COMS E6998 001/12650

Practical Deep Learning Systems Performance

Lecture 1 09/10/20

Class Introduction

- *Instructor*: Parijat Dube pd2637@columbia.edu
  Research Staff Member at IBM Research, NY

- *CA:* Brandon Liang xl2891@columbia.edu

- *Student Information Sheet:*
  - Introduction Survey posted at courseworks (under Quizzes) on Sept. 6
  -

- *Prerequisites:*
  - Basic knowledge of ML algorithms
  - Programming experience in python

Today's Agenda

- Introductory Survey Response
- Course Overview
  - Syllabus
  - Assignments and Grading
  - Logistics
- Introduction to Machine Learning / Deep Learning
- Real World Deep Learning Systems
- Machine Learning on Cloud and Model Lifecycle
- Model Performance and Complexity Tradeoffs

Survey Response: Your Expectations

- Practical deep learning techniques
- Improve efficiency of deep learning algorithms and networks
- Efficient deployment of large-scale DL systems in real–world scenarios
- Deep learning on cloud with GPUs including docker and Kubernetes
- Scientific approaches to DL system performance evaluation during training and inference
  - Available tools and techniques
- Manage the full lifecycle of a robust, scalable ML system
  - Taking the models from training to deployment
- Search for appropriate neural network architecture
- Benchmarking of deep learning systems
- Career as an ML/AI engineer

Course Information

- *What this course will cover ?*
  - DL training architectures, frameworks, hyperparameters
  - Cloud based DL systems and performance issues
  - DL systems performance evaluation tools, techniques, benchmarks
  - Programming assignments involving GPUs on cloud
  - Guest lectures (1 or 2) by experts in ML/DL systems from industry
  - Paper readings to solve assignment questions
- *What this course will not cover ?*
  - Details about specific ML/DL algorithms
  - Mathematical analysis of ML/DL algorithms

Course Contents: Module 1

Introduction to Machine Learning (ML) and Deep Learning (DL)

- ML revolution and cloud
- ML System and its constituents
- ML performance concepts/techniques: overfitting generalization, bias, variance, regularization
- Performance metrics: algorithmic and system Level
- DL training: backpropagation, activation functions, data preprocessing, batch normalization, SGD and its variants, exploding and vanishing gradients, weight initialization, learning rate schedule
- Regularization techniques in DL Training: dropout, early stopping, data augmentation

Course Contents: Module 2

DL Training: Architecture, Frameworks, Hyperparameters

- DL training architectures
  - Model and Data Parallelism
  - Single node training
  - Distributed training
- DL training hyperparameters
  - Batch size, Learning rate, Momentum, Weight decay; Convergence and Runtime
  - Relation between batch size and learning rate
- Hardware Acceleration: CPUs, GPUs, FPGAs, TPUs, Specialized systems for AI acceleration
- Specialized DL architectures: CNNs, RNNs, LSTMs, GANs

Course Contents: Module 3

ML and Cloud Technologies

- ML system stack on cloud
- Micro-services architecture, Docker, Kubernetes, Kubeflow
- Cloud storage: File, Block, Object storage; performance (latency, throughput), cost, scalability

Cloud Based ML Platforms

- ML as a service offering: AWS, Microsoft, Google, and IBM
- System stack, capabilities and tools support
- Monitoring and observability
- Performance and availability

Course Contents: Module 4

DL Performance Evaluation: Tools and Techniques

- Monitoring tools for GPU resources (nvprof, nvidia smi)
- Time series analysis of resource usage data
- Predictive performance modeling techniques
  - Black-box vs white-box modeling
  - Analytical modeling
- Predictive performance models for DL: accuracy and runtime

Course Contents: Module 5

ML Benchmarks

- DAWNBench, Mlperf
- Datasets: MNIST, CIFAR10/100, ImageNet, Kaggle, OpenML
- Performance metrics for DL jobs
  - Runtime, Cost, Response time, Accuracy, Time To Accuracy (TTA)
- Study of published numbers by different cloud service providers/vendors
- Compare performance scaling across GPUs for different models in MLperf
- Open Neural Network Exchange (ONNX)

Course Contents: Module 6

DL Systems Performance

- Training-logs: framework specific support, instrumentation, analysis
- Checkpointing: framework specific support, restarting from checkpoint
- Job Scheduling on Cluster:
  - Policies: FIFO, Gang, Earliest Deadline First
  - Job Scheduler : Kubernetes default, Gandiva, Optimus
- Job Elasticity: scaling GPUs during runtime, platform support
- Scalability: learners, batch size, single node, distributed

Course Contents: Module 7

Operational DL and MLOps

- Model Lifecycle management
- DL systems testing and quality
- Drift detection and re-training
- Robustness and Adversarial training
- Devops principle in machine learning
- MLOps and associated tool-chain

Course Contents: Module 8

Special Topics

- Transfer Learning: finetuning and pseudo-labeling techniques
- Deep Reinforcement Learning systems
- Neural Network synthesis and architecture search (NAS)
- Hyperparameter optimization
- Automated Machine Learning (AutoML)
- Federated learning and ML on edge devices
- TinyML and Tensorflow Lite

Survey Response Module Interest

- Hottest: Module 6 and 7 -- 2 lectures each
- Hotter: Module 2,3,4,5,8 -- 1.5 lecture each
- Hot: Module 1 -- 1 lecture

Survey Response: Interesting topics

- Training Architectures, Frameworks, Hyperparameters
- ML Systems stack on cloud
- Cloud based ML platforms from AWS, Microsoft, Google, and IBM
- L**wLL**
- **FL and ML on edge devices**
- docker, kubernetes, kubeflow, MLFlow, AWS
- DL deployment and production cycle
- Microservices architecture, cloud-based ML platforms, monitoring, checkpointing, scalability, MLOps, full lifecycle testing, drift detection and retraining, adversarial training, **automated ML, NN synthesis, ML on edge devices**
- DevOps principles in ML
- Automated ML
- **Deep Reinforcement learning systems**

Survey Response: Additional topics

- ML models deployment at large scale and ML pipelines
  - Will cover under Automated ML
- Data planning, preparation, and data pipeline
  - Will cover in the context of both ML and DL
- Training and finetuning using large scale NLP models
  - Will cover under Transfer learning and fine-tuning
- Federated learning
- Choice of models for different kind of real-world problems
- New DL architectures

Assignments and Grading

- **Distribution of marks**:
  - Assignments (50): 50%
  - Readings/Quizzes: 10%
  - Seminar: 15%
  - Final Project: 25%
- **Assignments (50%)**
  - 5 assignments
  - Assignments posted at the end of lectures 2, 4, 6, 8, 10; due in 2 weeks
  - Assignments to be submitted using courseworks
  - All programming assignments should be done as Jupyter notebooks
- **Readings/Quizzes (10%)**
  - Assigned article reading and answering questions based on them. Not be spending 1-2 hrs per reading assignment.
  - Quizzes with multiple choice and/or short answer questions. Each quiz will not have more than 10 questions.

Assignments and Grading (contd.)

- **Seminar (15%):** Team assignment. Team of 2**.** Read 4+ papers on a DL topic and prepare a 20 min technical presentation. Detailed rubric shall be provided. Discuss topic and related papers with me. Seminar grading:
  - Seminar presentation: 10%
  - Technical blog: 5%
- **Project (25%):** Team assignment. Team of 2. Any project around performance of DL systems. Discuss project ideas with me. 1-page project proposal due by Midterm. Detailed rubric shall be provided. Project grading:
  - Project proposal (5%) – due in mid October
  - Report (5%)
  - Github repo with README, documented code (5%)
  - Final presentation and demo (10%)

Example Projects from Previous Classes

- Serving ML models on edge devices with less memory and compute
  - Using post training quantization (TF-lite) and model optimization for target hardware (OpenVINO)
- Performance of Serverless vs Server Based Learning with Tensorflow.js
- Image classification on edge devices
  - Use of quantization and pruning to reduce model size by 90%
- Neural Architecture Search for Encrypted Classifiers
  - Integrate NAS and tf-encrypted encrypted classification
- Performance comparison of CPU/GPU and Edge TPU
- Cloud based service for hyperparameter optimization and performance evaluation
- Train an agent using reinforcement learning to find a diamond in Minecraft

Example Seminar from Previous Classes

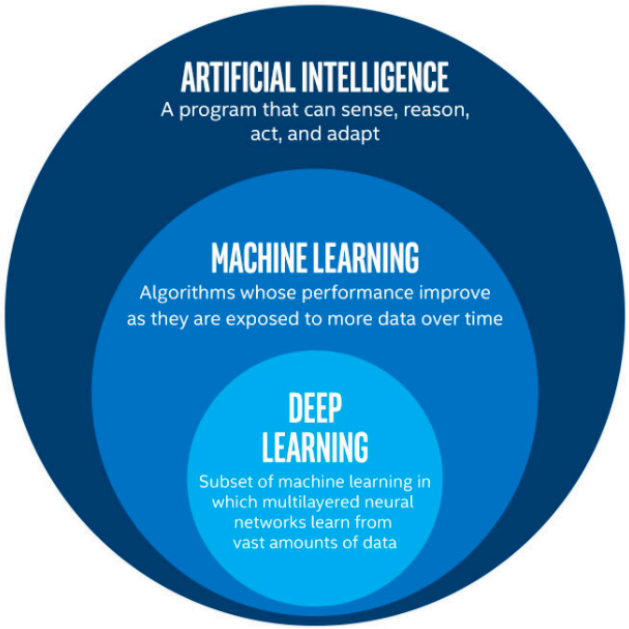
- Evaluating frameworks for Automated Machine Learning
- Batch Normalization (2)
- The Applications of Deep Learning in Finance
- Contemporary NLP Language Models and their Performance Issues
- Explaining Neural Networks for Transfer Learning
- Model Compression & Knowledge Distillation in Deep Learning NLP
- Image Forensics: Detecting Manipulated Images
- Distributed training (3)
- Choice of Learning Rate and Batch Size for Better Generalization
- Neural Architecture Search
- Deep Learning in Speech Recognition

Class Logistics

- Reach me: Office hours (Fridays 4-6 PM), email, piazza
- Reach the CAs
- Access to Compute and Storage Clusters
  - Amazon Web Services ($100 coupons for free)
  - Google Cloud Platform ($300 coupons will be given)
- Class communications: courseworks/Piazza
- Class website:

https://courseworks2.columbia.edu/courses/109828

AI, ML, DL



Source: Intel. "How to Get Started as a Developer in AI." October 2016

- AI is dubbed as the Fourth wave of Industrial Revolution
- Projection for 2030: Add $15.7tn to global economy; Contribute to +14% global GDP growth
- DL is a subset of ML and involves use of Deep Neural Networks (DNNs)

Examples of AI

- Visual Recognition
- Handwriting Recognition
- Self Driving Cars: Intel Mobileye, Waymo
- Conversational agents: IBM Watson Assistant
- Transportation: Uber AI
- Art: Generation of artworks

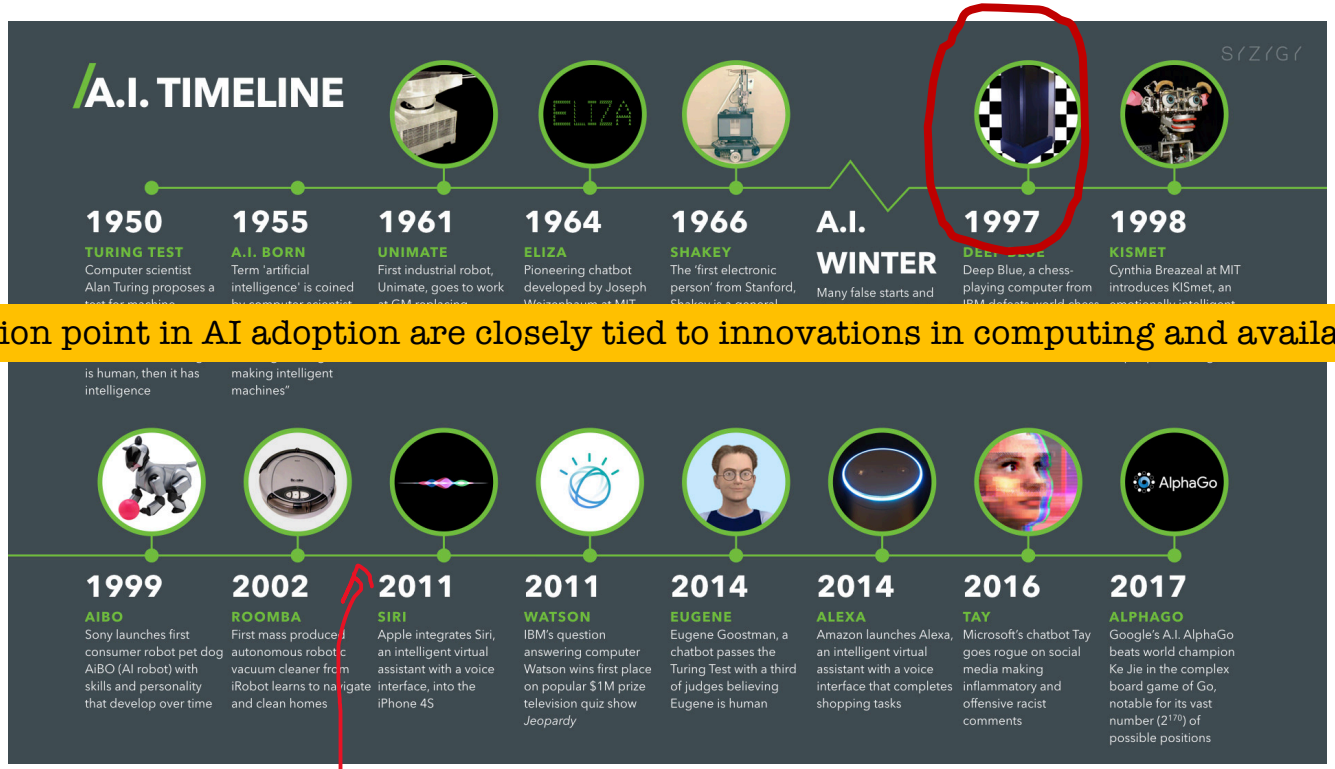
AI Blogs of Major Companies

- Facebook AI: https://ai.facebook.com/blog/
- Google AI Blog: https://ai.googleblog.com
- IBM Research Blog AI:

  https://www.ibm.com/blogs/research/category/ai/
- Microsoft The AI Blog: https://blogs.microsoft.com/ai/
- AWS Machine Learning Blog: https://aws.amazon.com/blogs/machine-learning/

AI Timeline

Cloud computing was coined



Inflection point in AI adoption are closely tied to innovations in computing and availability of big data

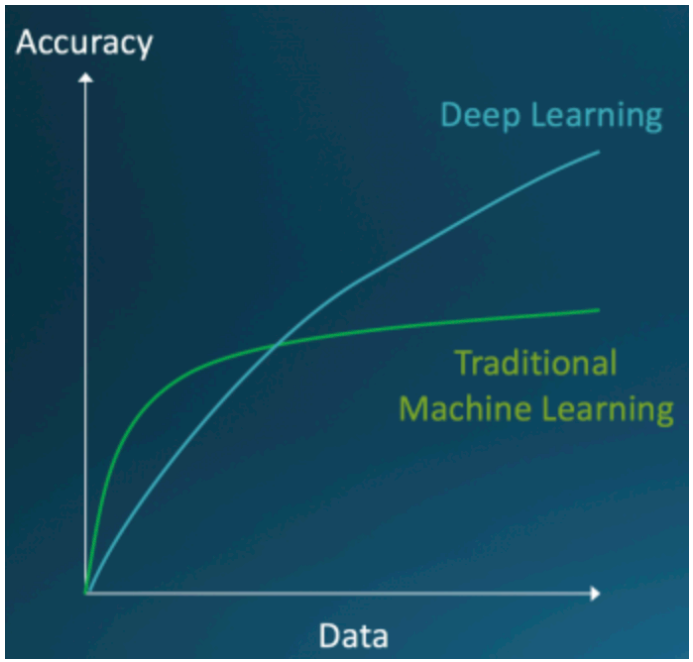
2006: Amazon elastic cloud and S3 was launched

Factors Contributing to AI Success

- **Algorithms, Data, Compute, Applications**
  - Distributed training algorithms scaling upto 100s of GPUs
  - Data growing at exponential rate; Internet, Social media, IoT
  - Compute power growth with specialized cores; GPUs, TPUs
  - Development of innovative applications
- **2012** Alexnet by Krizhevsky et al at ImageNet Competition
  - Simple convolutional neural network: 5 convolutional, 3 fully connected
  - GPU based; Beat other models by 11% margin
  - Triggered ”Cambrian Explosion” in deep learning technologies

  “*Neural networks are growing and evolving at an extraordinary rate, at a lightening rate,….What started out just five years ago with AlexNet…five years later, thousands of species of AI have emerged*.”

Deep Learning Revolutionized Machine Learning



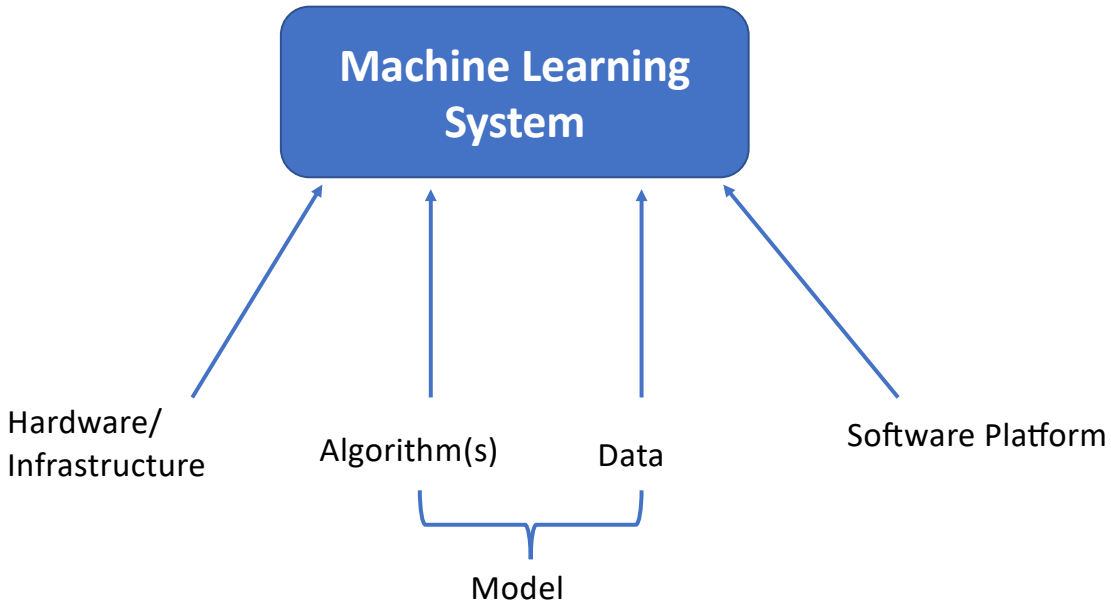
Machine Learning System

A composition of one or more software components, with possible interactions, deployed on a hardware platform with the purpose of achieving some performance objective.

A Machine Learning system is a system where one or more software components are machine learning based.

- Why study ML systems ?
  - Algorithms run on real and (possibly) faulty hardware in production environments
  - Theoretical performance is far away from observed
  - To characterize holistic performance of not just the algorithm but the end-to-end performance of the entire system

Constituents of a ML System



Infrastructure

- Compute units and accelerators, Memory, Storage, Network
- Resources can be acquired as bare metal, VMs/Containers on cloud
- Hardware can help improve performance pretty much everywhere in the pipeline
- Design better hardware
  - Adapt existing architectures to ML tasks.
  - Develop brand-new architectures for ML.
- Hardware compute precision affects performance

(Learning) Algorithm

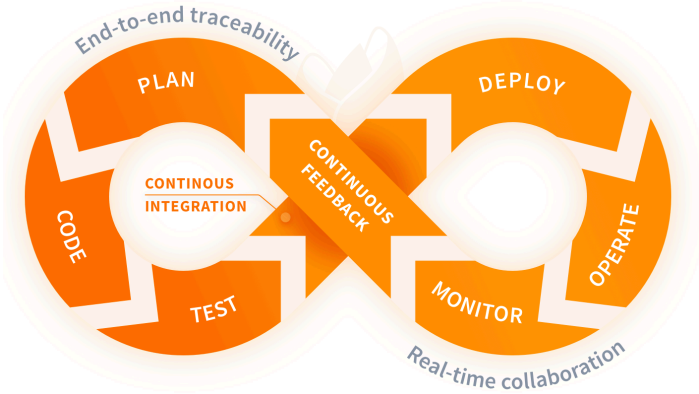
- General and domain specific architectures
- Hyperparameter tuning to extract the best performance
- Effects the resource requirements: compute (FLOPS), memory
- Performance (runtime) and scalability of an algorithm depends on:
  - Hardware/Infrastructure
  - Software platform (frameworks, libraries, drivers)

Data

- Data as a critical element; Data is the King in ML
- Different modalities: Audio, video, images, text
- Data sources, collection, labeling, quality, data storage
- Data type determines the choice of learning algorithm
- Making the data *business ready* is challenging
- Many data-driven organizations are spending 80 percent of their time on data preparation and find it a major bottleneck.
- DataOps

Software Engineering in ML Systems

- Machine learning applications run as pipelines that ingest data, compute features, identify model(s), discover hyperparameters, train model(s), validate and deploy model(s).
- Making a model as a production-capable web service
  - Containerization (docker), cluster deployment (K8s)
  - APIs exposed as web service (Tensorflow serving/ONNX runtime)
- Workflow engines (Kubeflow) automate the ML pipeline
- Deployment monitoring and operational analytics
- Devops principles applicable to ML Systems:
  - Continuous Integration, Continuous delivery (CI/CD)
  - Predictability
    - "A model may be unexplainable—but an API cannot be unpredictable"
  - Reproducibility and Traceability
    - Provenance for Machine Learning Artifacts



ML Specific testing and monitoring apart from traditional software testing

- Data testing
- Infrastructure testing
- Model testing
- Production testing

Creating an ML System

- **System Objective:** What problem will my system solve ? What is the target deployment scenario? What are the performance objectives ?
- **Solution Approach**: What is my solution and its components ?
- **Data Collection**
  - Identifying the data sources: What are my data sources ?
  - Collecting the data
- **Data Preparation**
  - Preparing the data: Is my data business ready ?
  - Ingesting the data: What is the right storage for my data ?
- **Model Development**
  - Identification and training
  - Model evaluation
  - Hyperparameter Tuning
- **Model Deployment**
  - Model optimization (if needed) for the deployment infrastructure
  - Model packaging and deployment
- **Monitoring and Feedback**
  - Is my deployed model performing as expected ?
  - Is there a drift in model performance which requires re-training ?

AI at US Open: IBM AI Highlights

- Read article: https://www.ibmbigdatahub.com/blog/how-ai-picks-most-exciting-moments-us-open-without-bias
- Visual recognition, Action detection, Facial emotions recognition
- **Features of input**: hand clapping, player actions, court cheering

AI for Agriculture

- Facebook AI Blog: PyTorch drives next-gen intelligent farming machines
- Chris Padwick medium article: AI for AG: Production machine learning for agriculture

Search for Exoplanets with Machine Learning

- Earth to exoplanet: Hunting for planets with machine learning
- Data from NASA's Kepler Mission: 200,000 stars over 4 years; image take every 30 seconds
- Roughly 14 billion data points
- Selected **15,000 labeled data points** across 677 stars to train a **Tensorflow model**
- **Test accuracy: 96%**
- Discovered 2 new planets: Kepler-90i and Kepler-80g
- Details at Google AI blog: Opensourcing the hunt for planets

AI for Healthcare

- Read article: Opinion article: Artificial Intelligence for Healthcare in Africa
- Stanford news: Deep learning algorithm does as well as dermatologists in identifying skin cancer
- Read article: The Real-World Benefits of Machine Learning in Healthcare
- Video: https://youtu.be/i8CQ8Ud0HN0

DeepMind's AlphaGo Zero

- AlphaGo Zero: Starting from Scratch
- 2017 NIPS Keynote by David Silver
- Nature article: Mastering the game of Go without Human Knowledge
- David Foster medium article: AlphaGo Zero Explained in one Diagram
- AlphaGo Zero Cheat Sheet

Inhibitors in Successful Implementation of ML Solutions

- Deployment and automation
- Reproducibility of models and predictions
- Diagnostics
- Governance and regulatory compliance
- Scalability
- Collaboration
- Monitoring and management

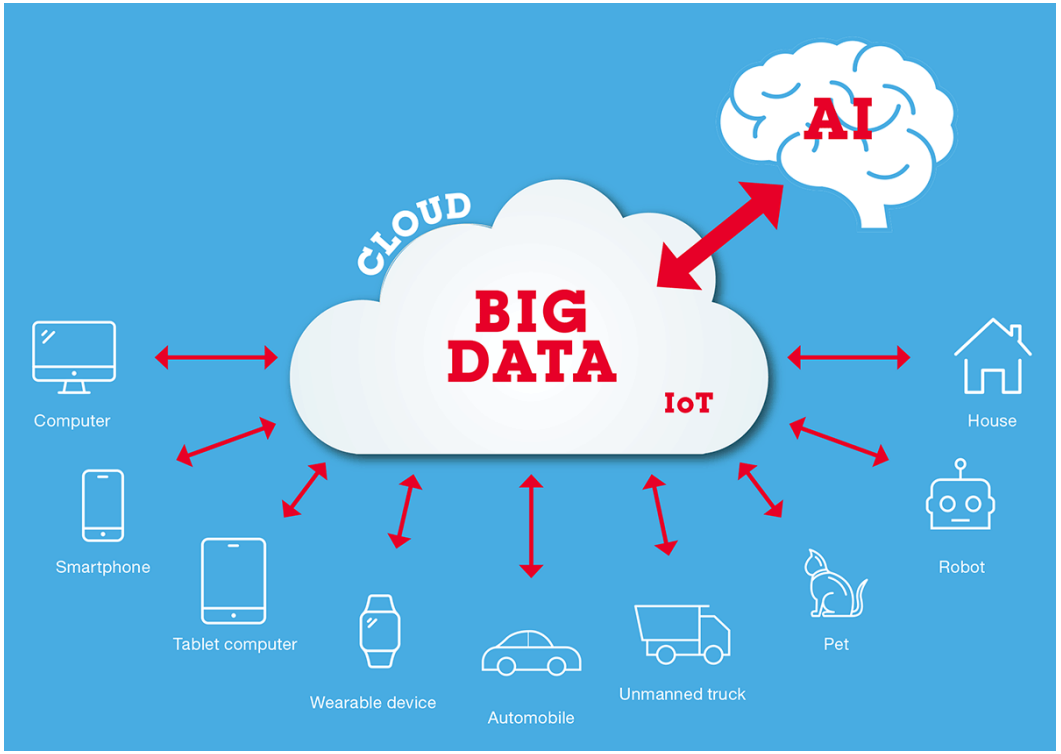
Cloud Computing

- Access to computing resources and storage on demand
- Pay-as-you go model
- Heterogeneous resources: GPUs, CPUs, storage type
- Different offering models: IaaS, PaaS, SaaS, MLaaS
- Different deployment models: Public, private, hybrid cloud
- Provisioning, maintenance, monitoring, life-cycle-management

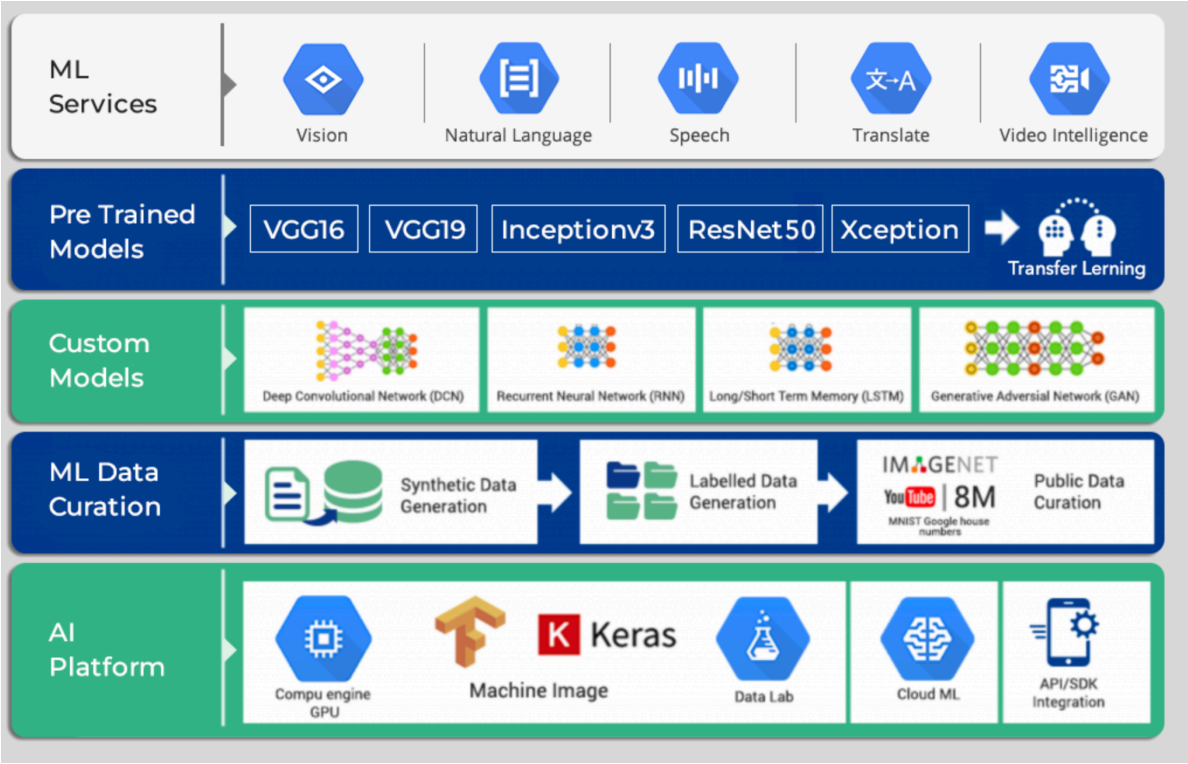
Marriage of Cloud and AI

- AI
  - Harness power of Big Data and compute
- Cloud
  - Access to Big Data
  - Platform to quickly develop, deploy, and test AI solutions
  - Ease in AI reachability

- Cloud + AI is the winning combination



Deep Learning on Cloud: System Stack



Cloud based Machine Learning Services

- IBM Watson Machine Learning

https://www.ibm.com/cloud/machine-learning

- Amazon Sagemaker

https://aws.amazon.com/sagemaker

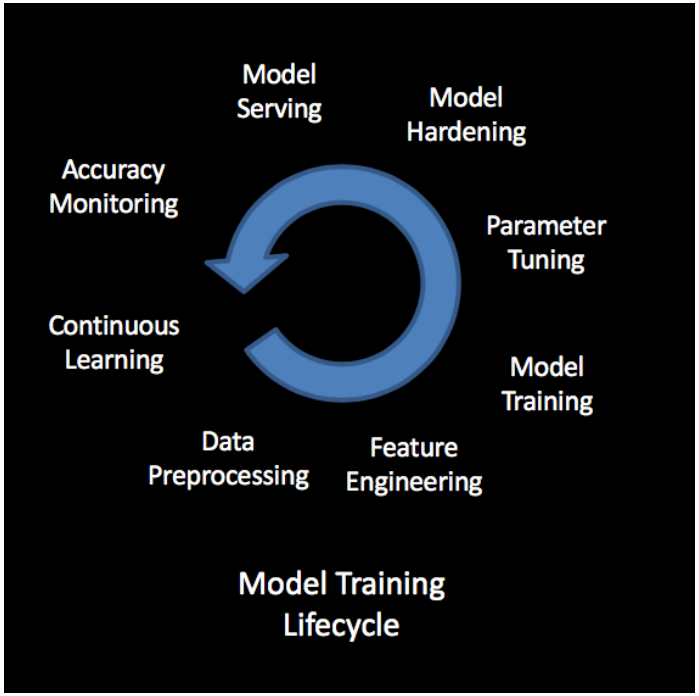
- Microsoft Azure Machine Learning

https://azure.com/ml

- Google Cloud Machine Learning

https://cloud.google.com/ml-engine

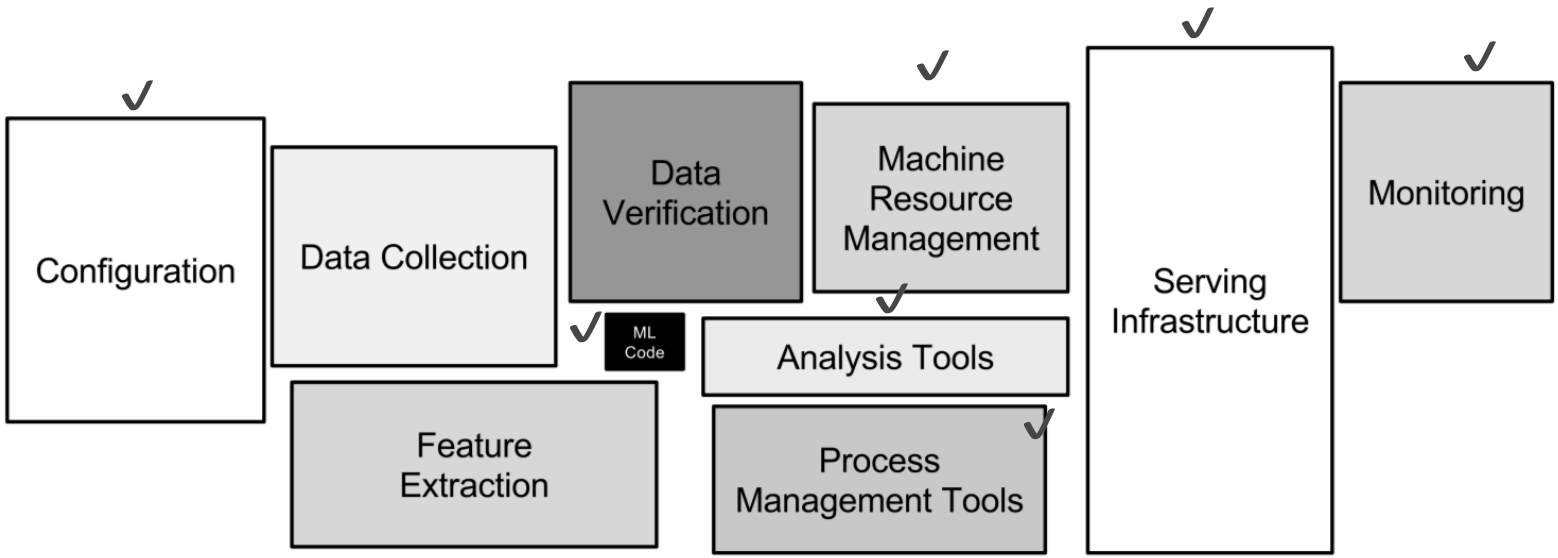
AI Model Training Lifecycle



Performance considerations at each stage

- Data preprocessing: de-noising, de-biasing, train/test set creation
- Feature engineering: search efficient data transformations
- Model training: model identification/synthesis, hyperparameter tuning, regularization
- Model hardening: efficient adversarial training
- Model serving: hardware, model pruning and compression
- Monitoring: response time, drift detection
- Continuous learning: model adaptability, retraining

Practical Machine Learning Systems



ML Systems Testing and Monitoring

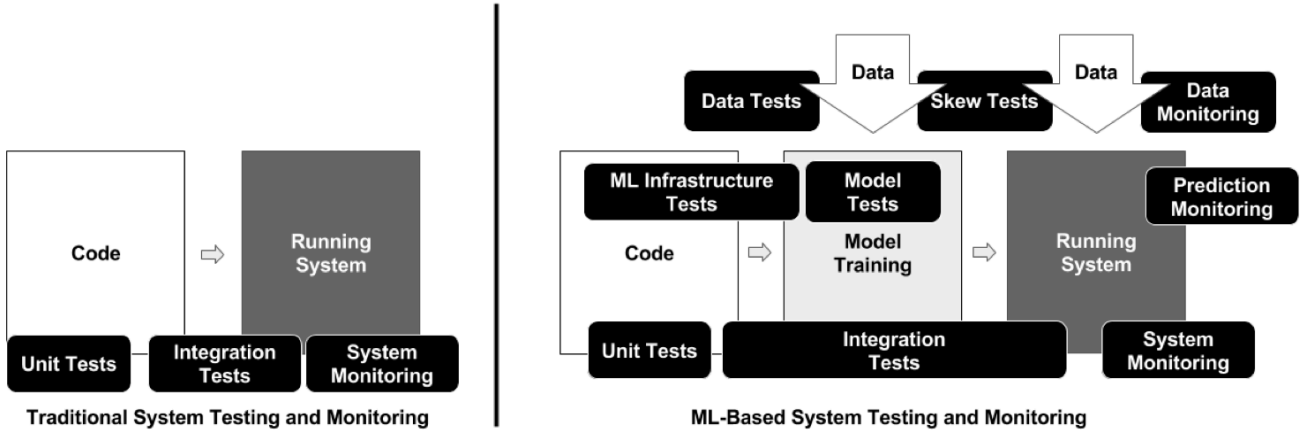


Figure 1. **ML Systems Require Extensive Testing and Monitoring.** The key consideration is that unlike a manually coded system (left), ML-based system behavior is not easily specified in advance. This behavior depends on dynamic qualities of the data, and on various model configuration choices.

ML Model Training and Inferencing

- What is ML model ?
  - A computer program trained to learn from data and preform tasks requiring human intelligence
- What is ML model training ?
  - Process of using data to learn the ML model
- What is inferencing ?
  - Process of using the trained model to make predictions on test data
- Why is data needed and important ?
  - ML is data-driven machine intelligence; ML model knowledge is all learned from the data
- Type of data for ML training ➔ Type of training (Supervised vs Unsupervised)

Types of Learning Algorithms

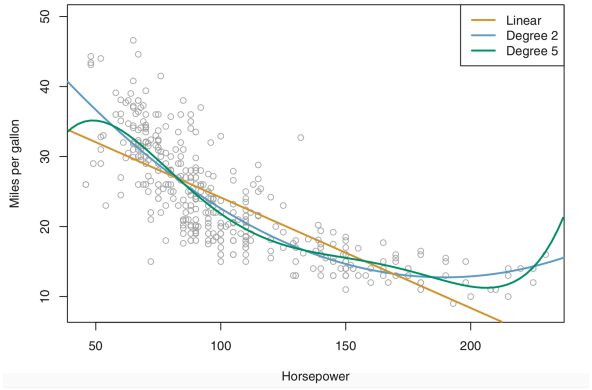
- **Supervised**: learning using labeled training dataset
  - Learning task it to infer a function mapping input to the output using labeled input-output pairs
    - Inferred function correctly predicts output labels for other inputs
  - Example Algorithms
    - Regression: Linear, Logistic
    - Support Vector Machine
    - Backpropagation based neural network
    - Tree based: Decision Trees, Random Forest
- **Unsupervised**: learning using un-labeled training dataset
  - Learning task is to infer relationships in the input dataset that can help in your understanding of the data
    - Infer clusters/patterns in data based on input features
  - Examples: Clustering, Autoencoder
- **Reinforcement**: learning by sensing, acting, and evaluating reward
- Also **Semi-supervised**

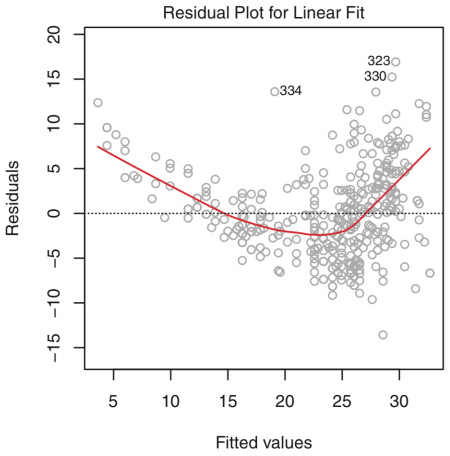
ML Tasks: Classification and Regression

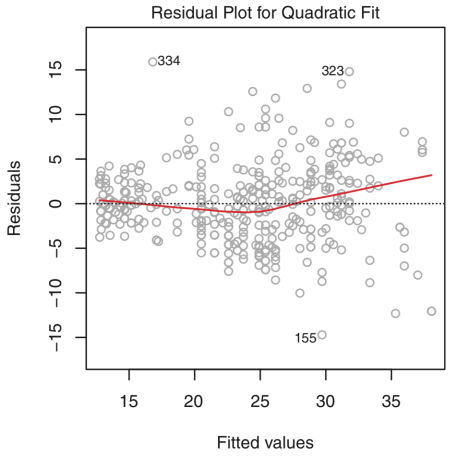
- **Classification**: Output is a categorical, unordered variable
  - Binary classification
    - Tumor: benign vs malignant
    - Food image : Greek vs Non-Greek
  - Multiclass classification
    - Tumor: benign, stage-1, stage-2, stage-3
    - Food image: falafel, salad, pita, …
- **Regression**: Output is a continuous valued real variable
  - Finance stock price prediction
  - House value price prediction

Linear Regression

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$







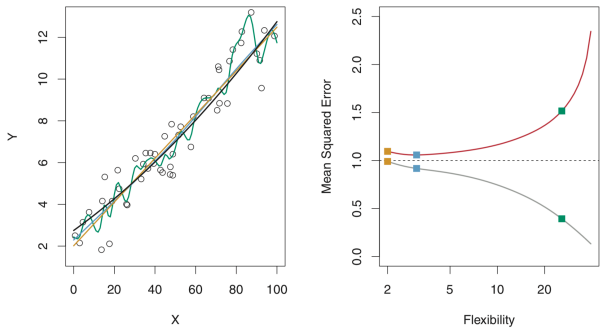
$$\text{TSS} = \sum (y_i - \bar{y})^2$$

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

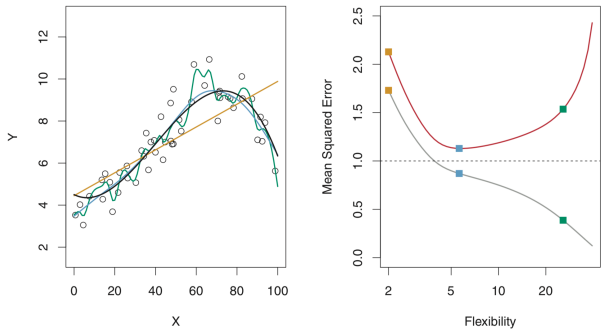
$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}}$$

Mean Square Error (MSE)

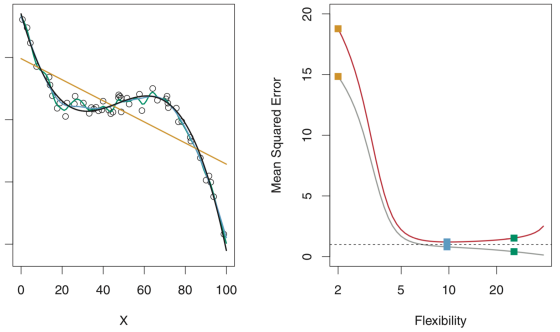
CASE 1



CASE 2



CASE 3



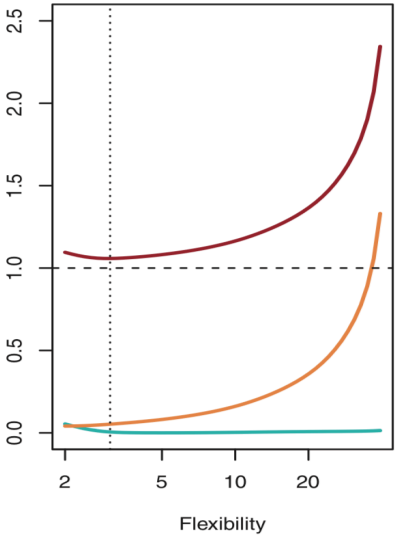
true label $Y = f(X) + \epsilon.$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

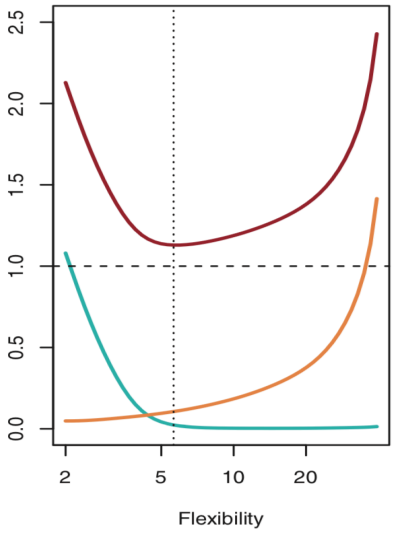
predicted $\hat{Y} = \hat{f}(X)$

Model Complexity Tradeofffs

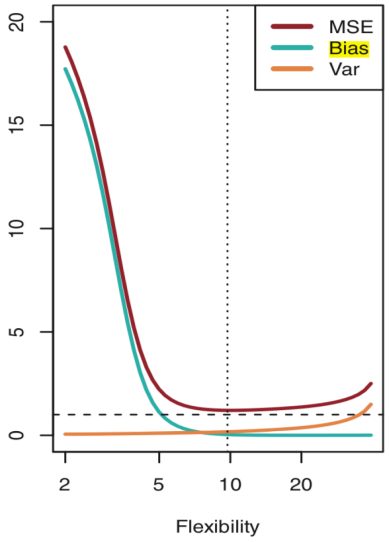
CASE 1



CASE 2



CASE 3

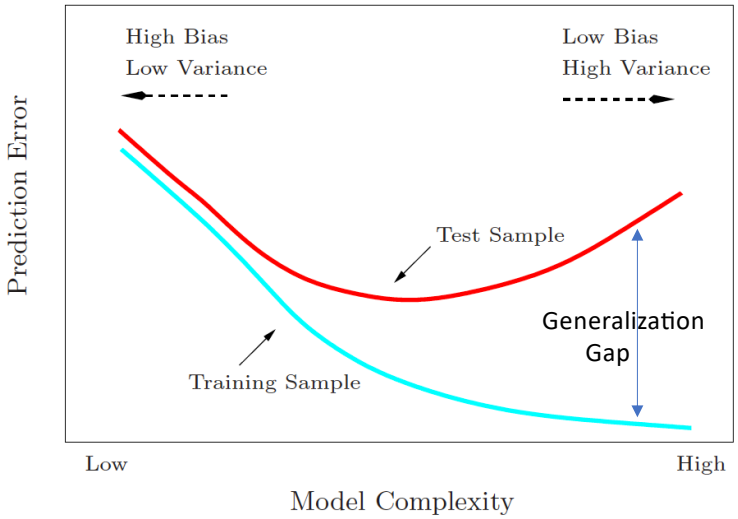


$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon)$$

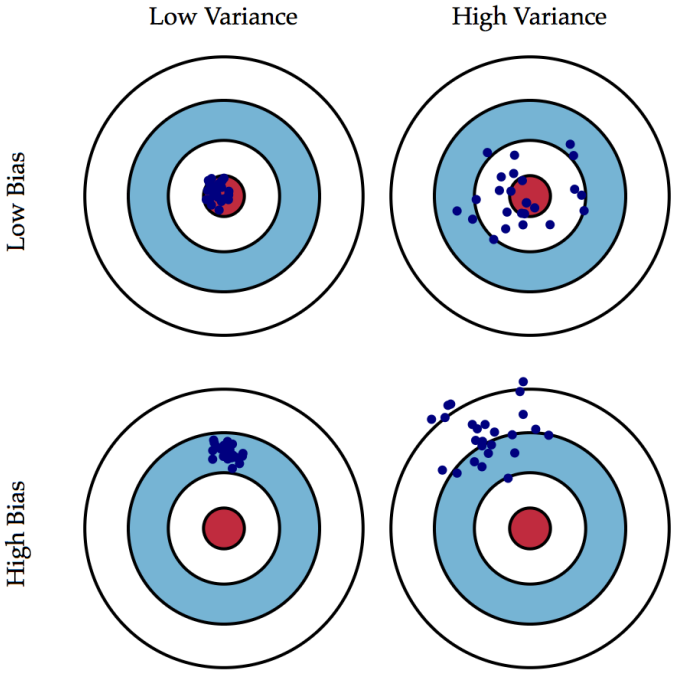
Irreducible

Model Complexity Tradeoffs

- Simple model
  - Fail to completely capture the relationship between features
  - Introduce bias: Consistent test error across different choices of training data
  - Low variance
  - Increasing training data does not help in reducing bias
- Complex model captures nuances in training data causing Overfitting
  - Low bias
  - Train error << Test error
  - With different training instances, the model prediction for same test instance will be very different – High Variance



Bias-Variance Tradeoff



$$MSE_{test} = Bias^2 + Variance + Irreducible\ error$$

High Bias

Increase model complexity

Low Bias

High Variance

Increase training data

Low Variance

Prepare for Lecture 2

- Get access to compute cluster
  - Amazon Educate https://aws.amazon.com/education/awseducate/
  - Google cloud platform
  - Reach me/TAs of you have any problem
- Take account and familiarize with cloud computing clusters
  - How to run Jupyter notebooks ?
  - How to run training jobs ?
- First home-work posted on 09/17, due by 10/01
- Look at class website on courseworks for syllabus, announcement and homework

Reference Materials

- Machine Learning Process
  - Blog: The 7 Steps of Machine Learning
  - How to build a machine learning model in 7 steps
  - Frameworks for Approaching the Machine Learning Process

Recommended Books

- This course does not follow a single text book
- List of books (covering similar topics)
  - Charu Aggarwal “Neural Networks and Deep Learning”, available at rd.springer.com
  - Goodfellow, Bengio, Courville, “Deep Learning”, available at http://www.deeplearningbook.org
  - Umberto Michelucci, “Applied Deep Learning: A Case-Based Approach to Understanding Deep Neural Networks”
  - G. James et al “Introduction to Statistical Learning Theory”, available at http://faculty.marshall.usc.edu/gareth-james/ISL/