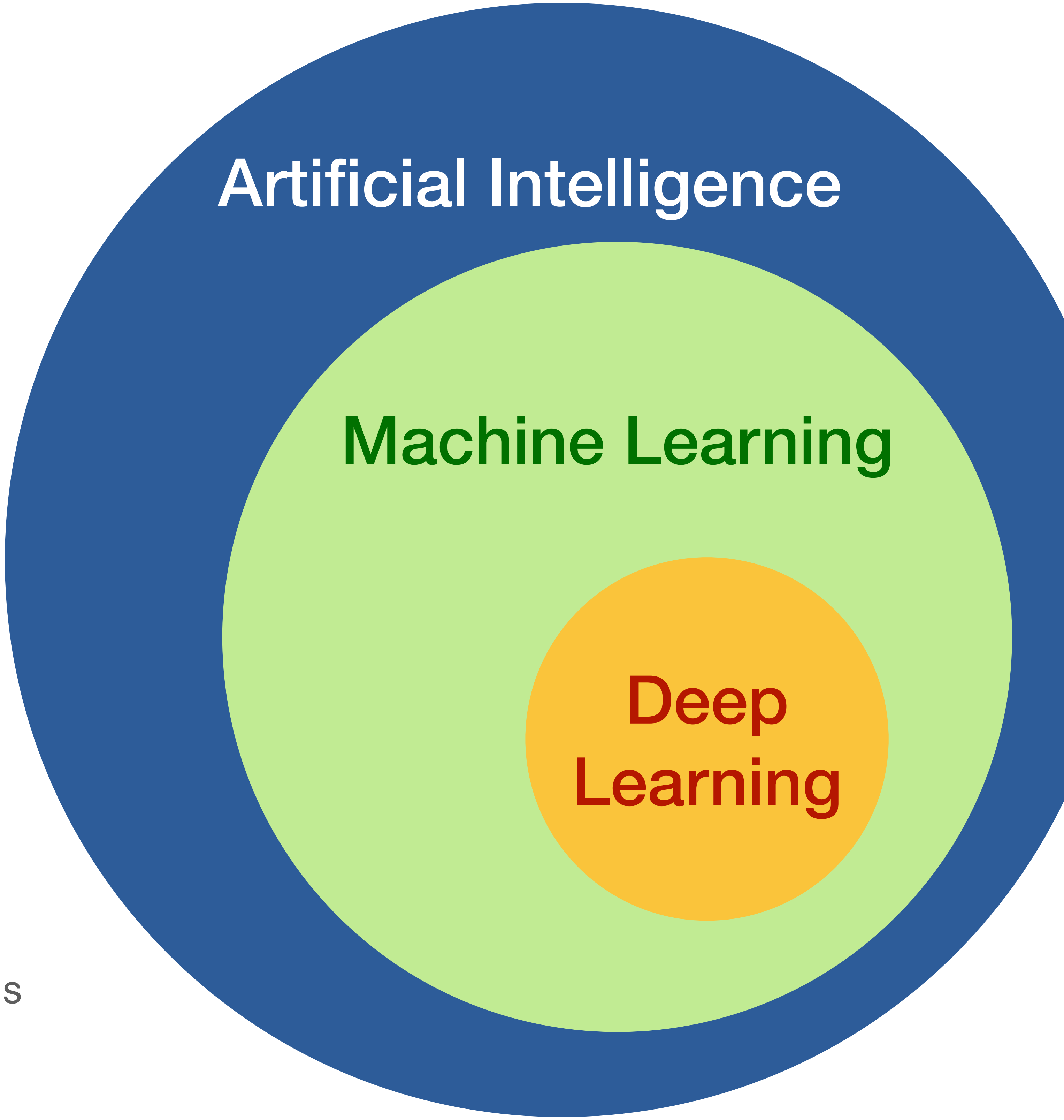
Overview of Machine Learning

AI, Machine Learning & Deep Learning

- **Artificial intelligence**
  Computer technologies to simulate **human intelligence**.

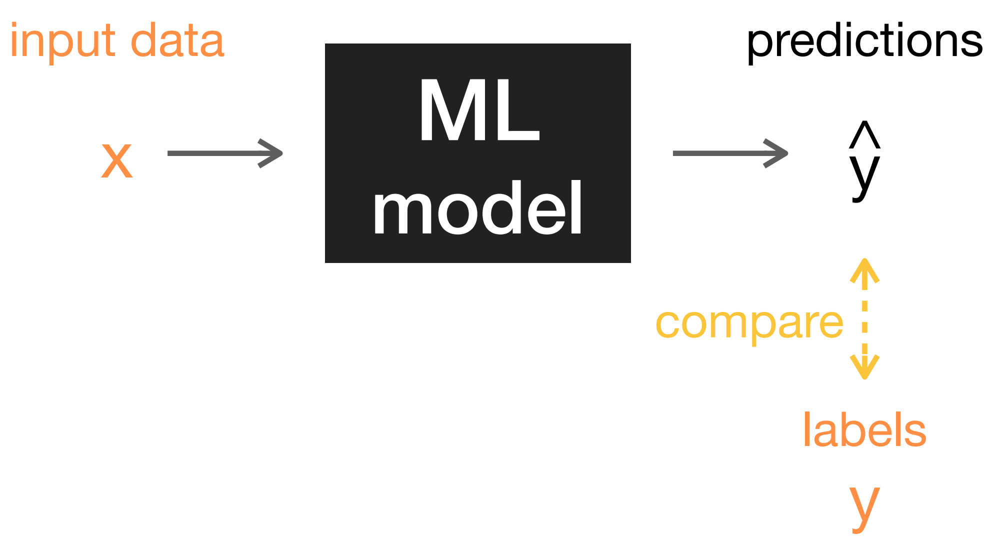
- **Machine Learning**
  [broader]
    Subset of AI techniques used to make predictions by **learning from data** and **improving from experience**.
    Learning process -> **Finding functions.**
  [narrower]
    Classical **statistical learning algorithms** (regression, clustering, SVM, random forests …)

- **Deep Learning**
  Subset of ML.
  Neural network based learning algorithms



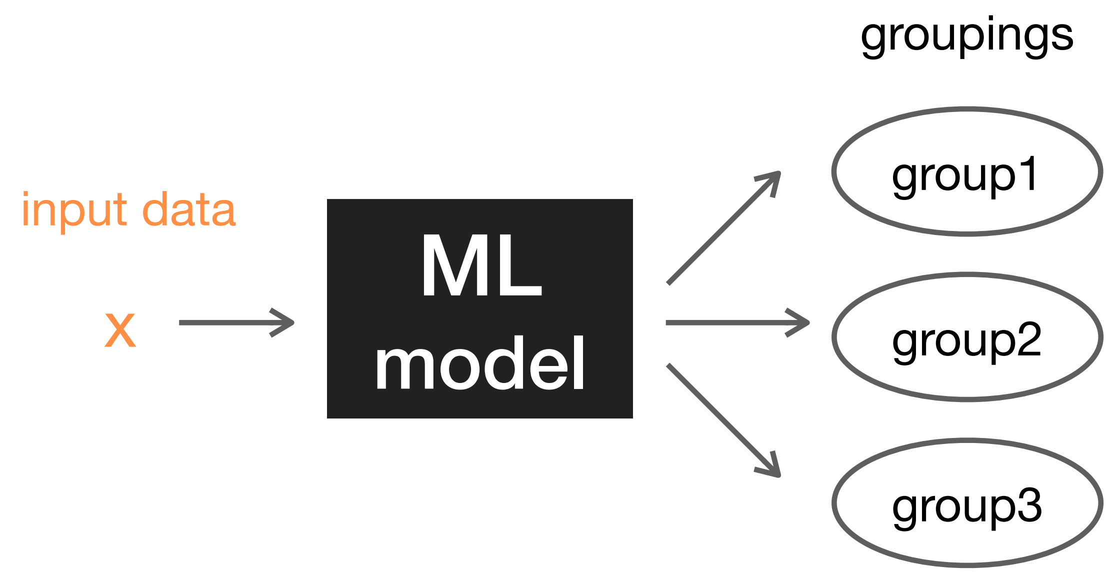
Types of Machine Learning

Supervised Learning



- Learning a function that maps an input to an output based on **labeled data**
- Goal : make predictions with high accuracy
- Applications : regression, classification
- Example algorithms : Linear / Logistic regression
  Image classification networks
- Challenges : labeled data is expensive

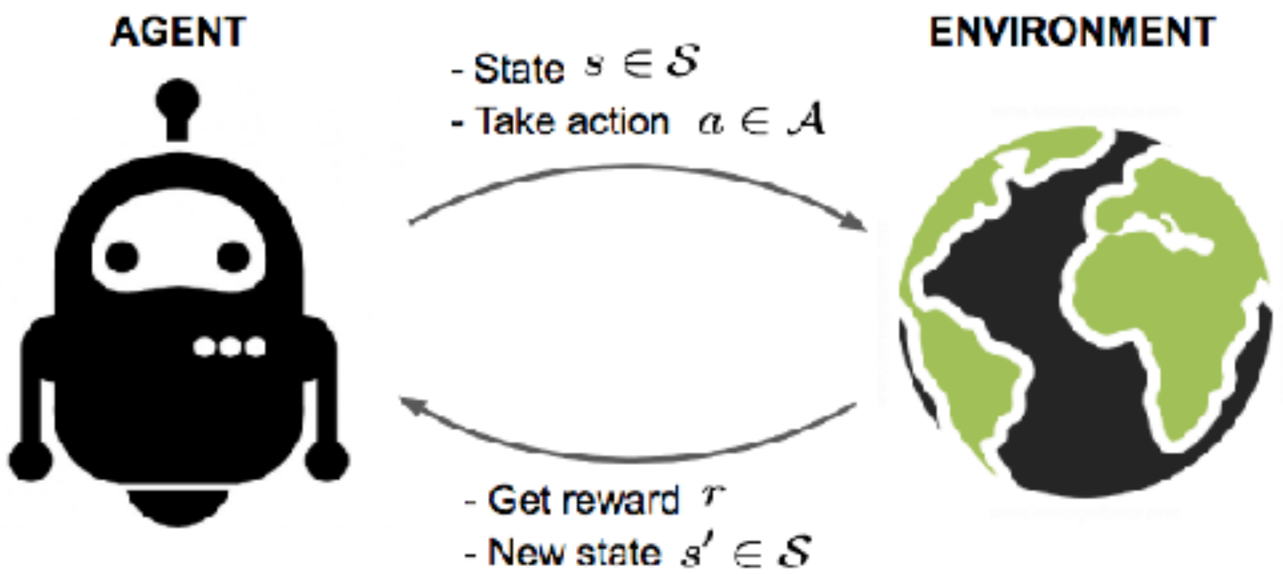
Unsupervised Learning



- Learning patterns / structures from **unlabeled data**
- Goal : identify interesting patterns in the data
- Applications : clustering, association
- Example algorithms : PCA
  AutoEncoder
- Challenges : evaluating whether the algorithm is learning something useful

Types of Machine Learning

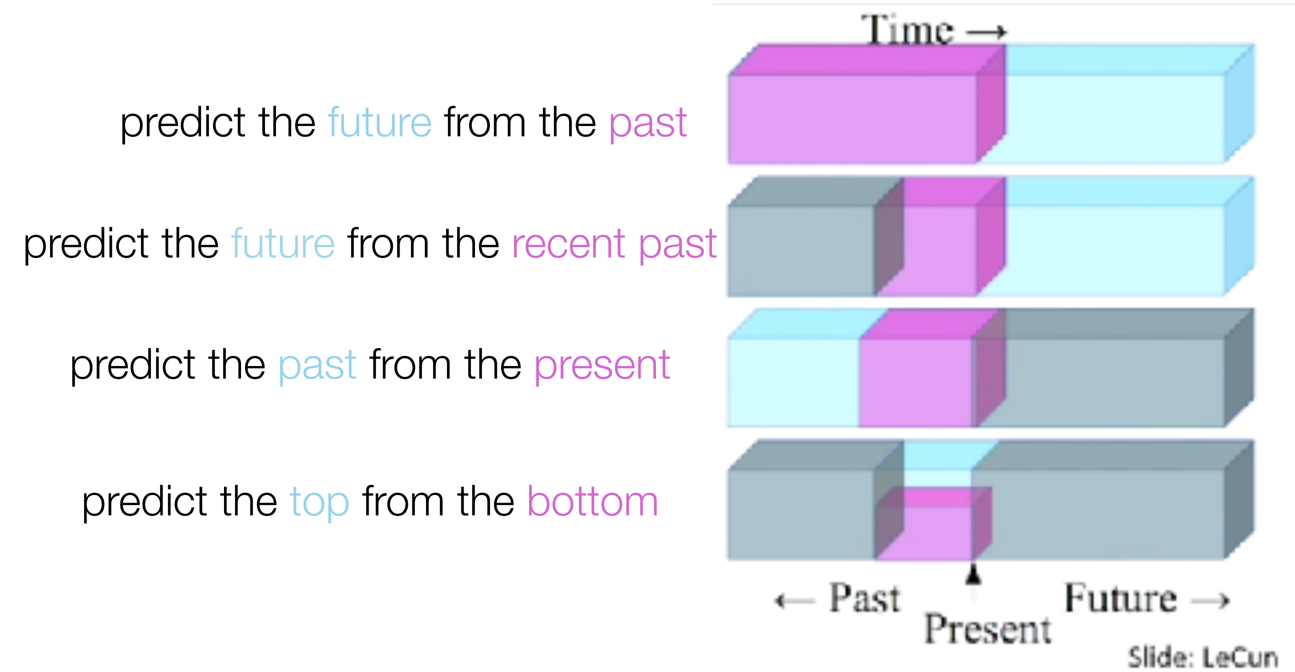
Reinforcement Learning



- Learning by trail and error
- Goal : maximizing the reward
- Example algorithms : AlphaZero
  (trained entirely from self-play)

Self-Supervised Learning

Machine learns to predict part of the input from any other part.



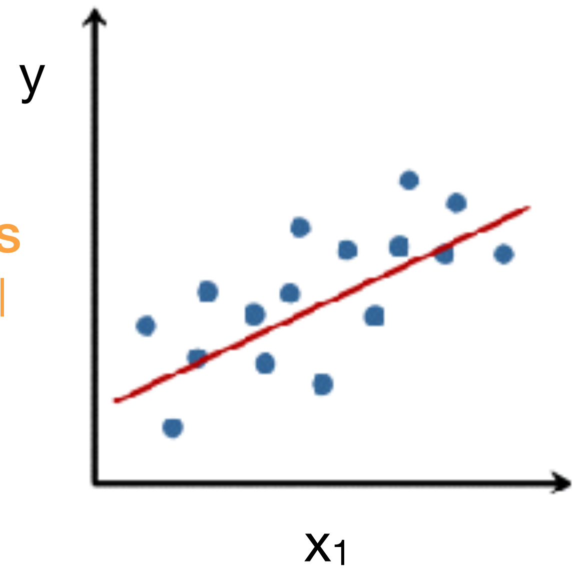
- Learning patterns by better utilizing **unlabeled data**.
- Goal : Learn **intermediate representations with good structural meanings** and can be beneficial to a variety of practical downstream tasks.
- “Most of what we learn as humans is in a self-supervised mode, not a reinforcement mode. It’s basically **observing the world** and **interacting with it a little bit**, mostly by observation in a test-independent way.” — Yann LeCun

Image credit : Lil’Log

Classical Machine Learning Methods

Linear Regression

Continuous
target label



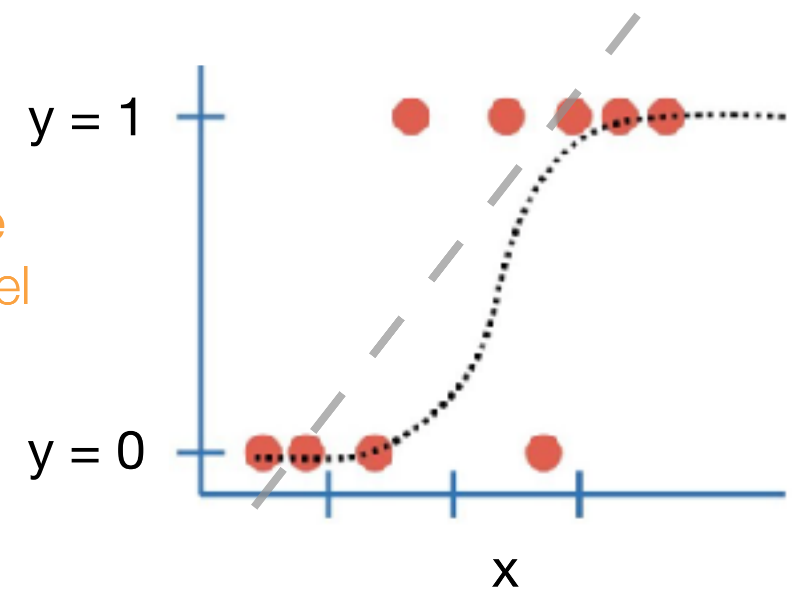
model prediction

$$\hat{y} \approx w_0 + w_1\, x_1$$

approximately modeled as

Logistic Regression

Discrete
target label



logistic function

$$\hat{y} = p(\,y=1 \mid x\,) \approx \frac{1}{1 + e^{-(w_0+w_1 x)}}$$

↖ logit / log-odd

$$log(\frac{\text{prob. of classfying to 1}}{\text{prob. of classifying to 0}}) = \log(\frac{p(y=1|x)}{p(y=0|x)})$$
$$= \log(\frac{\hat{y}}{1-\hat{y}}) = w_0 + w_1 x$$

Classical Machine Learning Methods

Linear Regression

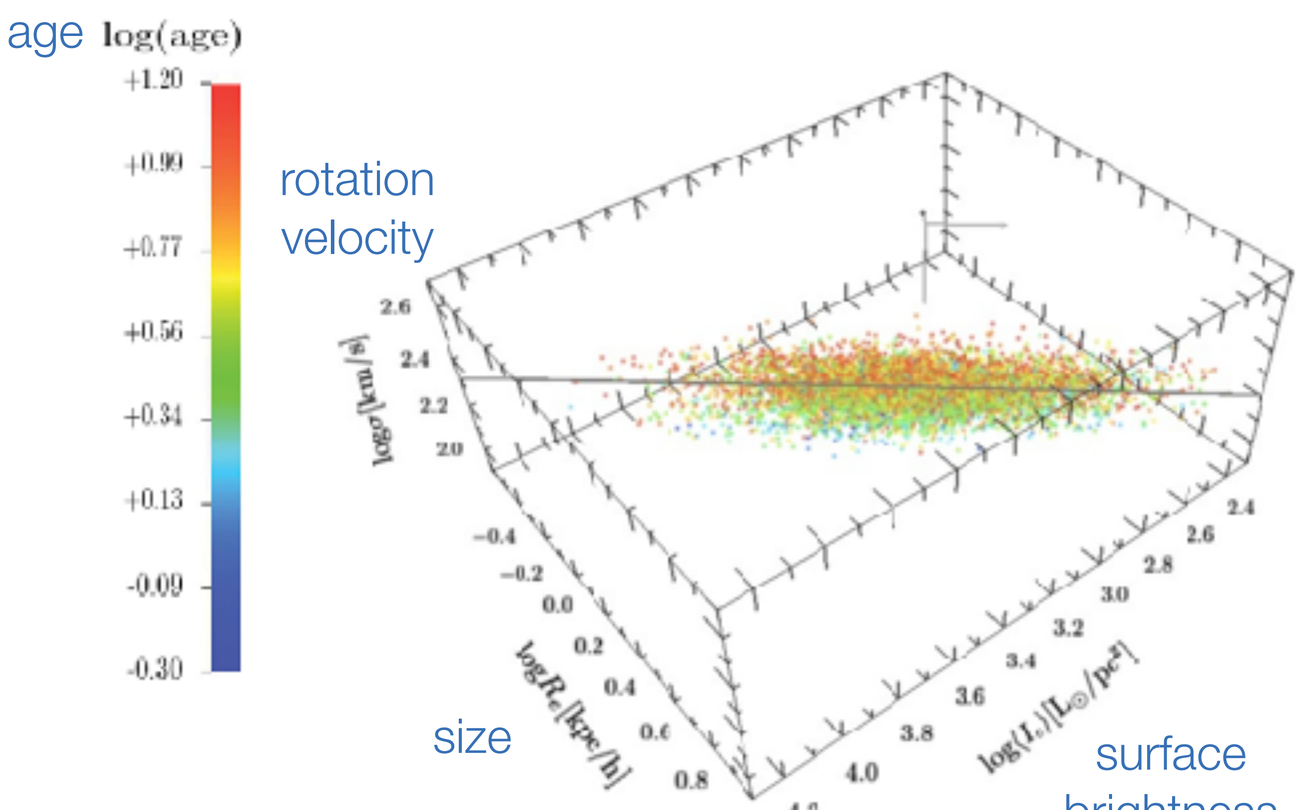
Logistic Regression

High Dimensional feature space

$$\hat{y} \approx w_0 + \Sigma_i w_i x_i$$

$$\hat{y} \approx \frac{1}{1 + e^{-(w_0 + \Sigma_i w_i x_i)}}$$

Fundamental Plane of galaxies



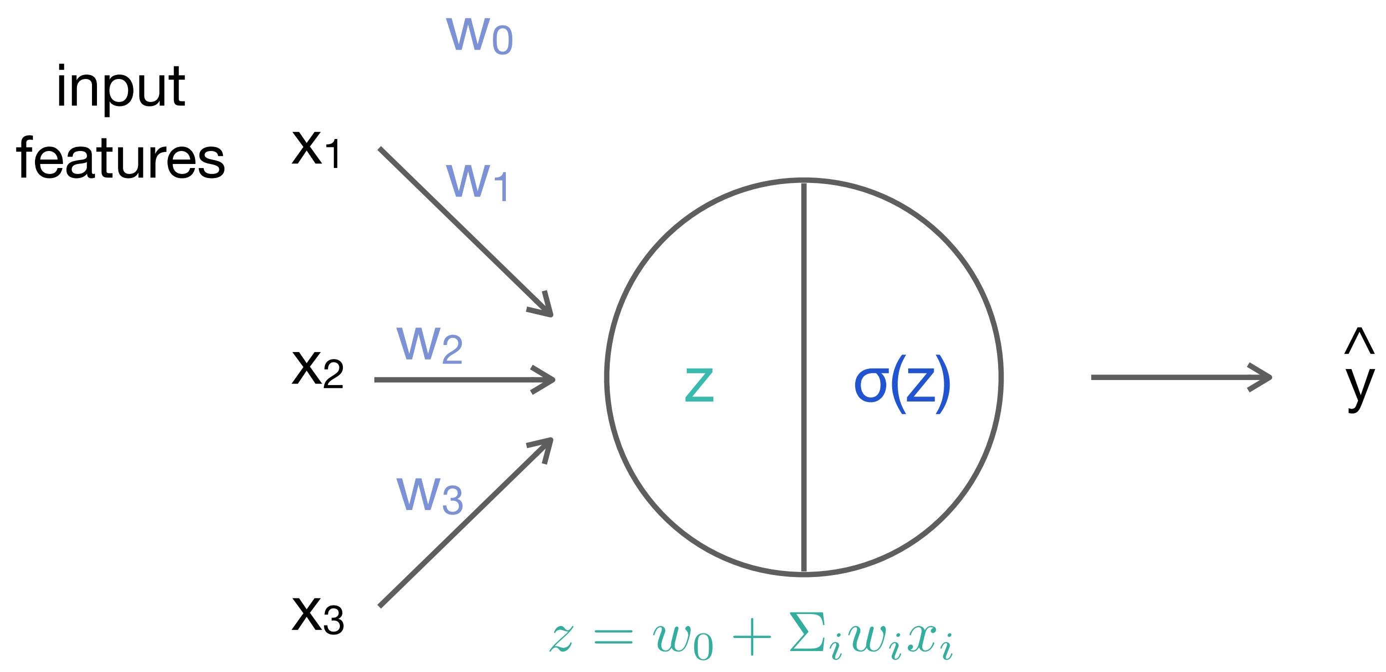
Magoulas+12

Active galaxy ?

y = 1 : active galaxy
y = 0 : quiescent galaxy

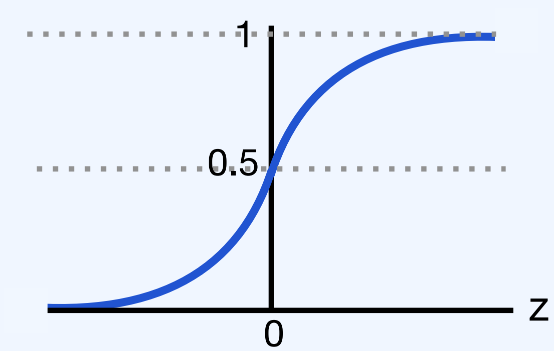
stellar mass
emission line intensity
black hole mass
color
…

Neural Network – Single Neuron

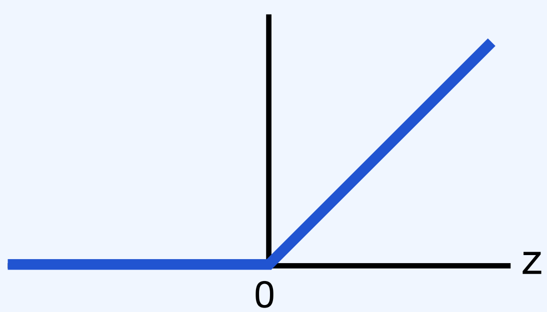


activation functions

sigmoid : $\sigma(z) = \dfrac{1}{1+e^{-z}}$

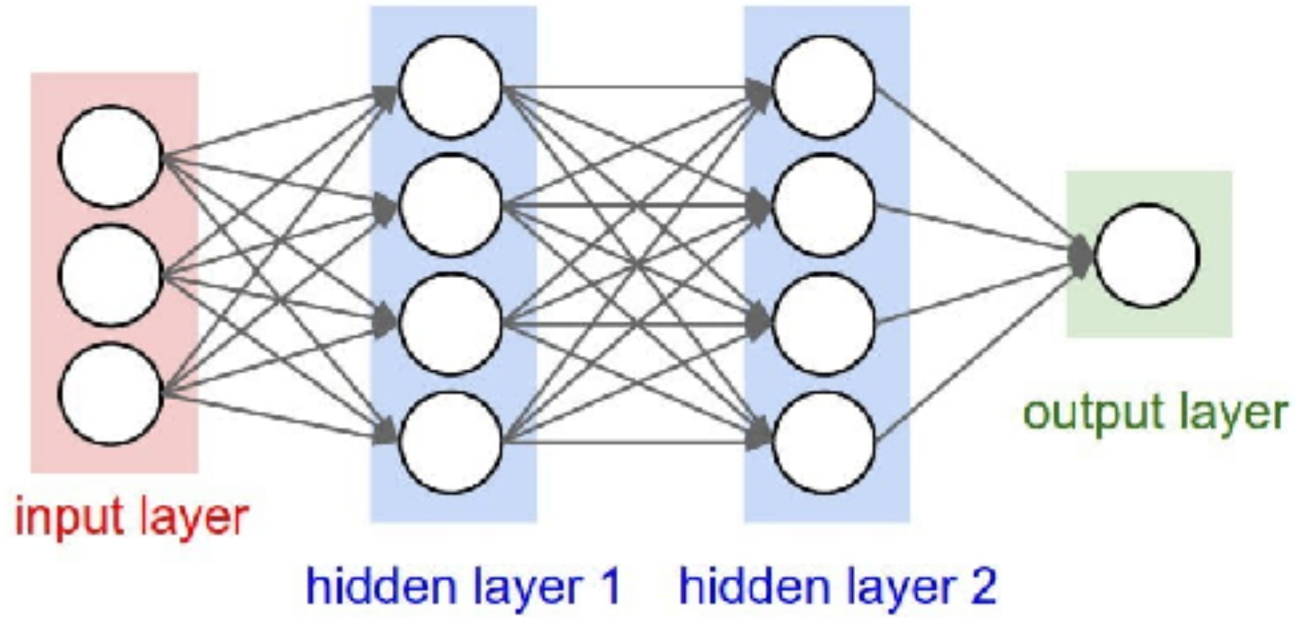


ReLU : σ(z) = max(0, z)



- Activation functions add non-linearity to neural network models.

Neural Network – Fully connected Multilayer Perceptrons (MLP)

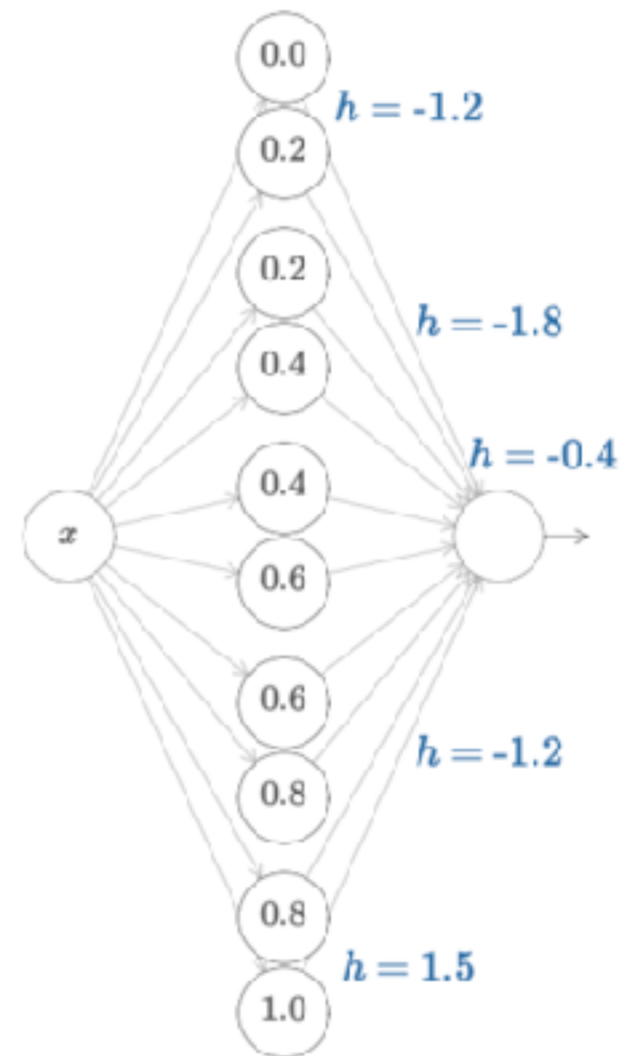


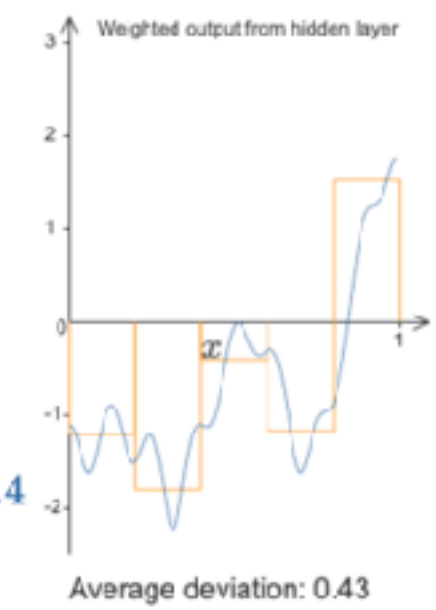
Universal Function Approximation Theorem

- By growing the network size, MLPs can approximate **any continuous functions** up to the **desired accuracy level**.

Width v.s Depth

- Deep networks can learn features in a **hierarchical** way
  Earlier layers : simple structures like edges
  Deeper layers : more complex representations
- Wide, shallow networks are more likely to have **overfitting** issue.

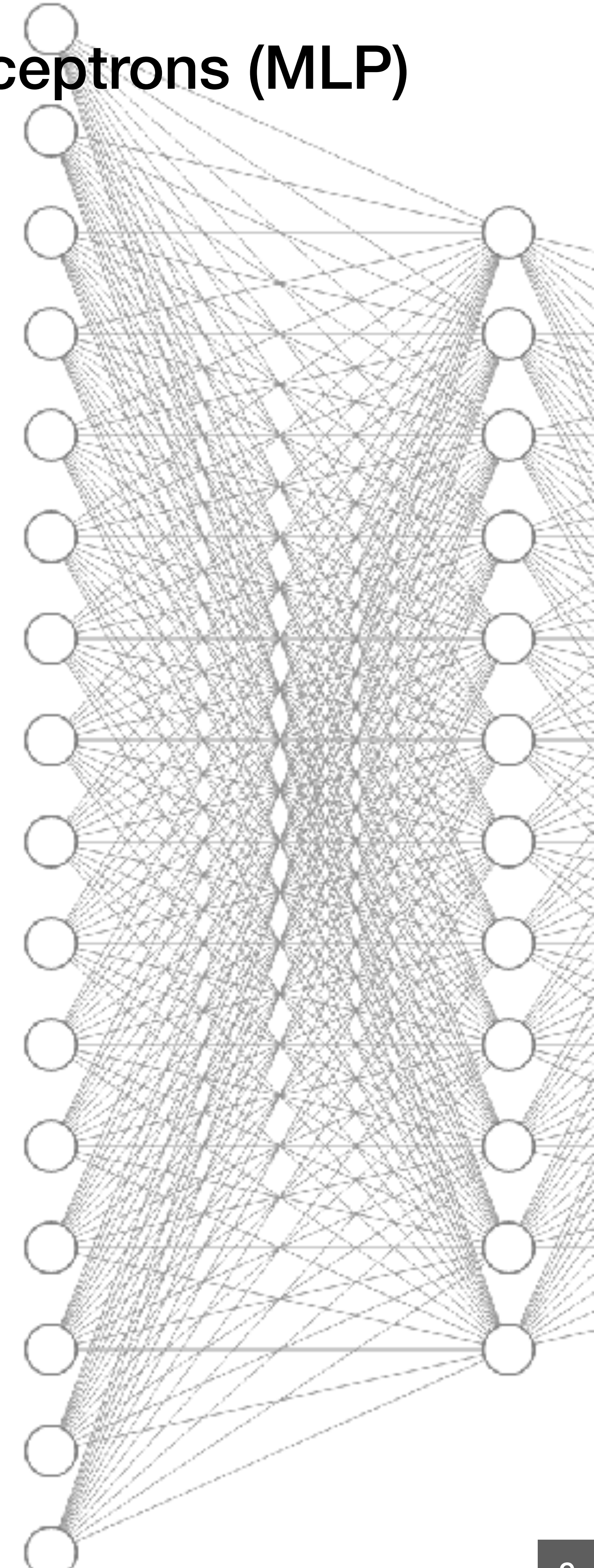




- Even single layer MLP can be so powerful. (visual proof)

Neural Network – Fully connected Multilayer Perceptrons (MLP)

128 x 128 pixels



Limitations of MLP

- **Too many parameters** due to fully connection.
- Low data efficiency : need lots of data to learn well.

Fully connected structure

↓

No prior assumption on how features interact from data.

Neural Network – Convolutional Neural Network (CNN)

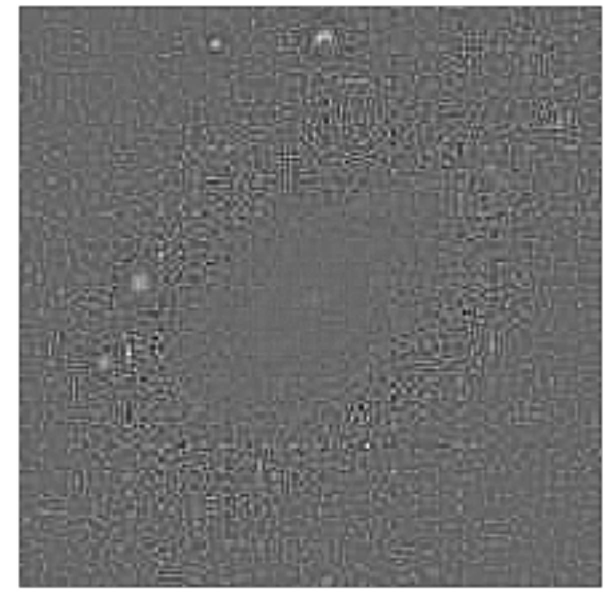
Image Convolution

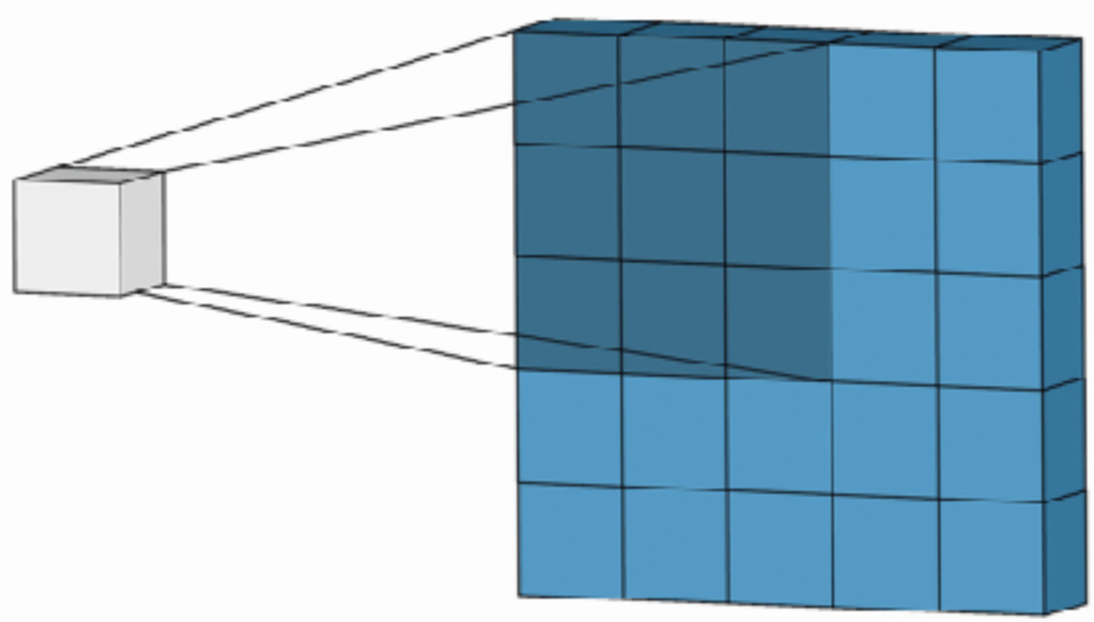
filter / kernel

$$* \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$

learnable parameters

feature map





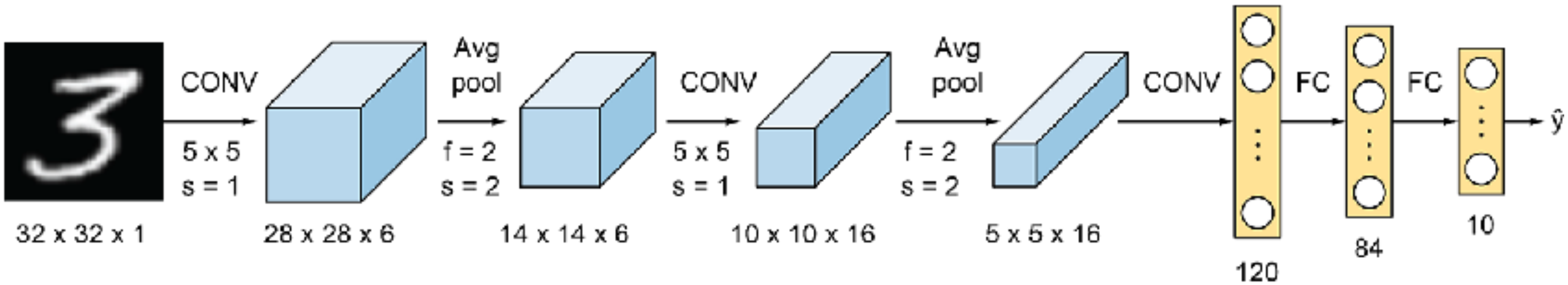
CNN Properties

- **Translational Symmetry**
  Weight sharing across the entire image
- **Local Connectivity**
  Drop connections between far away neurons

Image source

Convolutional Network structures

LeNet-5 ~ 60k parameters A **pioneering CNN network** by LeCun et al. 1998



ConV Pool ConV Pool ConV FC FC

Average Pooling Layer

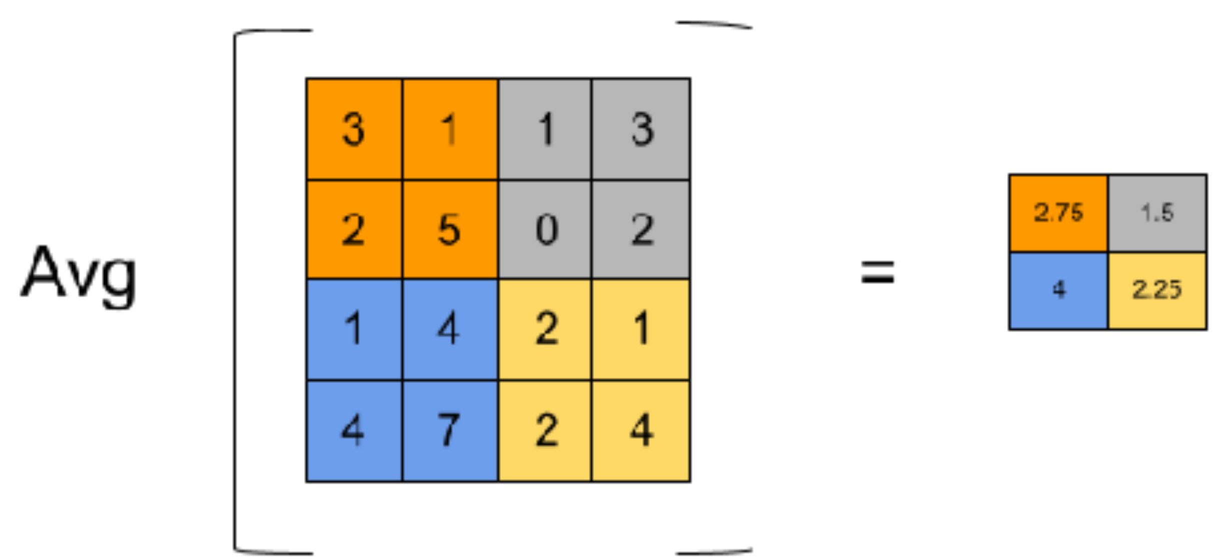
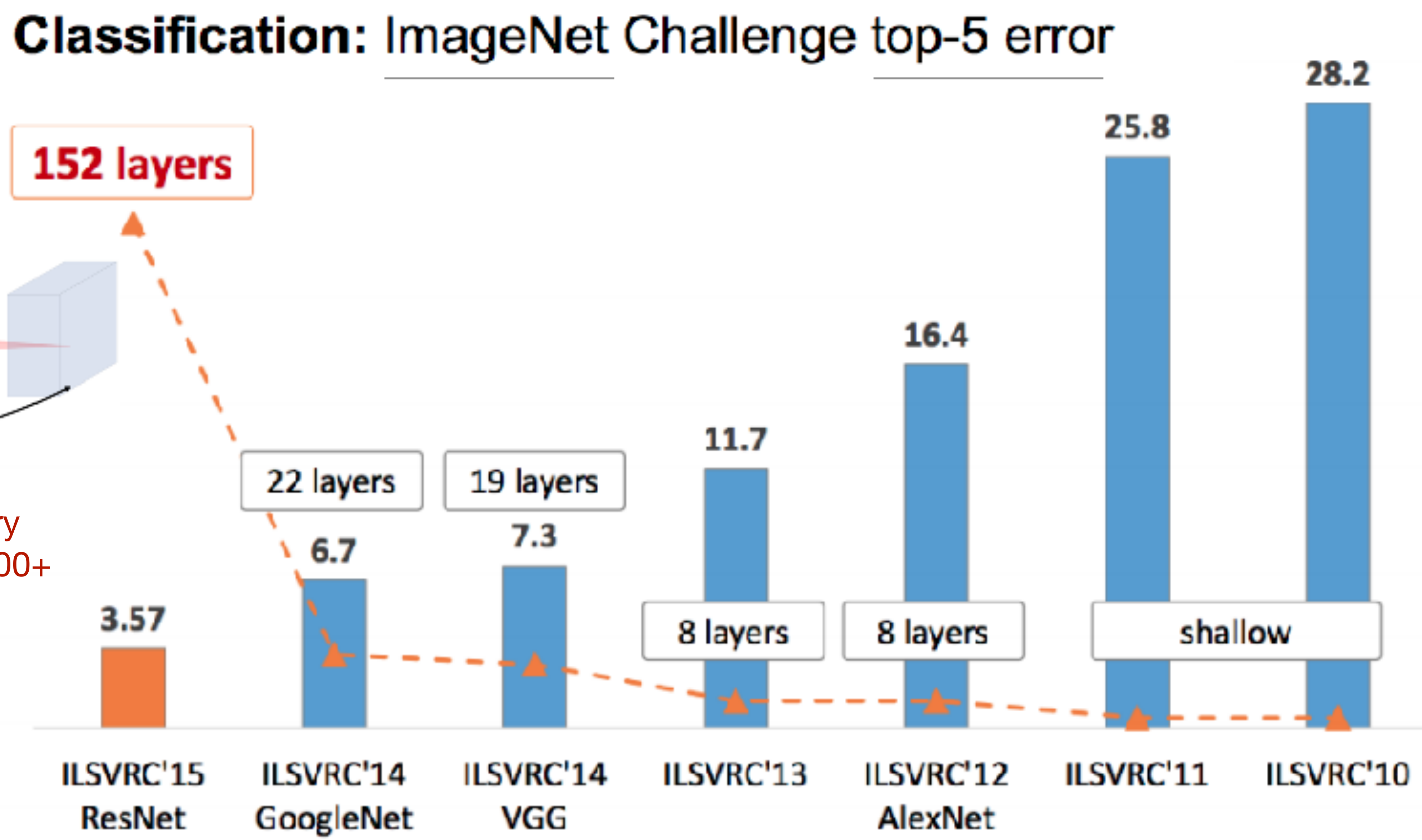


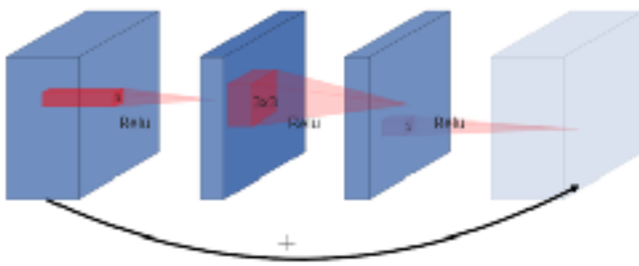
Image source

Evolution of CNNs

1000 categories

if the target label is one of the top 5 predictions
→ correct prediction



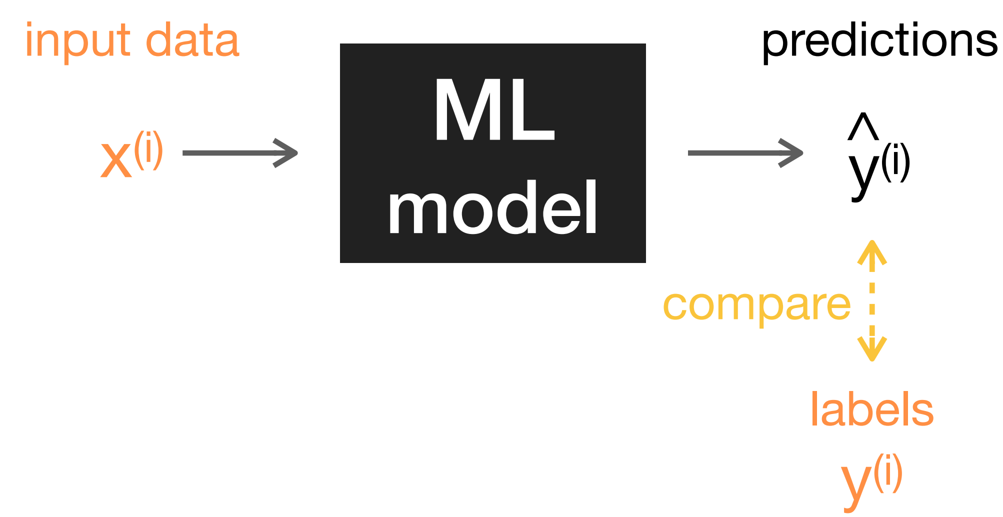


- skip connection → allowing training very deep networks of 100+ layers

	ResNet 2015	GoogLeNet 2014	VGG 2014	AlexNet 2012
trainable parameters	~ 60M	~ 7M	~ 144M	~ 60M

Image source

Training Process



0. Training Data

$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots (x^{(i)}, y^{(i)}) \dots , (x^{(m)}, y^{(m)})$

1. Define Model

- From simple → more complex network structures.
- For similar data types → Find existing network model and apply directly.

2. Define Loss

average across all training samples →

- Mean Squared Loss : $\frac{1}{2}(\hat{y} - y)^2$ → $J = \frac{1}{2m}\Sigma_{i=1}^{m}(\hat{y}^{(i)} - y^{(i)})^2$
- Cross Entropy Loss : $-(y \log \hat{y} + (1-y)\log(1-\hat{y}))$ → $J = \frac{-1}{m}\Sigma_{i=1}^{m}(y^{(i)} \log \hat{y}^{(i)} + (1-y^{(i)})\log(1-\hat{y}^{(i)}))$

3. Optimization

α : learning rate

- Gradient Descent $\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$
- Advanced optimization techniques / algorithms :
  Momentum, RMSProp, Adam, Learning rate decay

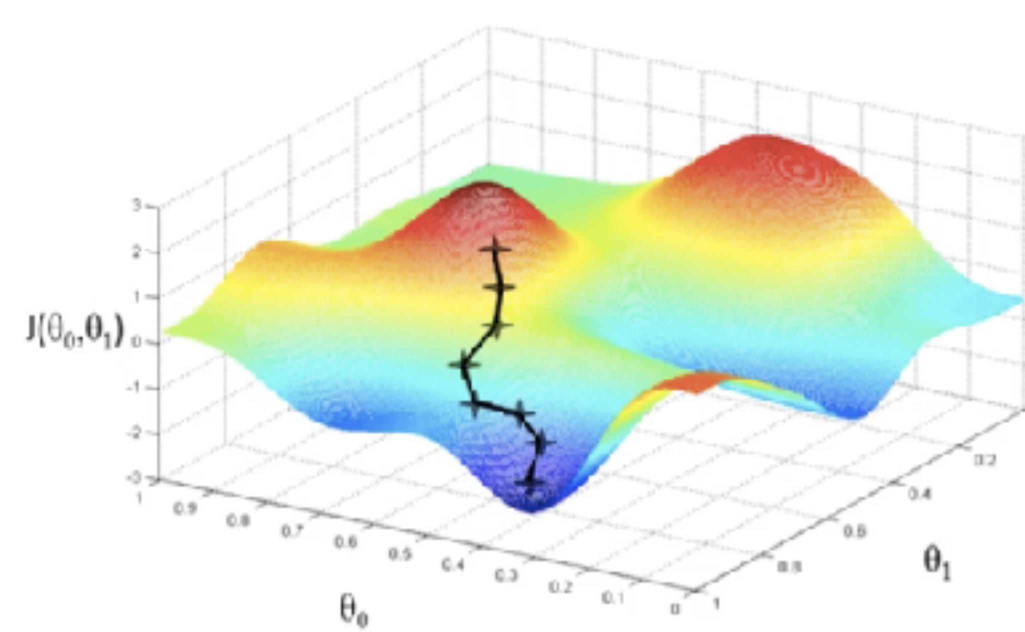
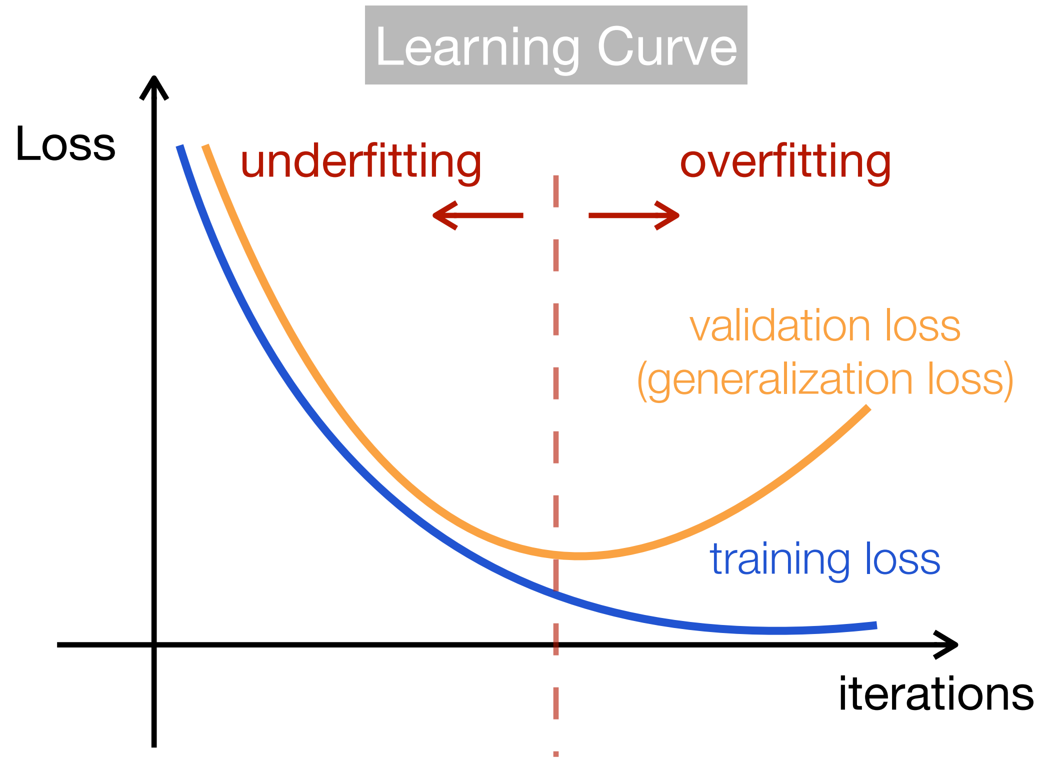
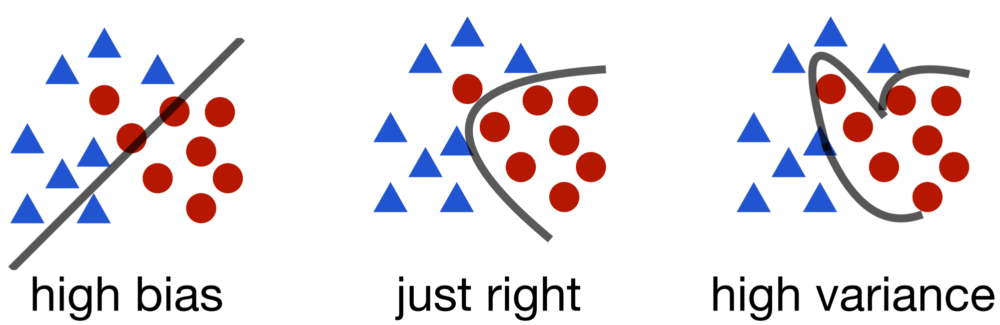


Image source

Overfitting & Regularization Techniques



- Bias & Variance



- **Regularization** : techniques that discouraging learning a more complex model to prevent overfitting.

Data augmentation

natural image : random shift, color change
galaxy image : random rotation, foreground contamination

Early stopping

Select the model that **performs the best on the validation set.**

L2 regularization

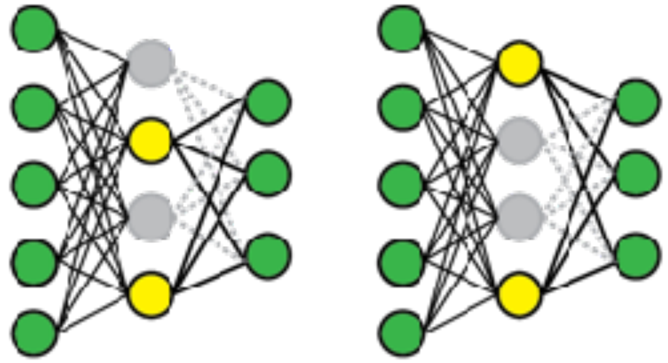
minimize (**Loss**(Data|Model) + **Complexity**(Model))

$$J(\boldsymbol{\theta}) = \frac{1}{m}\Sigma_{i=1}^{m}\mathcal{L}(\hat{y}^{(i)}, y) + \frac{\lambda}{2m}\|\boldsymbol{\theta}\|^2$$

λ : regularization parameter

Dropout

Randomly eliminate a fraction of nodes for each training example.



Summary

Classical Machine Learning

- Works better for **structured data** (catalog, tabular data).
- Limited performance on learning complex functions.
- Data Efficient. Easier to Train.
- Require some data manipulation/exploration before feeding to the algorithm (dimensionality reduction, feature extraction…).
- Easier to interpret.
- Usually with evaluable prediction uncertainty.
- Not covered in this course. But is really useful in physical science.
  - Statistics, Data Mining, and Machine Learning in Astronomy — Ivezic et al.
  - ASTR 502 2020 class notebook

Deep Learning

- Works especially well for **unstructured data** (e.g. image, audio signal).
- Superior performance on wide variety of tasks.
- Data efficiency is poor.
  Need lots of data to train.
- Directly pass the data into the network.
- Difficult to understand.
- Challenge to evaluate uncertainty.
- Focus of this course.
  - ML papers in cosmology

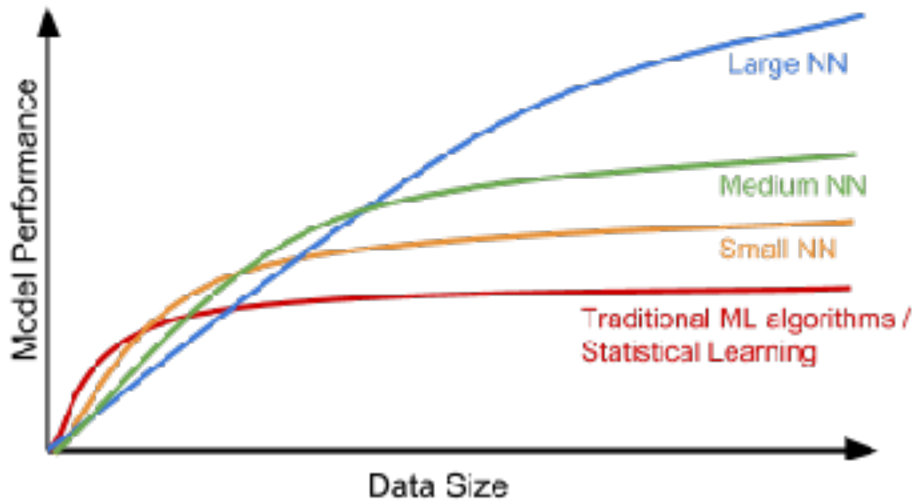


Image source