COMS E6998 012/15099

Practical Deep Learning Systems Performance

Lecture 1 09/05/19

Class Introduction

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

- *TAs:* Daniel Jeong dpj2108@columbia.edu
  Madhavan Seshadri ms5945@columbia.edu

- *Student Information Sheet*
  - Please submit by tomorrow 09/06/19

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

Student Information

- *Send me an email with following information*
  - Name, degree enrolled, department
  - Your expectations from the course
  - Specific things you would like to learn (apart from those listed in the syllabus)
  - Checklist on what you know (these are not prerequisites, just for my understanding)
    - ❑ Python
    - ❑ Machine learning
    - ❑Deep learning
    - ❑Git
    - ❑Cloud computing
  - Any specific question about the course ?

Lecture 1 Agenda

- Course Overview
  - Contents
  - Assignments and Grading
  - Logistics
- Introduction to Machine Learning / Deep Learning
- Real World Applications of Deep Learning
- Machine Learning on Cloud and Model Lifecycle
- Model Performance and Complexity Tradeoffs

Course Information

- *What this course will cover ?*
  - Performance related concepts of ML algorithms
  - ML training architectures, frameworks, hyperparameters
  - Cloud based ML systems and performance issues
  - ML systems performance evaluation tools, techniques, benchmarks
  - Paper reading and programming assignments
- *What this course will not cover ?*
  - Details of standard ML algorithms
  - Mathematical analysis of ML algorithms

Course Contents: Module 1

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

- ML revolution and cloud
- Overview of ML algorithms, supervised and unsupervised Learning
- ML performance concepts/techniques: bias, variance, generalization, regularization
- Performance metrics: algorithmic and system Level
- DL training: backpropagation, gradient descent, activation functions, data preprocessing, batch normalization, SGD and its variants, exploding and vanishing gradients, weight initialization, learning rate policies
- 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
  - Parameter server
- DL training frameworks: Spark, Caffe, **Tensorflow, Pytorch**, Keras
- DL training hyperparameters
  - Batch size, Learning rate, Momentum, Weight decay; Convergence and Runtime
- Hardware Acceleration: GPUs, Tensor cores
- 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 and flexibility
- Network support on cloud platforms

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: GPU resources (nvprof, nvidia smi), host system (top, iostat), network monitoring,
- Time series analysis of resource usage data
- Predictive performance modeling techniques
  - Black-box vs white-box modeling
  - Linear and non-linear regression
  - Analytical modeling
- Predictive performance models for DL: accuracy and runtime

Course Contents: Module 5

ML Benchmarks

- DAWNBench, MLperf suite, Tensorflow HPM
- Kaggle, OpenML
- Datasets: MNIST, CIFAR10/100, ImageNet
- 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, Gandiva, Optimus
- Job Elasticity: scaling GPUs during runtime, platform support
- Scalability: learners, batch size, single node, distributed
- Overview of conferences at intersection of ML and Systems

Course Contents: Module 7

Advanced Topics

- Transfer Learning: finetuning and pseudo-labeling techniques
- Reinforcement Learning
- Neural Network synthesis and architecture search
- Hyperparameter optimization
- Automated Machine Learning
- Robustness and Adversarial training
- Bias in models and De-biasing techniques
- Devops principle in machine learning; Model Lifecycle management
- Drift detection and Re-training

Assignments and Grading

- **Distribution of marks**:
  - Assignments: 60%
  - Technical community participation: 10%
  - Seminar/Project: 30%
- **Assignments**: **60%**
  - 6 assignments
  - Assignments posted at the end of lectures 2, 4, 6, 8, 10, 12; due in 2 weeks
- **Technical community participation: 10%**
  - Technical blogs e.g., Medium
  - Code contribution and reuse
  - Performance data sharing (we will create a repository)

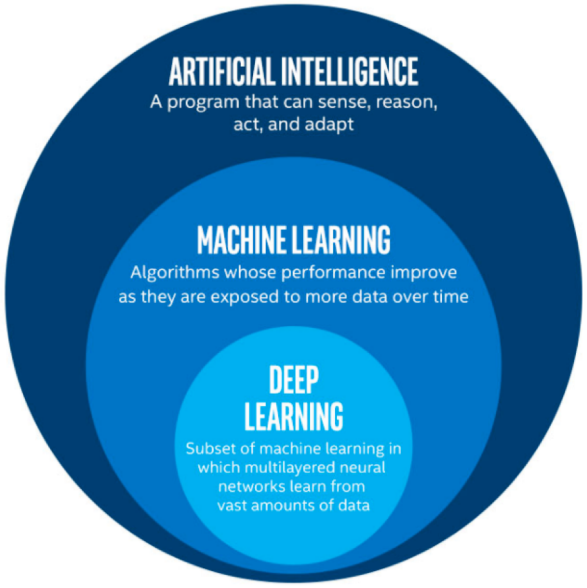
Assignments and Grading (contd.)

- **Seminar OR Project: 30%**
  - **Seminar:** Read 4+ papers on a DL topic and prepare a 10 min technical presentation. Grading will be based on your technical understanding of the papers (10%), slides content and presentation (10%), performance in the q/a session (10%). I can help you in identifying topic and related papers.
  - **Project:** Any project around performance of DL systems. Project can be done as a team of 2 students. Grading will be based on technical contents (10%), innovation (10%), and presentation (10%). Discuss project ideas with me.
- Assignments to be done using Courseworks and github
- All programming assignments should be done as Jupyter notebooks

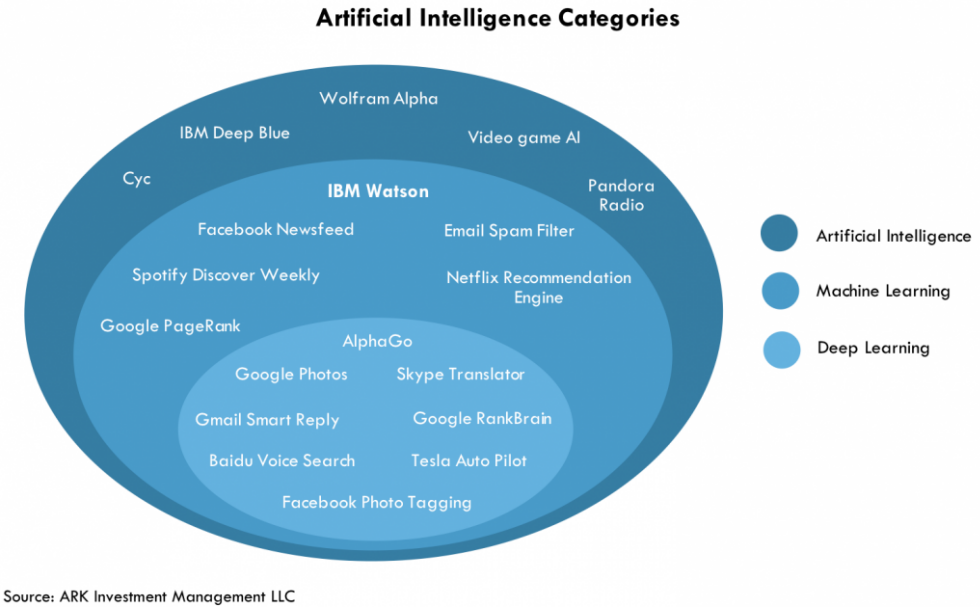
Class Logistics

- Reach me: Office hours (Thursdays 4-6 PM), email, courseworks
- Reach TAs
- Access to Compute and Storage Clusters
  - IBM Watson Machine Learning and Cloud Object Storage
  - Amazon Web Services and Amazon S3
  - Google Cloud Platform and Google Storage
- Working session to onboard and start using cloud computing clusters (if needed)
- Class communications: courseworks/Piazza

AI, ML, DL



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



Source: ARK Investment Management LLC

- AI is dubbed as the Fourth wave of Industrial Revolution
- AI is a disruptive technology
- Projection for 2030: Add $15.7tn to global economy; Contribute to +14% global GDP growth

Examples of AI

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

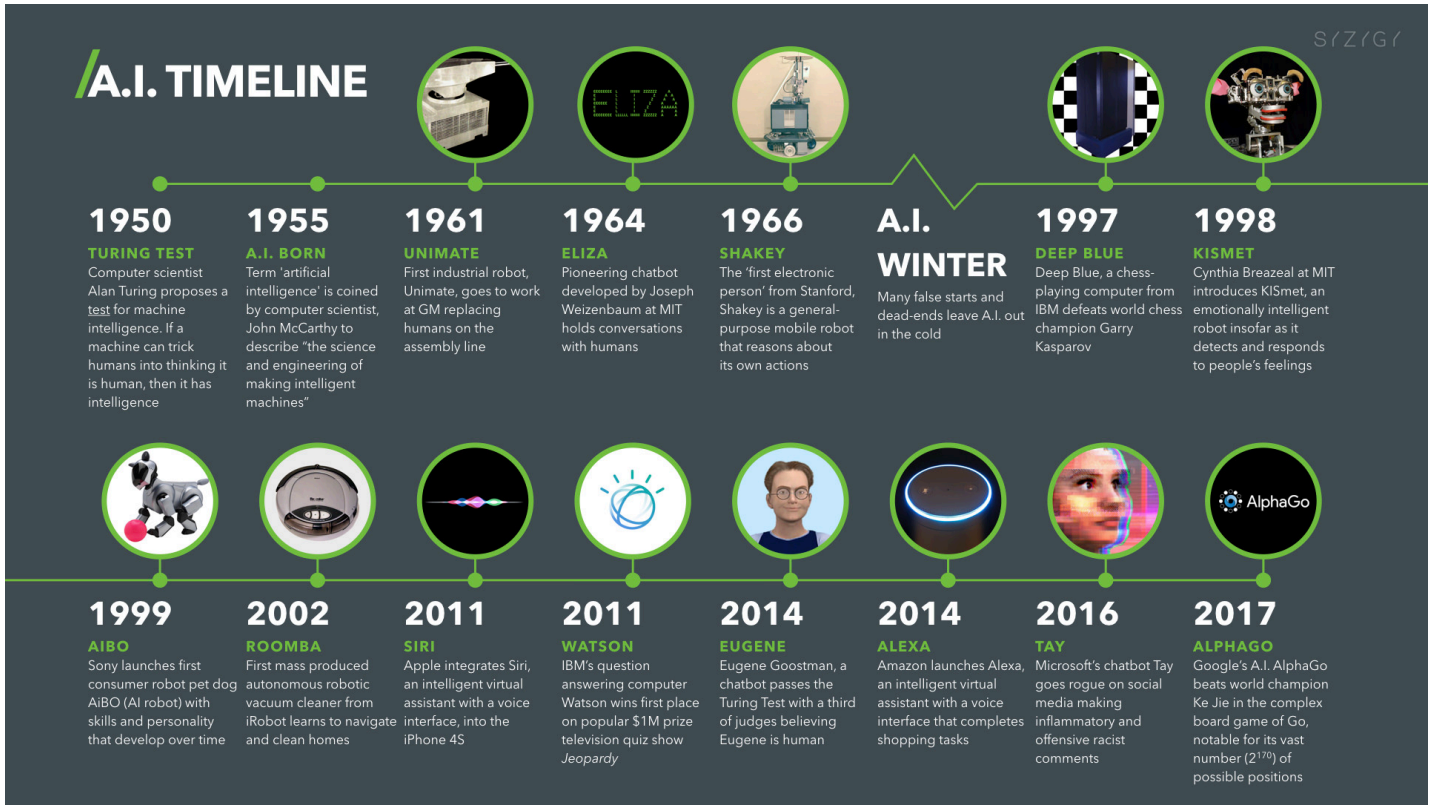
AI at US Open: IBM AI Highlights

- https://www.youtube.com/watch?v=-gRq5rNoWrQ
- Visual recognition, Action detection, Facial emotions recognition
- **Features of input**: hand clapping, player actions, court cheering

Search for Exoplanets 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 Timeline



https://cloudcomputing521.wordpress.com/2017/05/01/history-of-cloud-computing/

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*.”

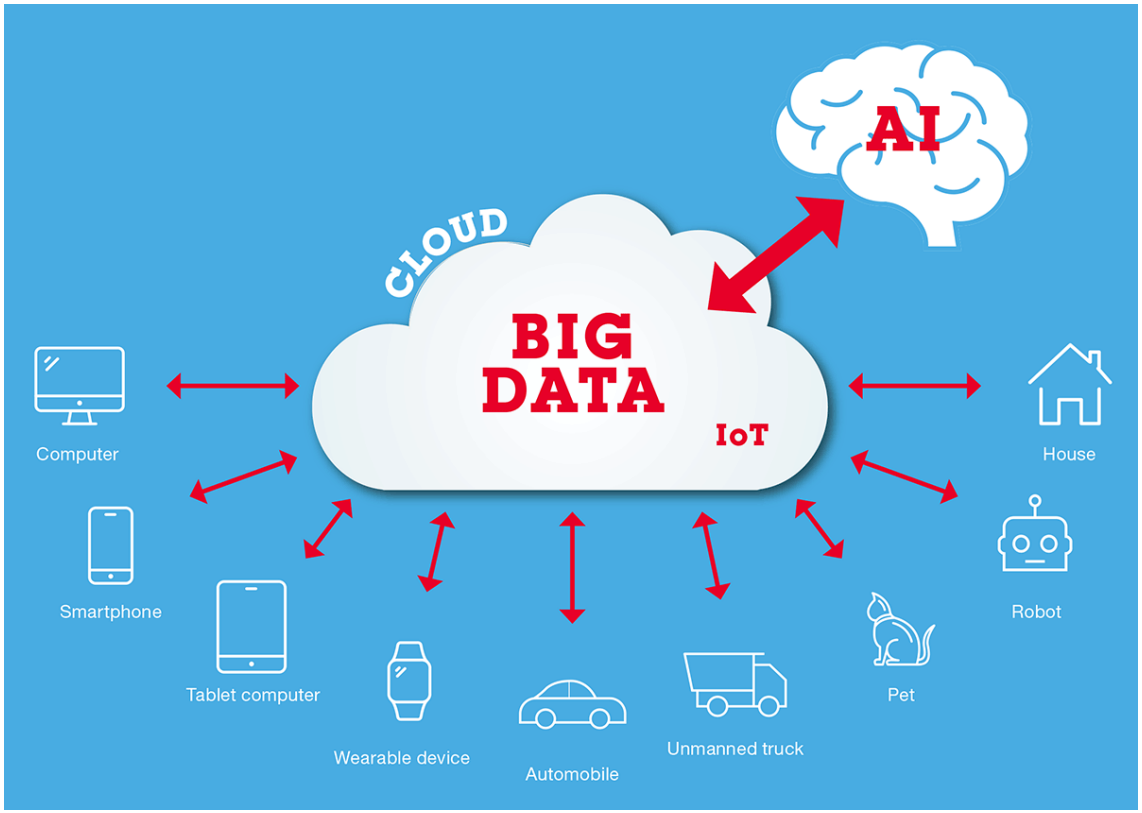
Jensen Huang, NVIDIA, 2017 GTC

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



Yuichi Yoda

Cloud based Machine Learning Services

- IBM Watson Machine Learning

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

- Amazon Sagemaker

https://aws.amazon.com/sagemaker

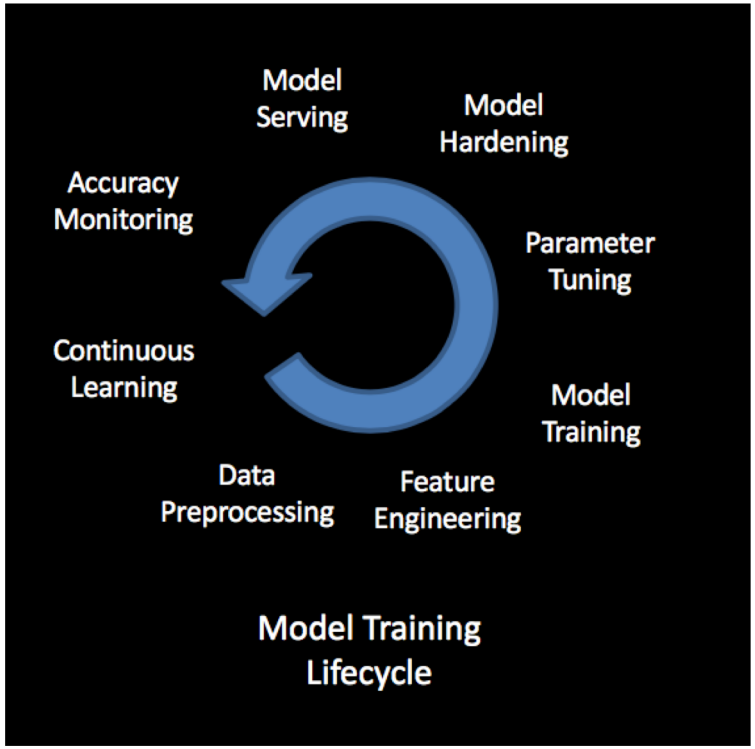
- Microsoft Azure Machine Learning

https://azure.com/ml

- Google Cloud Machine Learning

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

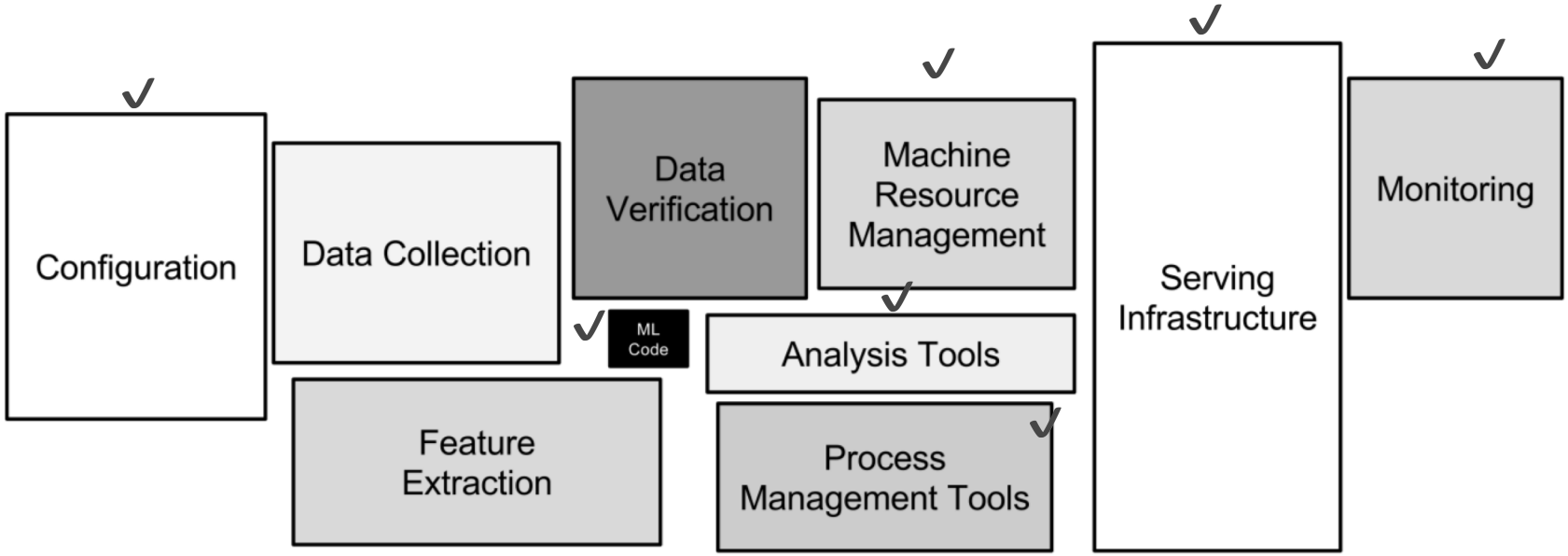
AI Model Training Lifecycle



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



Scully et al Hidden Technical Debt in Machine Learning Systems

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

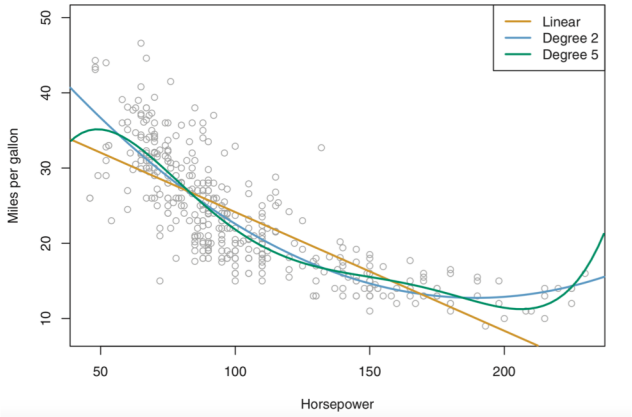
- **Supervised**: learning using labeled training dataset
  - Learning task it to infer a function mapping input to the output using labeled input-output pairs
    - Inferred function correctly predicts output labels for other inputs
  - Example Algorithms
    - **Regression: Linear, Logistic**
    - **Support Vector Machine**
    - **Backpropagation based neural network**
    - Tree based: Decision Trees, Random Forest
  - Quality measures: Mean Square Error, Maximum Likelihood Error
- **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

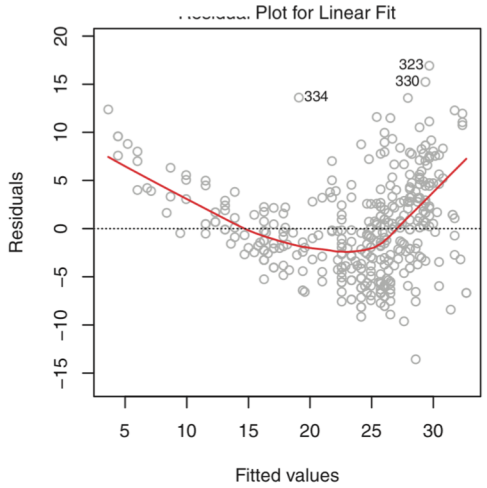
ML Tasks: Classification and Regression

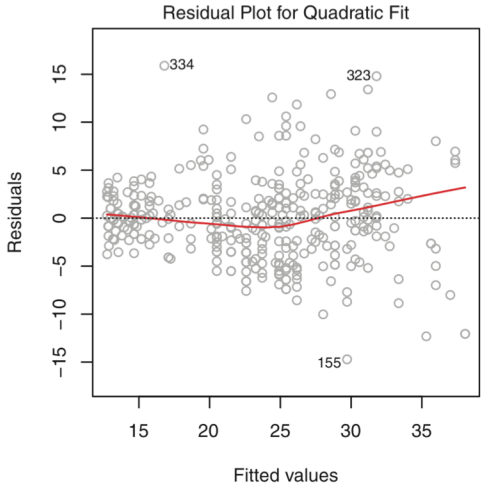
- **Classification**: Output is a categorical, unordered variable
  - Binary classification
    - Tumor: benign vs malignant
    - Food image : Greek vs Non-Greek
  - Multiclass classification
    - Tumor: benign, stage-1, stage-2, stage-3
    - Food image: falafel, salad, pita, …
- **Regression**: Output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

Linear Regression

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







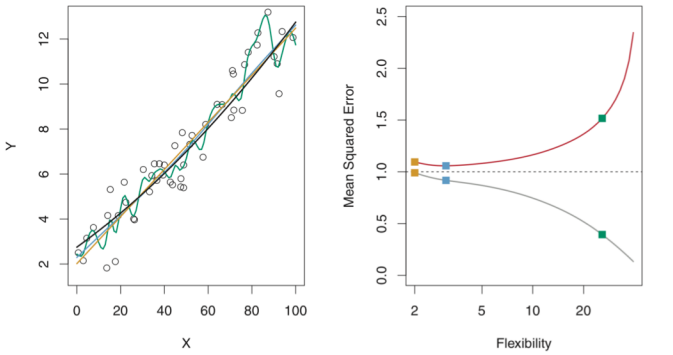
$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \qquad \text{TSS} = \sum (y_i - \bar{y})^2$$

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

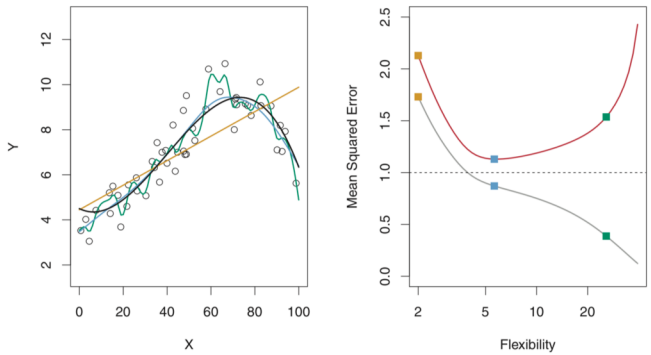
G. James et al, Introduction to Statistical Learning Theory

Mean Square Error (MSE)

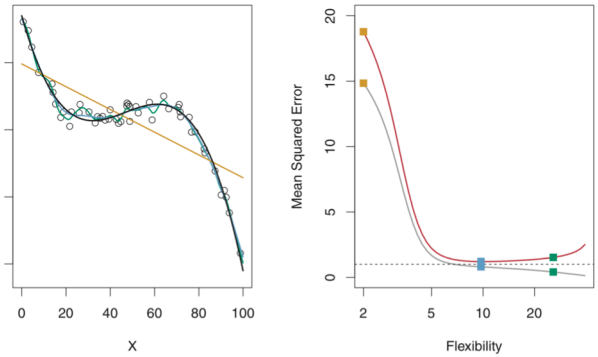
CASE 1



CASE 2



CASE 3



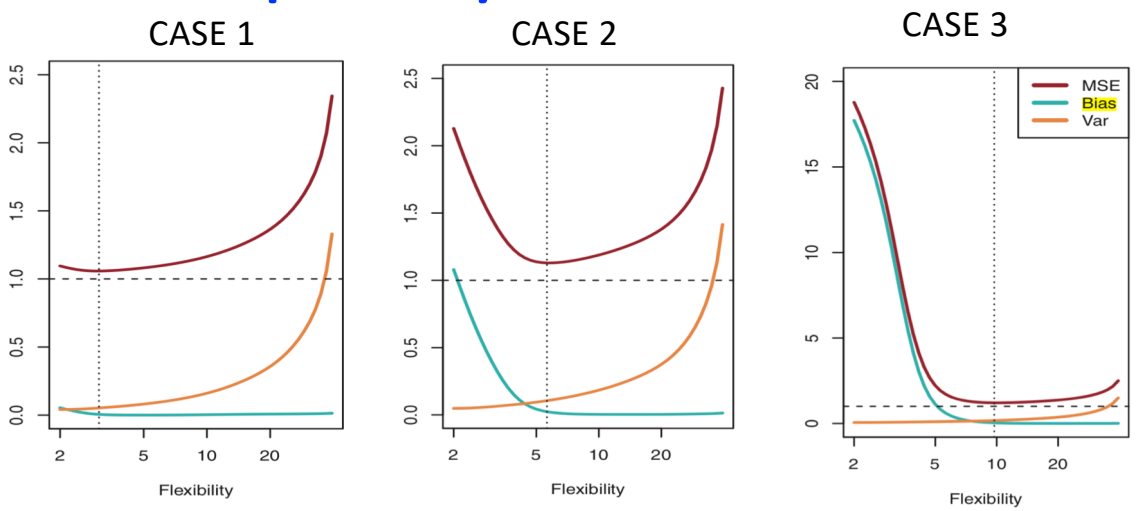
$$Y = f(X) + \epsilon. \qquad \hat{Y} = \hat{f}(X)$$

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

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

G. James et al, Introduction to Statistical Learning Theory

Model Complexity Tradeofffs



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

G. James et al, Introduction to Statistical Learning Theory

Prepare for Lecture 2

- Get access to compute cluster
  - Amazon Educate https://aws.amazon.com/education/awseducate/
  - IBM Academic Initiative https://www.research.ibm.com/university/
  - 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 09/012, due by 09/26
- Look at class website on courseworks for syllabus, announcement and homework

Recommended Books

- The 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/