zhihu.com/p/86263786

bagging 全称叫 bootstrap aggregating，每个基学习器都会对训练集进行有放回抽样得到子训练集，比较著名的采样法为 0.632 自助法。每个基学习器基于不同子训练集进行训练，并综合所有基学习器的预测值得到最终的预测结果。bagging 常用的综合方法是投票法，票数最多的类别为预测类别。

boosting 训练过程为阶梯状，基模型的训练是有顺序的，每个基模型都会在前一个基模型学习的基础上进行学习，最终综合所有基模型的预测值产生最终的预测结果，用的比较多的综合方式为加权法。

stacking 是先用全部数据训练好基模型，然后每个基模型都对每个训练样本进行的预测，其预测值将作为训练样本的特征值，最终会得到新的训练样本，然后基于新的训练样本进行训练得到模型，然后得到最终预测结果。

bagging 和 stacking 中的基模型为强模型（偏差低，方差高），而 boosting 中的基模型为弱模型（偏差高，方差低）。

bagging 的特点：

- 整体模型的期望等于基模型的期望，这也就意味着整体模型的偏差和基模型的偏差近似。
- 整体模型的方差小于等于基模型的方差，当且仅当相关性为 1 时取等号，随着基模型数量增多，整体模型的方差减少，从而防止过拟合的能力增强，模型的准确度得到提高。

所以，bagging 中的基模型一定要为强模型，如果 bagging 使用弱模型则会导致整体模型的偏差提高，而准确度降低。

boosting 的特点：

- 整体模型的方差等于基模型的方差，如果基模型不是弱模型，其方差相对较大，这将导致整体模型的方差很大，即无法达到防止过拟合的效果。因此，boosting 框架中的基模型必须为弱模型。
- boosting 框架中采用基于贪心策略的前向加法，整体模型的期望由基模型的期望累加而成，所以随着基模型数的增多，整体模型的期望值增加，整体模型的准确度提高。

gbdt 与 Adaboost 对比

相同：

- 都是 boosting，使用弱分类器；
- 都使用前向分布算法；

不同：

- 迭代思路不同：adaboost 是通过提升错分数据点的权重来弥补模型的不足（利用错分样本），而 GBDT 是通过算梯度来弥补模型的不足（利用残差）；
- 损失函数不同：adaBoost 采用的是指数损失，GBDT 使用的是绝对损失或者 Huber 损失函数；

Learning the click-through rate for rare/new ads from similar ads

Using boosted trees for click-through rate prediction for sponsored search

Improving Ad Relevance in Sponsored Search

Stochastic Gradient Boosted Distributed Decision Trees

https://zhuanlan.zhihu.com/p/148050748

#### 9.27.4 深度学习 ctr

https://daiwk.github.io/posts/dl-dl-ctr-models.html

#### 9.27.5 ctr 特征

稠密特征加入 CTR 预估模型的方法汇总

#### 9.27.6 HugeCTR

点击率预估的训练传统上存在着几个困扰着广大开发者的问题：巨大的哈希表（Embedding Table），较少的矩阵计算，大量的数据吞吐。

HugeCTR 是首个全部解决以上问题的开源 GPU 训练框架，与现有 CPU 和混合 CPU / GPU 解决方案相比，它的速度提高了 12 倍至 44 倍。HugeCTR 是一种端到端训练解决方案，其所有计算都在 GPU 上执行，而 CPU 仅用于 I / O。GPU 哈希表支持动态缩放。它利用 MPI 进行多节点训练，以支持任意大的嵌入尺寸。它还还支持混合精度训练，在 Volta GPU 及其后续版本上可以利用 Tensor cores 进一步加速。

如何解决点击率预估？英伟达专家详解 HugeCTR 训练框架（二）

Merlin HugeCTR 分级参数服务器简介

#### 9.27.7 阿里妈妈 CTR

阿里妈妈点击率预估中的长期兴趣建模

### 9.27.8 凤巢

大规模深度学习广告系统的分布式分层 GPU 参数服务器

Distributed Hierarchical GPU Parameter Server for Massive Scale Deep Learning Ads Systems

### 9.27.9 cvr

ecpc：用户给定一个粗粒度出价，模型可以在一定的范围内调价 ocpc：完全以模型出价为准

delay feedback https://zhuanlan.zhihu.com/p/555950153

### 9.27.10 时长预估

快手 kdd 2022

Deconfounding Duration Bias in Watch-time Prediction for Video Recommendation

### 9.27.11 APG

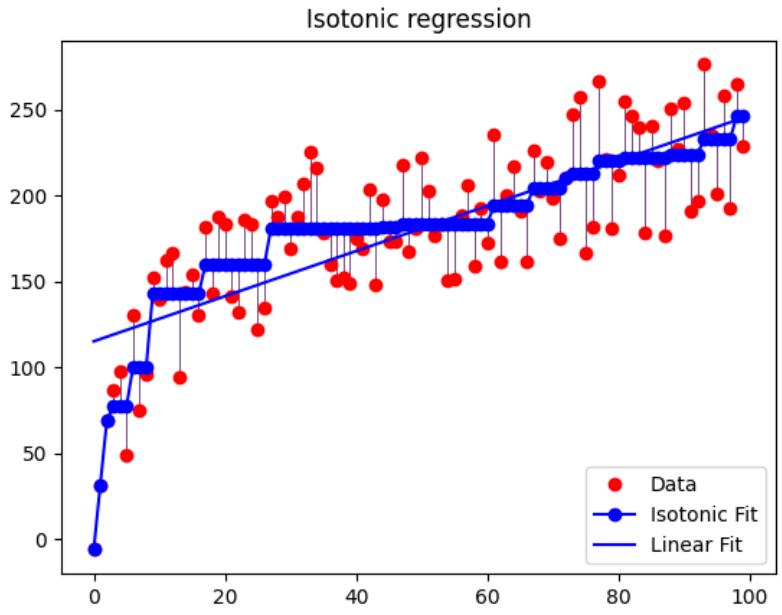
APG: 面向 CTR 预估的自适应参数生成网络

摘要：目前基于深度学习的 CTR 预估模型（即 Deep CTR Models）被广泛的应用于各个应用中。传统的 Deep CTR Models 的学习模式是相对静态的，即所有的样本共享相同的网络参数。然而，由于不同样本的特征分布不尽相同，这样一种静态方式很难刻画出不同样本的特性，从而限制了模型的表达能力，导致次优解。在本文中，我们提出了一个高效率、高效果的通用模块，称为自适应参数生成网络 (APG)。其可以基于不同的样本，动态的为 CTR 模型生成不同的模型参数。大量的实验表明，APG 能够被应用于各种 CTR 模型，并且显著的提升模型效果，同时能节省 38.7% 的时间开销和 96.6% 的存储。APG 已在阿里巴巴搜索广告系统部署上线，并获得 3% 的点击率增长和 1% 的广告收入增长。

APG: Adaptive Parameter Generation Network for Click-Through Rate Prediction

### 9.27.12 保序回归

参考https://zhuanlan.zhihu.com/p/88623159的代码，能画出下面的图




对于二分类问题，参考https://zhuanlan.zhihu.com/p/101766505

对 lr+gbdt 的负采样校准的方法

Practical Lessons from Predicting Clicks on Ads at Facebook

# 10 图神经网络

https://daiwk.github.io/posts/links-navigation-gnn.html

## 10.1 GNN 数据集

Bengio 参与、LeCun 点赞：图神经网络权威基准现已开源

https://github.com/graphdeeplearning/benchmarking-gnns

## 10.2 GNN 综述

图神经网络（Graph Neural Networks，GNN）综述

A Comprehensive Survey on Graph Neural Networks

网络图模型知识点综述

想入门图深度学习？这篇 55 页的教程帮你理清楚了脉络

A Gentle Introduction to Deep Learning for Graphs

2020 年，图机器学习将走向何方?

常见问题：QA 派 | GNN 工业应用-PinSAGE

https://keg.cs.tsinghua.edu.cn/jietang/publications/KDD23-MLG-keynote.pdf

## 10.3 GNN 理论研究

On The Equivalence Between Node Embeddings And Structural Graph Representations

在本文中，来自普渡大学计算机科学系的两位研究者提供了首个用于节点（位置）嵌入和结构化图表征的统一的理论框架，该框架结合了矩阵分解和图神经网络等方法。通过利用不变量理论（invariant theory），研究者表明，结构化表征和节点嵌入之间的关系与分布和其样本之间的关系类似。他们还证明，可以通过节点嵌入执行的任务也同样能够利用结构化表征来执行，反之亦然。此外，研究者还表明，直推学习和归纳学习的概念与节点表征和图表征无关，从而澄清了文献中的另一个困惑点。最后，研究者介绍了用于生成和使用节点嵌入的新的实践指南，从而修复了现在所使用的的标准操作流程的缺陷。

推荐：实证研究结果表明，在本文提出的理论框架加持下，节点嵌入可以成功地作为归纳学习方法使用，并且 non-GNN 节点嵌入在大多数任务上的准确度显著优于简单的图神经网络（GNN）方法。

## 10.4 图翻译

ICDM 2019 最佳论文：从图片、文本到网络结构数据翻译，一种新型的多属性图翻译模型

## 10.5 异构图 GNN

2019 年，异质图神经网络领域有哪些值得读的顶会论文?

## 10.6 HAN-GNN

Heterogeneous Graph Attention Network

https://github.com/Jhy1993/HAN

## 10.7 GTN

Graph Transformer Networks

https://github.com/seongjunyun/Graph_Transformer_Networks

## 10.8 HetGNN

Heterogeneous Graph Neural Network

https://github.com/chuxuzhang/KDD2019_HetGNN

## 10.9 HGAT

Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification

EMNLP 2019 开源论文：针对短文本分类的异质图注意力网络

短文本分类在新闻及微博等领域得到了广泛的应用。但是，目前的文本分类算法主要集中于长文本分类并且无法直接应用于短文本分类。这是由于短文本分类的两个独有挑战：

1. 数据的稀疏和歧义：短文本通常不超过 10 个词，提供的信息非常有限。经典的 Bi-LSTM+Attention 往往无法有效的捕获短文本中的语义信息。
2. 标签数量较少：传统的监督学习无法有效工作，尤其是传统深度学习算法需要大量的监督数据。

针对上述两个挑战，本文创新地将短文本建模为异质图（见 Figure 1），通过图数据的复杂交互来解决数据稀疏和歧义带来的挑战。同时，本文提出了一种异质图注意力 HGAT 来学习短文本的表示并进行分类。HGAT 是一种半监督学习算法可以更好的适用于标签数量较少的场景，如短文本的分类

## 10.10 MEIRec

Metapath-guided Heterogeneous Graph Neural Network for Intent Recommendation

## 10.11 GAS

CIKM 最佳应用论文：11 亿节点的大型图，看闲鱼如何用图卷积过滤垃圾评论

Spam Review Detection with Graph Convolutional Networks

## 10.12 AAAI2020 GNN

AAAI 2020 论文抢先看！图学习 GNN 火爆，谷歌、港科大、北邮最新力作

## 10.13 cluster-GCN

Google 图嵌入工业界最新大招，高效解决训练大规模深度图卷积神经网络问题

https://github.com/benedekrozemberczki/ClusterGCN

## 10.14 深层 GCN

从 3/4 层拓展到 56 层，如何训练超级深层的图卷积神经网络

Deep GCNs: Can GCNs Go as Deep as CNNs?

## 10.15 GNN 或者图模型的一些应用场景

## 10.16 风控关系

风控特征—关系网络特征工程入门实践

### 10.17 社区发现相关

最小熵原理：“层层递进”之社区发现与聚类

### 10.18 transformer+gnn

原来 Transformer 就是一种图神经网络，这个概念你清楚吗?

## 11 强化学习

### 11.1 RL 历史

漫画带你图解强化学习

强化学习 70 年演进：从精确动态规划到基于模型

### 11.2 RL 概述

强化学习之路——清华博士后解读 83 篇文献，万字长文总结

pg/ddpg 相关

https://daiwk.github.io/posts/rl-stepbystep-chap9.html

### 11.3 MAB 相关

### 11.4 multitask+mab

多任务学习时转角遇到 Bandit 老虎机

### 11.5 RL 基础

### 11.6 srl+drl

大白话之《Shallow Updates Deep Reinforcement Learning》

Shallow Updates Deep Reinforcement Learning

### 11.7 模仿学习

今晚，NeurIPS 2019 Spotlight 论文分享：不完备专家演示下的模仿学习

Imitation Learning from Observations by Minimizing Inverse Dynamics Disagreement

视频 | NeurIPS 2019 分享：清华大学孙富春组提出全新模仿学习理论

ppt：https://pan.baidu.com/s/1Zj59PAe4hYhDDh5zd4gWZg

### 11.8 推荐 + 强化学习

Deep Reinforcement Learning in Large Discrete Action Spaces

### 11.9 2019 强化学习论文

2019 年深度强化学习十大必读论文！DeepMind、OpenAI 等上榜

### 11.10 ICLR2020 强化学习相关

ICLR 2020 高质量强化学习论文汇总

## 11.11 HER

“事后诸葛亮”经验池：轻松解决强化学习最棘手问题之一：稀疏奖励

本文介绍了一个“事后诸葛亮”的经验池机制，简称为 HER，它可以很好地应用于稀疏奖励和二分奖励的问题中，不需要复杂的奖励函数工程设计。强化学习问题中最棘手的问题之一就是稀疏奖励。本文提出了一个新颖的技术：Hindsight Experience Replay（HER），可以从稀疏、二分的奖励问题中高效采样并进行学习，而且可以应用于所有的 Off-Policy 算法中。

Hindsight 意为”事后”，结合强化学习中序贯决策问题的特性，我们很容易就可以猜想到，“事后”要不然指的是在状态 s 下执行动作 a 之后，要不然指的就是当一个 episode 结束之后。其实，文中对常规经验池的改进也正是运用了这样的含义。

## 11.12 多智能体 RL

## 11.13 LIIR

LIIR: Learning Individual Intrinsic Reward in Multi-Agent Reinforcement Learning

nips2019 快手跟腾讯 AI Lab 和 Robotics X 合作，它希望智能体能快速学会利用自己所观测的信息来相互配合。比如说在星际争霸中，我们发现确实会产生多智能体合作的现象，模型会让一些防高血厚的单位去抗对方的输出，己方输出高的单元会躲到后方攻击。

虽然把所有的 agents 看成是一个 agent，理论上也可以学到最终的配合效果，但是效率会非常低，不具有可扩展性。我们的方法通过一种 intrinsic reward 的机制兼顾了可扩展性和效率，通过鼓励每个 agent 按照单体利益最大化的原则去学习自己的 policy，然后这种 intrinsic reward 的影响会越来越小最后快速达到学到整体最后的方案。

## 11.14 AlphaStar

Nature：闭关修炼 9 个月，AlphaStar 达成完全体，三种族齐上宗师，碾压 99.8% 活跃玩家

## 11.15 TVT

离人类更近一步！DeepMind 最新 Nature 论文：AI 会“回忆”，掌握调取记忆新姿势

Optimizing agent behavior over long time scales by transporting value

https://www.nature.com/articles/s41467-019-13073-w

https://github.com/deepmind/deepmind-research/tree/master/tvt

## 11.16 upside-down rl

超有趣！LSTM 之父团队最新力作：将强化学习“颠倒”过来

Reinforcement Learning Upside Down: Don’t Predict Rewards - Just Map Them to Actions

RL 算法要么使用价值函数预测奖励，要么使用策略搜索使其最大化。该研究提出一种替代方法：颠倒 RL(Upside-Down RL)，主要使用监督学习来解决 RL 问题。

标准 RL 预测奖励，而 UDRL 使用奖励作为任务定义的输入，以及时间范围的表示和历史数据以及可期的未来数据的其他可计算函数。

UDRL 学会将这些输入观察结果解释为命令，并根据过去 (可能是偶然的) 经历通过 SL 将它们映射为行为 (或行为概率)。UDRL 一般通过输入命令来实现高奖励或其他目标，例如：在一定时间内获得大量奖励！另一篇关于 UDRL 的首个实验的论文 (Training agents with upside-down reinforcement learning) 表明，UDRL 在某些具有挑战性的 RL 问题上可以胜过传统的 baseline 算法。

我们还提出了一种相关的简单而且通用的方法来教机器人模仿人类。首先，对人模仿机器人当前的行为进行录像，然后让机器人通过监督学习将视频 (作为输入命令) 映射到这些行为上，然后让其概括和模仿先前未知的人类行为。这种 Imitate-Imitator 的概念实际上可以解释为什么生物进化导致父母会模仿婴儿的咿呀学语。

## 11.17 游戏 +RL

## 11.18 游戏 AI 历史 (alphago 系列)

https://deepmind.com/research/case-studies/alphago-the-story-so-far

从 α 到 μ：DeepMind 棋盘游戏 AI 进化史

alphago：Mastering the game of Go with deep neural networks and tree search

alphago zero：Mastering the game of Go without Human Knowledge

alpha zero: Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm

mu zero: Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model

### 11.18.1 alphago

reward：1 胜，-1 负，中间状态 0

3 个网络：

- SL：学习棋谱的 P
- RL：自己和自己下，学习一个 P
- V：学习 s 的长期收益
- fast roolout：用一个简单的线性模型学习棋谱的 P

蒙特卡罗树搜索 + 深度学习-AlphaGo 原版论文阅读笔记

mcts：

选 Q+u 最大的 a，

首先模拟 n 次，

- N(s,a)：对于第 i 次，如果经过当前的 (s,a)，那么 +1
- Q(s,a)：对于第 i 次，如果走到叶子时经过了当前的 (s,a)，那么把 V(叶子) 加上，最后除以 N(s,a)
- V(叶子)：(1-lambda) * value network 的输出 +lambda * fastrollout 走到结束时的 reward
- u(s,a)：与 P(s,a)/(1+N(s,a)) 成正比
- P(s,a)：policy network 的输出

最开始还没 expand 时，Q 是 0，那 SL 的 P 就是 prior probabilities。P 还能起到减少搜索宽度的作用，普通点得分很低。比较难被 select 到。有趣的结论是，比较得出这里用 SL 比 RL 的要好！！模仿人类走棋的 SL 结果更适合 MCTS 搜索，因为人类选择的是 a diverse beam of promising moves。而 RL 的学的是最优的下法（whereas RL optimizes for the single best move）。所以人类在这一点暂时获胜！不过另一方面，RL 学出来的 value networks 在评估方面效果好。所以各有所长。**搜索次数 N 一多会扣分，鼓励 exploration 其他分支**。

### 11.18.2 alphago zero

模型输出 p 和 v，训练的时候通过 mcts 去选 action。loss 就是 p 的交叉熵 +v 的 rmse

### 11.18.3 alpha zero

主要是特征改了一下，使得可以适用于各种棋,loss 没变

### 11.18.4 muzero

模型加了个 r，loss 里加了个 r，

planning 需要考虑 r V P，还有次数 N

### 11.19 绝悟

不服 SOLO：腾讯绝悟 AI 击败王者荣耀顶尖职业玩家，论文入选 AAAI，未来将开源

Mastering Complex Control in MOBA Games with Deep Reinforcement Learning

以 MOBA 手游《王者荣耀》中的 1v1 游戏为例，其状态和所涉动作的数量级分别可达 10^600 和 10^18000，而围棋中相应的数字则为 10^170 和 10^360

为了实现有效且高效的训练，本文提出了一系列创新的算法策略：

- 目标注意力机制；用于帮助 AI 在 MOBA 战斗中选择目标。
- LSTM；为了学习英雄的技能释放组合，以便 AI 在序列决策中，快速输出大量伤害。
- 动作依赖关系的解耦；用于构建多标签近端策略优化（PPO）目标。
- 动作掩码；这是一种基于游戏知识的剪枝方法，为了引导强化学习过程中的探索而开发。
- dual-clip PPO；这是 PPO 算法的一种改进版本，使用它是为了确保使用大和有偏差的数据批进行训练时的收敛性。

### 11.20 RL+ 因果

华为诺亚 ICLR 2020 满分论文：基于强化学习的因果发现算法

### 11.21 RL+Active learning

Ready Policy One: World Building Through Active Learning

基于模型的强化学习（Model-Based Reinforcement Learning，MBRL）为样本高效学习提供了一个有前途的方向，通常可以实现连续控制任务（continuous control task）的 SOTA 结果。然而，许多现有的 MBRL 方法依赖于贪婪策略（greedy policy）与探索启发法的结合，甚至那些利用原则试探索奖金（exploration bonus）的方法也能够以特定方式构建双重目标。

### 11.22 ES

Evolution Strategies as a Scalable Alternative to Reinforcement Learning

在本文中，研究者介绍了 Ready Policy One（RP1），这是一种将 MBRL 视为主动学习问题的框架。研究者的目标是在尽可能少样本中改进世界模型（world model）。RP1 通过利用混合目标函数来实现这一目标，该函数在优化过程中的适应性调整至关重要，从而使算法可以权衡不同学习阶段的奖励与探索。此外，一旦拥有足够丰富的轨迹批（trajectory batch）来改进模型，研究者会引入一种原则式机制（principled mechanism）来终止样本收集。

## 12 Auto-ML

### 12.1 automl 综述

https://github.com/hibayesian/awesome-automl-papers

CVPR 2019 神经网络架构搜索进展综述

https://drsleep.github.io/NAS-at-CVPR-2019/

### 12.2 HM-NAS

ICCV Workshop 最佳论文提名：通过层级掩码实现高效神经网络架构搜索

### 12.3 FGNAS

Fine-Grained Neural Architecture Search

在本文中，研究者提出了一种优雅的细粒度神经架构搜索（fine-grained neural architecture search，FGNAS）框架，该框架允许在单层中采用多个异构运算，甚至可以使用几种不同的基础运算生成合成特征图。与其他方法相比，尽管搜索空间非常大，但 FGNAS 仍可高效运行，因为它能够通过随机梯度下降方法端对端地训练网络。此外，所提出的 FGNAS 框架允许在预定义的资源约束下根据参数数量、

FLOP 和时延来优化网络。FGNAS 已应用于资源要求很高的计算机视觉任务中的两个关键应用-大型图像分类和图像超分辨率，结果证明可以通过灵活的运算搜索和通道剪枝展示了 SOTA 性能。

### 12.4 NAT

NeurIPS 2019 | 自动优化架构，这个算法能帮工程师设计神经网络

NAT: Neural Architecture Transformer for Accurate and Compact Architectures

### 12.5 NASP

比可微架构搜索 DARTS 快 10 倍，第四范式提出优化 NAS 算法

神经架构搜索（NAS）因其比手工构建的架构更能识别出更好的架构而备受关注。近年来，可微分的搜索方法因可以在数天内获得高性能的 NAS 而成为研究热点。然而，由于超级网的建设，其仍然面临着巨大的计算成本和性能低下的问题。

在本文中，我们提出了一种基于近端迭代（NASP）的高效 NAS 方法。与以往的工作不同，NASP 将搜索过程重新定义为具有离散约束的优化问题和模型复杂度的正则化器。由于新的目标是难以解决的，我们进一步提出了一种高效的算法，由近端启发法进行优化。

通过这种方式，NASP 不仅比现有的可微分的搜索方法速度快，而且还可以找到更好的体系结构并平衡模型复杂度。最终，通过不同任务的大量实验表明，NASP 在测试精度和计算效率上均能获得更好的性能，在发现更好的模型结构的同时，速度比 DARTS 等现有技术快 10 倍以上。此外，NASP 消除了操作之间的关联性。

Efficient Neural Architecture Search via Proximal Iterations

https://github.com/xujinfan/NASP-codes

### 12.6 NASP+ 推荐系统

Efficient Neural Interaction Functions Search for Collaborative Filtering

https://github.com/quanmingyao/SIF

https://www.tuijianxitong.cn/cn/school/openclass/27

https://www.tuijianxitong.cn/cn/school/video/26

### 12.7 automl+nlp

超强大自动 NLP 工具！谷歌推出 AutoML 自然语言预训练模型

### 12.8 nni

长期盘踞热榜，微软官方 AutoML 库教你三步学会 20+ 炼金基本功

https://github.com/microsoft/nni

### 12.9 视频 NAS

比手工模型快 10~100 倍，谷歌揭秘视频 NAS 三大法宝

## 13 few-shot & meta-learning

https://daiwk.github.io/posts/ml-few-shot-learning.html

英伟达小样本换脸 AI：金毛一秒变二哈，还有在线试玩

### 13.1 meta-learning

NeurIPS 提前看 | 四篇论文，一窥元学习的最新研究进展

## 13.2 few-shot 数据集

FewRel 2.0 数据集：以近知远，以一知万，少次学习新挑战

## 13.3 incremental few-shot

多伦多大学提出注意式吸引器网络，实现渐进式少量次学习

Incremental Few-Shot Learning with Attention Attractor Networks

https://github.com/renmengye/inc-few-shot-attractor-public

## 13.4 few-shot 无监督 img2img

Few-Shot Unsupervised Image-to-Image Translation

本项目的灵感来自人类自身。人可以从少量示例中获取新对象的本质，并进行概括。本项目实现了一种无监督模式的图像到图像转换算法，在测试时仅由几个示例图像加以确定，就能用于之前未见过的新目标类。

https://github.com/NVlabs/FUNIT

## 13.5 TADAM

https://blog.csdn.net/liuglen/article/details/84193555

TADAM:Task dependent adaptive metric for improved few-shot learning

meta-learning with latent embedding optimization

nips18 的 tadam 还有这篇，基本思路都是把问题先转化成做样本 matching 的 deep metric learning 任务 并对类目信息做 task condition

## 13.6 AutoGRD

CIKM 2019 最佳论文

https://daiwk.github.io/assets/autogrd.pdf

近来，非机器学习人士也希望能够使用相关的算法进行应用。其中一个主要的挑战是，他们需要选择算法并用它来解决问题。如果能够选择正确的算法，在给定数据集、任务和评价方法的情况下可以使算法得到很好的效果。

本文中，研究者提出了一个名为 AutoGRD 的算法，这是一种新颖的元学习算法，用于算法推荐。AutoGRD 首先将数据表示为图，并将其隐式表示提取出来。提取出来的表示会被用来训练一个排序元模型，这个模型能够精确地对未见数据集提供表现最好的算法。研究者将这一算法在 250 个数据集上进行了测试，在分类和回归任务上都表现出了很高的性能，而且 AutoGRD 比现有的元学习 SOTA 模型和贝叶斯算法表现得都要好。

## 13.7 few-shot 的一些应用

论文推荐 | 基于单阶段小样本学习的艺术风格字形图片生成

## 13.8 nips 2019 few-shot

NeurIPS 2019 少样本学习研究亮点全解析

# 14 压缩与部署

## 14.1 压缩综述

深度学习助力数据压缩，一文读懂相关理论

## 14.2 layer dropout

模型压缩实践系列之——layer dropout

## 14.3 剪枝相关

2019 年的最后一个月，这里有 6 种你必须要知道的最新剪枝技术

## 14.4 slimmable networks

深度学习模型剪枝：Slimmable Networks 三部曲

## 14.5 TAS(NAS+ 剪枝)

Network Pruning via Transformable Architecture Search

https://github.com/D-X-Y/NAS-Projects

网络剪枝是深度学习的一个有趣的领域。其思路是分析神经网络的结构，并在其中找到“死角”和有用的参数。然后按照估计好的深度和宽度建立一种新架构，称为剪枝网络。然后，可以将来自原网络中的有用参数传输到新网络。这种方式对于深度卷积神经网络（CNN）特别有用，如果在嵌入式系统中进行部署，网络规模可能会变得很大且不切实际。在前一种情况下，网络剪枝可以减少超参数数量，降低 CNN 的计算成本。

本文实际上一开始就进行了大型网络的训练。然后通过传输体系结构搜索（TAS）提出了搜索小型网络的深度和宽度的建议。最后，使用知识提炼将大型网络中的知识转移到小型网络中。

## 14.6 GDP

GDP：Generalized Device Placement for Dataflow Graphs

大型神经网络的运行时间和可扩展性会受到部署设备的影响。随着神经网络架构和异构设备的复杂性增加，对于专家来说，寻找合适的部署设备尤其具有挑战性。现有的大部分自动设备部署方法是不可行的，因为部署需要很大的计算量，而且无法泛化到以前的图上。为了解决这些问题，研究者提出了一种高效的端到端方法。该方法基于一种可扩展的、在图神经网络上的序列注意力机制，并且可以迁移到新的图上。在不同的表征深度学习模型上，包括 Inception-v3、AmoebaNet、Transformer-XL 和 WaveNet，这种方法相比人工方法能够取得 16% 的提升，以及比之前的最好方法有 9.2% 的提升，在收敛速度上快了 15 倍。为了进一步减少计算消耗，研究者在一系列数据流图上预训练了一个策略网络，并使用 superposition 网络在每个单独的图上微调，在超过 50k 个节点的大型图上得到了 SOTA 性能表现，例如一个 8 层的 GNMT。

推荐：本文是谷歌大脑的一篇论文，通过图网络的方法帮助将模型部署在合适的设备上。推荐收到硬件设备限制，需要找到合适部署图的方法的读者参考。

ICCV 2019 提前看 | 三篇论文，解读神经网络压缩

## 14.7 metapruning

MetaPruning: Meta Learning for Automatic Neural Network Channel Pruning

旷视。近年来，有研究表明无论是否保存了原始网络的权值，剪枝网络都可以达到一个和原始网络相同的准确率。因此，通道剪枝的本质是逐层的通道数量，也就是网络结构。鉴于此项研究，Metapruning 决定直接保留裁剪好的通道结构，区别于剪枝的裁剪哪些通道。

本文提出来一个 Meta network，名为 PruningNet，可以生成所有候选的剪枝网络的权重，并直接在验证集上评估，有效的搜索最佳结构。

## 14.8 data-free student

Data-Free Learning of Student Networks

该篇论文是华为提出的一篇蒸馏方向的论文，其主要的创新点是提出的蒸馏过程**不需要原始训练数据的参与**。

### 14.9 样本相关性用于蒸馏

Correlation Congruence for Knowledge Distillation

这篇论文是由商汤提出的一篇蒸馏方向论文，其主要的亮点在于研究**样本之间的相关性**，利用这种相关性作为蒸馏的知识输出。

### 14.10 pu learning+ 压缩

视频 | NeurIPS 2019 分享：华为诺亚方舟提出基于少量数据的神经网络模型压缩技术

Positive-Unlabeled Compression on the Cloud

NeurIPS 2019，华为诺亚方舟。神经网络的小型化已经在 cnn 网络中取得了巨大的成功，并且能够在终端设备上例如手机、相机等进行落地应用。然而，由于隐私方面等限制，在实际进行神经网络的小型化时，可能只存在少量标记过的训练数据。如何在这种情况下压缩神经网络并取得好的结果，是急需解决的问题。本讲座首先介绍了半监督学习的正类与未标记学习（pu learning）方法，之后介绍了基于 **pu learning** 的少量数据下的**神经网络模型压缩**方法。

### 14.11 对抗训练 + 压缩

Adversarially Trained Model Compression: When Robustness Meets Efficiency

快手 nips2019，把各种压缩方法都集中到一起，并做一种联合优化，这和之前按照 Pipeline 的形式单独做压缩有很大的不同。与此同时，模型还能抵御对抗性攻击。

### 14.12 GSM

Global Sparse Momentum SGD for Pruning Very Deep Neural Networks

快手 nips2019，一般剪枝方法都需要大量调参以更好地保留模型性能，而全局稀疏动量 SGD 会端到端地学习到底哪些权重比较重要，重要的就少压缩一点，不重要的就多压缩一点。核心思想在于，我们给定一个压缩率，模型在训练中就能自己剪裁，并满足这个压缩率。

### 14.13 无监督量化

ResNet 压缩 20 倍，Facebook 提出新型无监督模型压缩量化方法

And the Bit Goes Down: Revisiting the Quantization of Neural Networks

### 14.14 Autocompress

AAAI 2020 | 滴滴 & 东北大学提出自动剪枝压缩算法框架，性能提升 120 倍

AutoCompress: An Automatic DNN Structured Pruning Framework forUltra-High Compression Rates

## 15 硬件

### 15.1 硬件综述

2 小时演讲，近 140 页 PPT，这个 NeurIPS Tutorial 真是超硬核的 AI 硬件教程

Efficient Processing of Deep Neural Network: from Algorithms to Hardware Architectures

### 15.2 TPU

TPU 的起源，Jeff Dean 综述后摩尔定律时代的 ML 硬件与算法

【Jeff Dean 独自署名论文】深度学习革命引领计算机架构与 AI 芯片未来

The Deep Learning Revolution and Its Implications for Computer Architecture and Chip Design

Kaggle 竞赛硬件如何选择？不差钱、追求速度，那就上 TPU 吧

## 15.3 pixel 4

数十亿次数学运算只消耗几毫瓦电力，谷歌开源 Pixel 4 背后的视觉模型

2019 年 11 月，谷歌发布了 MobileNetV3 和进行了 Pixel 4 Edge TPU 优化后的 MobileNetEdgeTPU 模型的源代码及检查点（checkpoint）。这些模型是最新的可感知硬件（hardware-aware）的自动机器学习（AutoML）技术与一些新型架构设计的结晶。

http://ai.googleblog.com/2019/11/introducing-next-generation-on-device.html

MobileNetV3 和 MobileNetEdgeTPU 的代码，以及用于 ImageNet 分类的浮点和量化的检查点，都可以在 MobileNet 的 github 主页上找到：https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet。

在 Tensorflow 目标检测 API 中提供了 MobileNetV3 和 MobileNetEdgeTPU 目标检测的开源实现：https://github.com/tensorflow/models/tree/master/research/object_detection。

MobileNetV3 语义分割的 TensorFlow 版开源实现可以在 DeepLab 中找到：https://github.com/tensorflow/models/tree/master/research/deeplab。

## 15.4 MNNKit

阿里开源 MNNKit：基于 MNN 的移动端深度学习 SDK，支持安卓和 iOS

## 15.5 deepshift

把 CNN 里的乘法全部去掉会怎样？华为提出移动端部署神经网络新方法

DeepShift: Towards Multiplication-Less Neural Networks

https://github.com/mostafaelhoushi/DeepShift

## 15.6 传感器相关

基于传感器的人类行为识别 DL 方法难在哪？这篇综述列了 11 项挑战

# 16 架构

## 16.1 分布式机器学习

MLSys 提前看 | 机器学习的分布式优化方法

机器学习系统 (MLsys) 综述：分布式、模型压缩与框架设计

## 16.2 JAX

DeepMind 发布神经网络、强化学习库，网友：推动 JAX 发展

Jax 生态再添新库：DeepMind 开源 Haiku、RLax

Haiku：https://github.com/deepmind/haiku

RLax：https://github.com/deepmind/rlax

## 16.3 trax

谷歌大脑开源 Trax 代码库，你的深度学习进阶路径

https://github.com/google/trax

## 16.4 spark

Spark 通信篇 | spark RPC 模块篇

PySpark 源码解析，教你用 Python 调用高效 Scala 接口，搞定大规模数据分析

## 16.5 kaggle 相关工具

Kaggle 最流行 NLP 方法演化史，从词袋到 Transformer

## 16.6 blink

伯克利与微软联合发布：任意网络结构下的最优 GPU 通信库 Blink

Blink: Fast and Generic Collectives for Distributed ML

## 16.7 cortex

模型秒变 API 只需一行代码，支持 TensorFlow 等框架

https://github.com/cortexlabs/cortex

https://daiwk.github.io/posts/platform-cortex.html

## 16.8 optuna

召唤超参调优开源新神器：集 XGBoost、TensorFlow、PyTorch、MXNet 等十大模块于一身

https://github.com/optuna/optuna

## 16.9 DALI

英伟达 DALI 加速技巧：让数据预处理速度比原生 PyTorch 快 4 倍

DALI 和 TensorFlow 自带的 DataLoader 类似，是一个专门用于加速数据预处理过程的库。英伟达数据加载库 DALI 是一个便捷式开源库，用于图像或视频的解码及增强，从而加速深度学习应用。通过并行训练和预处理过程，减少了延迟及训练时间，并为当下流行的深度学习框架中的内置数据加载器及数据迭代器提供了一个嵌入式替代器，便于集成或重定向至不同框架。

## 16.10 tf

## 16.11 tf-lite

借助 TFLite GPU Delegate 的实时推理扫描书籍

## 16.12 tensorboard.dev

可视化 ML 实验数据：谷歌推出免费托管服务，TensorBoard.dev 可在线一键共享

## 16.13 pytorch

### 16.13.1 torch 里的 categorical 分布 (类别分布)

https://blog.csdn.net/qq_37388085/article/details/127251550

https://zhuanlan.zhihu.com/p/59550457

### 16.13.2 pytorch 设计思路

NeurIPS 顶会接收，PyTorch 官方论文首次曝光完整设计思路

PyTorch: An Imperative Style, High-Performance Deep Learning Library

### 16.13.3 pytorch 分布式训练

了解 Pytorch 分布式训练，这一篇足够了!

### 16.13.4 Texar-PyTorch

Texar-PyTorch：在 PyTorch 中集成 TensorFlow 的最佳特性

https://github.com/asyml/texar

## 16.14 Streamlit

从 Python 代码到 APP，你只需要一个小工具：GitHub 已超 3000 星

## 16.15 paddle

提速 1000 倍，预测延迟少于 1ms，百度飞桨发布基于 ERNIE 的语义理解开发套件

20+ 移动端硬件，Int8 极速推理，端侧推理引擎 Paddle Lite 2.0 正式发布

AI 产业化应用落地不用愁，这里有份国产最大框架上手完整解析

强化学习、联邦学习、图神经网络，飞桨全新工具组件详解

YOLOv3 比原作高 10 个点，飞桨更新至 73 个视觉算法、203 个预训练模型

## 16.16 TensorRT

手把手教你采用基于 TensorRT 的 BERT 模型实现实时自然语言理解

https://developer.nvidia.com/deep-learning-software

7 倍 AI 算力芯片，TensorRT 重大更新，英伟达 GTC 新品全介绍

## 16.17 开源 gnn 平台

https://daiwk.github.io/posts/platform-gnn-frameworks.html

## 16.18 dgl

Amazon 图神经网络库 DGL 零基础上手指南

## 16.19 plato

十亿节点大规模图计算降至「分钟」级，腾讯开源图计算框架柏拉图

## 16.20 angel(java)

https://github.com/Angel-ML/angel

腾讯大数据将开源高性能计算平台 Angel，机器之心专访开发团队

跻身世界顶级 AI 项目：腾讯机器学习平台 Angel 从 LF AI 基金会毕业

## 16.21 bytegraph

字节跳动自研万亿级图数据库 & 图计算实践

## 16.22 tensorlayer

几行代码轻松实现，Tensorlayer 2.0 推出深度强化学习基准库

## 16.23 MAX

https://github.com/IBM/MAX-Framework

https://github.com/IBM/MAX-Skeleton

cikm 2019 最佳 Demo 奖

Model Asset eXchange: Path to Ubiquitous Deep Learning Deployment

IBM 的研究者提出了一种名为 Model Asset Exchange（MAE）的系统，使得开发人员可以方便地利用当前最新的 DL 模型。无论底层的 DL 编程框架是什么，该模型都能提供一个开源的 Python 库（MAX 框架），该库封装 DL 模型，并使用标准化的 RESTful API 统一编程接口。这些 RESTful API 使得开发者能够在推理任务中利用封装的 DL 模型，无需完全理解不同的 DL 编程框架。利用 MAX，研究者封装并开源了来自不同研究领域的 30 多个 SOTA DL 模型，包括计算机视觉、自然语言处理和信号处理等。

## 16.24 CogDL

集成图网络模型实现、基准测试，清华推出图表示学习工具包

http://keg.cs.tsinghua.edu.cn/cogdl/index.html

与其他图表示学习工具包相比，CogDL 具有以下特点：

- 稀疏性：在具有数千万节点的大规模网络上实现快速网络嵌入；
- 任意性：能够处理属性化、多路和异构等不同图结构的网络；
- 并行处理：在多个 GPU 上实现不同种子和模型的并行训练并自动输出结果表格；
- 可扩展性：轻松添加新的数据集、模型和任务并在所有现有的模型/数据集上测试。

pytorch 版本：

https://github.com/THUDM/cogdl/

tf 版本：

https://github.com/imsheridan/CogDL-TensorFlow

## 16.25 auto-ml 架构

比谷歌 AutoML 快 110 倍，全流程自动机器学习平台应该是这样的

## 16.26 megengine

国产深度学习框架 MegEngine，旷视打造，三月底即将开源

## 16.27 异构硬件加速

广告深度学习计算：异构硬件加速实践

## 16.28 GPU 基础知识

CPU 与 GPU 之间通过 PCIe 进行连接

# 17 课程资源

## 17.1 分布式系统课程

MIT 经典课程“分布式系统”视频版上线！网友：终于来了非偷拍清晰版本

## 17.2 微软 ml 基础课程 (win10 版)

GitHub 6600 星，面向中国人：微软 AI 教育与学习共建社区 2.0 登场!

https://github.com/microsoft/ai-edu

## 17.3 无监督课程

14 周无监督学习课程，UC 伯克利出品，含课件、视频

## 17.4 tf2.0 课程

全新版本，李沐《动手学深度学习》TF2.0 版本来了

https://github.com/TrickyGo/Dive-into-DL-TensorFlow2.0

TensorFlow 2.0 深度学习开源书:

TensorFlow 2.0 中文开源书项目：日赞 700，登上 GitHub 热榜

https://github.com/dragen1860/Deep-Learning-with-TensorFlow-book

https://daiwk.github.io/posts/platform-tensorflow-2.0.html

TensorFlow 2.0 常用模块 1：Checkpoint

TensorFlow 2.0 常用模块 2：TensorBoard

TensorFlow 2.0 常用模块 3：tf.data

TensorFlow 2.0 常用模块 4：TFRecord

TensorFlow 2.0 常用模块 5：@tf.function

TensorFlow 2.0 中的 tf.keras 和 Keras 有何区别？为什么以后一定要用 tf.keras?

上线俩月，TensorFlow 2.0 被吐槽太难用，网友：看看人家 PyTorch

快速掌握 TensorFlow 中张量运算的广播机制

## 17.5 tf 2.0 分布式课程

TensorFlow 2.0 分布式训练

## 17.6 深度学习 + 强化学习课程

Bengio、Sutton 的深度学习 & 强化学习暑期班又来了，2019 视频已放出

https://dlrlsummerschool.ca/

https://www.youtube.com/playlist?list=PLKlhhkvvU8-aXmPQZNYG_e-2nTd0tJE8v

### 17.7 统计学习方法课程

学它！李航《统计学习方法》课件，清华大学深圳研究院教授制作

https://pan.baidu.com/s/1HUw0MeBD-1LP-r441oykhw

GitHub 趋势榜首：李航《统计学习方法》Python 代码实现

### 17.8 Colab 相关课程

帮初学者快速上手机器学习，这有一份 Colab 资源大全

https://www.google-colab.com/

https://github.com/firmai/awesome-google-colab

https://github.com/toxtli/awesome-machine-learning-jupyter-notebooks-for-colab

### 17.9 李宏毅机器学习课程

你离开学只差这个视频：李宏毅机器学习 2020 版正式开放上线

### 17.10 nlp+ 社交课程

社科 NLP 课程来了：斯坦福开年公开课主讲 NLP 和社交网络应用

### 17.11 图机器学习课程

【斯坦福大牛 Jure Leskovec 新课】CS224W：图机器学习，附课程 PPT 下载

http://web.stanford.edu/class/cs224w/

ppt 也在这个网页上

### 17.12 deeplearning.ai 课程

吴恩达 deeplearning.ai 新课上线：TensorFlow 移动和 web 端机器学习

### 17.13 多任务与元学习课程

斯坦福 CS330 2019 秋季课程视频全新上线，专注多任务与元学习

## 18 一些网址和应用

### 18.1 一些笔记

超干货！一位博士生 80 篇机器学习相关论文及笔记下载

机器学习研究者的养成指南，吴恩达建议这么读论文

### 18.2 各类数据集

https://datasetsearch.research.google.com/

### 18.3 数据集搜索

谷歌数据集搜索正式版出炉：全面升级，覆盖 2500 万数据集

https://www.datasetlist.com/

## 18.4 烂番茄影评数据集

https://github.com/nicolas-gervais/6-607-Algorithms-for-Big-Data-Analysis/blob/master/scraping%20all%20critic%20reviews%20from%20rotten%20tomatoes

https://drive.google.com/file/d/1N8WCMci_jpDHwCVgSED-B9yts-q9_Bb5/view

## 18.5 numpy 实现 ml 算法

https://github.com/ddbourgin/numpy-ml

## 18.6 pytorch 实现 rl

作者列出了 17 种深度强化学习算法的 PyTorch 实现。包括 DQN，DQN-HER，DoubleDQN，REINFORCE，DDPG，DDPG-HER，PPO，SAC，离散 SAC，A3C，A2C 等。

https://github.com/p-christ/Deep-Reinforcement-Learning-Algorithms-with-PyTorch

## 18.7 一些有趣应用

一键抠图，毛发毕现：这个 GitHub 项目助你快速 PS

https://github.com/pymatting/pymatting

“狗屁不通文章生成器”登顶 GitHub 热榜，分分钟写出万字形式主义大作

实时可视化 Debug：VS Code 开源新工具，一键解析代码结构

# 19 基础知识

## 19.1 C++

百度 C++ 工程师的那些极限优化（内存篇）

https://daiwk.github.io/posts/knowledge-c++-useful-features.html

## 19.2 python

https://daiwk.github.io/posts/knowledge-python.html

## 19.3 core 相关

https://daiwk.github.io/posts/knowledge-stack-heap-core.html

tf.keras 中的预训练修改模型中 layer 的名字，解决“The name is used 2 times ···all layer names should be unique.”问题。

https://nrasadi.medium.com/change-model-layer-name-in-tensorflow-keras-58771dd6bf1b

```
import tensorflow as tf

def add_prefix(model, prefix: str, custom_objects=None):
    '''Adds a prefix to layers and model name while keeping the pre-trained weights
    Arguments:
        model: a tf.keras model
        prefix: a string that would be added to before each layer name
        custom_objects: if your model consists of custom layers you shoud add them pass them as a dicti
            For more information read the following:
            https://keras.io/guides/serialization_and_saving/#custom-objects
    Returns:
```

```
    new_model: a tf.keras model having same weights as the input model.
'''

config = model.get_config()
old_to_new = {}
new_to_old = {}

for layer in config['layers']:
    new_name = prefix + layer['name']
    old_to_new[layer['name']], new_to_old[new_name] = new_name, layer['name']
    layer['name'] = new_name
    layer['config']['name'] = new_name

    if len(layer['inbound_nodes']) > 0:
        for in_node in layer['inbound_nodes'][0]:
            in_node[0] = old_to_new[in_node[0]]

for input_layer in config['input_layers']:
    input_layer[0] = old_to_new[input_layer[0]]

for output_layer in config['output_layers']:
    output_layer[0] = old_to_new[output_layer[0]]

config['name'] = prefix + config['name']
new_model = tf.keras.Model().from_config(config, custom_objects)

for layer in new_model.layers:
    layer.set_weights(model.get_layer(new_to_old[layer.name]).get_weights())

return new_model
```

# 20 其他领域

## 20.1 信息安全

破解神经网络、攻击 GPU，AI 黑客教程来了，已登 GitHub 热榜

https://github.com/Kayzaks/HackingNeuralNetworks

NeurIPS 2019 论文分享 | 微众银行：重新思考 DNN 所有权验证，以数字护照抵御模糊攻击

密码学重大里程碑！科学家暴力破解迄今最长 RSA 密钥，功劳却不在摩尔定律

阿里的 AI 安全武功秘籍：迁移 + 元学习开路，小样本数据能用跨模态

## 20.2 量子计算

https://daiwk.github.io/posts/dl-quantum-computing.html

量子导航即将上路：实时更新，全局优化，不仅更快还能解决拥堵

对标谷歌、IBM、微软，亚马逊正式推出量子计算云服务 Braket

## 20.3 机器人

四大视角、万字长文，欧盟 MuMMER 项目之商场服务机器人深入解读

## 20.4 落地的一些思考

AI 模型走下高科技神坛、走进大规模量产、深入渗透产业界 | 百度研究院 2020 十大预测

分析了自家 150 个 ML 模型之后，这家全球最大的旅行网站得出了 6 条经验教训

你的算法耗尽全球 GPU 算力都实现不了，DeepMind 阿尔法系列被华为怒怼，曾登 Nature 子刊

不要只关注算法与模型，这里有份产品级深度学习开发指南

一日千星的「机器学习系统设计指南」，这个英伟达小姐姐的项目火了

https://github.com/chiphuyen/machine-learning-systems-design/blob/master/build/build1/consolidated.pdf

https://github.com/chiphuyen/machine-learning-systems-design

AI 落地的 2019：注定残酷，真实终会浮出水面

Nature 发表 Google 新型 AI 系统！乳腺癌筛查完胜人类专家

用户增长怎么做？UG 涉及哪些技术领域

「拥抱产业互联网」一年后，腾讯首次完整披露 20 年技术演进之路

阿里达摩院发布 2020 十大科技趋势！量子计算、类脑计算系统崛起

从工具选择到团队沟通，看 ML 工程师一步步打造生产级机器学习

疫情期间如何让快递送得更快？菜鸟网络 AAAI 论文用深度学习驱动 MIP 求解