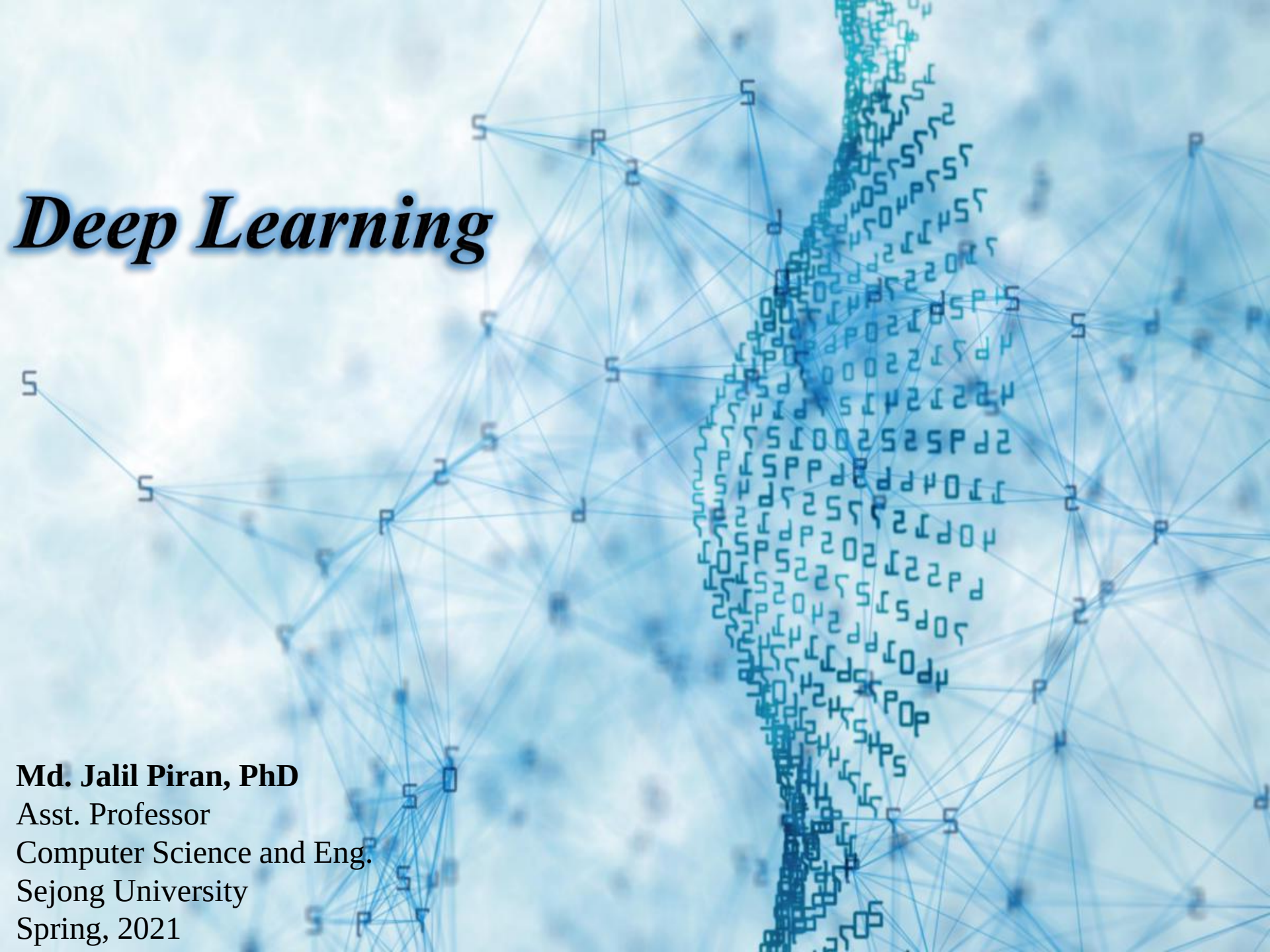


Deep Learning

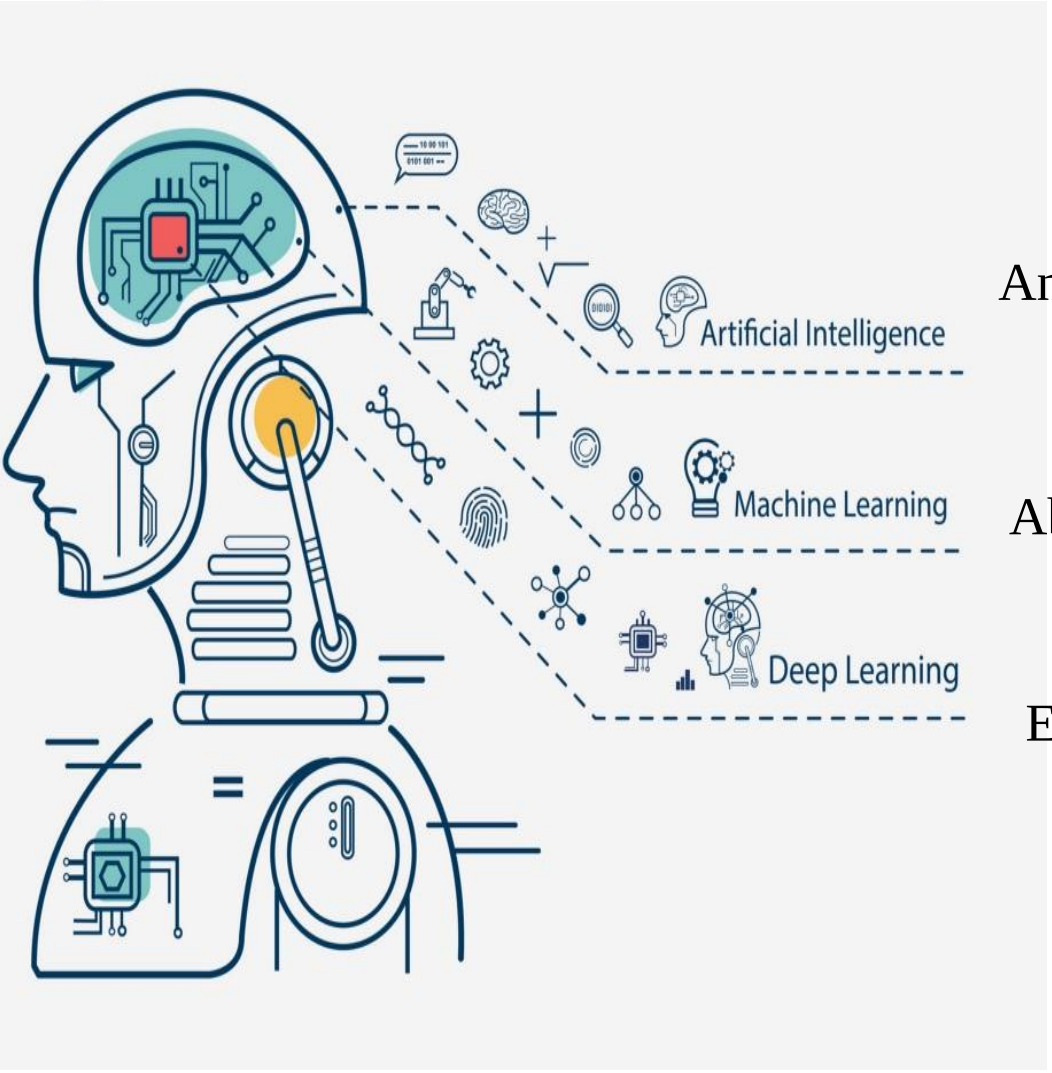
Md. Jalil Piran, PhD
Asst. Professor
Computer Science and Eng.
Sejong University
Spring, 2021

Lecture Type

WebEx	Blackboard	Face-to-face
0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0
3	13	3

INTRODUCTION OF DEEP LEARNING

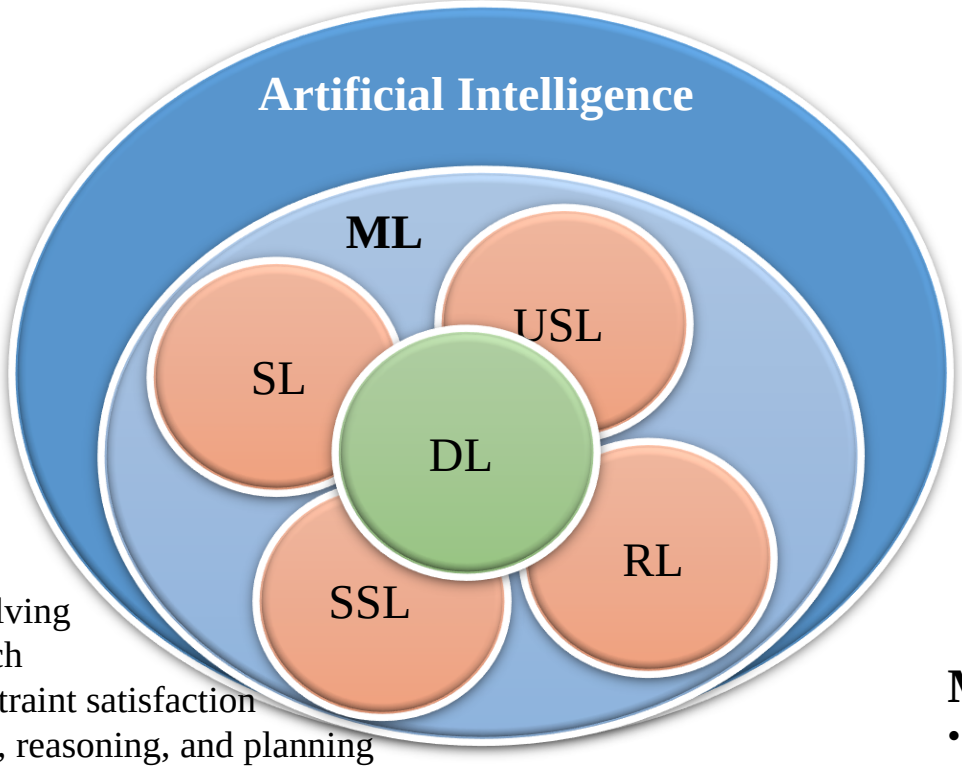
Artificial Intelligence



Any technique that enables computers to mimic human behavior.

Ability to learn without explicitly being programmed.

Extract patterns from data using neural networks.



AI Branches

- Expert systems
- Closed-loop control systems
- Case-based reasoning
- Fuzzy interference system
- Learning-based systems

AI Techniques

- Problem solving
  - Search
  - Constraint satisfaction
- Knowledge, reasoning, and planning
  - Logical agents
  - First-order logic
  - Planning and acting
  - Knowledge representation
  - Probabilistic reasoning
  - Decision making
- Learning
  - Learning from examples
  - Knowledge in learning
  - Learning probabilistic models
- Communications, perceiving, and acting
  - Natural language processing
  - Perception
  - Robotics
- …

ML

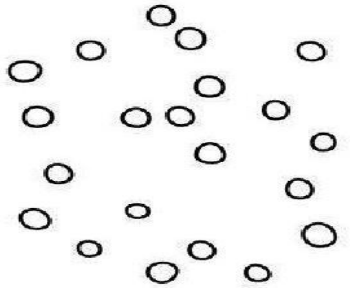
- Linear Regression
- Logistic Regression
- Bayesian Networks
- Clustering
- Decision Tree Learning
- Genetic Algorithm
- Inductive Logic Programming
- Representation Learning
- Rule-based ML
- Similarity and Metric Learning
- Sparse Dictionary Learning
- Support Vector Machine
- Q-Learning
- Deep Learning
- …

…

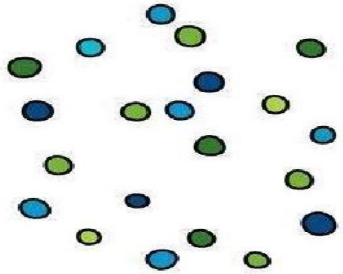
Data

Machine Learning: Data + Model + Optimization

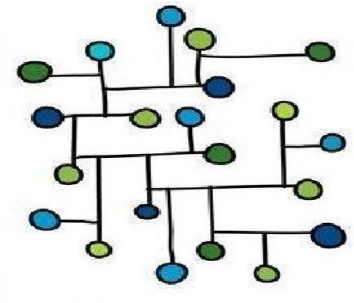
Data



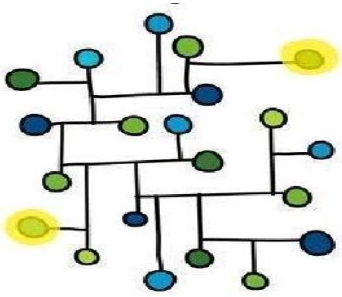
Information



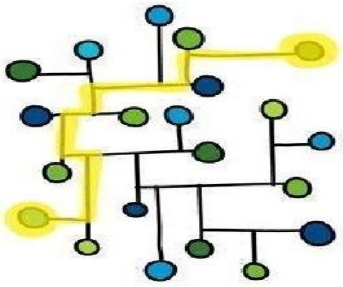
Knowledge



Insight



Wisdom

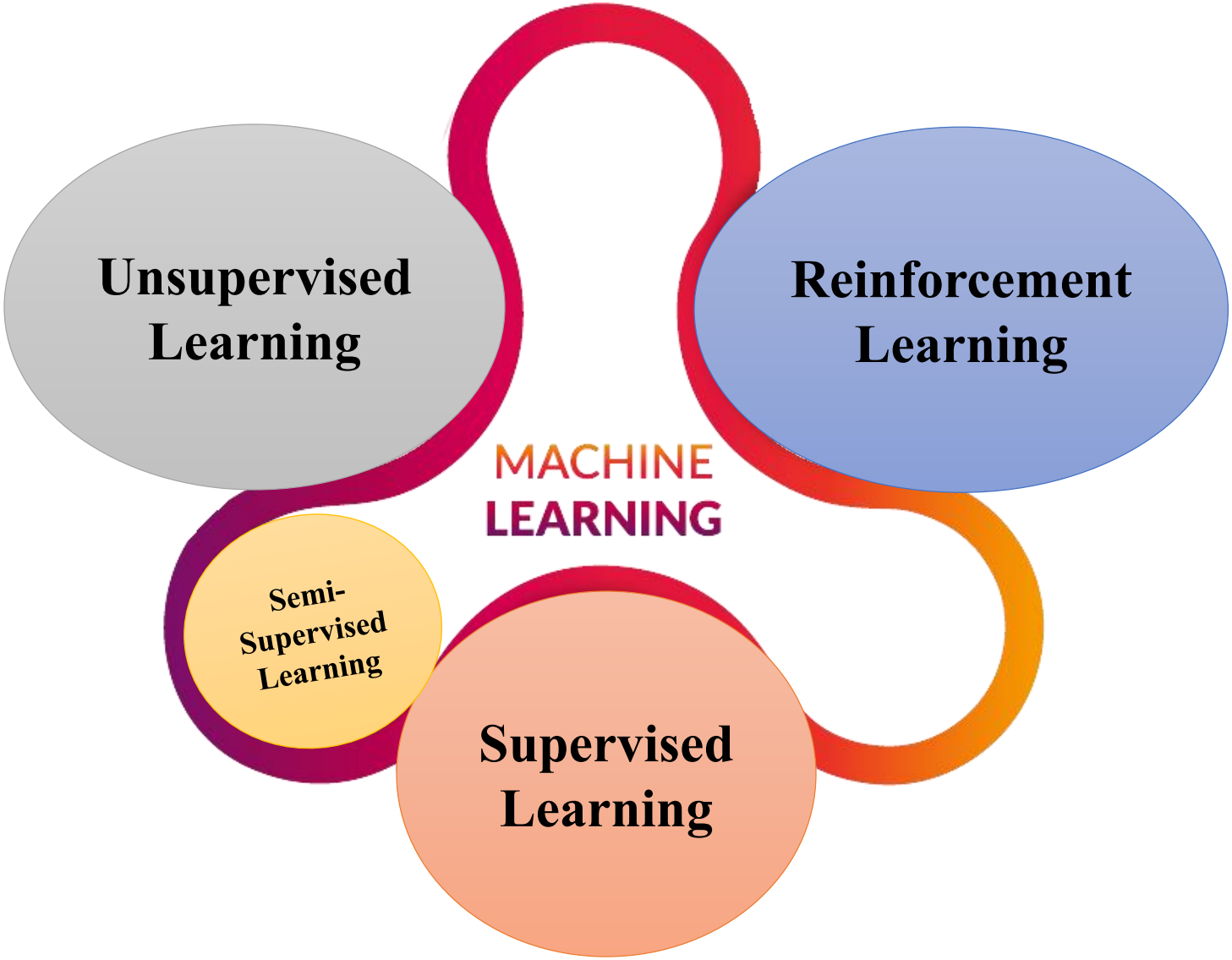


Impact

DL Beginner Checklist

- **Personality**
  - Interest in AI
  - Curiosity
  - Patience
- **Mathematics**
  - Linear algebra
  - Statistics
- **Data**
  - Data preparation
  - Data scaling
- **Programming**
  - Python and related tools.
- **Neural networks**

Types of Learning



Potential Tasks

- **Classification** – assigning a category to each item
- **Regression** – predicting a value for each item
- **Ranking** – learning to order items.
- **Clustering** – partitioning items into homogeneous subsets
- **Dimensionality reduction** (manifold learning) –
  - Transforming a representation of items into a
  - Lower-dimensional one
- **Anomaly Detection**

Types of Machine Learning

- **Supervised Learning (SL)**
  - Learn a mapping from the input to an output using a set of labeled examples
- **Unsupervised Learning (UL)**
  - Find the regularities of data using a set of unlabeled examples
- **Semi-supervised Learning (SSL)**
  - The training sample consisting of both labeled and unlabeled data

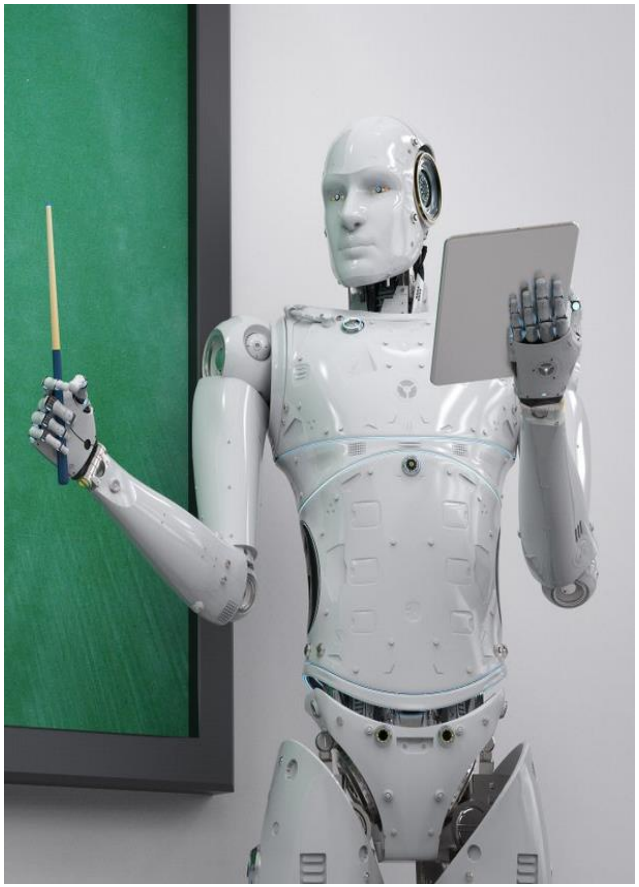
Types of Machine Learning

- **Reinforcement Learning (RL)**
  - During learning, the correct answers are not provided but hints or delayed rewards
- **On-line Learning**
  - Training and testing phases are intermixed.
- **Active Learning**
  - The learner adaptively or interactively collects training examples

SL

- **Model** relationships and dependencies between the target prediction output and the input features

- Used to **predict** the output values for new data based on those relationships which it learned from the previous data sets.



SL

- The majority of practical machine learning uses SL.

$$Y = f(X)$$

X: input variables

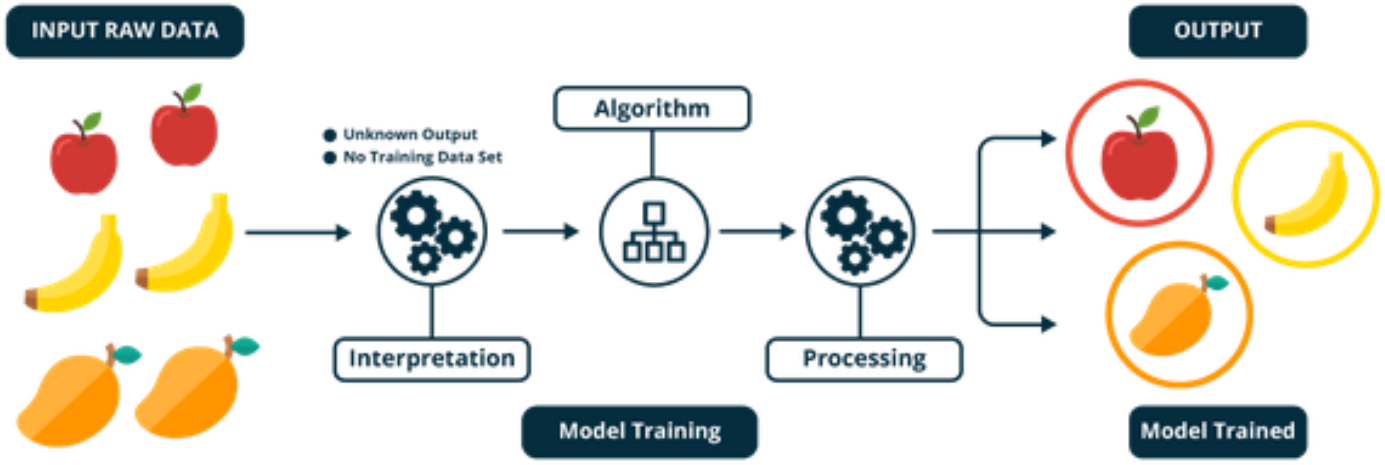
Y: an output variable

$f(.)$: an algorithm to learn the mapping function from the input to the output.

- **Goal**: to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.

SL

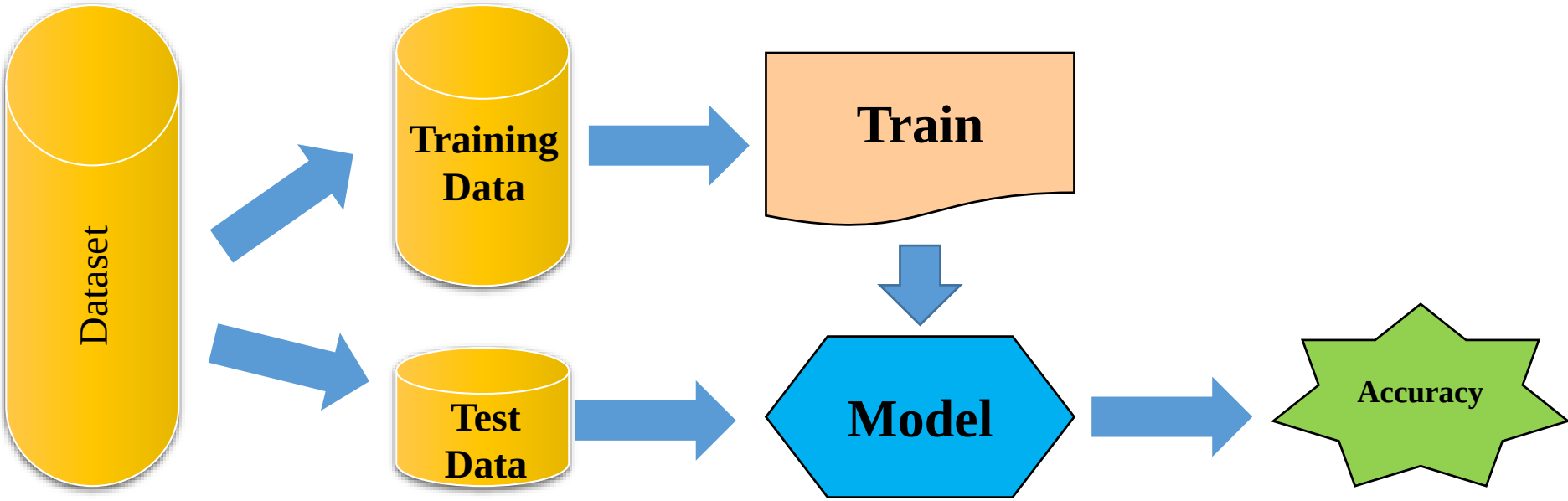
- SL enable machines to **classify** / **predict** objects, problems or situations based on labeled data fed to the machine.



SL process: two steps

- **Learning (training):** Learn a model using the **training data**
- **Testing:** Test the model using **unseen test data** to assess the model accuracy

$$Accuracy = \frac{Number\ of\ correct\ classification}{total\ number\ of\ test\ cases}$$



What is Learning?

- Given
  - a **data set** D,
  - a **task** T, and
  - a **performance measure** M,

- A computer system is said to **learn** from ***D*** to perform the task ***T*** if after learning the system's performance on ***T*** improves as measured by ***M***.

- In other words, the learned model helps the system to perform ***T*** better as compared to no learning.

SL problems

- **Classification**: when the output variable is a category,
  - such as “red” or “blue” or “disease” and “no disease”.

- **Regression**: when the output variable is a real value,
  - such as “dollars” or “weight”.

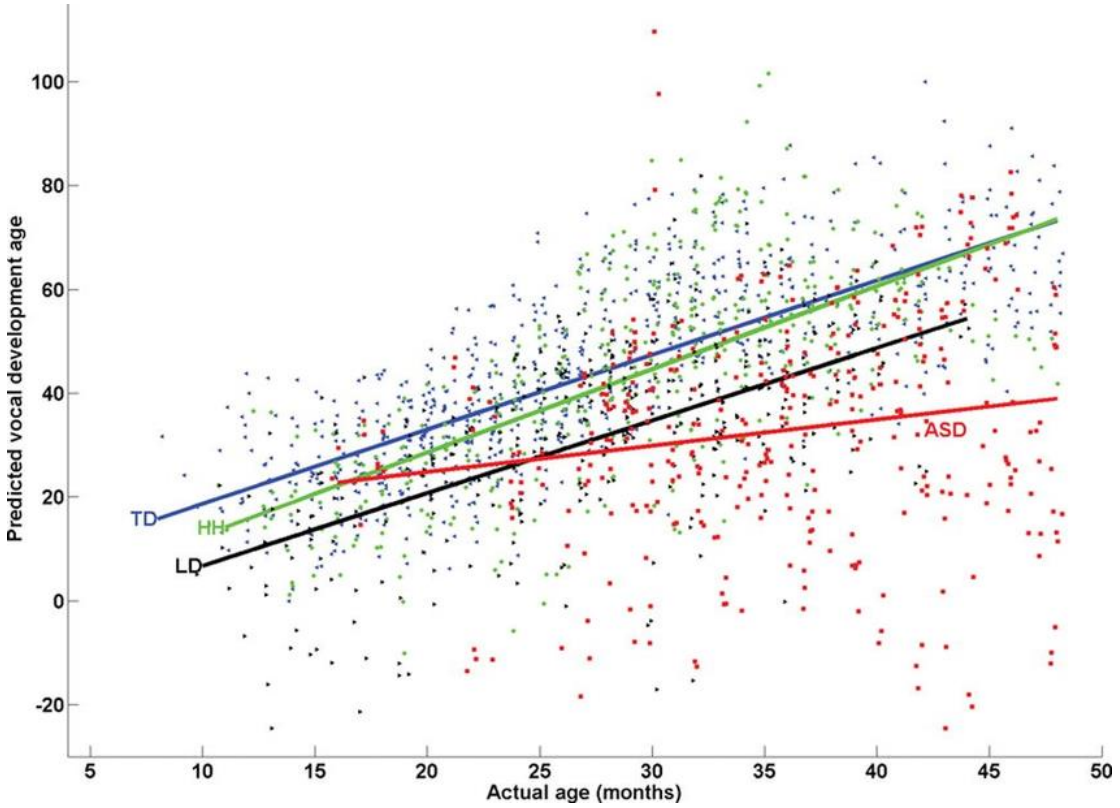
- Some common types of problems built on top of classification and regression include **recommendation** and **time series prediction** respectively.

Algorithms

- **Regression**
- **Random forest** for classification and regression problems
- **Support vector machines** for classification problems
- **Artificial neural networks** (ANN)
- **Deep neural networks (DNN)**
- **Linear discriminant analysis**
- **Decision trees**
- **Similarity learning**
- **Bayesian logic**
- **Supervised classifier**
- **Probabilistic Learning**

Linear Regression

- Regression **maps** an input to an output based on example input-output pairs,
- It predicts **continuous** valued output.



Classification

- Given a collection of records (**training set**)
  - Each record contains a set of **attributes**, one of the attributes is the **class**.

- Find a **model** for class attribute as a function of the values of other attributes.

SL

Many classifiers to choose;

- Logistic regression
- Neural networks
- Naïve Bayes
- Bayesian network
- SVM
- Randomized Forests
- Boosted Decision Trees
- K-nearest neighbor
- RBMs
- …

UL

- UL algorithms try to use techniques on the input data to **mine for rules**, **detect patterns**, and **summarize and group the data points**

- UL help in deriving meaningful insights and describe the data better to the users.

Learn patterns from (unlabeled) data.

- **UL** is used when only have input data (X) and no corresponding output variables.

UL

- **Problems**

- **Clustering**: where you want to discover the inherent groupings in the data, such *as grouping customers by purchasing behavio*r.

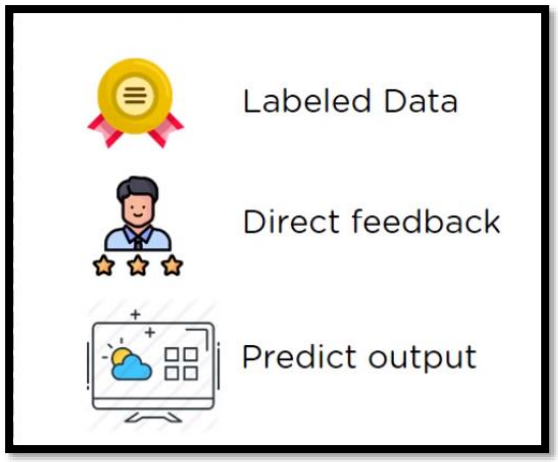
- **Association**: where you want to discover rules that describe large portions of your data, *such as people that buy X also tend to buy Y*.

Approaches

- Clustering (similarity-based)
- Density estimation (e.g., EM algorithm)
- Performance Tasks
- Understanding and visualization
- Anomaly detection
- Information retrieval
- Data compression

Comparison

Supervised vs. Unsupervised



Algorithms

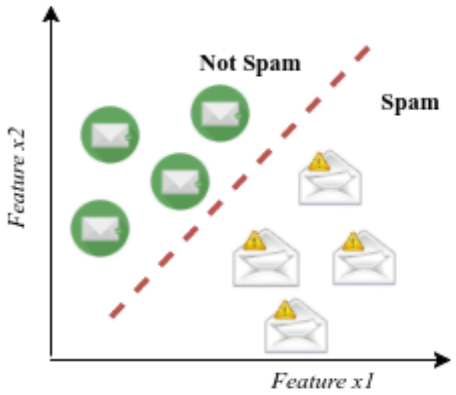
- K-means clustering,
- Hierarchical clustering,
- Unsupervised soft-clustering,
- Affinity propagation clustering
- Self-organizing map learning
- Autoencoders
- Adversarial autoencoders
- Non-parametric Bayesian Learning
- Generative Deep Neural networks (GDNN)
- Apriori,
- PCA
- Mixture model
- Gaussian mixture model, Expectation Maximization (EM)
- Dirichlet process

Comparison

Regression	Classification	Clustering
SL	SL	UL
map an input to an output based on example input-output pairs		partitioning the dataset into groups, e.g. clusters
predicts continuous valued output	predicts discrete number of values	Split up the data in such a way that points within single cluster are very similar and points in different clusters are different.
used to predict the numeric data instead of labels.	data is categorized under different labels	determines grouping among unlabeled data.
identify the distribution trends based on the available data or historic data	the labels are predicted for the data.	

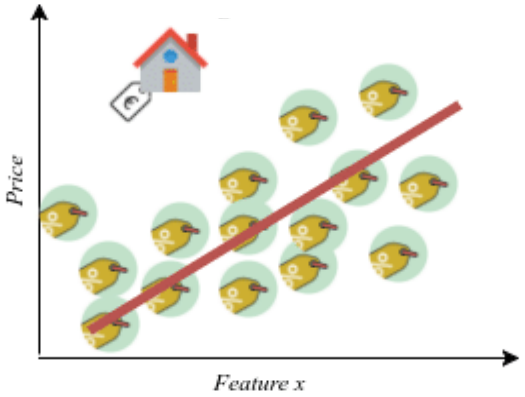
UL

Classification



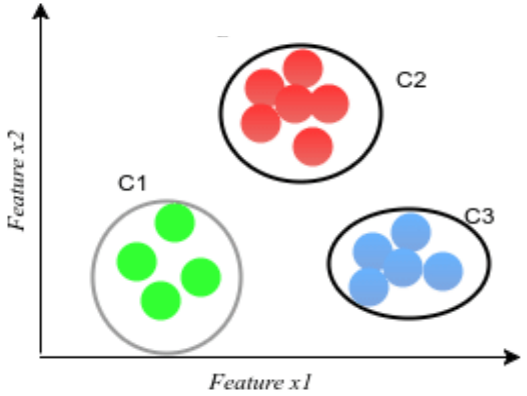
Spam filtering as a classification task

Regression



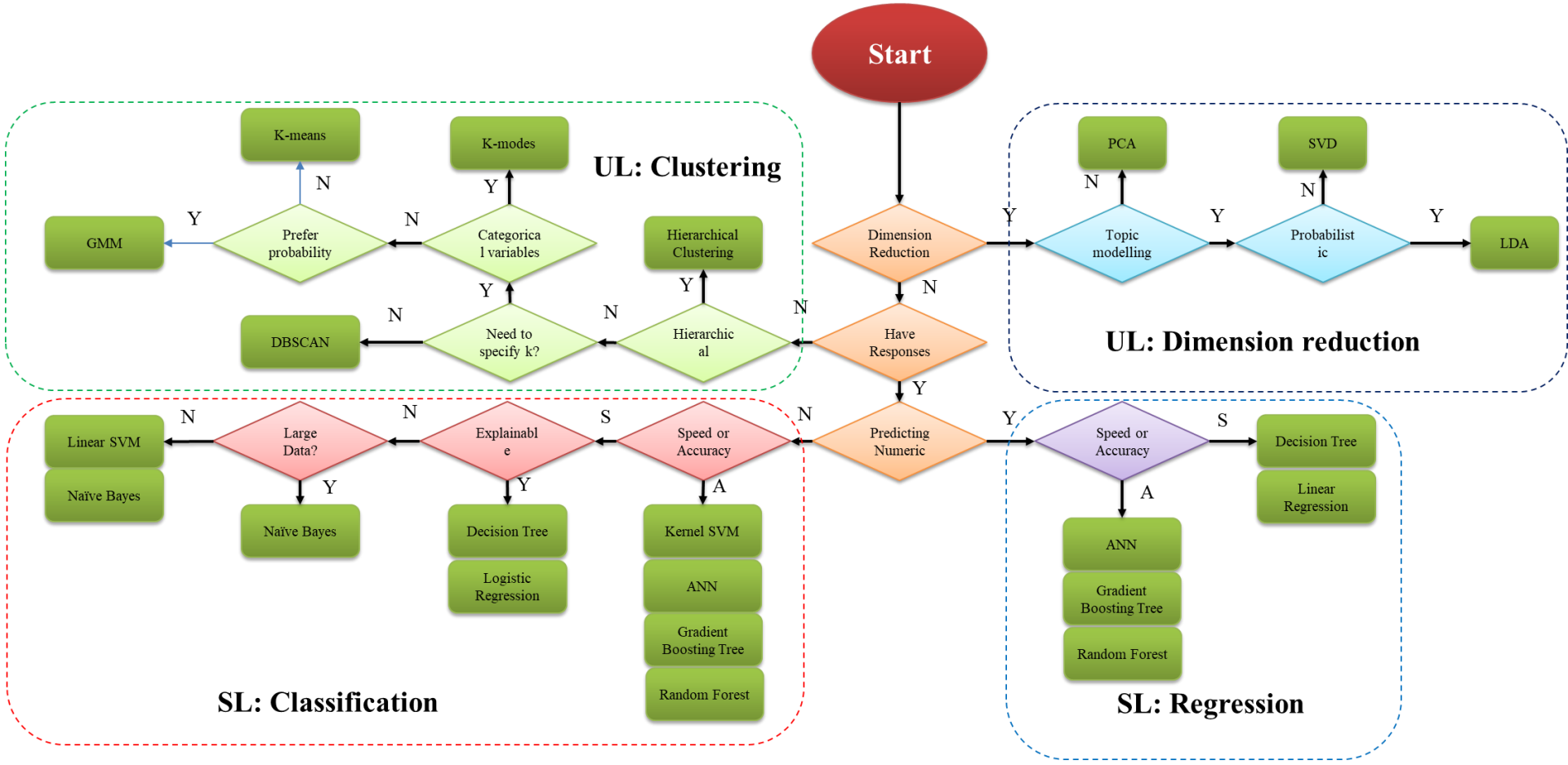
House price estimation as a regression task

Clustering



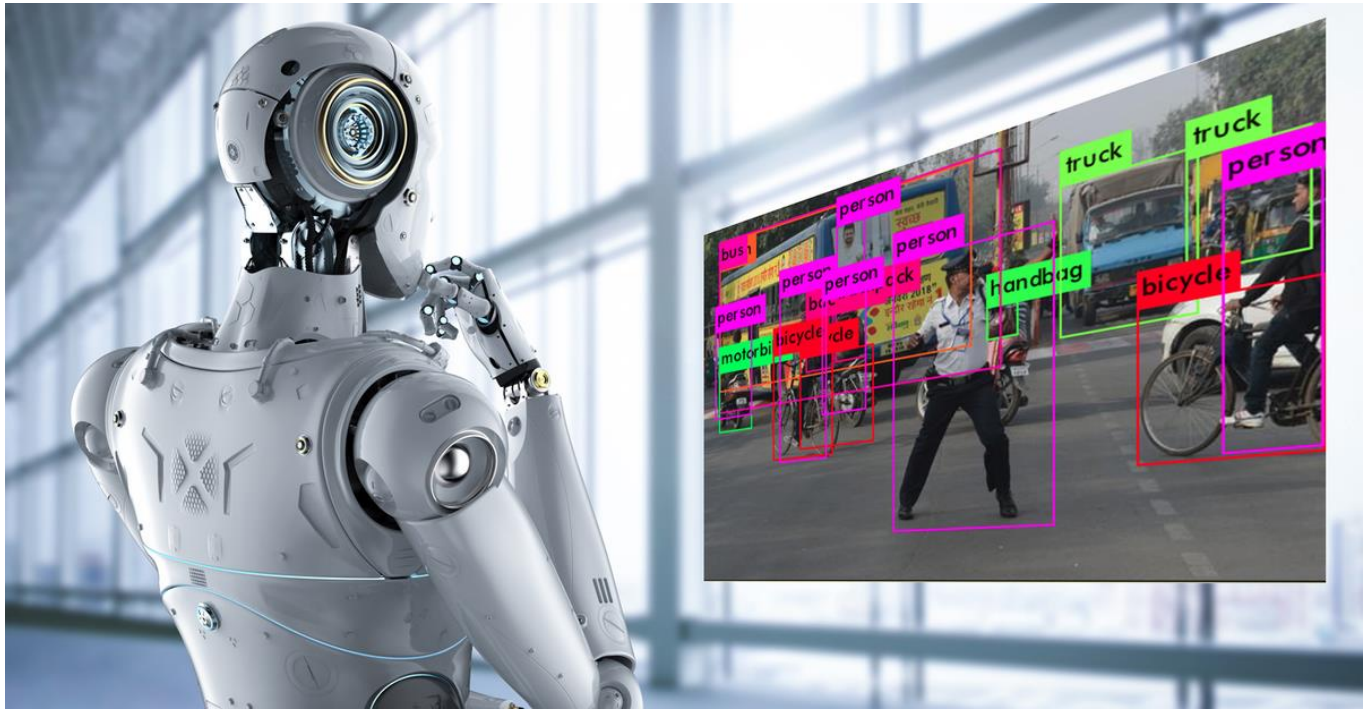
customers are grouped into three different categories based on their purchasing behavior.

SL and UL Cheat-sheet



SSL

- A training dataset with both **labeled** and **unlabeled** data.
- This method is particularly useful when extracting relevant features from the data is difficult, and labeling examples is a time-intensive task for experts.



SSL

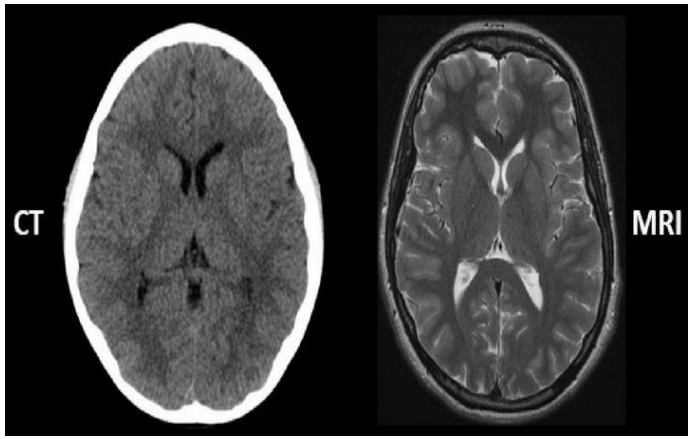
- **Problems**: where you have a large amount of input data (X) and only some of the data is labeled (Y).

- These problems sit in between both SL and UL.

- A good example is a photo archive where only some of the images are labeled, (e.g. dog, cat, person) and the majority are unlabeled.

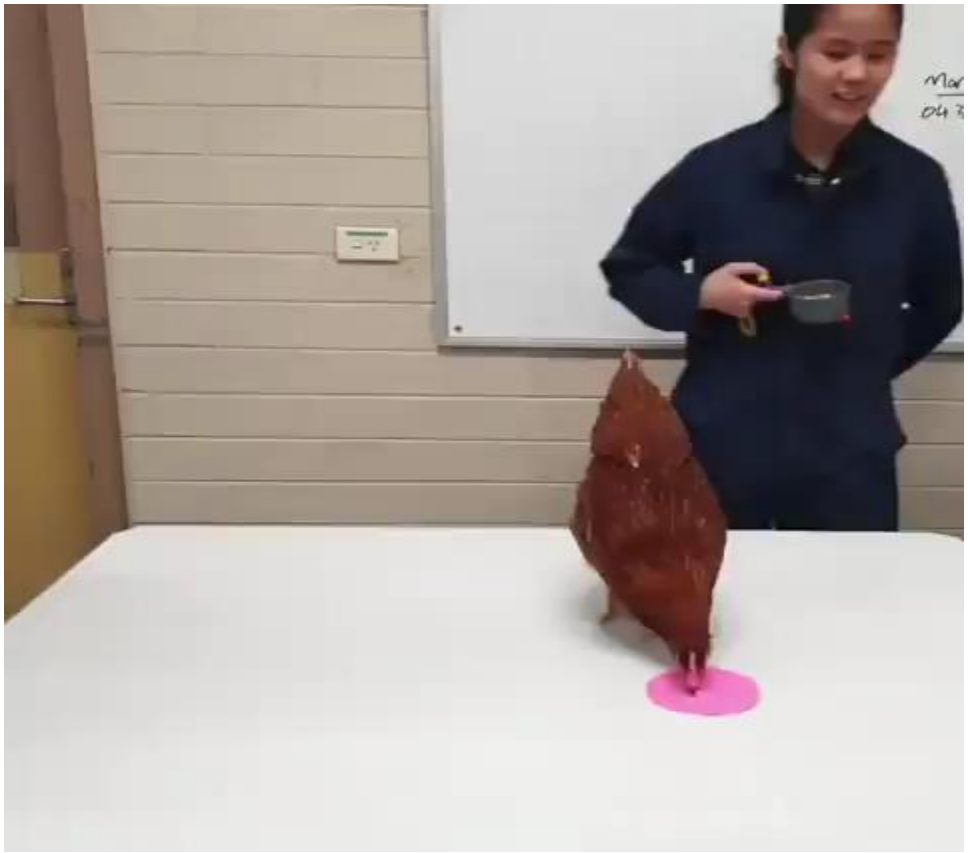
Applications

- **Medical images**
- e.g. CT scans or MRIs
- A trained radiologist can go through and label a small subset of scans for tumors or diseases.
- It would be too time-intensive and costly to manually label all the scans
- but the DL network can still benefit from the small proportion of labeled data and improve its accuracy compared to a fully unsupervised model.



Reinforcement learning

- An agent learns how to behave in an environment by performing actions and seeing the results.



RL

- **Video games** are full of reinforcement cues.
- Complete a level and earn a badge.
- Defeat the bad guy in a certain number of moves and earn a bonus. Step into a trap — game over.
- These cues help players learn how to improve their performance for the next game.
- Without this feedback, they would just take random actions around a game environment in the hopes of advancing to the next level.

Comparison

SL

- Labeled dataset
- Establish relationship between input and output
- Generate output for new data points
- Reliable models but expensive and limited

- Classification: Associative classifiers, Decision Trees, Instance Learning, Bayesian Learning, Kernel machines, Neural Networks, Genetic Algorithms, etc.

- Regression: Linear Regression, …

UL

- Unlabeled dataset
- Decipher structure of the data
- Output attributes are not defined
- Clustering: K-means, DBScan, Hierarchical algorithms, Self Organizing Maps, etc.
- Associations: Apriori, FP-Growth, …

RL

- Maximizing the rewards from the results
- Aka. credit assessment learning
- Additional decision about rewards
- Explore the tradeoff between exploring and exploiting the data

Steps

- In order to produce intelligent programs (also called agents), reinforcement learning goes through the following steps:
  - **Input state** is observed by the agent.
  - **Decision making function** is used to make the agent perform an action.
  - After the action is performed, the agent receives **reward** or reinforcement from the environment.
  - The **state-action pair information** about the reward is stored.

Algorithms

- Q-Learning
- MDP / POMDP
- Temporal Difference (TD)
- Deep Adversarial Networks
- Multi-armed bandit
- Actor-critic
- Deep reinforcement learning (DRL)

ML Algorithms Cheat Sheet

What do you want to do?

Extract information from text

Text Analytics

Derives high-quality information from text

Answers questions like: What info is in this text?

- **Extract N-Gram Features from Text** ← Creates a dictionary of n-grams from a column of free text
- **Feature Hashing** ← Converts text data to integer encoded features using the Vowpal Wabbit library
- **Preprocess Text** ← Performs cleaning operations on text, like removal of stop-words, case normalization
- **Word2Vector** ← Converts words to values for use in NLP tasks, like recommender, named entity recognition, machine translation

Predict values

Regression

Makes forecasts by estimating the relationship between values

Answers questions like: How much or how many?

- **Fast Forest Quantile Regression** ← Predicts a distribution
- **Poisson Regression** ← Predicts event counts
- **Linear Regression** ← Fast training, linear model
- **Bayesian Linear Regression** ← Linear model, small data sets
- **Decision Forest Regression** ← Accurate, fast training times
- **Neural Network Regression** ← Accurate, long training times
- **Boosted Decision Tree Regression** ← Accurate, fast training times, large memory footprint

Generate recommendations

Recommenders

Predicts what someone will be interested in

Answers the question: What will they be interested in?

- **SVD Recommender** ← Collaborative filtering, better performance with lower cost by reducing dimensionality

Discover structure

Clustering

Separates similar data points into intuitive groups

Answers questions like: How is this organized?

- **K-Means** ← Unsupervised learning

Find unusual occurrences

Anomaly Detection

Identifies and predicts rare or unusual data points

Answers the question: Is this weird?

- **One Class SVM** ← Under 100 features, aggressive boundary
- **PCA-Based Anomaly Detection** ← Fast training times

Predict between several categories

Multiclass Classification

Answers complex questions with multiple possible answers

Answers questions like: Is this A or B or C or D?

- **Multiclass Logistic Regression** ← Fast training times, linear model
- **Multiclass Neural Network** ← Accuracy, long training times
- **Multiclass Decision Forest** ← Accuracy, fast training times
- **One-vs-All Multiclass** ← Depends on the two-class classifier
- **Multiclass Boosted Decision Tree** ← Non-parametric, fast training times and scalable

Predict between two categories

Two-Class Classification

Answers simple two-choice questions, like yes or no, true or false

Answers questions like: Is this A or B?

- **Two-Class Support Vector Machine** ← Under 100 features, linear model
- **Two-Class Averaged Perceptron** ← Fast training, linear model
- **Two-Class Decision Forest** ← Accurate, fast training
- **Two-Class Logistic Regression** ← Fast training, linear model
- **Two-Class Boosted Decision Tree** ← Accurate, fast training, large memory footprint
- **Two-Class Neural Network** ← Accurate, long training times

Classify images

Image Classification

Classifies images with popular networks

Answers questions like: What does this image represent?

- **DenseNet** ← High accuracy, better efficiency

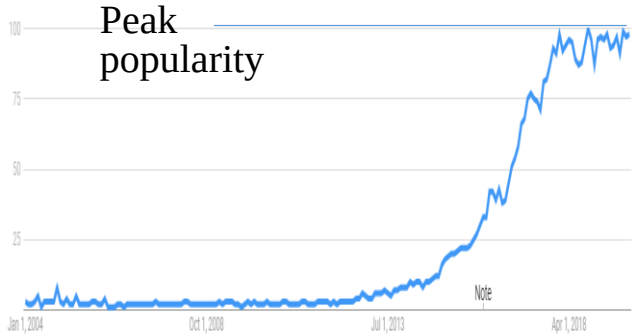
In nutshell…

- **What?**
  - *Extract useful patterns from data.*
- **Why?**
  - *To reduce human intervention.*
- **How?**
  - *Data + ANN + Optimization.*
- **Tools?**
  - *Python + TensorFlow & Friends*
- **Hard part:**
  - *“Good questions” + “clean and useful data”*
- **Why so popular?**
  - *Data, HW, community, tools, investment, efficient algorithms for complex problems*
- **Current status?**
  - *Most big question of intelligence have not been answered nor properly formulated!*

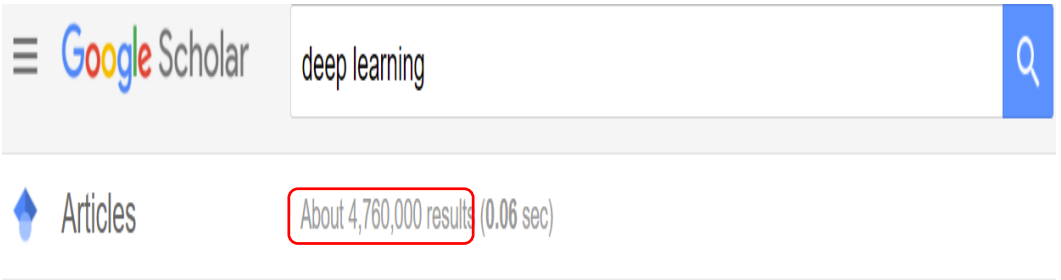
…

DL Popularity

- **Popularity**



- **Number of papers**



- **Interest by region**

1	South Korea	100
2	Singapore	54
3	St. Helena	49
4	Hong Kong	41
5	China	37

Values are calculated on a scale from 0 to 100, where 100 is the location with the most popularity as a fraction of total searches in that location.

• • •

Advances in Computer Vision

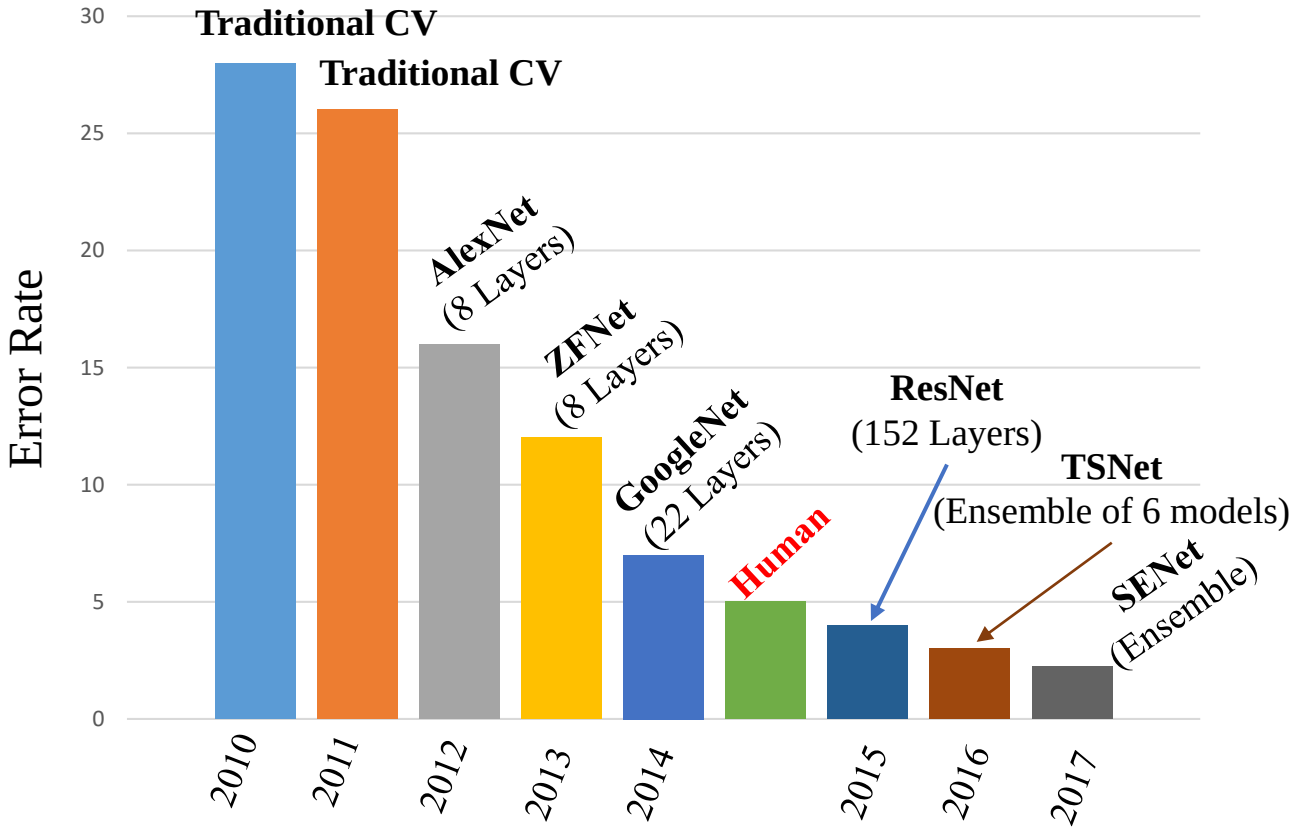
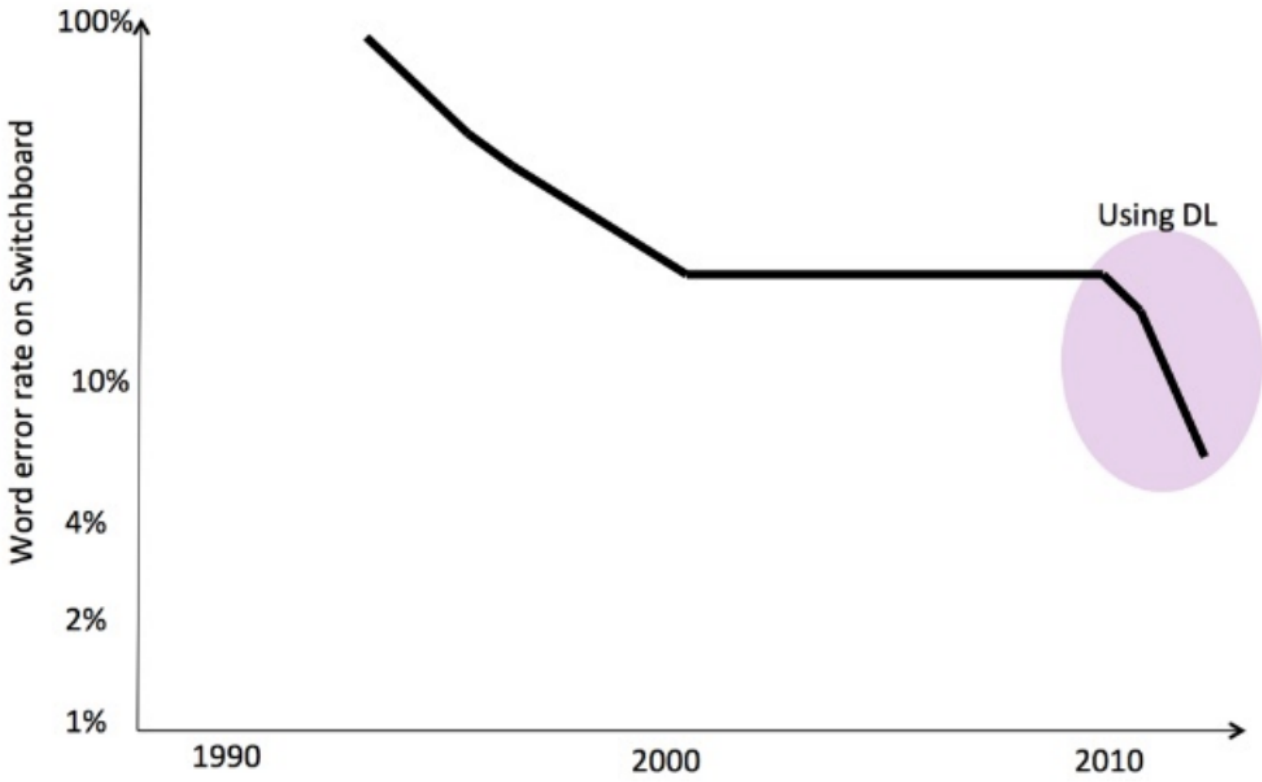


Image classifiers have surpassed human level accuracy.

...

Advances in Speech Recognition



DL improved the accuracy of speech recognition significantly!

…

DL History

https://machinelearningknowledge.ai/brief-history-of-deep-learning/

...

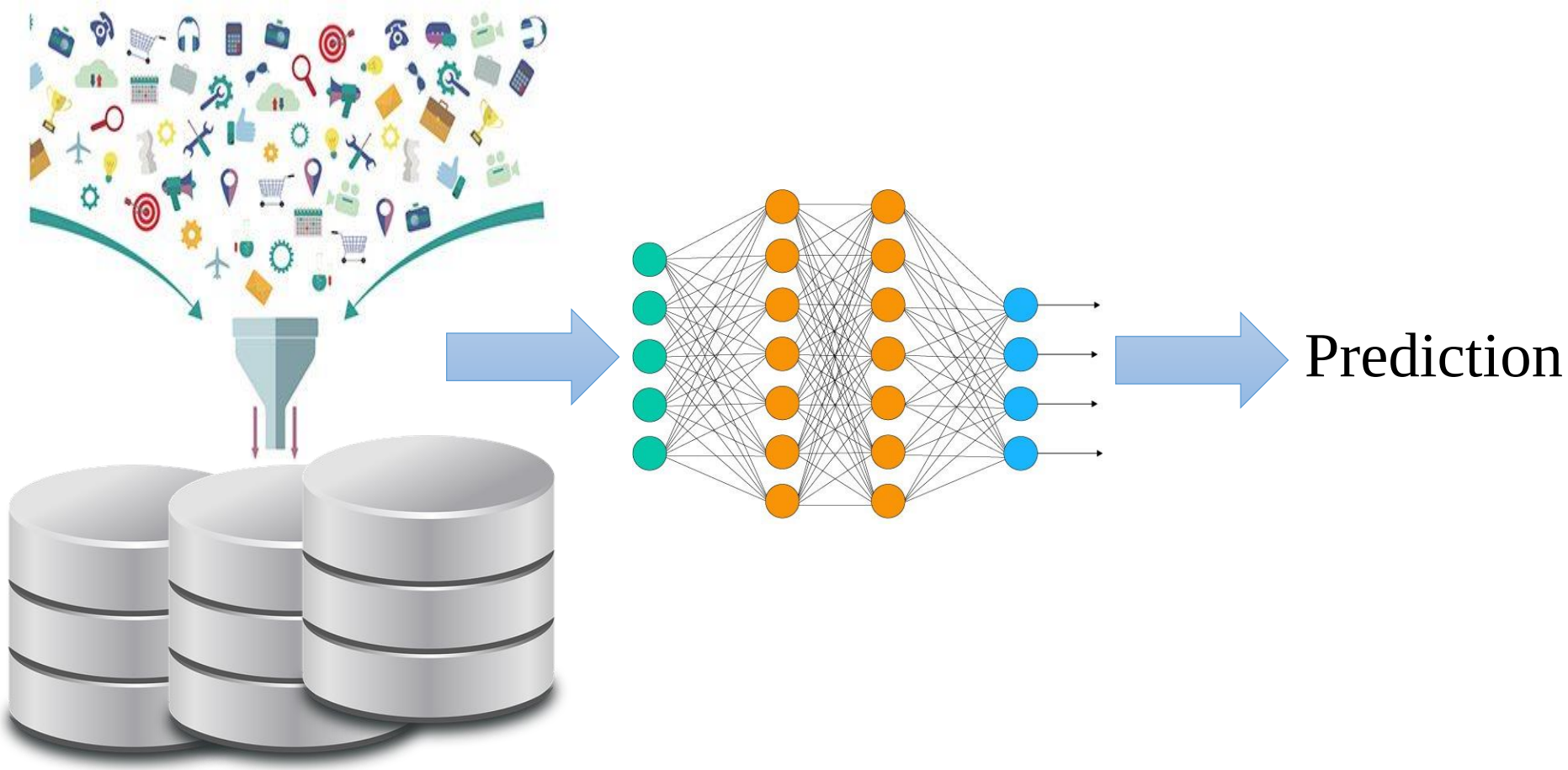
DL Popularity

- Why currently become so hot and popular?

DL Popularity

Data

- Availability of “Digitized Data”:

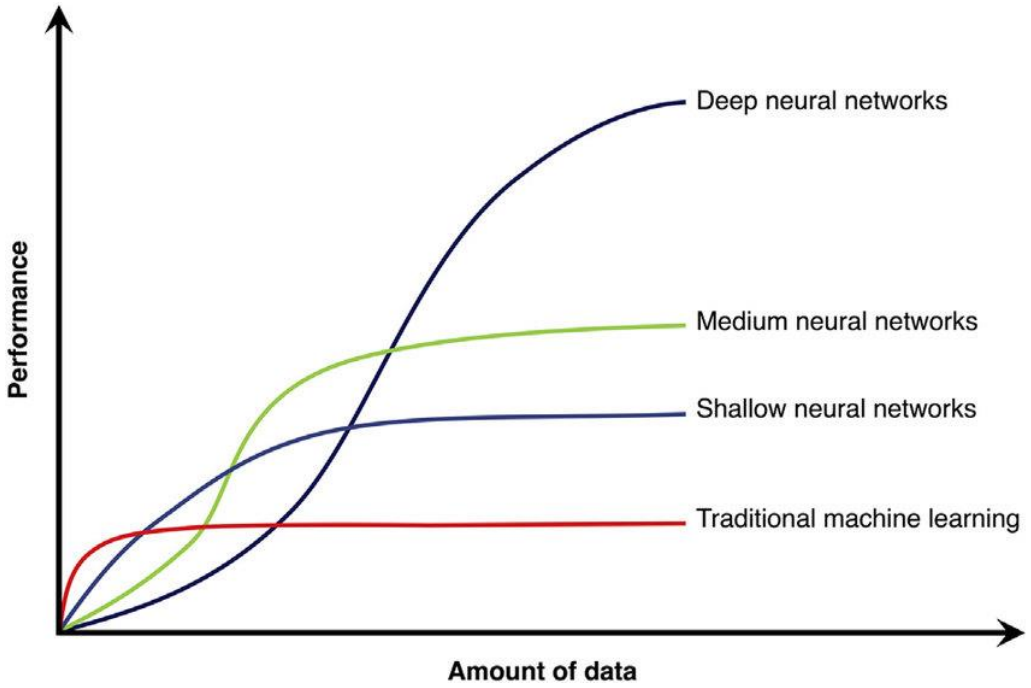


…

DL Popularity

Data

- The amount of data has more impact on the performance of DL compared to traditional ML algorithms.

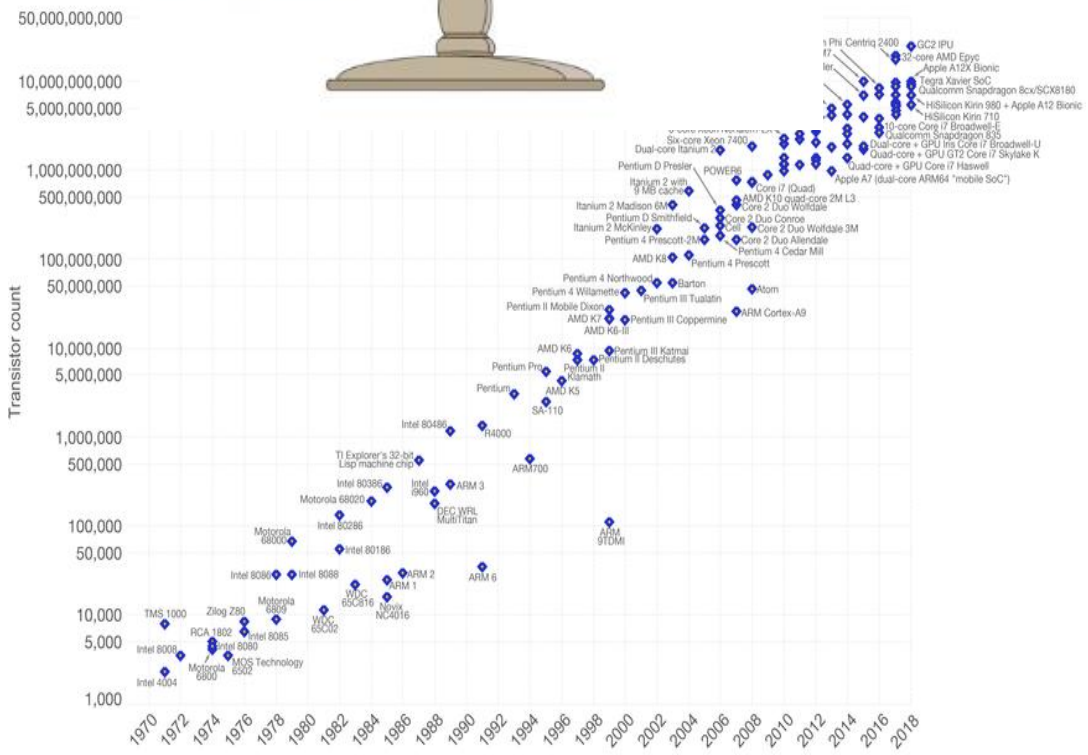


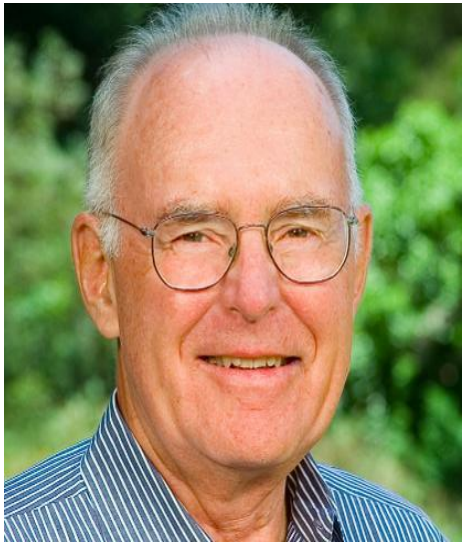
- Time increases based on the amount of data and the number of hidden layers.

...

DL Popularity

Computational power





"The number of transistors in a dense integrated circuit doubles about every two years."

- Gordon Moore

...

DL Popularity

- **CPU**: serial, general purpose, everyone has one.
- **GPU**: parallelizable, still general purpose.
- **Cloud Tensor Processing Units** (**TPUs**): custom application-specific integrated circuit (ASIC) by Google, specialized for ML.

https://cloud.google.com/tpu/docs/tpus

…

DL Popularity

Computational power

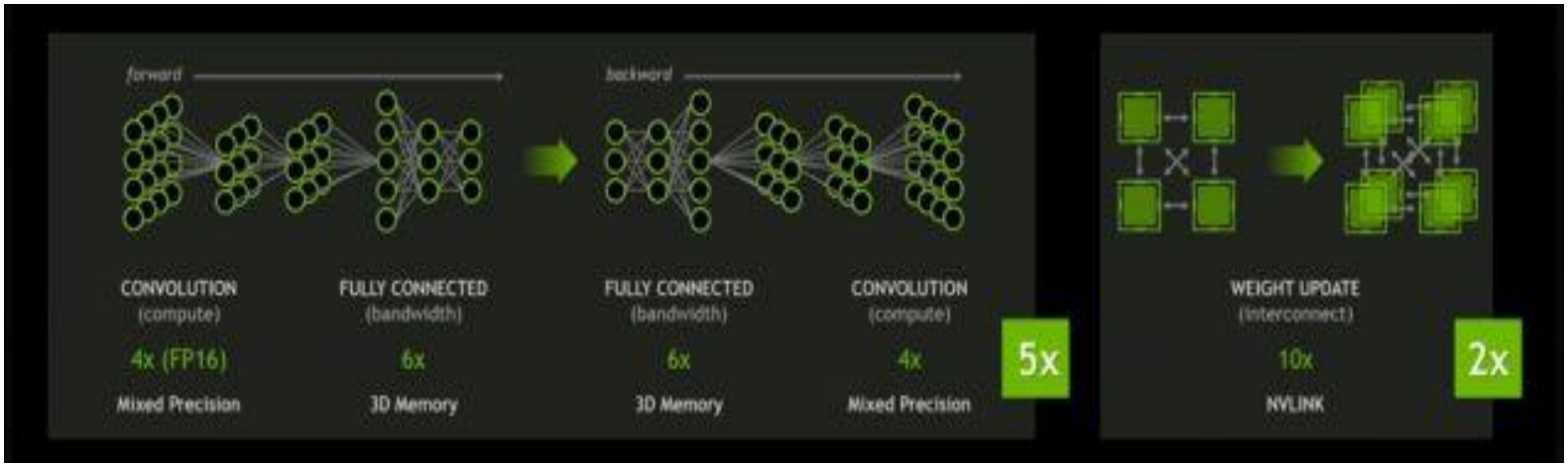
- The **Digits DevBox** for customized DL;
  - The world's best hardware (4 GPUs),
  - The best Software,
  - Can fit under a desk,
  - Cost $15,000.

...

DL Popularity

Computational power

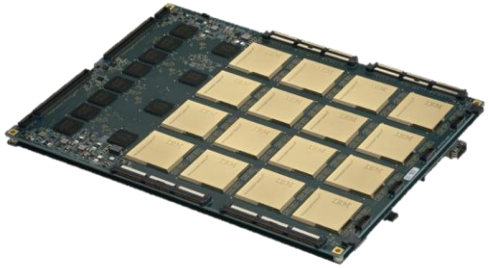
- **NAVIDI's Pascal GPU;**
  - Accelerate DL applications up to 10x beyond the speed of its current-generation Maxwell processors.



...

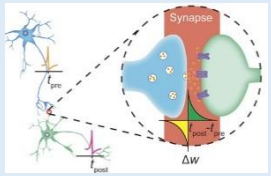
DL Popularity

Computational power

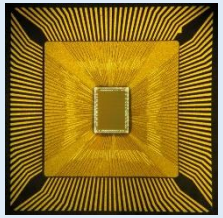


- **IBM TrueNorth**
  - A brain-inspired computer chip.
  - Implements networks of integrated and fire spiking artificial neurons.
  - Uses only a tiny 70 mw of power – orders of magnitude less energy than traditional chips.

1 million
Programmable
Neurons

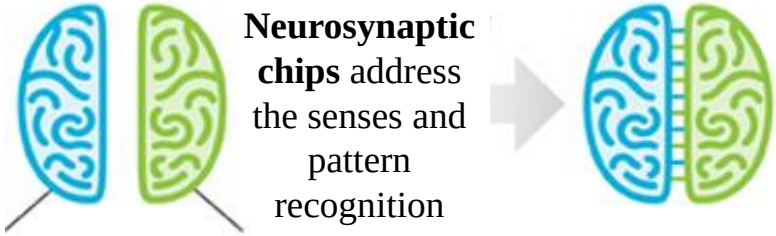


256 million
Programmable
Synapses



4096 Neurosynaptic Cores

Traditional computers focus on language and analytical thinking (**left brain**)



Neurosynaptic chips address the senses and pattern recognition (**Right brain**)

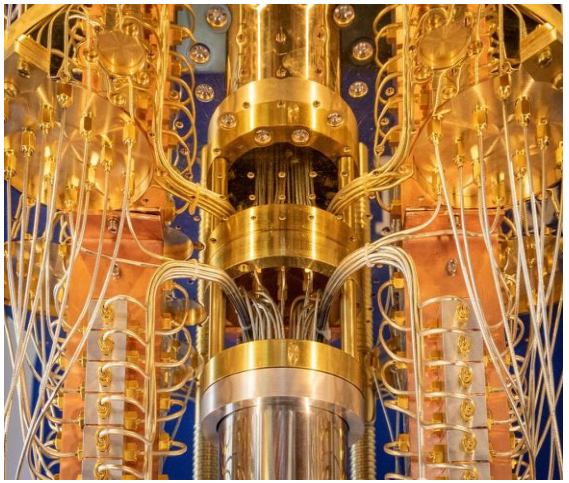
Over the coming years, **IBM** scientists hope to meld the two capabilities together to create **holistic computing intelligence**.

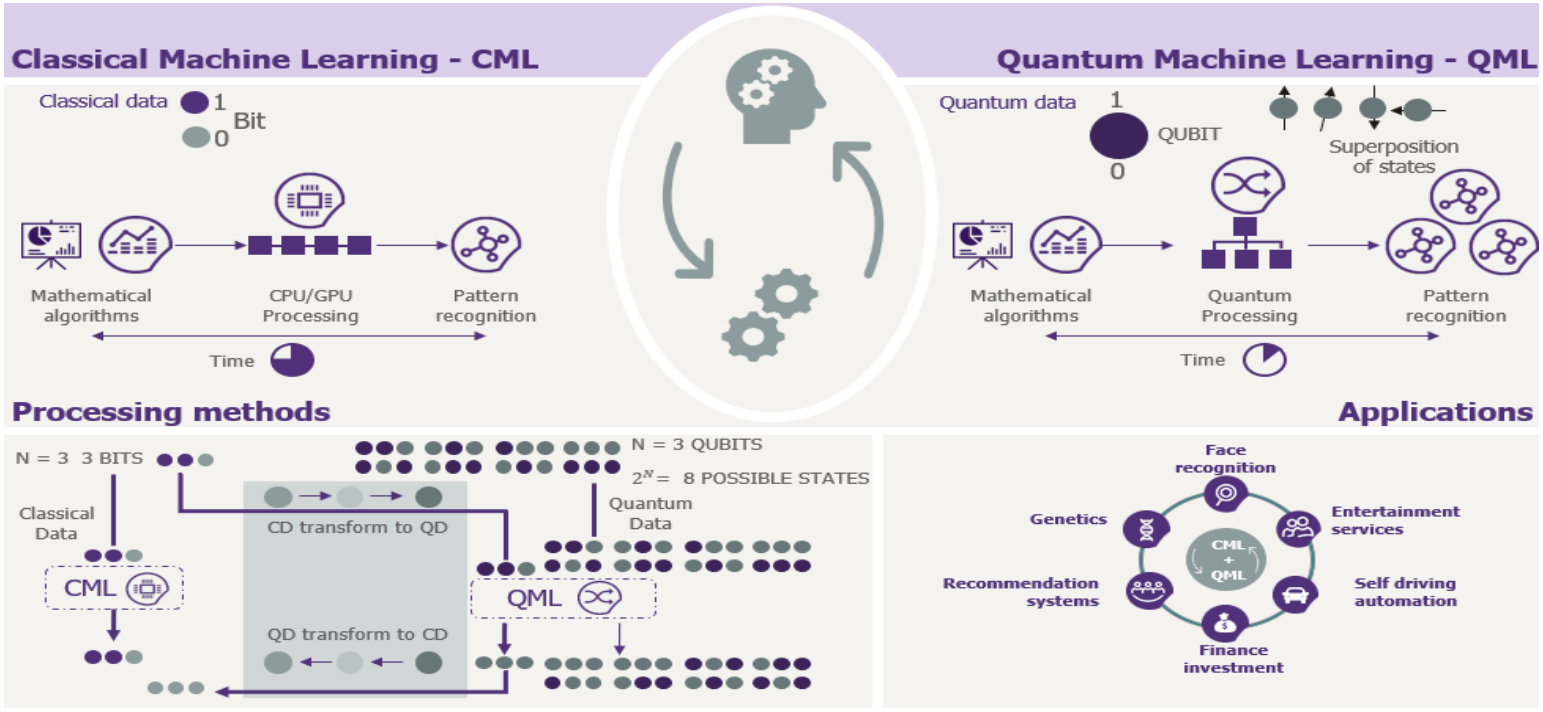
...

DL Popularity

Quantum Computing

- A new field is introduced as “quantum-enhanced machine learning”
- Employ quantum computers to learn pattern in data that cannot be learned by classical algorithm.

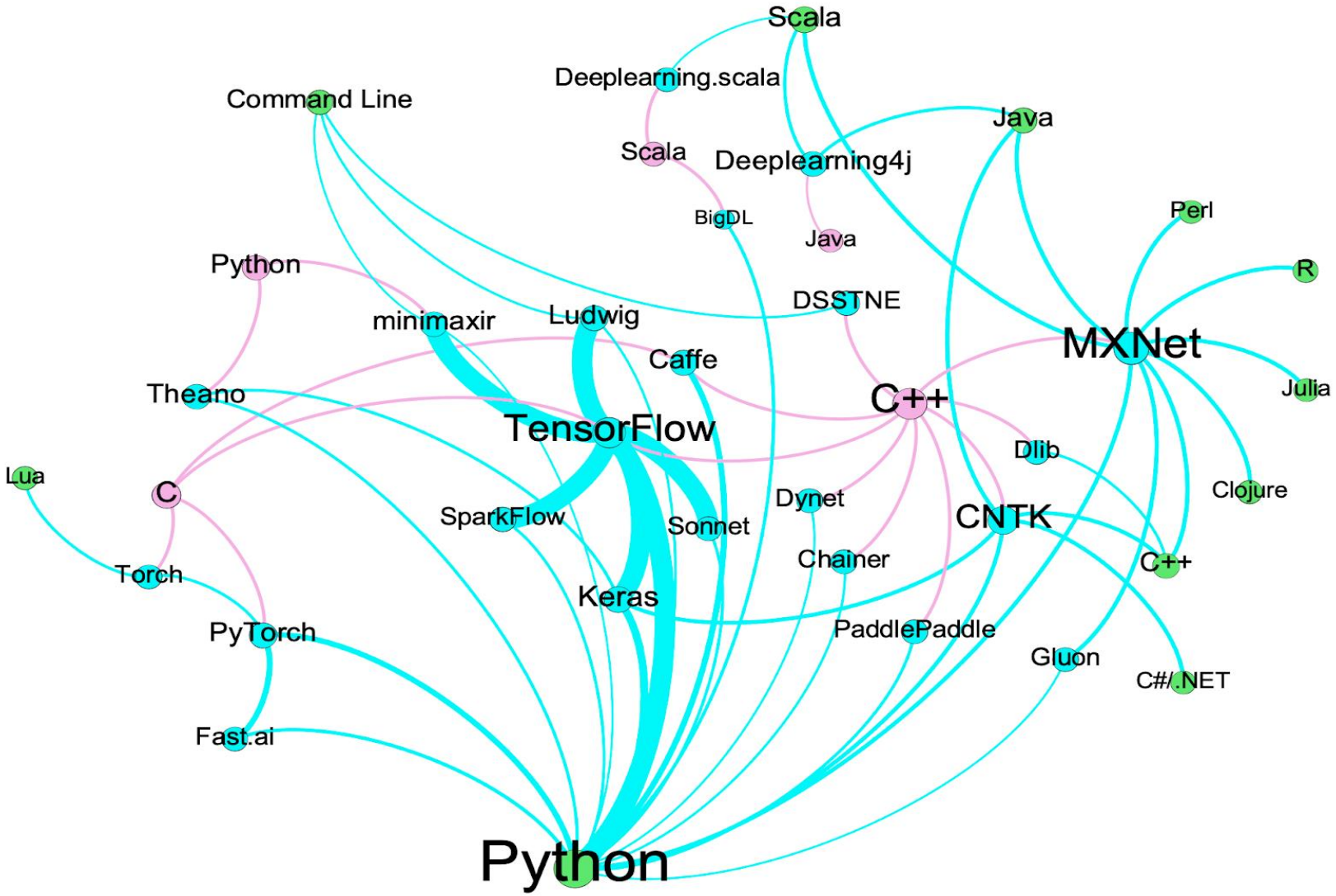




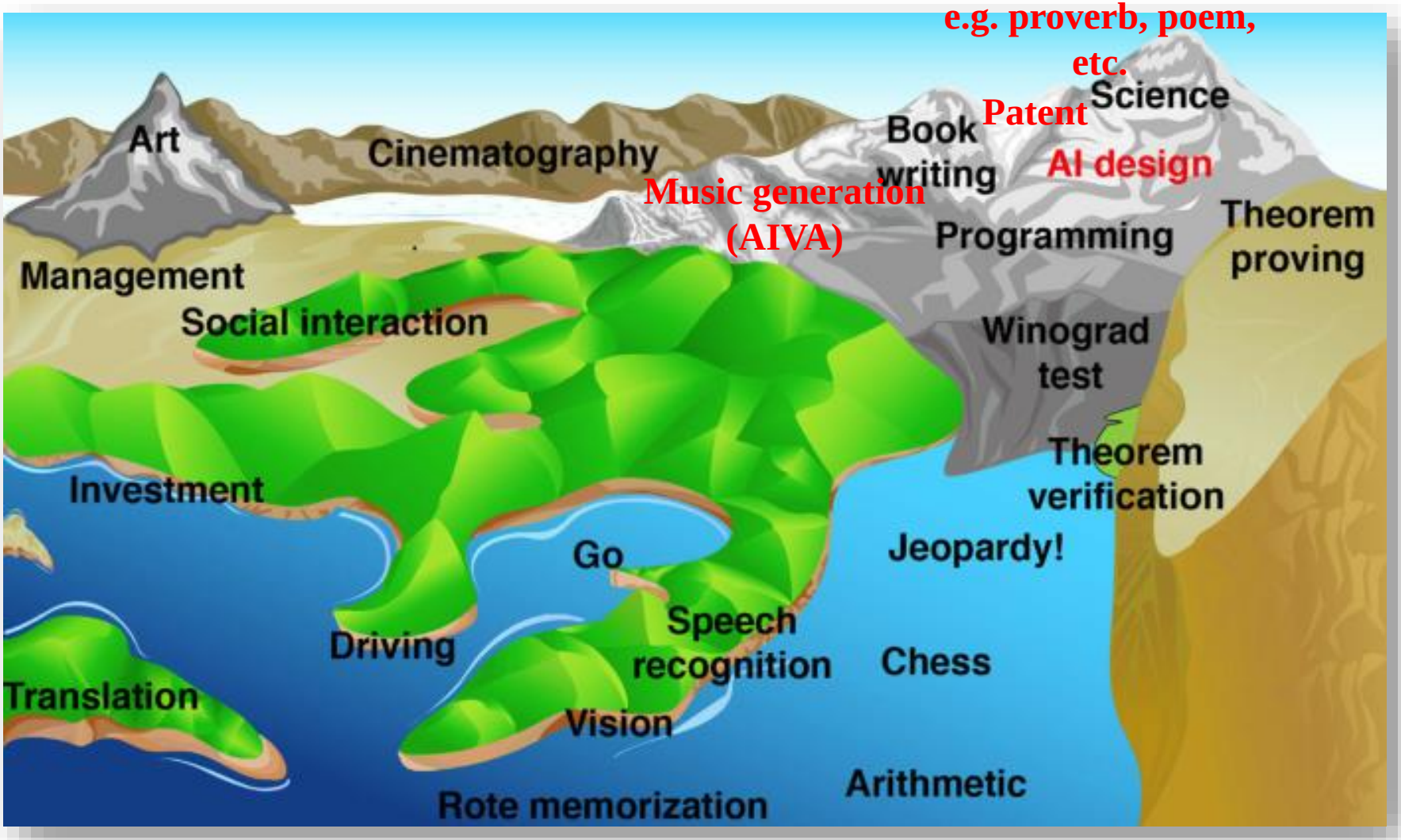
...

DL Popularity

Great Frameworks



DL Advances



Max Tegmark's sea visualization

…

DL Advances

- **Technical papers**:
- Predict future scientific discoveries by simply extracting meaningful data from research publications.

  https://perssongroup.lbl.gov/papers/dagdelen-2019-word-embeddings.pdf

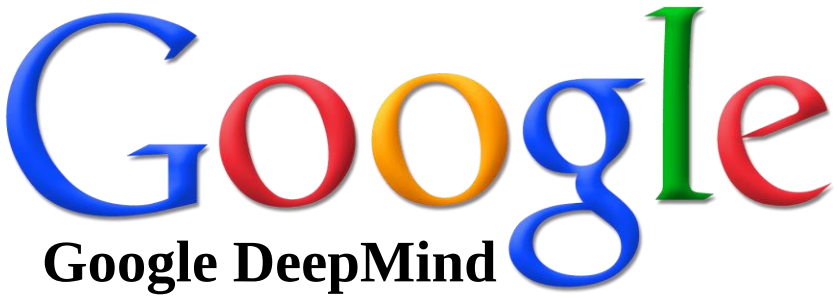
- **Patent**

DABUS, Surrey University

https://www.surrey.ac.uk/news/world-first-patent-applications-filed-inventions-generated-solely-artificial-intelligence

DL Pioneers

Companies



Google DeepMind

Microsoft

NVIDIA®

Alphabet

Baidu 百度

YAHOO!

...

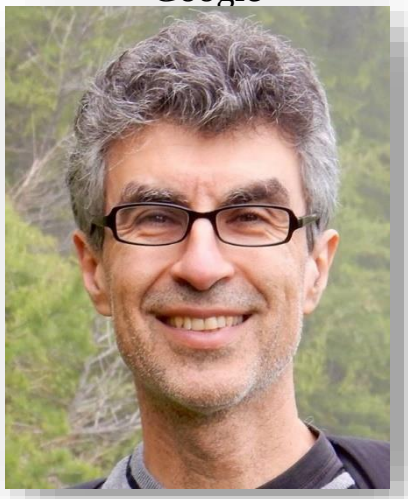
DL Pioneers

Individual

Geoffery Hinton
University of Toronto &
Google

Yann LeCun
New York University &
Facebook

Andrew NG
Stanford University &
Baidu



Yoshua Bengio
University of Montreal

Jurgen Schmidhuber
Swiss AI Lab &
NNAISENSE

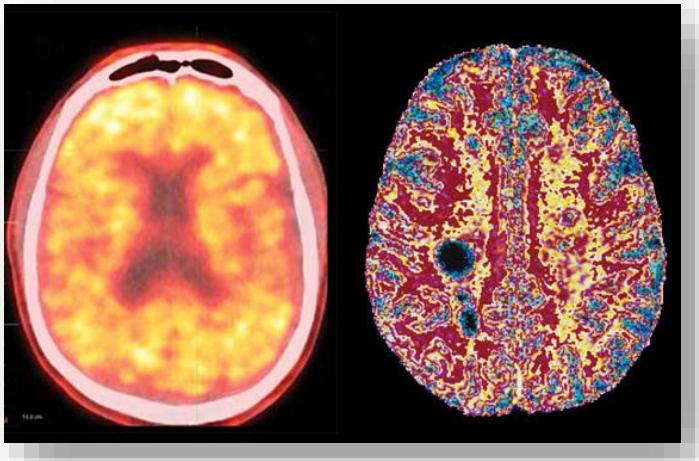
...

DL State-of-the-Art

- BERT and Natural Language Processing
- Tesla Autopilot Hardware v2+: natural networks at scale
- AdaNet: AutoML with ensembles
- AutoAugment: Deep RL data augmentation
- Training deep networks with synthetic data
- Segmentation annotation with Polygon-RNN++
- DAWNBench: training fast and cheap
- BigGAN: state-of-the-art in image synthesis
- Video-to-video synthesis
- Semantic segmentation
- AlphaZero & OpenAI five
- DL frameworks

DL Applications

Customer Support



MRI Image Analysis

Self Driving Cars

TESLA

NISSAN

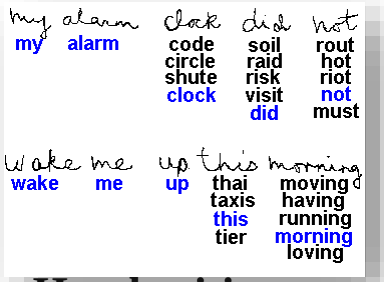
HYUNDAI

… …

DL Applications

Speech Recognition

Face Recognition



Handwriting Transcription

Text-to-speech Generation

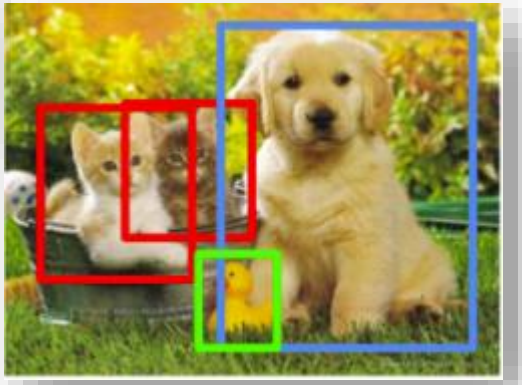


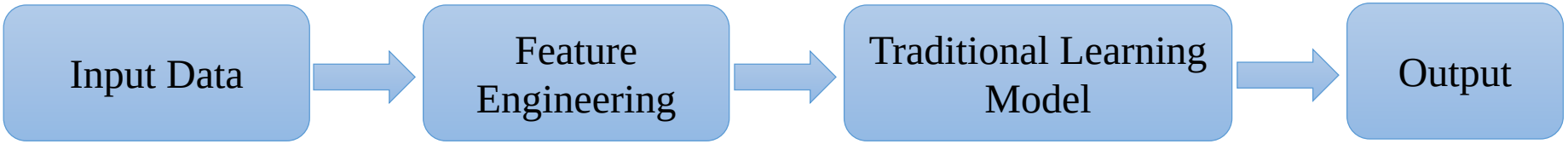
Image Classification

Machine Translations

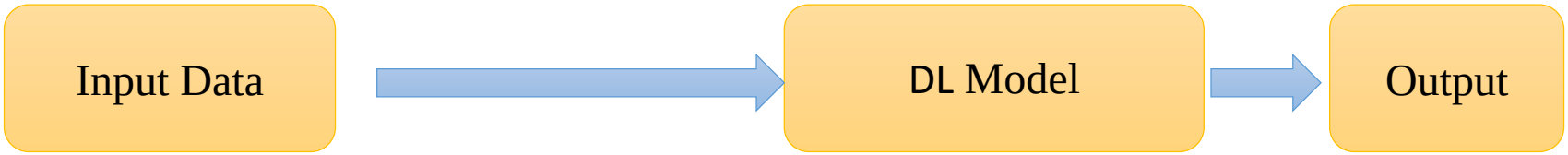
DL Basics

"Modern Reincarnation of Artificial Neural Networks"

- A deep neural network consists of a hierarchy of layers.
- Each layer transforms input data into more abstract representations.
- e.g. edge > nose > face.
- The output layer combines those features to make predictions.

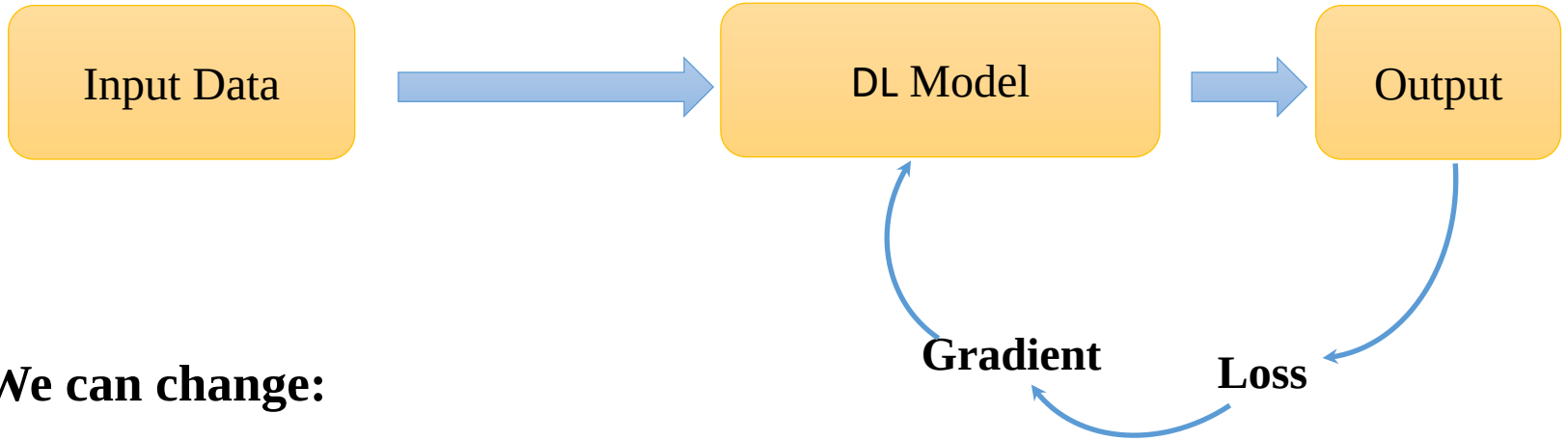


Costs lots of time



…

DL Basics



We can change:

- **Input**
- **Output** (regression, classification, multi-class, etc.)
- **Architecture** (ANN, RNN, CNN, etc.)
  - **Activation function**
  - **Optimizer** (Adam, stochastic gradient descent, batch gradient descent)
  - **Hyperparameters** (learning rate, batch,
- **Loss function**

…

When to use DL!

- Simpler models (logistic regression) don't achieve the **accuracy** level for our use-case need.
- We have **complex pattern** matching in images, NLP, or audio to deal with.
- We have **low** **quality** data.
- We have **high dimensionality** data.
- We have the dimension of time in our vectors (**sequences**).

DL Architecture

Main Architectures:

- **Feedforward Neural Networks**
- **Unsupervised Pre-trained Networks (UPNs)**
  - Autoencoders
  - Deep Belief Networks (DBNs)
  - Generative Adversarial Networks (GANs)
- **Convolutional Neural Networks (CNNs)**
  - LeNet
  - AlexNet
  - ZFNet
  - GoogLeNet
  - VGGNet
  - ResNet
- **Recurrent Neural Networks**
  - Bidirectional Recurrent Neural Networks (BRNN)
  - LSTM
- **Recursive Neural Networks**

Office: #432, Dayang AI Center, **Mobile**: 010-8999-8586, **email**: *piran@sejong.ac.kr*