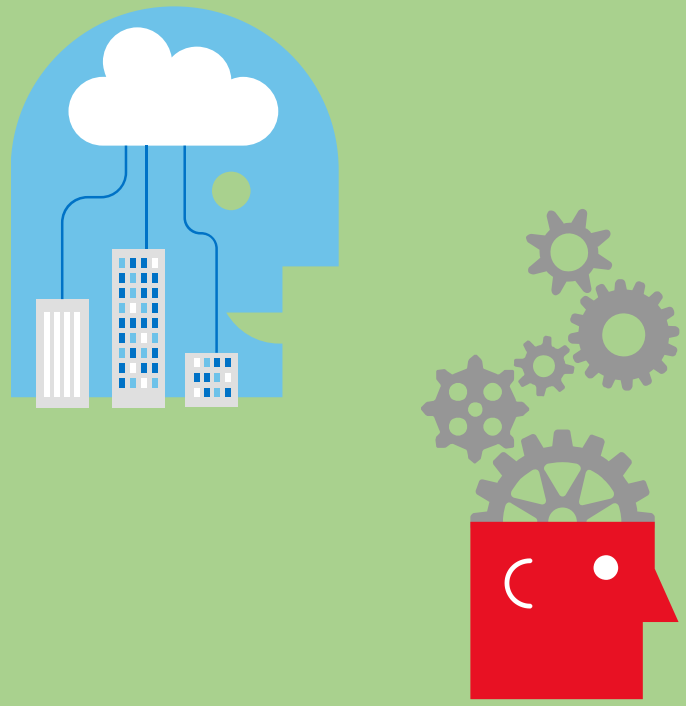
Artificial Intelligence and Machine Learning



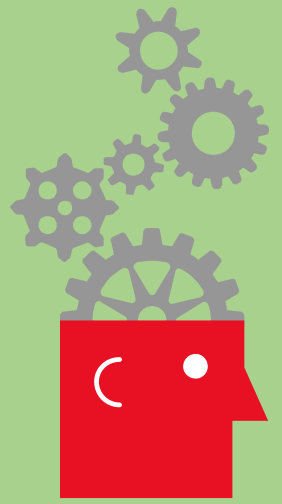
Microsoft

MSA
Microsoft Student Accelerator

1000kg

40mpg

3000kg



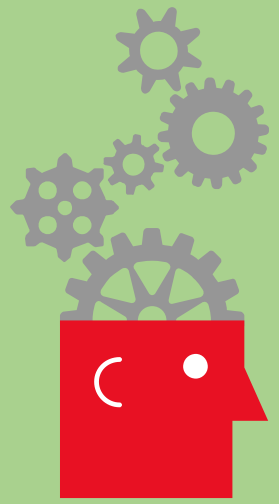
20mpg

2000kg

?

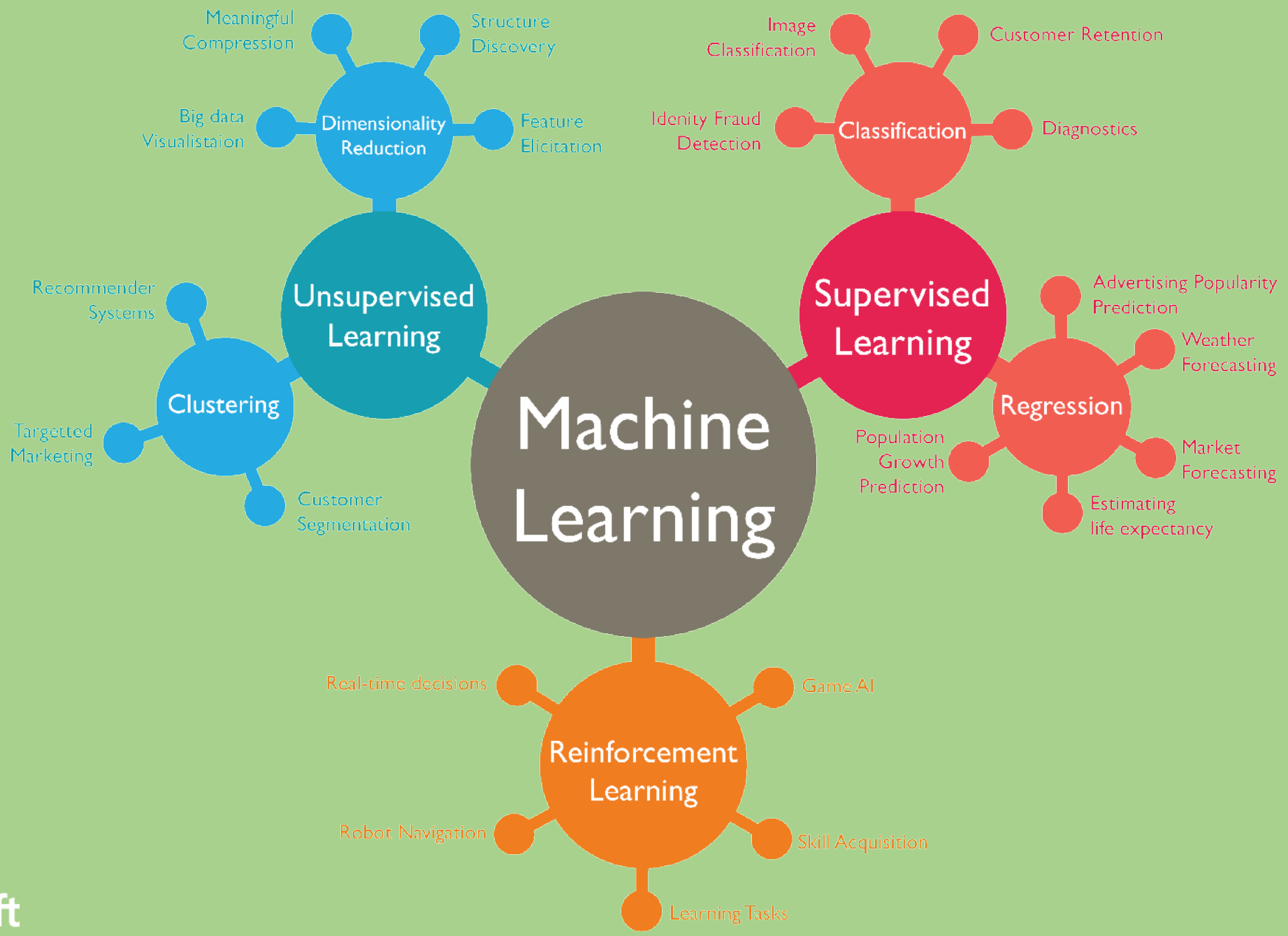
Microsoft

MSA
Microsoft Student Accelerator

	inputs		*output*
	1000kg		40mpg
	3000kg		20mpg
	2000kg		30mpg

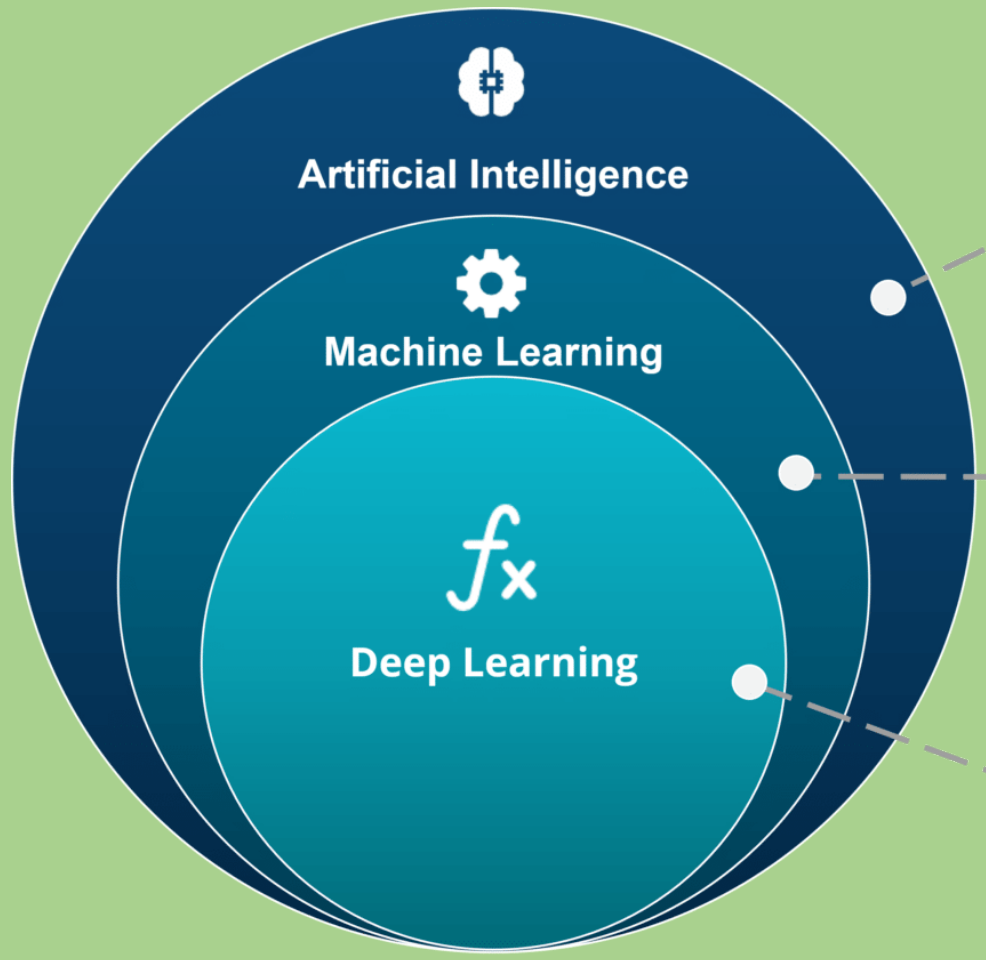
Microsoft

MSA
Microsoft Student Accelerator



Microsoft

MSA
Microsoft Student Accelerator



ARTIFICIAL INTELLIGENCE

A technique which enables machines to mimic human behaviour

MACHINE LEARNING

Subset of AI technique which use statistical methods to enable machines to improve with experience

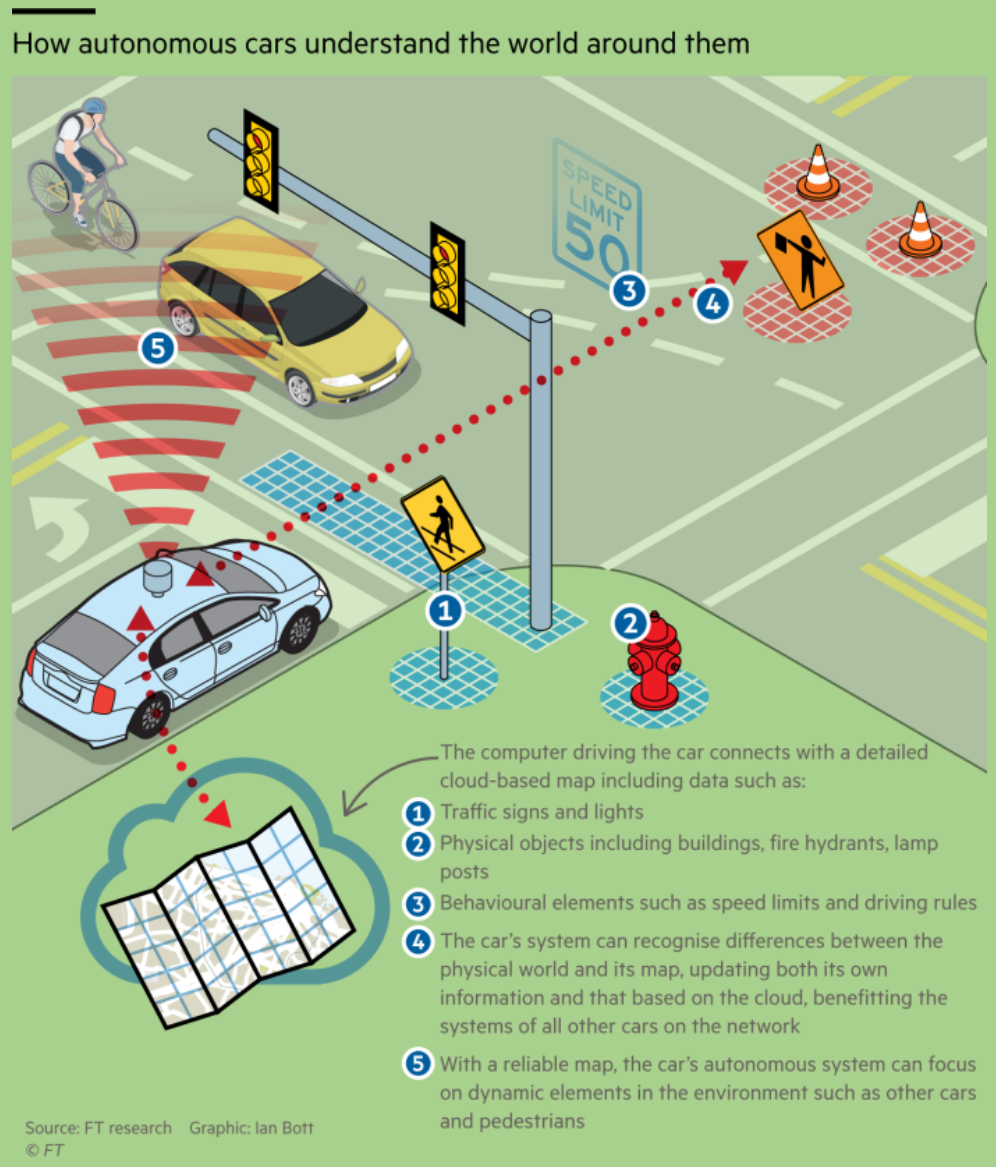
DEEP LEARNING

Subset of ML which make the computation of multi-layer neural network feasible

Microsoft

MSA
Microsoft Student Accelerator

AI Example



Microsoft

MSA
Microsoft Student Accelerator

Perception

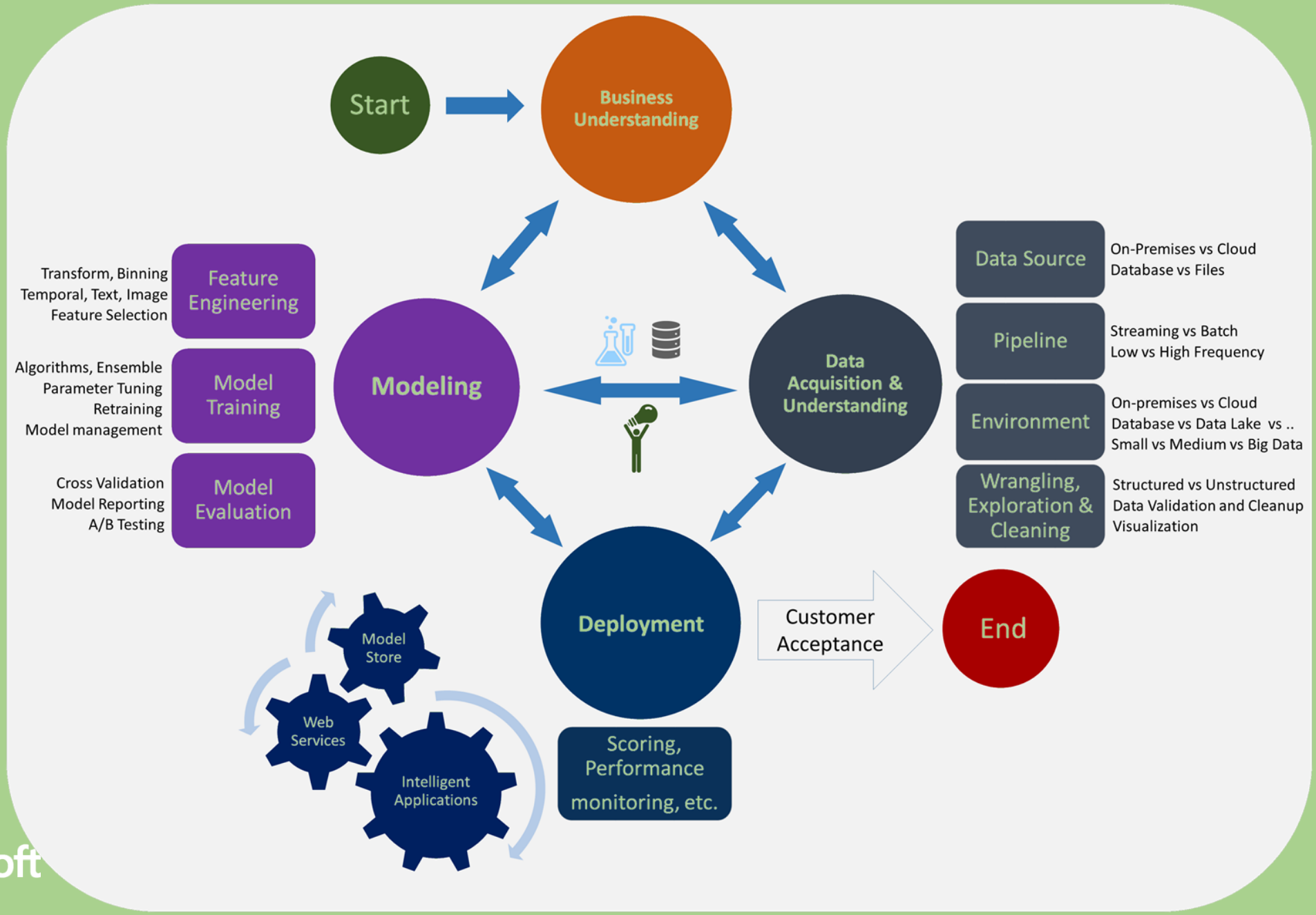
Prediction

Planning

Microsoft

MSA
Microsoft Student Accelerator

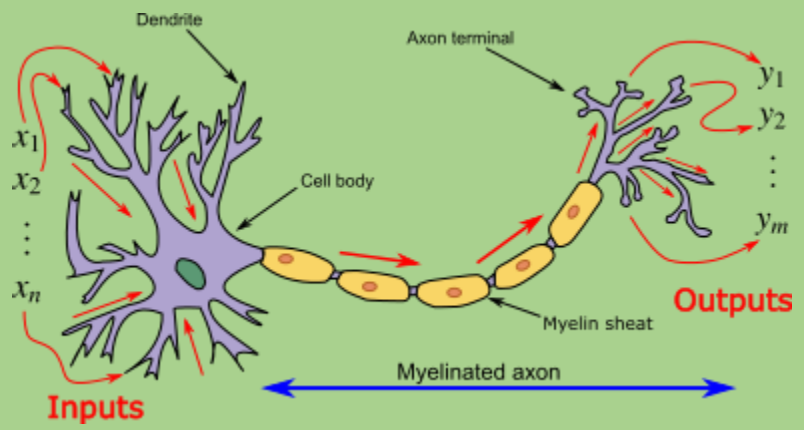
Data Science Lifecycle

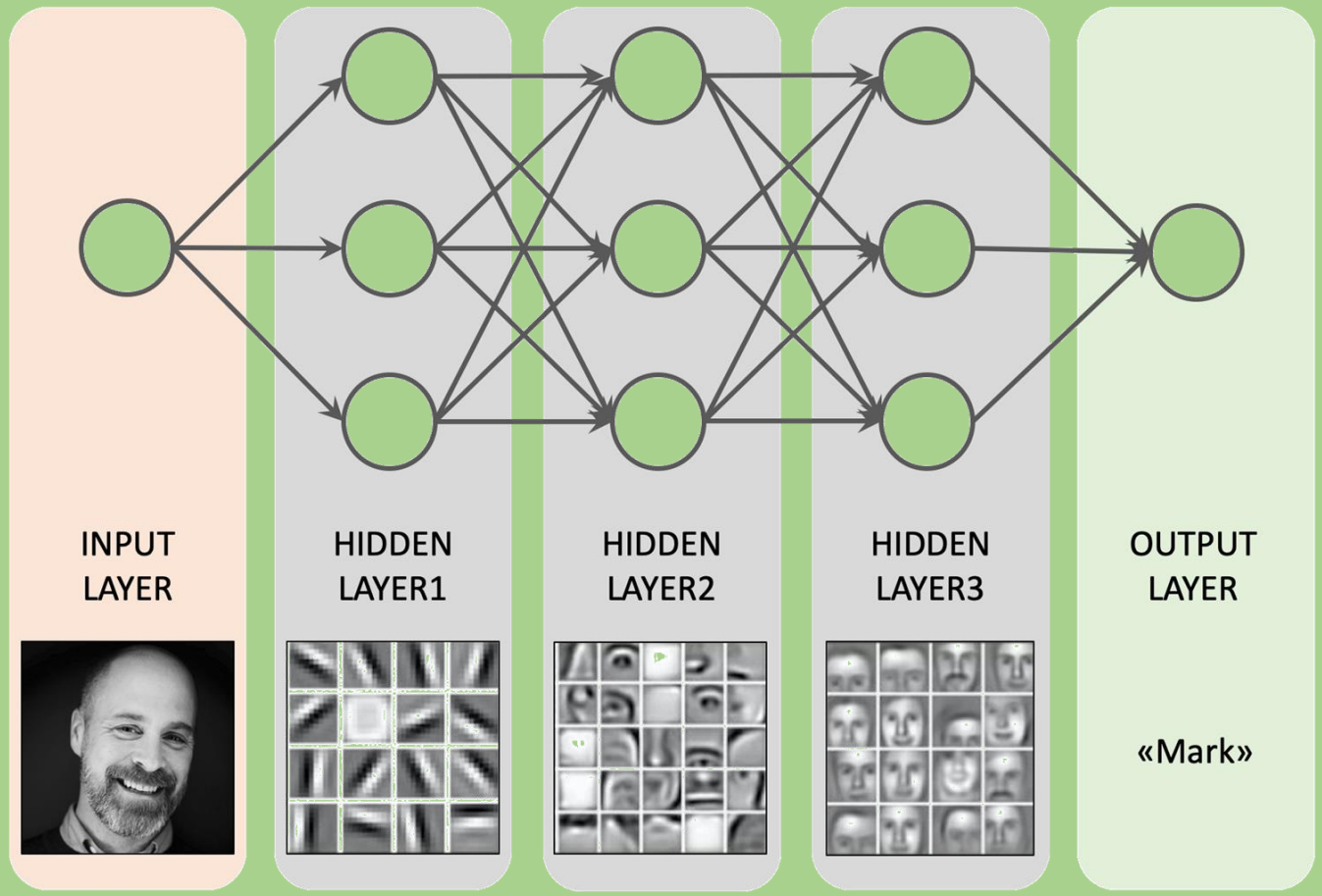


Microsoft

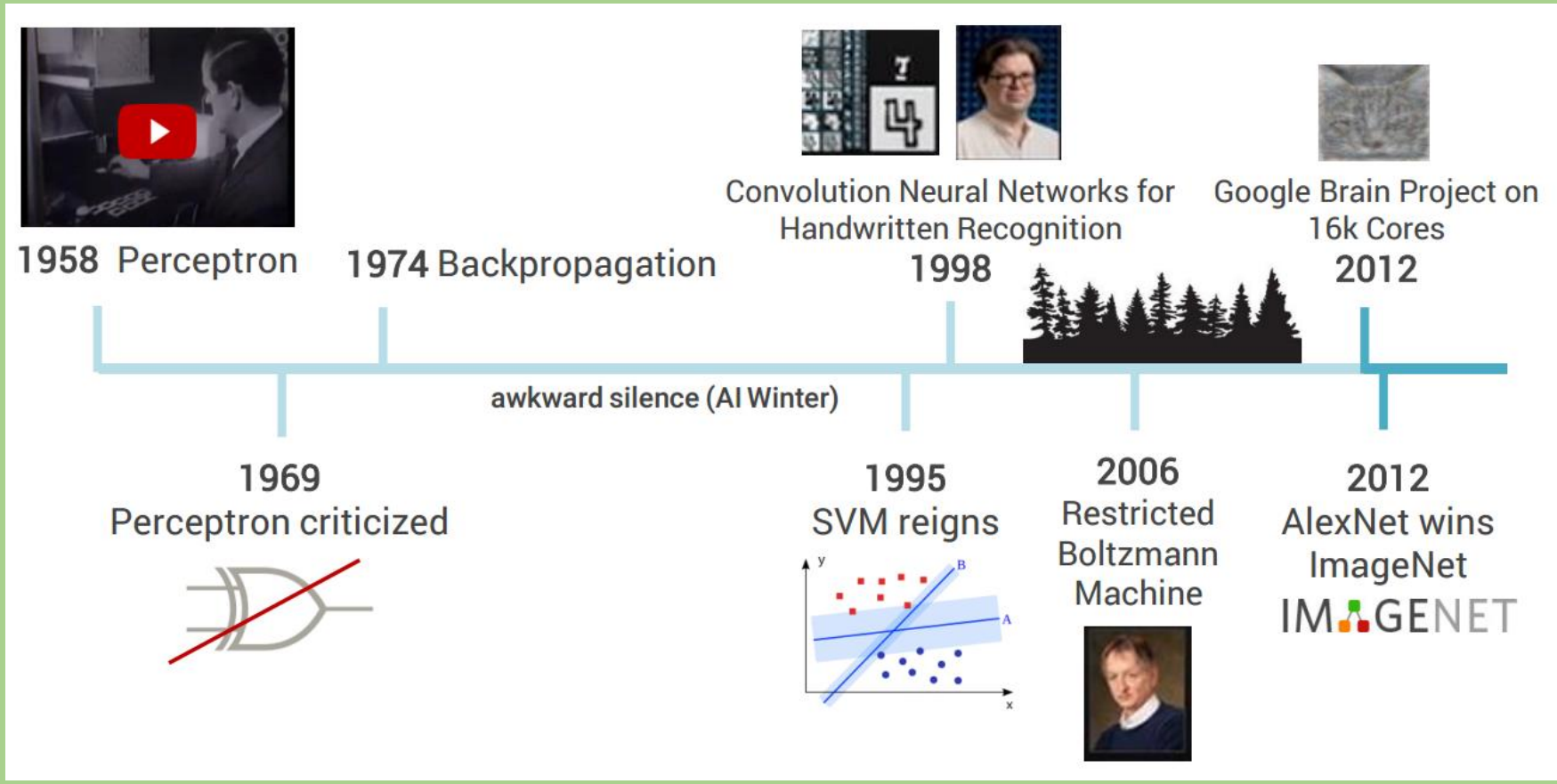
MSA
Microsoft Student Accelerator

Deep Learning / Neural Networks





Deep Learning



Neural Information Processing Systems Foundation

NIPS : Conferences : 2009 : Program

Li Deng, Dong Yu, Geoffrey Hinton

Microsoft Research; Microsoft Research; University of Toronto

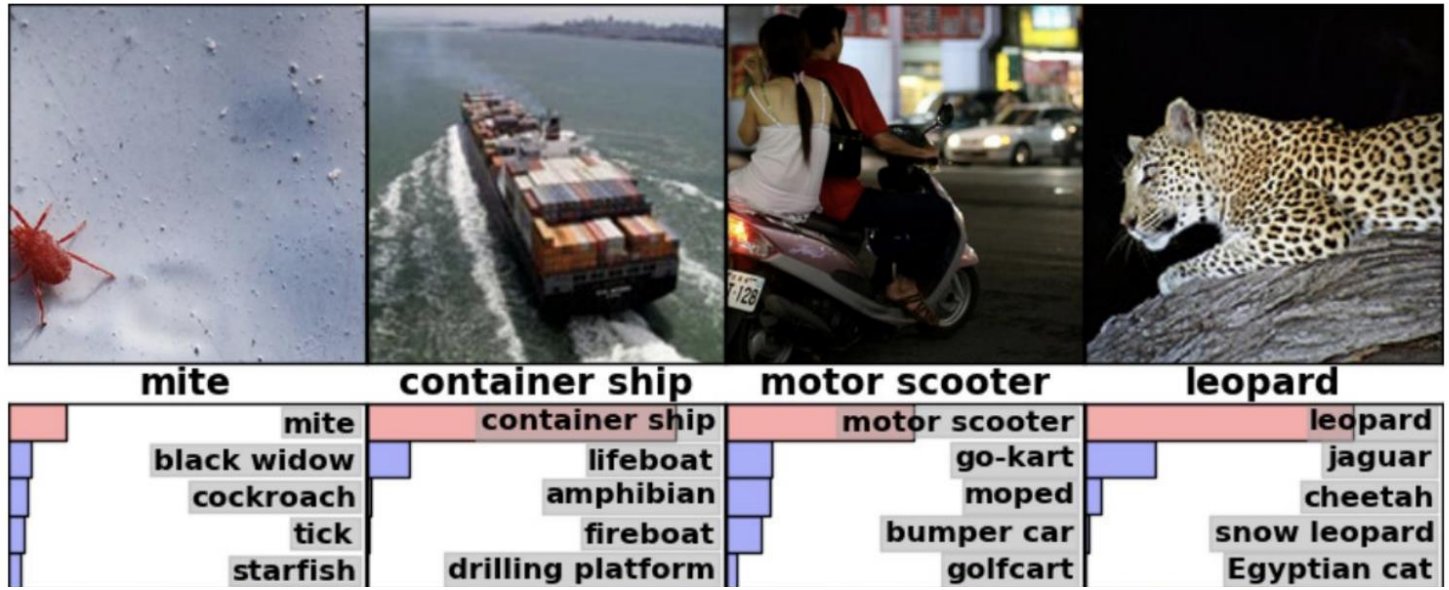
Deep Learning for Speech Recognition and Related Applications

7:30am - 6:30pm Saturday, December 12, 2009

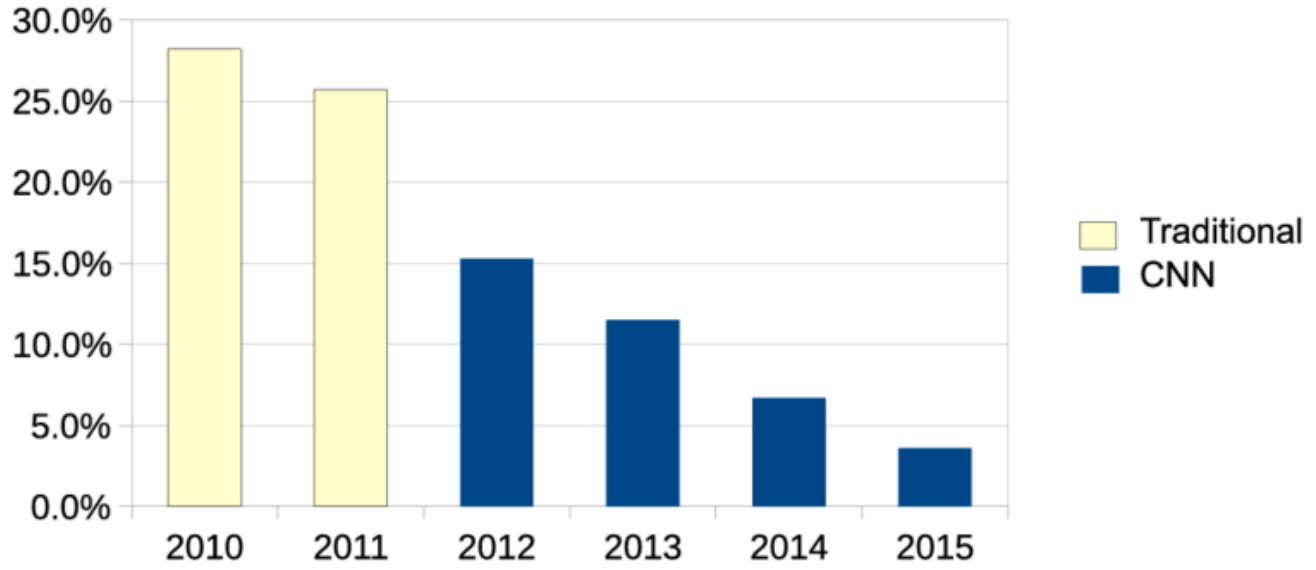
Location: Hilton: Cheakamus

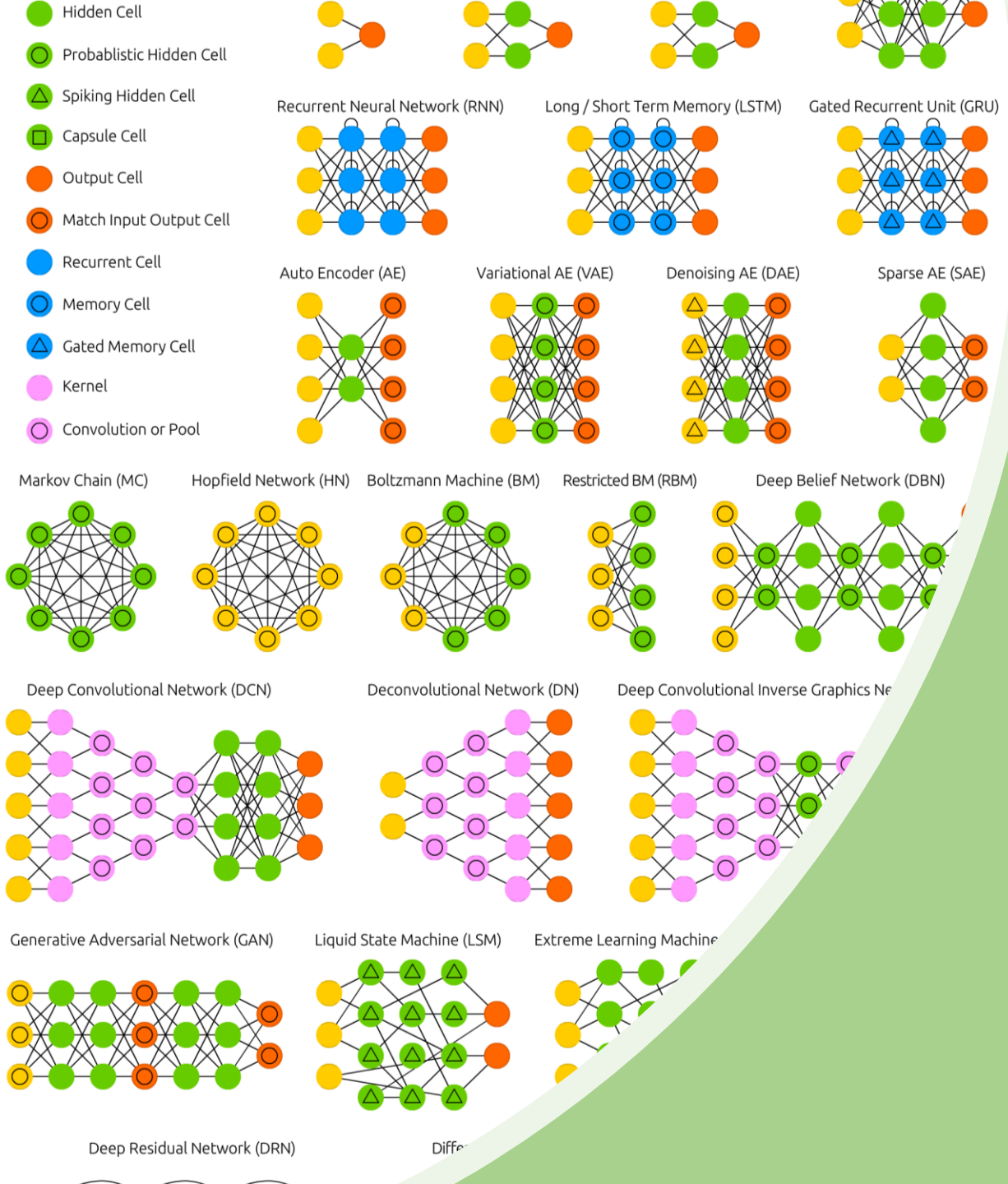
Abstract: Over the past 25 years or so, speech recognition technology has been dominated by a "shallow" architecture --- hidden Markov models (HMMs). Significant technological success has been achieved using complex and carefully engineered variants of HMMs. The next generation of the technology requires solutions to remaining technical challenges under diversified deployment environments. These challenges, not adequately addressed in the past, arise from the many types of variability present in the speech generation process. Overcoming these challenges is likely to require "deep" architectures with efficient learning algorithms. For speech recognition and related sequential pattern recognition applications, some attempts have been made in the past to develop computational architectures that are "deeper" than conventional HMMs, such as

ImageNet Challenge



- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.





Neural Networks

Time for some Math

Fish and Chips

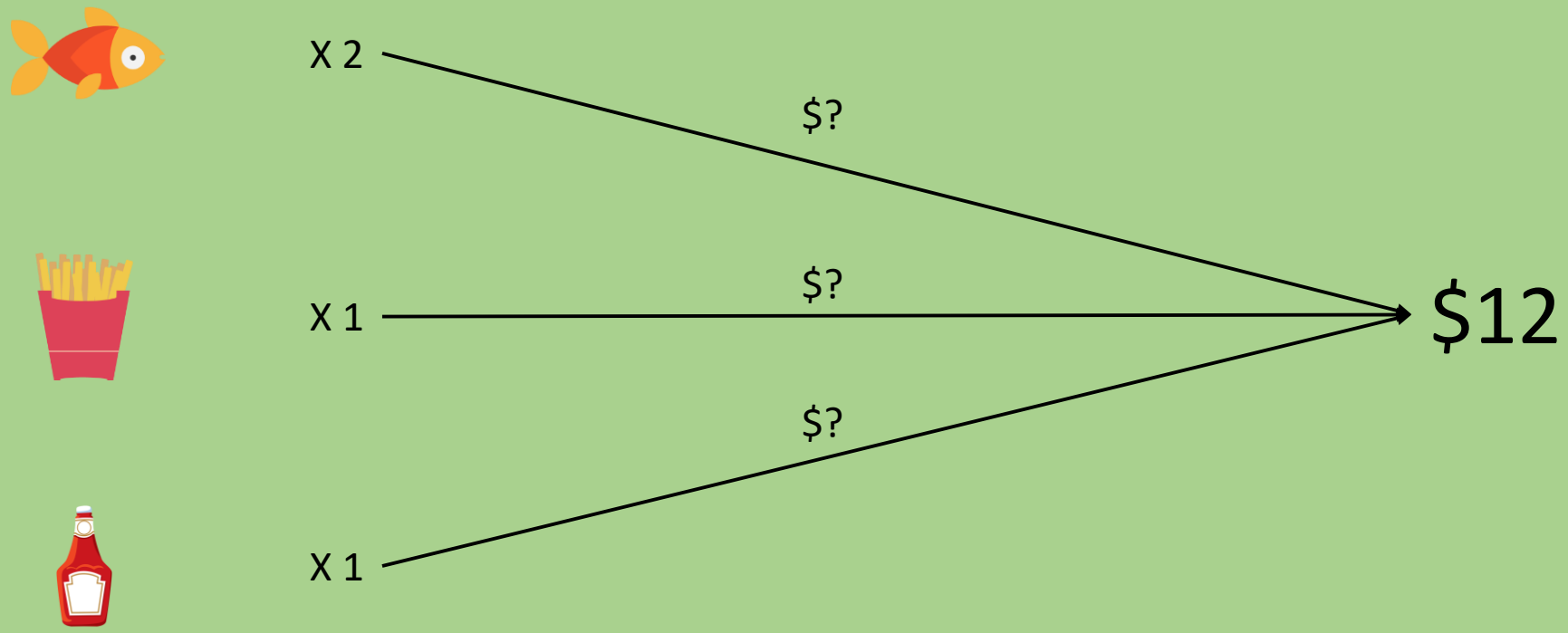
X 2

X 1

$12

X 1

Fish and Chips



Dataset



Fish	Chips	Sauce	Total
2	1	1	$12
3	2	4	$15
1	1	0	$7
0	2	2	$8
1	1	0	$7

Linear Regression

X 2

X 1

X 1

m_1

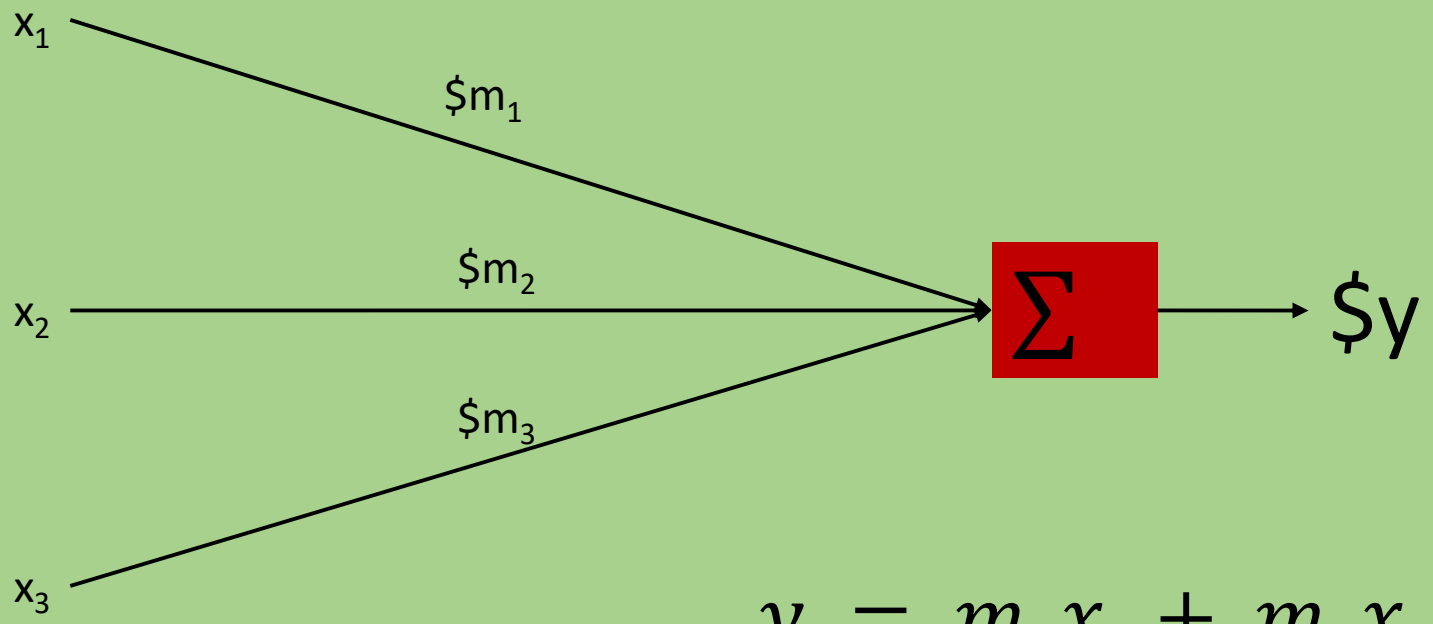
m_2

m_3

+

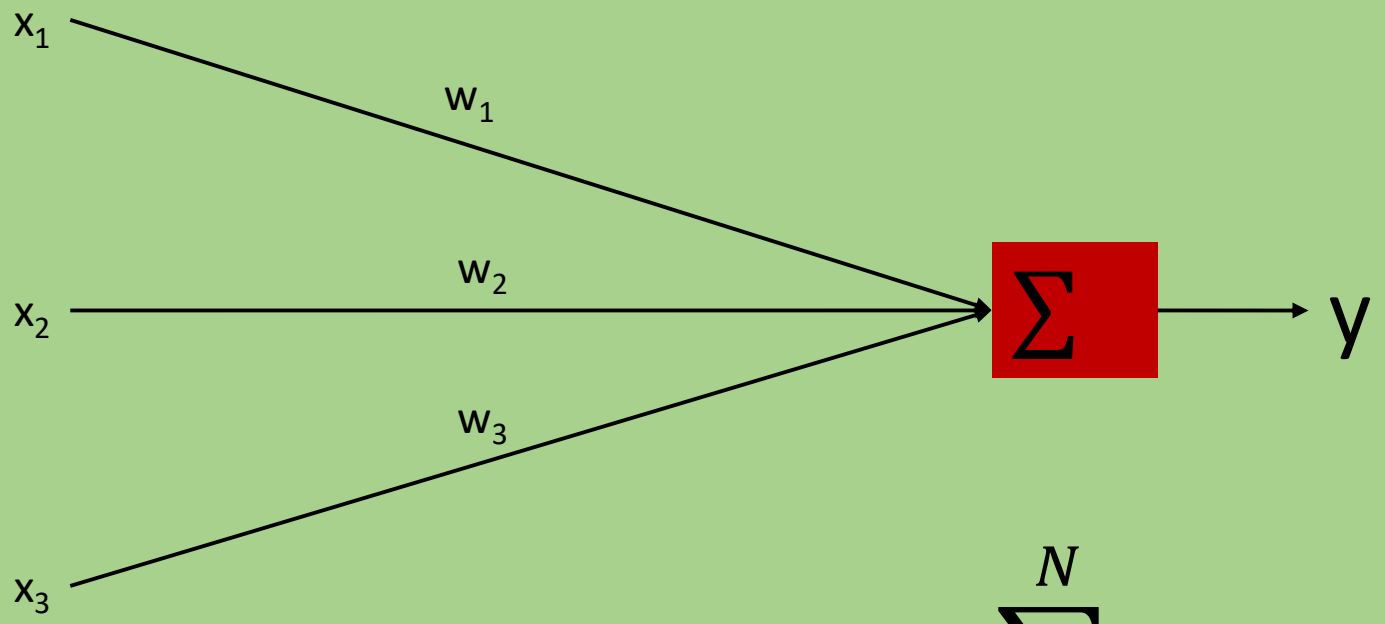
$12

Linear Regression



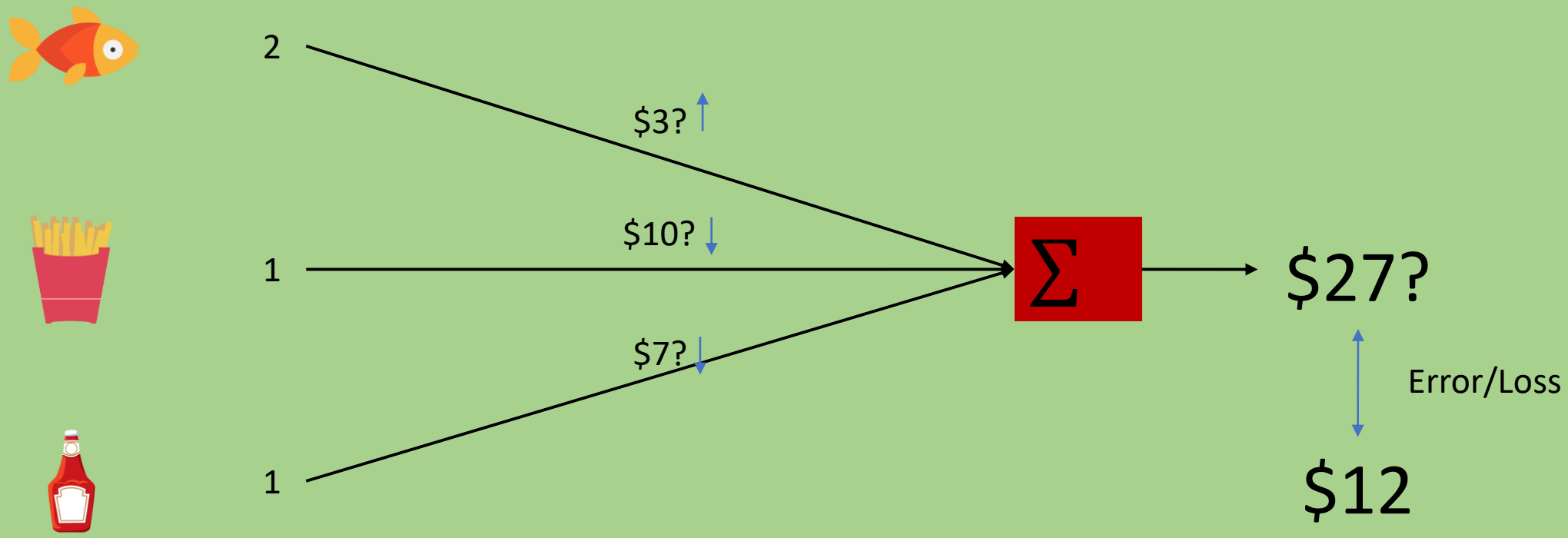
$$y = m_1x_1 + m_2x_2 + m_3x_3$$

Linear Regression



$$y = \sum_{i=1}^{N} w_i x_i$$

Fish and Chips



Loss Functions - Regression

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

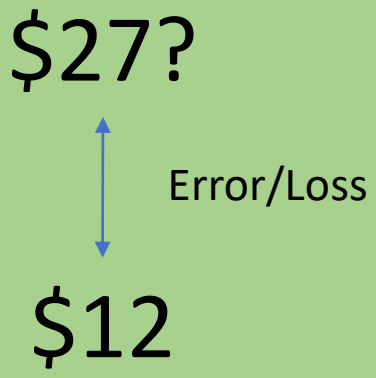
$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

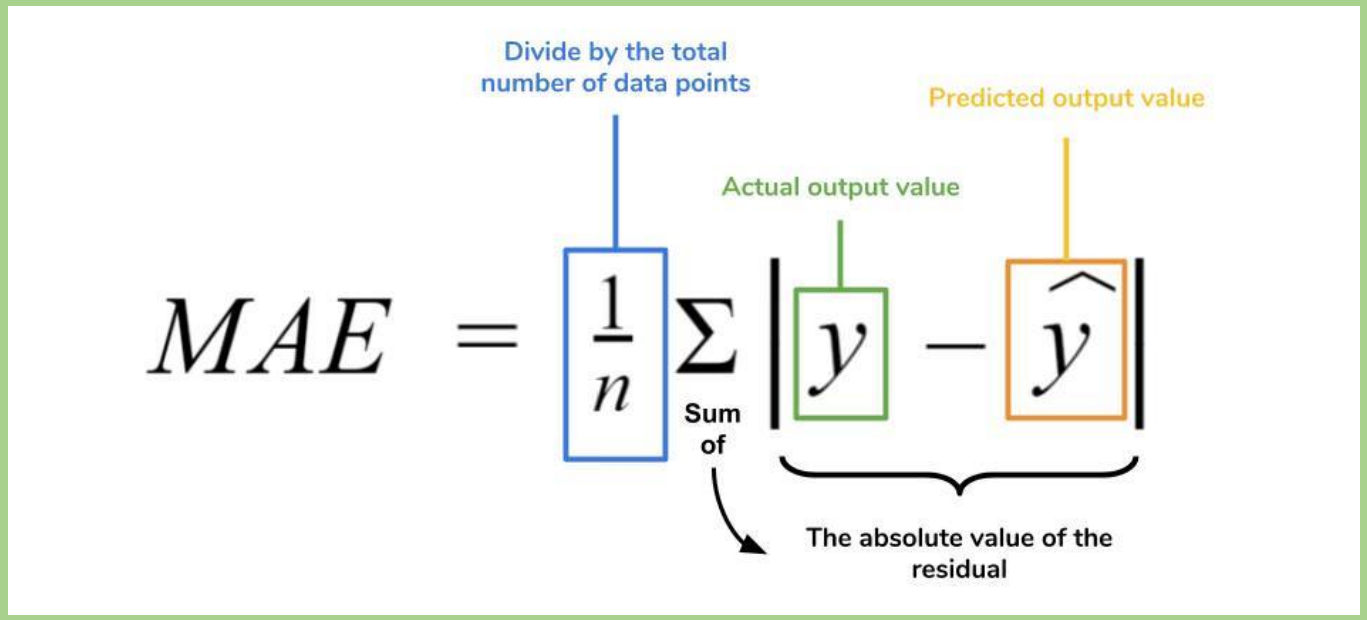
Where,

$\hat{y}$ – *predicted value of y*

$\bar{y}$ – *mean value of y*



Error Metrics



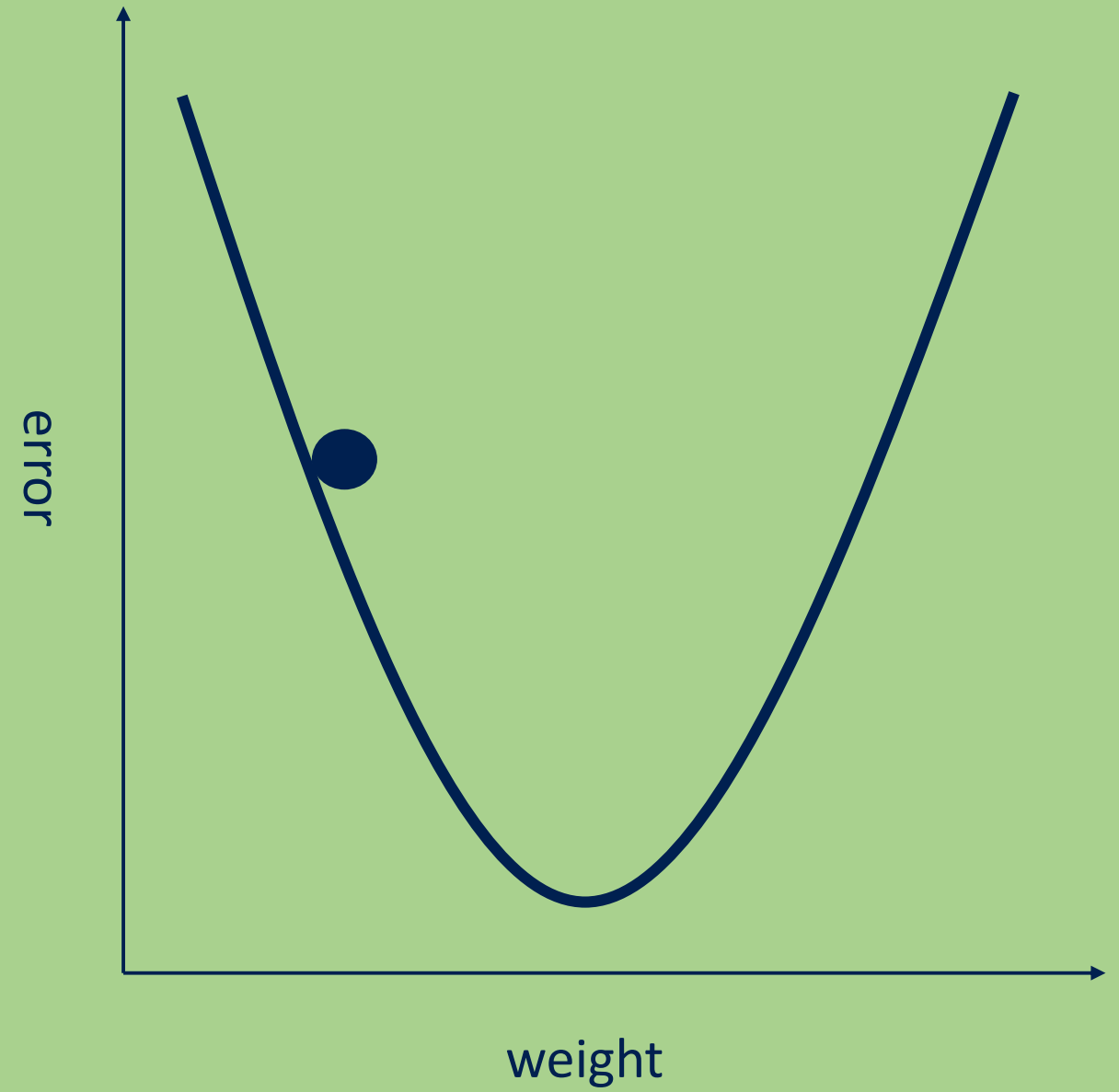
$27?

Error/Loss

$12

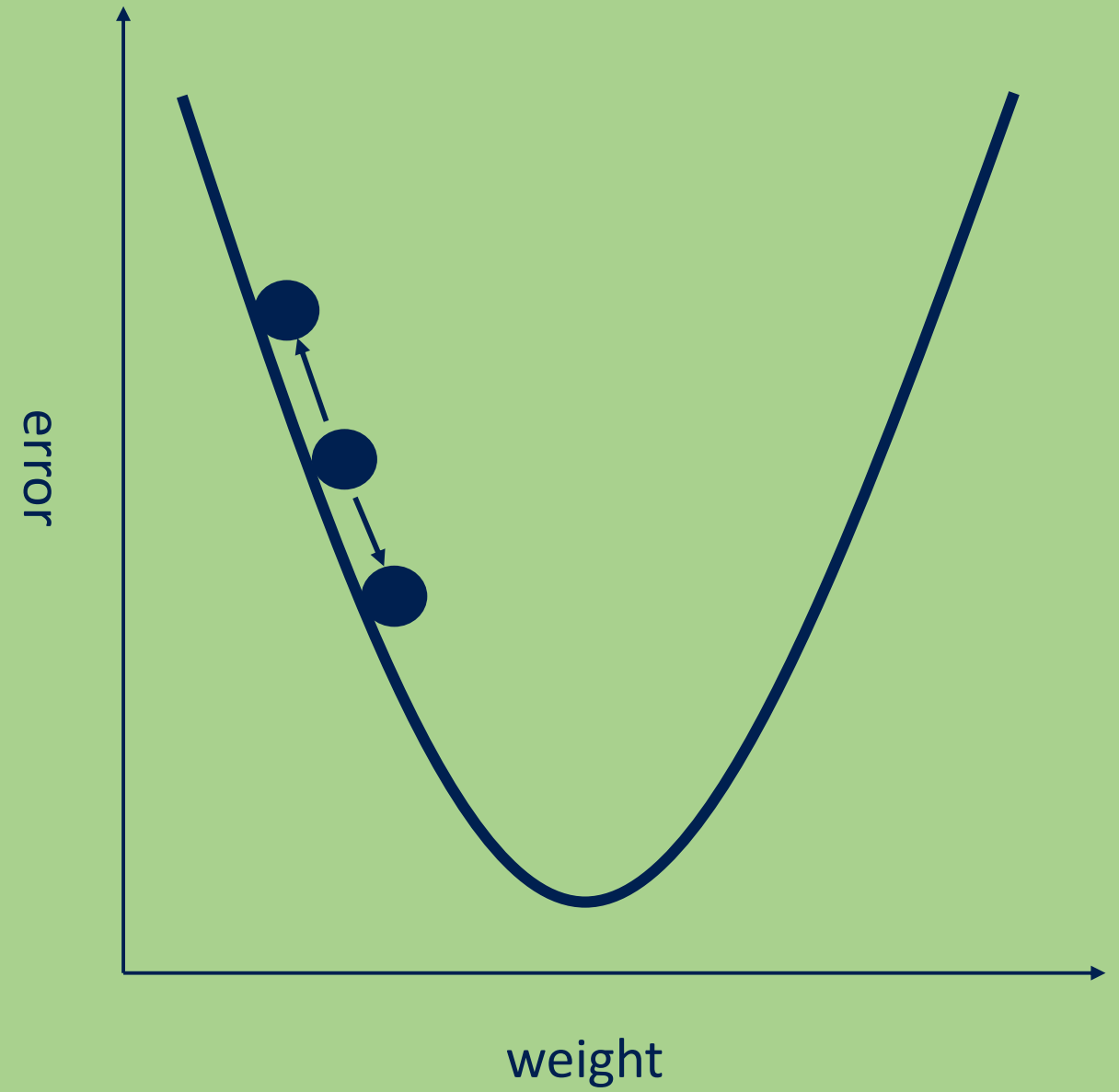
Gradient descent

For each input feature and voting weight, adjust it up and down a bit and see how the error changes.

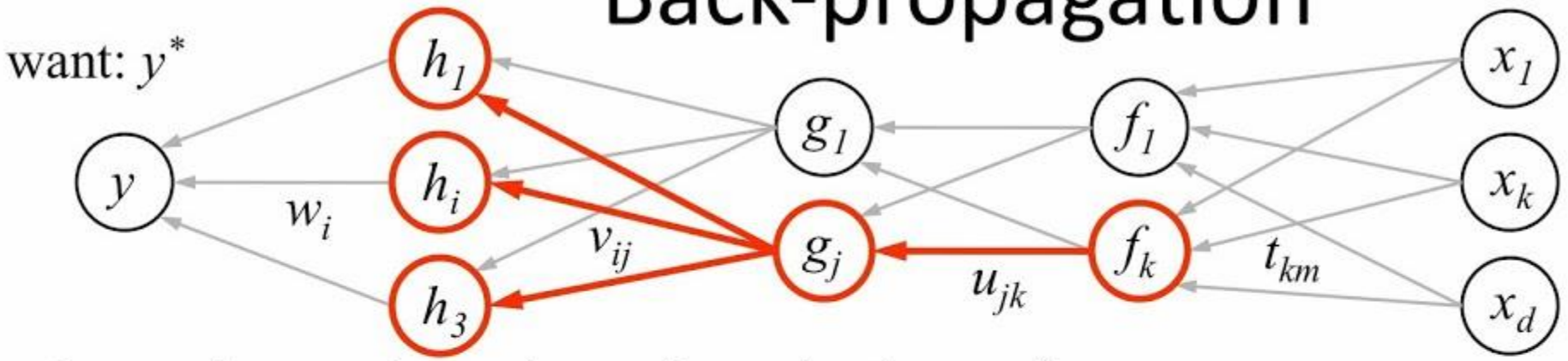


Gradient descent

For each input feature and voting weight, adjust it up and down a bit and see how the error changes.



Back-propagation



1. receive new observation $\boldsymbol{x} = [x_1 \ldots x_d]$ and target y^*
2. **feed forward:** for each unit g_j in each layer 1…L
   compute g_j based on units f_k from previous layer: $g_j = \sigma\left(u_{j0} + \sum_k u_{jk} f_k\right)$
3. get prediction y and error $(y\text{-}y^*)$
4. **back-propagate error:** for each unit g_j in each layer L…1

(a) compute error on g_j

$$\underbrace{\frac{\partial E}{\partial g_j}}_{\text{should } g_j \text{ be higher or lower?}} = \sum_i \underbrace{\sigma'(h_i) v_{ij}}_{\text{how } h_i \text{ will change as } g_j \text{ changes}} \underbrace{\frac{\partial E}{\partial h_i}}_{\text{was } h_i \text{ too high or too low?}}$$

(b) for each u_{jk} that affects g_j

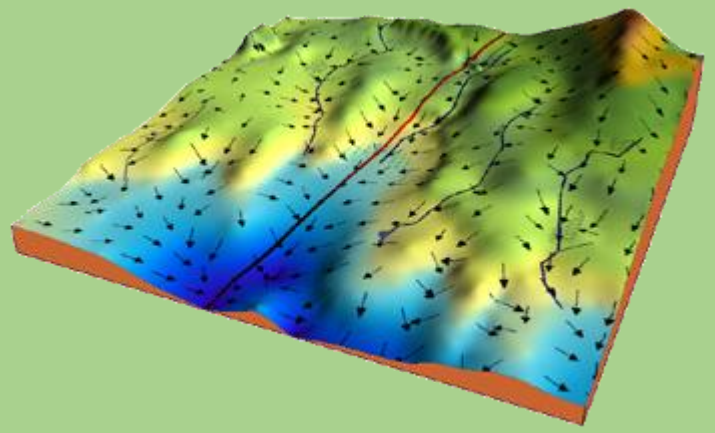
(i) compute error on u_{jk}

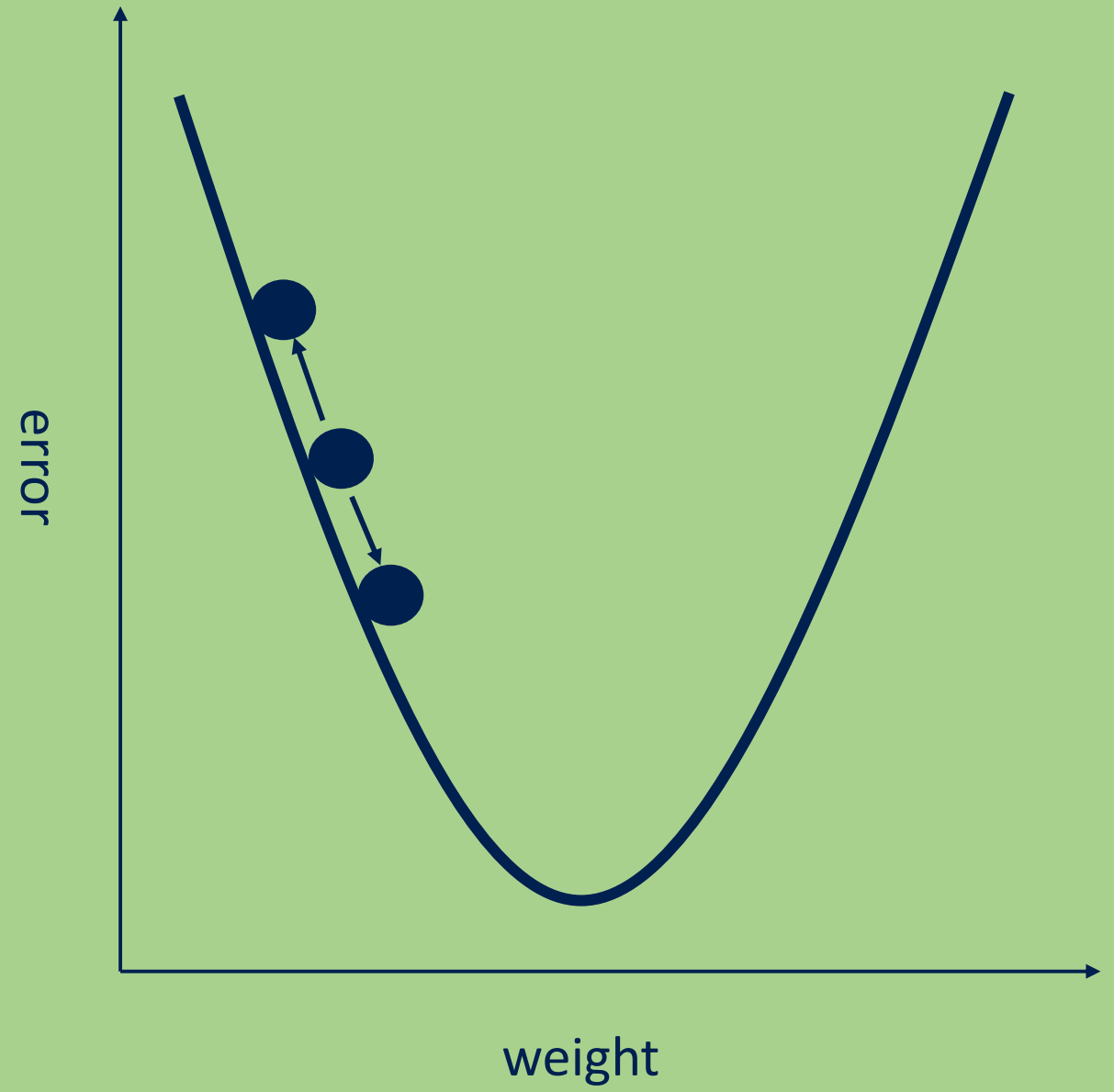
$$\frac{\partial E}{\partial u_{jk}} = \underbrace{\frac{\partial E}{\partial g_j}}_{\text{do we want } g_j \text{ to be higher/lower}} \underbrace{\sigma'(g_j) f_k}_{\text{how } g_j \text{ will change if } u_{jk} \text{ is higher/lower}}$$

(ii) update the weight

$$u_{jk} \leftarrow u_{jk} - \eta \frac{\partial E}{\partial u_{jk}}$$

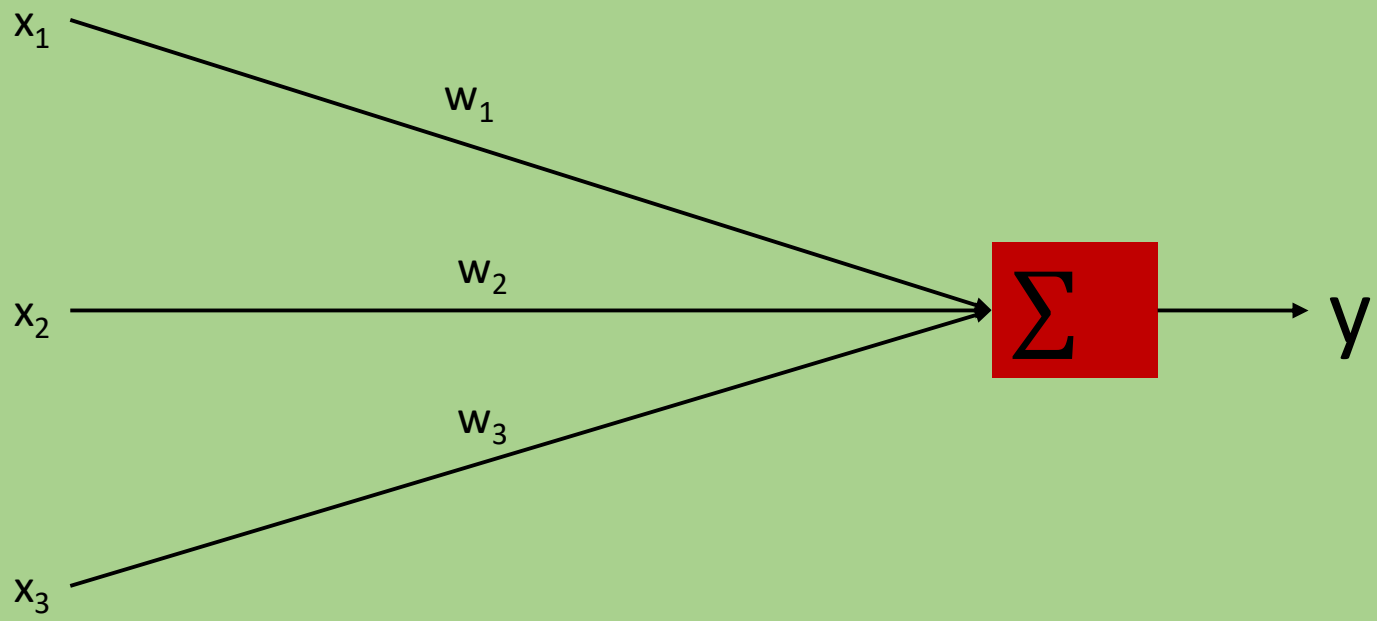
Gradient descent





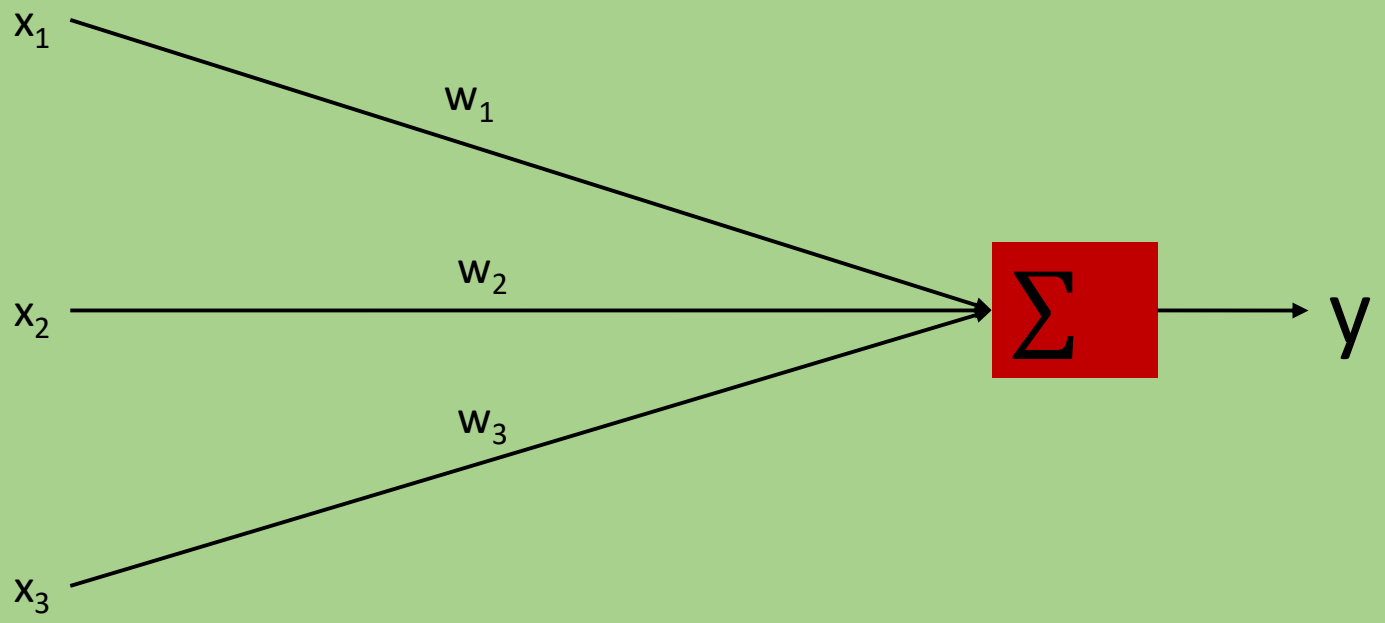
Weight Update

$$w^{new} = w^{old} + learning\ rate * gradient$$

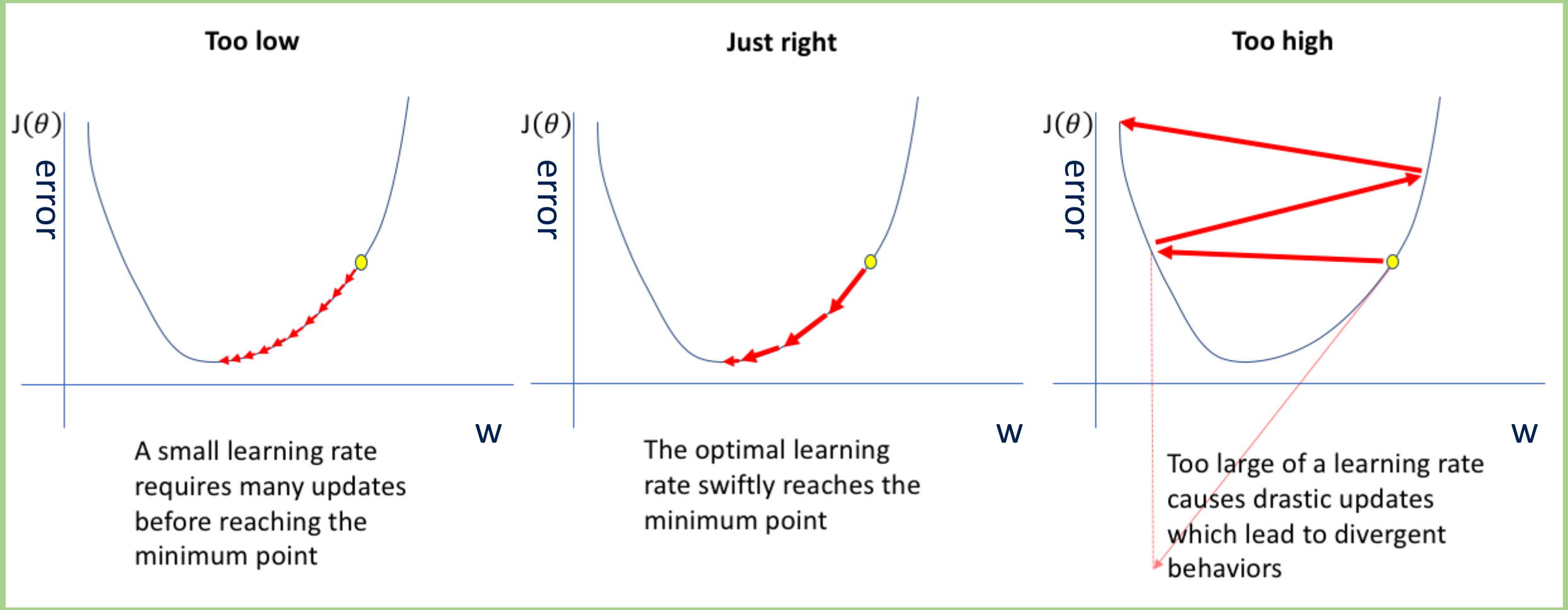


Weight Update

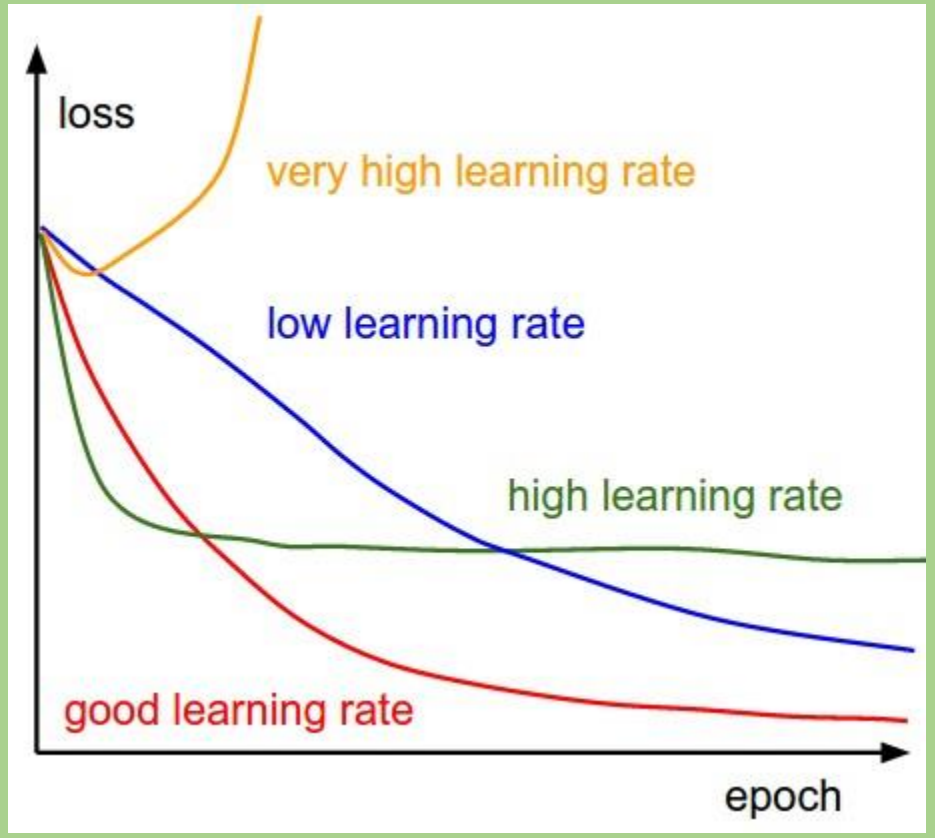
$$w^{new} = w^{old} + \alpha \frac{\Delta y}{\Delta x}$$



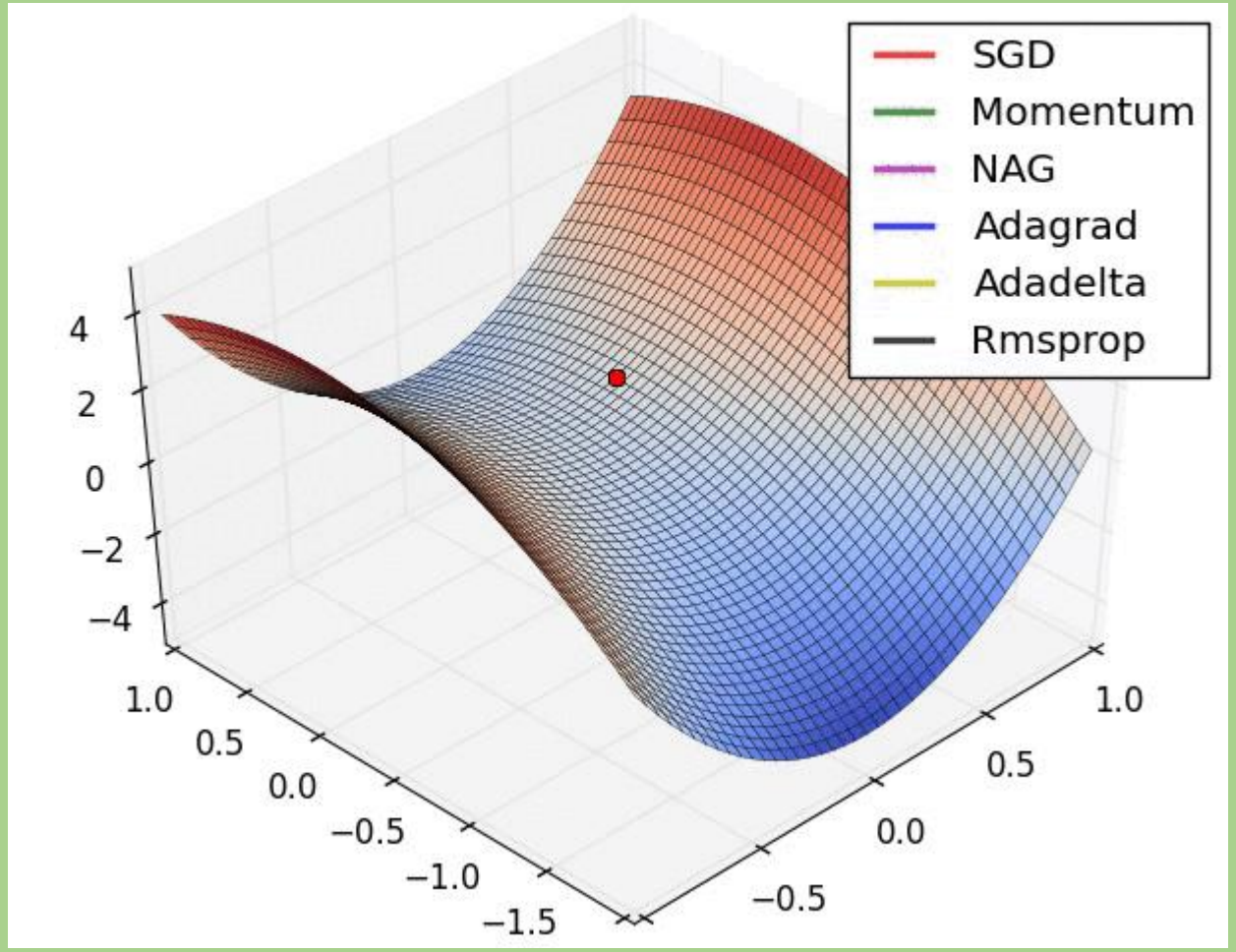
Learning Rate



Learning Rate



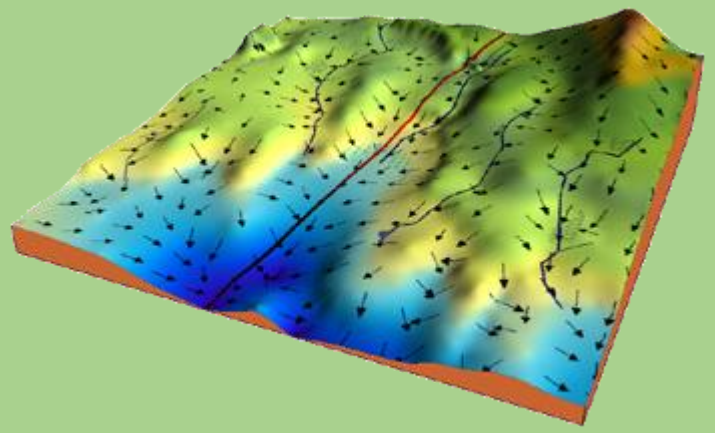
Different Optimizers

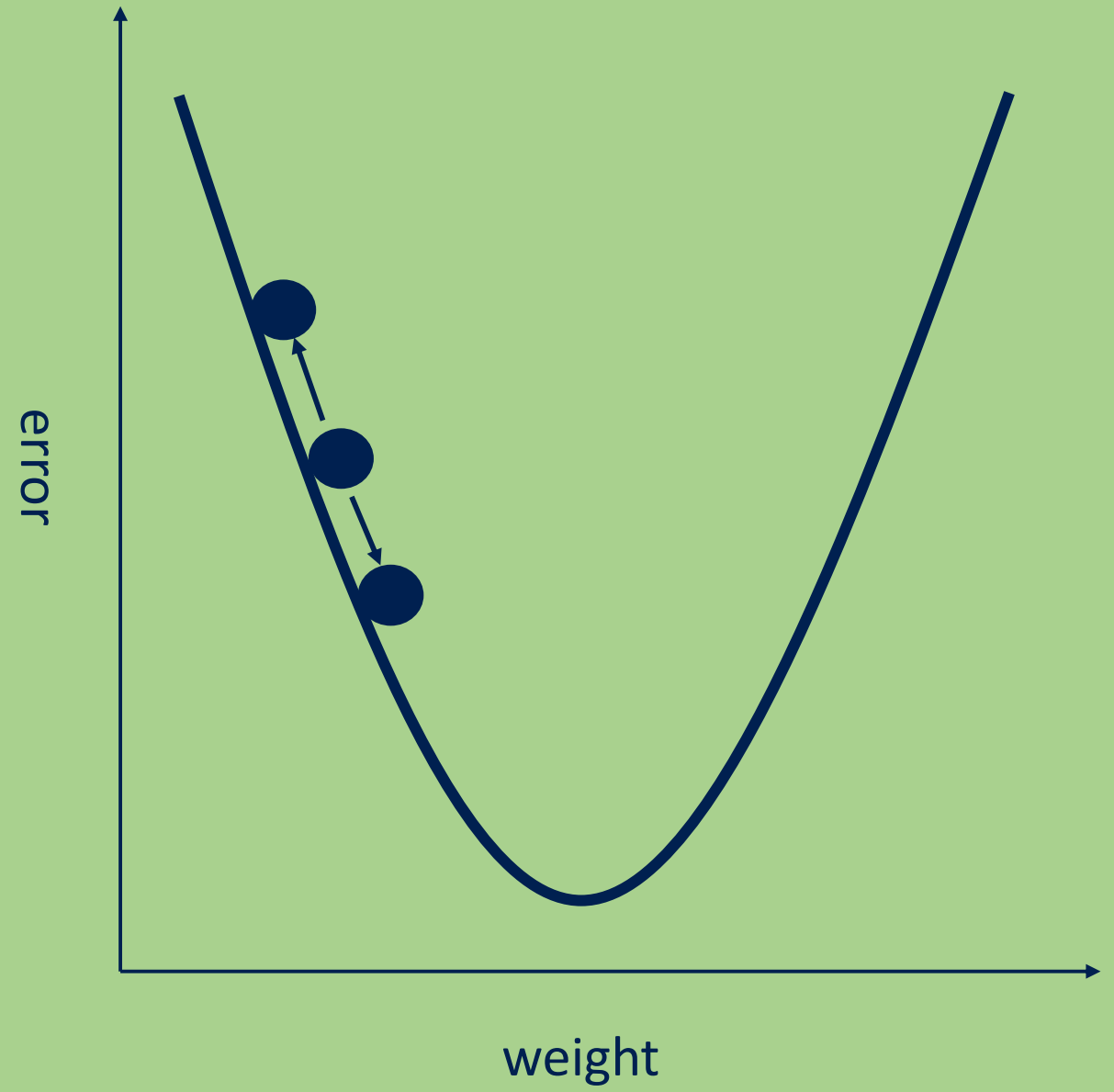


Microsoft

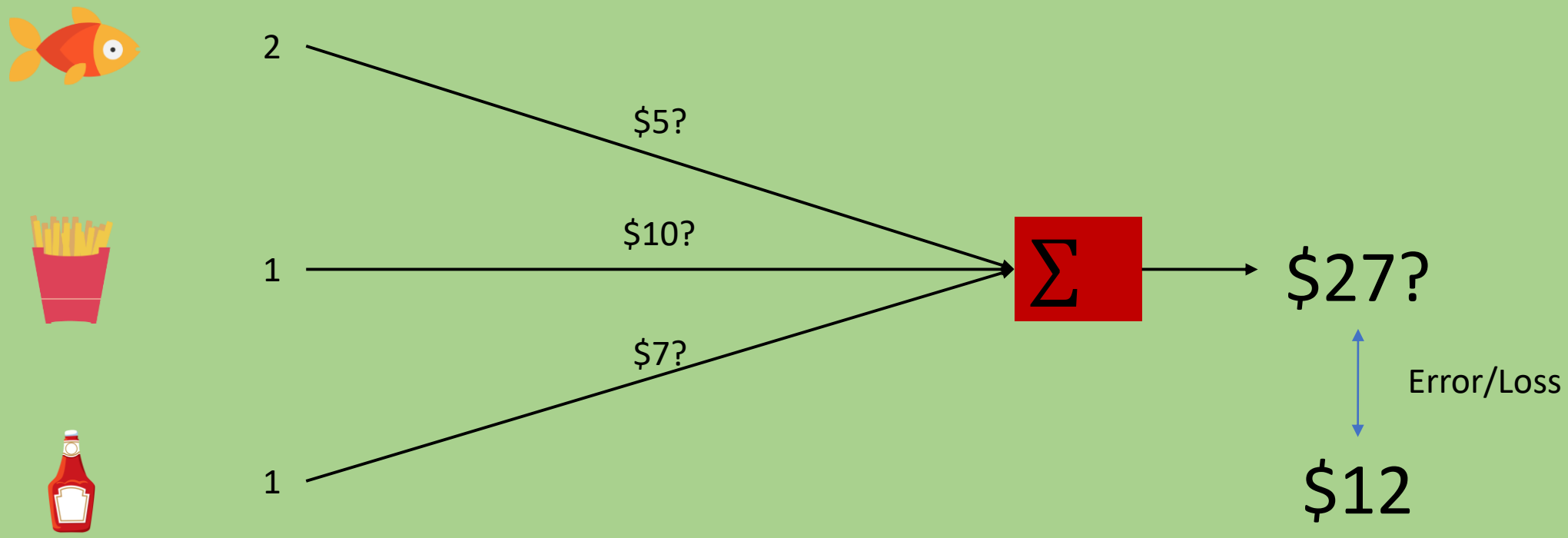
MSA
Microsoft Student Accelerator

Gradient descent

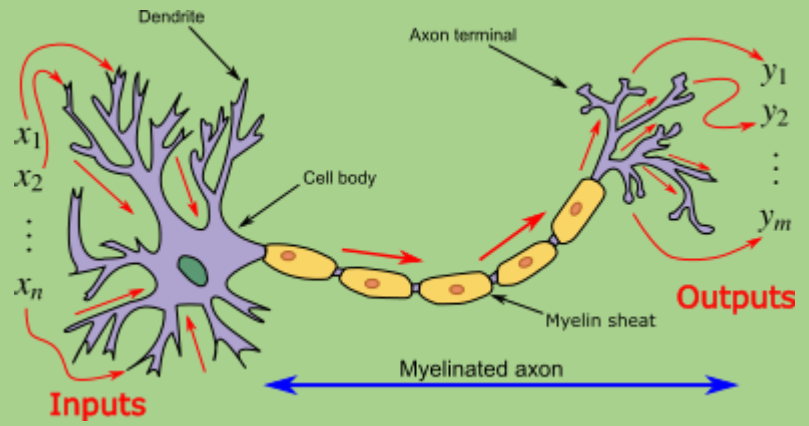


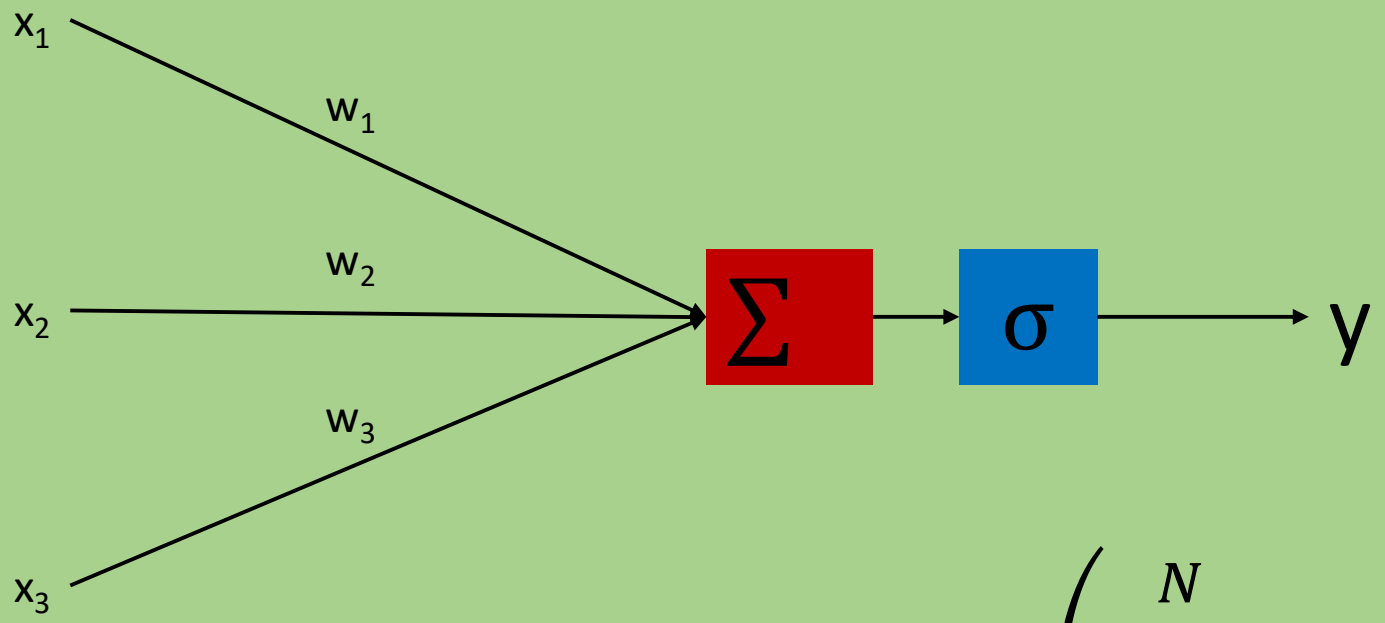


Fish and Chips



Neuron



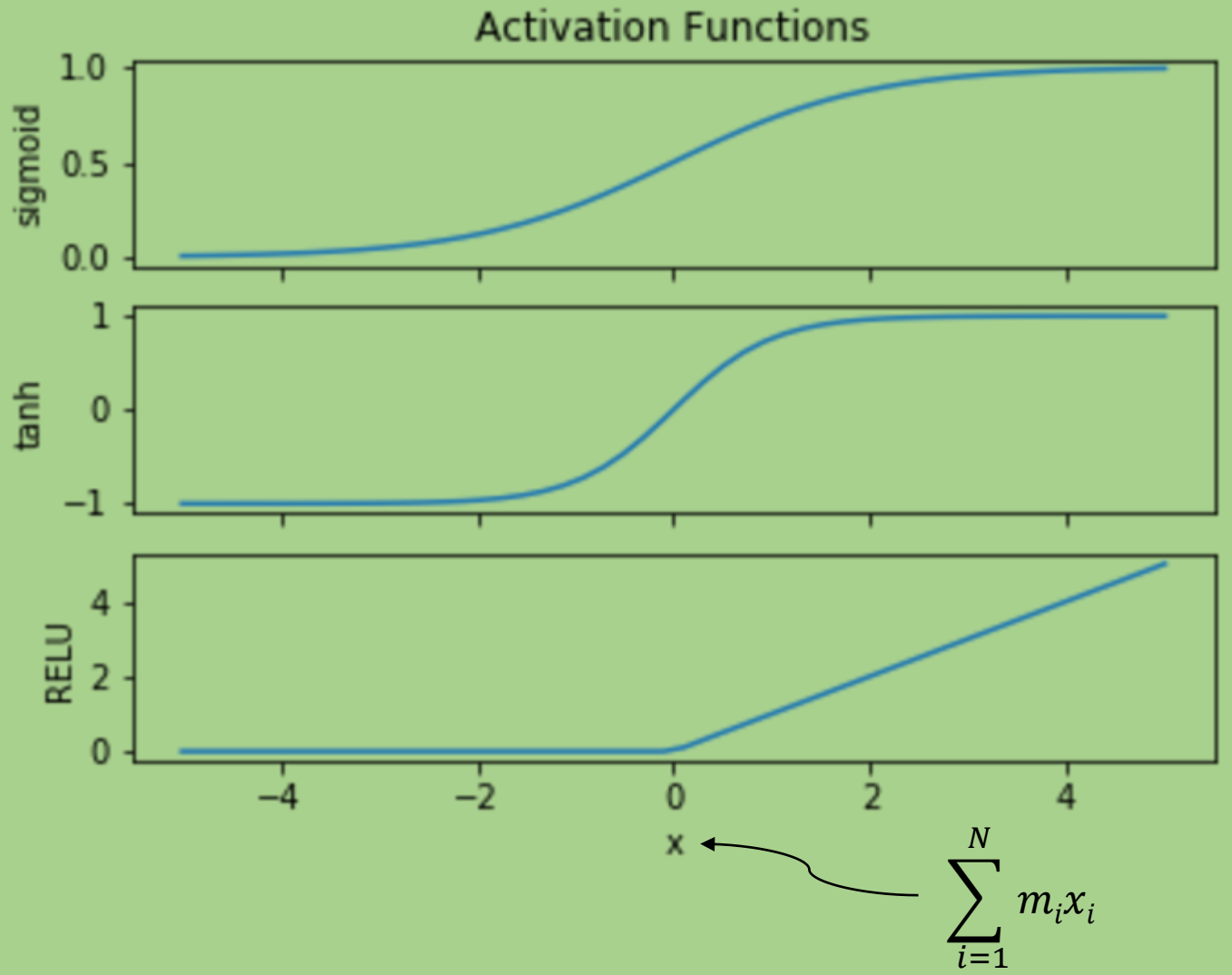


$$y = \sigma\left(\sum_{i=1}^{N} w_i x_i\right)$$

Microsoft

MSA
Microsoft Student Accelerator

σ

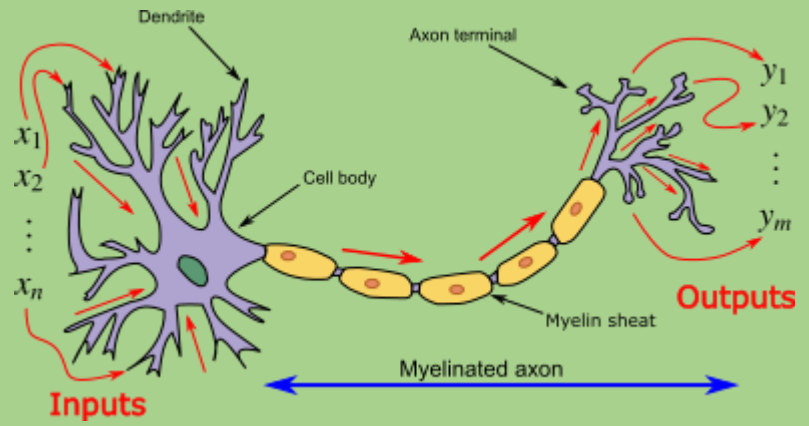


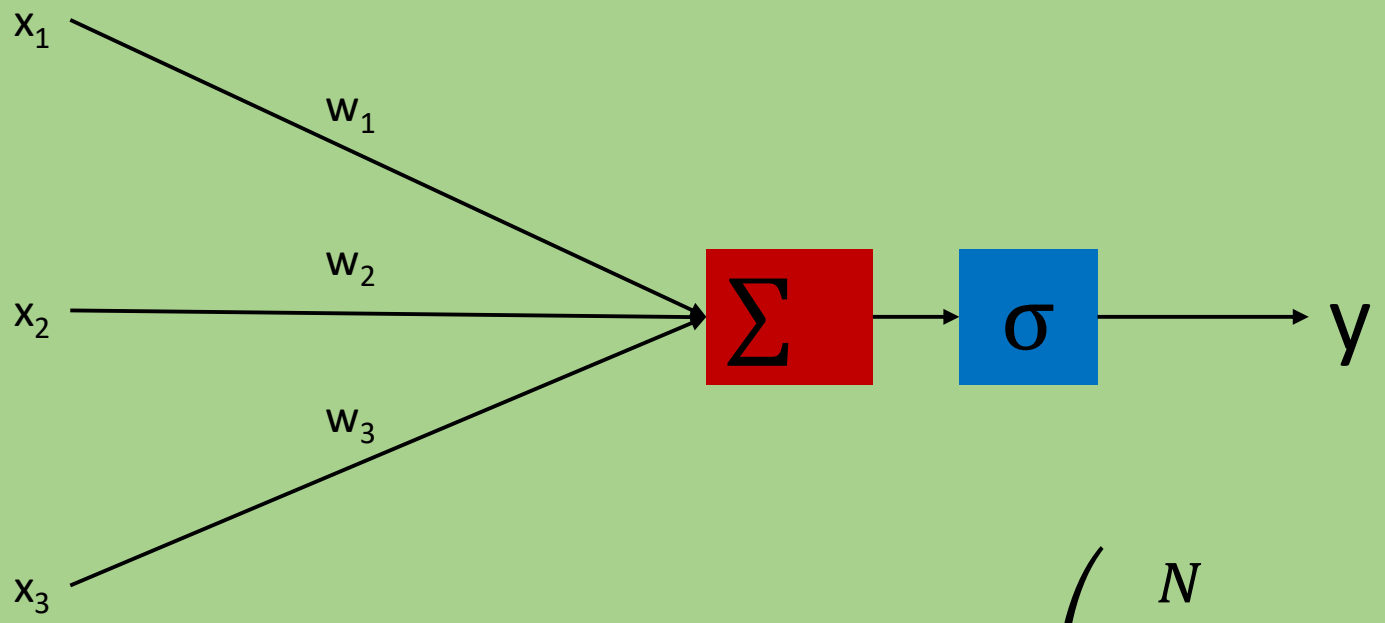
$$y = \sigma\left(\sum_{i=1}^{N} m_i x_i\right)$$

Microsoft

MSA
Microsoft Student Accelerator

Neuron



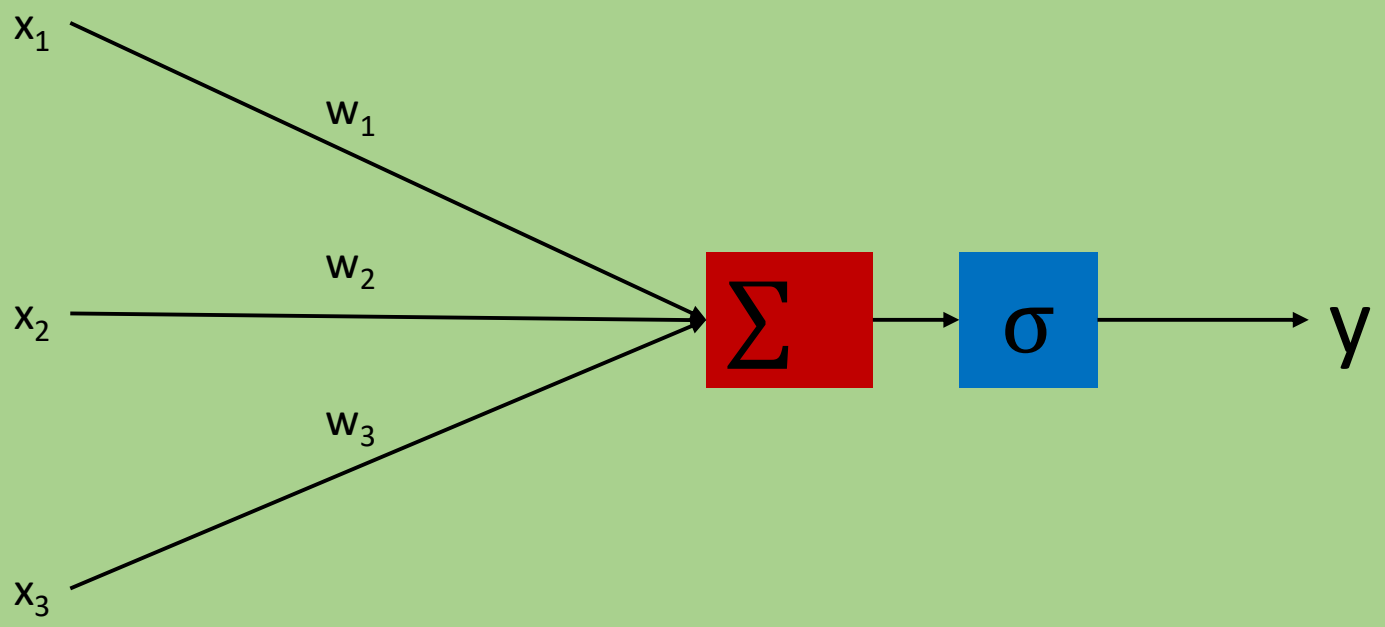


$$y = \sigma\left(\sum_{i=1}^{N} w_i x_i\right)$$

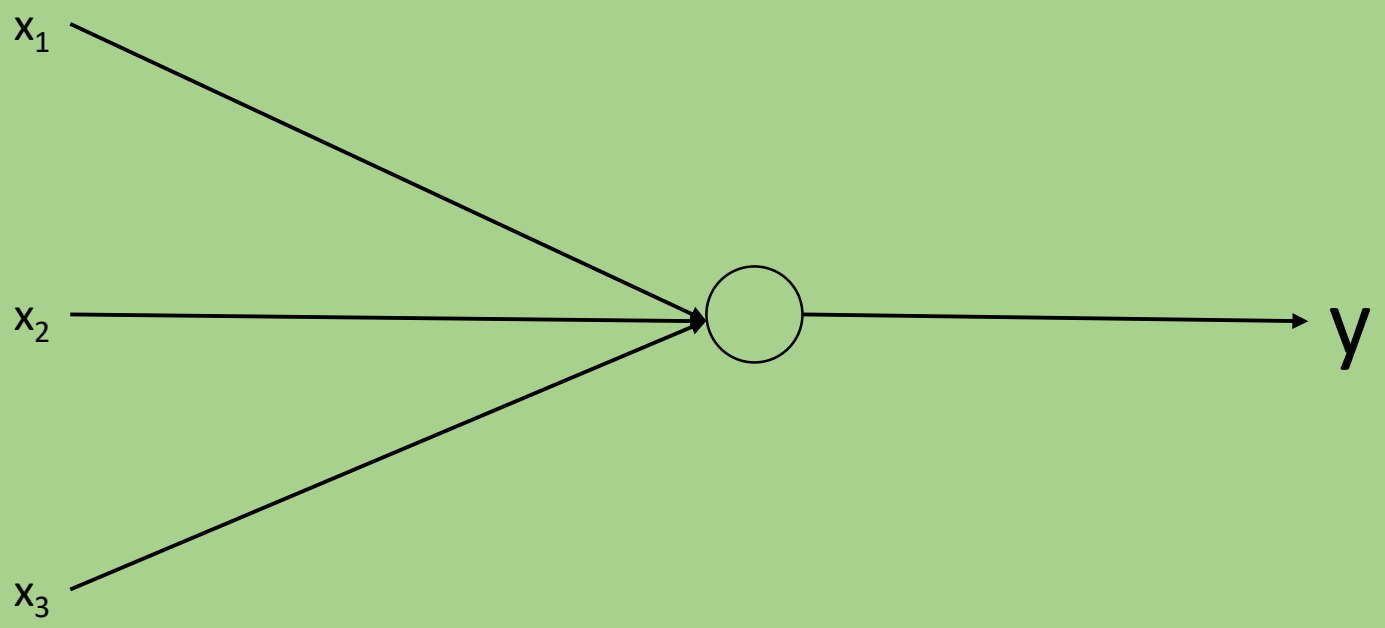
Microsoft

MSA
Microsoft Student Accelerator

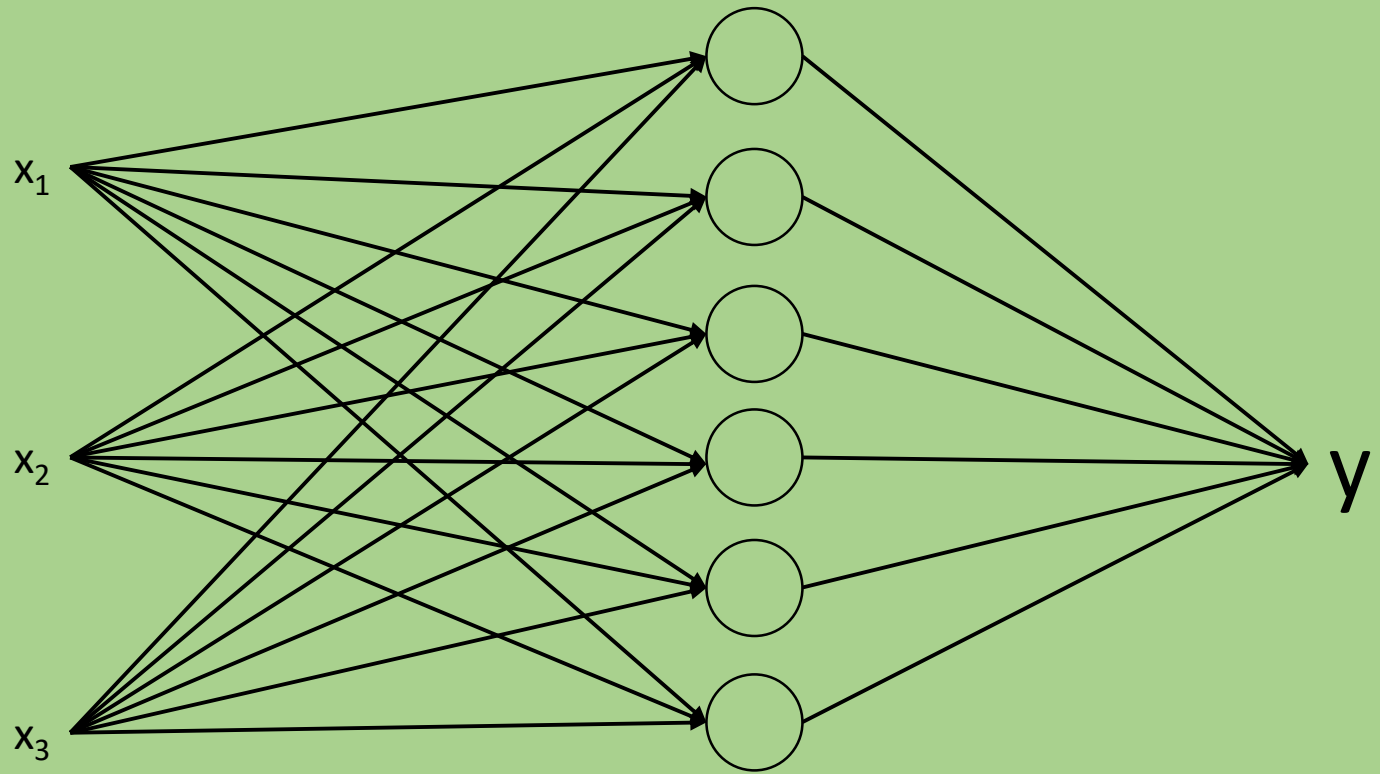
Neuron

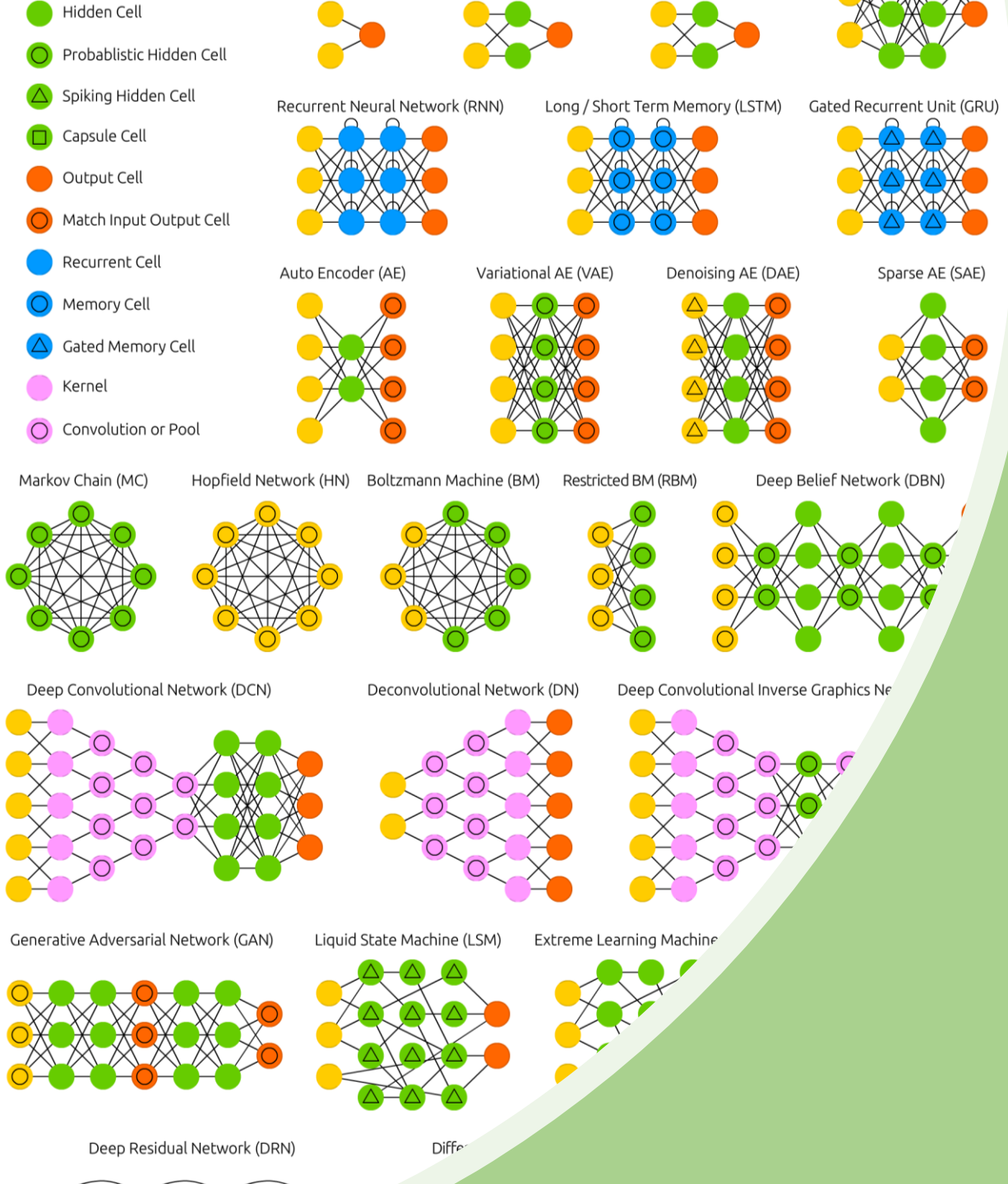


Neuron



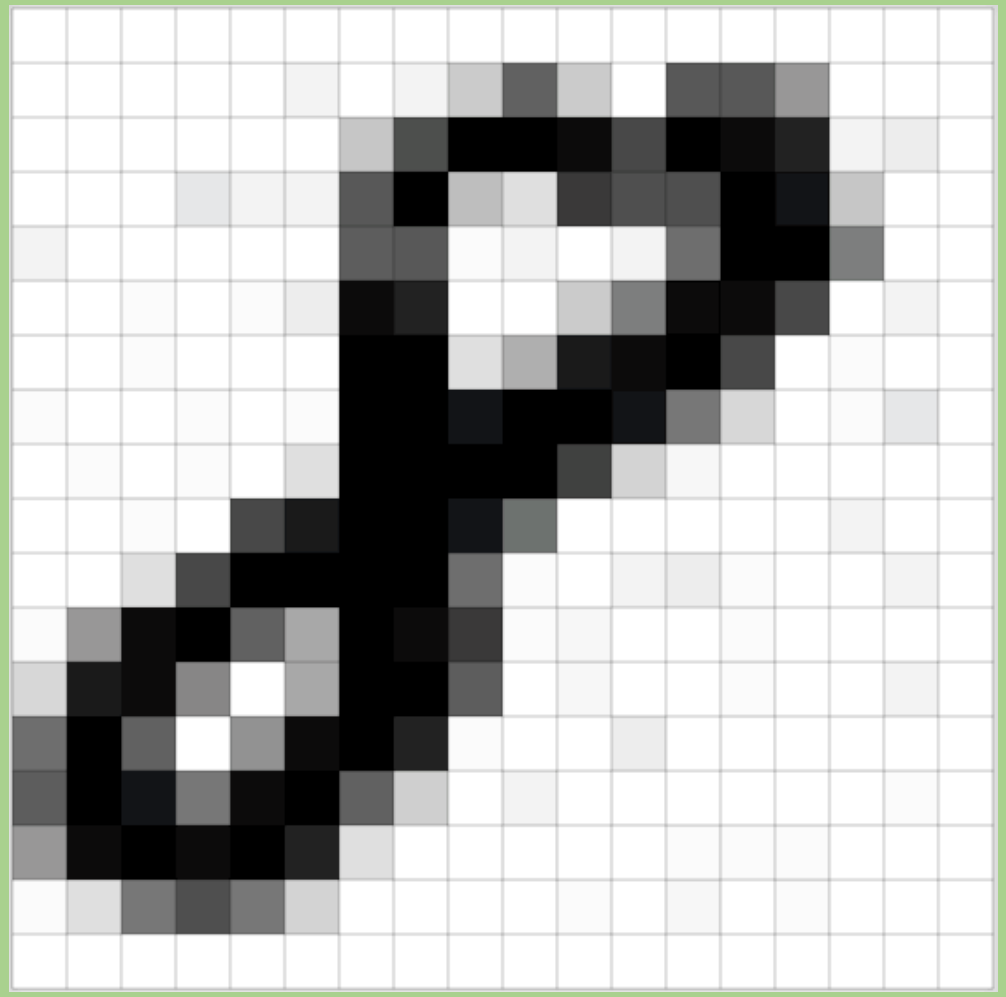
Neural Network





Neural Networks

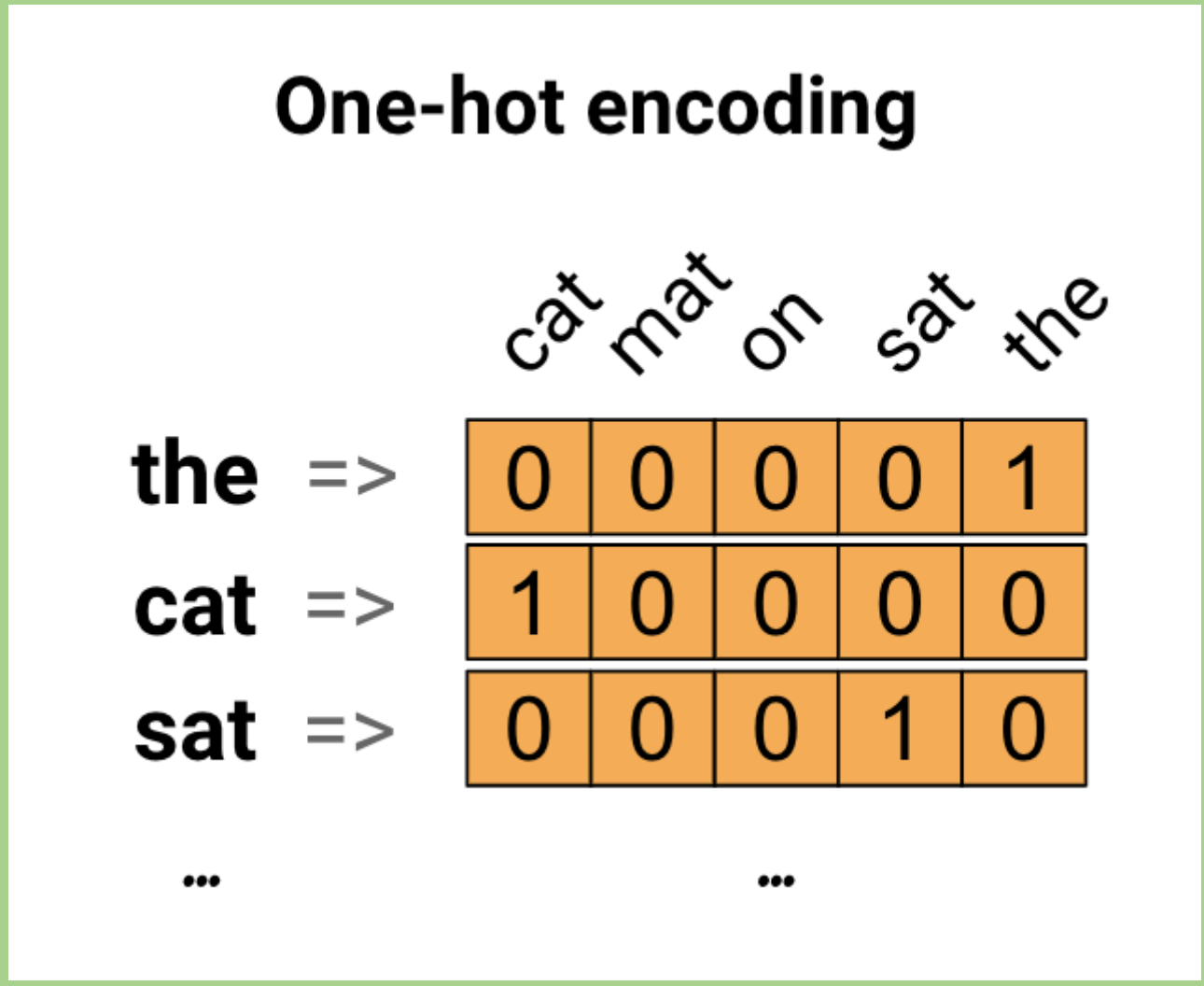
Can only deal with numbers



Microsoft

MSA
Microsoft Student Accelerator

The Cat sat on the Mat



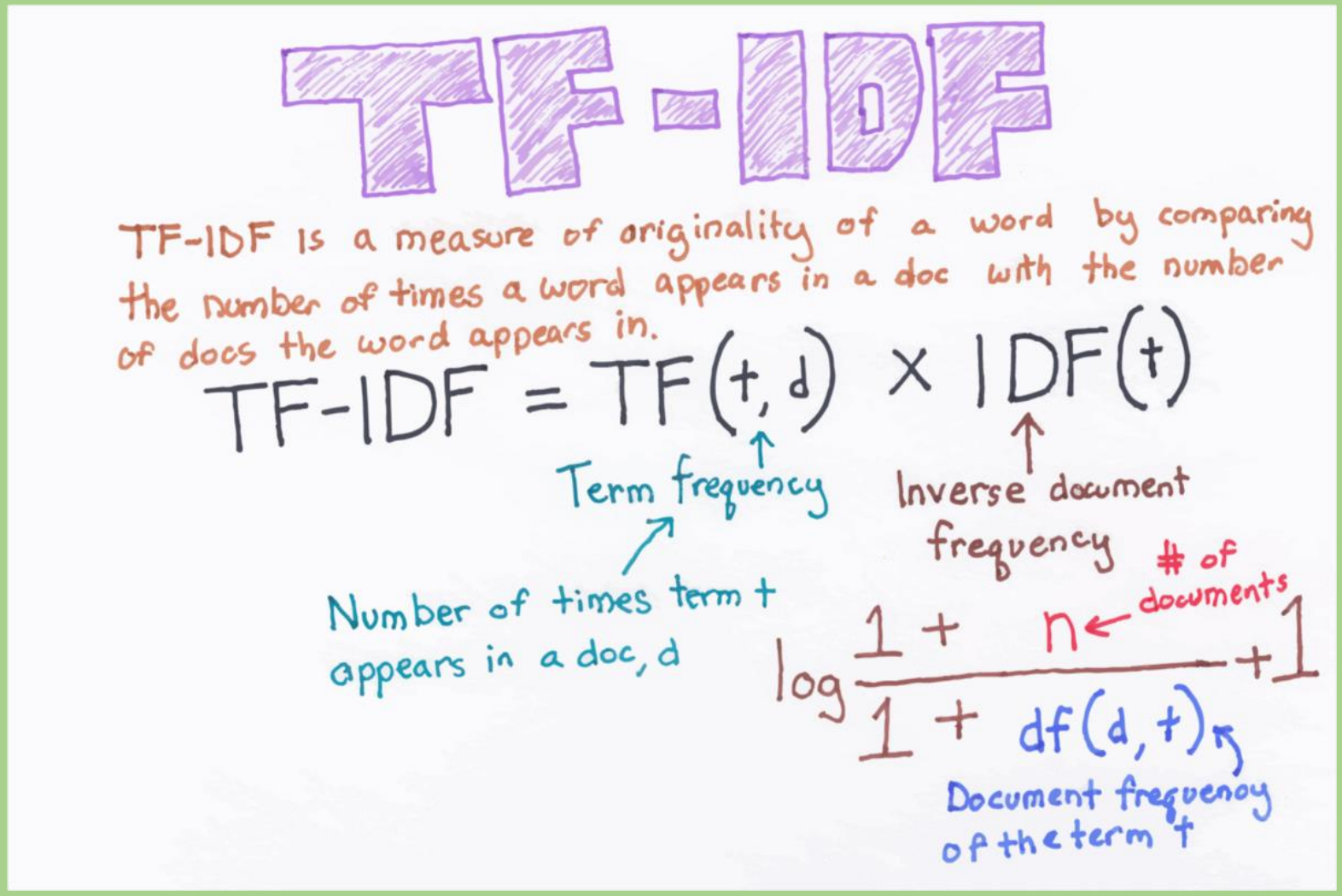
The Cat sat on the Mat

The Dog sat on my bed

	on	mat	sat	the	cat	my	dog	bed
0	1	1	1	2	1	0	0	0
1	1	0	1	1	0	1	1	1

TFIDF

- Rather than just counting, we can use the TF-IDF score of a word to rank it's importance.



TFIDF

The Cat sat on the Mat

The Dog sat on my bed

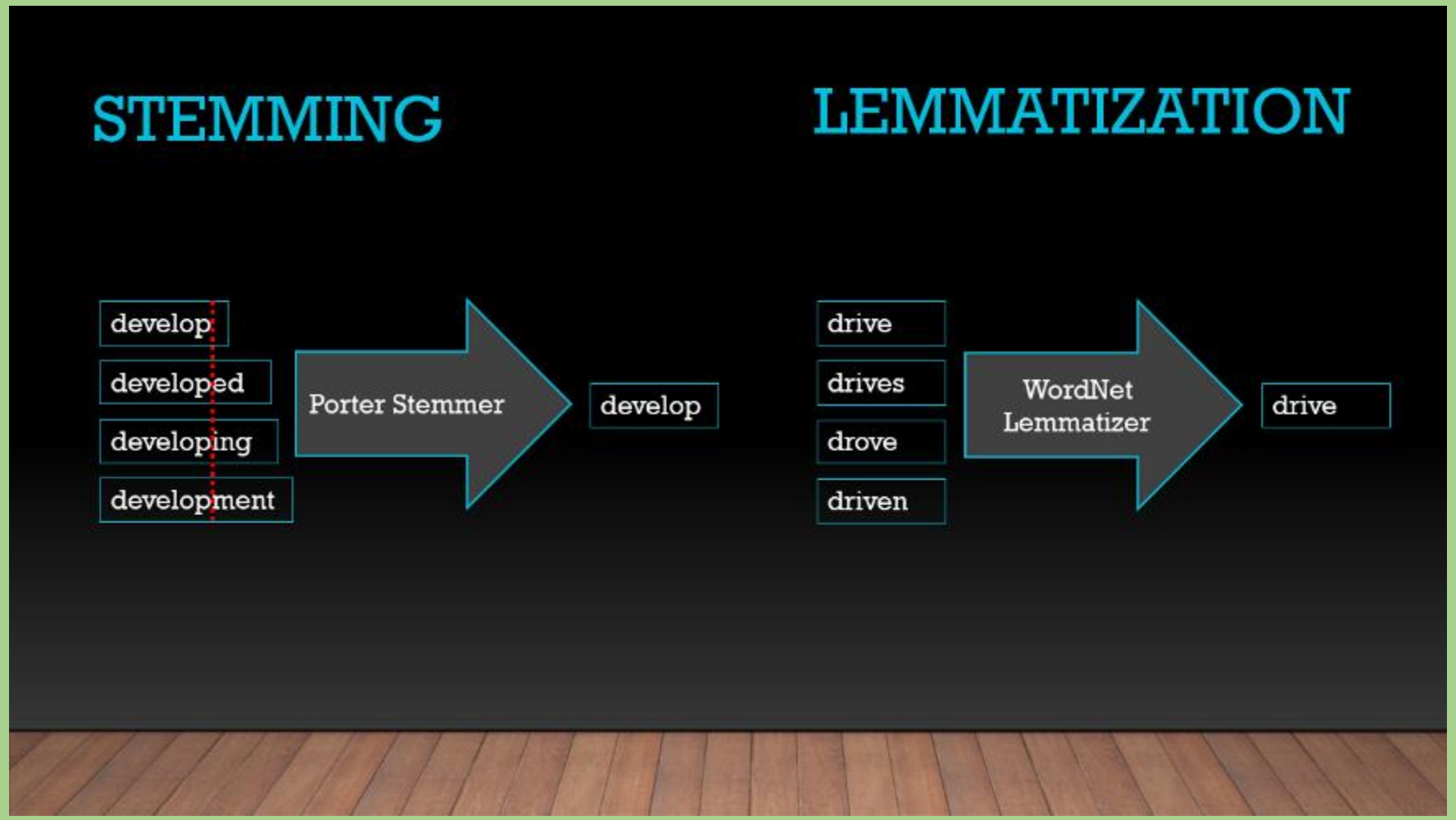
	on	mat	sat	the	cat	my	dog	bed
0	1	1	1	2	1	0	0	0
1	1	0	1	1	0	1	1	1

The Cat sat on the Mat

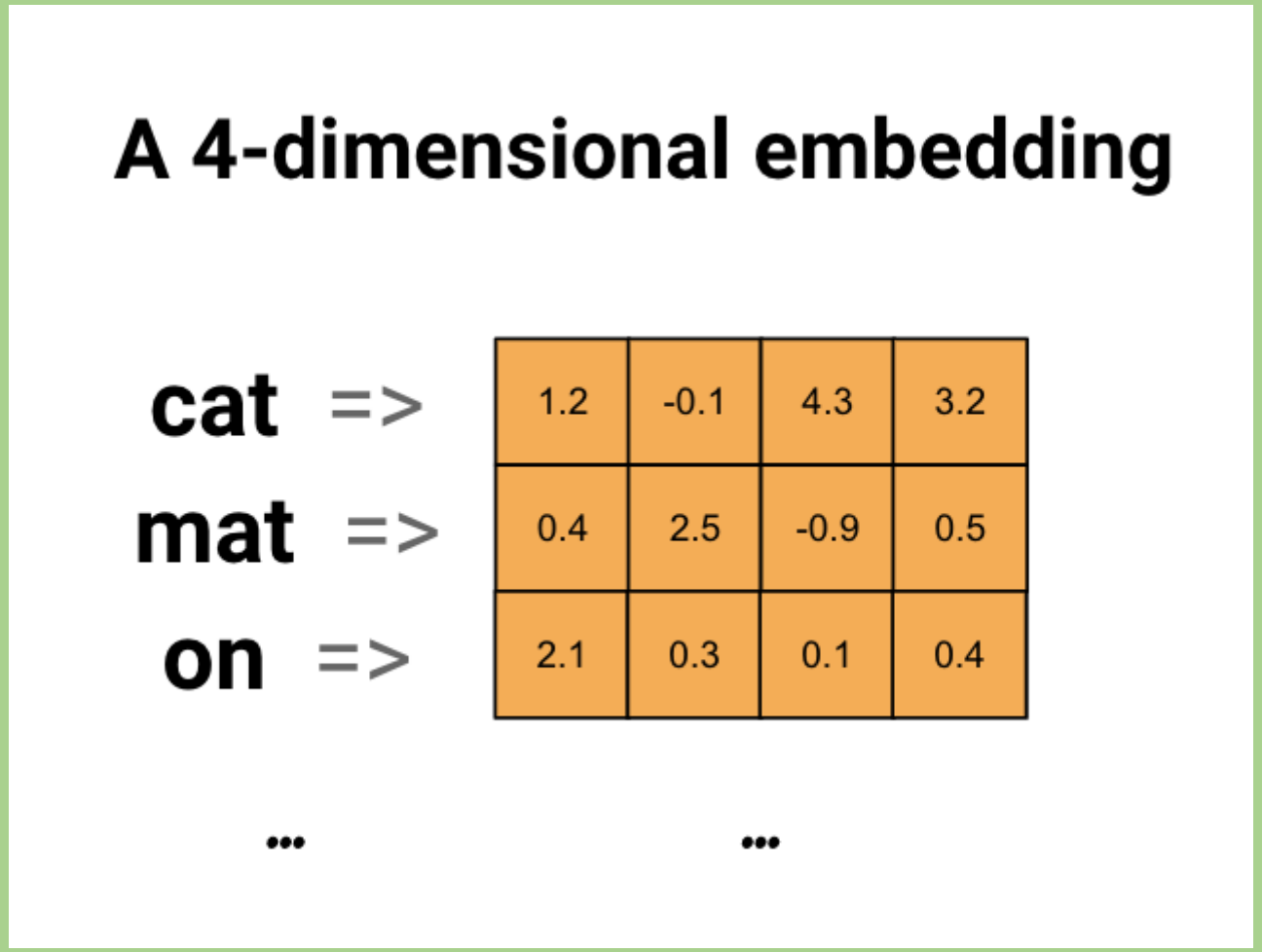
The Dog sat on my bed

	on	mat	sat	the	cat	my	dog	bed
0	0.0	0.115525	0.0	0.0	0.115525	0.000000	0.000000	0.000000
1	0.0	0.000000	0.0	0.0	0.000000	0.115525	0.115525	0.115525

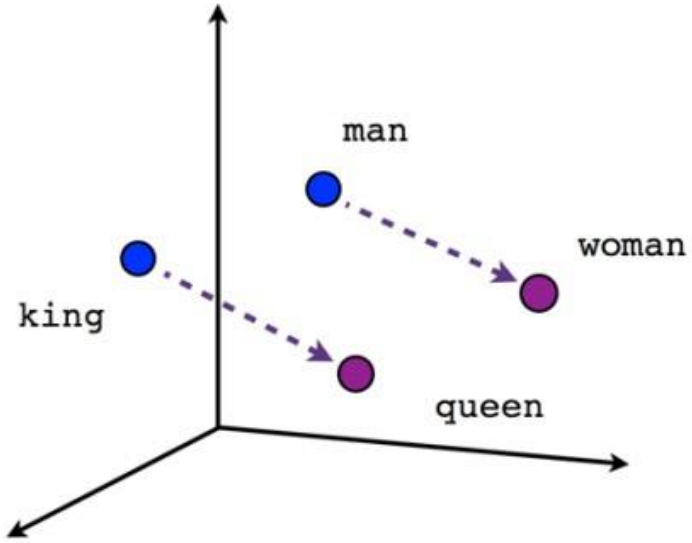
Other NLP techniques



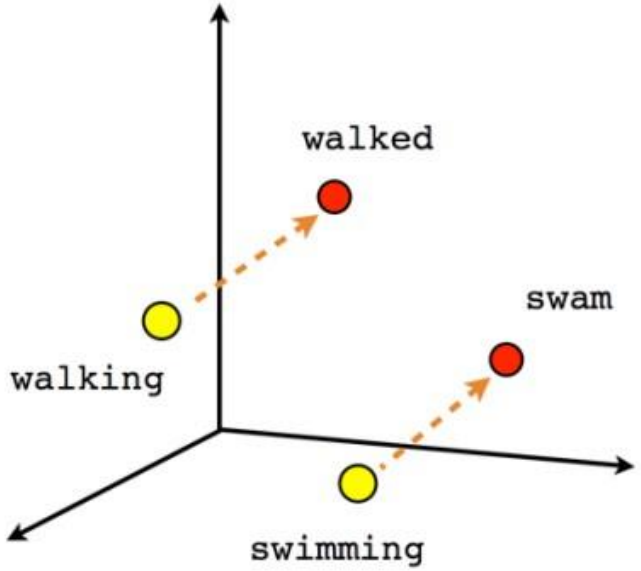
The Cat sat on the Mat



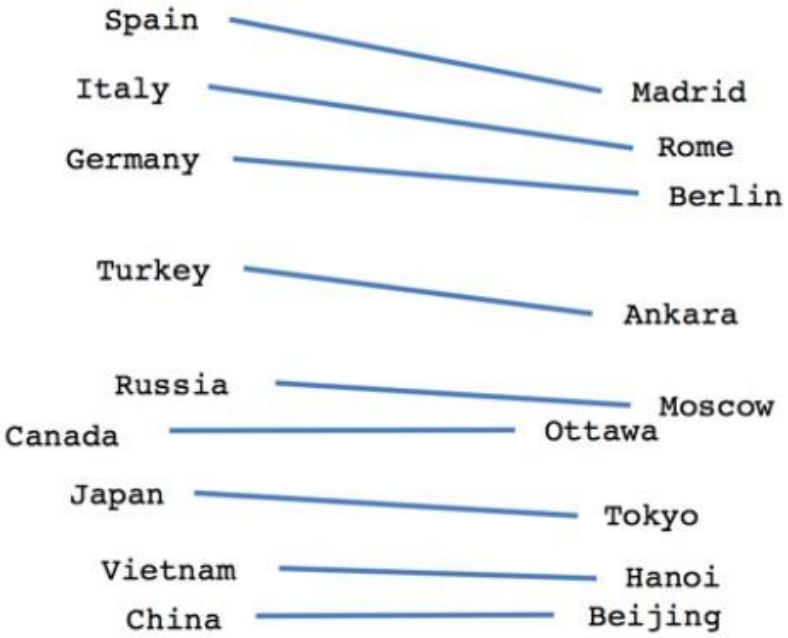
Word Embeddings



Male-Female



Verb tense

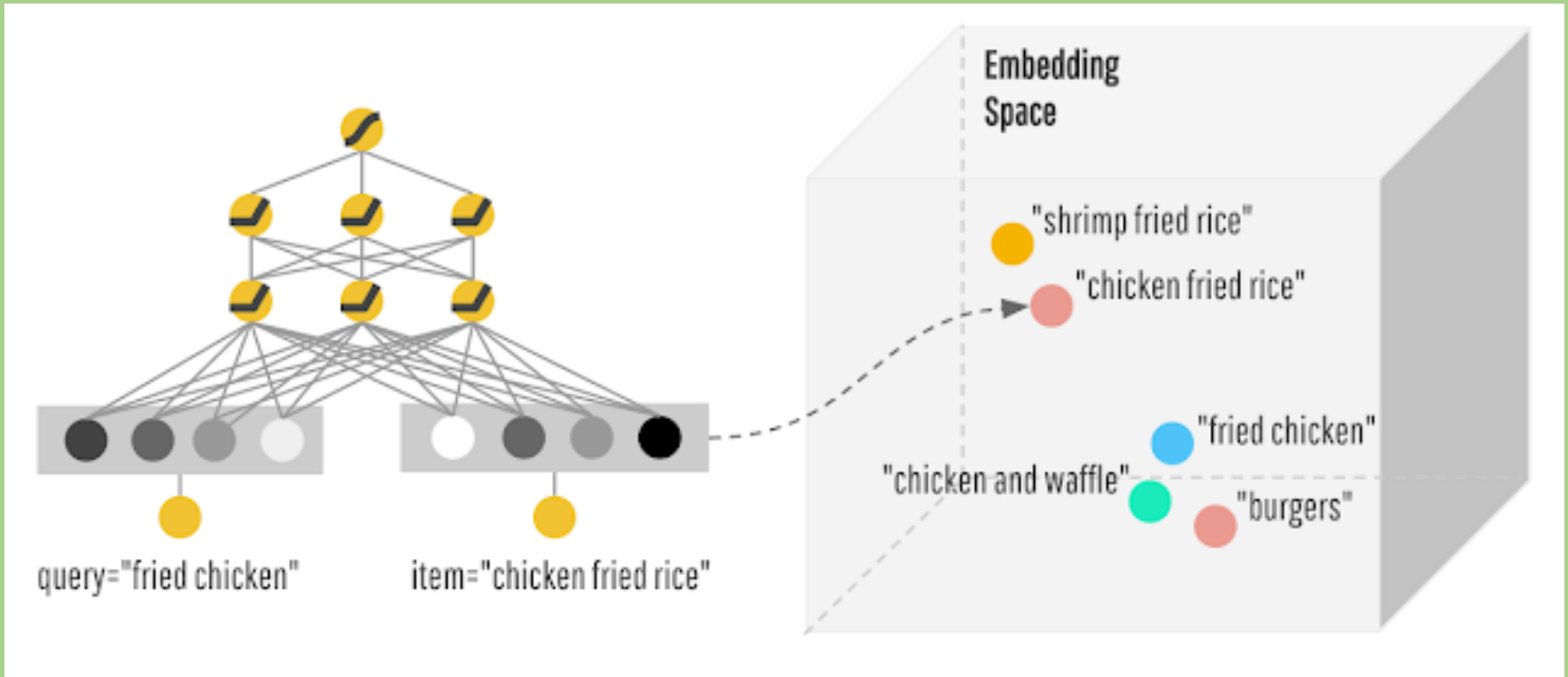


Country-Capital

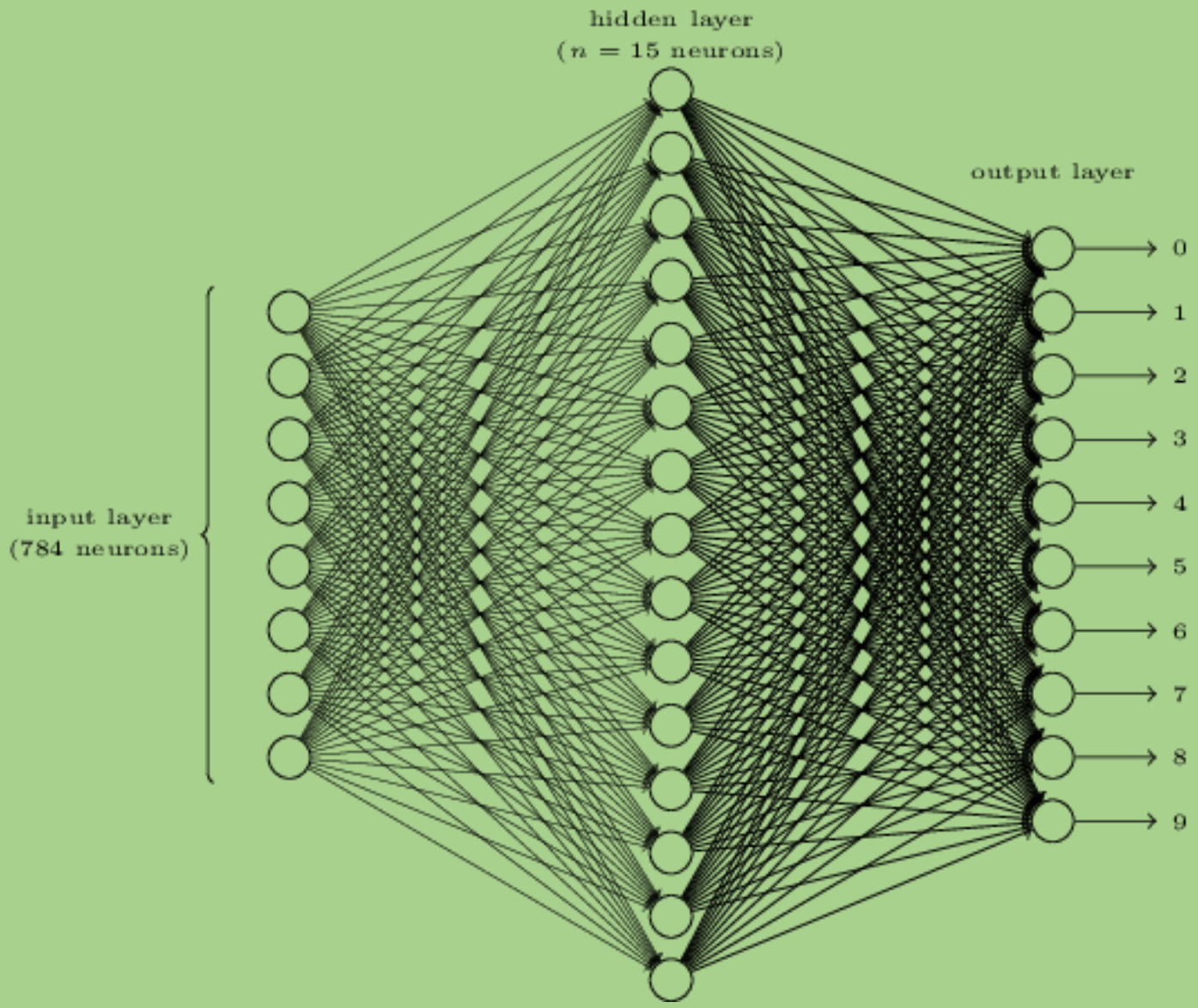
Microsoft

MSA
Microsoft Student Accelerator

Word Embeddings



Let’s build a ML Model

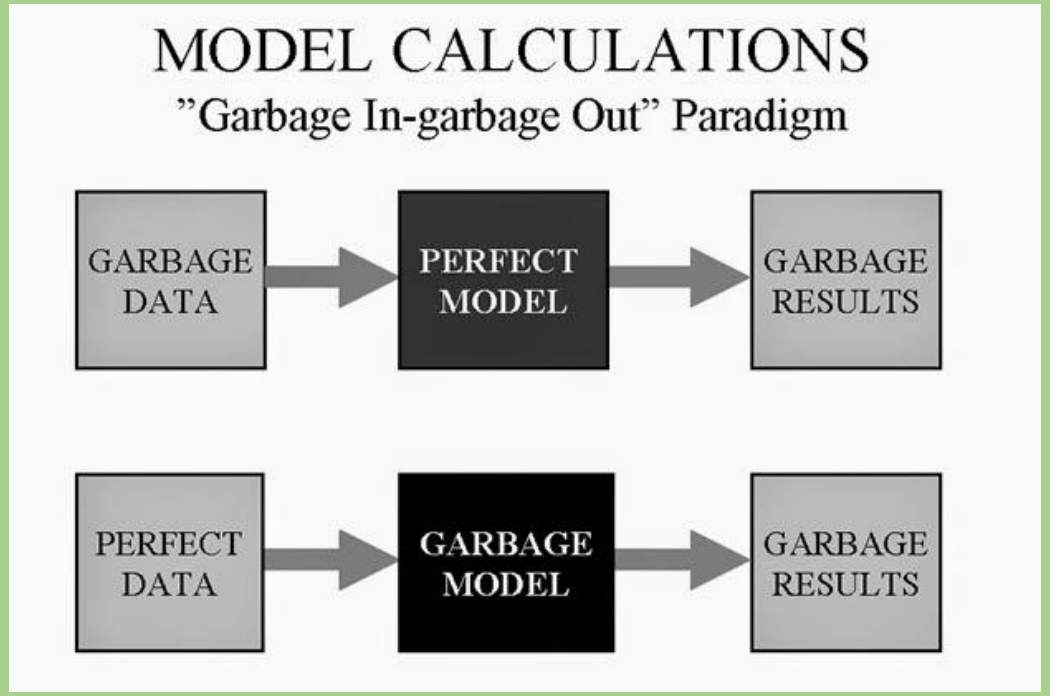


Microsoft

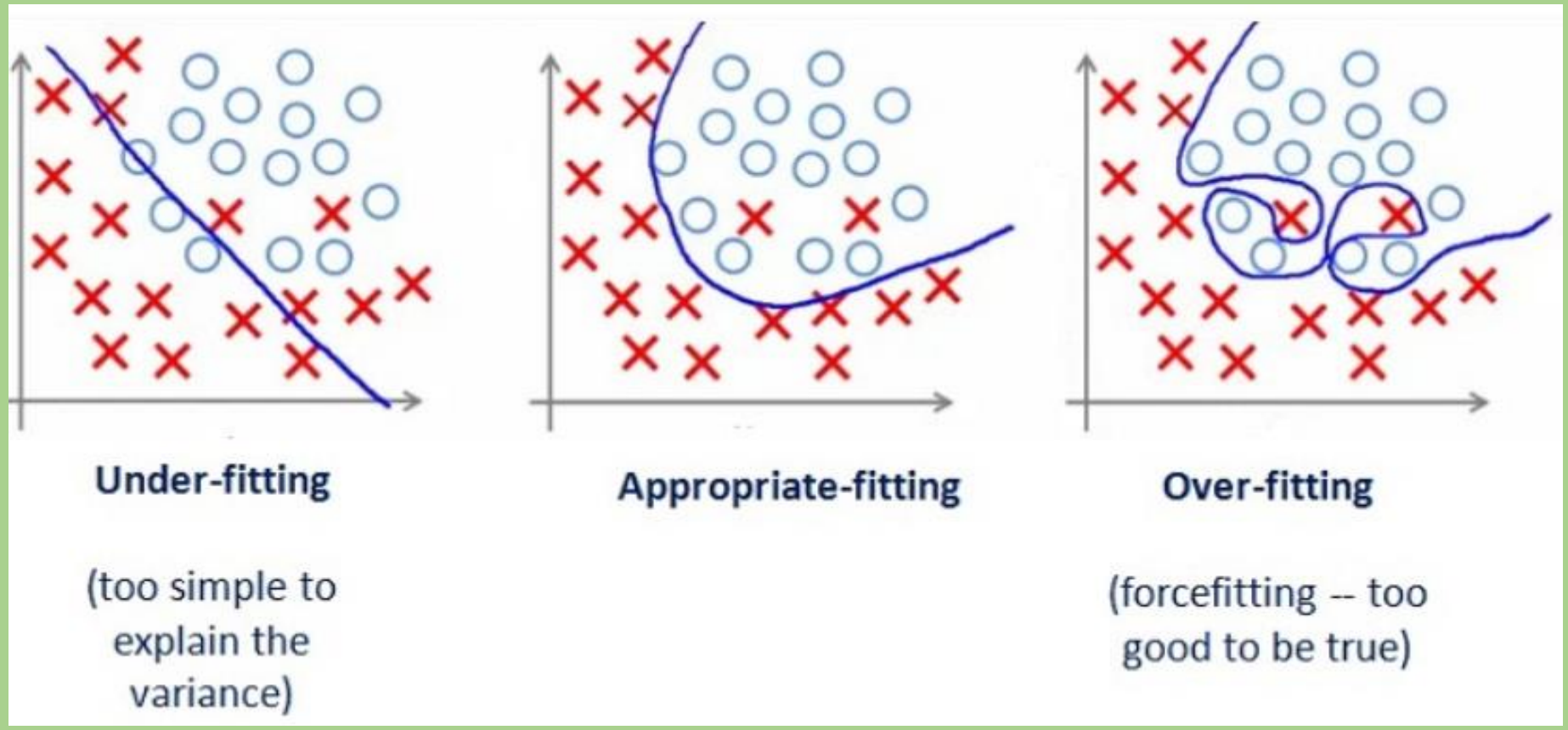
MSA
Microsoft Student Accelerator

Training testing and validation

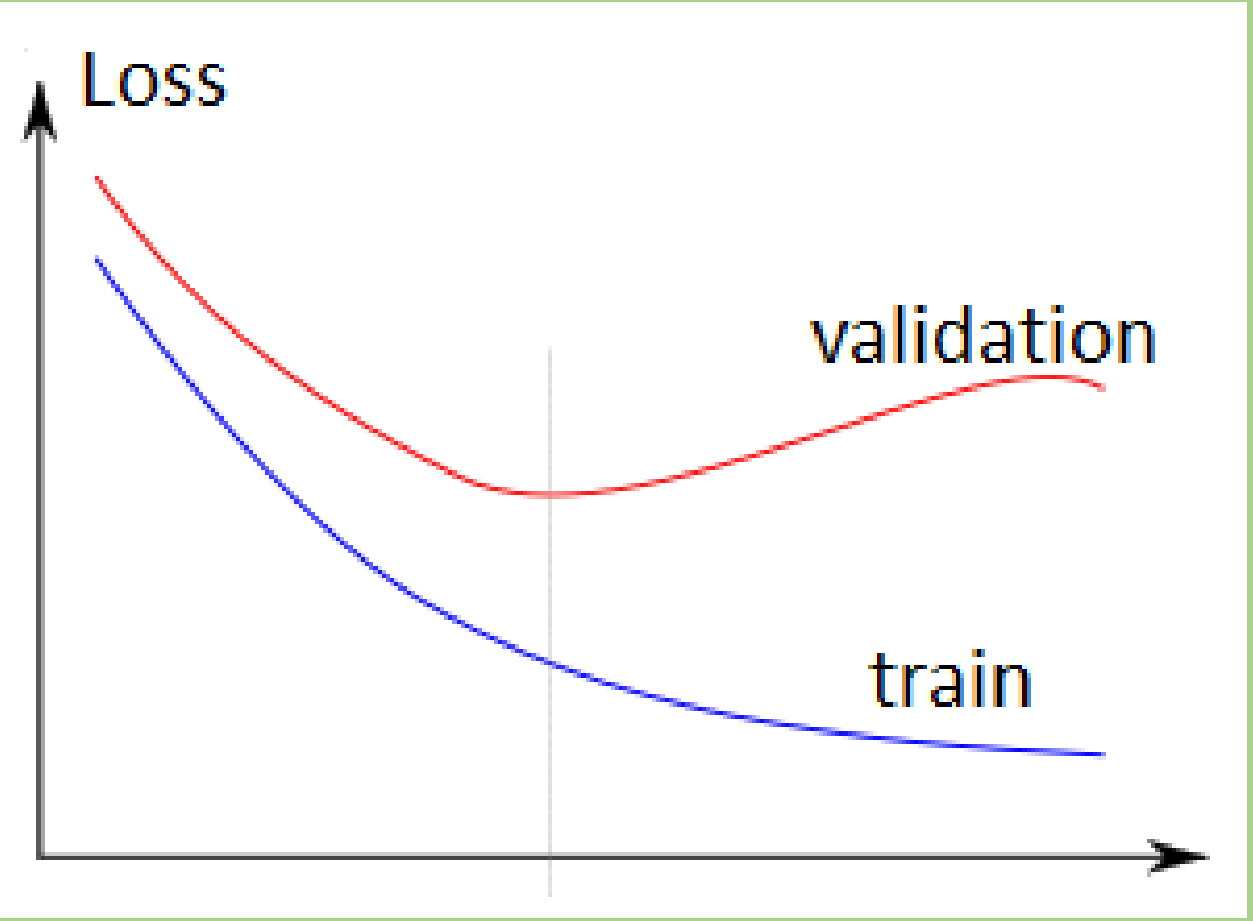
1. Training data
   The data your algorithm will ‘learn’ from.
   garbage in = garbage out.
2. Validation data
   Data used to tune hyperparameters.
   Results will be used to tweak the model.
3. Testing data
   Used to validate final accuracy.
   Should not be used for hyperparameter tuning.



Overfitting/Underfitting



Overfitting



Resources

- Youtube
  - http://3b1b.co/neural-networks
  - https://www.youtube.com/playlist?list=PLZbbT5o_s2xq7LwI2y8_QtvuXZedL6tQU
  - https://www.youtube.com/watch?v=JB8T_zN7ZC0
  - … + a LOT more

- MOOCs
  - fast.ai
  - deeplearning.ai
  - http://cs231n.stanford.edu