Course 3 Optional References

**Machine Learning Modeling Pipelines in Production**

This is a compilation of resources including URLs and papers appearing in lecture videos. If you wish to dive more deeply into the topics covered this week, feel free to check out these optional references.

**Overall resources:**

Towards ML Engineering - History of TFX:

<https://arxiv.org/abs/2010.02013>

Challenges in Deploying ML:

<https://arxiv.org/abs/2011.09926>

**Week 1: Neural Architecture Search**

Neural Architecture Search:

<https://arxiv.org/pdf/1808.05377.pdf>

Bayesian Optimization:

<https://distill.pub/2020/bayesian-optimization/>

Neural Architecture Search with Reinforcement Learning:

<https://arxiv.org/pdf/1611.01578.pdf>

Progressive Neural Architecture Search:

<https://arxiv.org/pdf/1712.00559.pdf>

Network Morphism:

<https://arxiv.org/abs/1603.01670>

Amazon SageMaker Autopilot

<https://aws.amazon.com/sagemaker/autopilot>

Microsoft Azure Automated Machine Learning

<https://azure.microsoft.com/en-in/services/machine-learning/automatedml/>

Google Cloud AutoML

<https://cloud.google.com/automl>

**Week 2: Model Resource Management Techniques**

High dimensional spaces visualization:

<https://colab.research.google.com/drive/1GTBYAcMsiKDDQeDpyOIi_DGuPVleJAf0?usp=sharing>

Word embeddings:

<https://heartbeat.fritz.ai/coreml-with-glove-word-embedding-and-recursive-neural-network-part-2-d72c1a66b028>

Curse of dimensionality:

<https://builtin.com/data-science/curse-dimensionality>

<https://www.visiondummy.com/2014/04/curse-dimensionality-affect-classification/>

Sparsity:

<https://www.kdd.org/exploration_files/parsons.pdf>

Feature engineering:

<https://quantdare.com/what-is-the-difference-between-feature-extraction-and-feature-selection/>

<https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/>

PCA:

<https://scikit-learn.org/stable/modules/decomposition.html>

<https://www.coursera.org/lecture/machine-learning/principal-component-analysis-problem-formulation-GBFTt>

<https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues/140579#140579>

<https://elitedatascience.com/dimensionality-reduction-algorithms>

ICA:

<https://scikit-learn.org/stable/modules/decomposition.html>

<https://scikit-learn.org/stable/auto_examples/decomposition/plot_ica_vs_pca.html>

NMF:

<https://scikit-learn.org/stable/modules/decomposition.html#non-negative-matrix-factorization-nmf-or-nnmf>

Mobile model deployment:

<https://developers.google.com/ml-kit>

<https://www.tensorflow.org/lite>

Quantization:

<https://www.qualcomm.com/news/onq/2019/03/12/heres-why-quantization-matters-ai>

<https://petewarden.com/2016/05/03/how-to-quantize-neural-networks-with-tensorflow/>

<https://arxiv.org/abs/1712.05877>

<https://blog.tensorflow.org/2020/04/quantization-aware-training-with-tensorflow-model-optimization-toolkit.html>

<https://www.tensorflow.org/lite/performance/best_practices>

Post-training quantization:

<https://medium.com/tensorflow/introducing-the-model-optimization-toolkit-for-tensorflow-254aca1ba0a3>

Quantization aware training:

<https://blog.tensorflow.org/2020/04/quantization-aware-training-with-tensorflow-model-optimization-toolkit.html>

Pruning:

<https://blog.tensorflow.org/2019/05/tf-model-optimization-toolkit-pruning-API.html>

<http://yann.lecun.com/exdb/publis/pdf/lecun-90b.pdf>

<https://towardsdatascience.com/can-you-remove-99-of-a-neural-network-without-losing-accuracy-915b1fab873b>

<https://arxiv.org/abs/1803.03635>

<https://numenta.com/blog/2019/08/30/case-for-sparsity-in-neural-networks-part-1-pruning>

<https://www.tensorflow.org/model_optimization/guide/pruning>

**Week 3: High Performance Modeling**

Distribution strategies:

<https://www.tensorflow.org/guide/distributed_training>

Changes in data parallelism:

<https://arxiv.org/abs/1806.03377>

Pipeline parallelism:

<https://ai.googleblog.com/2019/03/introducing-gpipe-open-source-library.html>

GPipe:

<https://github.com/tensorflow/lingvo/blob/master/lingvo/core/gpipe.py>

<https://arxiv.org/abs/1811.06965>

GoogleNet:

<https://arxiv.org/abs/1409.4842>

Knowledge distillation:

<https://ai.googleblog.com/2018/05/custom-on-device-ml-models.html>

<https://arxiv.org/pdf/1503.02531.pdf>

<https://nervanasystems.github.io/distiller/knowledge_distillation.html>

DistilBERT:

<https://blog.tensorflow.org/2020/05/how-hugging-face-achieved-2x-performance-boost-question-answering.html>

Two-stage multi-teacher distillation for Q & A:

<https://arxiv.org/abs/1910.08381>

EfficientNets:

<https://arxiv.org/abs/1911.04252>

**Week 4: Model Performance Analysis**

TensorBoard:

<https://blog.tensorflow.org/2019/12/introducing-tensorboarddev-new-way-to.html>

Model Introspection:

<https://www.kaggle.com/c/dogs-vs-cats/data>

Optimization process:

<https://cs231n.github.io/neural-networks-3/>

TFMA architecture:

<https://www.tensorflow.org/tfx/model_analysis/architecture>

TFMA:

<https://blog.tensorflow.org/2018/03/introducing-tensorflow-model-analysis.html>

Aggregate versus slice metrics:

<https://blog.tensorflow.org/2018/03/introducing-tensorflow-model-analysis.html>

What-if tool:

<https://pair-code.github.io/what-if-tool/>

[https://www.google.com/url?q=https://www.youtube.com/playlist?list%3DPLIivdWyY5sqK7Z5A2-sftWLlbVSXuyclr&sa=D&source=editors&ust=1620676474220000&usg=AFQjCNEF\_ONMs8YkdUtgUp2-stfKmDdWtA](https://www.youtube.com/playlist?list=PLIivdWyY5sqK7Z5A2-sftWLlbVSXuyclr)

Partial Dependence Plots:

<https://github.com/SauceCat/PDPbox>

<https://github.com/AustinRochford/PyCEbox>

Adversarial attacks:

<http://karpathy.github.io/2015/03/30/breaking-convnets/>

<https://arxiv.org/pdf/1707.08945.pdf>

Informational and behavioral harms:

<https://fpf.org/wp-content/uploads/2019/09/FPF_WarningSigns_Report.pdf>

Clever Hans:

<https://github.com/cleverhans-lab/cleverhans>

Foolbox:

<https://foolbox.jonasrauber.de/>

Defensive distillation:

<https://arxiv.org/abs/1511.04508>

Concept Drift detection for Unsupervised Learning:

<https://arxiv.org/pdf/1704.00023.pdf>

Cloud providers:

<https://cloud.google.com/ai-platform/prediction/docs/continuous-evaluation>

<https://aws.amazon.com/sagemaker/model-monitor>

<https://docs.microsoft.com/en-us/azure/machine-learning/how-to-monitor-datasets>

Fairness:

<https://www.tensorflow.org/responsible_ai/fairness_indicators/guide>

Model Remediation:

<https://www.tensorflow.org/responsible_ai/model_remediation>

AIF360:

<http://aif360.mybluemix.net/>

Themis ML:

<https://github.com/cosmicBboy/themis-ml>

LFR:

<https://arxiv.org/pdf/1904.13341.pdf>

**Week 5: Explainability**

Fooling DNNs:

<https://arxiv.org/pdf/1607.02533.pdf>

<https://arxiv.org/pdf/1412.6572.pdf>

XAI:

<http://www.cs.columbia.edu/~orb/papers/xai_survey_paper_2017.pdf>

Interpretable models

<https://christophm.github.io/interpretable-ml-book/>

<https://www.tensorflow.org/lattice>

Dol bear law:

<https://en.wikipedia.org/wiki/Dolbear%27s_law>

TensorFlow Lattice:

<https://www.tensorflow.org/lattice>

<https://jmlr.org/papers/volume17/15-243/15-243.pdf>

PDP:

<https://github.com/SauceCat/PDPbox>

<https://scikit-learn.org/stable/auto_examples/inspection/plot_partial_dependence.html>

Permutation Feature Importance:

<http://arxiv.org/abs/1801.01489>

Shapley values:

<https://en.wikipedia.org/wiki/Shapley_value>

SHAP:

<https://github.com/slundberg/shap>

 TCAV:

<https://arxiv.org/pdf/1711.11279.pdf>

 LIME:

<https://github.com/marcotcr/lime>

Google Cloud XAI

<https://storage.googleapis.com/cloud-ai-whitepapers/AI%20Explainability%20Whitepaper.pdf>

Integrated gradients:

<https://arxiv.org/pdf/1703.01365.pdf>