AI in Built Environment DCP4300

Lec07-08: Deep Learning

Part A

Dr. Chaofeng Wang
Jianhao Gao (TA)
University of Florida
College of Design Construction and Planning

1

Artificial Intelligence Vs. Machine Learning Vs. Deep Learning?

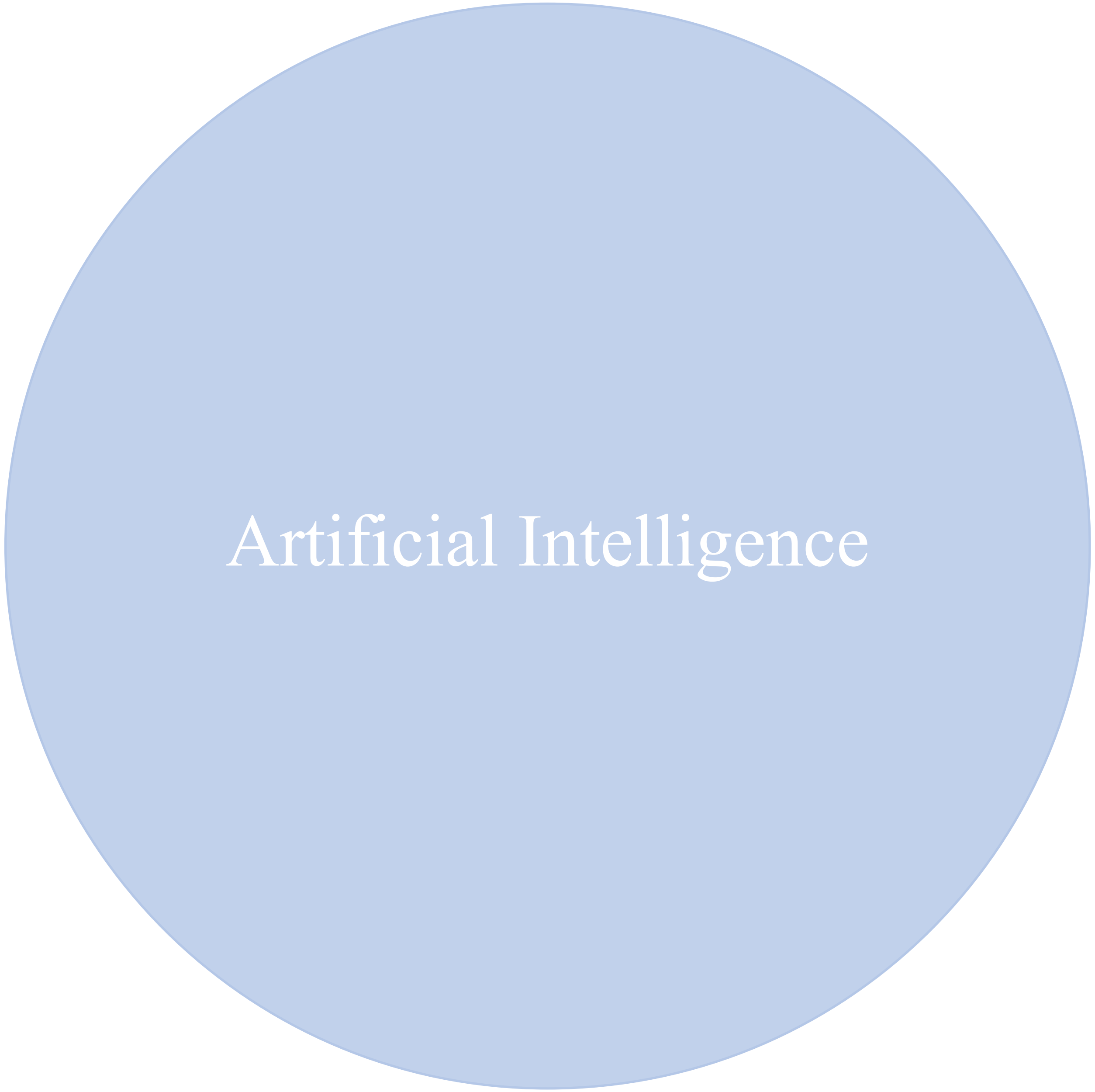
AI IN THE BUILT ENVIRONMENT
DCP4300

Week 2: Deep Learning

Part A

Dr. Chaofeng ‘Charles' Wang

University of Florida
College of Design Construction and Planning



AI: Techniques that enable machines to mimic human.

Robotics

Machine Learning

Neural Networks

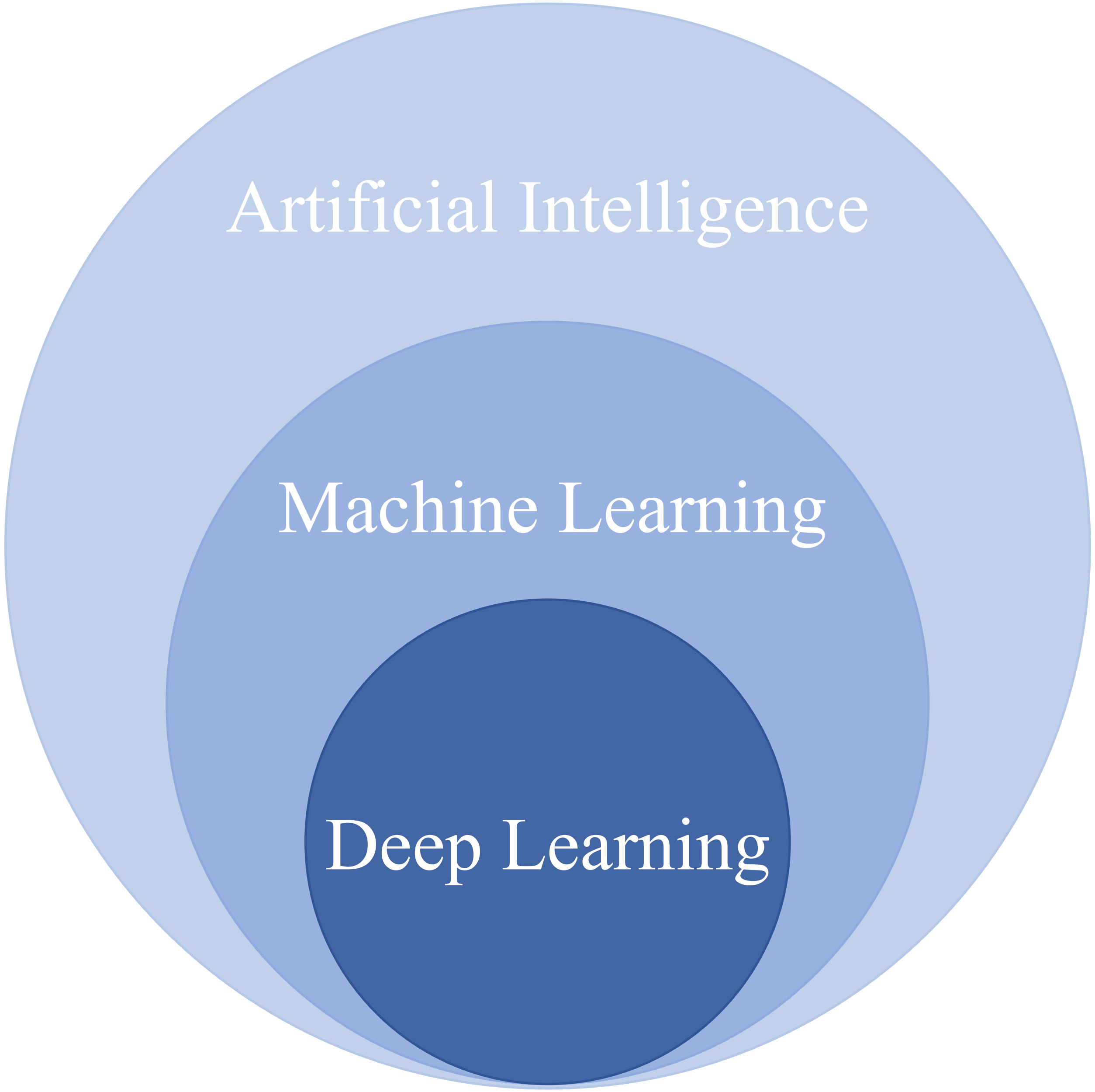
Computer Vision

Natural Language Processing

Expert System

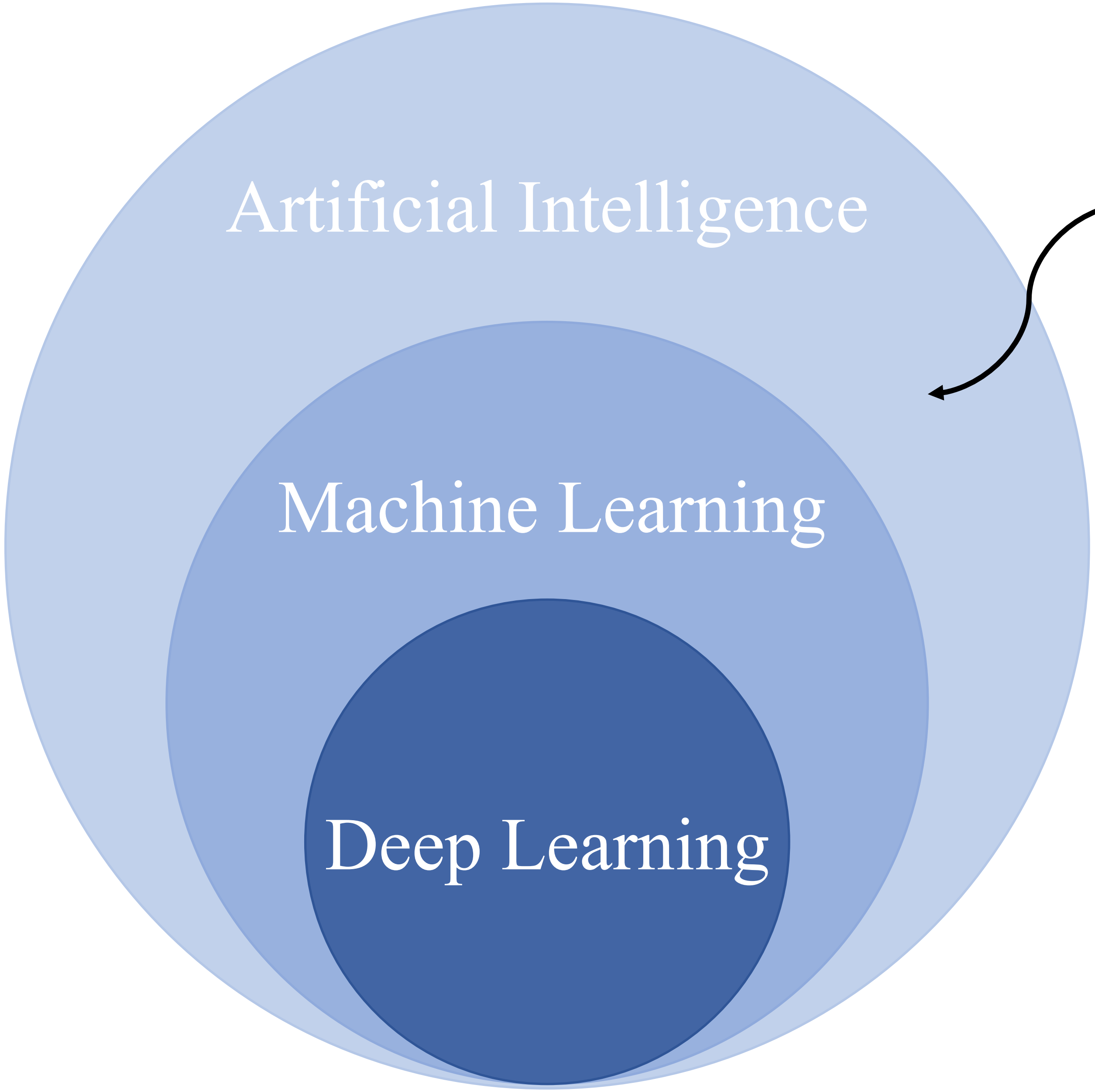
Fuzzy Logic

…

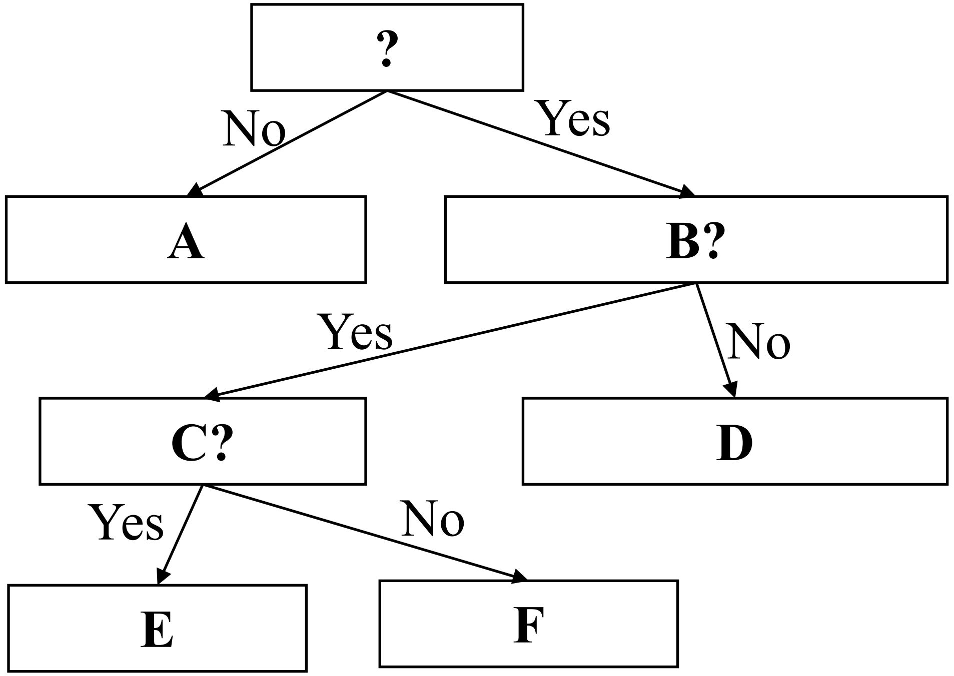


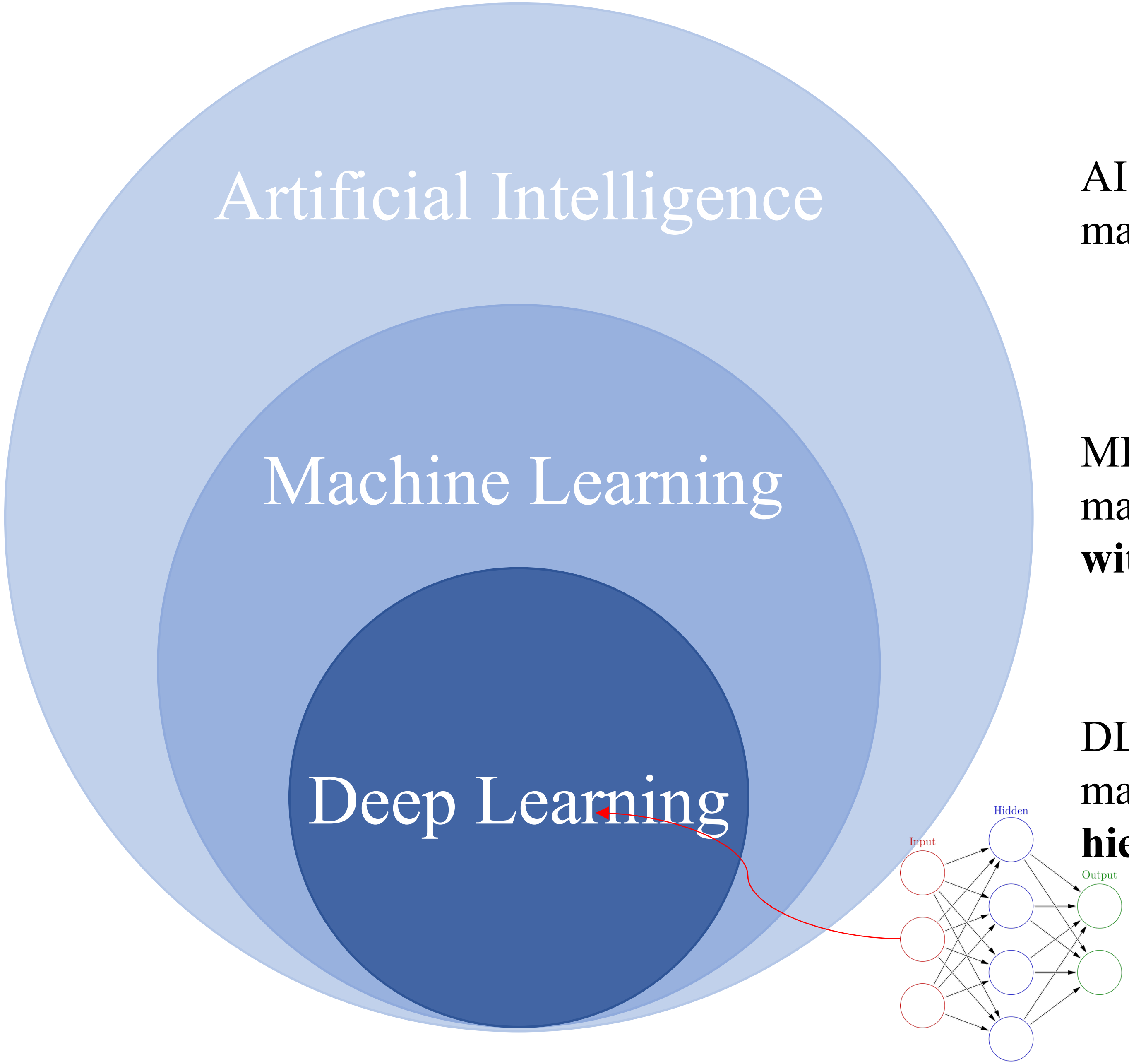
AI: Techniques that enable machines to mimic human.

ML: Techniques that enable machines to **learn from data**, **without being explicitly programmed**.



What is AI but not ML? Doesn’t learn. Explicitly programmed.
Example: Programmed expert systems.

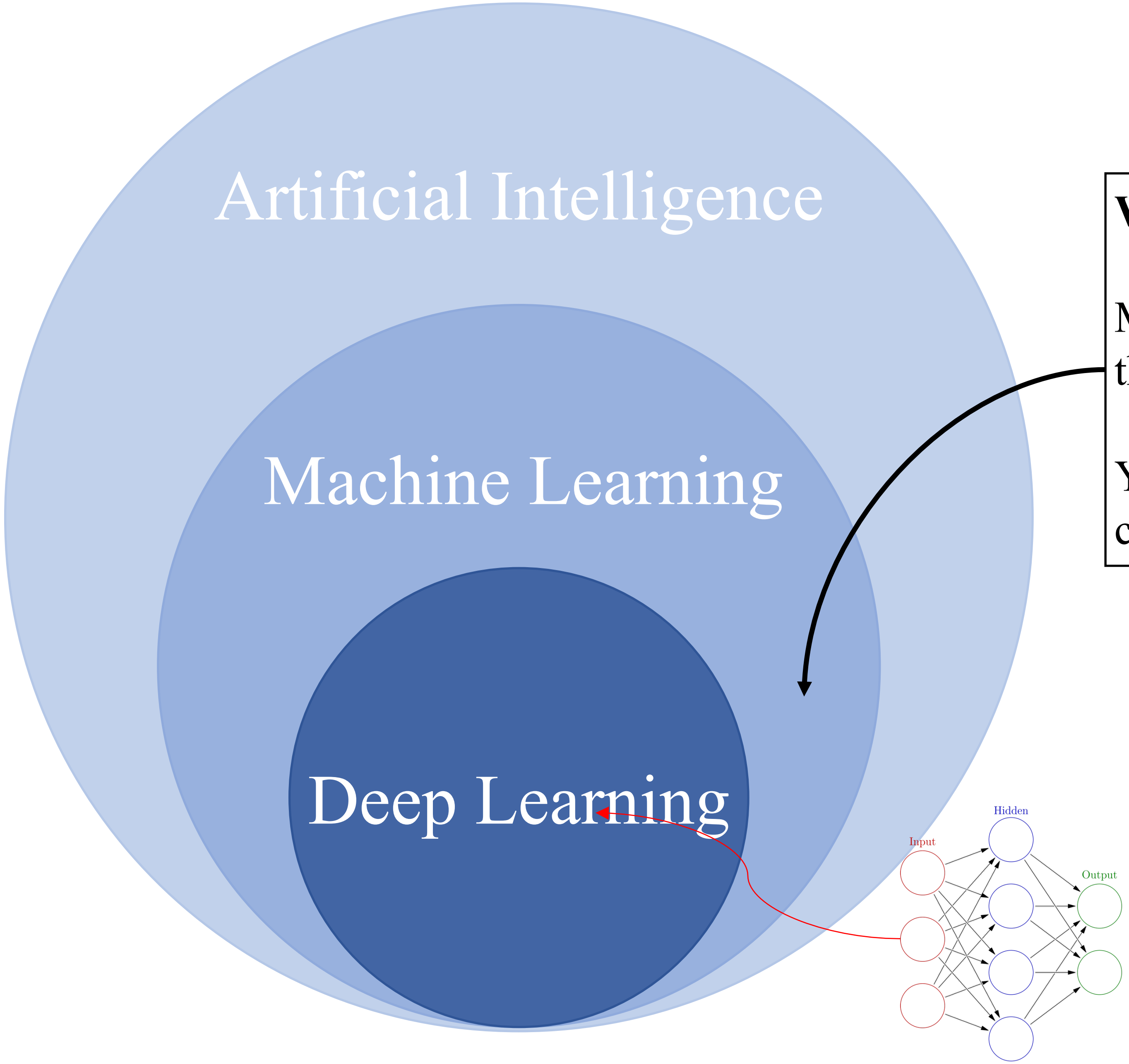




AI: Techniques that enable machines to mimic human.

ML: Techniques that enable machines to **learn from data**, **without being explicitly programmed**.

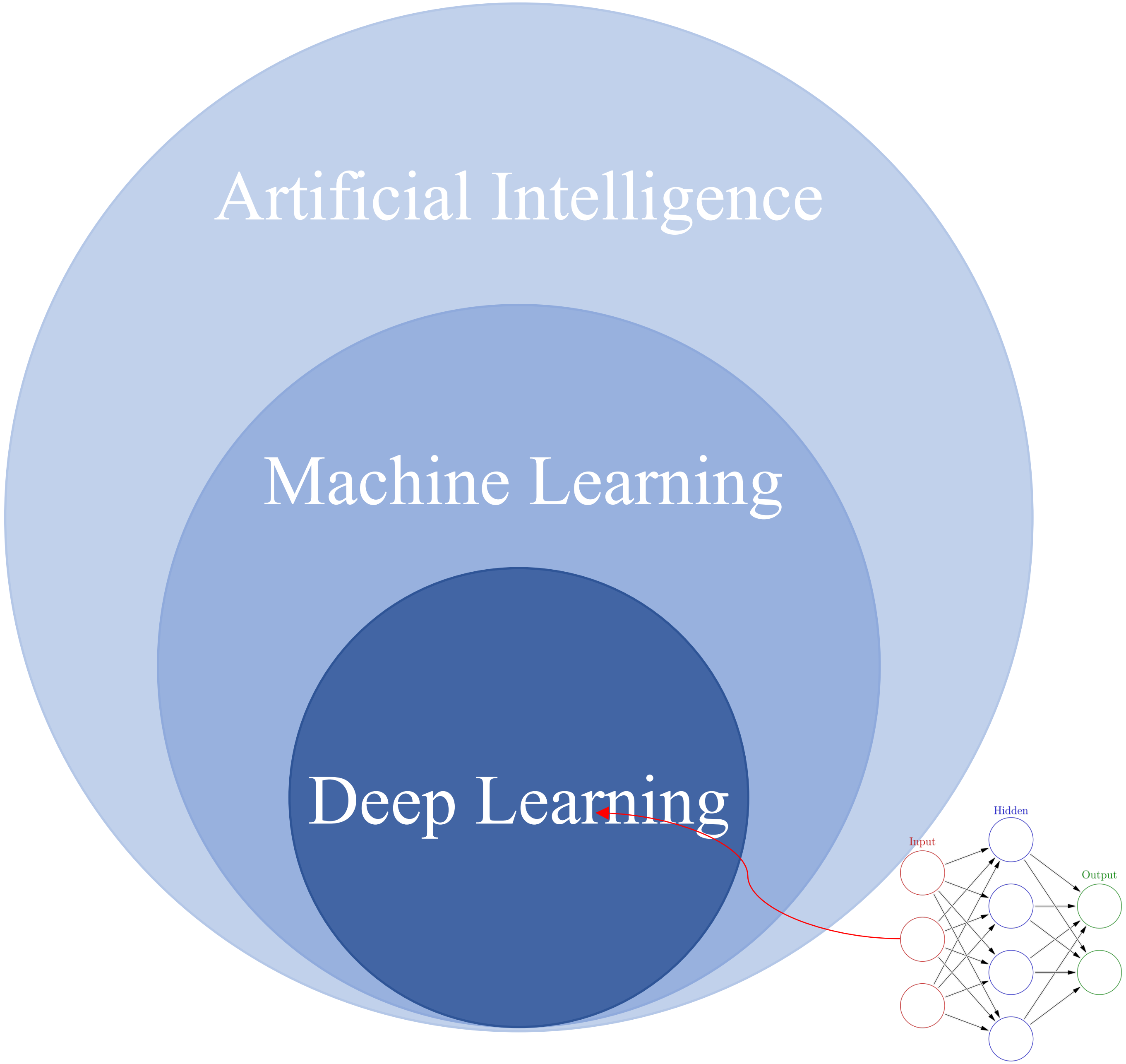
DL: Techniques that enable machines to learn from data, **hierarchically**, using **neural networks**.



What is ML but not DL?

ML that doesn't use (deep) neural networks that are capable of hierarchical learning.

You can call them **traditional ML** or conventional ML



How ‘deep’ should it be for DL?

Deep Learning ? Deep Neural Networks

2

Types of Machine Learning, what are the differences

Machine Learning:

Techniques that enable machines to
learn from data, **without being explicitly programmed**.

Types of Machine Learning:

Supervised Learning: Learn a function from **labeled** data.
- Classification
- Regression

Semi-supervised Learning

Unsupervised Learning: Learn the pattern from **unlabeled** data.
- Clustering
- Dimension reduction

Reinforcement Learning: Learn to react to an environment by **trial and error**.
- Decision making
- Robotics
- …

Fit a function: $f: X \rightarrow Y$

Supervised Learning: Learn a function from **labeled** data.

- Classification
- Regression

Algorithms:

Regressions
Decision trees
Support Vector Machines
Linear discriminant analysis
K-nearest neighbor algorithm
Multilayer perceptron
…

x1	x2	x3	x4	x5	y
0.21	0.20	0.65	0.87	0.29	0.22
0.83	0.47	0.14	0.77	0.43	0.63
0.42	0.31	0.41	0.43	0.11	0.92
0.83	0.49	0.52	0.01	0.94	0.17
0.99	0.05	0.47	0.72	0.01	0.60
0.31	0.31	0.74	0.41	0.93	0.13
0.29	0.03	0.32	0.16	0.24	0.35
0.91	0.91	0.24	0.23	0.51	0.23
0.47	0.04	0.17	0.77	0.34	0.08
0.10	0.10	0.73	0.82	0.32	0.23
0.09	0.66	0.10	0.98	0.21	0.66
0.00	0.35	0.38	0.18	0.89	0.02

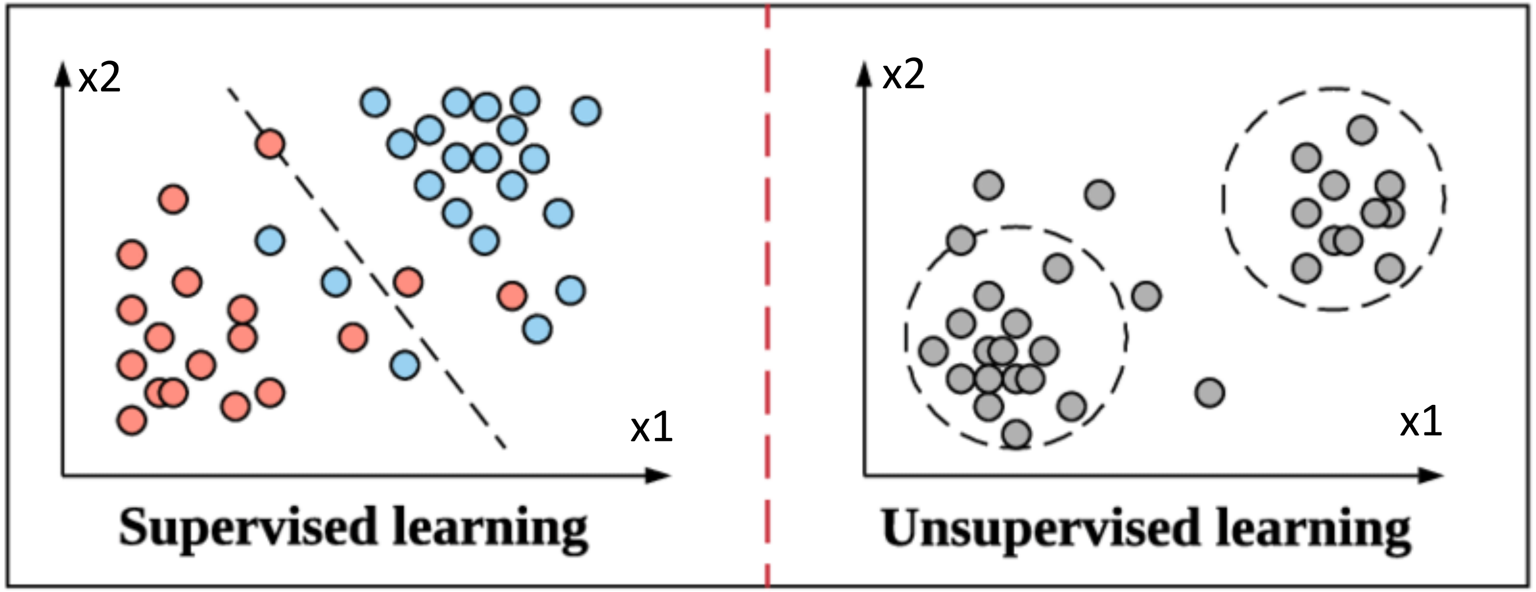
Basic frame of Supervised Learning:
Use data to set the parameters of a model to fit the labels.

Unsupervised Learning: Learn the patterns from **unlabeled** data.

- Clustering
- Dimension reduction

Algorithms:

K-means
Principal component analysis
Autoencoder
Generative adversarial networks



Classification

Trained with labeled data

Can predict the class name

Sky

Vegetation

Clustering

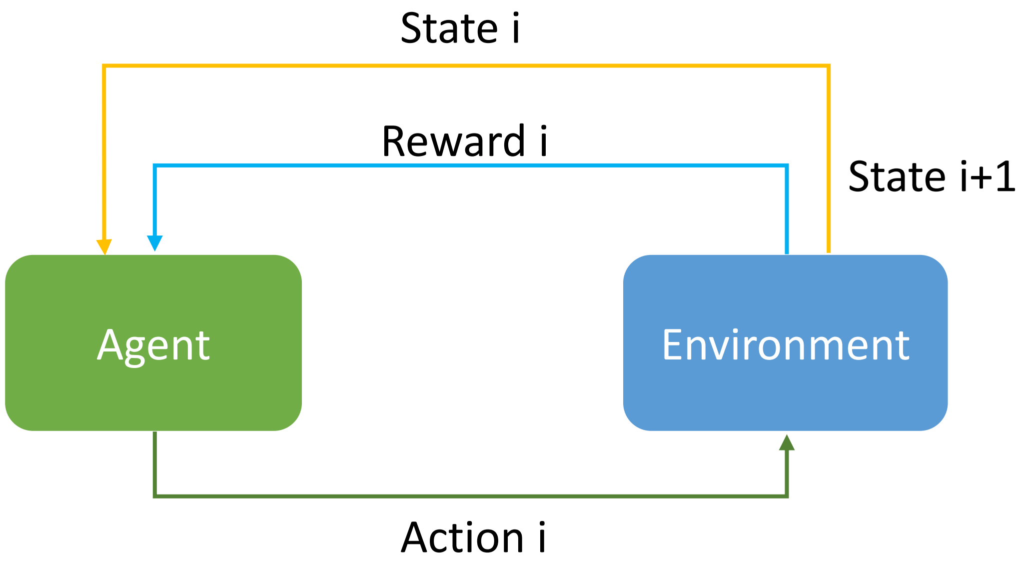
Trained with unlabeled data
Similar data points are grouped together

Group A

Group B

Reinforcement Learning:
Learn to react to an environment by **trial and error**.

- Decisions
- Robotics
- …



Algorithms:

Q-learning
SARSA
DQN
…

Reinforcement Learning:
Learn to react to an environment by **trial and error**.

- Decisions
- Robotics
- …

Algorithms:

Q-learning
SARSA
DQN
…

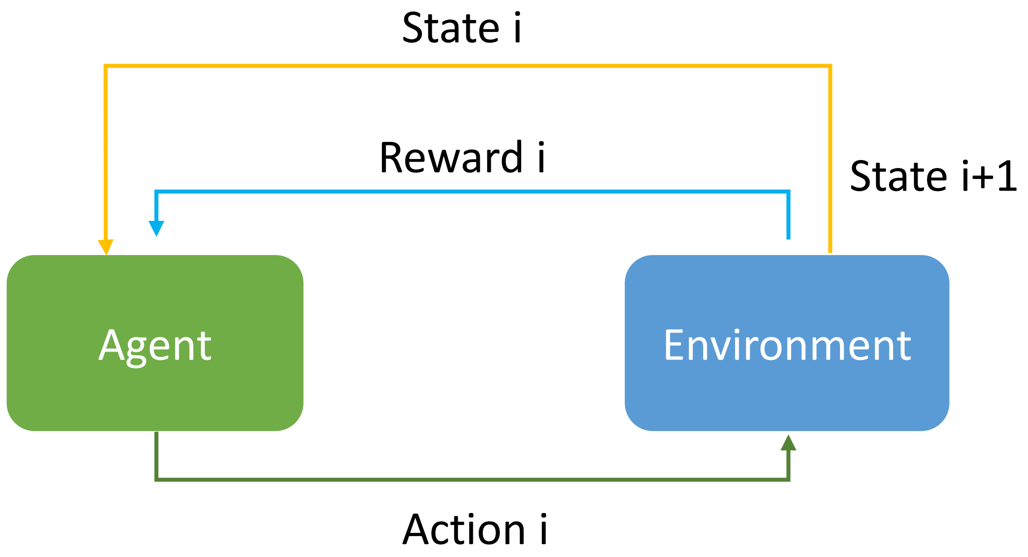
Supervised Learning Vs Reinforcement Learning

They are both trained to learn some functions: $f: X \to Y$

x1	x2	x3	x4	x5	y
0.21	0.20	0.65	0.87	0.29	0.22
0.83	0.47	0.14	0.77	0.43	0.63
0.42	0.31	0.41	0.43	0.11	0.92
0.83	0.49	0.52	0.01	0.94	0.17
0.99	0.05	0.47	0.72	0.01	0.60
0.31	0.31	0.74	0.41	0.93	0.13
0.29	0.03	0.32	0.16	0.24	0.35
0.91	0.91	0.24	0.23	0.51	0.23
0.47	0.04	0.17	0.77	0.34	0.08
0.10	0.10	0.73	0.82	0.32	0.23
0.09	0.66	0.10	0.98	0.21	0.66
0.00	0.35	0.38	0.18	0.89	0.02

X is a space of [x1,x2,x3,x4,x5]

Y is a space of [y]



X is a space of [state]

Y is a space of [action]

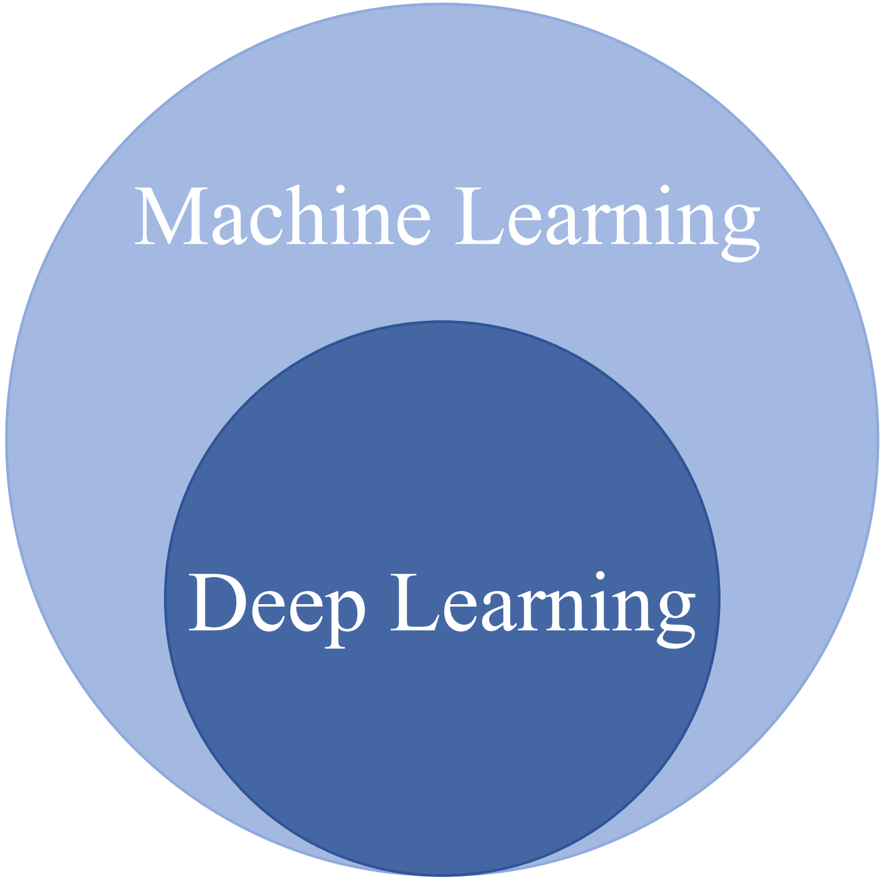
The differences:

Training data is labeled:
pairs of ([x1,x2,x3,x4,x5], [y])

No training data.
The agent has to interact with environment
to generate the data on the run.
And the generated data is unlabeled.

3

Deep Learning

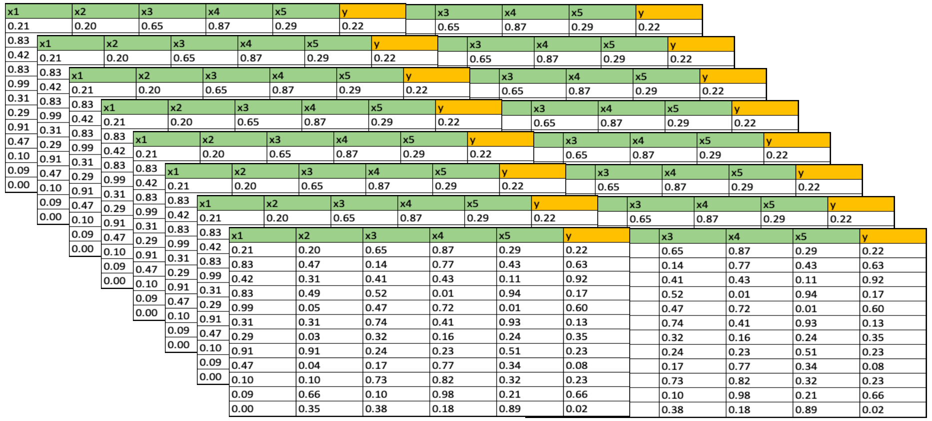


DL: Techniques that enable machines to to learn from data, **hierarchically**, using **neural networks**.

Why deep learning?

x1	x2	x3	x4	x5	y
0.21	0.20	0.65	0.87	0.29	0.22
0.83	0.47	0.14	0.77	0.43	0.63
0.42	0.31	0.41	0.43	0.11	0.92
0.83	0.49	0.52	0.01	0.94	0.17
0.99	0.05	0.47	0.72	0.01	0.60
0.31	0.31	0.74	0.41	0.93	0.13
0.29	0.03	0.32	0.16	0.24	0.35
0.91	0.91	0.24	0.23	0.51	0.23
0.47	0.04	0.17	0.77	0.34	0.08
0.10	0.10	0.73	0.82	0.32	0.23
0.09	0.66	0.10	0.98	0.21	0.66
0.00	0.35	0.38	0.18	0.89	0.02

Small data



Big data

Size

Why deep learning?

x1	x2	x3	x4	x5	y
0.21	0.20	0.65	0.87	0.29	0.22
0.83	0.47	0.14	0.77	0.43	0.63
0.42	0.31	0.41	0.43	0.11	0.92
0.83	0.49	0.52	0.01	0.94	0.17
0.99	0.05	0.47	0.72	0.01	0.60
0.31	0.31	0.74	0.41	0.93	0.13
0.29	0.03	0.32	0.16	0.24	0.35
0.91	0.91	0.24	0.23	0.51	0.23
0.47	0.04	0.17	0.77	0.34	0.08
0.10	0.10	0.73	0.82	0.32	0.23
0.09	0.66	0.10	0.98	0.21	0.66
0.00	0.35	0.38	0.18	0.89	0.02

5 columns

x1	x2	x3	x4	x5		y
0.21	0.20	0.65	0.87	0.29		0.22
0.83	0.47	0.14	0.77	0.43		0.63
0.42	0.31	0.41	0.43	0.11		0.92
0.83	0.49	0.52	0.01	0.94		0.17
0.99	0.05	0.47	0.72	0.01		0.60
0.31	0.31	0.74	0.41	0.93		0.13
0.29	0.03	0.32	0.16	0.24		0.35
0.91	0.91	0.24	0.23	0.51		0.23
0.47	0.04	0.17	0.77	0.34		0.08
0.10	0.10	0.73	0.82	0.32		0.23
0.09	0.66	0.10	0.98	0.21		0.66
0.00	0.35	0.38	0.18	0.89		0.02

>thousands of columns

Dimension

Why deep learning?

128x128 pixels

x1	x2	x3	x4	x5	y
0.21	0.20	0.65	0.87	0.29	0.22
0.83	0.47	0.14	0.77	0.43	0.63
0.42	0.31	0.41	0.43	0.11	0.92
0.83	0.49	0.52	0.01	0.94	0.17
0.99	0.05	0.47	0.72	0.01	0.60
0.31	0.31	0.74	0.41	0.93	0.13
0.29	0.03	0.32	0.16	0.24	0.35
0.91	0.91	0.24	0.23	0.51	0.23
0.47	0.04	0.17	0.77	0.34	0.08
0.10	0.10	0.73	0.82	0.32	0.23
0.09	0.66	0.10	0.98	0.21	0.66
0.00	0.35	0.38	0.18	0.89	0.02

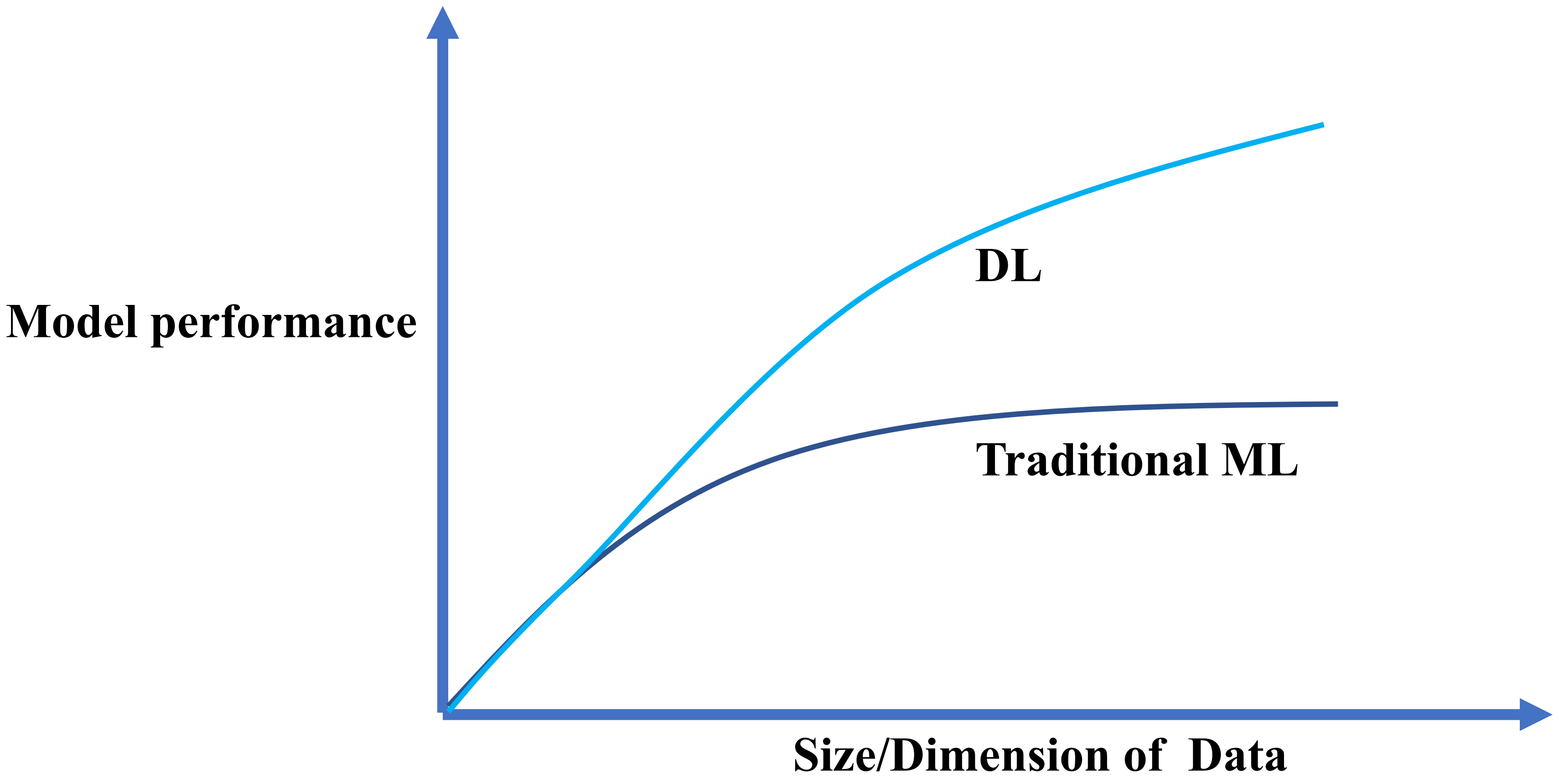
5 columns

x1	x2	x3	x4	x5		y
0.21	0.20	0.65	0.87	0.29		0.22
0.83	0.47	0.14	0.77	0.43		0.63
0.42	0.31	0.41	0.43	0.11		0.92
0.83	0.49	0.52	0.01	0.94		0.17
0.99	0.05	0.47	0.72	0.01		0.60
0.31	0.31	0.74	0.41	0.93		0.13
0.29	0.03	0.32	0.16	0.24		0.35
0.91	0.91	0.24	0.23	0.51		0.23
0.47	0.04	0.17	0.77	0.34		0.08
0.10	0.10	0.73	0.82	0.32		0.23
0.09	0.66	0.10	0.98	0.21		0.66
0.00	0.35	0.38	0.18	0.89		0.02

>thousands of columns

Dimension

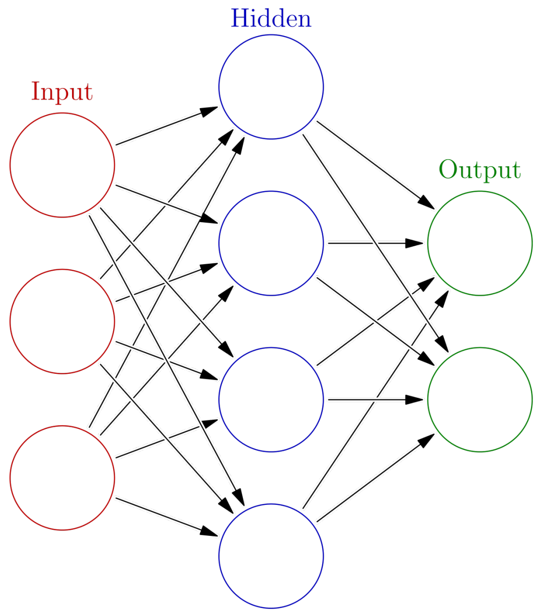
Why deep learning?



4

Neural networks

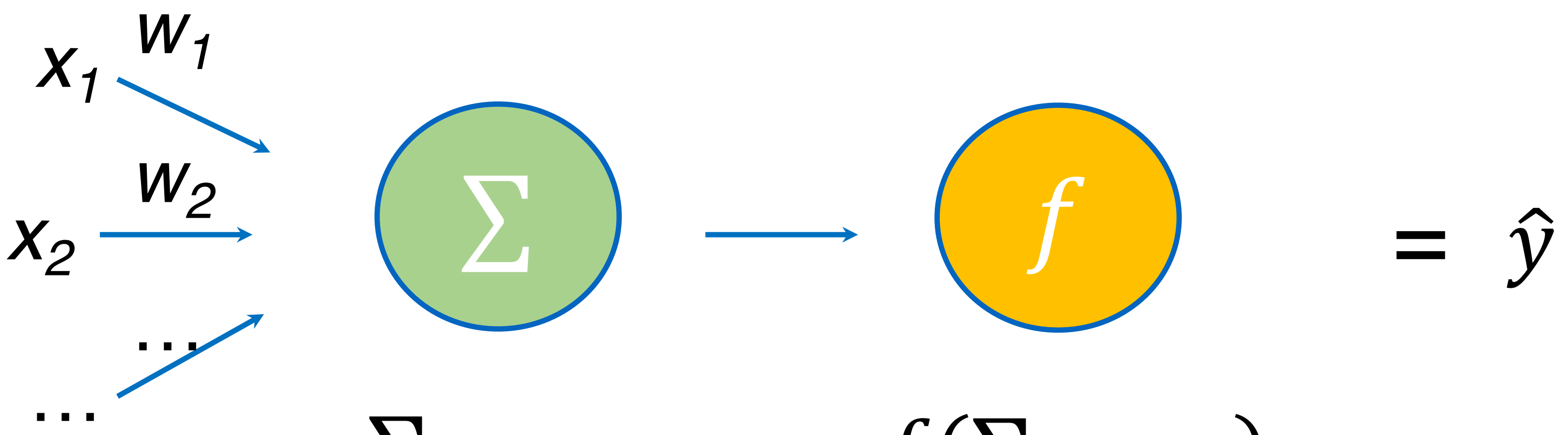
Deep learning ≈ (Deep) neural networks



Nuts and Bolts of a neural network (multilayer perceptron)

Neuro (perceptron)
Activation
Architecture
Loss function
Learning rate
Optimizer
Hyperparameter
…

Perceptron

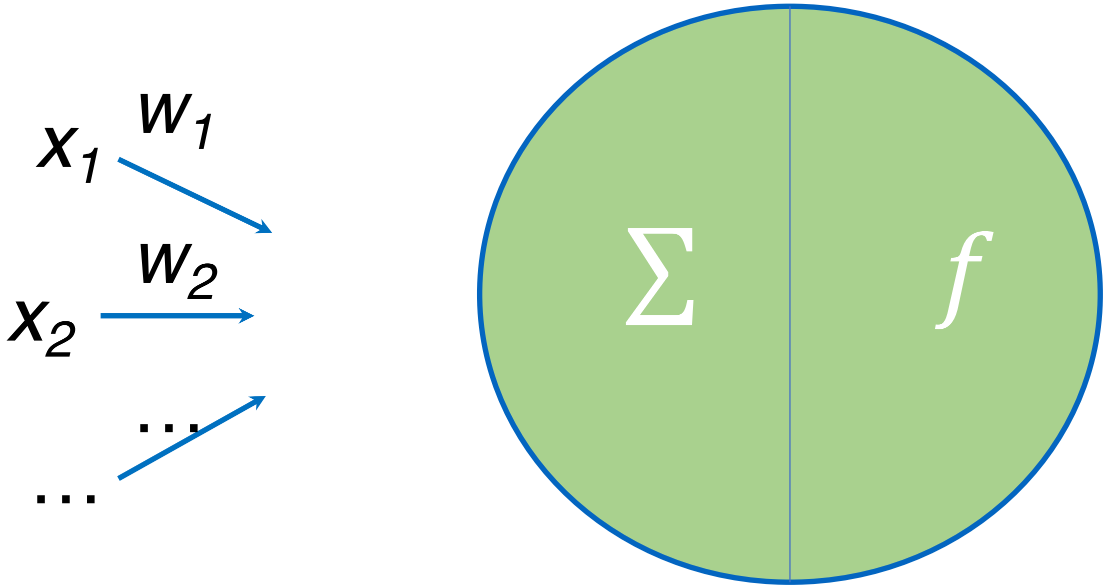


$\sum x_n w_n$ $f(\sum x_n w_n)$

f is the activation function

Sum then Activate

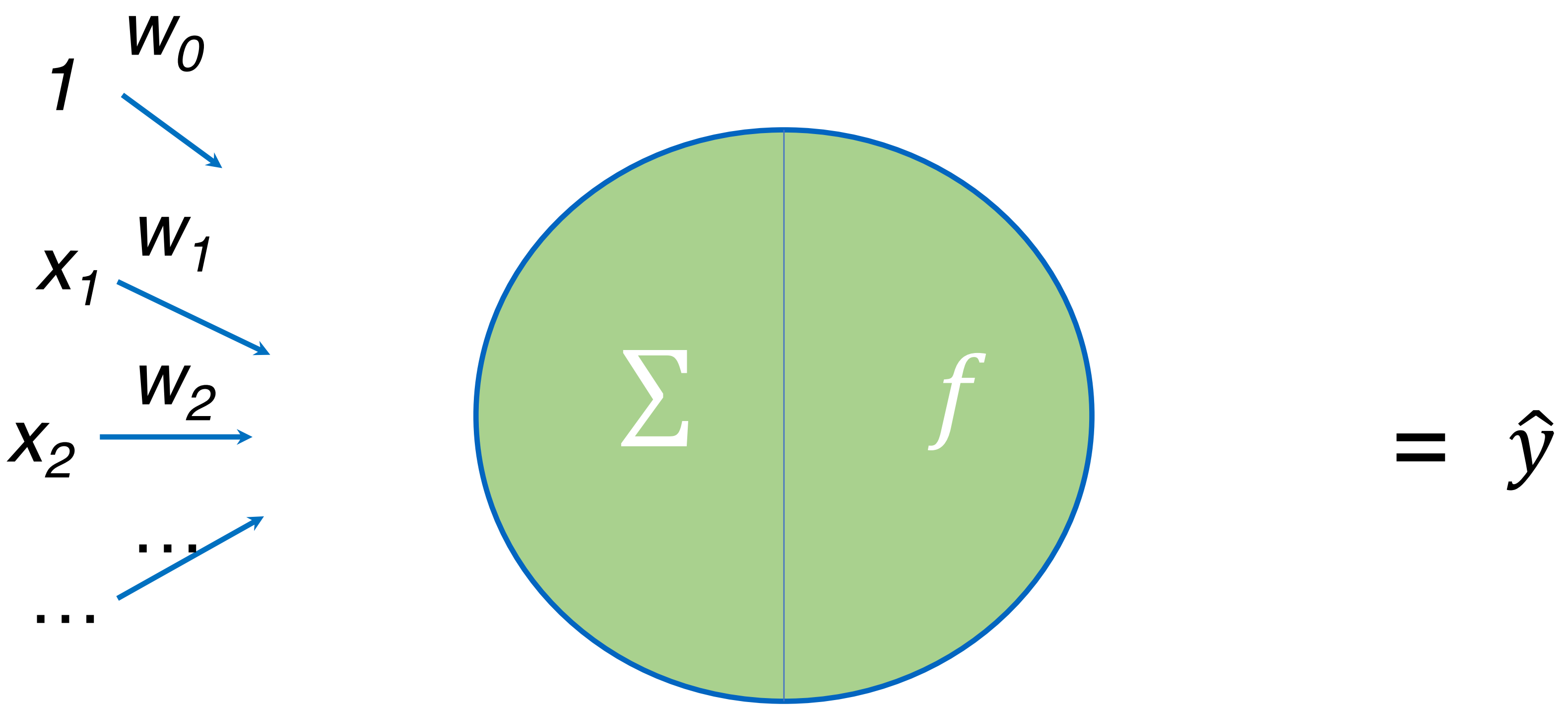
Perceptron



$f\left(\sum x_n w_n\right)$

$= \hat{y}$

Perceptron



$$f\,(w_0 + \sum x_n w_n)$$

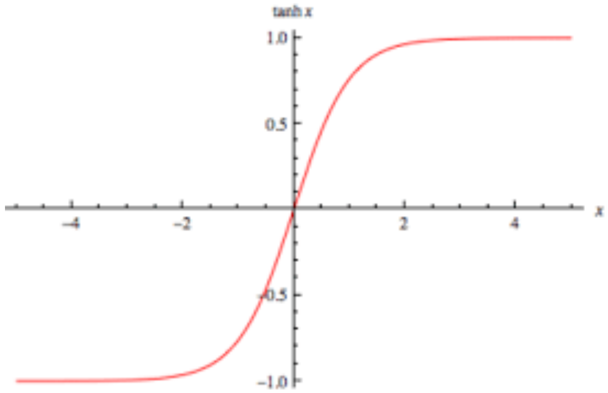
bias

Activation functions: why they are needed?

1. Limit the output

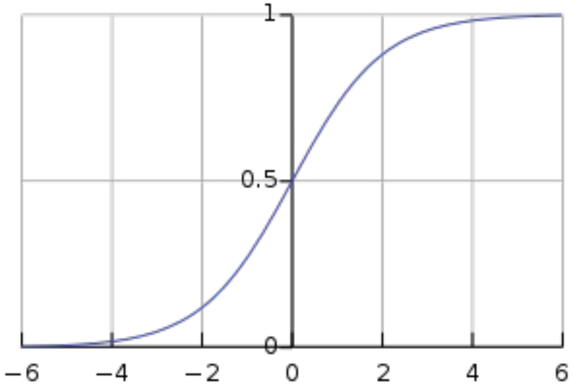
Hyperbolic tangent (tanh)

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



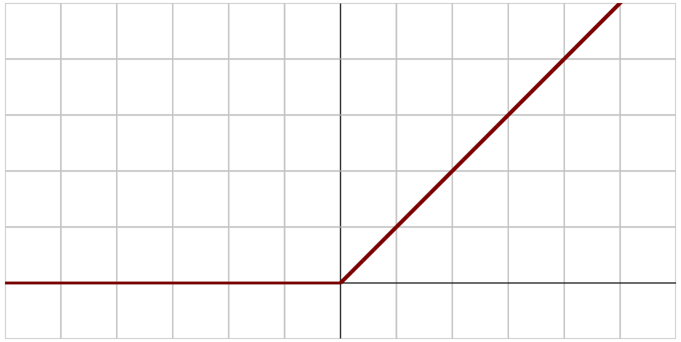
Sigmoid

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



Rectified linear unit (ReLU)

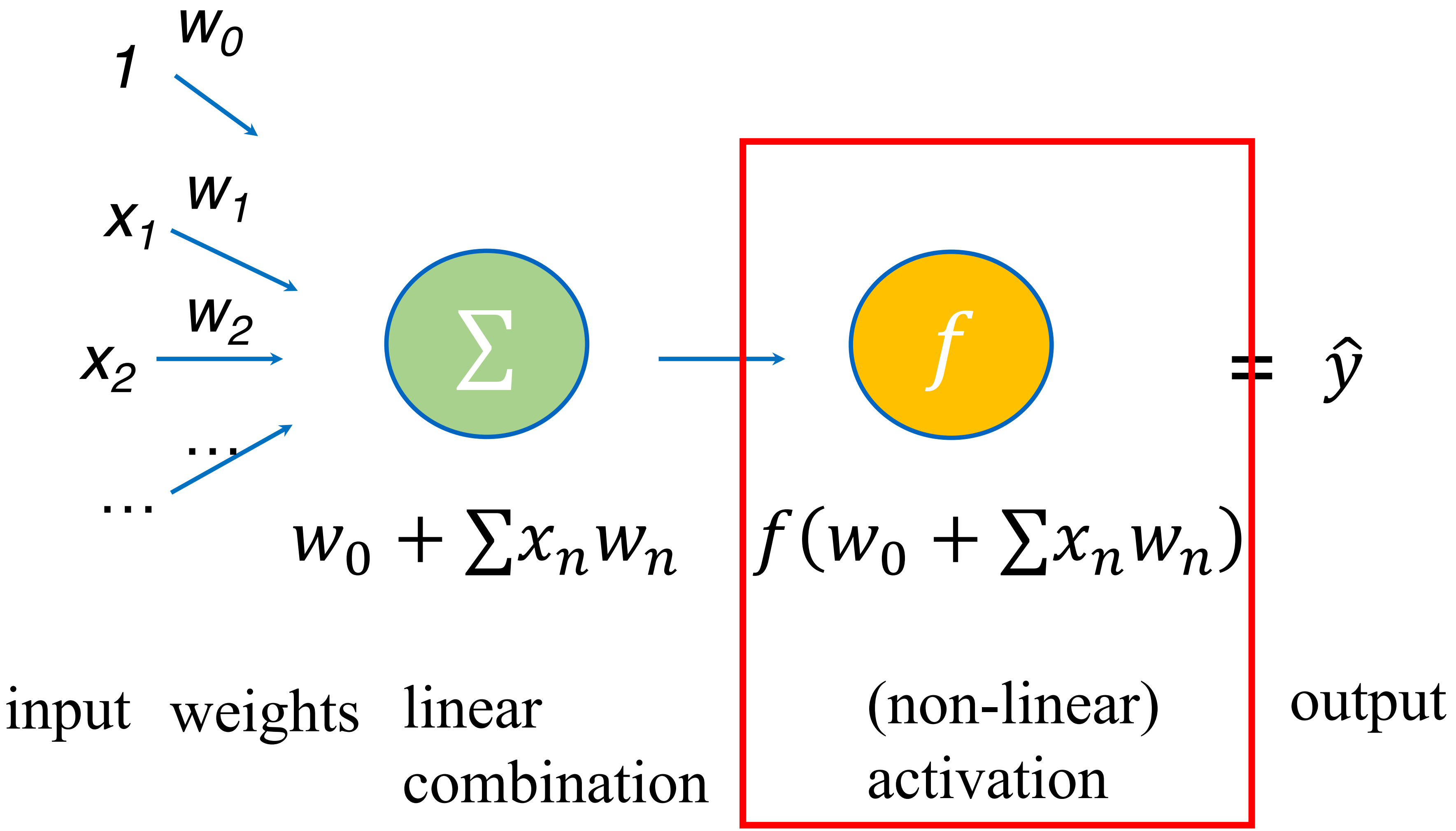
$$f(x) = x^+ = \max(0, x)$$



More: https://www.tensorflow.org/api_docs/python/tf/keras/activations

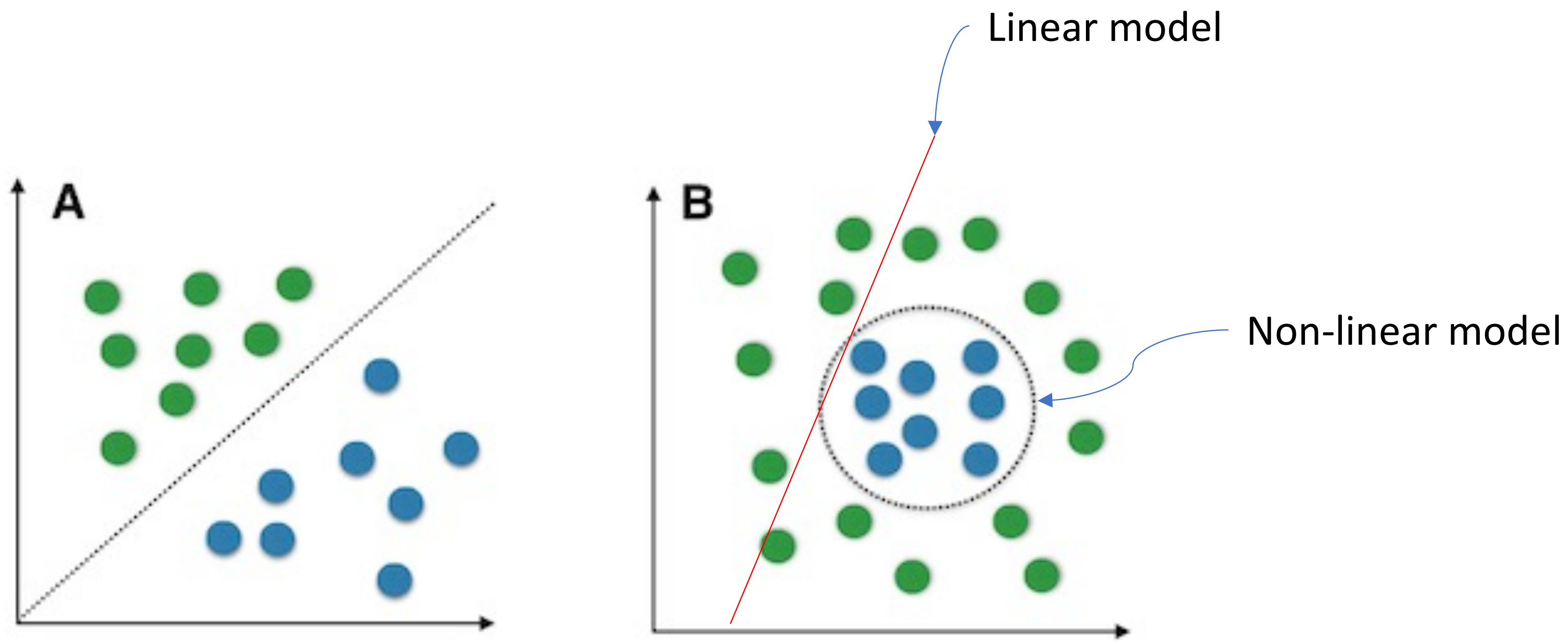
Activation functions: why they are needed?

2. Provide non-linearity



Activation functions: why they are needed?

2. Provide non-linearity



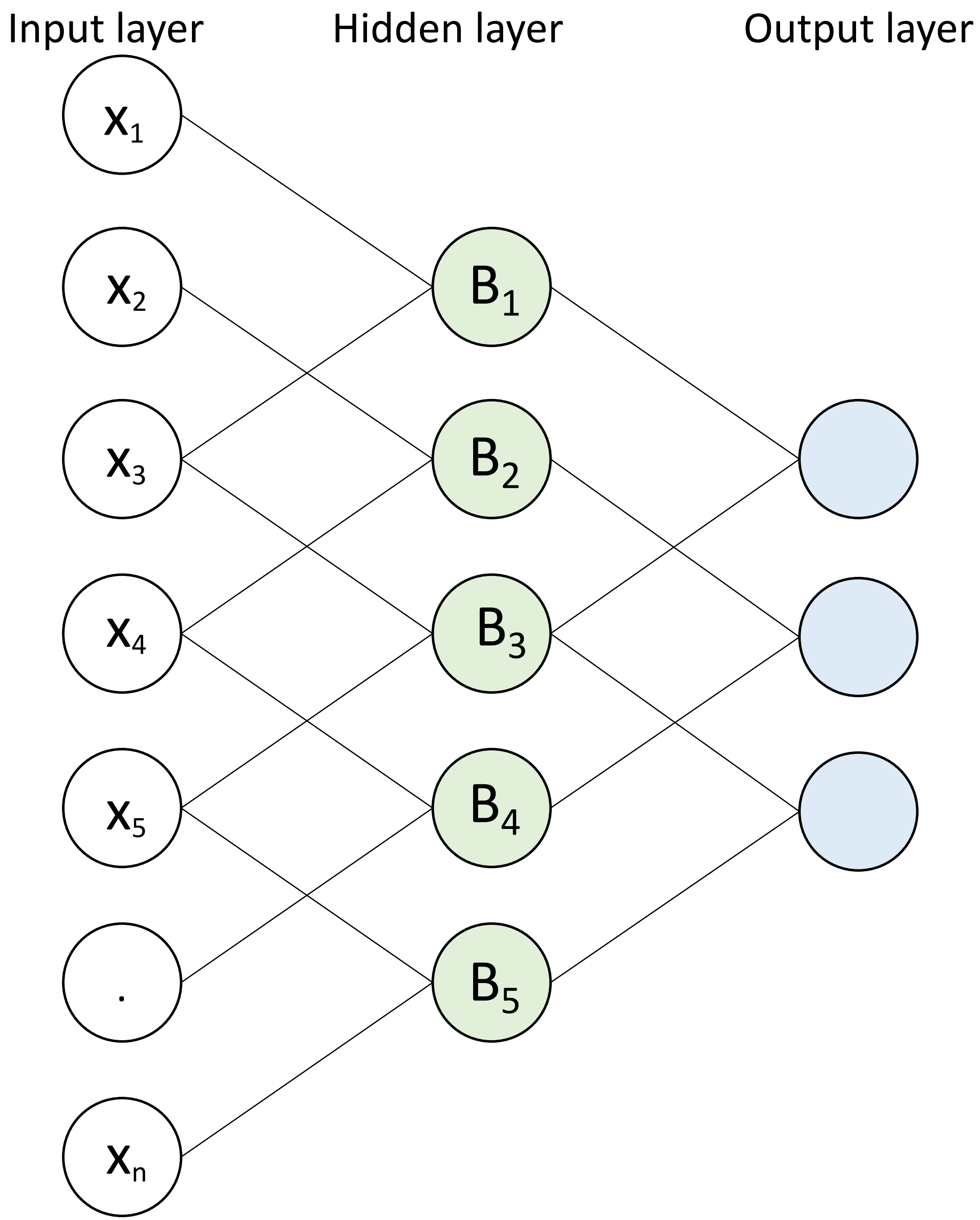
Linear data

Non-linear data (most data)

Neural network (Multilayer perceptron)

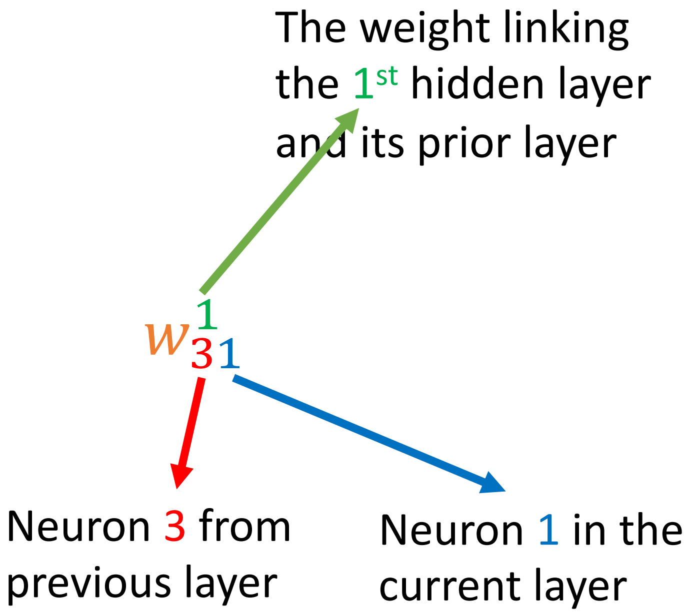
Architecture
The structure (topology and size) of the network

Hyperparameter
A parameter controlling the learning process.
e.g., values that describe the architecture,
learning rate

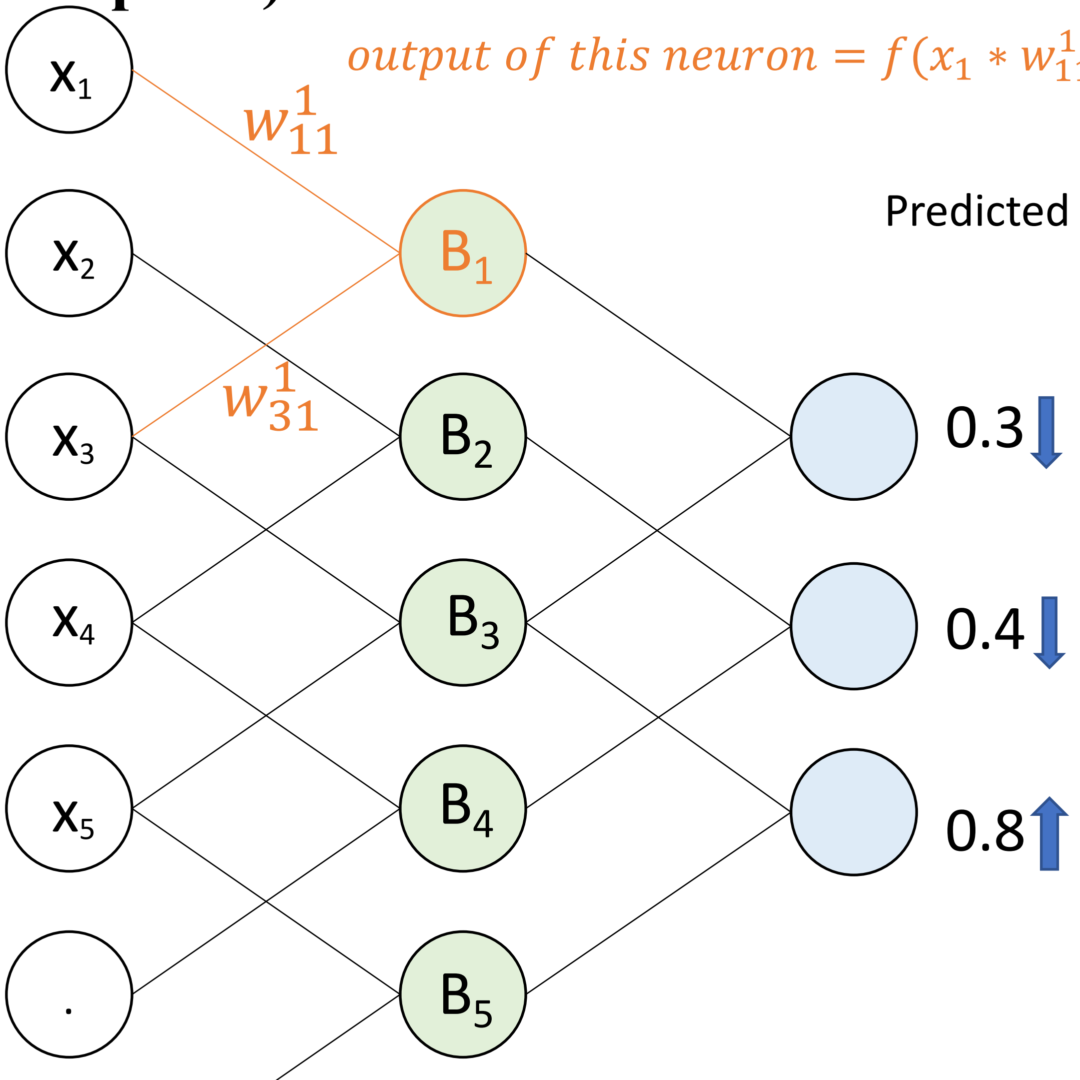


Neural network (Multilayer perceptron)

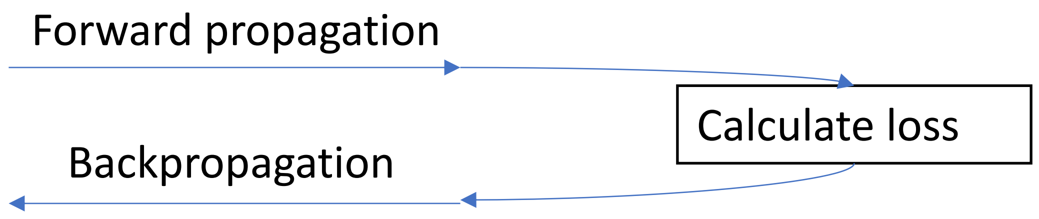
Weights



$$output\ of\ this\ neuron = f(x_1 * w_{11}^{1} + x_3 * w_{31}^{1} + B_1)$$



True Value (target)	Error
0	-0.3
0	-0.4
1	0.2



Loss function (Cost function / Objective function)

A function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event.

Example: Mean Squared Error

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\underbrace{Y_i}_{\text{Truth}} - \underbrace{\hat{Y}_i}_{\text{Prediction}})^2$$

More loss functions:
https://www.tensorflow.org/api_docs/python/tf/keras/losses

How to choose a loss function

For regression models (the output of the model is a real-valued quantity)

Mean Squared Error Loss $MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$

Mean Squared Logarithmic Error Loss $MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(Y_i) - \log(\hat{Y}_i))^2$

Mean Absolute Error Loss $MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i|$

How to choose a loss function

For classification models (the output of the model is a categorical quantity)

Binary classification

Binary Cross-Entropy $$BCE = \frac{1}{n}\sum_{i=1}^{n}\left(\boxed{-Y_i(class0)log\hat{Y}_i(class0)} - Y_i(class1)log\hat{Y}_i(class1)\right)$$

Entropy for class0

Multi-class classification

Multi-class Cross-Entropy $$BCE = \frac{1}{n}\sum_{i=1}^{n}\left(-Y_i(class0)log\hat{Y}_i(class0) - Y_i(class1)log\hat{Y}_i(class1) - \cdots\right)$$

More loss functions:
https://www.tensorflow.org/api_docs/python/tf/keras/losses
Choose a loss function:
https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/

Neural network (Multilayer perceptron)

Weights

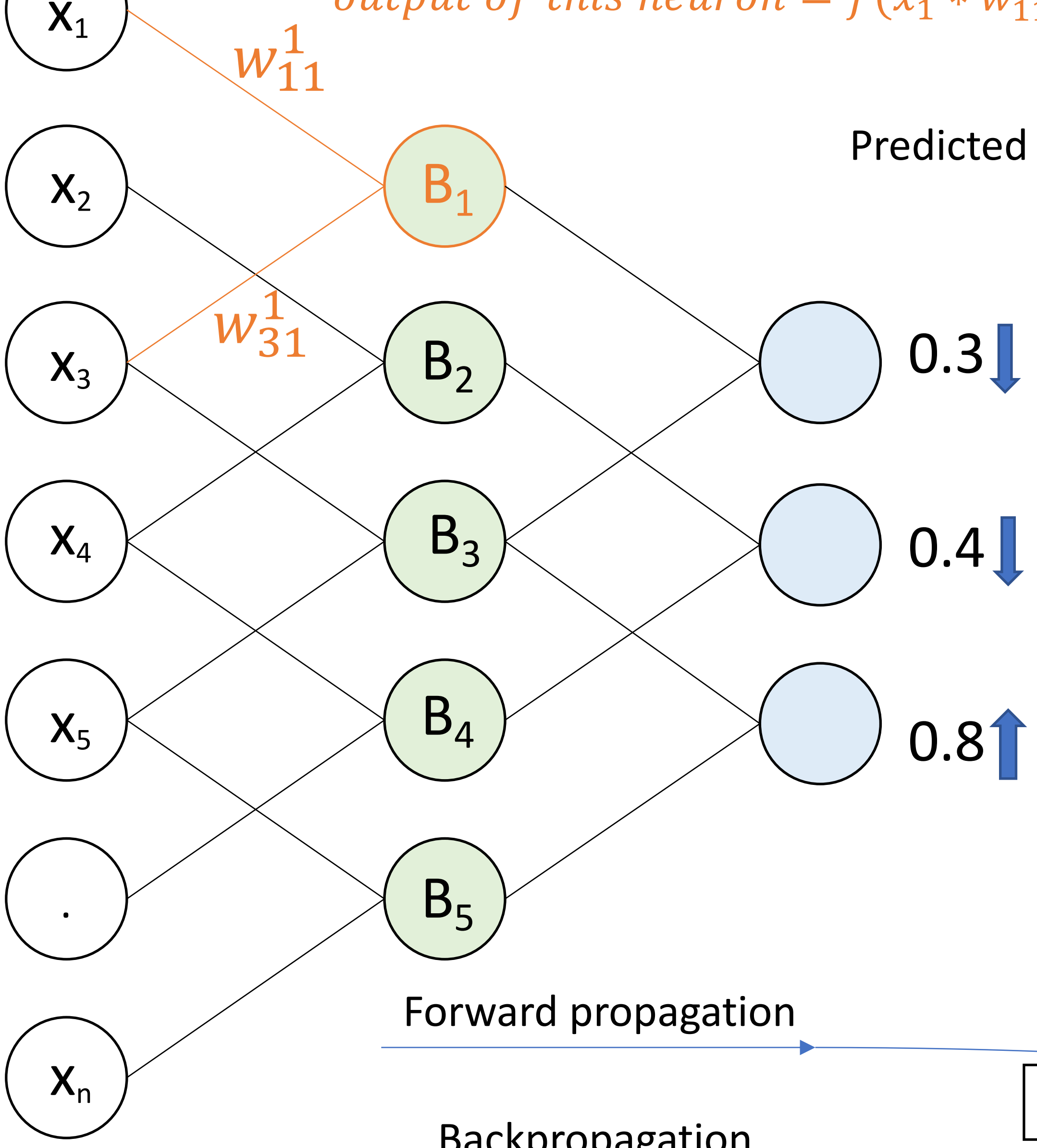
The weight linking the 1st hidden layer and its prior layer

w_{31}^{1}

Neuron 3 from previous layer

Neuron 1 in the current layer

$$output\ of\ this\ neuron = f(x_1 * w_{11}^{1} + x_3 * w_{31}^{1} + B_1)$$



True Value (target)	Error
0	-0.3
0	-0.4
1	0.2

Forward propagation

Calculate loss

Backpropagation

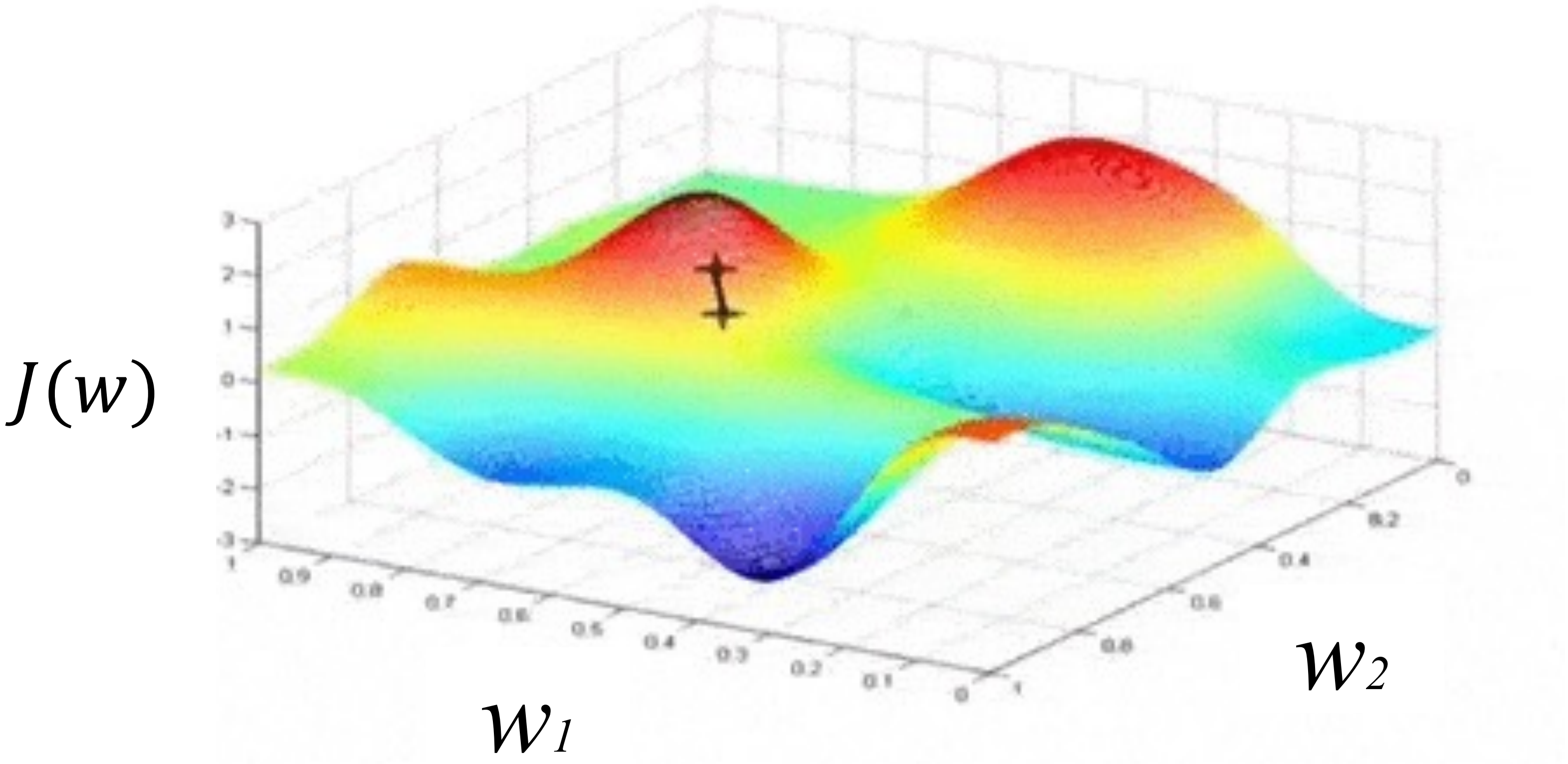
The training of a neural network is the process of minimizing the loss function.

How?

Gradient Descent + Backpropagation

Gradient of a (loss) function:

J can be any function.
w is the parameter of the function.
(It is the weights in a neural network.)



Gradient descent:

1. Compute the slope (gradient) at the current step $\frac{\partial J}{\partial w}$

2. Make a move in the direction opposite to the slope

(Modified From Andrew Ng)

This means we need to modify the parameters (*w*) during the backpropagation.

Learning rate

Gradient descent:

1. Compute the slope (gradient) at the current step $\frac{\partial J}{\partial w}$

2. Make a $\boxed{\textbf{move}}$ **in the direction opposite to the slope**

The move = $-\eta \frac{\partial J}{\partial w}$

Learning rate

Next class:

Python

ML algorithms and demos

Build a neural network

You can install Python and Jupyter notebook on your computer,
or use Google Colab in the browser instead, which needs your Google account.

For deep learning demonstrations, we’ll always use Google Colab.

Run codes on your computer

1.Install Python: https://www.python.org/downloads/

2.Install Jupyter: Run this in the command line

```
pip3 install --upgrade pip
pip3 install jupyter
```

3.How to use Jupyter: https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/execute.html

Run codes in the cloud: Google Colab

Google Colab:

1. First you need a Google Account
2. Go to https://colab.research.google.com
3. Create a New notebook