LECTURER: TAI LE QUY

DATA SCIENCE

TOPIC OUTLINE

UNIT 6

SELECTED ARTIFICIAL INTELLIGENCE TECHNIQUES

STUDY GOALS

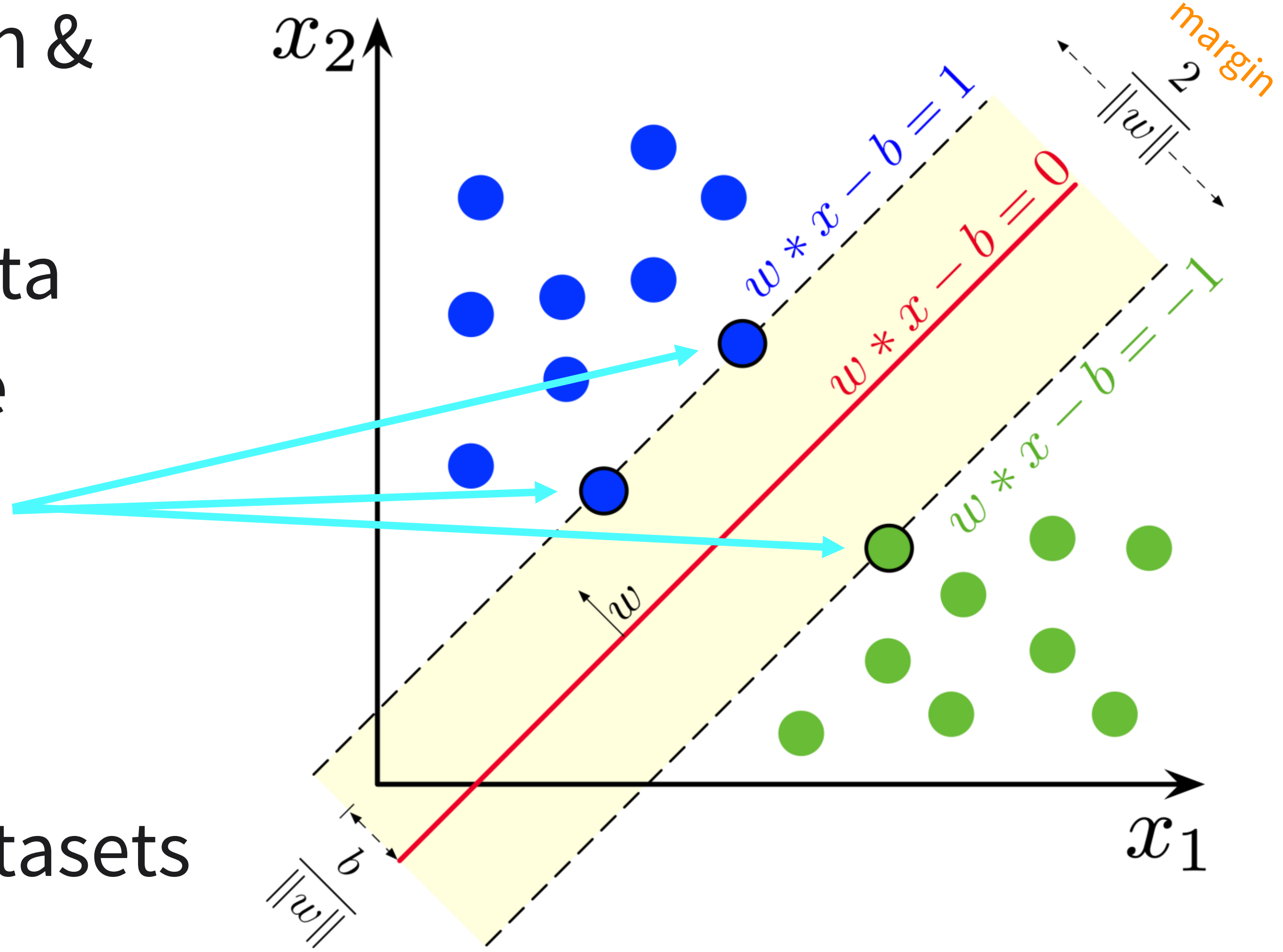
On completion of this unit, you will have learned …

- data classification by support vector machines.
- the feedforward neural network structure.
- the back propagation algorithm in neural networks. how to develop an artificial neural networks prediction model.
- recurrent networks and reinforcement learning.
- basics about genetic algorithms, fuzzy logic, and Naïve Bayes classification.

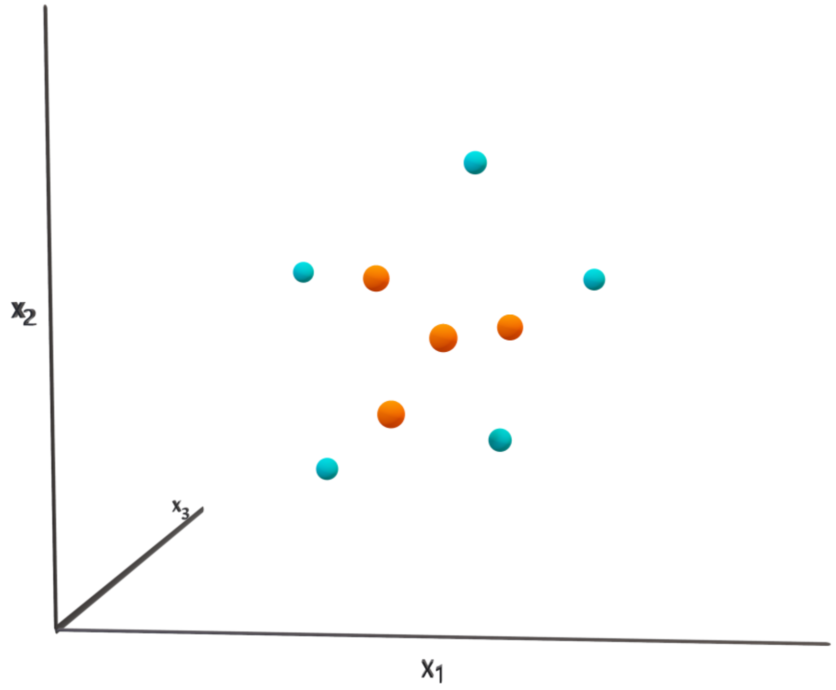
EXPLAIN SIMPLY

1. Explain the concept of Support Vector Machines (SVM) and the usage of the kernel tricks.
2. Name two activation functions. Can you draw them?
3. Describe the usage of Gradient Descent in Neural Networks.

- model used for regression & classification tasks
- identify hyperplane in data space that maximizes the margin between support vectors
- apply kernel trick for nonlinearly separable datasets

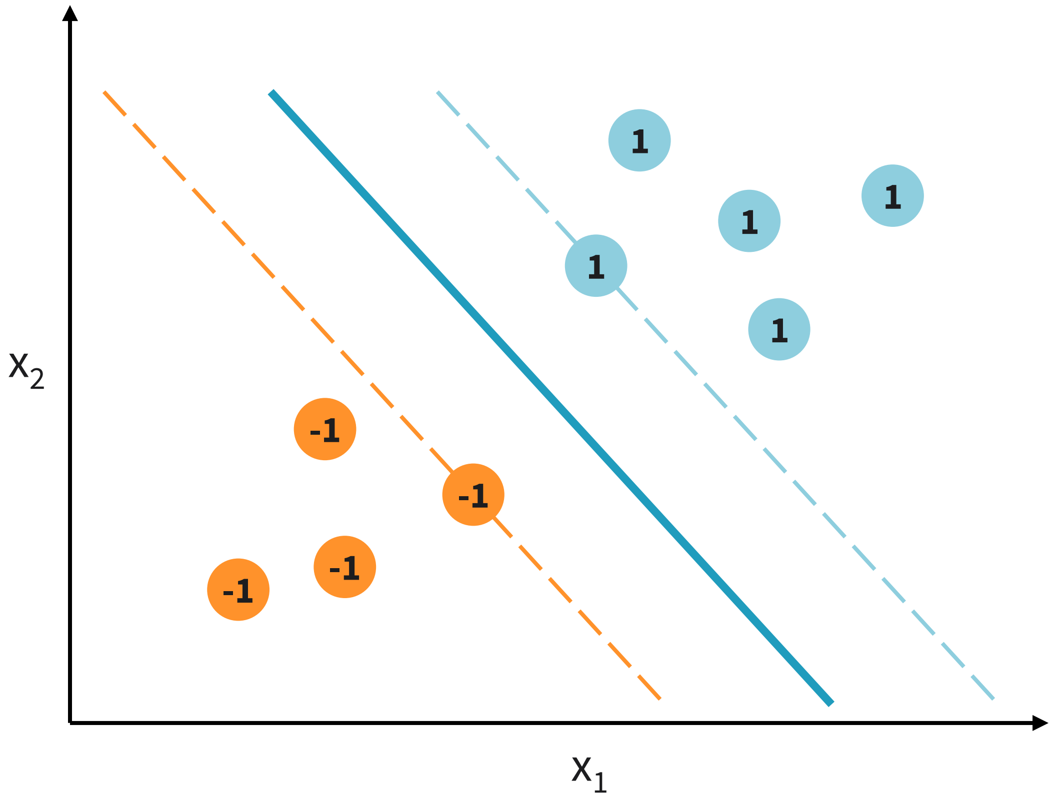


- **n-dimensional feature space**
- **n-1-dimensional decision boundary** hyperplane
- **Example**
  - 3 Features → 2-dimensional decision boundary plane
  - 2 Features → 1-dimensional decision boundary line
- Classes might **only be separable** in **higher dimensions**



Source of text: Boehmke & Greenwell, 2019, Hastie et al., 2017
Source of image 1: Christian Müller-Kett, 2021

- **Soft margin** allows samples to be misclassified
- Larger margin = **many misclassifications**
- Narrower margin = **few misclassifications**

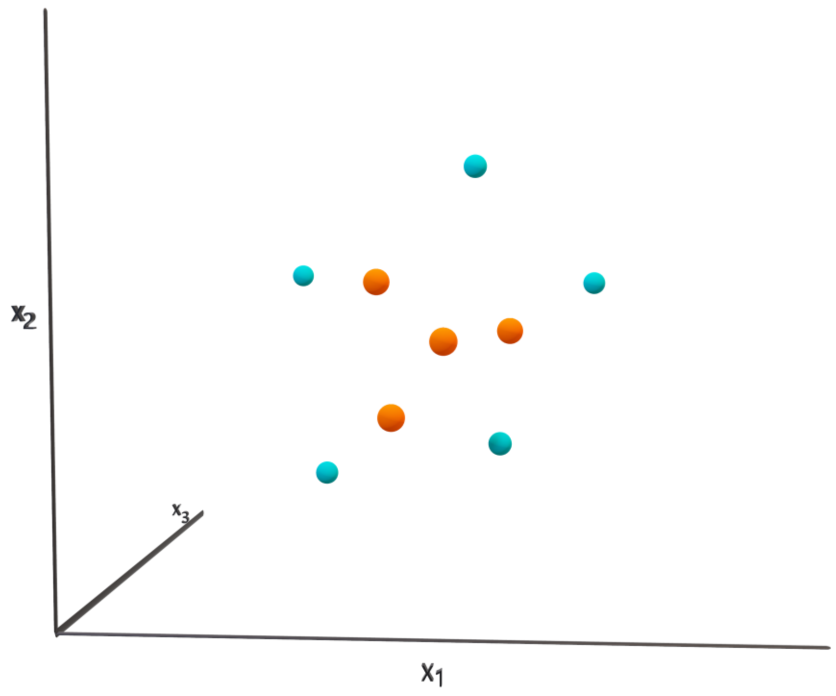


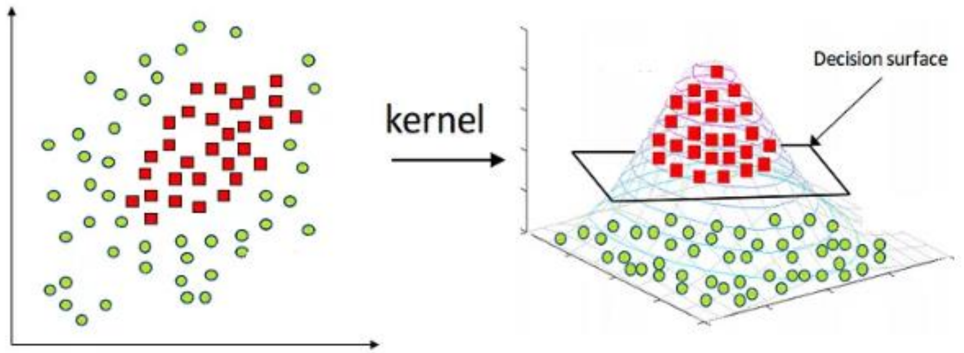
Source of text: Boehmke & Greenwell, 2019, Hastie et al., 2017
Source of image 1: Christian Müller-Kett, 2021

THE KERNEL TRICK

Classes might **only be separable** in **higher dimensions**

- Tranform the data to higher dimensions
- Calculate the **dot product** of the **transformed data**
- **Kernel functions** give the **same results** as the dot product of the transformed data
- **We do not have to transform** the data to higher dimensions
- Common Kernel functions include polynomial, sigmoid, and radial



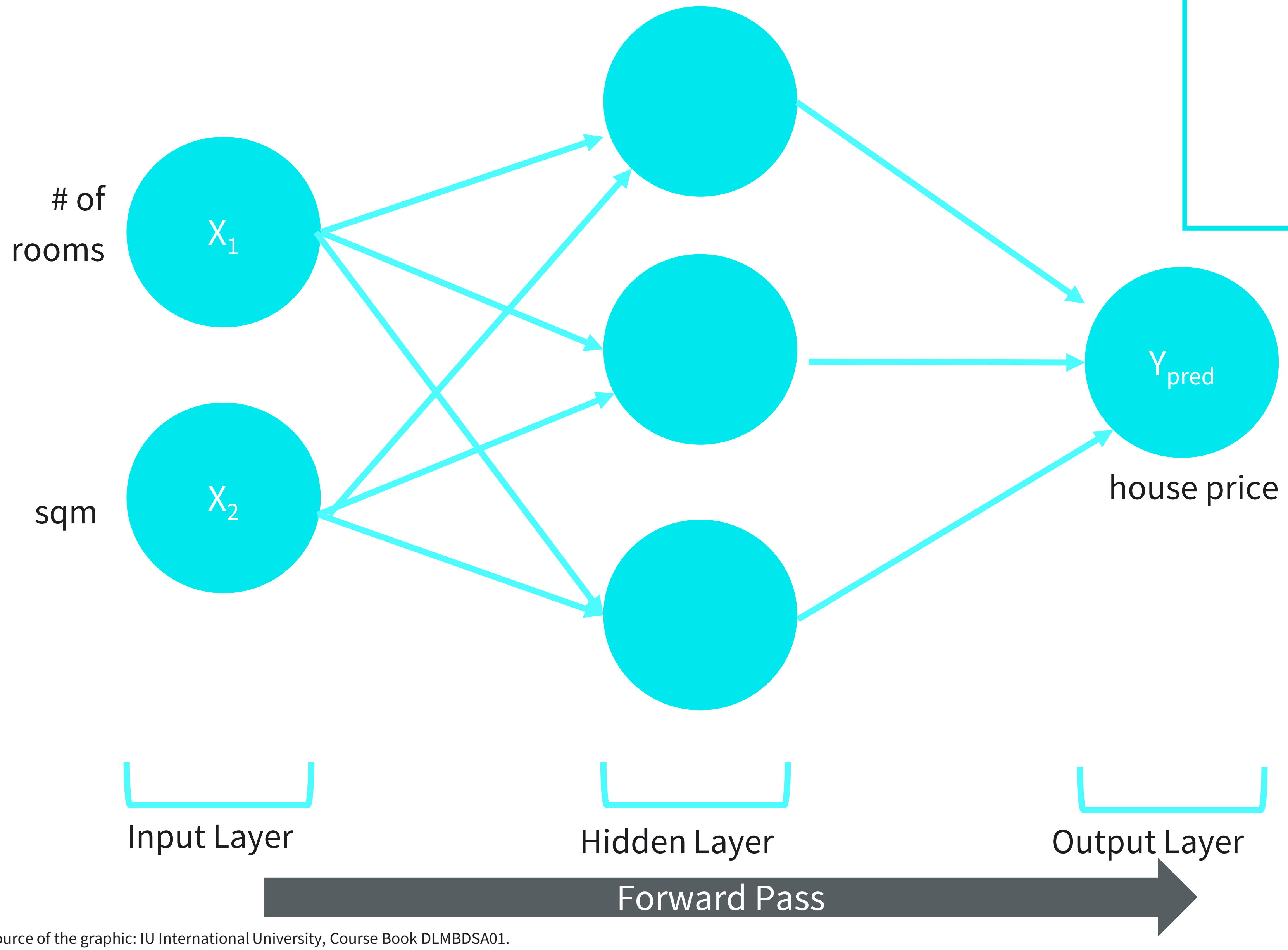


https://scikit-learn.org/stable/modules/svm.html

Source of image 1: Christian Müller-Kett, 2021
Source of image 2: https://medium.com/@Suraj_Yadav/what-is-kernel-trick-in-svm-interview-questions-related-to-kernel-trick-97674401c48d

FEEDFORWARD NEURAL NETWORKS

Forward Pass



Source of the graphic: IU International University, Course Book DLMBDSA01.

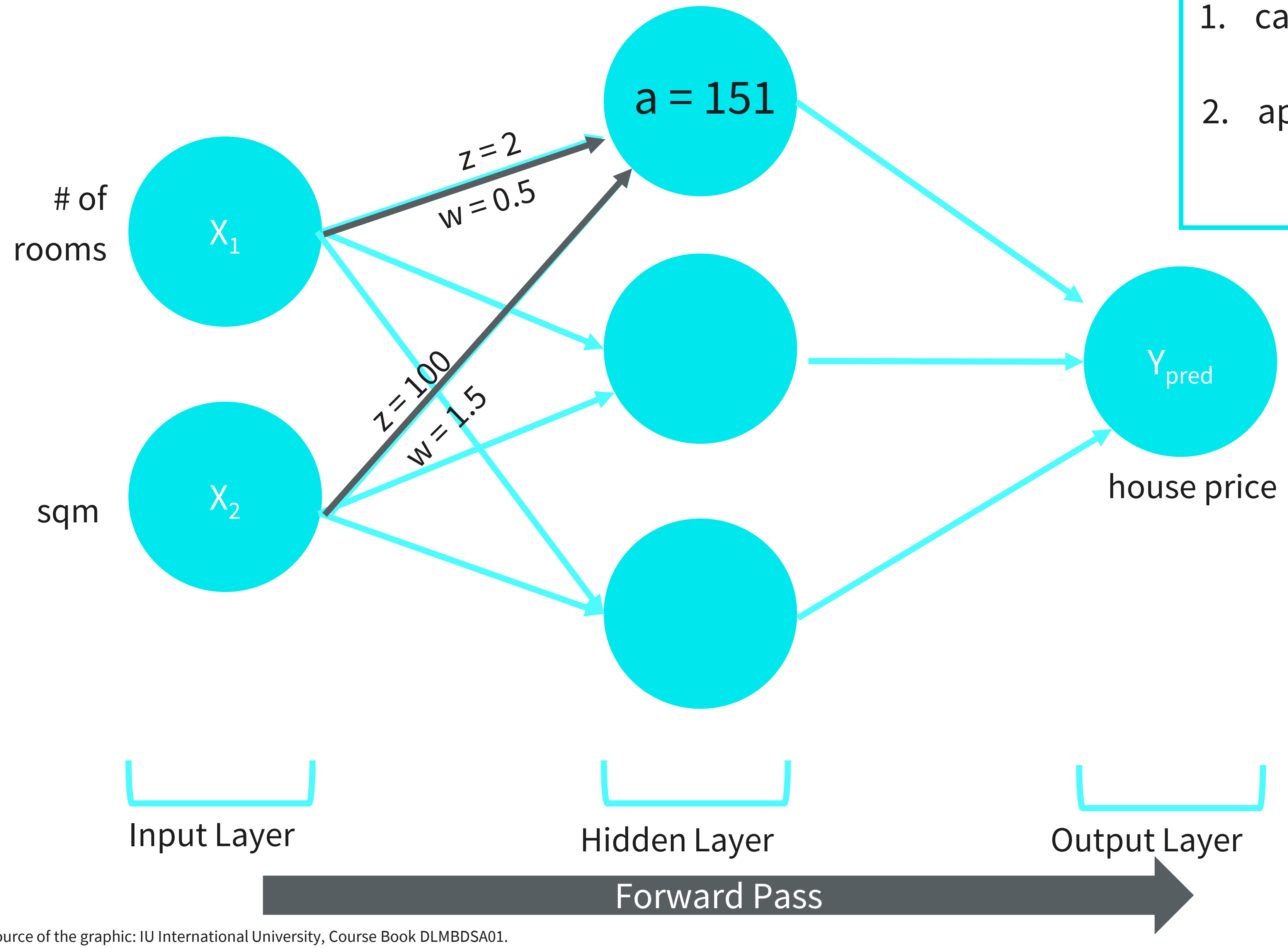
FEEDFORWARD NEURAL NETWORKS

Forward Pass

1. calculate weighted sum a from X_1 & X_2

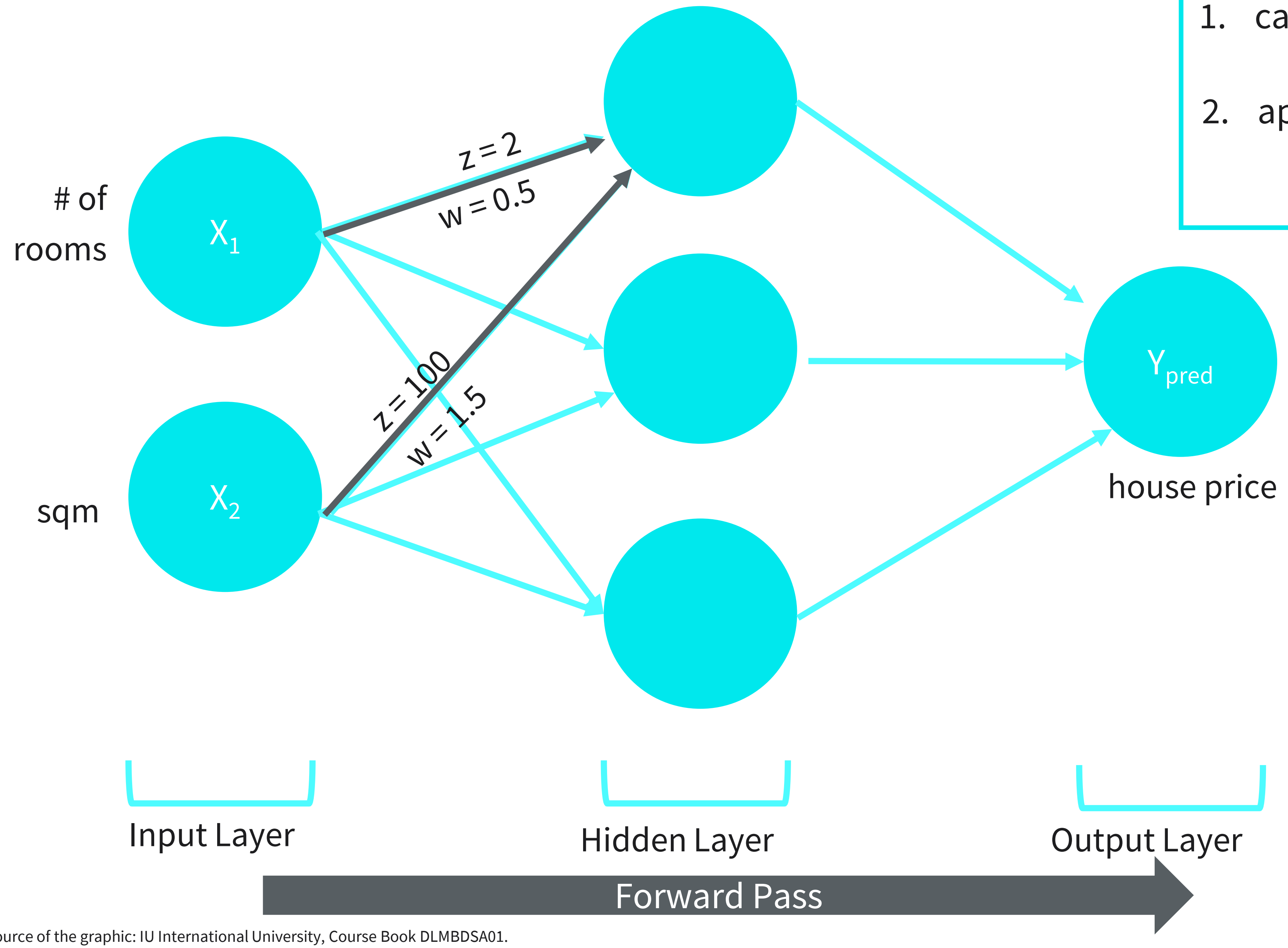
$$a = 2 \cdot 0.5 + 100 \cdot 1.5 = 151$$

2. apply Activation Function f(a) to get z



Source of the graphic: IU International University, Course Book DLMBDSA01.

FEEDFORWARD NEURAL NETWORKS



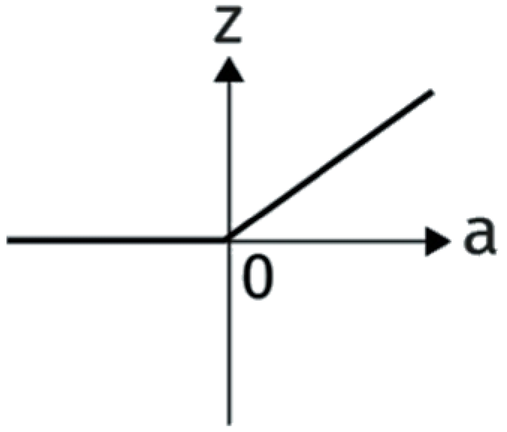
Forward Pass

1. calculate weighted sum a from X_1 & X_2

   $a = 2 \cdot 0.5 + 100 \cdot 1.5 = 151$

2. apply Activation Function f(a) to get z

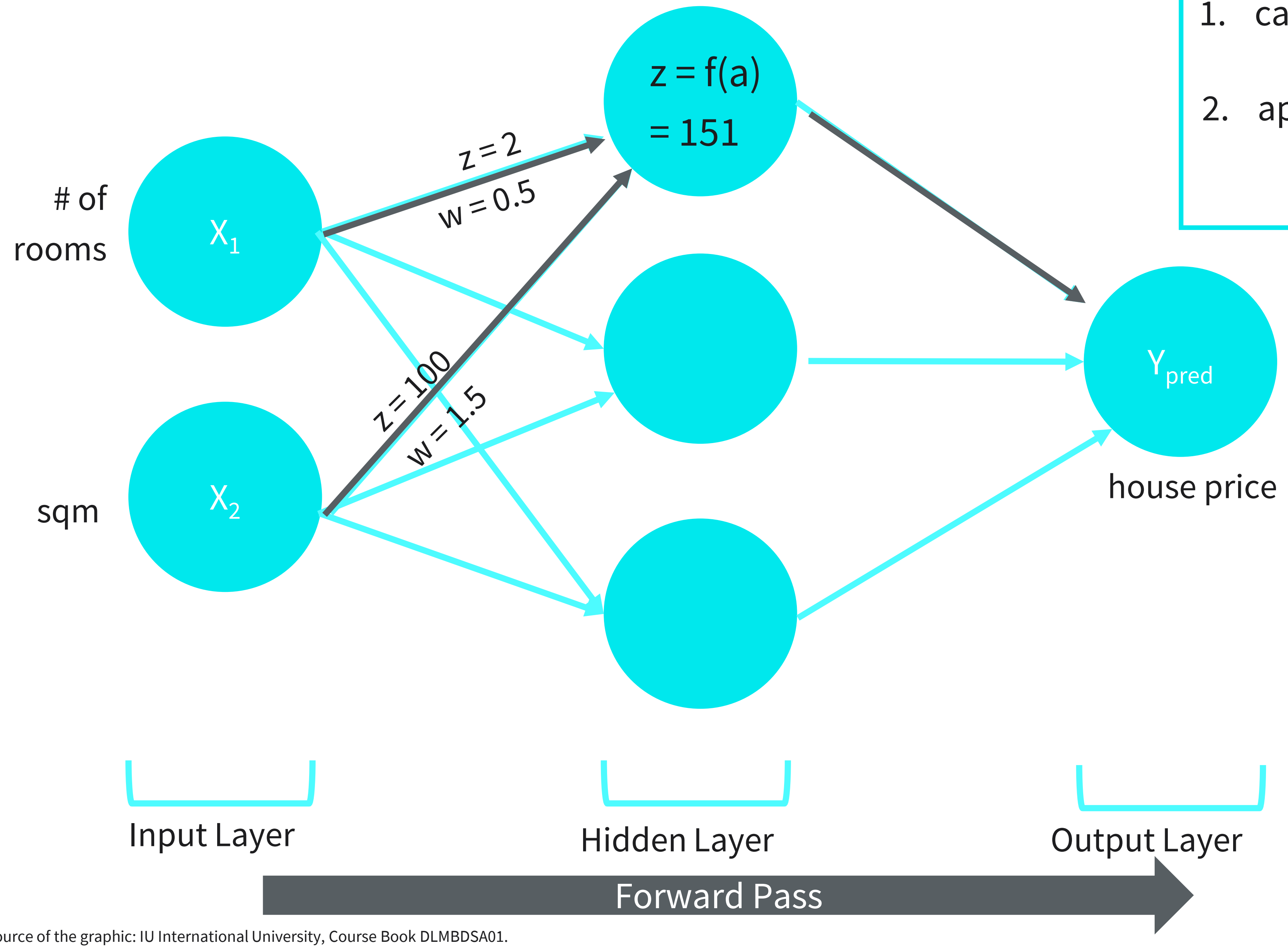
Rectified Linear Unit (ReLU)



$$z = \max(a, 0)$$

Source of the graphic: IU International University, Course Book DLMBDSA01.

FEEDFORWARD NEURAL NETWORKS



Forward Pass

1. calculate weighted sum a from X_1 & X_2

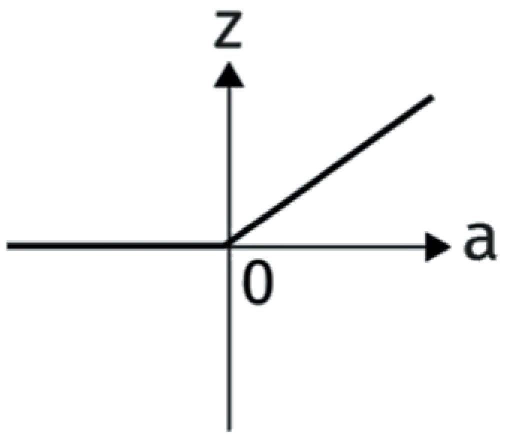
$$a = 2 \cdot 0.5 + 100 \cdot 1.5 = 151$$

2. apply Activation Function f(a) to get z

$$z = f(151) = 151$$

... repeat until Output Neuron

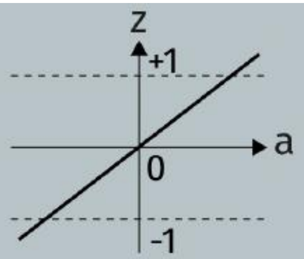
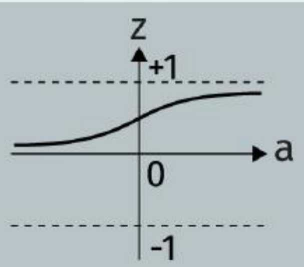
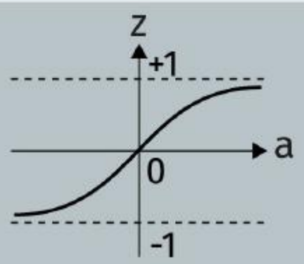
Rectified Linear Unit (ReLU)

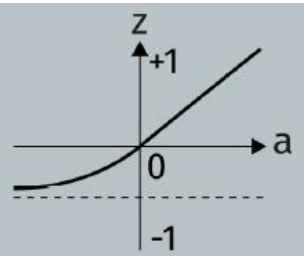
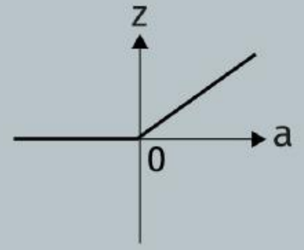


$$z = \max(a, 0)$$

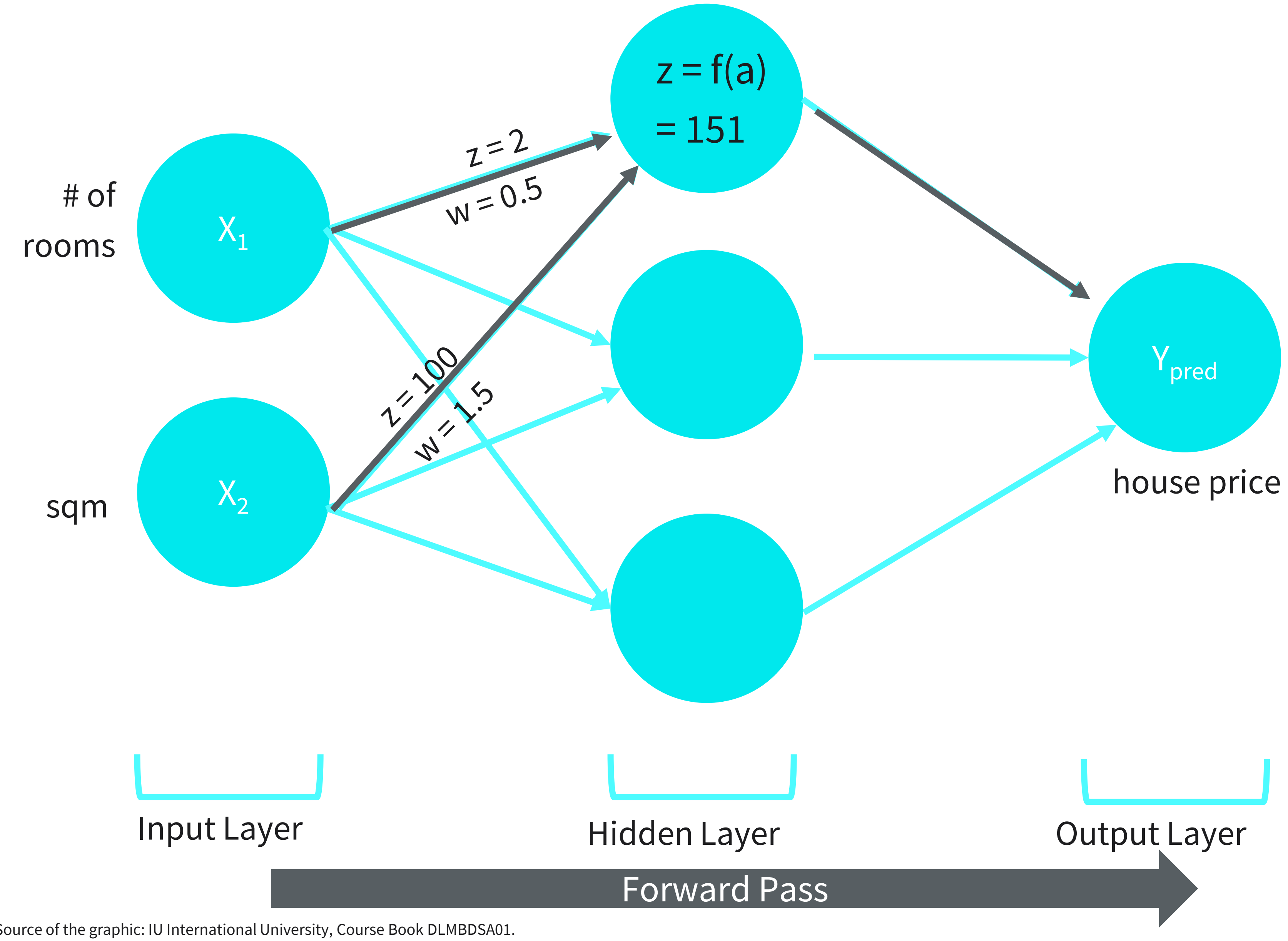
Source of the graphic: IU International University, Course Book DLMBDSA01.

ACTIVATION FUNCTIONS

Linear (lin)	 $z = a$	The lin function generates outputs which are not confined to a specific range.
log-sigmoid (logsig)	 $z = \frac{1}{1 + e^{-a}}$	The logsig function generates outputs between 0 and 1.
tan-sigmoid (tansig)	 $z = \tanh(a) = \frac{e^{2a} - 1}{e^{2a} + 1}$	The tansig function generates outputs between -1 and +1.

Exponential linear unit (ELU)	 $z = \begin{cases} a & a > 0 \\ \alpha (e^{a} - 1) & a \leq 0 \end{cases}$ where α is a positive value, normally equals 0.01	The ELU function generates outputs which are not confined to a specific range. It has a linear shape for positive inputs and an exponential shape for negative outputs.
Rectified linear unit (ReLU)	 $z = \max(a,0)$	The ReLU function generates outputs which are 0 for inputs with negative values. For all other inputs, the output will equal the input number.

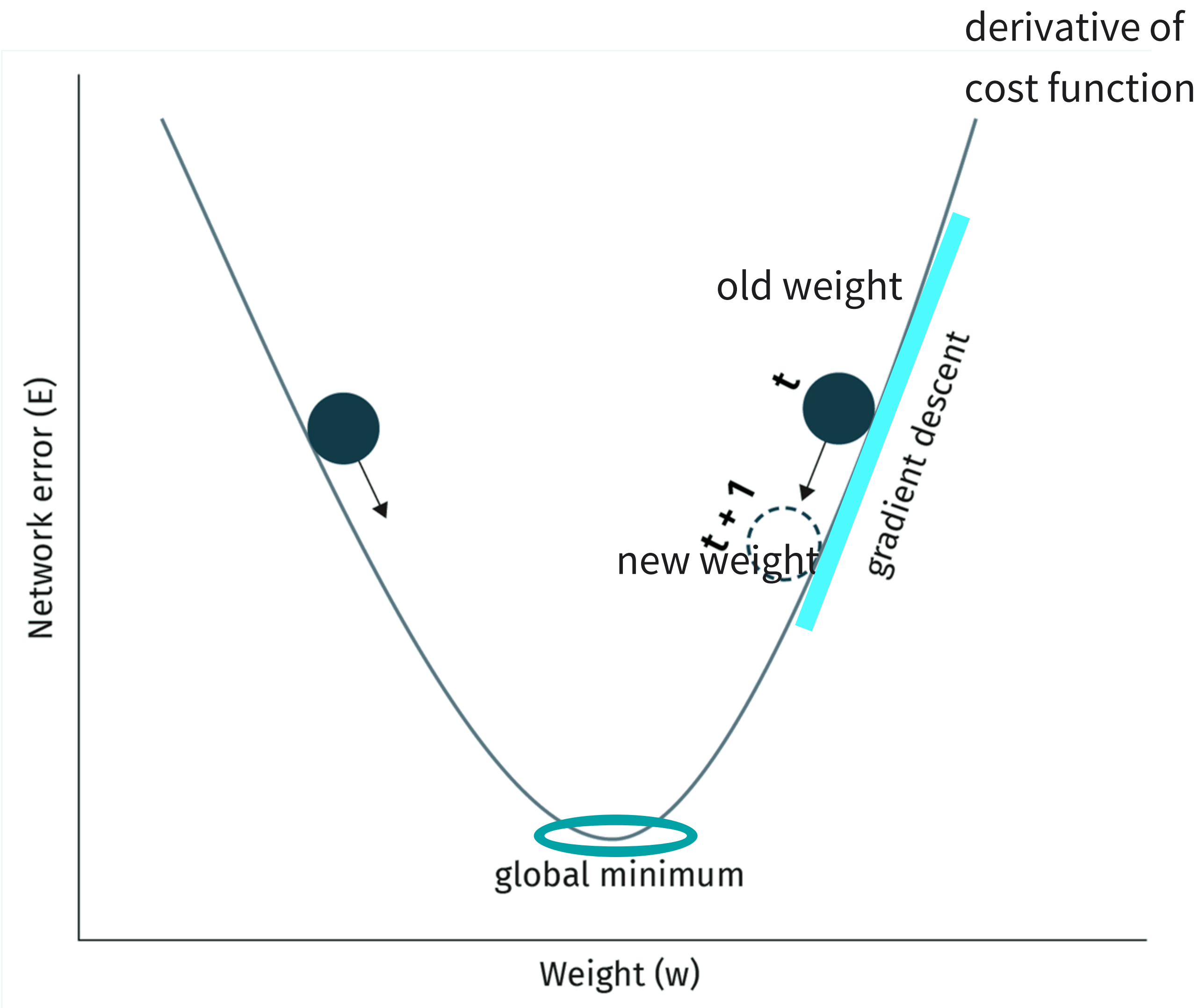
FEEDFORWARD NEURAL NETWORKS



… how do we define network weights w?
→ backpropagation

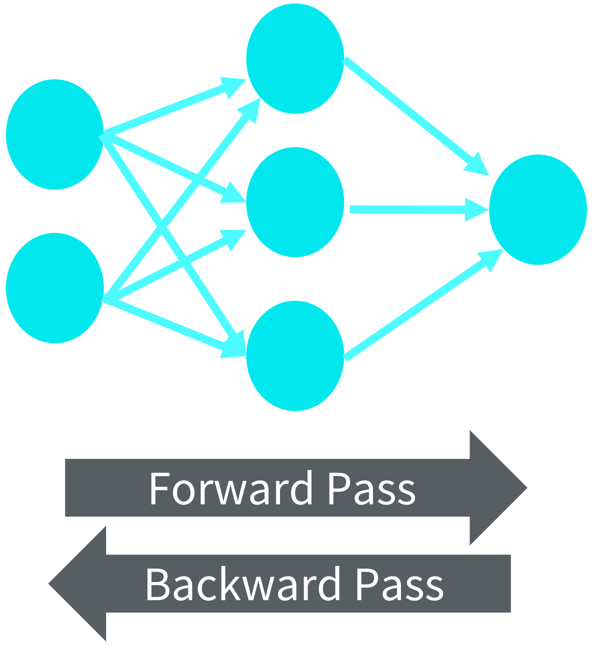
Source of the graphic: IU International University, Course Book DLMBDSA01.

- Algorithm for finding a local minimum of a differentiable function
- in ML: **find weights that minimize the error function**
- calculate the gradient of the error function with respect to network weights



Source of the graphic: IU International University, Course Book DLMBDSA01.

BACKPROPAGATION ALGORITHM



1. Randomly initialize weights
2. Calculate output of every neuron
3. Calculate the error for 2.
4. Update the weights with GD

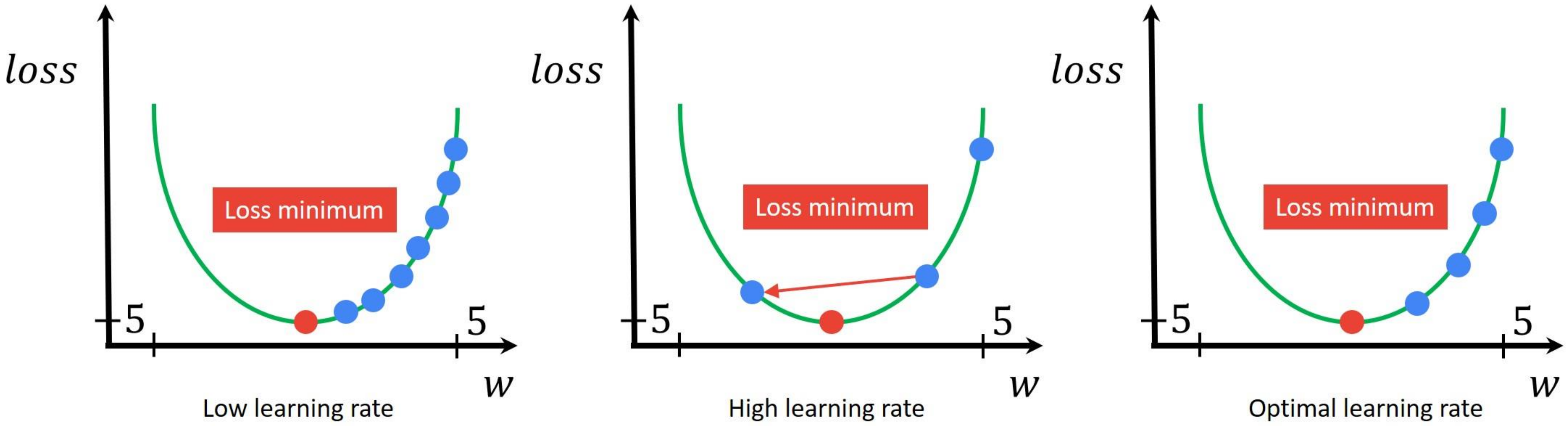
$$w_{new} = w_{old} - \eta \left(\frac{\partial \text{ Error}}{\partial \, w_{old}}\right)$$

Derivative of Error with respect to weights

learning rate

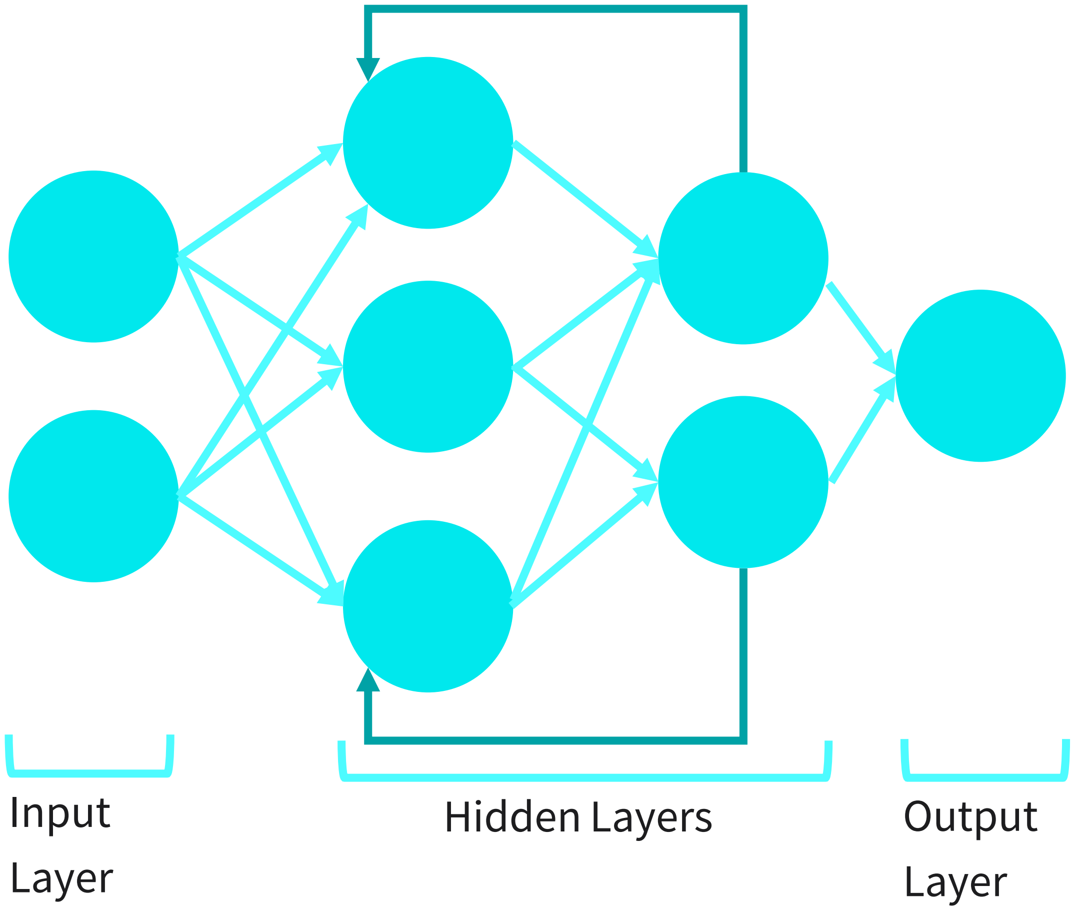
5. Start new forward pass with updated weights
6. Repeat steps 2-4 until no improvement in Error achieved

LEARNING RATE



Source of the image: https://datahacker.rs/category/pytorch/page/3/

Feedforward Neural Networks

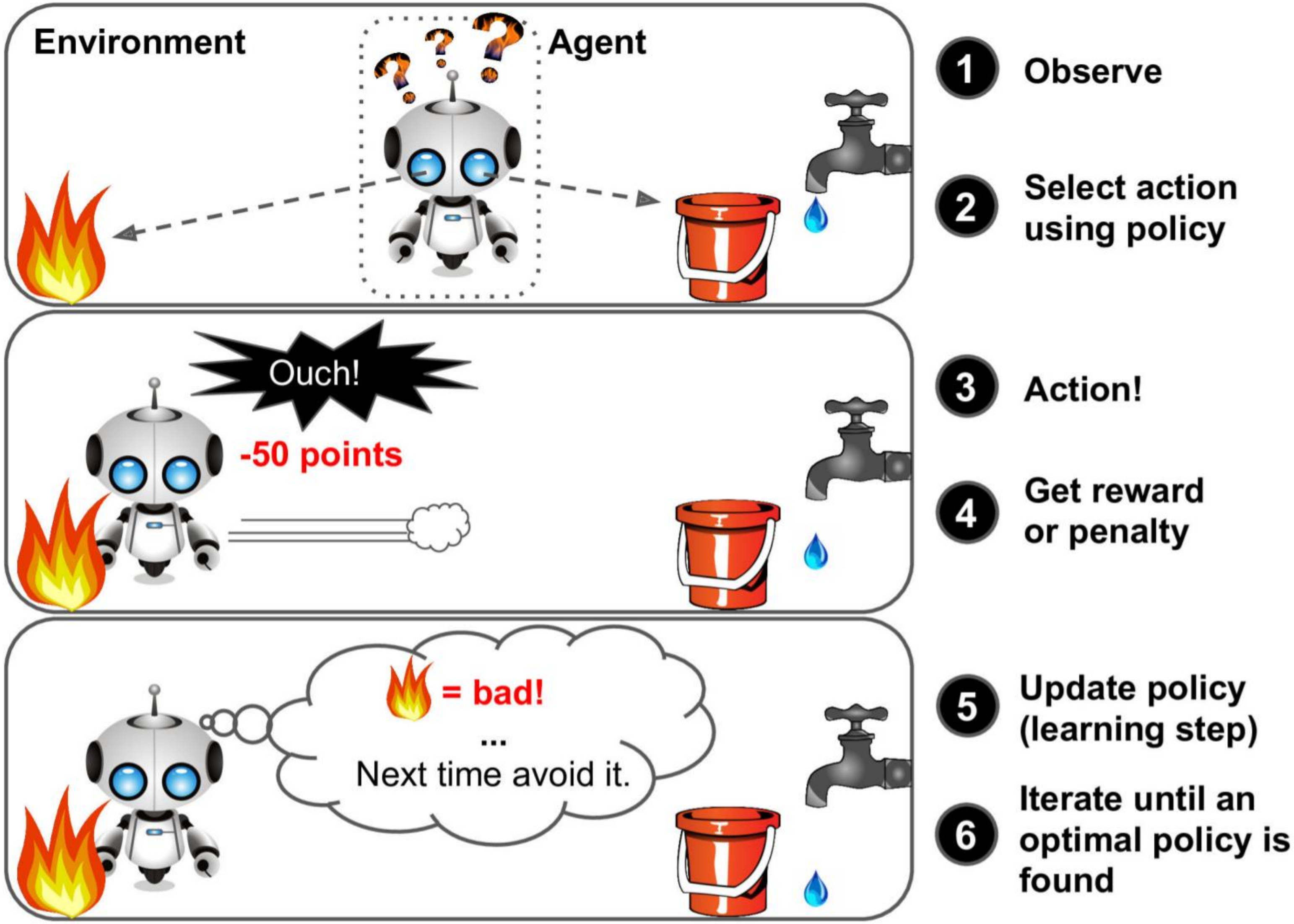


Recurrent Neural Networks

- Allow connections to previous layers
- Memory cells to retain information in deeper neural networks

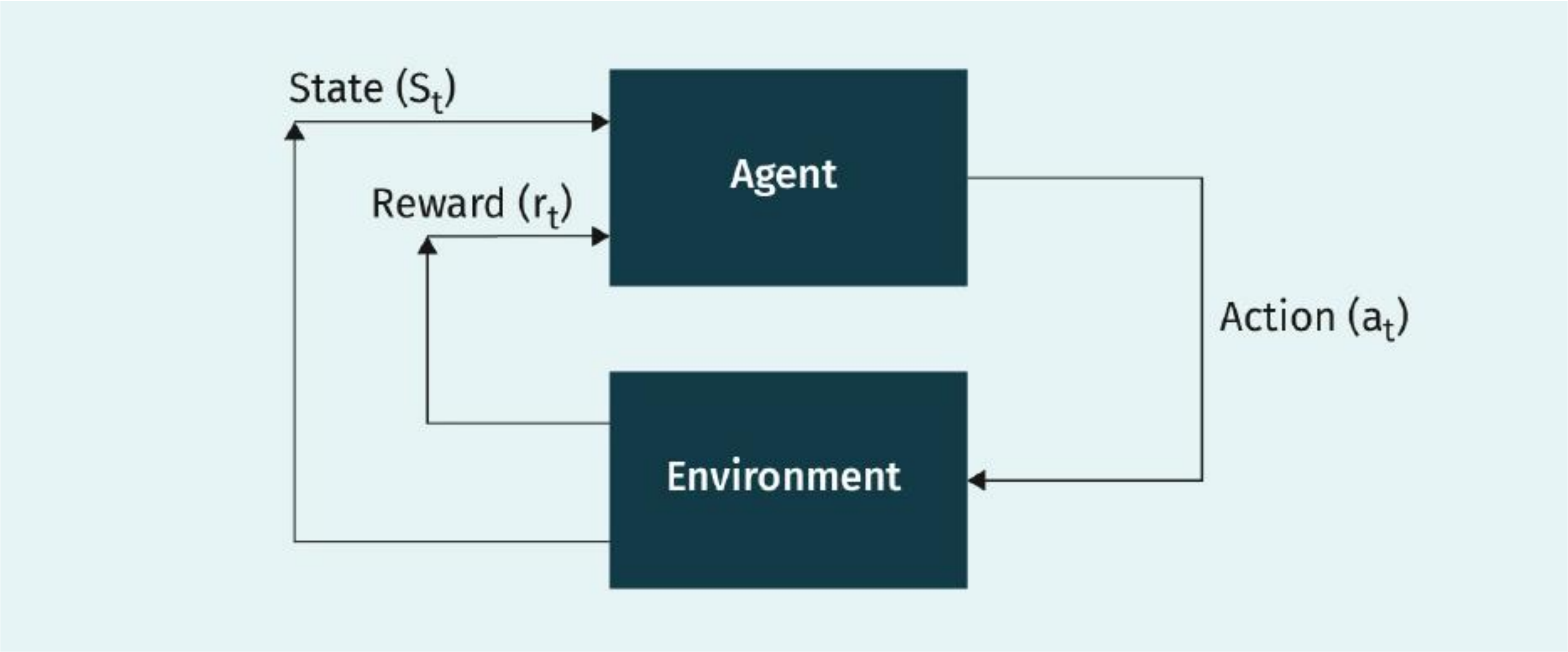
Source of the graphic: IU International University, Course Book DLMBDSA01.

- Algorithm learns a policy how to act in a given environment through trial-and-error actions
- goal: maximize the reward for the agent



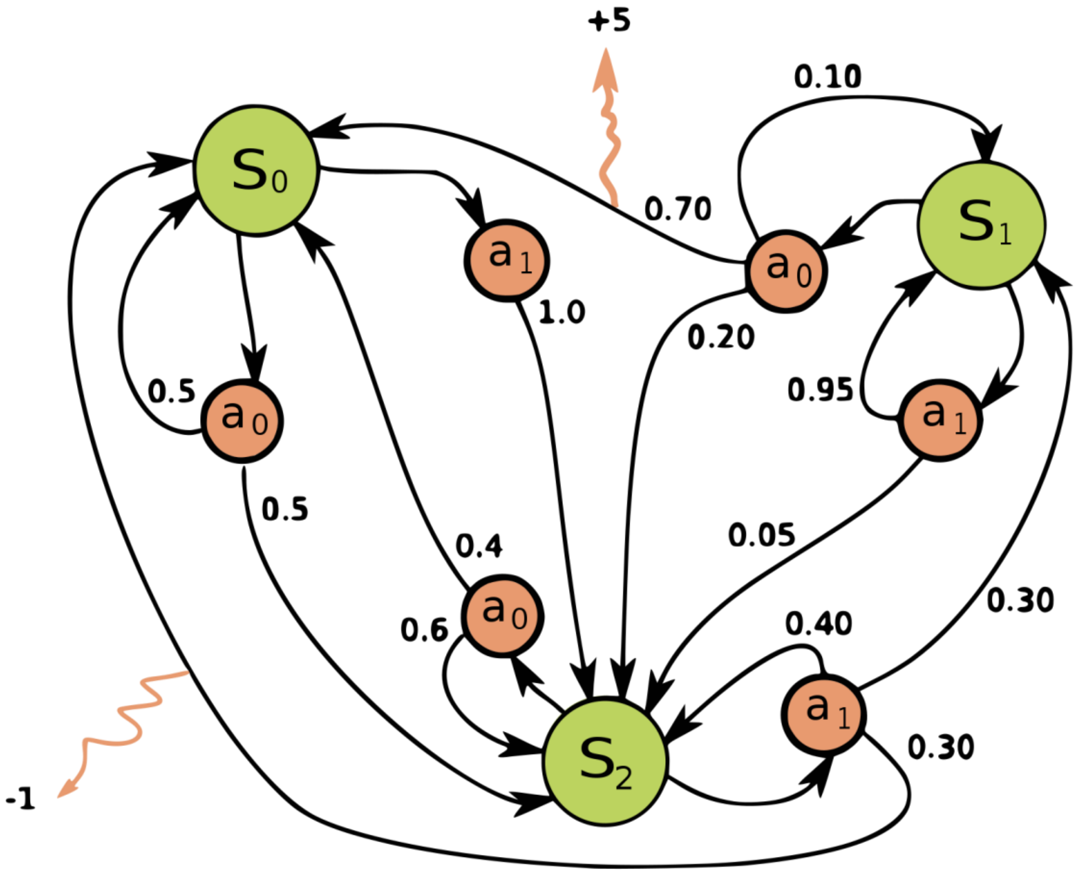
Source of graphics: IU International University, Course Book DLMBDSA01; Géron, 2019, p. 15.

REINFORCEMENT LEARNING



Source of the image: Coursebook

- framework to solve reinforcement learning problems
- set of states $\{s_0, s_1, s_2\}$
- set of actions to take a path $\{a_0, a_1 \ldots\}$
- set of rewards {+10,+40,-50}
- policy for selected path $\{s_0 \rightarrow s_1 \rightarrow s_2\}$



Source of the graphic: Waldo Alvarez, 2017, https://commons.wikimedia.org/wiki/File:Markov_Decision_Process.svg, CC BY-SA 4.0.

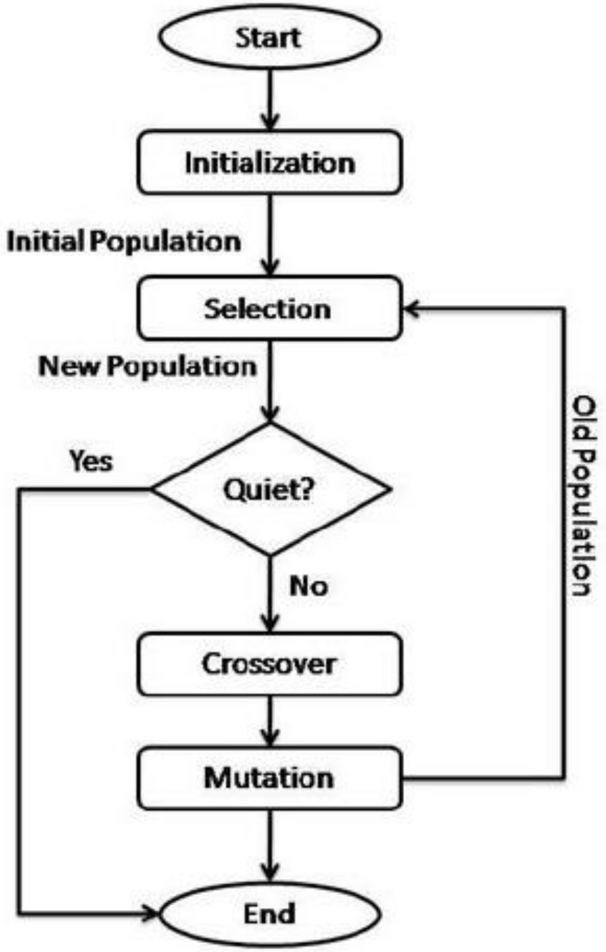
The parameters:

- Set of states $\{S_1, S_2, \ldots S_t, \ldots\}$
- Set of actions to take a specific path (e.g., $\{S_2 \rightarrow S_3\}$)
- Set of rewards $\{r_1, r_2, \ldots, r_t, \ldots\}$
- The policy, which is the selected path to complete the task (e.g., $\{S_1 \rightarrow S_3 \rightarrow S_5\}$)
- The value (V): total reward achieved by following this policy

Algorithm for policy-based:

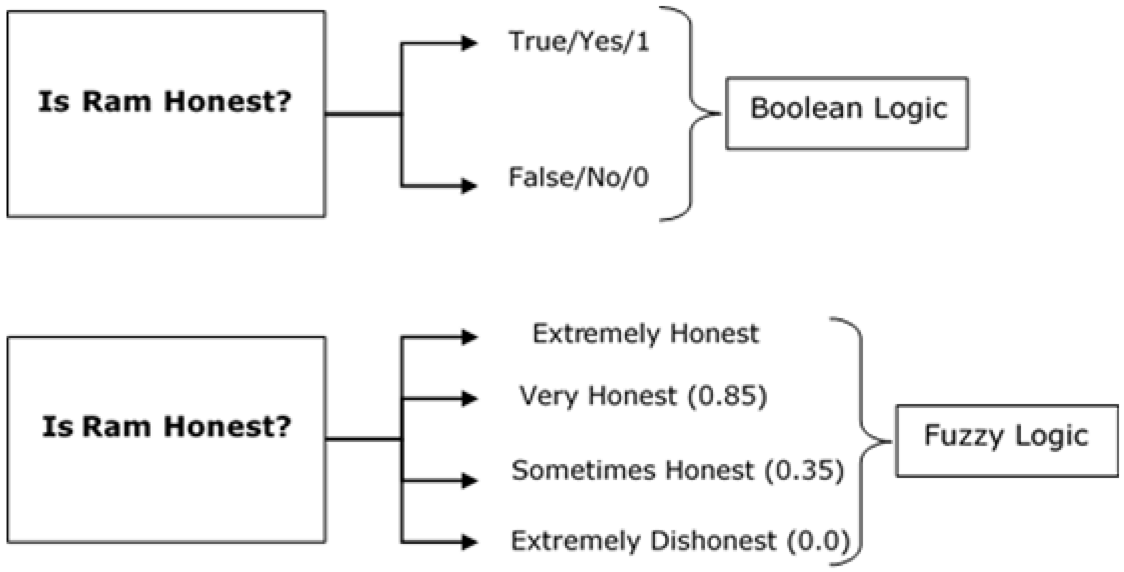
1. Initialize the parameters {S, A, R, P, and V}.
2. Observe the current state $\{S_t\}$.
3. Choose an action {rt+1} according to the maximum possible reward for the next state.
4. Take the action and reach the new state {St+1}.
5. Update the value {V}.
6. Repeat the process until the terminated (end) state is reached.

- Solving optimization problems based on natural selection.
- Starts by generating a population of possible solutions and then applies selection rules to randomly select individuals from the current population to be parents.
- The crossover rules are applied to combine two parents and form children for the next generation.
- Mutation rules are applied with random changes to the parents to form different children.
- Over successive generations, the population evolves toward the optimal solution (in this case, children with the best genetic combinations).



Source of image: https://www.researchgate.net/publication/261028418_Used_Genetic_Algorithm_for_Support_Artificial_Neural_Network_in_Intrusion_Detection_System

- Deals with classes of objects with unsharp boundaries.
- The membership of these clusters is based on degrees of truth instead of the usual {+1,–1} assignments.
- The first step of fuzzy logic is to fuzzify (decompose) all input values into truth values, which are any real numbers between "0" (completely false) and "1" (completely true).
- The fuzzy output is computed by the execution of a group of "IF-THEN" rules in a way that mimics Boolean logic operators ("AND", "OR", and "NOT").
- Finally, a de-fuzzification step is performed on the fuzzy truth values to get a continuous output value



NAÏVE BAYES CLASSIFIER

Example

$$P(x \mid c) = P(Sunny \mid Yes) = 3/9 = 0.33$$

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

Likelihood Table		Play Golf		
		Yes	No	
Outlook	Sunny	3/9	2/5	5/14
	Overcast	4/9	0/5	4/14
	Rainy	2/9	3/5	5/14
		9/14	5/14	

$$P(x) = P(Sunny) = 5/14 = 0.36$$

$$P(c) = P(Yes) = 9/14 = 0.64$$

Posterior Probability: $P(c \mid x) = P(Yes \mid Sunny) = 0.33 \times 0.64 \div 0.36 = 0.60$

$$P(x \mid c) = P(Sunny \mid No) = 2/5 = 0.4$$

Frequency Table		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
		9	5	14

$$P(x) = P(Sunny) = 5/14 = 0.36$$

$$P(c) = P(No) = 5/14 = 0.36$$

Posterior Probability: $P(c \mid x) = P(No \mid Sunny) = 0.40 \times 0.36 \div 0.36 = 0.40$

Source of image: https://www.saedsayad.com/naive_bayesian.htm

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Bayes theorem

$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\ \overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Strong independence assumption between random variables

NAÏVE BAYES CLASSIFIER

Frequency Table

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Likelihood Table

		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Outlook	Temp	Humidity	Windy	Play
Rainy	Cool	High	True	?

$$P(Yes \mid X) = P(Rainy \mid Yes) \times P(Cool \mid Yes) \times P(High \mid Yes) \times P(True \mid Yes) \times P(Yes)$$

$$P(Yes \mid X) = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.00529 \rightarrow 0.2 = \frac{0.00529}{0.02057 + 0.00529}$$

$$P(No \mid X) = P(Rainy \mid No) \times P(Cool \mid No) \times P(High \mid No) \times P(True \mid No) \times P(No)$$

$$P(No \mid X) = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.02057 \rightarrow 0.8 = \frac{0.02057}{0.02057 + 0.00529}$$

Source of image: https://www.saedsayad.com/naive_bayesian.htm

You have learned …

- data classification by support vector machines.
- the feedforward neural network structure.
- the back propagation algorithm in neural networks. how to develop an artificial neural networks prediction model.
- recurrent networks and reinforcement learning.
- basics about genetic algorithms, fuzzy logic, and Naïve Bayes classification.

SESSION 6

TRANSFER TASK

TRANSFER TASK 1

1. Discuss the parameter of ***learning rate η*** in the context of Gradient Descent.
2. How does it influence the process?
3. Can you foresee challenges in choosing the adequate learning rate?

TRANSFER TASK 2

Working in groups

- Select your domain (e.g., healthcare, education, finance, etc.)
- Select a classification and/or regression task (e.g., credit scoring, customer demand prediction, diabetes prediction, etc.)
- Describe how SVM and/or artificial neural networks and/or Naïve Bayes techniques can be applied?
- Present your finding in 5 minutes

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.
The results will be discussed in plenary.

1. The Naïve Bayes approach assumes that the independent variables are…

   a) random variables.
   b) orthogonal variables.
   c) normalized variables.
   d) structured data variables.

2. A memory cell is a concept which exists in …

   a) feedforward networks.
   b) recurrent networks.
   c) reinforcement learning.
   d) support vector machines.

3. The Kernel trick is employed in support vector machines to…

a) maximize the margin between the two classes.
b) minimize the classification error.
c) deal with nonlinearly separable dataset.
d) define the set of support vectors.

How did you like the course?

LIST OF SOURCES

Alvarez, W. (2017). *Markov Decision Process* [Image]. https://commons.wikimedia.org/wiki/File:Markov_Decision_Process.svg, CC BY-SA 4.0.
Géron, A. (2019). *Hands-on machine learning with scikit-learn, keras, and tensorflow : Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Incorporated.
Larhmam (2018). *SVM-Margin* [Image]. https://commons.wikimedia.org/wiki/File:SVM_margin.png, CC BY-SA 4.0.
Jordon, J. (2018). *Setting the learning rate of your neural network*. [Image]. https://www.jeremyjordan.me/nn-learning-rate/