#io18extended

Machine Learning Project Lifecycle

Deep Neural Networks Training Focus

Google I/O 2018 Extended

LaFactory, Technopark, Casablanca, Morocco

May 12, 2018

Abdelhak Mahmoudi
abdelhak.mahmoudi@um5.ac.ma
linkedIn

Content

- Machine Learning Everywhere
- ML Project Lifecycle
- Deep NN Training Focus
- How can I Learn?
- How can I Apply?
- Demos

Machine Learning Everywhere

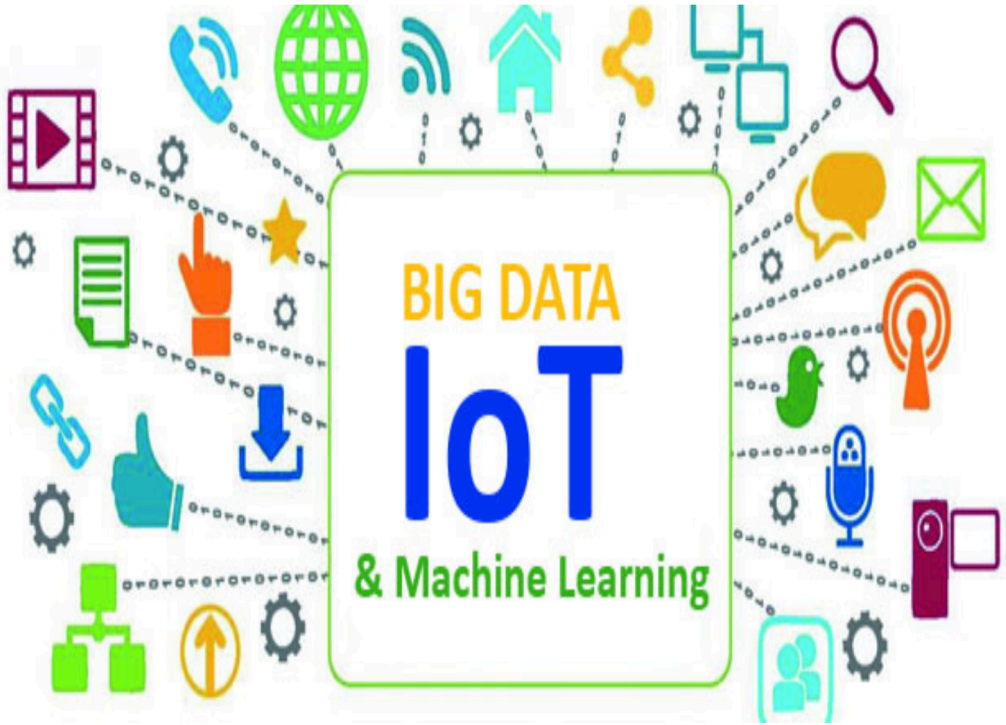
Use Cases

Forbes: The Top 10 AI And Machine Learning Use Cases Everyone Should Know About

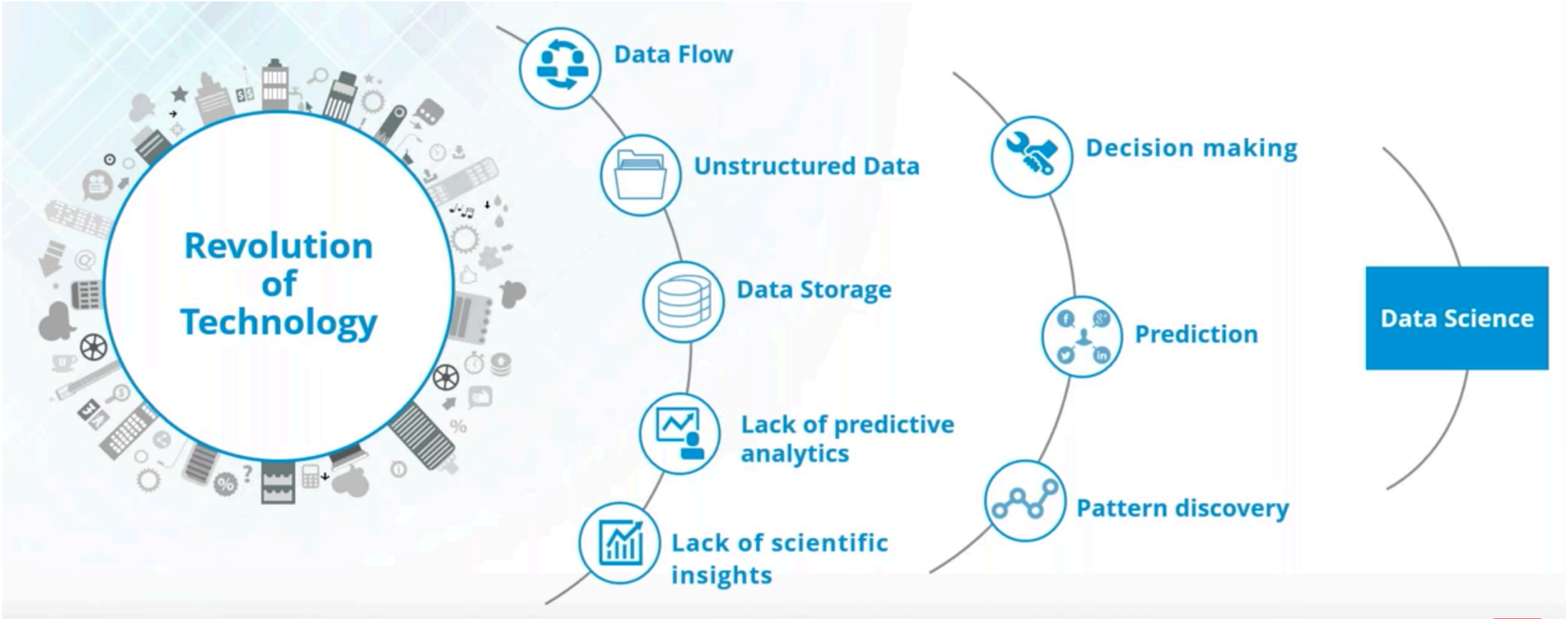
1. Data Security,
2. Personal Security,
3. Financial Trading,
4. Healthcare,
5. Marketing personalization,
6. Fraud Detection,
7. Recommendations,
8. Online Search,
9. NLP,
10. Smart Cars

Internet of Things

- 25 and 50 billion Internet-connected devices By 2020,
- IoT generates Big Data
  - **Volume**
  - **Velocity** in terms of time and location dependency,
  - **Variety** of multiple modalities
  - **Varying** data quality
- Nowadays, Train in the **Cloud** or **Fog**, predict locally
- Tomorrow, Train and predict **locally**

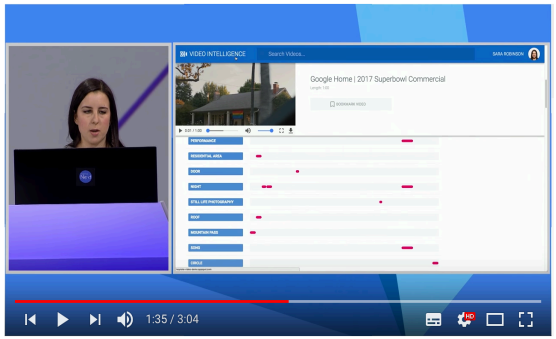


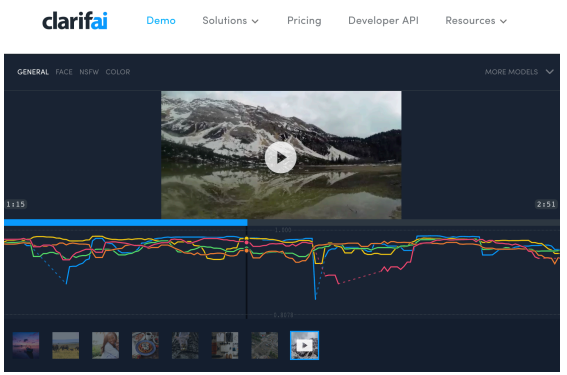
Data Science



Products

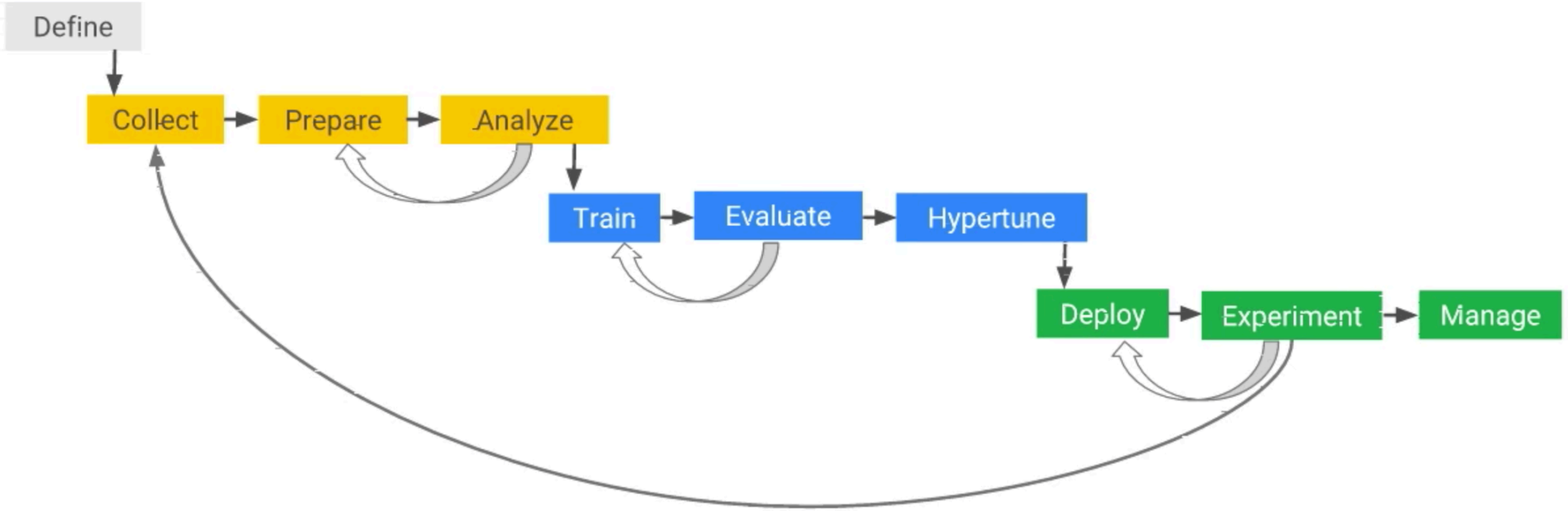
- Text Analysis
  - uclassify.com
- Speech <-> Text Recognition
  - Google Cloud
  - IBM Watson
- Image and Video Classification
  - clarifai.com
- Etc, etc, etc.





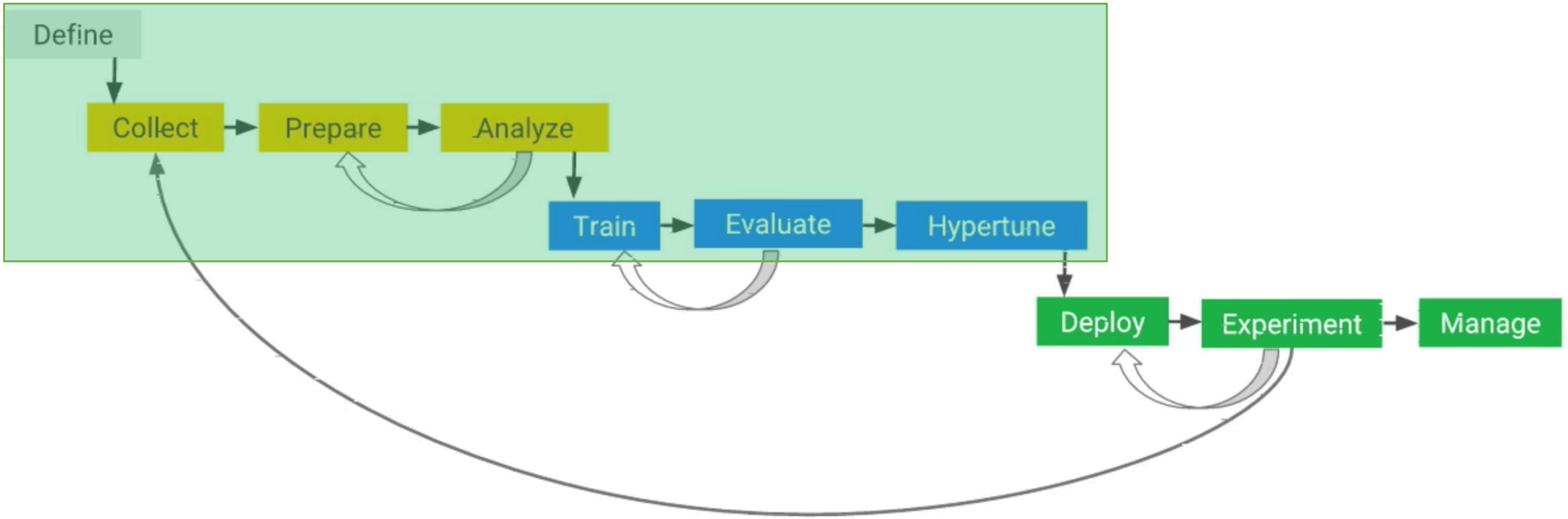
ML Project Lifecycle

ML Project Lifecycle

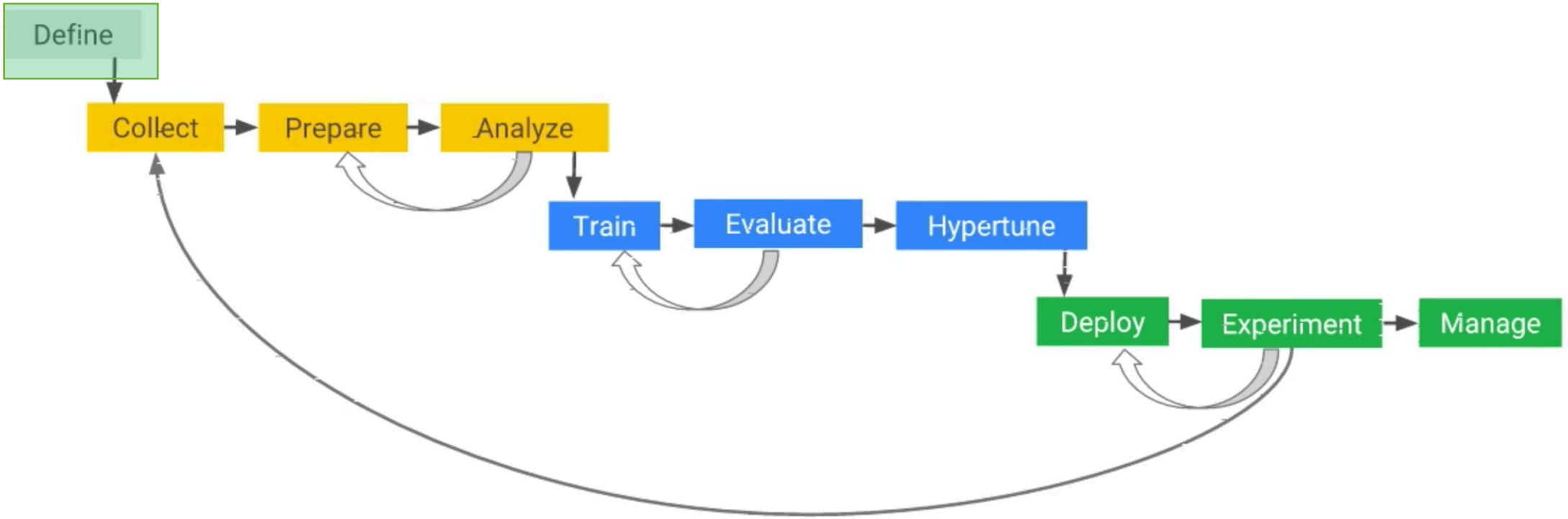


Google Cloud Next '17: https://youtu.be/gkmAnu8DtiM?list=PLIivdWyY5sqI8RuUibiH8sMb1ExIw0lAR

ML Project Lifecycle



ML Project Lifecycle



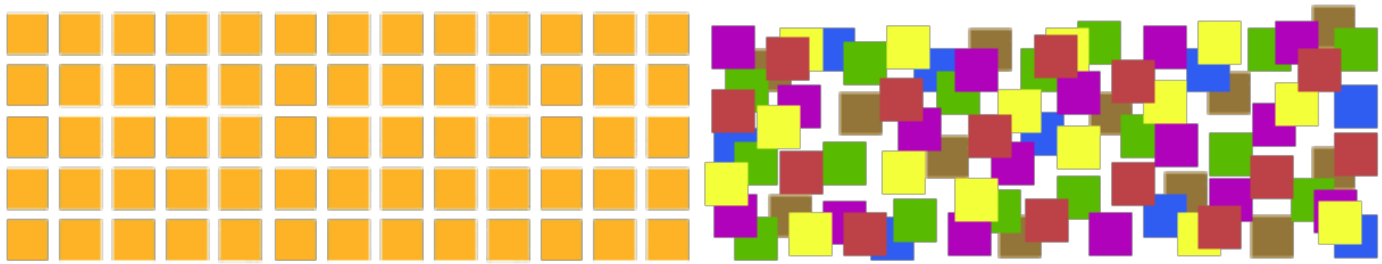
0. Define

- Data
  - Rows/Example/Instance/Input/ /Observation/Record/Point /Sample/Entity : $x^{(i)}$
  - Columns/Feature/Variable/Predictor /Characteristic/Field/Attribute : $x^{(i)}_j$
    - Quantitative (numeric, continue)
    - Qualitative (textual, category)
  - Dimension, Visualization
    - m Examples: i = 1..m
    - n Features: j = 1..n
  - Label/class/output : y_i
    - For each example (0/1)

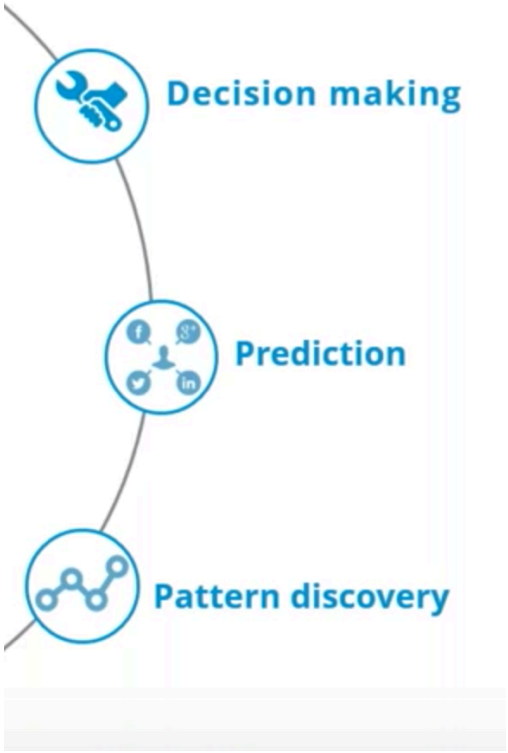
University	F1	F2	F3	F4	F5	F6	Region	Budget (MD)	Insertion (%)	Shanghai ranking?
U1	979	561	186	786	835	536	N	7	72.00	1
U2	895	247	985	206	870	246	N	7	67.00	1
U3	344	889	643	951	783	162	E	2	35.00	0
U4	400	959	999	312	981	254	W	8	57.00	0
U5	243	521	393	596	400	138	E	6	29.00	1
U6	882	722	518	541	425	551	W	10	80.00	1
U7	814	193	829	192	334	597	E	8	68.00	0
U8	186	972	763	968	217	772	S	1	20.00	0
U9	647	656	527	393	738	813	S	7	65.00	1
U10	568	570	458	799	682	530	N	4	59.00	1

0. Define

- Data
  - Structured
    - CSV, XML, JSON, XLSX, etc.
  - Unstructured
    - DOC, HTML, PDF, PNG, MP3, MP4, etc.



0. Define



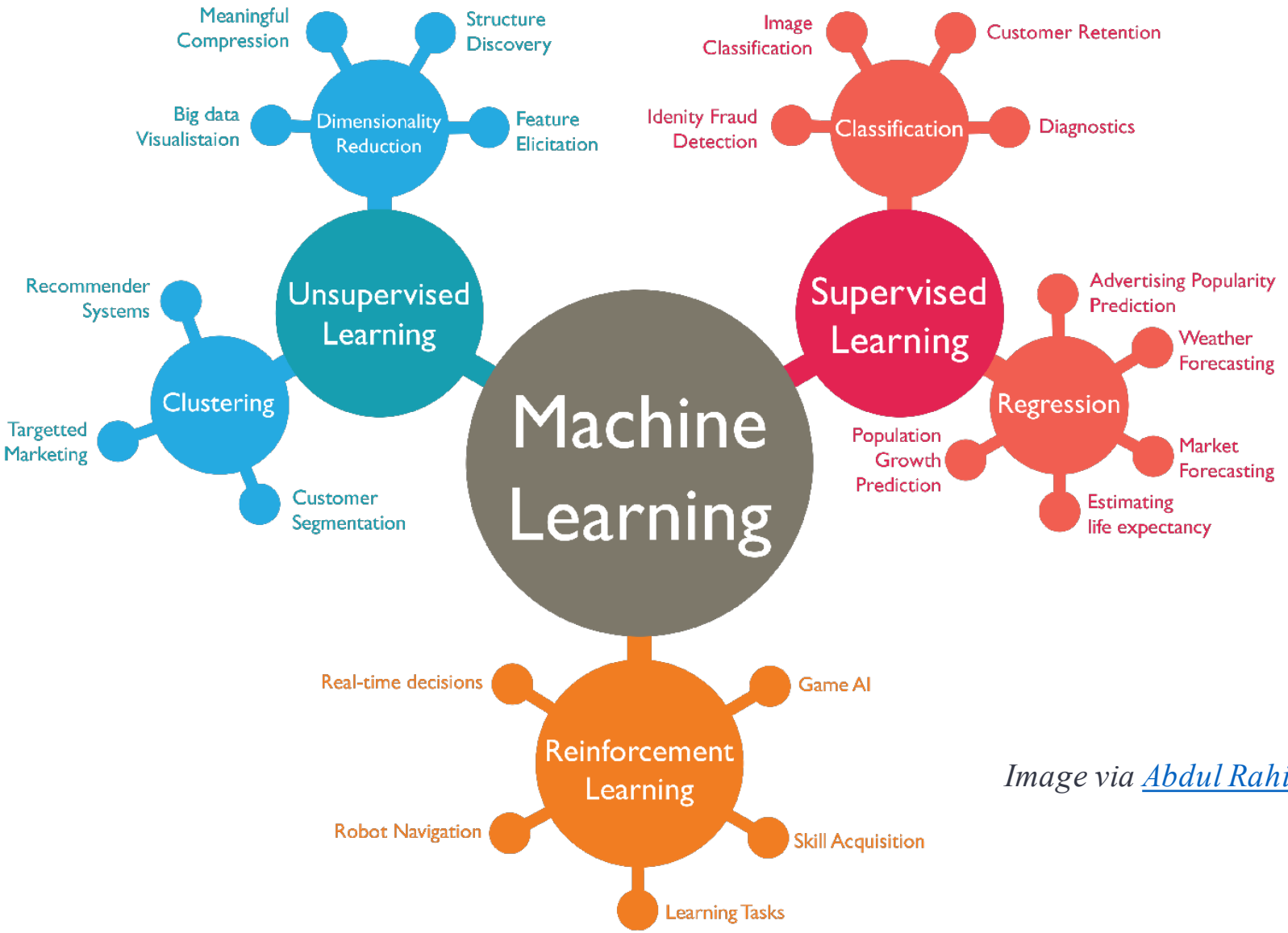
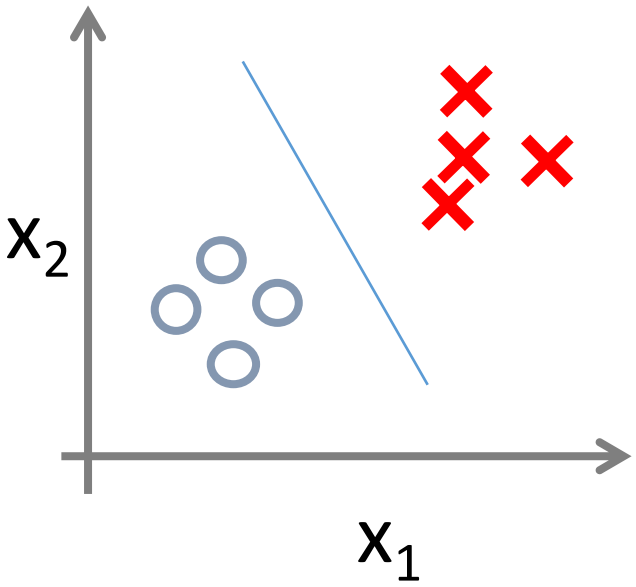


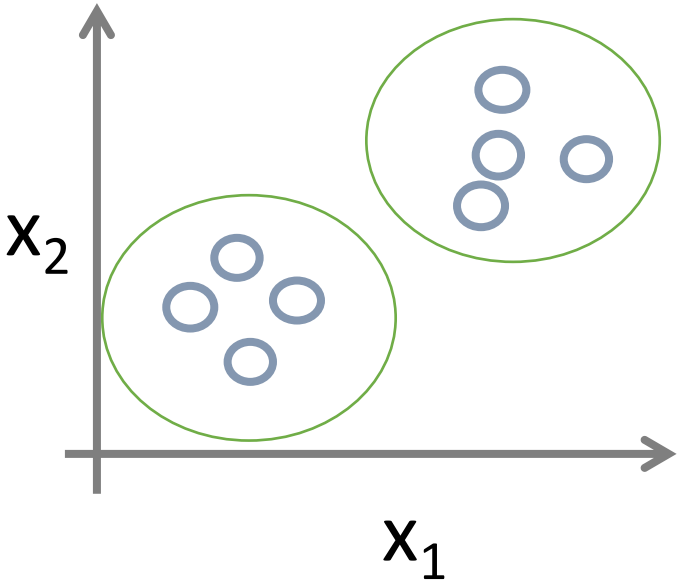
Image via [Abdul Rahid]

0. Define

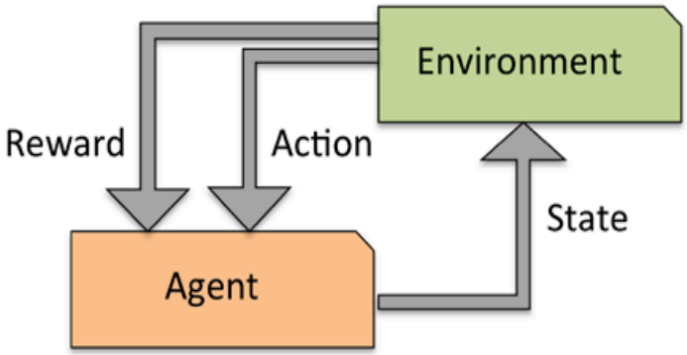
Supervised Learning



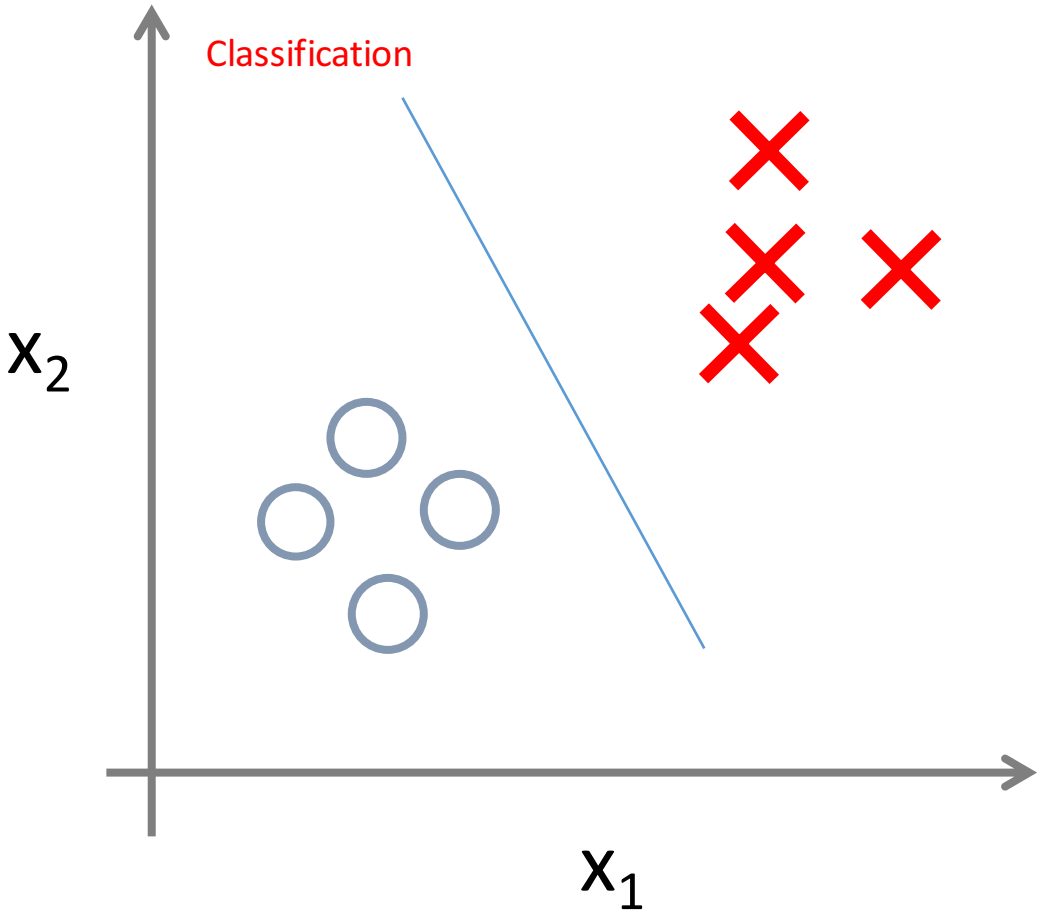
Unsupervised Learning
Clustering

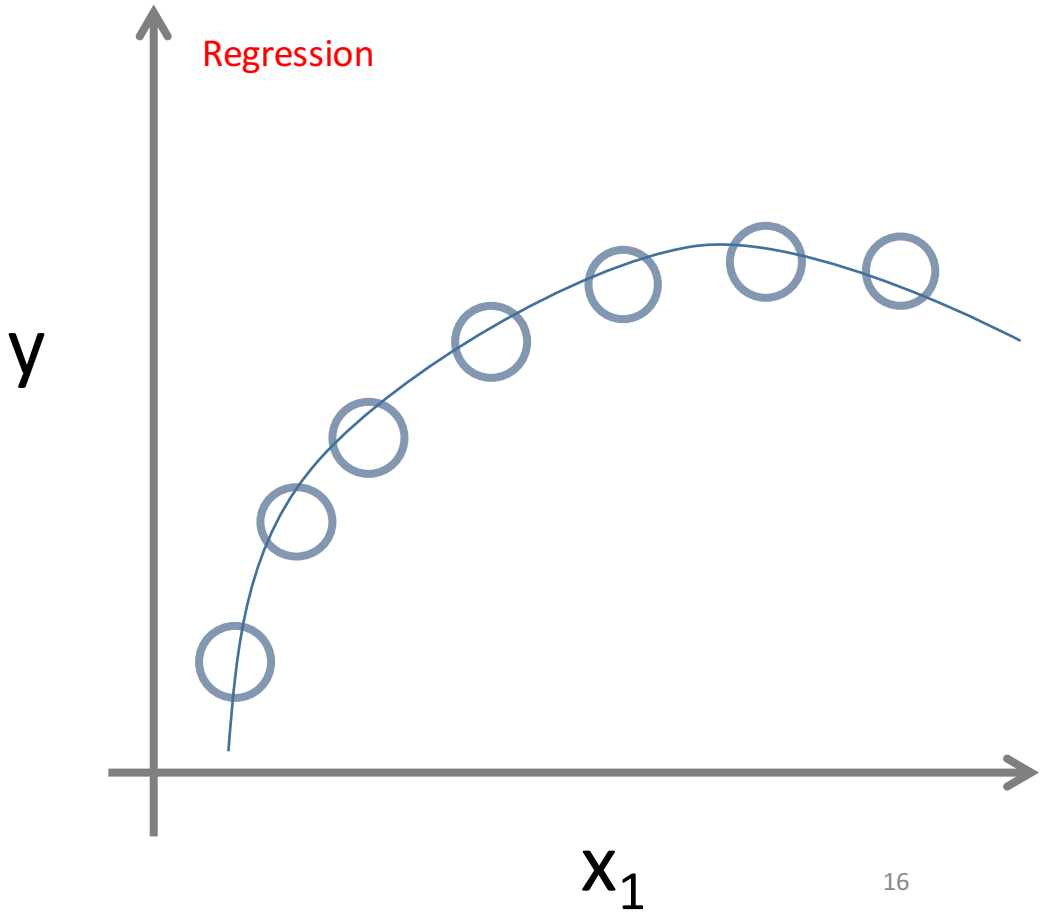


Reinforcement Learning

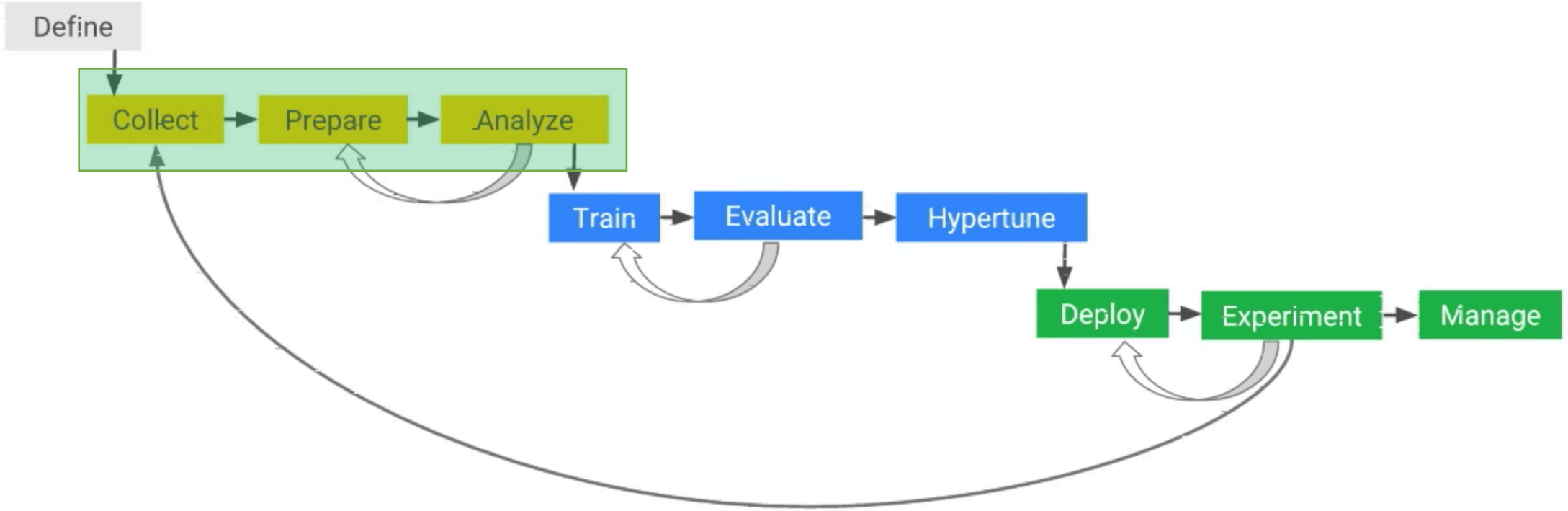


0. Define





ML Project Lifecycle

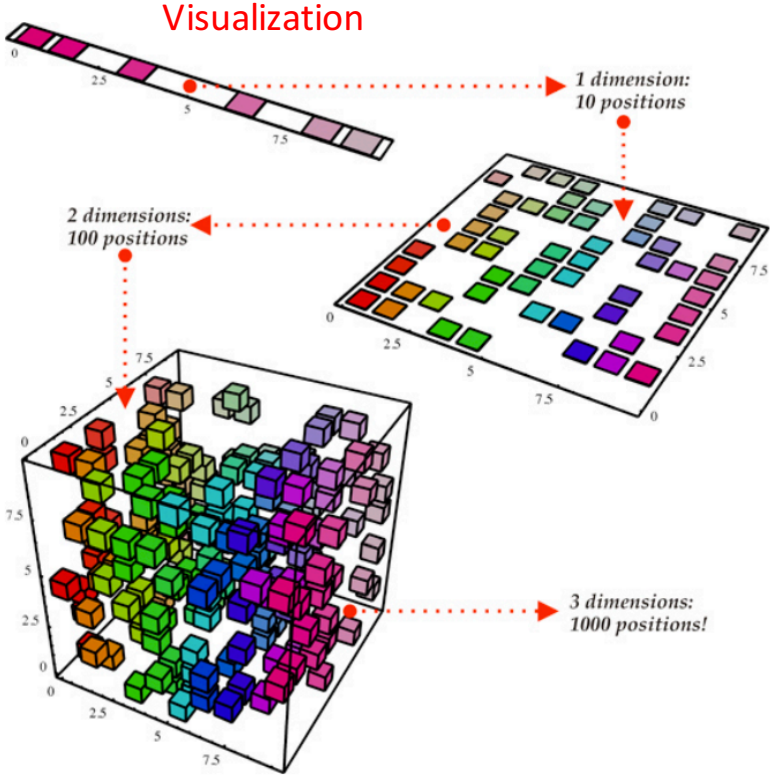


1. Collect

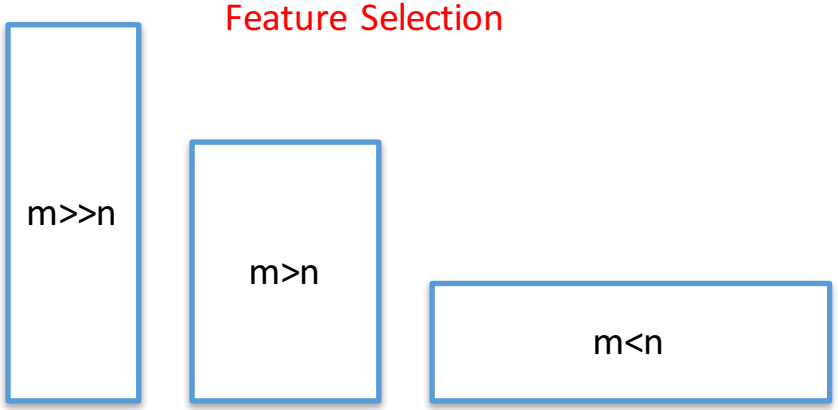
- Government, NGOs, Scientific Communities, etc.
- Open Data Platforms,
- Web data: AWS, Twitter API, Facebook API, Scraping tools, etc.
- Mobile Data: google maps, logs, etc
- Enterprise data: Databases, Datawerhouses
- IoT, Industrial IoT data
- More: Forbes: Big Data: 33 Brilliant And Free Data Sources Anyone Can Use

2. Prepare

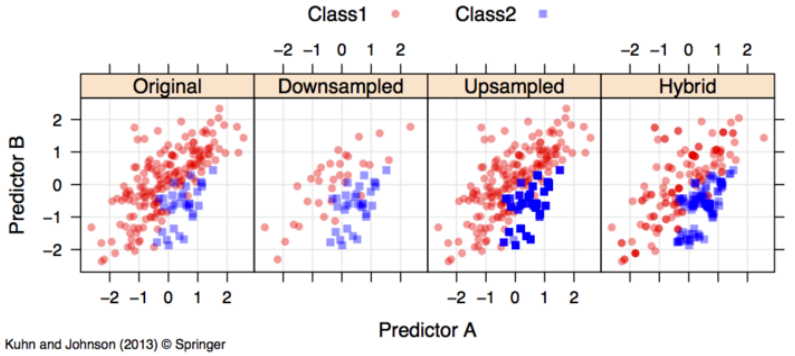
Dimensionality Reduction
Visualization



Feature Extraction
Feature Selection

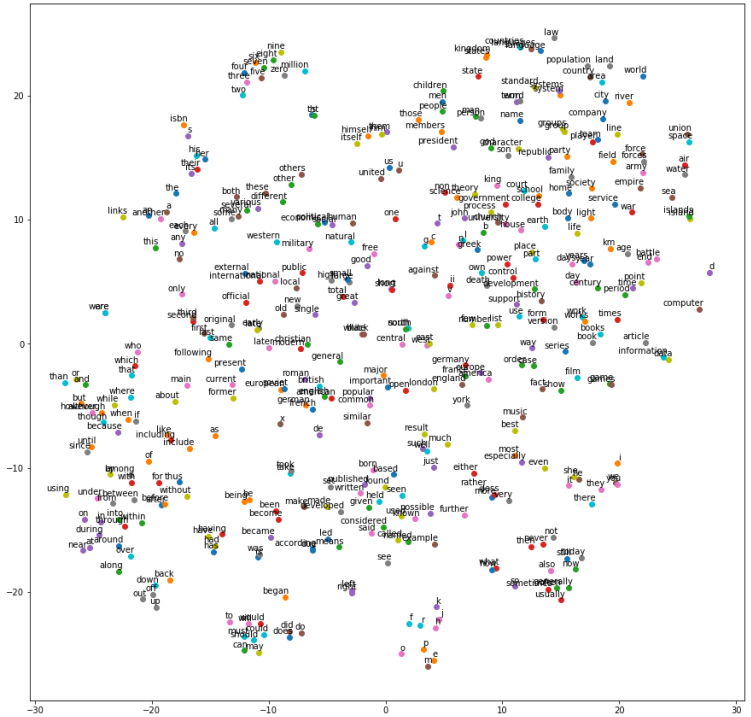


Data Unbalance



Kuhn and Johnson (2013) © Springer

3. Analyze

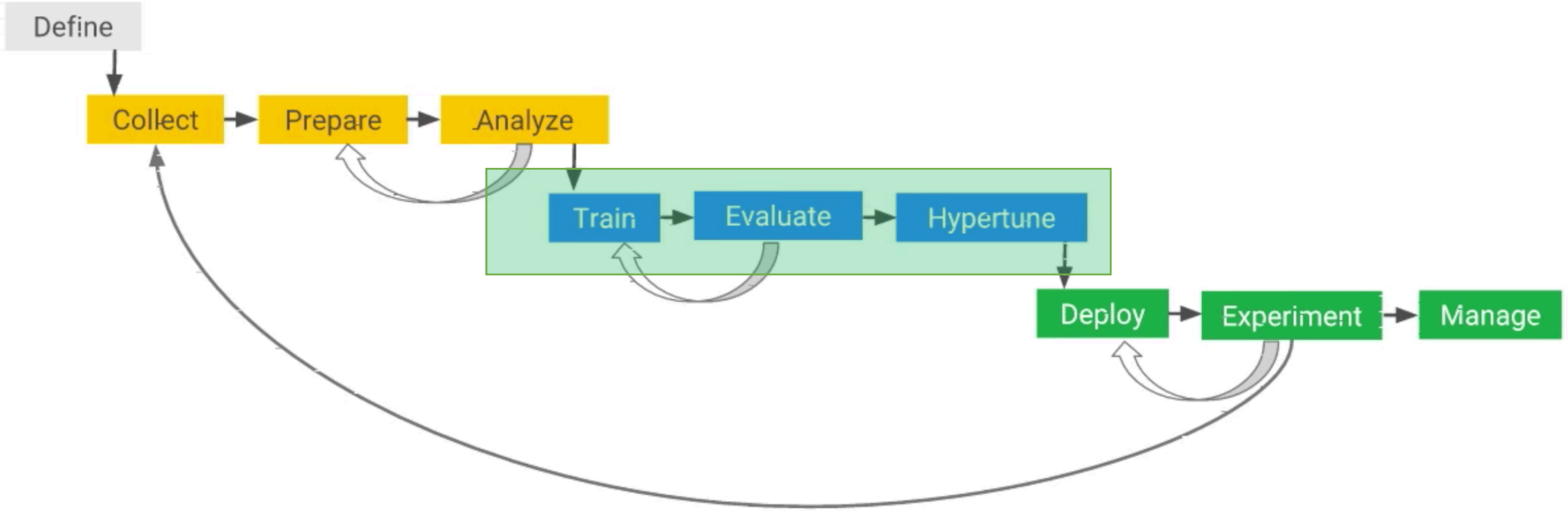


T-SNE

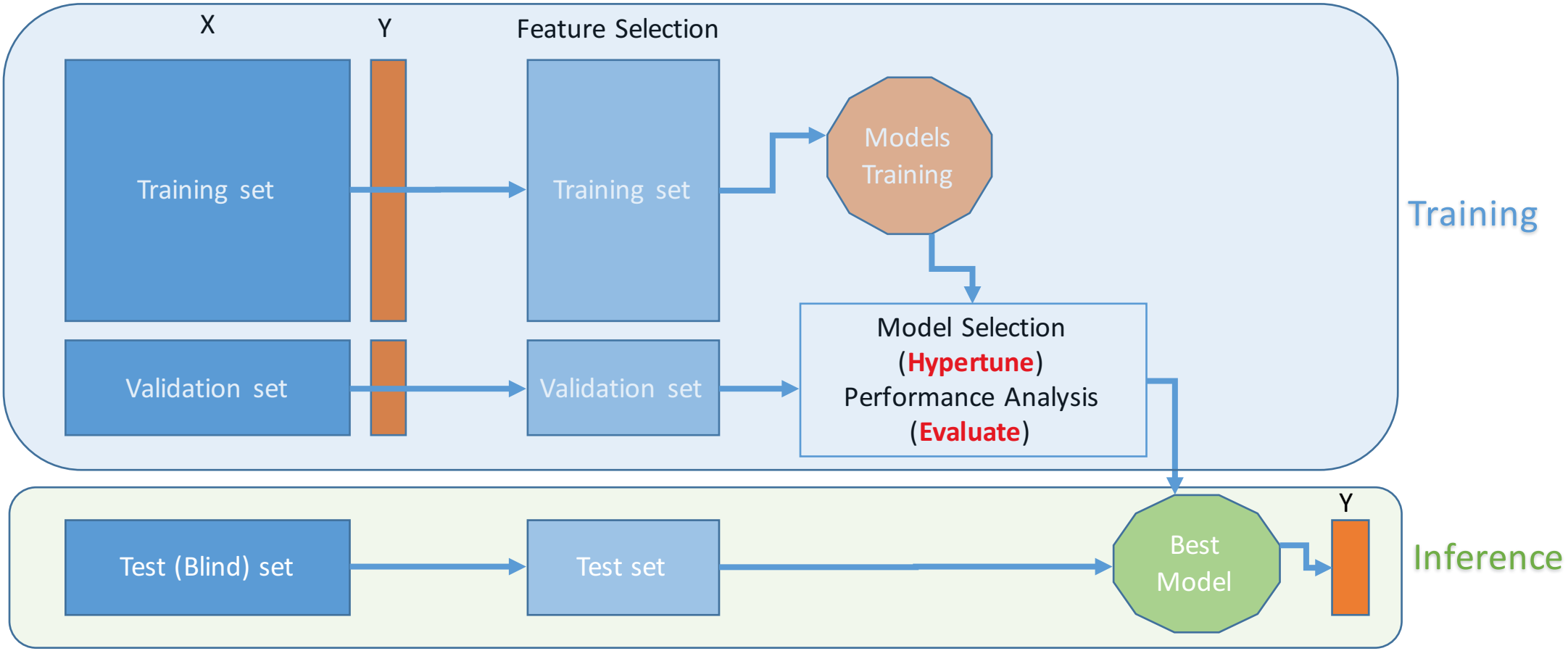
Stats , Transform, Visualize

https://datavizcatalogue.com/index.html

ML Project Lifecycle



4. Train


X
Y
Feature Selection
Training set
Training set
Models Training
Validation set
Validation set
Model Selection
(Hypertune)
Performance Analysis
(Evaluate)
Training
Test (Blind) set
Test set
Best Model
Y
Inference

4. Train


Machine Learning Algorithms
Deep Learning
Deep Boltzmann Machine (DBM)
Deep Belief Networks (DBN)
Convolutional Neural Network (CNN)
Stacked Auto-Encoders
Ensemble
Random Forest
Gradient Boosting Machines (GBM)
Boosting
Bootstrapped Aggregation (Bagging)
AdaBoost
Stacked Generalization (Blending)
Gradient Boosted Regression Trees (GBRT)
Neural Networks
Radial Basis Function Network (RBFN)
Perceptron
Back-Propagation
Hopfield Network
Regularization
Ridge Regression
Least Absolute Shrinkage and Selection Operator (LASSO)
Elastic Net
Least Angle Regression (LARS)
Rule System
Cubist
One Rule (OneR)
Zero Rule (ZeroR)
Repeated Incremental Pruning to Produce Error Reduction (RIPPER)
Regression
Linear Regression
Ordinary Least Squares Regression (OLSR)
Stepwise Regression
Multivariate Adaptive Regression Splines (MARS)
Locally Estimated Scatterplot Smoothing (LOESS)
Logistic Regression
Bayesian
Naive Bayes
Averaged One-Dependence Estimators (AODE)
Bayesian Belief Network (BBN)
Gaussian Naive Bayes
Multinomial Naive Bayes
Bayesian Network (BN)
Decision Tree
Classification and Regression Tree (CART)
Iterative Dichotomiser 3 (ID3)
C4.5
C5.0
Chi-squared Automatic Interaction Detection (CHAID)
Decision Stump
Conditional Decision Trees
M5
Dimensionality Reduction
Principal Component Analysis (PCA)
Partial Least Squares Regression (PLSR)
Sammon Mapping
Multidimensional Scaling (MDS)
Projection Pursuit
Principal Component Regression (PCR)
Partial Least Squares Discriminant Analysis
Mixture Discriminant Analysis (MDA)
Quadratic Discriminant Analysis (QDA)
Regularized Discriminant Analysis (RDA)
Flexible Discriminant Analysis (FDA)
Linear Discriminant Analysis (LDA)
Instance Based
k-Nearest Neighbour (kNN)
Learning Vector Quantization (LVQ)
Self-Organizing Map (SOM)
Locally Weighted Learning (LWL)
Clustering
k-Means
k-Medians
Expectation Maximization
Hierarchical Clustering

4. Train

Hypothesis

$$h_\theta(x) = \theta_0 + \theta_1 x \qquad h_\theta(x) = \theta^T x$$

Cost function

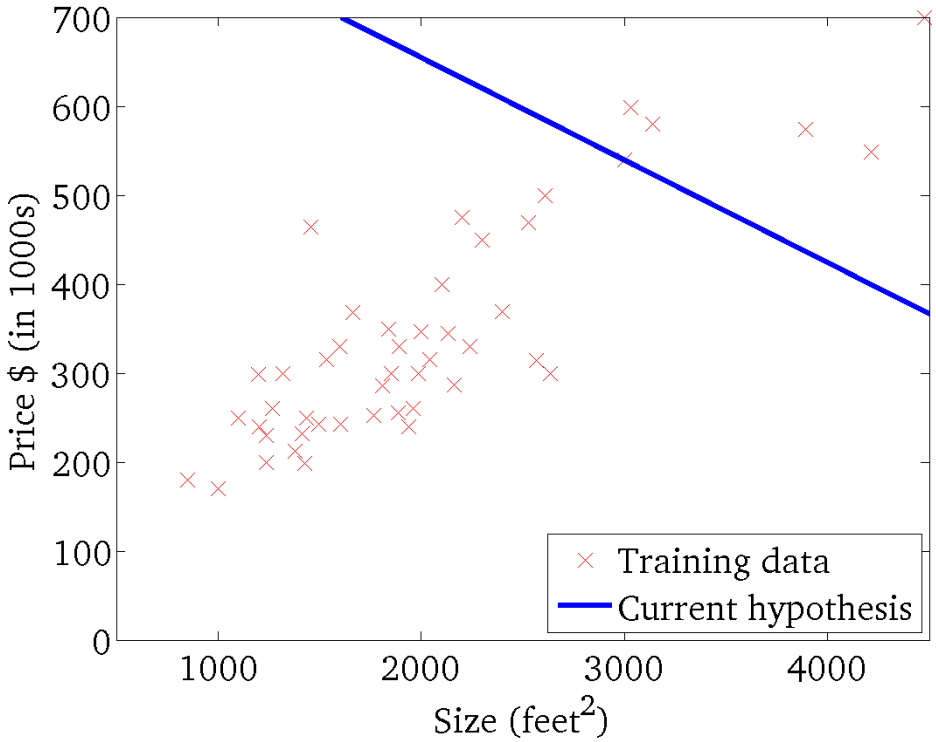
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

Optimization

$$\min_\theta J(\theta)$$

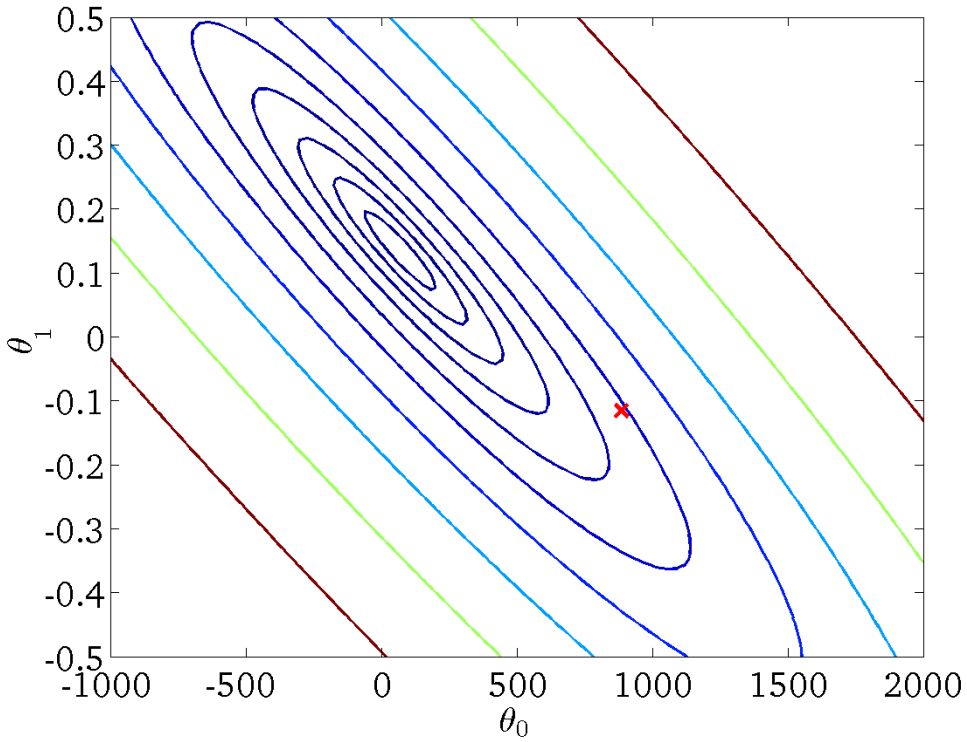
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



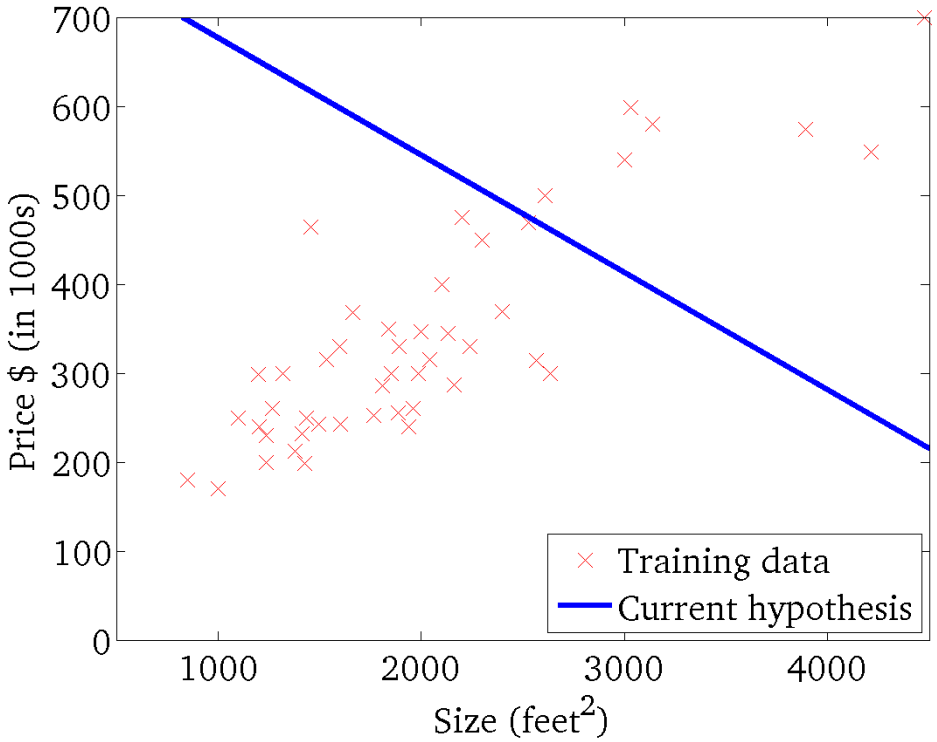
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



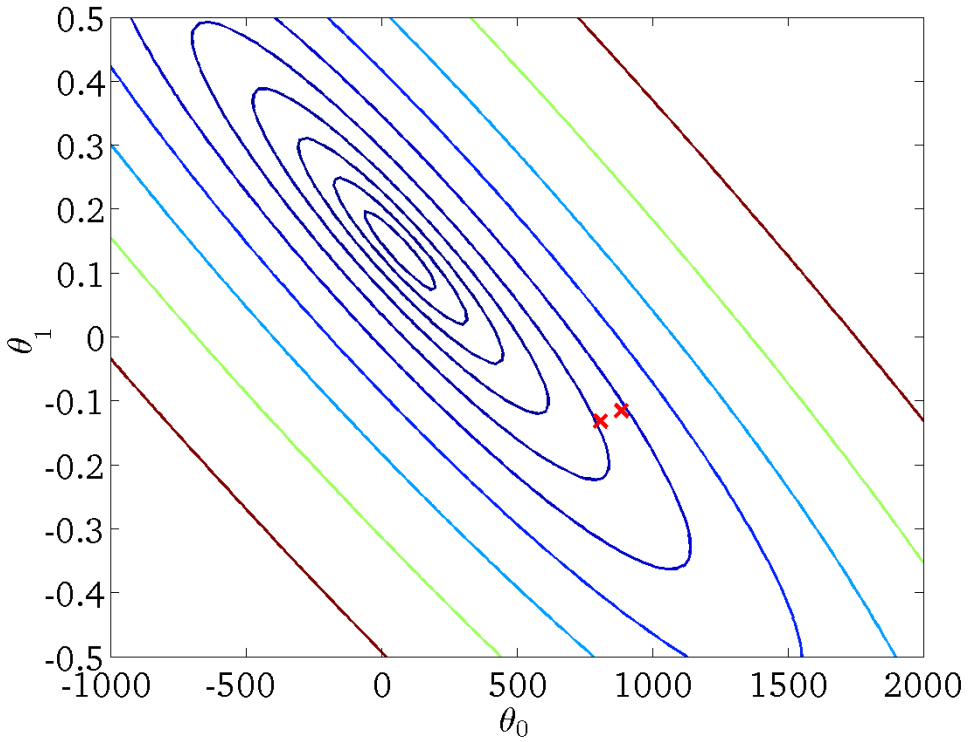
$h_\theta(x)$

(for fixed θ_0, θ_1 this is a function of x)



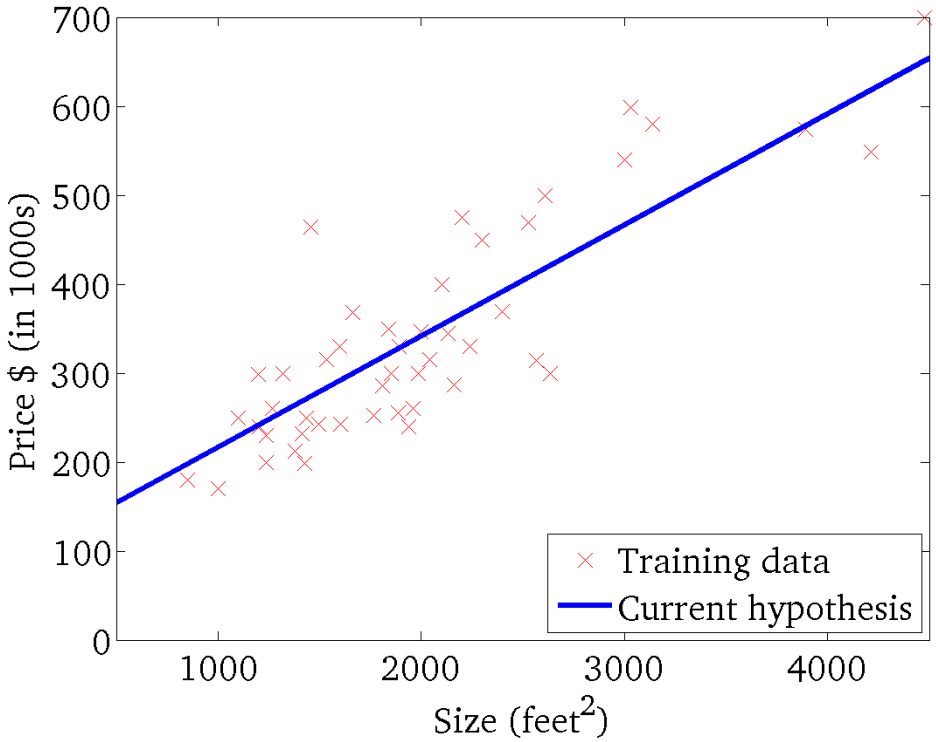
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



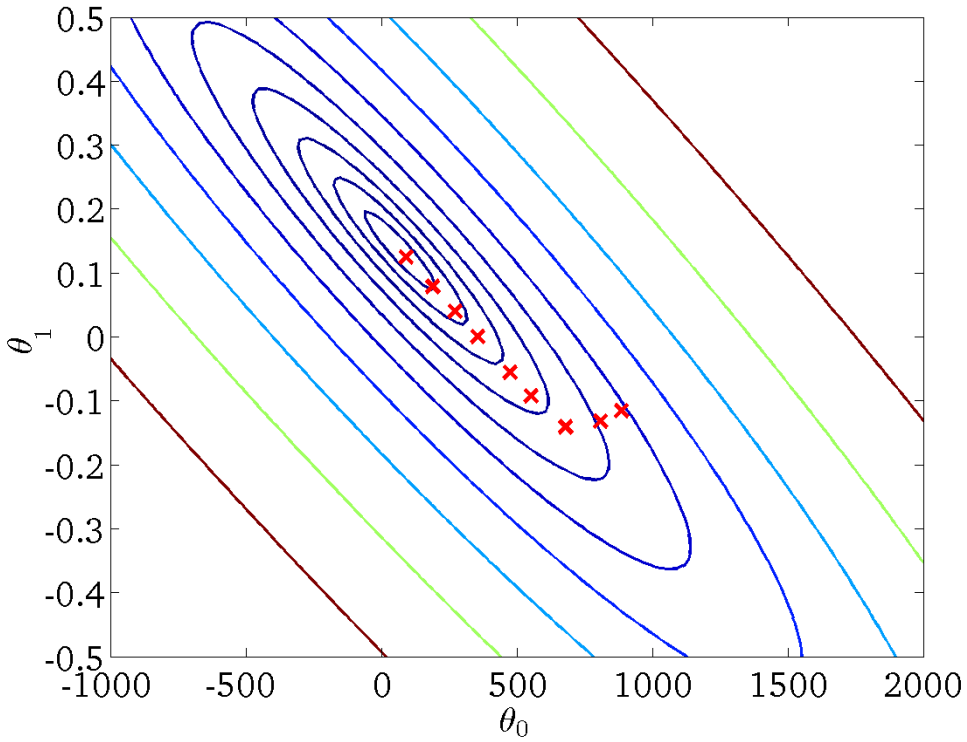
$h_\theta(x)$

(for fixed θ_0, θ_1 this is a function of x)



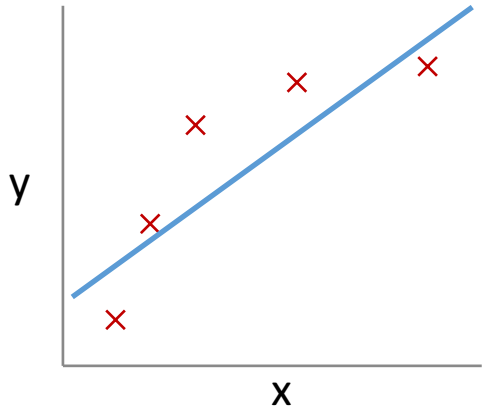
$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)



4. Train

Linear Regression



$$h_\theta(x) = \theta^T x$$

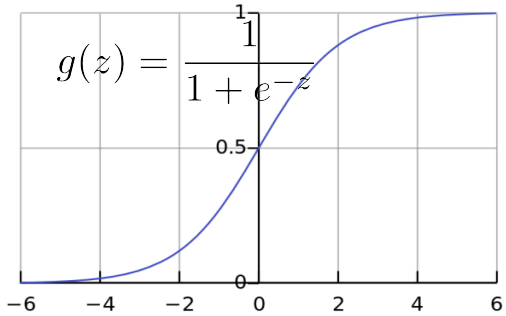
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

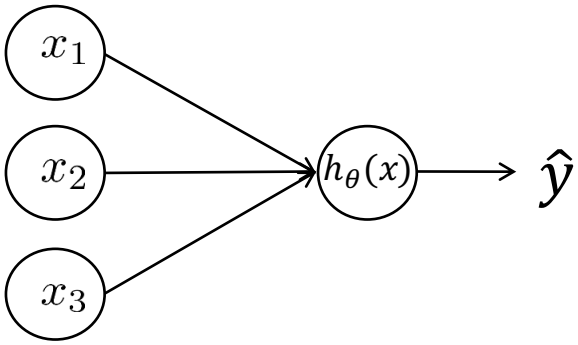
$$\min_\theta J(\theta)$$

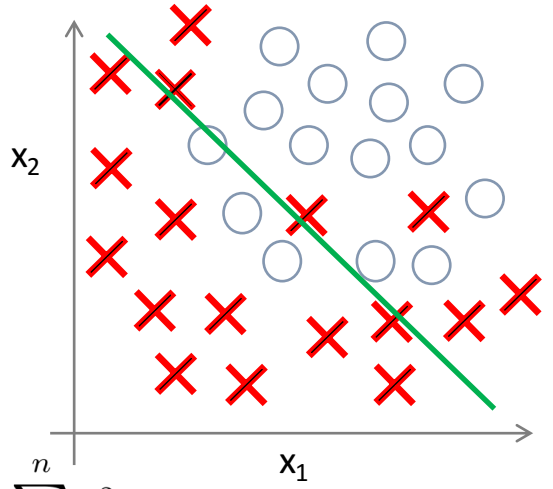
Regularization term

4. Train

Logistic Regression







Want $0 \leq h_\theta(x) \leq 1$

$$h_\theta(x) = g(\ \theta^T x\ )$$

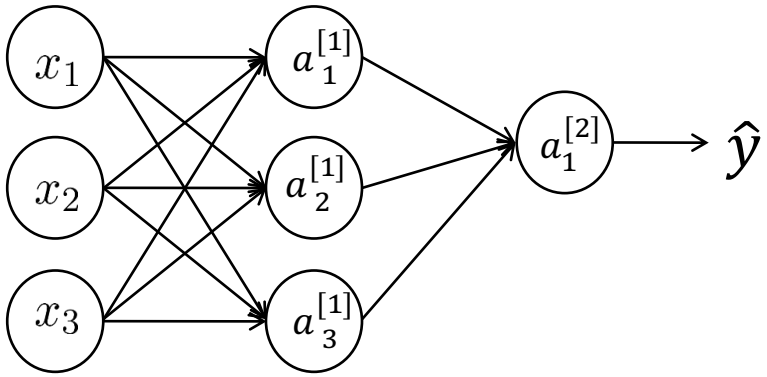
$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2$$

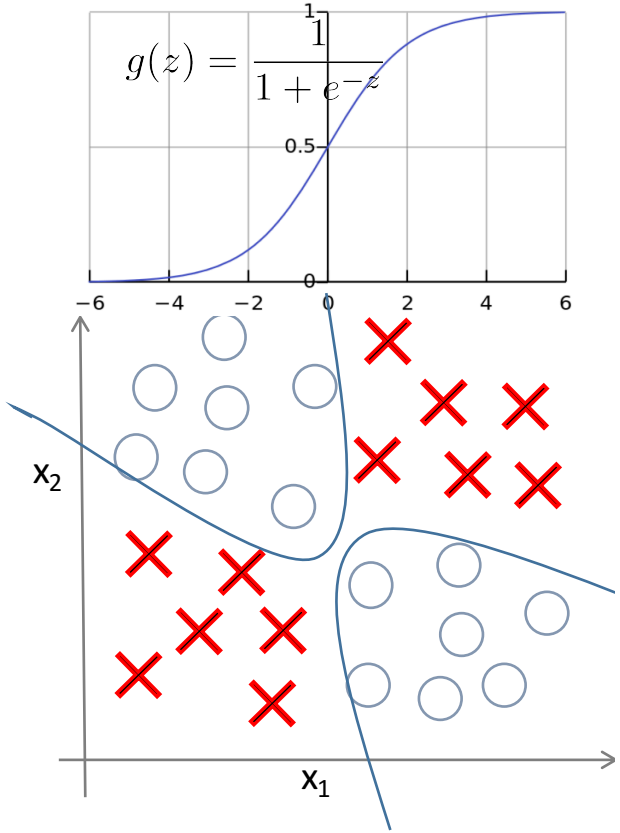
Regularization term

$$\min_\theta J(\theta)$$

4. Train

ANN





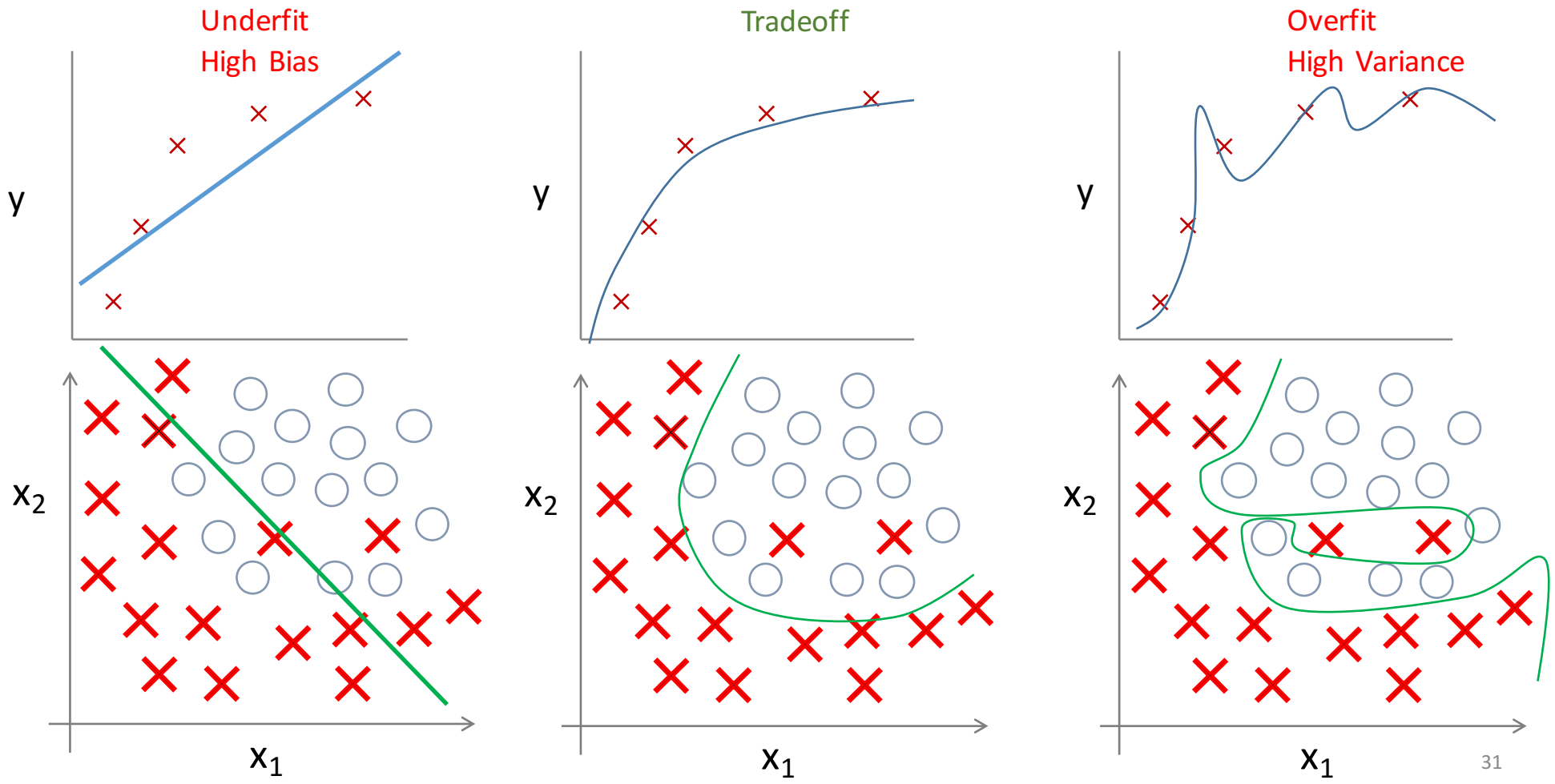
$$z_1^{[1]} = w_1^{[1]T} x + b_1^{[1]},\ a_1^{[1]} = g(z_1^{[1]})$$
$$z_2^{[1]} = w_2^{[1]T} x + b_2^{[1]},\ a_2^{[1]} = g(z_2^{[1]})$$
$$z_3^{[1]} = w_3^{[1]T} x + b_3^{[1]},\ a_3^{[1]} = g(z_3^{[1]})$$
$$z_1^{[2]} = w_1^{[2]T} x + b_1^{[2]},\ a_1^{[2]} = g(z_1^{[2]})$$

$$J = -\frac{1}{m}\sum_{i=0}^{m}\left(y^{(i)}\log\left(a^{[2](i)}\right) + (1-y^{(i)})\log\left(1-a^{[2](i)}\right)\right) + \frac{\lambda}{2m}\sum_{j=1}^{n}\sum_{l=1}^{L} {W_j^l}^2$$

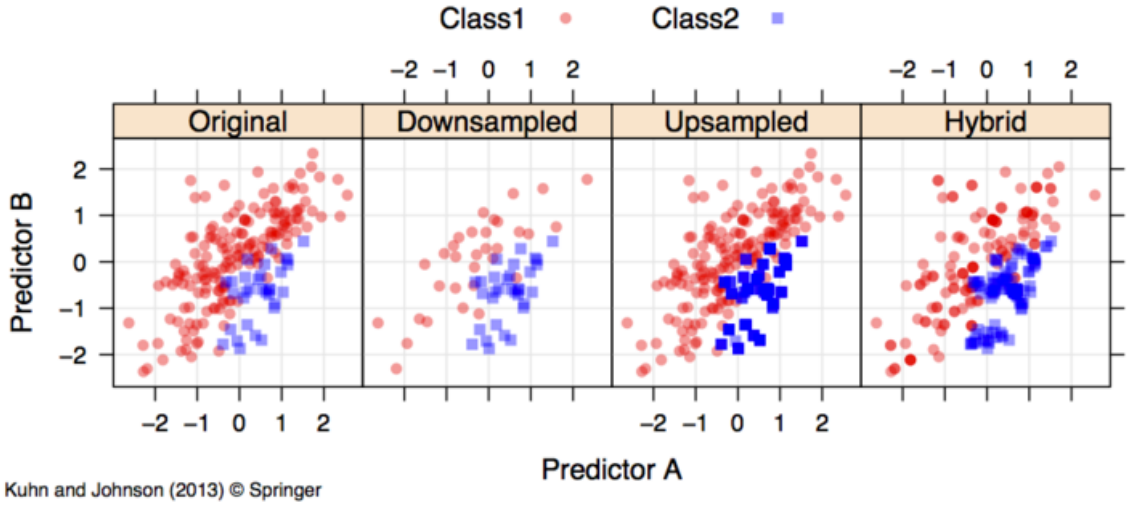
Regularization term

$$\min_{W,b} J(W,b)$$

5. Evaluate

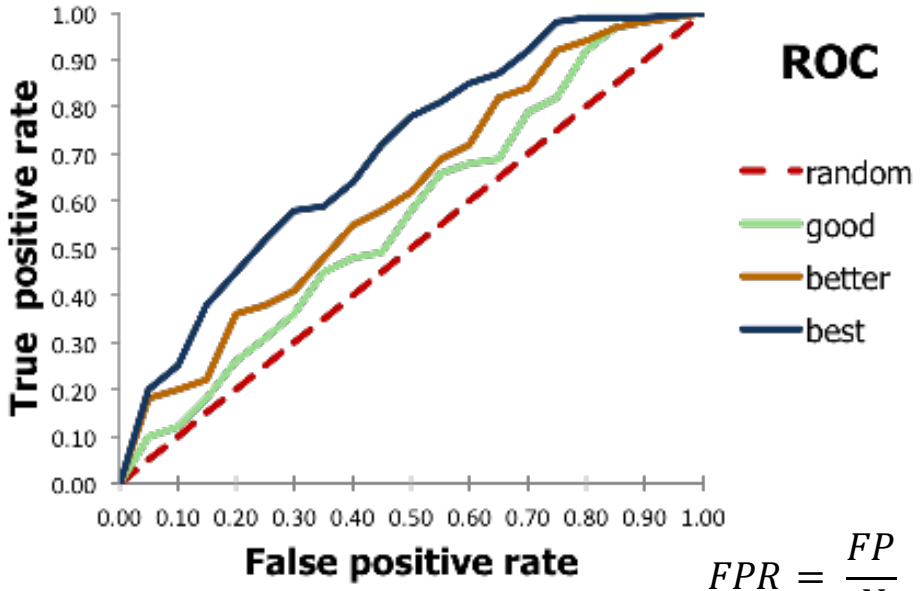

Underfit
High Bias
y
x_2
x_1
Tradeoff
y
x_2
x_1
Overfit
High Variance
y
x_2
x_1

5. Evaluate



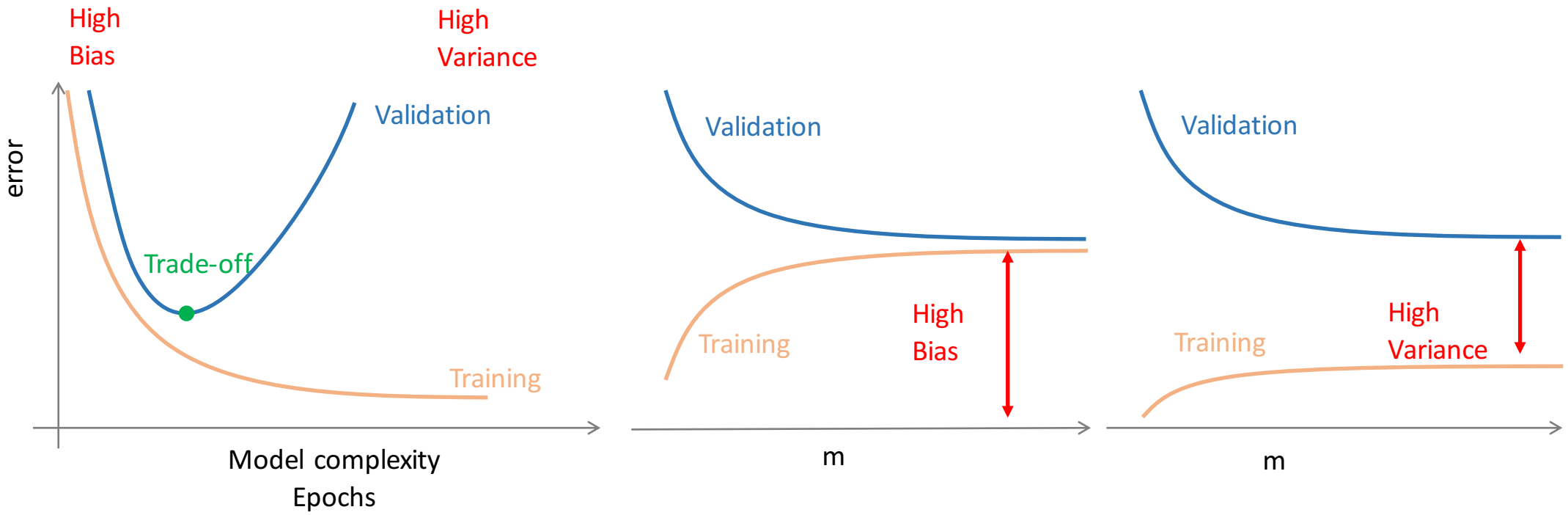
Kuhn and Johnson (2013) © Springer

$$TPR = \frac{TP}{P}$$

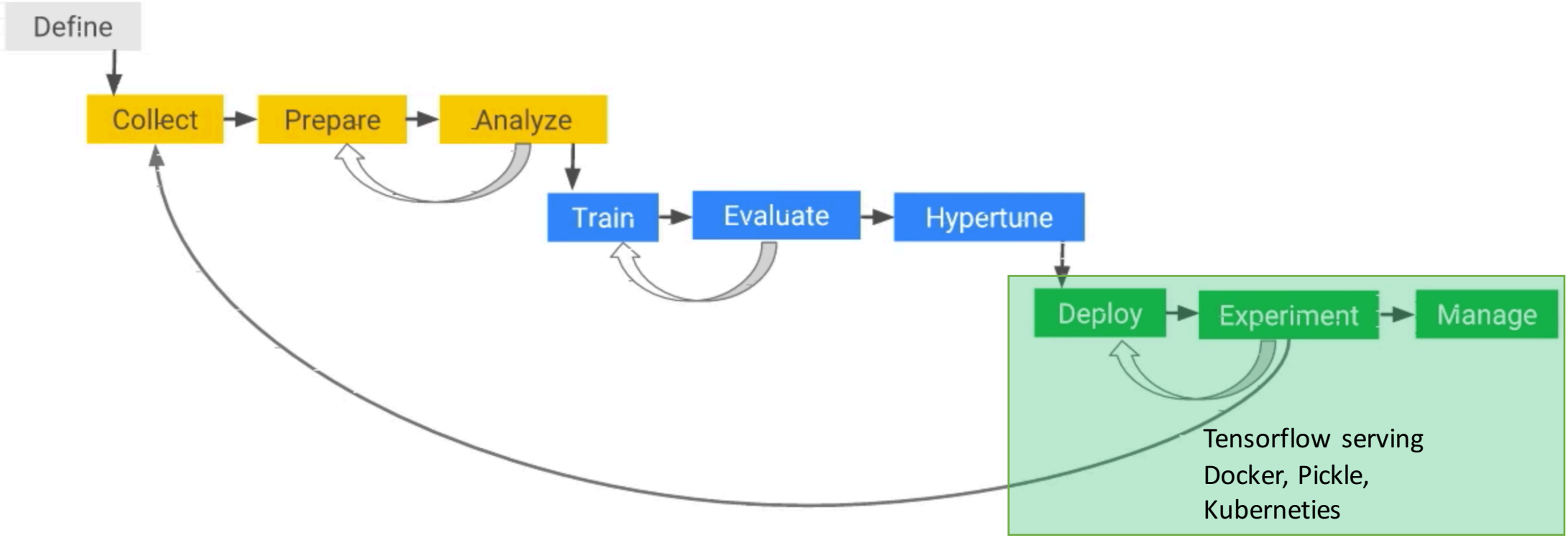


$$FPR = \frac{FP}{N}$$

7. Hypertune

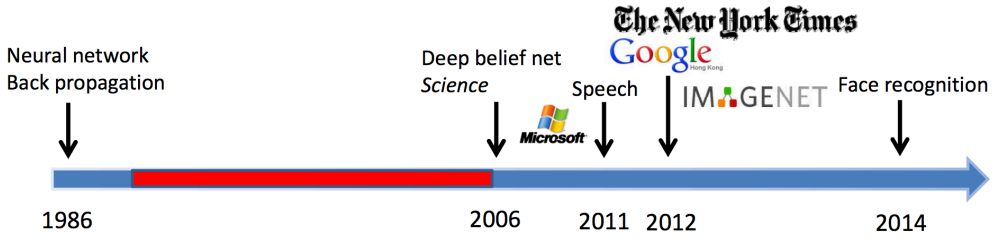

High
Bias
High
Variance
Validation
error
Trade-off
Training
Model complexity
Epochs
Validation
Training
High
Bias
m
Validation
Training
High
Variance
m

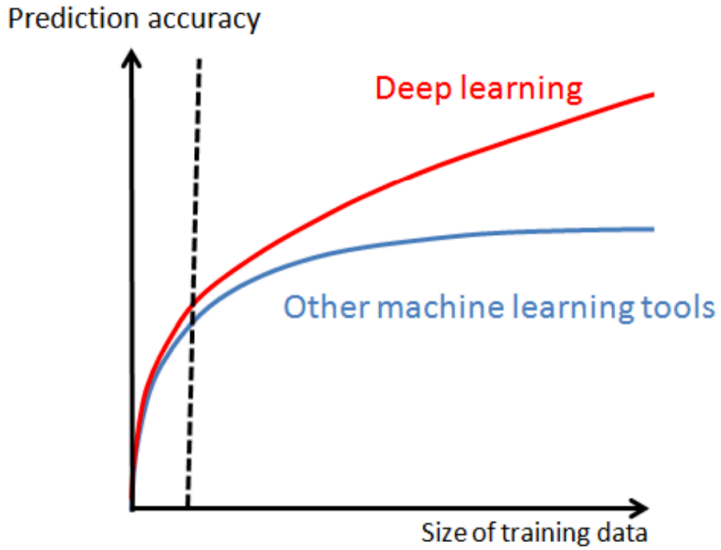
8. Deploy, 9. Experiment, 10. Manage

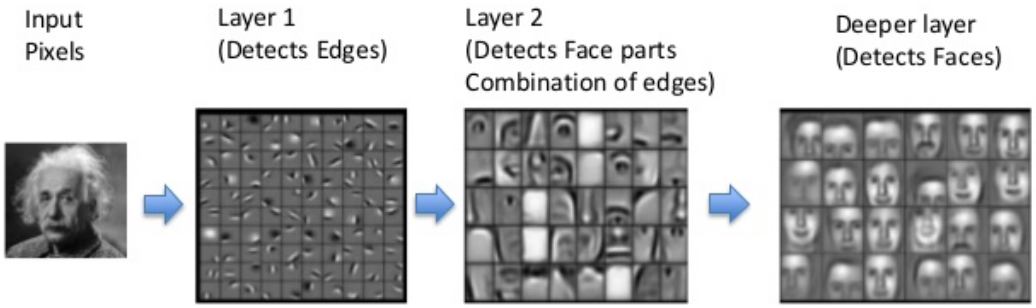


Deep NN Training Focus

Why Deep Learning?

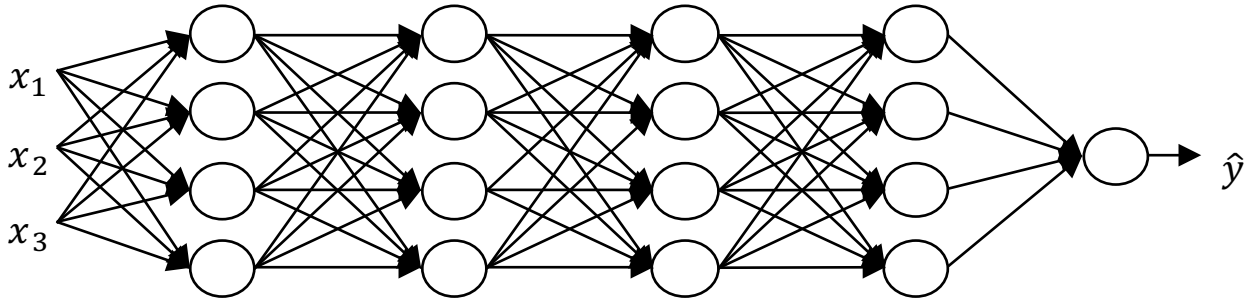






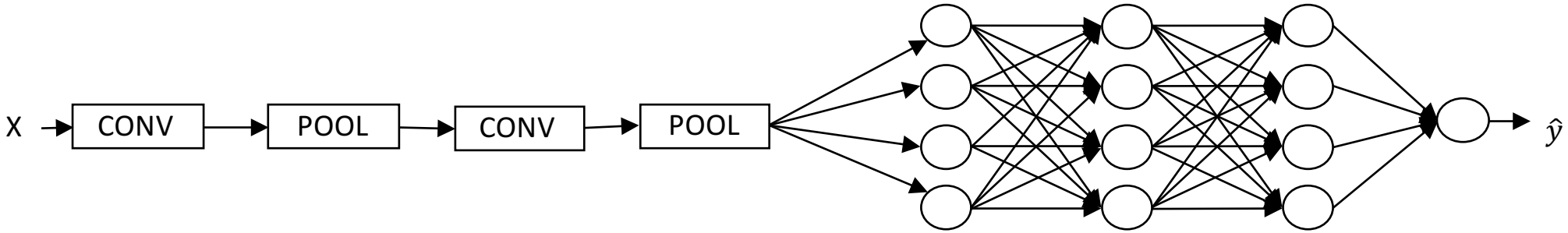
Deep NN main architectures

- Feed Forward NN
  - One hidden layer
- Deep NN
  - Multiple hidden layer



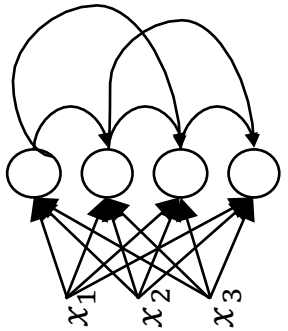
Deep NN main architectures

- Convolutional NN
  - Computer Vision
  - LeNet-5, AlexNet, VGG-16, ResNet, Inception

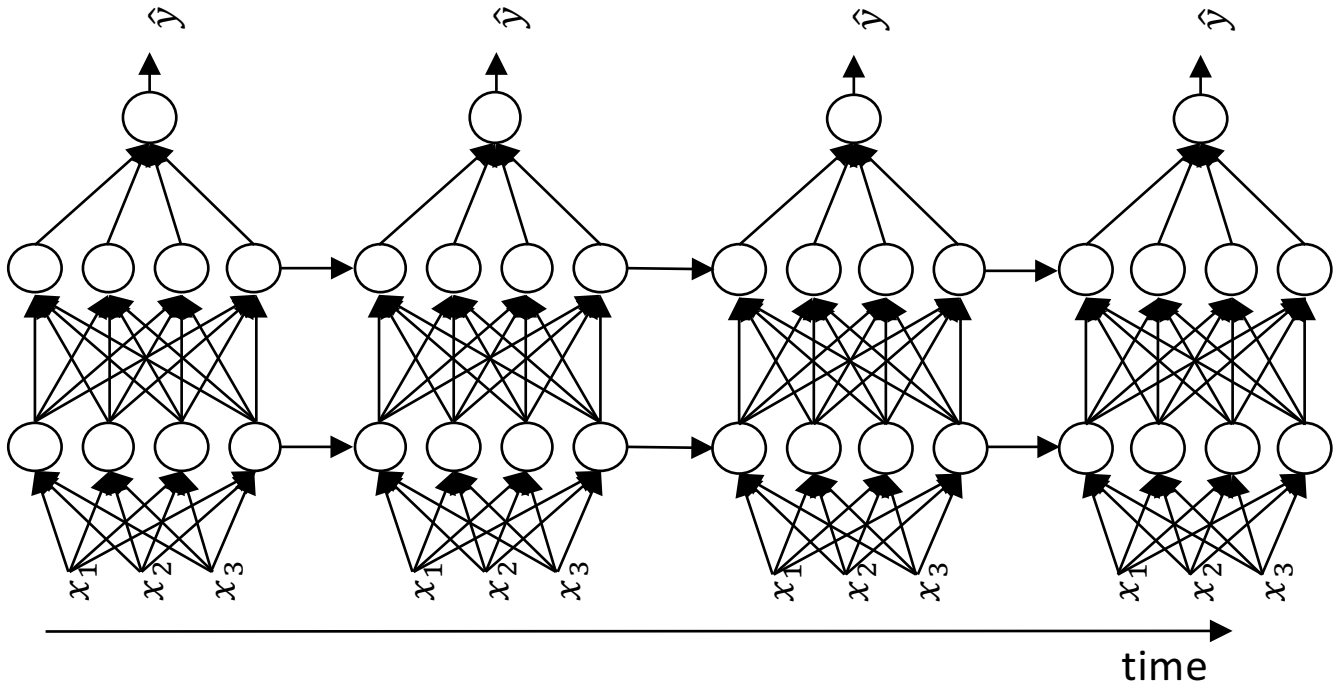


Deep NN main architectures

General Recurrent NN

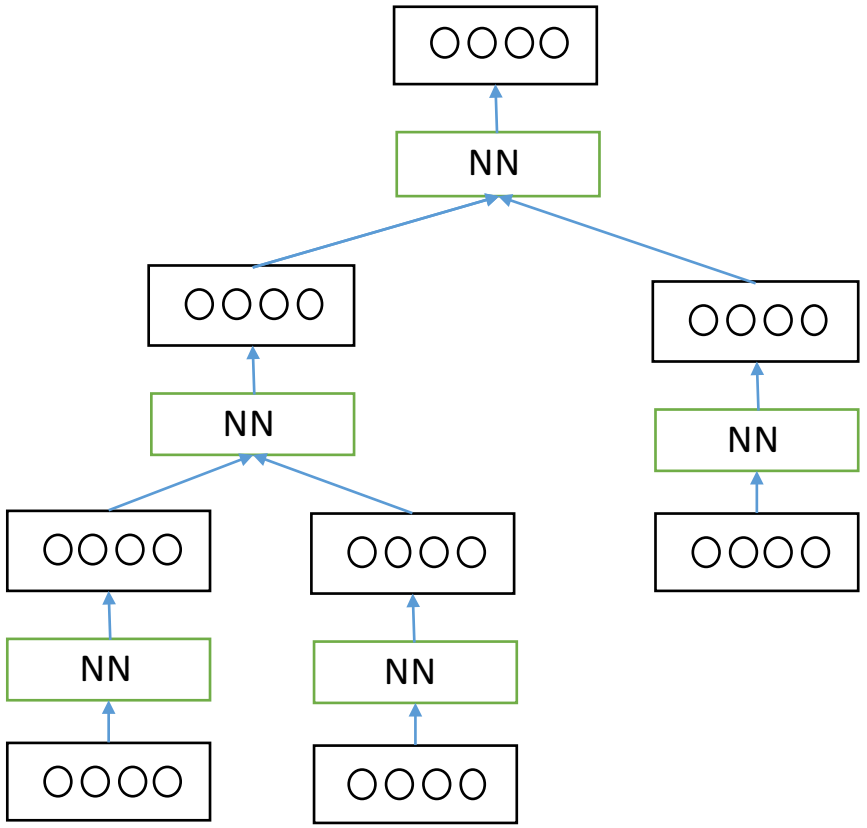


- Recurrent NN
  - Sequential models
    - NLP, Video, Signal
  - Problem of vanishing gradient
  - Gating Units: LSTM, GRU

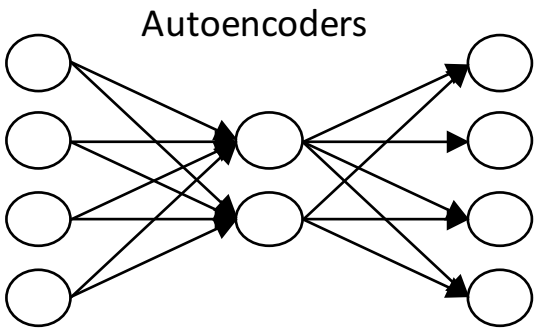


Deep NN main architectures

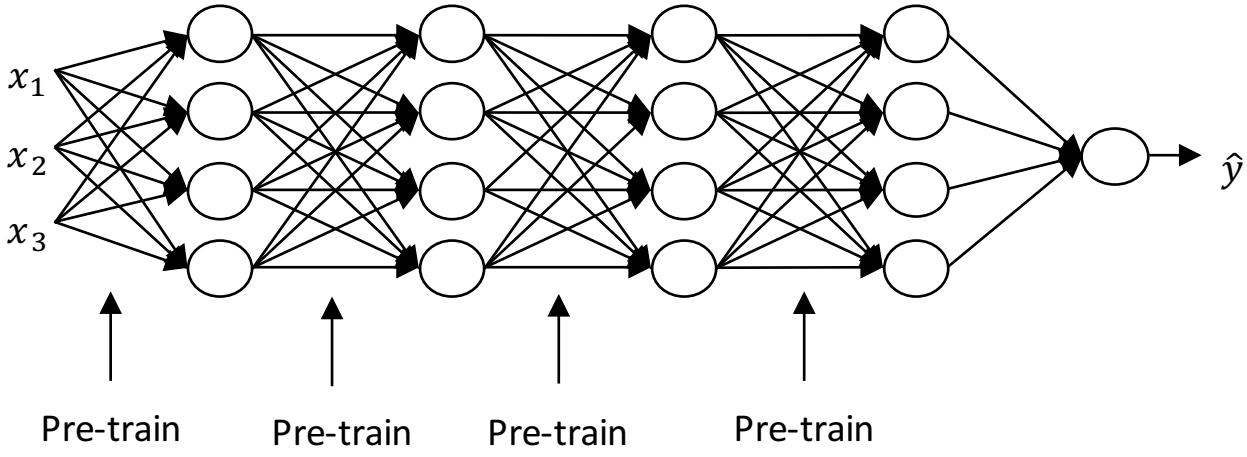
- Tree Recursive NN
  - Understand compositionality
  - NLP parsing, Vision
  - Tree based
  - Recursive



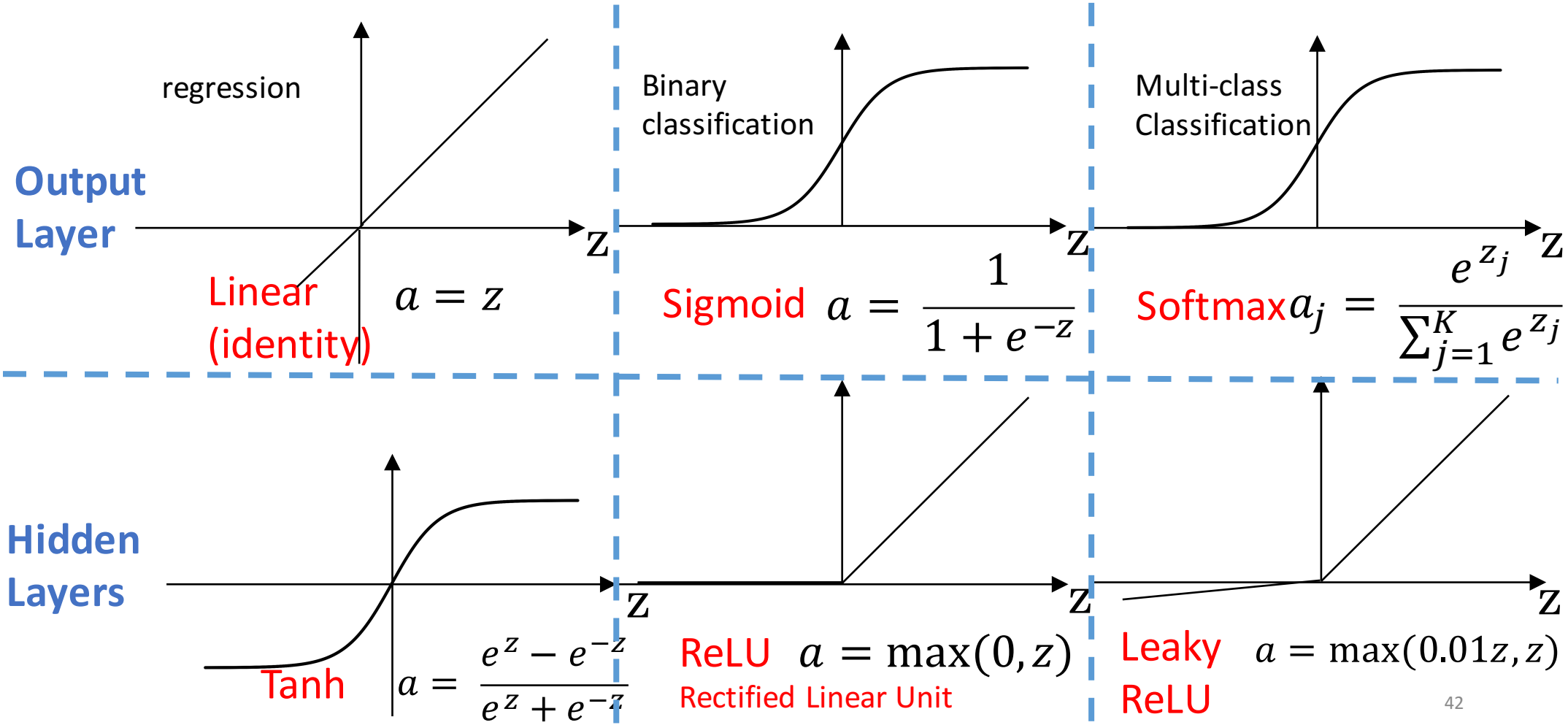
Deep NN main architectures



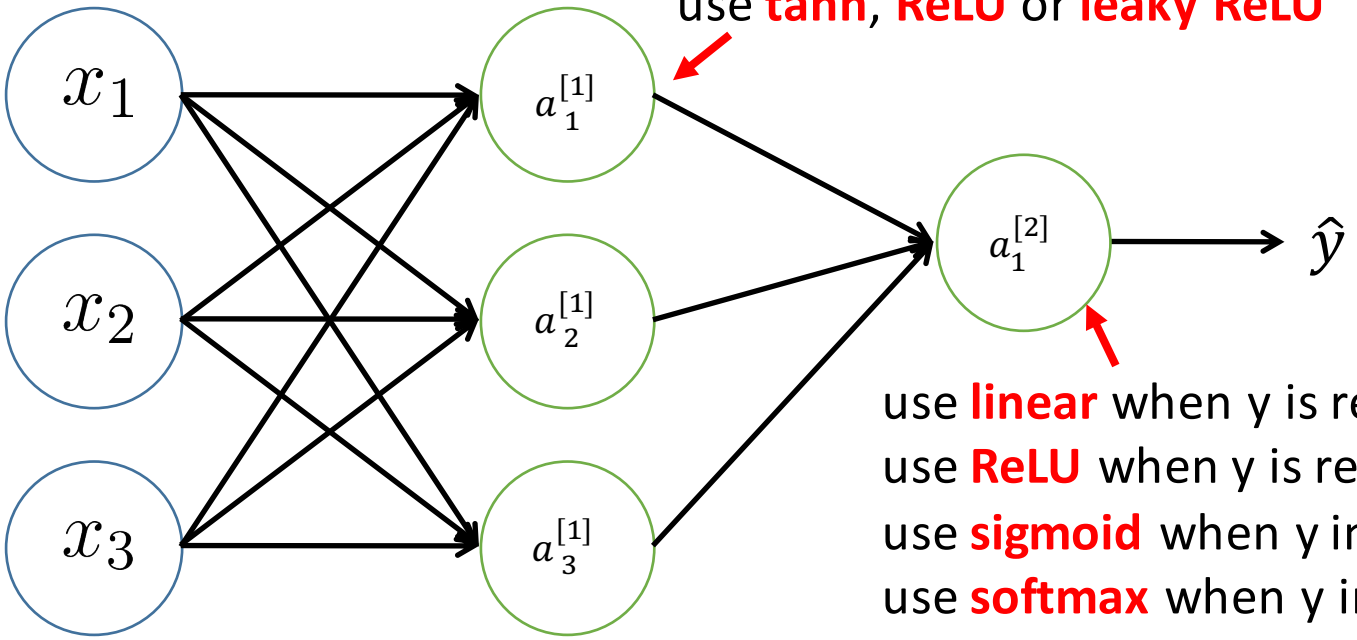
- Unsupervised Pre-trained NN
  - No Random Initialization
  - Parameters are initialized with unsupervised manner
    - Restricted Botzman Machines
    - Autoencoders
  - Regularization effect
    - Solves Overfitting
  - Transfer Learning
    - Model trained on one task is re-purposed on a second related task



Activation Functions

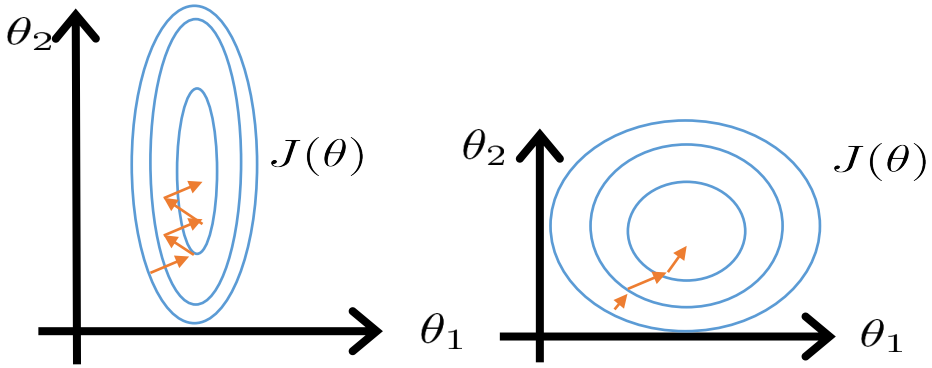


Activation Functions



Optimization (solve underfitting)

- **Back-Propagation**
- **Gradient Descent**
- **Normalizing** Inputs to speed up leaning
- **Gradient checking** to prevent Vanishing/Exploding gradient
- **Momentum** to prevent oscillations

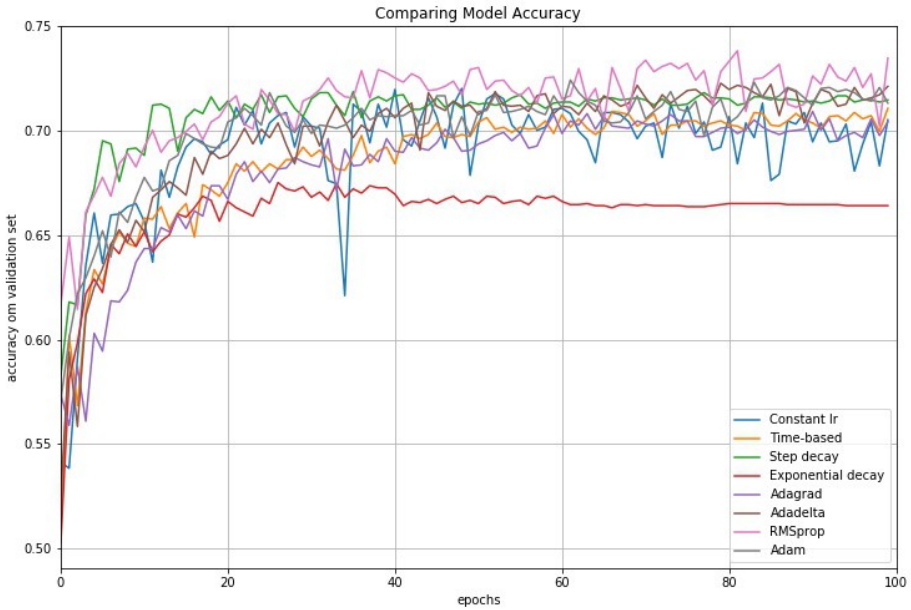


$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

Learning Rate

Optimization (solve underfitting)

- **Gradient Descent** (Batch, Mini-batch, Stochastic)
  - Constant Learning Rate Decay
  - Time based decay: lr = lr0/(1+kt)
  - Step decay: lr = lr0 * drop^floor(epoch / epochs_drop)
  - Exponential decay lr = lr0 * exp(−kt)
- **Adaptive** Learning Rate Decay
  - Adagrad
  - Adadelta
  - RMSprop
  - Adam
- More details: Yoshua Bengio Paper



Regularization (solve overfitting)

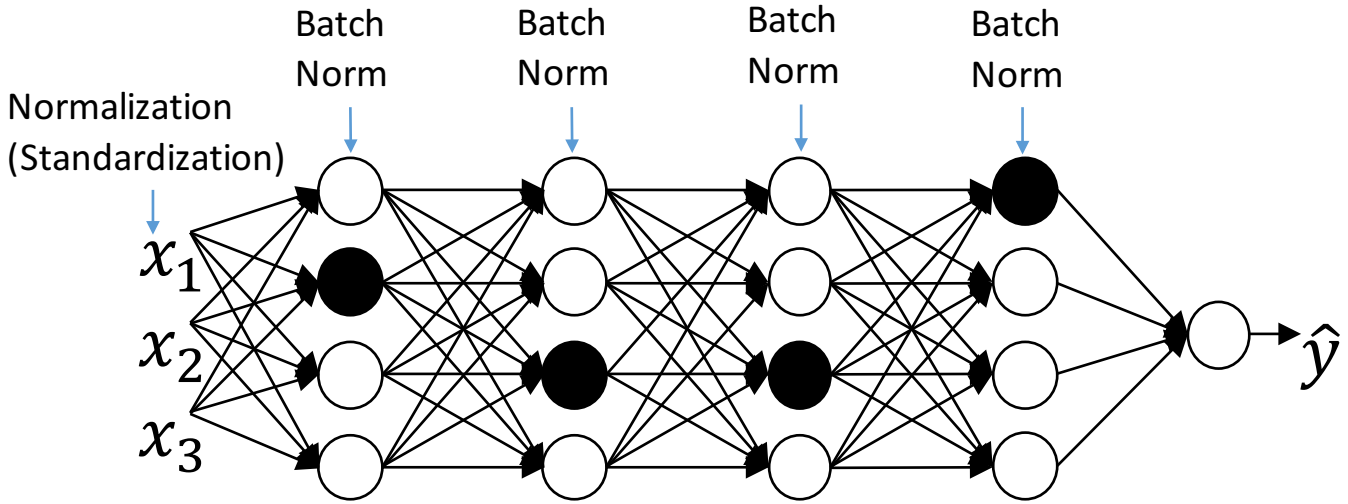
- **Dropout**
- **Batch Normalization**

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2$$

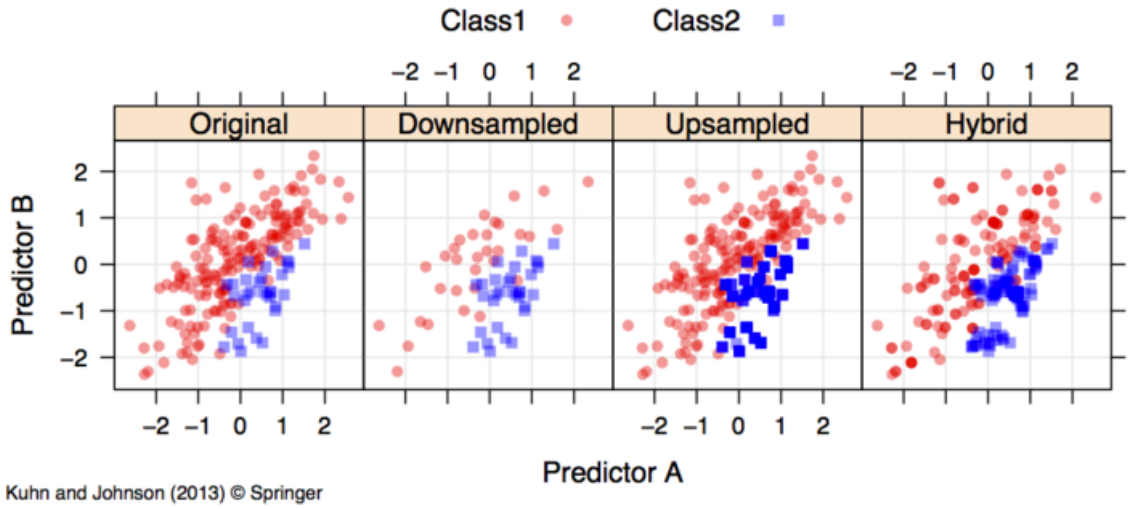
$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i)$$

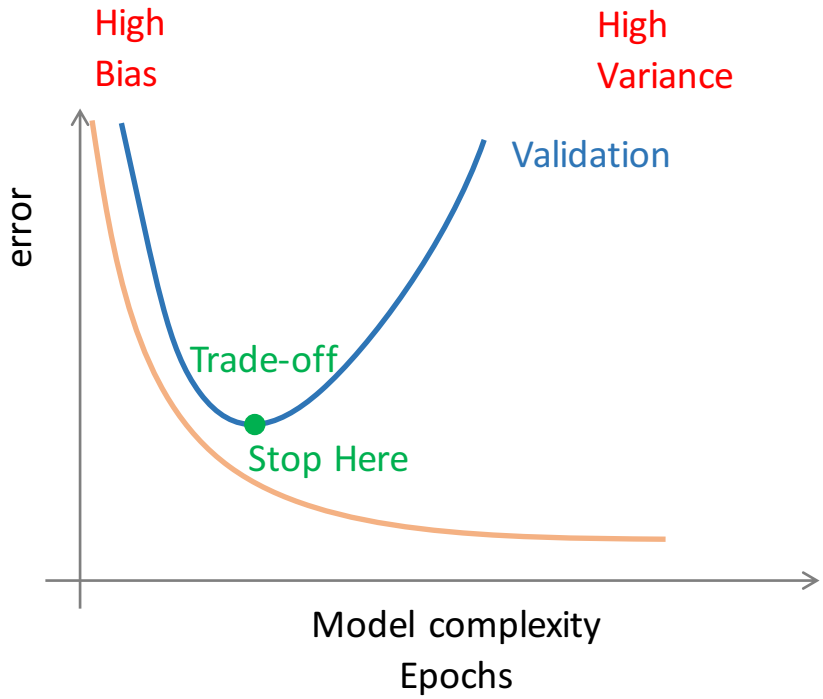


Regularization (solve overfitting)

- **Data augmentation**
- **Early stopping**



Kuhn and Johnson (2013) © Springer



How Can I Learn?

How can I Learn?

- Math
  - Statistics, Probabilistic Graphical Models, Algebra, Optimization
- Programming Languages
  - Python, R, others
- Books
  - Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. “An introduction to statistical learning with applications in R”. 2013.
  - Tom M. Mitchell. “Machine Learning”. 1997
  - Others

How can I Learn?

- MOOCs
  - Coursera: Machine learning, Deep learning specialization
  - Udacity, Udemy, Fun, etc.
- StackOverflow
- Research Papers
  - Read and rewrite algorithms from scratch
- Follow People
  - Androw Ng, Yann LeCun, Jeff Hinton, Yoshua Bengio, Chris Manning, Sebastian Thrun, etc.

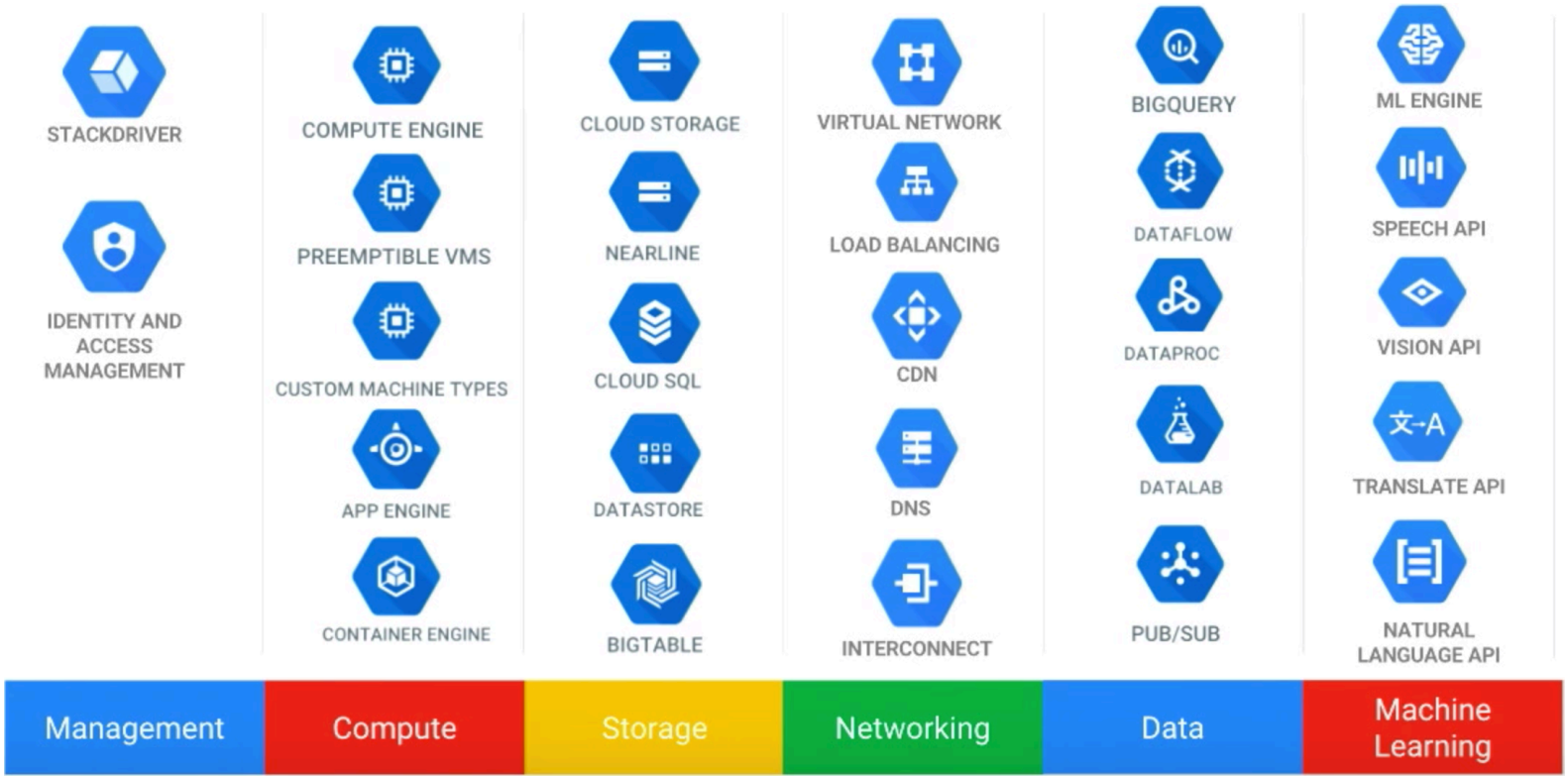
How Can I Apply?

How can I Apply?

- Start small projects and use Framworks
  - TensorFlow, Keras, Caffe, Microsoft Cognitive Toolkit (CNTK 2), MXNet, Scikit-learn, Spark MLlib, etc.
- Kaggle, UCI Machine Learning Repo
  - Find data
  - Go for competitions
- Notebooks:
  - Jupyter notebook, google cloud Datalab
- Github
  - Find codes
  - Share your code
- Softwares
  - Knime: open source data analytics
  - IBM SPSS Modeler: data mining and text analytics software application from IBM

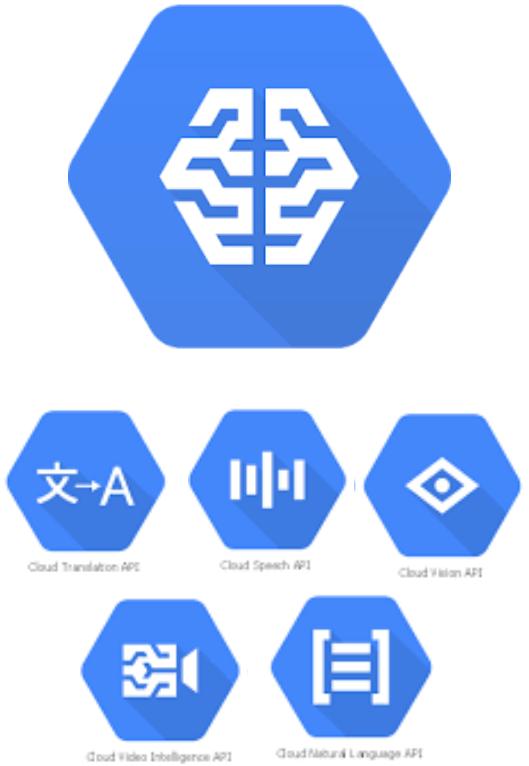
How can I Apply?

Google Cloud Services



How can I Apply?

- Google ML APIs
  - Vision API
  - Video Intelligence API
  - Speech API
  - Natural Language API
  - Translation API

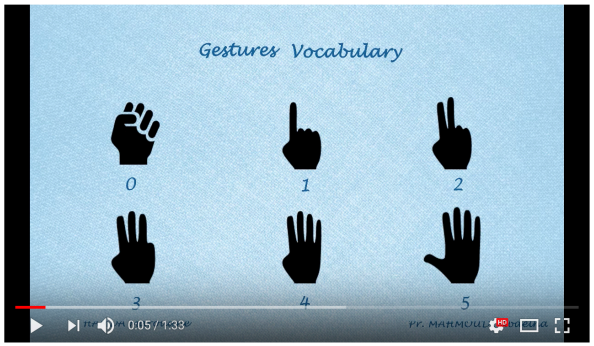


Demos

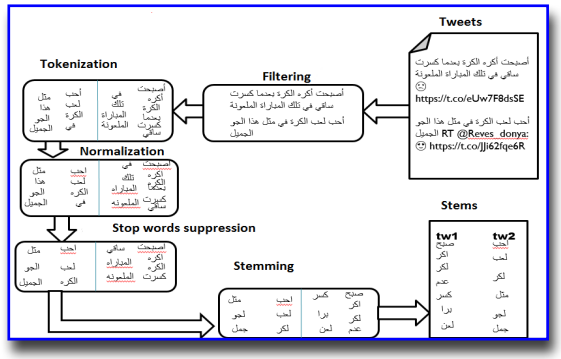
My PhD students demos (Youtube)

Younes Choubik

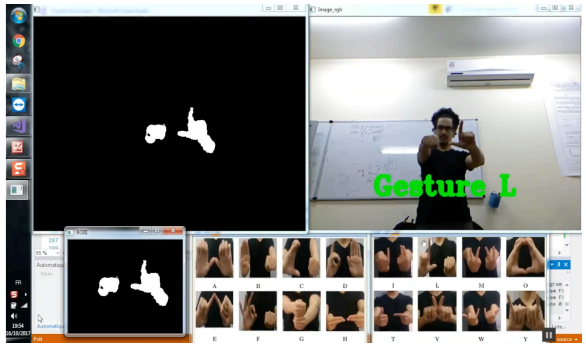
Marouane Hamda



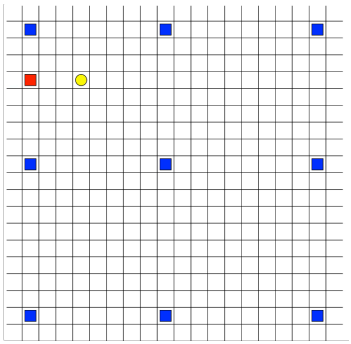
Hicham Boudkik



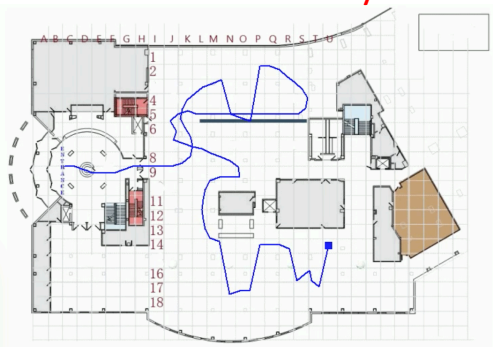
Marouane Benmoussa



Mohamed El Kaddoury



Mohamed Es-Salhy



THANKS!