Project 03 – Applications

Arthur J. Redfern
arthur.redfern@utdallas.edu
Nov 11, 2020

1 Logistics

- Assigned Nov 11, 2020 and due Dec 02, 2020
- This is an individual project, no help from others is allowed
- You cannot choose a project that is similar to a project you've done previously (in another class, for fun, for work, …); if the project I select for you falls into this category you need to alert me immediately and a different project will be selected
- The use of any and all online resources is allowed
- All text should be written by you – do not simply copy text from others

2 Goals

- This class covered a lot, from fundamental math to network design, training and implementation to a variety of applications; you have a strong basis to work from
- In this project you will take a deep dive into a topic that we didn't cover in detail in this class, demonstrating your ability to learn and summarize new topics

3 Project

Rank the topics below based on your interest and immediately send me an email with your 1 (most interested) – N (least interested) list. I'll reply back with the project that I've selected for you. While this is an individual project, I'm limiting the number of people that can select the same topic.

If you have an appropriate topic that you're very interested in that's not on this list, there's a small possibility I'll allow it. Send me an email with the topic, key references and your justification along with the above 1 – N list for backup and again, I'll let you know the project I've selected for you.

After receiving your project selection confirmation from me create a ≤ 20 slide tutorial that includes:

- Title slide
- Motivation slide
- Brief summary of previous work / background slide(s)
- Topic summary covering key ideas slides
- References (not included in the ≤ 20 slide limit)

Limit the number of equations to just the key ones and maximize the number of figures and plots used to illustrate the key ideas. Text should come from you, not be copied from others. If you copy a figure from somewhere else, provide a reference.

I'm not looking for a disjoint regurgitation of 5 separate papers. I'm looking for someone who understands an area and who is able to create a coherent summary of the area built from understanding and shaping multiple key papers / ideas into a clear presentation / story arc. An interesting story takes the reader on a journey that is both exciting and logically flowing. Think about the story your slides tell.

The resulting slides should enable a person who is generally skilled in the area of xNNs (e.g., 1 of your classmates or me) to learn the basics of a new topic. I like slides that a person can read offline and understand and also work in the context of an oral presentation (even though we're not going to do oral presentations). Make your slides with this in mind as I'm the 1 doing the grading.

The following are the 11 potential project topics to be ranked by you 1 – N in your email to me. Emails that do not contain a ranking of all N will be ignored. Key references to start with are provided for all cases, but feel free to expand your search as appropriate. You don't need to include material from all the references if it doesn't fit into the story you tell.

3.1 Topic 1: Design / Conditional Computation And Dynamic Networks

WeightNet: revisiting the design space of weight networks
https://arxiv.org/abs/2007.11823

DyNet: dynamic convolution for accelerating convolutional neural networks
https://arxiv.org/abs/2004.10694

Dynamic region-aware convolution
https://arxiv.org/abs/2003.12243

Conditional convolutions for instance segmentation
https://arxiv.org/abs/2003.05664

CondConv: conditionally parameterized convolutions for efficient inference

https://arxiv.org/abs/1904.04971

Squeeze-and-excitation networks
https://arxiv.org/abs/1709.01507

3.2 Topic 2: Design / Graph Neural Networks

Deep graph library: towards efficient and scalable deep learning on graphs
https://arxiv.org/abs/1909.01315
https://github.com/dmlc/dgl

Fast graph representation learning with PyTorch geometric
https://arxiv.org/abs/1903.02428
https://github.com/rusty1s/pytorch_geometric

A comprehensive survey on graph neural networks
https://arxiv.org/abs/1901.00596

Graph neural networks: a review of methods and applications
https://arxiv.org/abs/1812.08434

How powerful are graph neural networks?
https://arxiv.org/abs/1810.00826

Stanford CS224W: machine learning with graphs
See the slides from lectures 8, 9, 10, 18 and 19
http://web.stanford.edu/class/cs224w/
https://colab.research.google.com/drive/1DIQm9rOx2mT1bZETEeVUThxcrP1RKqAn

Graph deep learning
https://towardsdatascience.com/graph-deep-learning/home

The graph neural network model
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1015.7227&rep=rep1&type=pdf

An introduction to graph neural networks: models and applications
https://www.youtube.com/watch?v=zCEYiCxrL_0

3.3 Topic 3: Training / Modern ImageNet Training Optimizations

Fixing the train-test resolution discrepancy: FixEfficientNet
https://arxiv.org/abs/2003.08237

Fixing the train-test resolution discrepancy
https://arxiv.org/abs/1906.06423

Big transfer (BiT): general visual representation learning
https://arxiv.org/abs/1912.11370

Adversarial examples improve image recognition
https://arxiv.org/abs/1911.09665

Self-training with noisy student improves ImageNet classification
https://arxiv.org/abs/1911.04252

Billion-scale semi-supervised learning for image classification
https://arxiv.org/abs/1905.00546

Bag of tricks for image classification with convolutional neural networks
https://arxiv.org/abs/1812.01187

AutoAugment: learning augmentation policies from data
https://arxiv.org/abs/1805.09501

Mixup: beyond empirical risk minimization
https://arxiv.org/abs/1710.09412

3.4 Topic 4: Implementation / Binary Neural Networks

Training binary neural networks with real-to-binary convolutions
https://arxiv.org/abs/2003.11535

ReActNet: towards precise binary neural network with generalized activation functions
https://arxiv.org/abs/2003.03488

MeliusNet: can binary neural networks achieve MobileNet-level accuracy?
https://arxiv.org/abs/2001.05936

TentacleNet: a pseudo-ensemble template for accurate binary convolutional neural networks
https://arxiv.org/abs/1912.10103

XNOR-Net++: improved binary neural networks
https://arxiv.org/abs/1909.13863

Structured binary neural networks for image recognition
https://arxiv.org/abs/1909.09934

MoBiNet: a mobile binary network for image classification
https://arxiv.org/abs/1907.12629

Structured binary neural networks for accurate image classification and semantic segmentation
https://arxiv.org/abs/1811.10413

Bi-Real Net: enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm
https://arxiv.org/abs/1808.00278

Towards accurate binary convolutional neural network
https://arxiv.org/abs/1711.11294

XNOR-Net: ImageNet classification using binary convolutional neural networks
https://arxiv.org/abs/1603.05279

3.5 Topic 5: Implementation / Graph Compilers

The deep learning compiler: a comprehensive survey
https://arxiv.org/abs/2002.03794

TVM: an automated end-to-end optimizing compiler for deep learning
https://arxiv.org/abs/1802.04799

Apache TVM (incubating) an end to end deep learning compiler stack for CPUs, GPUs and accelerators
https://tvm.apache.org

TVM documentation
https://tvm.apache.org/docs/

2019 TVM and deep learning compilation conference: morning keynote & session 1
https://www.youtube.com/watch?v=npqO0hVXZwU

3.6 Topic 6: Vision / Anchor Box Free Detection And Segmentation

SOLOv2: dynamic, faster and stronger
https://arxiv.org/abs/2003.10152

SOLO: segmenting objects by locations
https://arxiv.org/abs/1912.04488
https://github.com/aim-uofa/AdelaiDet/

Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection
https://arxiv.org/abs/1912.02424
https://github.com/sfzhang15/ATSS

CenterMask : real-time anchor-free instance segmentation
https://arxiv.org/abs/1911.06667
https://github.com/youngwanLEE/CenterMask

NAS-FCOS: fast neural architecture search for object detection
https://arxiv.org/abs/1906.04423

Objects as points
https://arxiv.org/abs/1904.07850
https://github.com/xingyizhou/CenterNet

FCOS: fully convolutional one-stage object detection
https://arxiv.org/abs/1904.01355
https://github.com/tianzhi0549/FCOS

3.7 Topic 7: Vision / Point Cloud Processing

Deep learning for 3D point clouds: a survey
https://arxiv.org/abs/1912.12033

A convolutional decoder for point clouds using adaptive instance normalization
https://arxiv.org/abs/1906.11478

Learning object bounding boxes for 3D instance segmentation on point clouds
https://arxiv.org/abs/1906.01140

FlowNet3D: learning scene flow in 3d point clouds
https://arxiv.org/abs/1806.01411

VoxelNet: end-to-end learning for point cloud based 3D object detection
https://arxiv.org/abs/1711.06396

PointNet++: deep hierarchical feature learning on point sets in a metric space
https://arxiv.org/abs/1706.02413

PointNet: deep learning on point sets for 3D classification and segmentation
https://arxiv.org/abs/1612.00593

3.8 Topic 8: Vision / Transformers

An image is worth 16x16 words: transformers for image recognition at scale
https://arxiv.org/abs/2010.11929
https://github.com/google-research/vision_transformer
https://github.com/lucidrains/vit-pytorch
https://www.youtube.com/watch?v=TrdevFK_am4

End-to-end object detection with transformers
https://arxiv.org/abs/2005.12872

Exploring self-attention for image recognition
https://arxiv.org/abs/2004.13621

Adaptive attention span in computer vision
https://arxiv.org/abs/2004.08708

On the relationship between self-attention and convolutional layers
https://arxiv.org/abs/1911.03584

Stand-alone self-attention in vision models
https://arxiv.org/abs/1906.05909

Image transformer
https://arxiv.org/abs/1802.05751

3.9 Topic 9: Language / Large Scale Language Models

Hugging face transformers
https://github.com/huggingface/transformers
https://huggingface.co/transformers/

Linformer: self-attention with linear complexity
https://arxiv.org/abs/2006.04768

Reformer: the efficient transformer
https://arxiv.org/abs/2001.04451

BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension
https://arxiv.org/pdf/1910.13461.pdf

Exploring the limits of transfer learning with a unified text-to-text transformer
https://arxiv.org/abs/1910.10683

DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter
https://arxiv.org/abs/1910.01108

ALBERT: a lite BERT for self-supervised learning of language representations
https://arxiv.org/abs/1909.11942

RoBERTa: a robustly optimized BERT pretraining approach
https://arxiv.org/abs/1907.11692

XLNet: generalized autoregressive pretraining for language understanding
https://arxiv.org/abs/1906.08237

Transformer-XL: attentive language models beyond a fixed-length context
https://arxiv.org/abs/1901.02860

BERT: pre-training of deep bidirectional transformers for language understanding
https://arxiv.org/abs/1810.04805

3.10 Topic 10: Speech / Text To Speech

ESPnet-TTS: unified, reproducible, and integratable open source end-to-end text-to-speech toolkit
https://arxiv.org/abs/1910.10909
https://github.com/espnet/espnet

The LJ speech dataset
https://keithito.com/LJ-Speech-Dataset/

LibriTTS: a corpus derived from LibriSpeech for text-to-speech
https://arxiv.org/abs/1904.02882
http://www.openslr.org/60/

FastSpeech 2: fast and high-quality end-to-end text to speech
https://arxiv.org/abs/2006.04558

FastSpeech: fast, robust and controllable text to speech
https://arxiv.org/abs/1905.09263

Neural speech synthesis with transformer network
https://arxiv.org/abs/1809.08895

Transfer learning from speaker verification to multispeaker text-to-speech synthesis
https://arxiv.org/abs/1806.04558

Style tokens: unsupervised style modeling, control and transfer in end-to-end speech synthesis
https://arxiv.org/abs/1803.09017

Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions
https://arxiv.org/abs/1712.05884

Tacotron: towards end-to-end speech synthesis
https://arxiv.org/abs/1703.10135

WaveNet: a generative model for raw audio
https://arxiv.org/abs/1609.03499

3.11 Topic 11: Games / StarCraft II And Or Dota2

3.11.1 StarCraft II

AlphaStar: Grandmaster level in StarCraft II using multi-agent reinforcement learning
https://deepmind.com/blog/article/AlphaStar-Grandmaster-level-in-StarCraft-II-using-multi-agent-reinforcement-learning
https://www.nature.com/articles/s41586-019-1724-z.epdf?author_access_token=lZH3nqPYtWJXfDA10W0CNNRgN0jAjWel9jnR3ZoTv0PSZcPzJFGNAZhOlk4deBCKzKm70KfinloafEF1bCCXL6IlHHgKaDkaTkBcTEv7aT-wqDoG1VeO9-wO3GEoAMF9bAOt7mJ0RWQnRVMbyfgH9A%3D%3D

AlphaStar: Mastering the Real-Time Strategy Game StarCraft II
https://deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii

DeepMind and Blizzard open StarCraft II as an AI research environment
https://deepmind.com/blog/announcements/deepmind-and-blizzard-open-starcraft-ii-ai-research-environment

PySC2 - StarCraft II Learning Environment
https://github.com/deepmind/pysc2

AlphaStar: An Evolutionary Computation Perspective
https://arxiv.org/abs/1902.01724

StarCraft II: A New Challenge for Reinforcement Learning
https://arxiv.org/abs/1708.04782

3.11.2 Dota2

OpenAI five
https://openai.com/blog/openai-five/

Automatic player identification in Dota 2
https://arxiv.org/abs/2008.12401

Long-term planning and situational awareness in OpenAI five
https://arxiv.org/abs/1912.06721

Neural network surgery with sets
https://arxiv.org/abs/1912.06719

Dota 2 with large scale deep reinforcement learning
https://arxiv.org/abs/1912.06680

Time to die: death prediction in Dota 2 using deep learning
https://arxiv.org/abs/1906.03939

4 What To Turn In Via eLearning

- Upload individual files, not a zip file
- Don't write any comments in eLearning
- A pdf file of your slides
- Any optional material you'd like to include (e.g., if you do some work in PyTorch related to the project and want to share the code / results); the more impressed I am by the optional material the more additional points you can earn