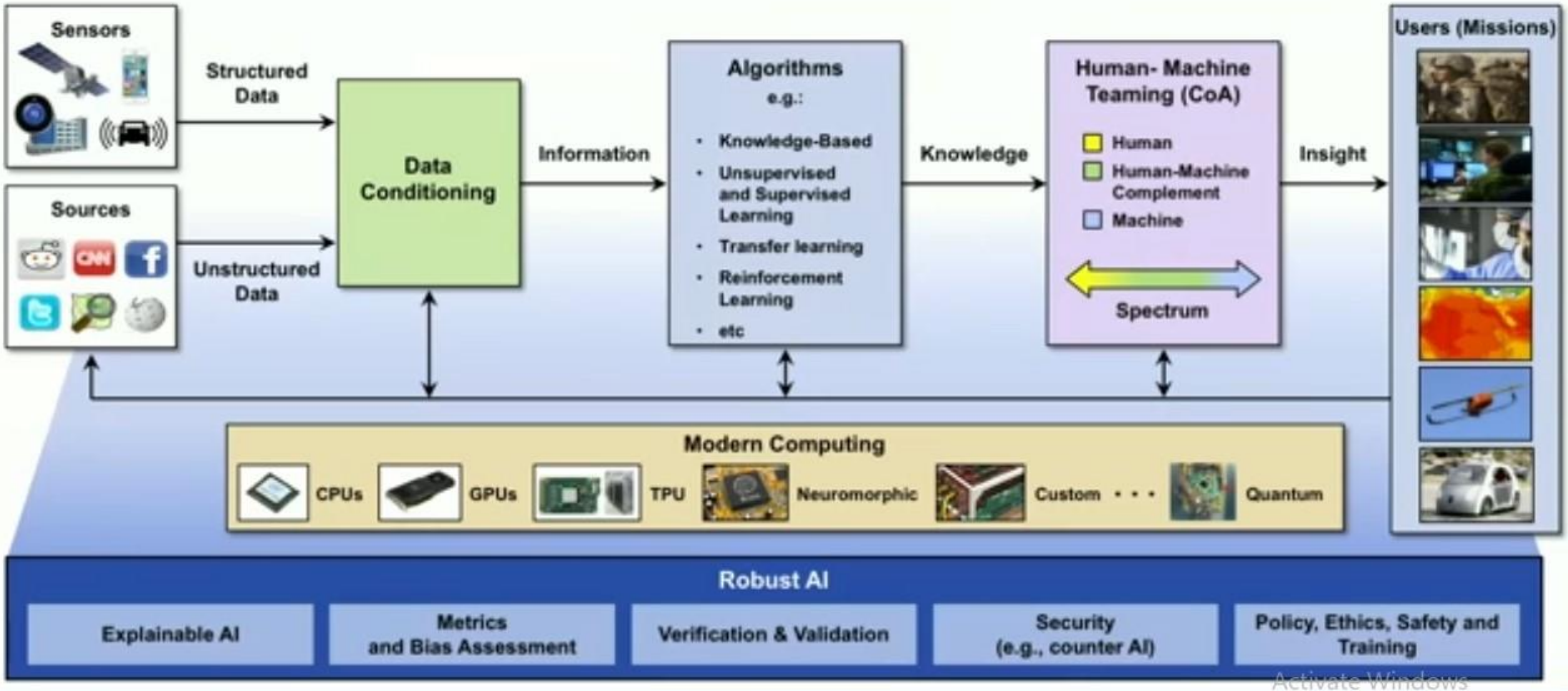
MACHIN LEARNING MIT OPEN COURS

Win+w

Outline

- Artificial Intelligence Overview
- Machine Learning Deep Dives
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
- Conclusions/Summary

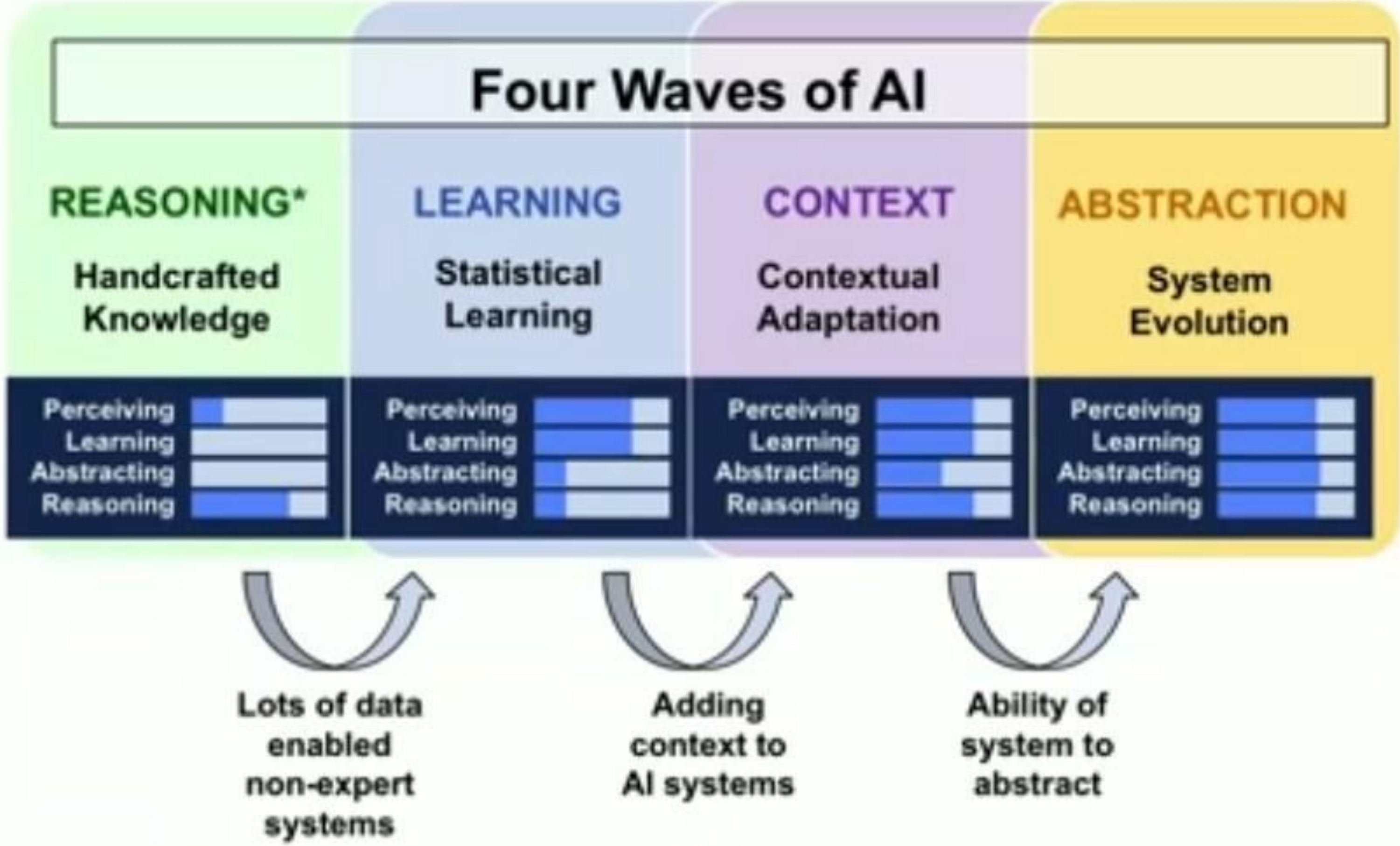
AI Canonical Architecture



GPU = Graphics Processing Unit CoA = Courses of Action
TPU = Tensor Processing Unit

Artificial Intelligence Evolution

* Waves adapted from John Launchbury, Director I2O, DARPA



Unstructured and Structured Data

Structured Data Types

Unstructured Data Types

Data Conditioning/Storage Technologies

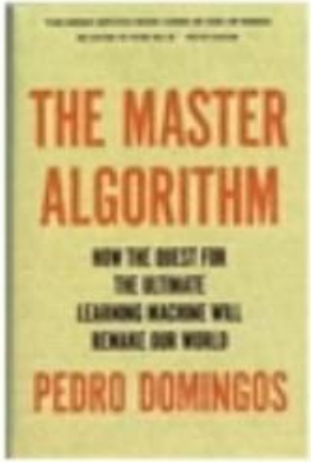
- Data to Information -

Technologies	Capabilities Provided
Infrastructure/Databases  ORACLE accumulo	• Indexing/Organization/Structure • Domain Specific Languages • High Performance Data Access • Declarative Interfaces
Data Curation 	• Unsupervised machine learning • Dimensionality Reduction • Clustering/Pattern Recognition • Outlier Detection
Data Labeling 	• Initial data exploration • Highlight missing or incomplete data • Reorient sensors/recapture data • Look for errors/biases in collection

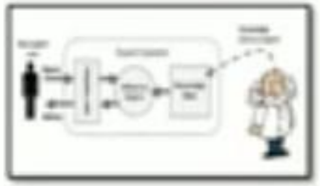
Often takes up 80+% of overall AI/ML development work

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

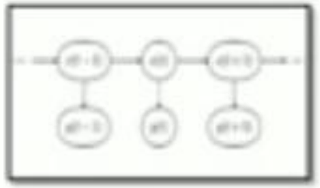
Machine Learning Algorithms Taxonomy



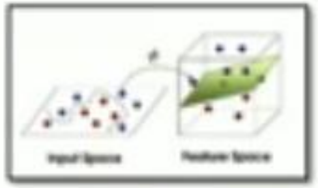
Algorithms*



Symbolists (e.g., exp. sys.)



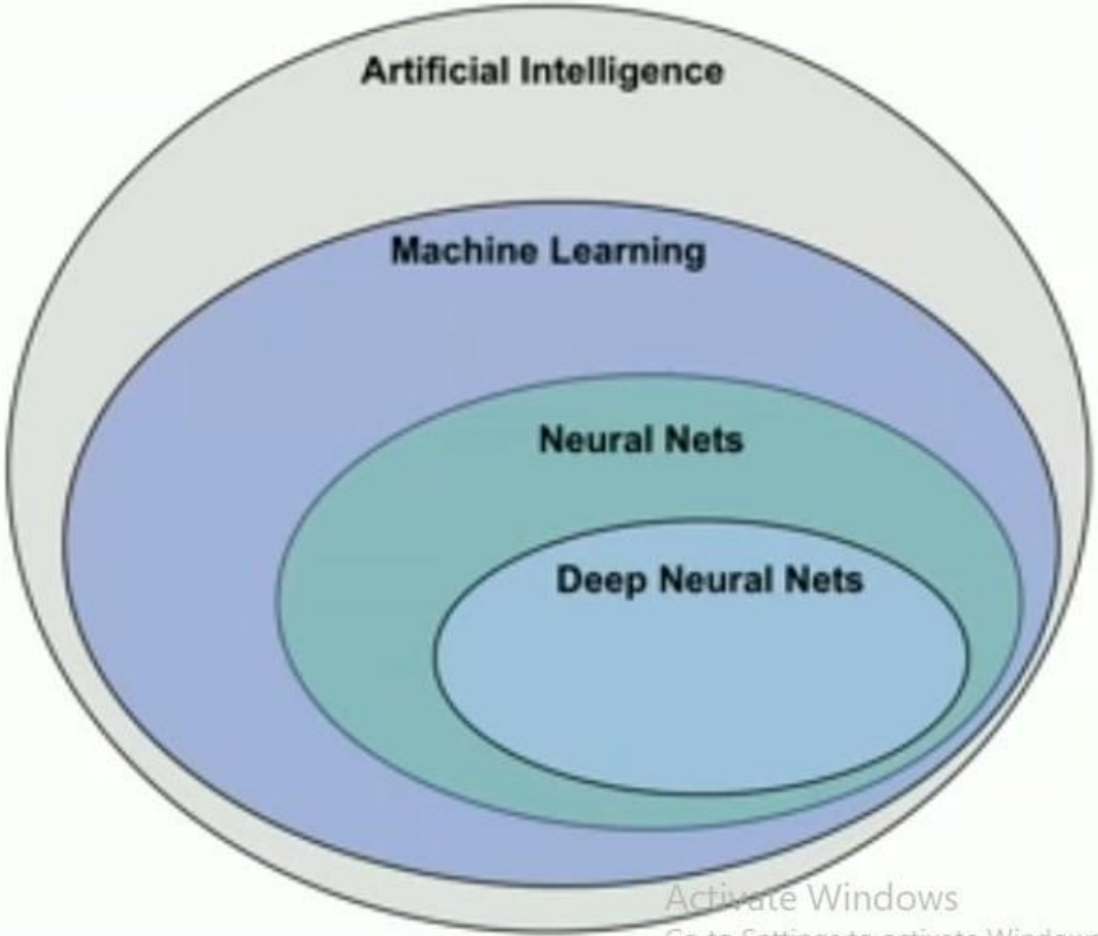
Bayesians (e.g., naive Bayes)



Analogizers (e.g., SVM)

Connectionists (e.g., DNN)

Evolutionaries (e.g., genetic programming)



* "The Five Tribes of Machine Learning", Pedro Domingos

DNN = Deep Neural Networks
SVM = Support Vector Machines
Exp. Sys. = Expert Systems

Image Adapted From "Deep Learning" by Ian Goodfellow, Yoshua Bengio and Aaron Courville

Modern AI Computing Engines

Computing Class — **What It Provides to AI**

CPU
- Most popular computing platform
- General purpose compute

GPU
- Used by most for training algorithms (good for NN backpropagation)

TPU
- Speeds up inference time (domain specific architecture)

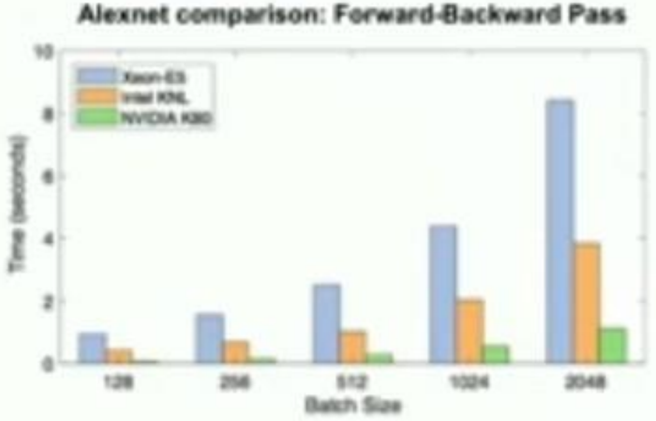
Neuromorphic
- Active research area

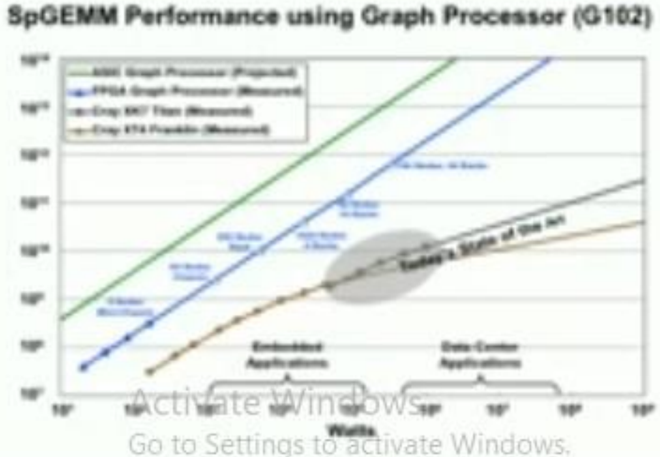
Custom
- Ability to speed up specific computations of interest (e.g. graphs)

⋮

Quantum
- Benefits unproven until now
- Recent results on HHL (linear system of equations)

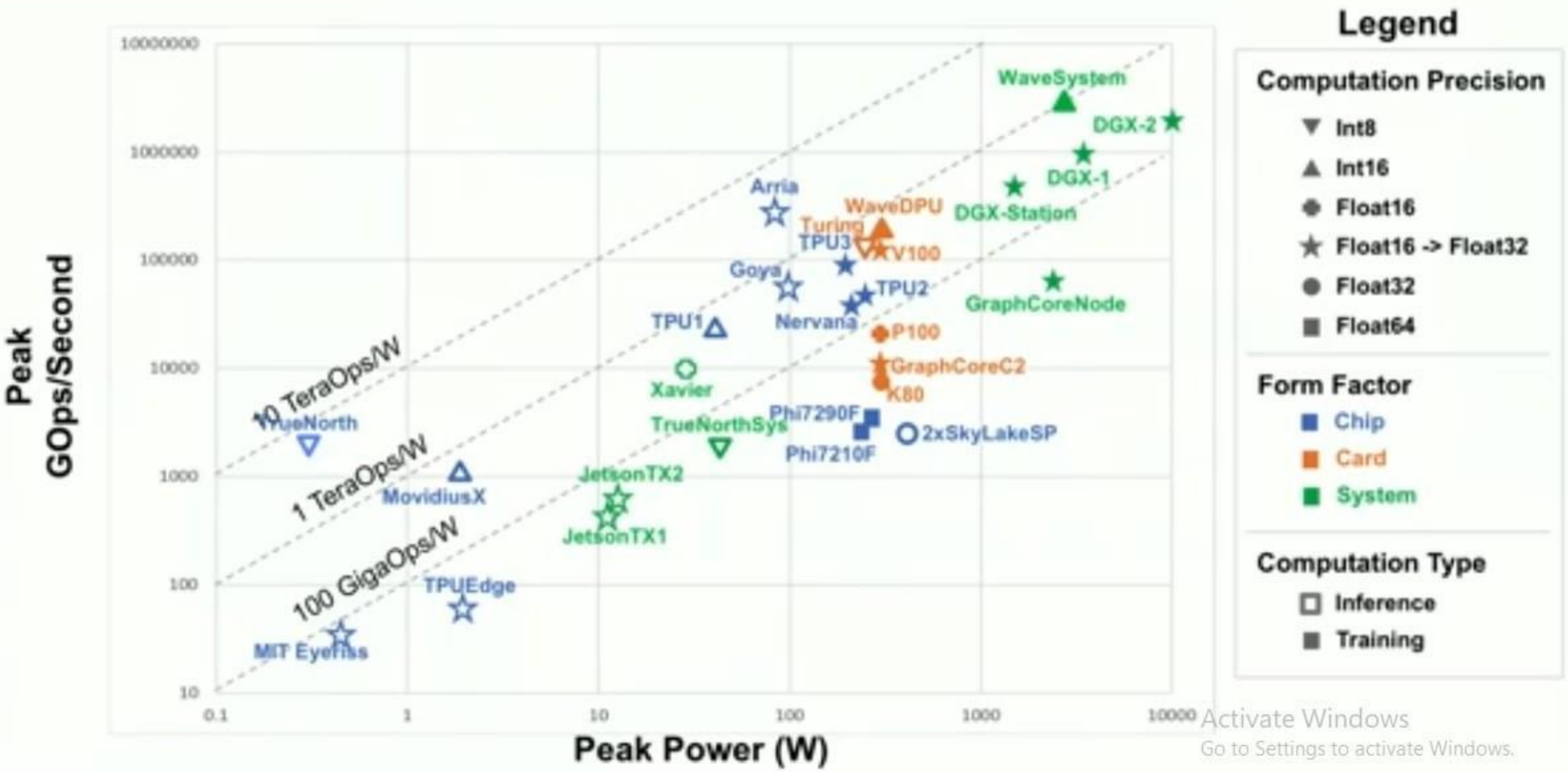
Selected Results





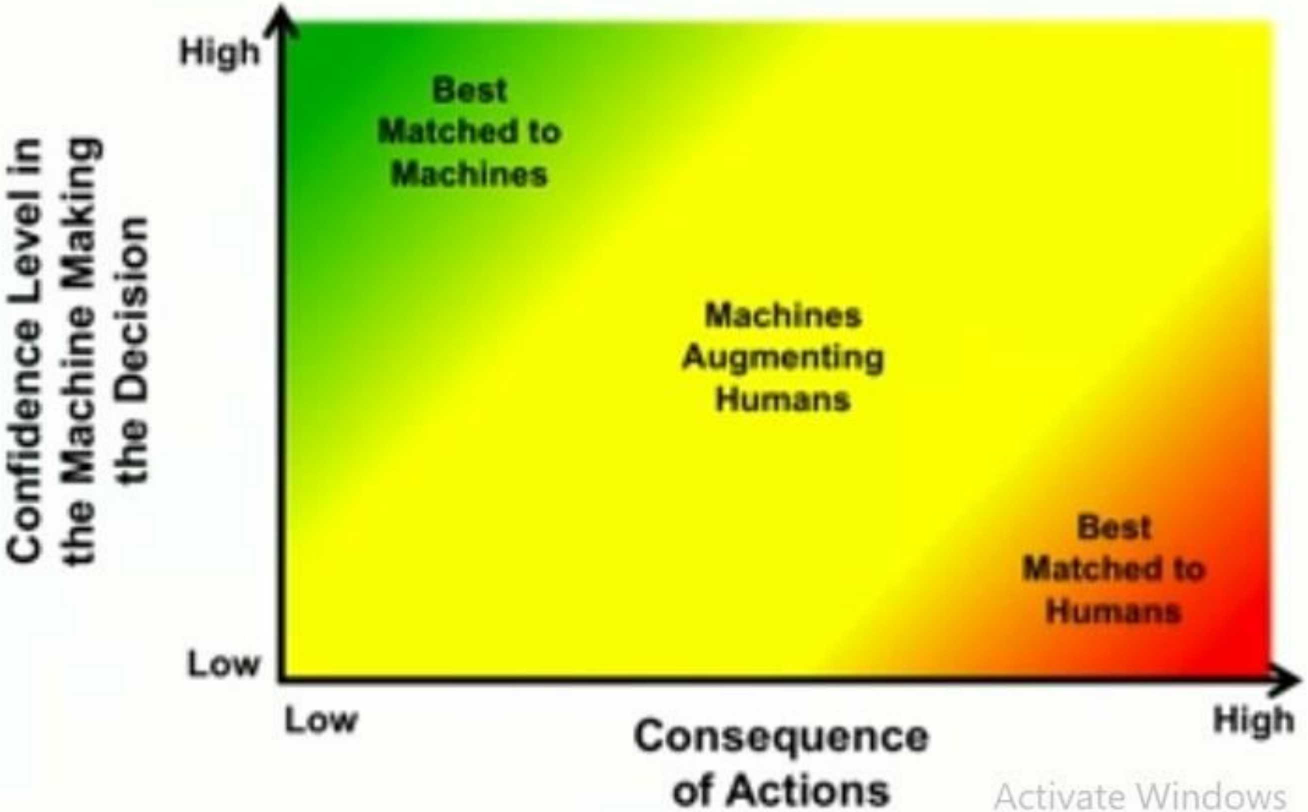
GPU = Graphics Processing Unit
TPU = Tensor Processing Unit
HHL = Harrow-Hassidim-Lloyd quantum algorithm

Neural Network Processing Performance



Reuther, Albert, et al. "Survey and Benchmarking of Machine Learning Accelerators." *arXiv preprint arXiv:1908.11348* (2019).

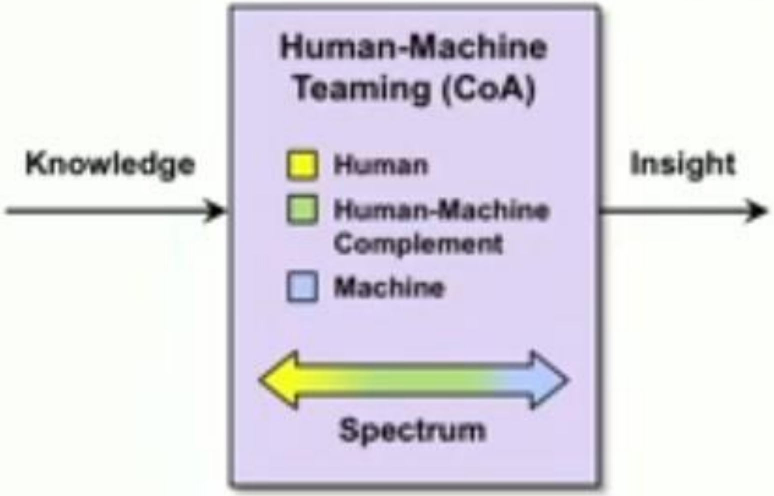
Robust AI: Preserving Trust



Importance of Robust AI

Robust AI Feature	Issue	Example	Solutions
Explainable AI	User unfamiliarity or mistrust leads to lack of adoption		Seamless integration, model expansion, transparent uncertainty
Metrics	Unknown relationship between arbitrary input and machine output		Explainability, dimensionality reduction, feature importance inference
Validation & Verification	Algorithms need to meet mission specifications		Robust training, "portfolio" methods, regularization
Security	System vulnerable to adversarial action (both cyber and physical)		Model failure detection, red teaming
Policy, Ethics, Safety, and Training	Unwanted actions when controlling heavy or dangerous machinery		Risk sensitivity, robust inference, high decision thresholds

Human-Machine Teaming



Human-Machine teaming will consist of intelligent assistants enabled by artificial intelligence

Scale vs. Application Complexity

Confidence Level vs. Consequence of Actions

Critical Element of AI: Understanding how humans and machines can work together for applications

What is Machine Learning?

- Machine Learning
  - Study of algorithms that improve their performance at some task with experience (data)
  - Optimize based on performance criterion using example data or past experience
- Combination of techniques from statistics, computer science communities
- Getting computers to program themselves
- Common tasks:
  - Classification
  - Regression
  - Prediction
  - Clustering
  - …

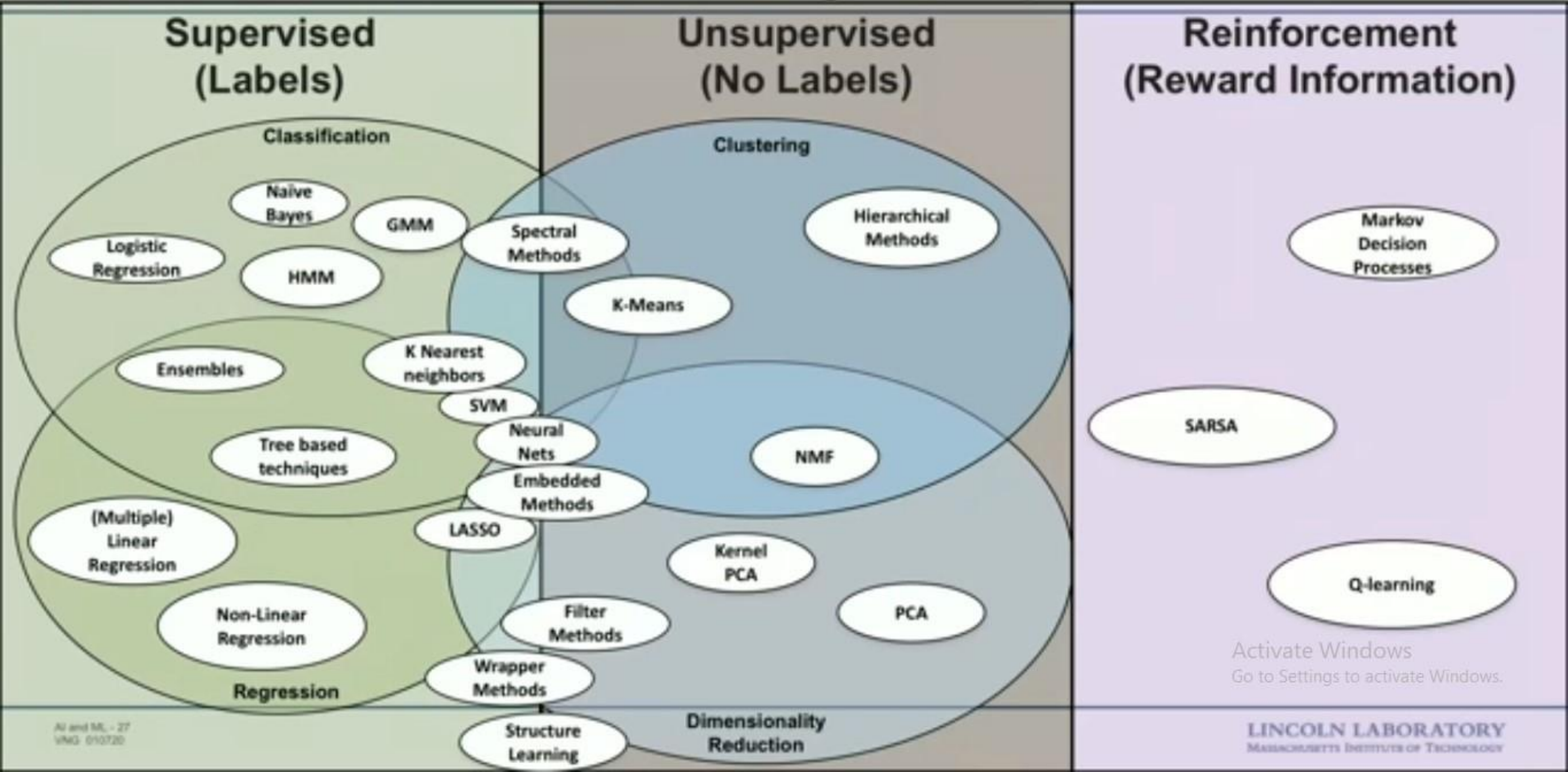
Traditional Programming vs. Machine Learning

Traditional Programming

Machine Learning

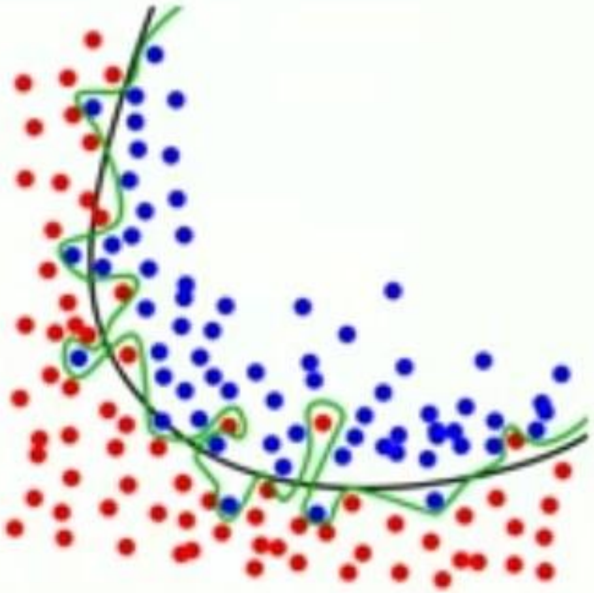
Source: Pedro Domingos, University of Washington

Machine Learning Techniques



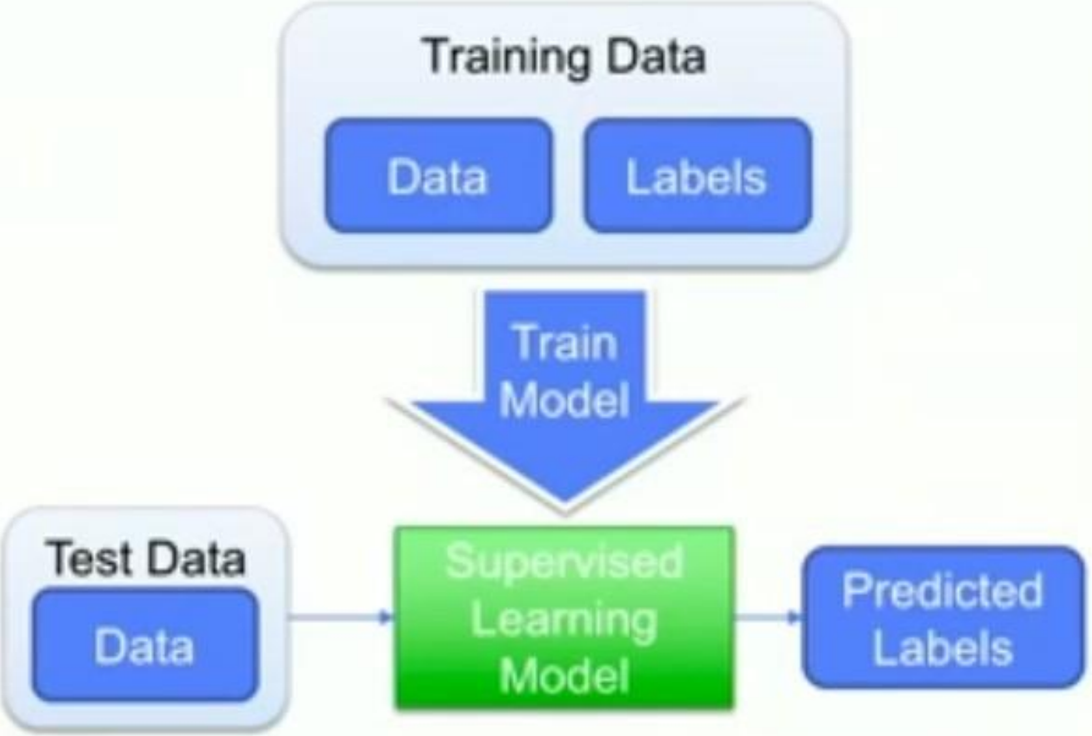
Common ML Pitfalls

- Over-fitting vs. Under-fitting
- Bad/noisy/missing data
- Model selection
- Lack of success metrics
- Linear vs. Non-linear models
- Ignoring outliers
- Training vs. testing data
- Computational complexity, curse of dimensionality
- Correlation vs. Causation



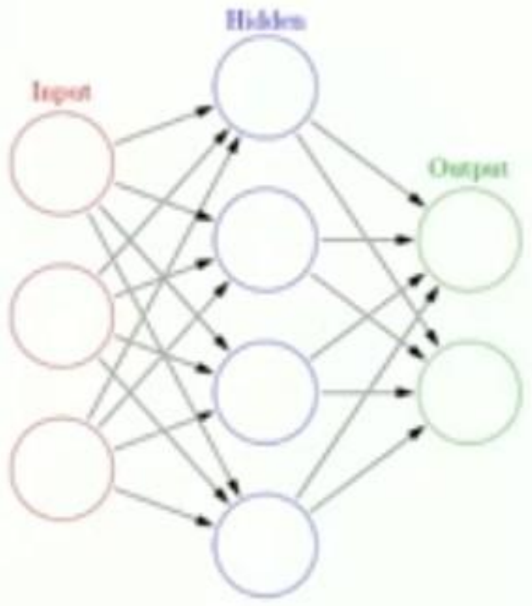
Supervised Learning

- Starting with labeled data (ground truth)
- Build a model that predicts labels
- Two general goals:
  - Regression: predict continuous variable
  - Classification: predict a class or label
- Generally has a training step that forms the model

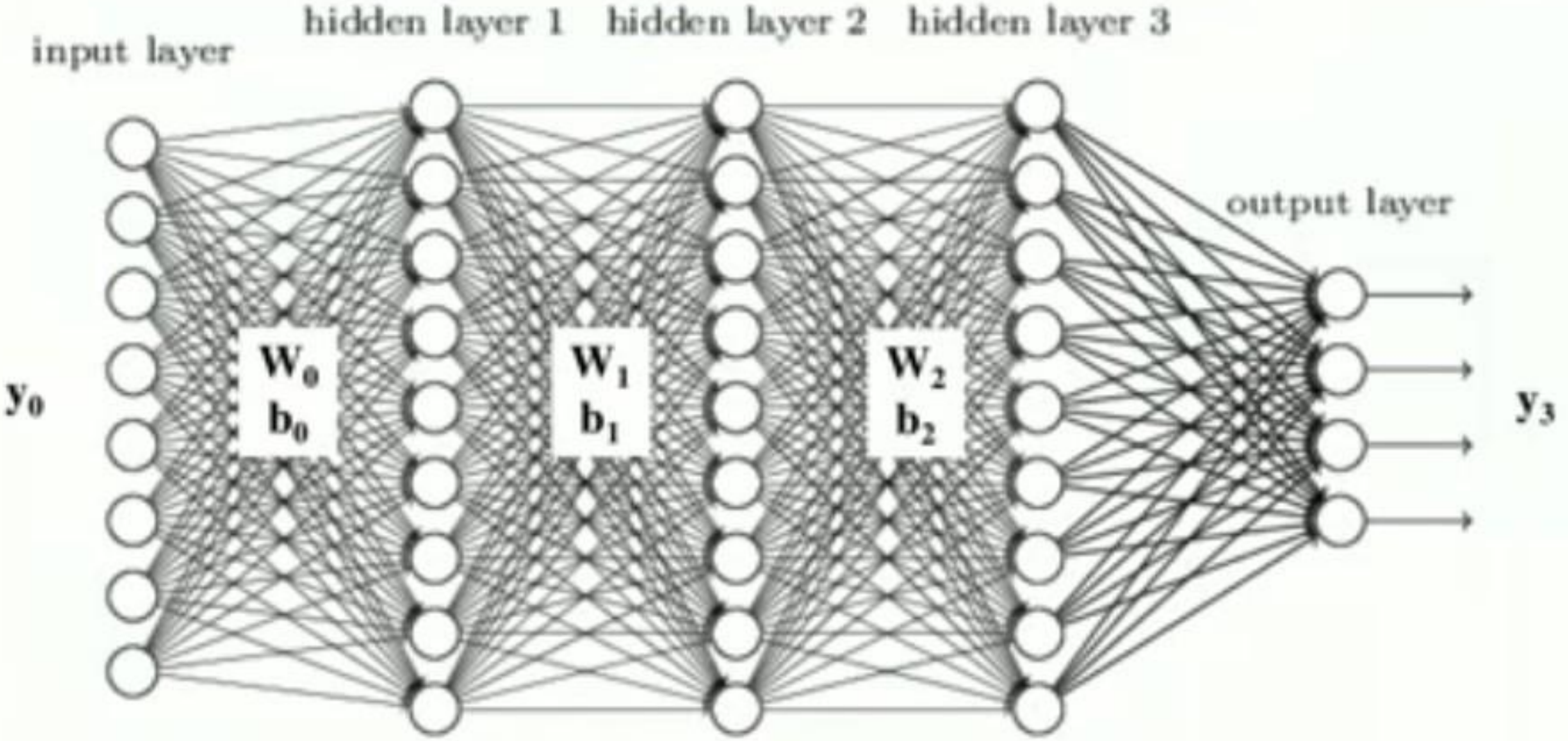


Artificial Neural Networks

- Computing systems inspired by biological networks
- Systems learn by repetitive training to do tasks based on examples
  - Generally a supervised learning technique (though unsupervised examples exist)
- Components: Inputs, Layers, Outputs, Weights
- Deep Neural Network: Lots of “hidden layers”
- Popular variants:
  - Convolutional Neural Nets
  - Recursive Neural Nets
  - Deep Belief Networks
- Very popular these days with many toolboxes and hardware support



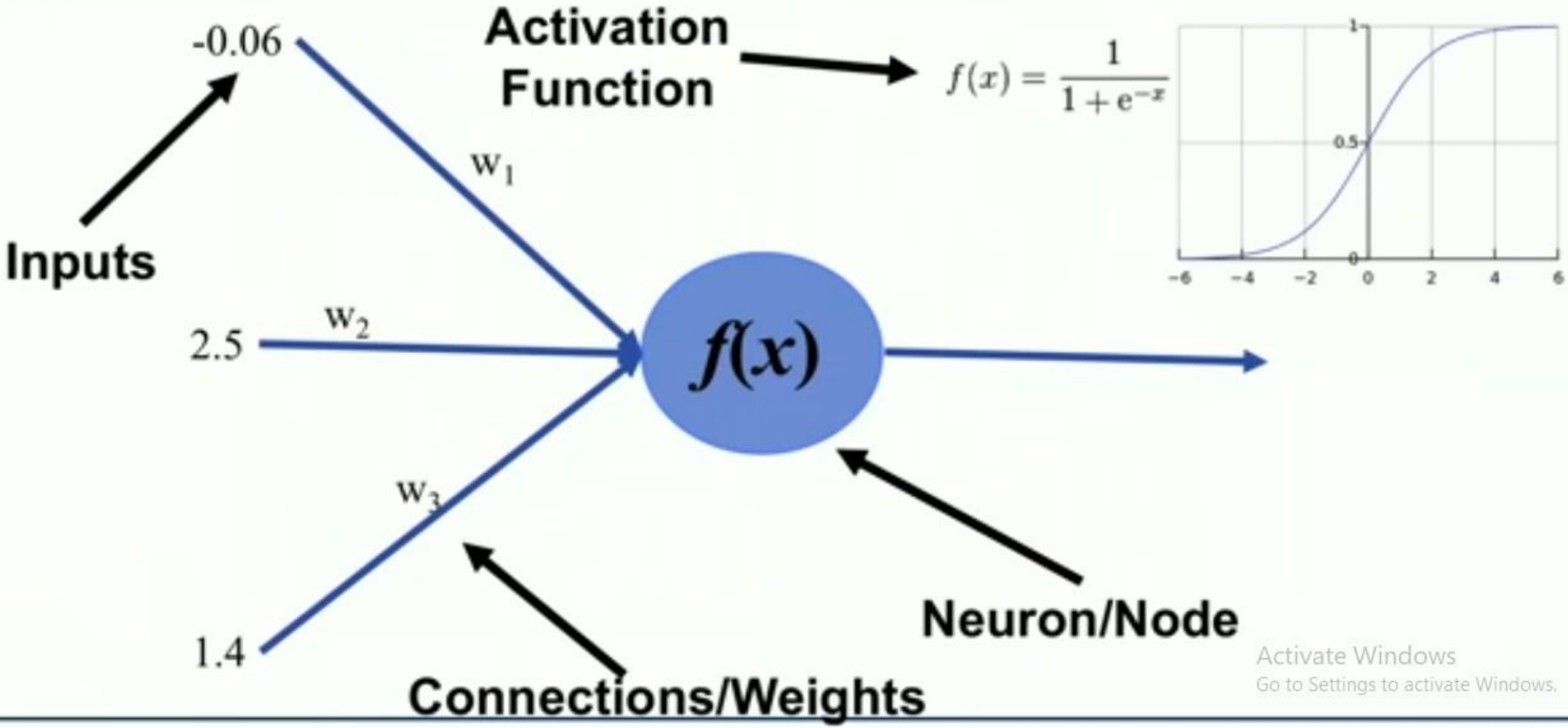
Deep Neural Networks



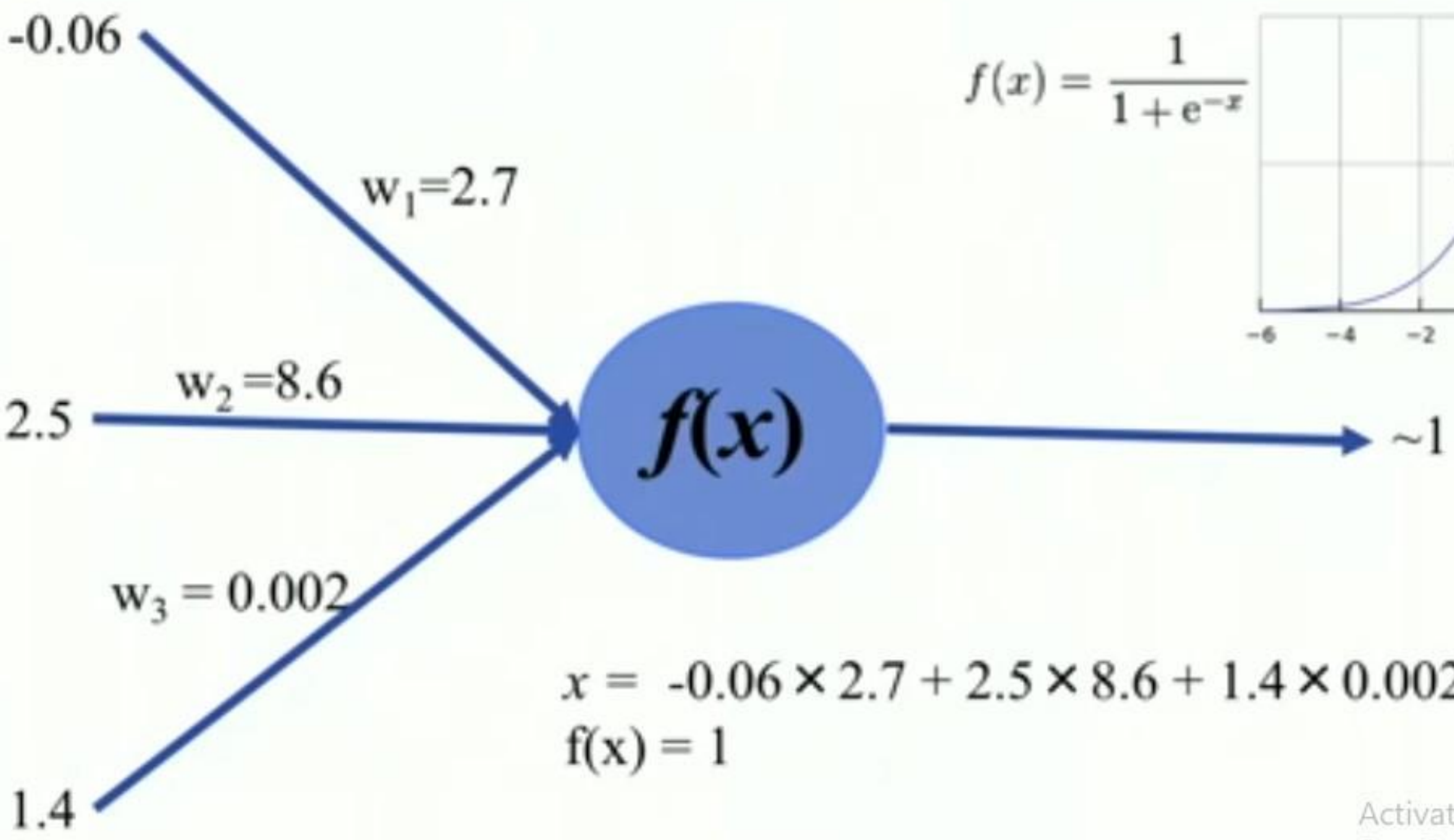
$$y_{i+1} = f(W_i\, y_i + b_i)$$

Image source: http://www.rsipvision.com/exploring-deep-learning/

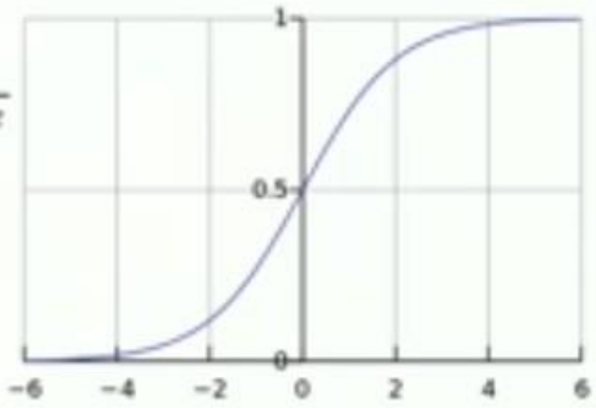
Components of an Artificial Neural Network



Components of an Artificial Neural Network



$$f(x) = \frac{1}{1 + e^{-x}}$$

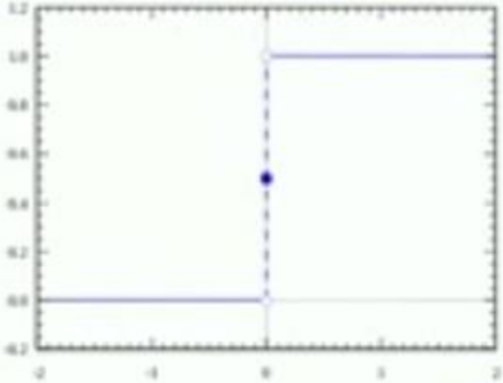


$$x = -0.06 \times 2.7 + 2.5 \times 8.6 + 1.4 \times 0.002 = 21.34$$

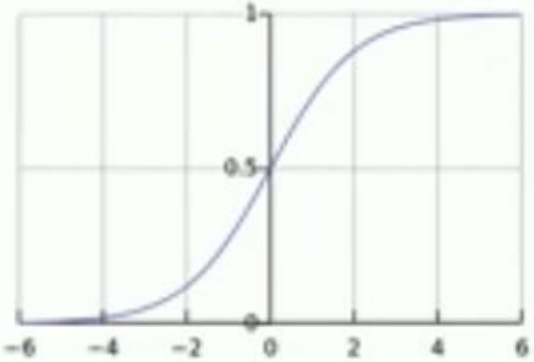
$$f(x) = 1$$

Common Activation Functions

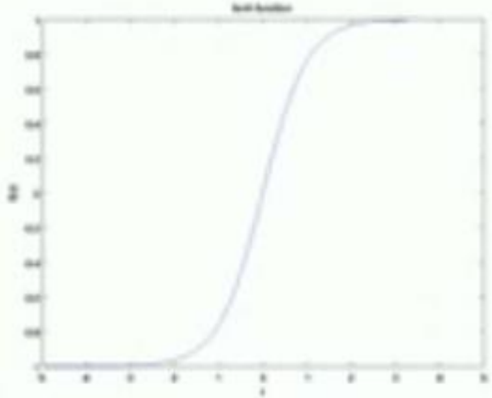
- **Step Function:** $f(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases}$



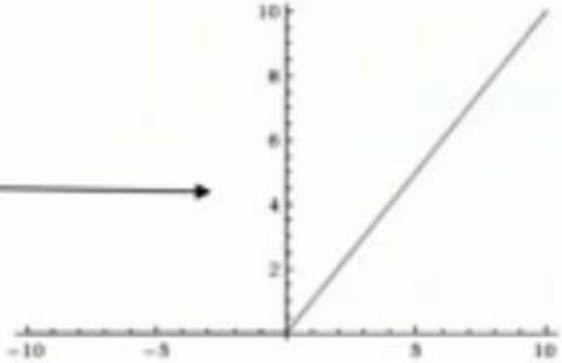
- **Sigmoid Function:** $f(x) = \frac{1}{1+e^{-x}}$



- **Tanh Function:** $f(x) = tanh(x)$



- **Rectified Linear Unit (ReLU):** $f(x) = max(0, x)$



Neural Network Landscape

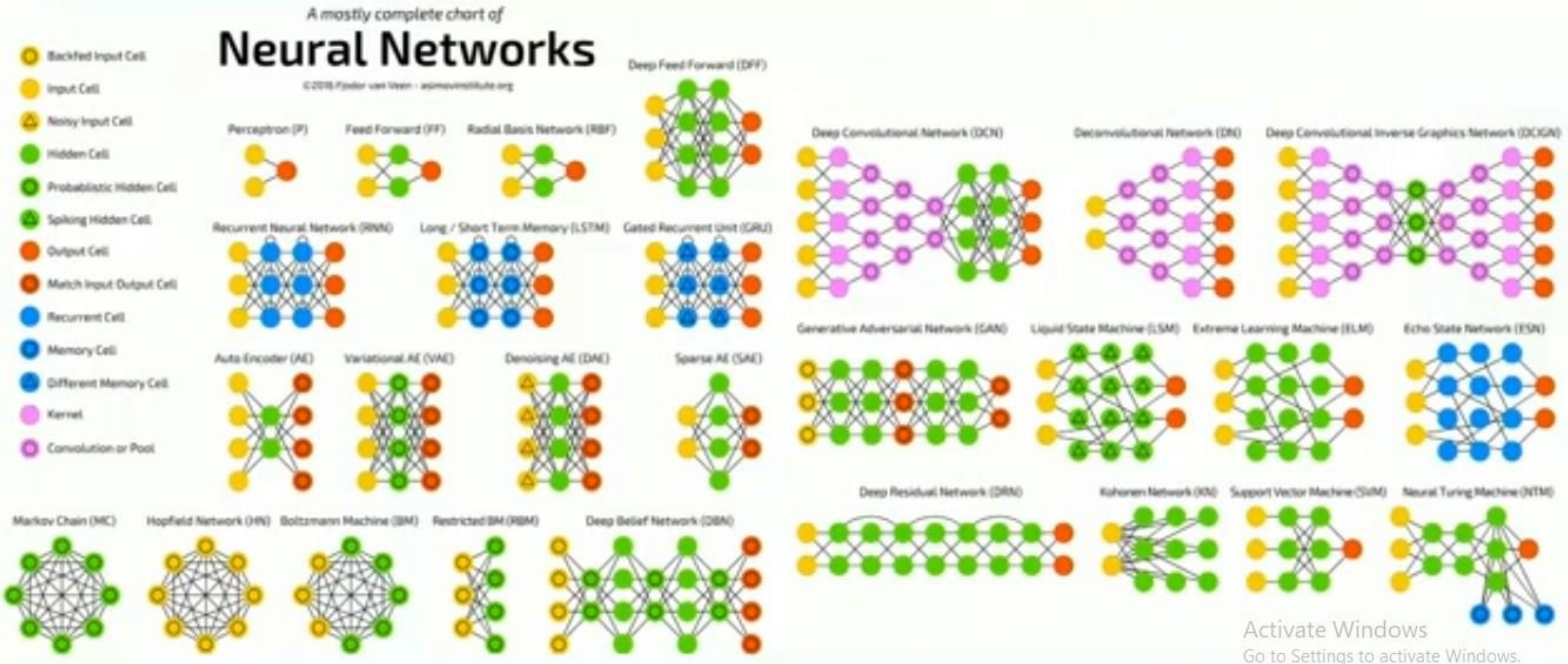


Image Source: http://www.asimovinstitute.org/neural-network-zoo/

Neural Network Training

Key Idea: Adjusting the weights changes the function represented by the neural network (*learning = optimization in weight space*).

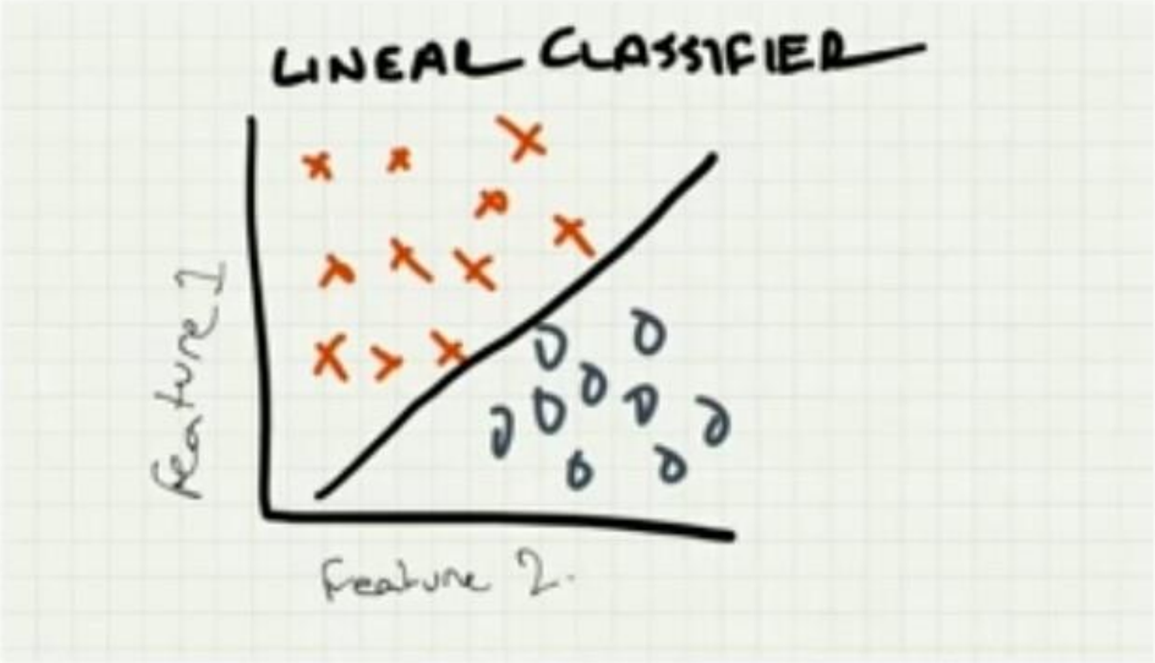
Iteratively *adjust weights* to reduce *error* (difference between network output and target output)

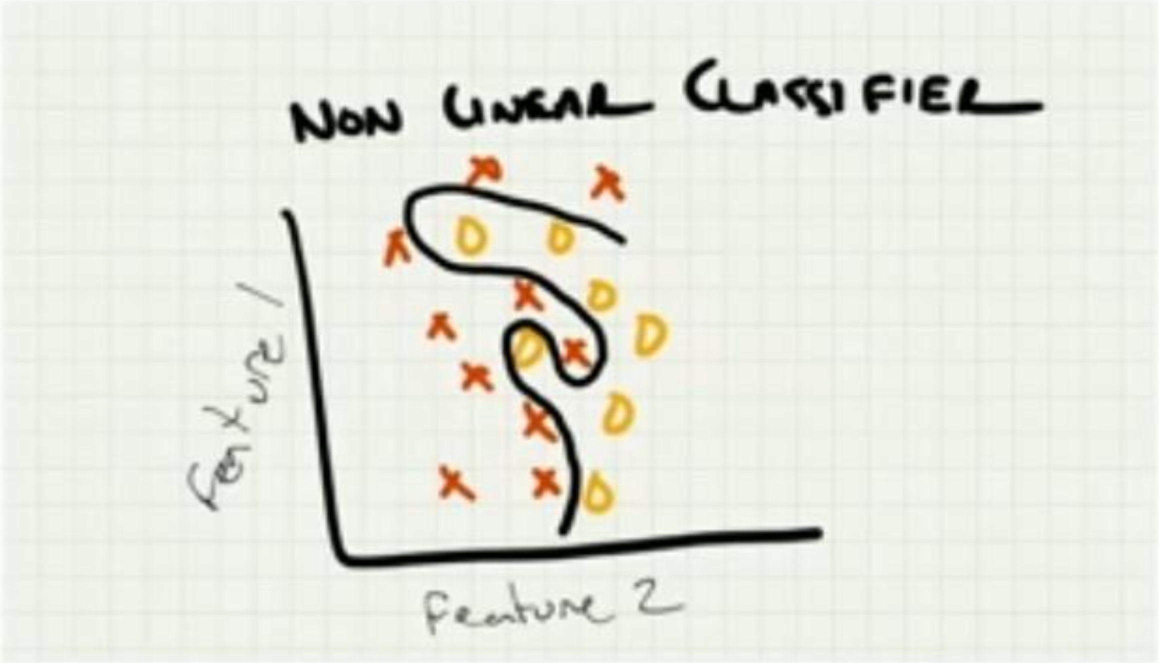
Weight Update

- *perceptron training rule*
- *linear programming*
- *delta rule*
- *Backpropagation*

Real neural network architectures can have 1000s of input data points, hundreds of layers and millions of weight changes per iteration

Neural Network Learning: Decision Boundary





Designing a Neural Network

- Designing a neural network can be a complicated task.
- Many choices:

- Depth (number of layers)
- Inputs (number of inputs)
- Type of Network:
  - Convolutional Neural Network
  - Deep Feedforward Neural Network
  - Deep Belief Network
  - Long/Short Term Memory
  - ...
- Types of layers:
  - MaxPool
  - Dropout
  - Convolutional
  - Deconvolutional
  - Softmax
  - Fully Connected
  - Skip Layer
  - ...
- Training Algorithm
  - Performance vs. Quality
  - Stopping criteria
  - Performance function
- Metrics:
  - False positive
  - ROC curve
  - ...

Unsupervised Learning

- Task of describing hidden structure from unlabeled data
- More formally, we observe features X_1, X_2,…,X_n and would like to observe patterns among these features.
  - We are not interested in predcition because we don't know what an output *Y* would look like.
- Typical tasks and associated algorithms:
  - Clustering
  - Data projection/Preprocessing
- Goal is to discover interesting things about the dataset: subgroups, patterns, clusters?

More on Unsupervised Learning

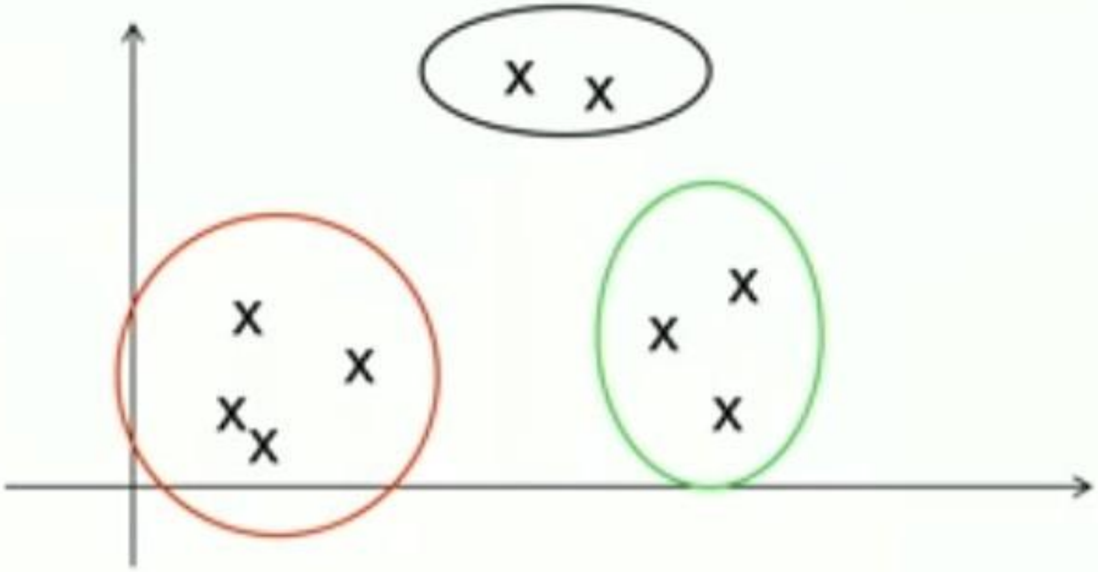
- There is no good simple goal (such as maximizing certain probability) for the algorithm
- Very popular because techniques work on unlabeled data
  - Labeled data can be difficult and expensive
- Common techniques:
  - Clustering
    - K-Means
    - Nearest neighbor search
    - Spectral clustering
  - Data projection/preprocessing
    - Principal component analysis
    - Dimensionality Reduction
    - Scaling

Clustering

- Group objects or sets of features such that objects in the same cluster are more similar than those of another cluster
- Optimal clusters should
  - Minimize intra-cluster distance
  - Maximize inter-cluster distance
- Example of intra-cluster measure
  - Squared error se

$$se = \sum_{i=1}^{k} \sum_{p \in c_i} \left\| p - m_i \right\|^2$$

where m_i is the mean of all features in cluster c_i



Dimensionality Reduction

- Process of reducing number of random variables under consideration
  - Key idea: Reduce large dataset to much smaller dataset using only high variance dimensions
- Often used to simplify computation or representation of a dataset
- Typical tasks:
  - Feature Selection: try to find a subset of original variables
  - Feature Extraction: try to represent data in lower dimensions
- Often key to good performance for other machine learning techniques such as regression, classification, etc.
- Other uses:
  - Compression: reduce dataset to smaller representation
  - Visualization: easy to see low dimensional data

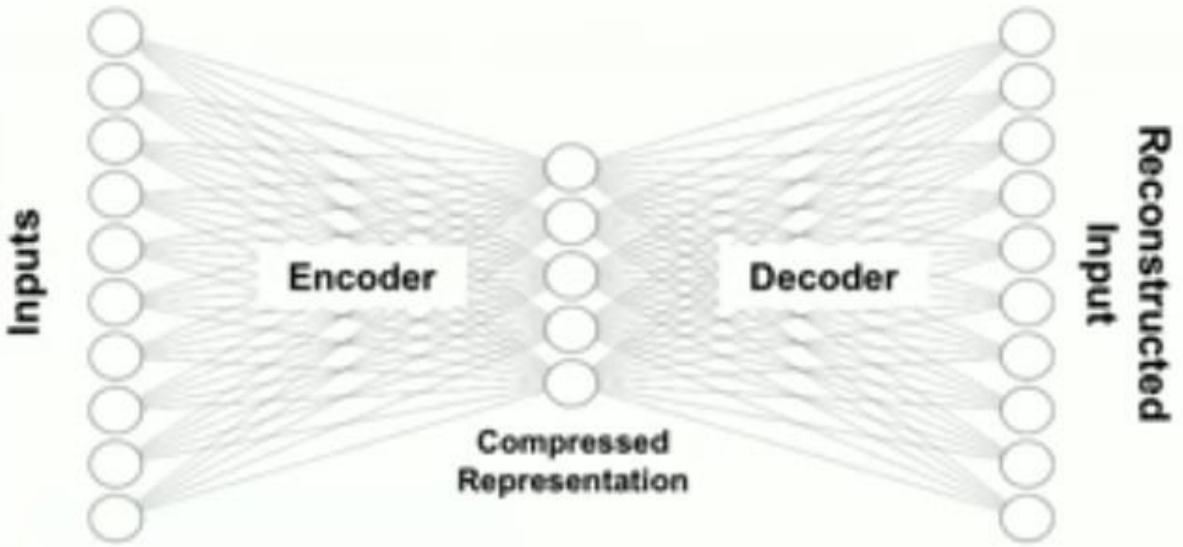
Neural Networks and Unsupervised Learning

- Traditional applications of neural networks such as Image classification fall into the realm of supervised learning:
  - Given example inputs x and target output y, learn the mapping between them.
  - A trained network is supposed to give the correct target output for any input stimulus
  - Training is learning the weights

- Largely used to find better representations for data: clustering and dimensionality reduction
- Non linear capabilities

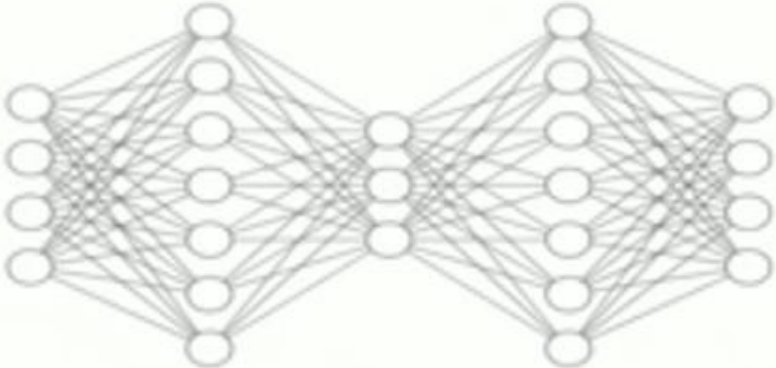
Example: Autoencoders

- Neural Network architecture designed to find a compressed representation for data
- Feedforward, multi layer perceptron.
- Input layer number of features = output layer number of features
- Similar to dimensionality reduction but allows for much more complex representations

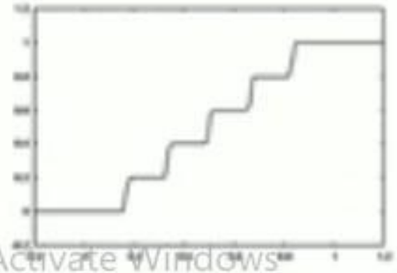


Example: Replicator Neural Network

- Conceptually, very similar to autoencoders
- Used extensively for anomaly detection (looking for outliers)
- Example architecture



- Salient differences from an autoencoder: Step Activation Function, Inclusion of dropout layers
  - Step activation squeezes the middle layer outputs into a number of clusters
  - Dropout layers help with overfitting



Step Activation Function

Hawkins, Simon, et al. "Outlier detection using replicator neural networks." *International Conference on Data Warehousing and Knowledge Discovery*. Springer, Berlin, Heidelberg, 2002.

Reinforcement Learning

- What makes reinforcement learning different from other machine learning paradigms?
  - There is no supervisor, only a reward signal
  - Feedback is delayed, not instantaneous
  - Time really matters (sequential, often inter-dependent data)
  - Agent's actions affect the subsequent data it receives

- Example: Playing Atari game
  - Rules of the game are unknown
  - Learn directly from interactive game-play
  - Pick actions on joystick, see pixels and scores

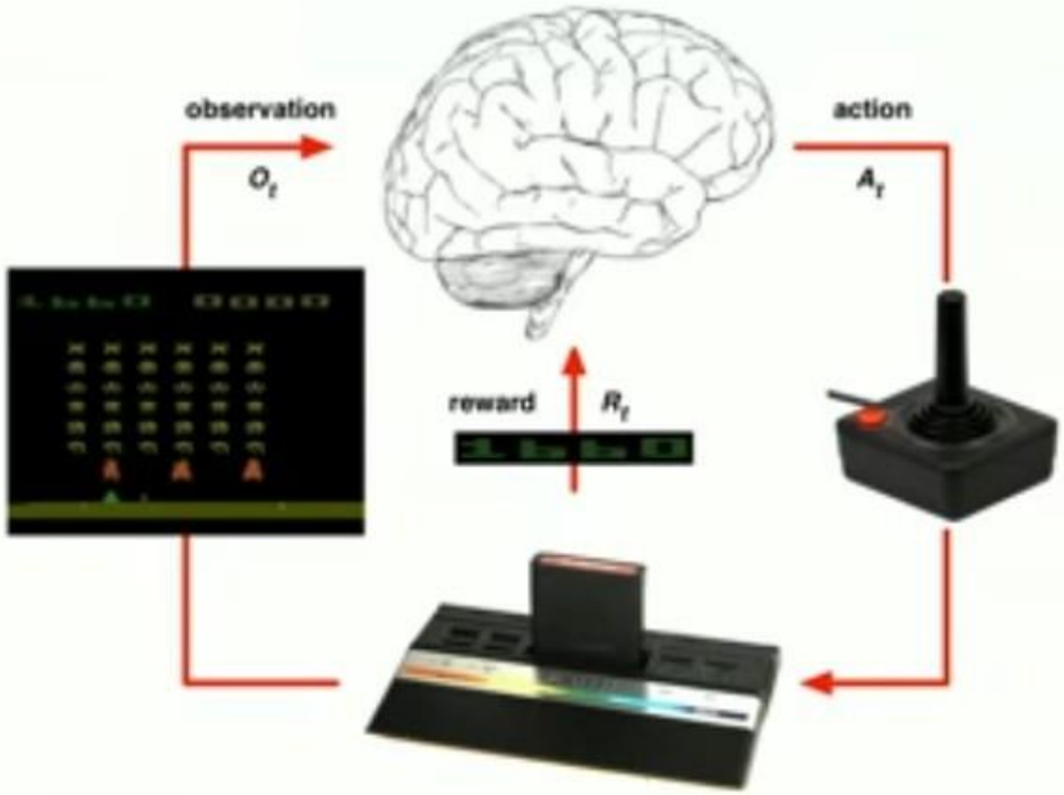


Image Source: "Introduction to Reinforcement Learning" by David Silver (Google DeepMind)

Other Reinforcement Learning Examples

- Fly stunt maneuvers in a helicopter
  - + reward for following desired trajectory
  - − reward for crashing
- Defeat the world champion at Backgammon
  - +/− reward for winning/losing a game
- Manage an investment portfolio
  - + reward for each $ in bank
- Control a power station
  - + reward for producing power
  - − reward for exceeding safety thresholds
- Make a humanoid robot walk
  - + reward for forward motion
  - − reward for falling over