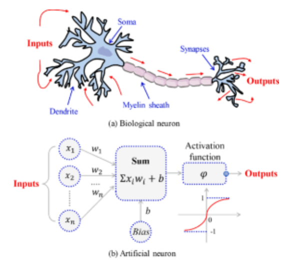
NEURAL NETWORK

10S3001 – Artificial Intelligence

by Samuel I. G. Situmeang



Faculty of Informatics and Electrical Engineering

OBJECTIVES

- Students are able to explain the concept of neural networks, including their common architectures and types.
- Students are able to describe commonly used activation functions in neural networks.
- Students are able to apply the simple perceptron algorithm to construct a classification model and use the classification model to perform accurate inference.

- Mahasiswa mampu menjelaskan konsep jaringan saraf, termasuk arsitektur dan jenisnya yang umum.
- Mahasiswa mampu menjelaskan fungsi aktivasi yang umum digunakan dalam jaringan saraf.
- Mahasiswa mampu menerapkan algoritma *perceptron* sederhana untuk membangun model klasifikasi dan menggunakan model klasifikasi untuk melakukan inferensi yang akurat.

BASIC CONCEPTS OF NEURAL NETWORK

NEURAL NETWORKS

- The definition of a neural network, more properly referred to as an 'artificial' neural network (ANN), is provided by the inventor of one of the first neurocomputers, Dr. Robert Hecht-Nielsen. He defines a neural network as:

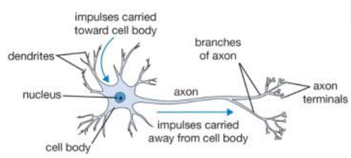
"...a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.“

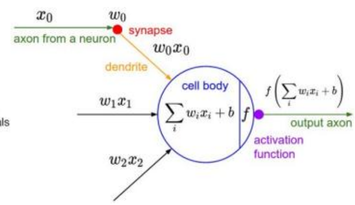
- Or you can also think of Artificial Neural Network as computational model that is inspired by the way biological neural networks in the human brain process information.

Neural networks and deep learning are big topics in Computer Science and in the technology industry, they currently provide the best solutions to many problems in image recognition, speech recognition and natural language processing.

BIOLOGICAL MOTIVATION AND CONNECTIONS

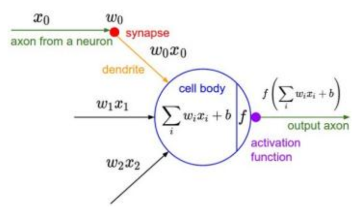
- The basic computational unit of the brain is a **neuron**.
- Approximately 86 billion neurons can be found in the human nervous system and they are connected with approximately $10^{14} - 10^{15}$ **synapses**.
- The diagram below shows a cartoon drawing of a biological neuron (left) and a common mathematical model (right).





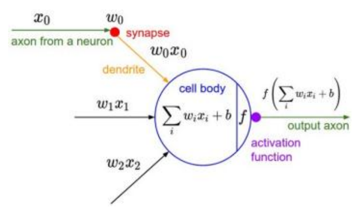
BIOLOGICAL MOTIVATION AND CONNECTIONS

- The basic unit of computation in a neural network is the neuron , often called a **node or unit**.
- It receives input from some other nodes, or from an external source and computes an output.
- Each input has an associated weight (w), which is assigned on the basis of its relative importance to other inputs.
- The node applies a function to the weighted sum of its inputs.



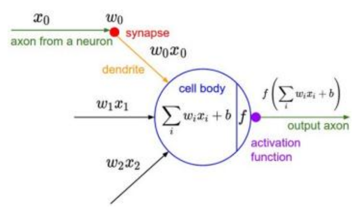
BIOLOGICAL MOTIVATION AND CONNECTIONS

- The idea is that the synaptic strengths (the weights w) are learnable and control the strength of influence and its direction: excitory (positive weight) or inhibitory (negative weight) of one neuron on another.
- In the basic model, the dendrites carry the signal to the cell body where they all get summed.
- If the final sum is above a certain threshold, the neuron can *fire*, sending a spike along its axon.



BIOLOGICAL MOTIVATION AND CONNECTIONS

- In the computational model, we assume that the precise timings of the spikes do not matter, and that only the frequency of the firing communicates information.
- We model the *firing rate* of the neuron with an **activation function** *(e.x sigmoid function)*, which represents the frequency of the spikes along the axon.



NEURAL NETWORK ARCHITECTURE

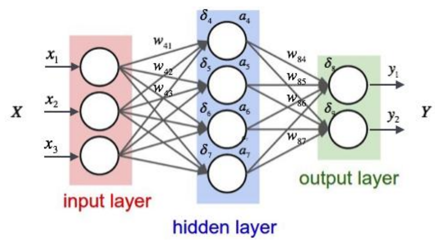
NEURAL NETWORK ARCHITECTURE

- From the previous explanation we can conclude that a neural network is made of neurons.
- Biologically the neurons are connected through synapses where informations flows (weights for out computational model).
- When we train a neural network, we want the neurons to fire whenever they learn specific patterns from the data, and we model the fire rate using an activation function.

- But, that's not everything…

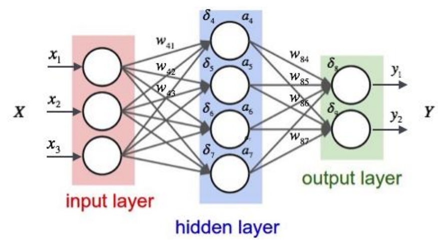
NEURAL NETWORK ARCHITECTURE

- **Input nodes (input layer):** No computation is done here within this layer, they just pass the information to the next layer (hidden layer most of the time). A block of nodes is also called **layer**.



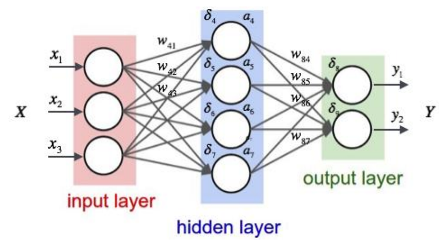
NEURAL NETWORK ARCHITECTURE

- **Hidden nodes (hidden layer):** In Hidden layers is where intermediate processing or computation is done, they perform computations and then transfer the weights (signals or information) from the input layer to the following layer (another hidden layer or to the output layer).
- It is possible to have a neural network without a hidden layer.



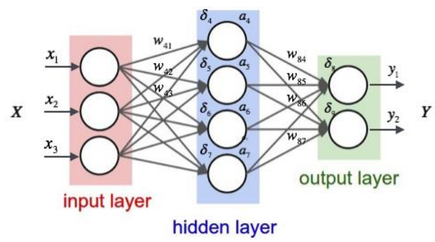
NEURAL NETWORK ARCHITECTURE

- **Output nodes (output layer):** Here we finally use an activation function that maps to the desired output format (e.g. softmax for classification).



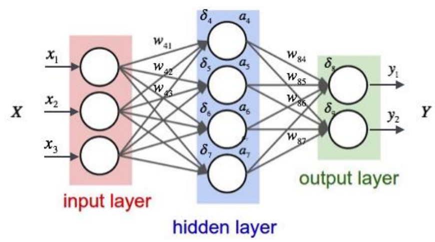
NEURAL NETWORK ARCHITECTURE

- **Connections and weights:** The *network* consists of connections, each connection transferring the output of a neuron i to the input of a neuron j.
- In this sense i is the predecessor of j and j is the successor of i, each connection is assigned a weight w_{ij}.



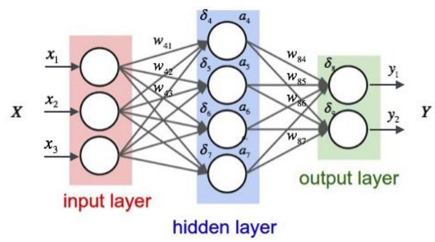
NEURAL NETWORK ARCHITECTURE

- **Activation function:** the **activation function** of a node defines the output of that node given an input or set of inputs.
- A standard computer chip circuit can be seen as a digital network of activation functions that can be "ON" (1) or "OFF" (0), depending on input.
- This is similar to the behavior of the linear perceptron in neural networks.
- However, it is the *nonlinear* activation function that allows such networks to compute nontrivial problems using only a small number of nodes. In artificial neural networks this function is also called the transfer function.



NEURAL NETWORK ARCHITECTURE

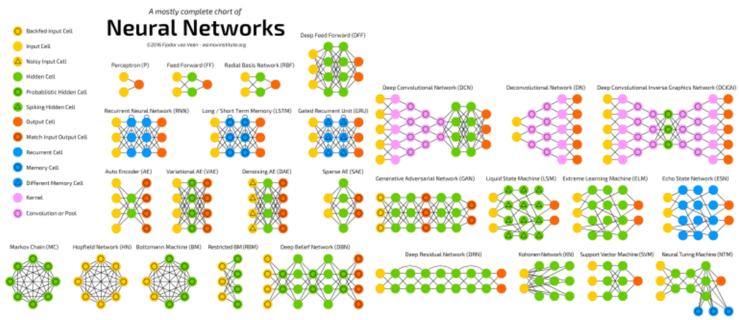
- **Learning rule**: The *learning rule* is a rule or an algorithm which modifies the parameters of the neural network, in order for a given input to the network to produce a favored output.
- This *learning* process typically amounts to modifying the weights and thresholds.



TYPES OF NEURAL NETWORKS

TYPES OF NEURAL NETWORKS

- There are many classes of neural networks and these classes also have sub-classes.



Source: http://www.asimovinstitute.org/neural-network-zoo/

TYPES OF NEURAL NETWORKS

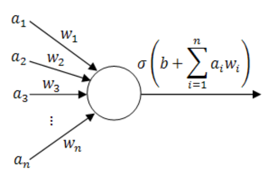
- Here, we will list the most used ones and make things simple to move on in this journey to learn neural networks.
  1. Feedforward Neural Network
     1.1. Single-layer Perceptron
     1.2. Multi-layer Perceptron (MLP)
     1.3. Convolutional Neural Network (CNN)
  2. Recurrent neural networks

FEEDFORWARD NEURAL NETWORK

- A feedforward neural network is an artificial neural network where connections between the units do *not* form a cycle.
- In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.
- We can distinguish two types of feedforward neural networks:
  - Single-layer Perceptron
  - Multi-layer Perceptron (MLP)

SINGLE-LAYER PERCEPTRON

- This is the simplest feedforward neural Network and does not contain any hidden layer, which means it only consists of a single layer of output nodes.
- This is said to be single because when we count the layers we do not include the input layer, the reason for that is because at the input layer no computations is done, the inputs are fed directly to the outputs via a series of weights.



Simple Perceptron

SINGLE-LAYER PERCEPTRON ALGORITHM

1. Inisialisasi semua bobot dan bias (umumnya nilai awal $w_i = b = 0$)
2. Selama ada elemen vektor masukan yang respon unit keluarannya tidak sama dengan target, lakukan:

2.1 Set aktivasi unit masukan $x_i = s_i$, dimana $i = 1, \dots, n$

2.2 Hitung respon unit keluaran: $net = \sum_i x_i w_i + b$

$$f(net) = \begin{cases} 1 & \text{if } net > \theta \\ 0 & \text{if} -\theta \le net \le \theta \\ -1 & \text{if } net < -\theta \end{cases}$$

2.3 Perbaiki bobot pola yang mengadung kesalahan menurut persamaan:

$w_i(baru) = w_i(lama) + \Delta w$ *dimana* $(i = 1, \dots, n)$ dengan $\Delta w = \alpha t x_i$

$b(baru) = b(lama) + \Delta b$ *dengan* $\Delta b = \alpha t$

Dimana: α = Laju pembelajaran (learning rate) yang ditentukan

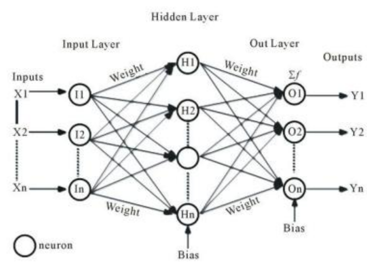
ɵ = Threshold yang ditentukan

t = Target

2.4 Ulangi iterasi sampai perubahan bobot ($\Delta W_n = 0$) tidak ada

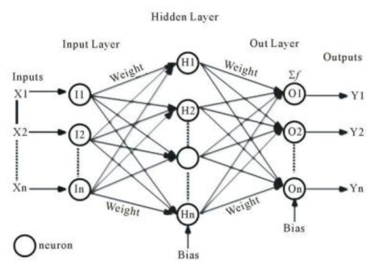
MULTI-LAYER PERCEPTRON

- This class of networks consists of multiple layers of computational units, usually interconnected in a feed-forward way.
- Each neuron in one layer has directed connections to the neurons of the subsequent layer.



MULTI-LAYER PERCEPTRON

- In many applications, the units of these networks apply a sigmoid function as an activation function.
- MLP are very more useful and one good reason is that, they are able to learn non-linear representations (most of the cases the data presented to us is not linearly separable).

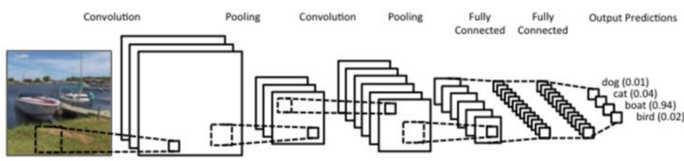


CONVOLUTIONAL NEURAL NETWORK (CNN)

- Convolutional Neural Networks are very similar to ordinary Neural Networks, they are made up of neurons that have learnable weights and biases.
- In convolutional neural network (CNN, or ConvNet or shift invariant or space invariant) the unit connectivity pattern is inspired by the organization of the visual cortex.
- Units respond to stimuli in a restricted region of space known as the receptive field.
- Receptive fields partially overlap, over-covering the entire visual field. Unit response can be approximated mathematically by a convolution operation.

CONVOLUTIONAL NEURAL NETWORK (CNN)

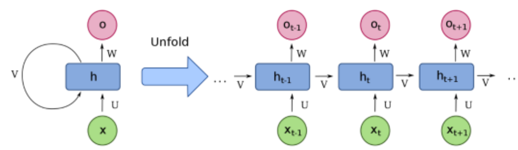
- They are variations of multilayer perceptrons that use minimal preprocessing.
- Their wide applications is in image and video recognition, recommender systems and natural language processing.
- CNNs requires large data to train on.



CNN for image classification

RECURRENT NEURAL NETWORKS

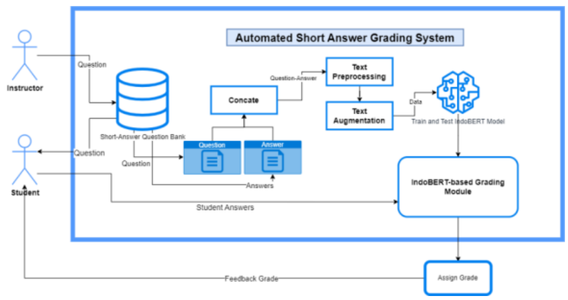
- In recurrent neural network (RNN), connections between units form a directed cycle (they propagate data forward, but also backwards, from later processing stages to earlier stages).



- This allows it to exhibit dynamic temporal behavior. Unlike feedforward neural networks, RNNs can use their internal memory to process arbitrary sequences of inputs.
- This makes them applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and other general sequence processors.

APPLICATION OF DEEP LEARNING

AUTOMATIC SHORT ANSWER GRADING (ASAG)



- ASAG is a scoring system where specific short answers in free text form can be assessed automatically.
- IndoBERT is a state-of-the-art deep learning architecture specifically designed for Indonesian language processing.

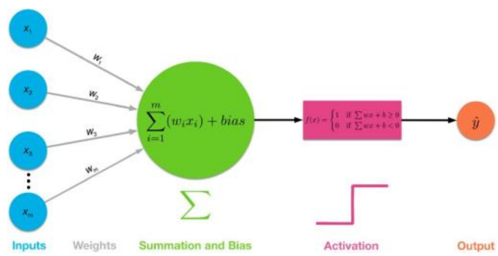
- [Situmeang *et al.*] proposed an IndoBERT-based ASAG model for regression analysis.

Samuel Indra Gunawan Situmeang, Raja Muda Gading Tulen Sihite, Humasak Simanjuntak, and Junita Amalia. 2023. A Deep Learning-Based Regression Approach to Indonesian Short Answer Grading System. In Proceedings of the 8th International Conference on Sustainable Information Engineering and Technology (SIET '23). Association for Computing Machinery, New York, NY, USA, 201–209. https://doi.org/10.1145/3626641.3626929

COMMONLY USED ACTIVATION FUNCTIONS

COMMONLY USED ACTIVATION FUNCTIONS

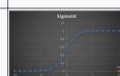
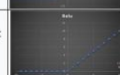
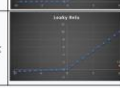
- ***Activation functions** also known as transfer function is used to map input nodes to output nodes in certain fashion.*
- Every activation function (or *non-linearity*) takes a single number and performs a certain fixed mathematical operation on it.



Source: https://towardsdatascience.com/

COMMONLY USED ACTIVATION FUNCTIONS

- There are many activation functions used in Machine Learning.
- Here are some activations functions you will often find in practice:

Name	Plot	Equation	Derivative
Sigmoid		$f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh		$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$	$f'(x) = 1 - f(x)^2$
Rectified Linear Unit (relu)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Leaky Rectified Linear Unit (Leaky relu)		$f(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0.01 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$

Source: https://engmrk.com/

TRAINING AND TESTING

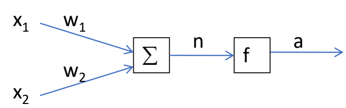
EXAMPLE

- Buat jaringan Perceptron untuk menyatakan fungsi logika AND dengan menggunakan masukan biner dan keluaran bipolar. Pilih $\alpha = 1$ dan $\theta = 0.2$

EXAMPLE: TRAINING

- Pola hubungan masukan-target :

x_1	x_2	t
0	0	-1
0	1	-1
1	0	-1
1	1	1



EXAMPLE: TRAINING

Masukan			Target		Output	Perubahan bobot $\Delta w = \alpha x_i t$ $\Delta b = \alpha t$			Bobot baru $w_{baru} = w_{lama} + \Delta w$ $b_{baru} = b_{lama} + \Delta b$		
x_1	x_2	1	t	n	a=f(n)	Δw_1	Δw_2	Δb	w_1	w_2	b
Epoch ke - 1									0	0	0
0	0	1	-1								
0	1	1	-1								
1	0	1	-1								
1	1	1	1								

istilah *epoch* digunakan karena ketika melakukan satu kali iterasi dilakukan dengan rambatan balik. misalnya satu iterasi melibatkan proses a-b-c-d, maka satu *epoch* melibatkan a-b-c-d-c-b-a.

EXAMPLE: TRAINING

Masukan			Target		Output	Perubahan bobot $\Delta w = \alpha x_i t \quad \Delta b = \alpha t$			Bobot baru $w_{baru} = w_{lama} + \Delta w$ $b_{baru} = b_{lama} + \Delta b$		
x_1	x_2	1	t	n	a=f(n)	Δw_1	Δw_2	Δb	w_1	w_2	b
Epoch ke - *											
0	0	1	-1								
0	1	1	-1								
1	0	1	-1								
1	1	1	1								

EXAMPLE: TRAINING

$$f(net) = \begin{cases} 1 & \text{if } net > \theta \\ 0 & \text{if } -\theta \le net \le \theta \\ -1 & \text{if } net < -\theta \end{cases}$$

$$net = \sum_i x_i w_i + b$$

$$\Delta b = \alpha t$$

	X1	X2	T	y_in	fnet	Δw1	Δw2	Δb	w1	w2	b
Epoch 1	1	1	1	0	0	1	1	1	1	1	1
	1	0	-1	2	1	-1	0	-1	0	1	0
	0	1	-1	1	1	0	-1	-1	0	0	-1
	0	0	-1	-1	-1	0	0	0	0	0	-1
	1	1	1	-1	-1	1	1	1	1	1	0
Epoch 2	1	0	-1	1	1	-1	0	-1	0	1	-1
	0	1	-1	0	0	0	-1	-1	0	0	-2
	0	0	-1	-2	-1	0	0	0	0	0	-2
	1	1	1	-2	-1	1	1	1	1	1	-1
Epoch 3	1	0	-1	0	0	-1	0	-1	0	1	-2
	0	1	-1	-1	-1	0	0	0	0	1	-2
	0	0	-1	-2	-1	0	0	0	0	1	-2
Epoch 4	1	1	1	-1	-1	1	1	1	1	2	-1
	1	0	-1	0	0	-1	0	-1	0	2	-2
	0	1	-1	0	0	0	-1	-1	0	1	-3
	0	0	-1	-3	-1	0	0	0	0	1	-3
	1	1	1	-2	-1	1	1	1	1	2	-2
Epoch 5	1	0	-1	-1	-1	0	0	0	1	2	-2
	0	1	-1	0	0	0	-1	-1	1	1	-3
	0	0	-1	-3	-1	0	0	0	1	1	-3
	1	1	1	-1	-1	1	1	1	2	2	-2

$$\Delta w = \alpha t x_i$$

$$w_i(baru) = w_i(lama) + \Delta w$$

$$b(baru) = b(lama) + \Delta b$$

EXAMPLE: TRAINING

	1	1	1	-1	-1	1	1	1	2	2	-2
Epoch 6	1	0	-1	0	0	-1	0	-1	1	2	-3
	0	1	-1	-1	-1	0	0	0	1	2	-3
	0	0	-1	-3	0	0	0	0	1	2	-3
	1	1	1	0	0	1	1	1	2	3	-2
	1	0	-1	0	0	-1	0	-1	1	3	-3
Epoch 7	0	1	-1	0	0	0	-1	-1	1	2	-4
	0	0	-1	-4	-1	0	0	0	1	2	-4
	1	1	1	-1	-1	1	1	1	2	3	-3
	1	0	-1	-1	-1	0	0	0	2	3	-3
Epoch 8	0	1	-1	0	1	0	-1	-1	2	2	-4
	0	0	-1	-4	-1	0	0	0	2	2	-4
	1	1	1	0	0	1	1	1	3	3	-3
	1	0	-1	0	0	-1	0	-1	2	3	-4
Epoch 9	0	1	-1	-1	-1	0	0	0	2	3	-4
	0	0	-1	-4	-1	0	0	0	2	3	-4
	1	1	1	2	1	0	0	0	2	3	-4
	1	0	-1	-2	-1	0	0	0	2	3	-4
Epoch 10	0	1	-1	-1	-1	0	0	0	2	3	-4
	0	0	-1	-4	-1	0	0	0	2	3	-4

Iterasi dihentikan pada epoch ke-10 karena $f(net)$ sudah sama dengan target nya

EXAMPLE: TESTING

- Model yang didapatkan dari training:

$$net = 2x_1 + 3x_2 - 4$$

x_1	x_2	t
0	0	-1
0	1	-1
1	0	-1
1	1	1

- Jika dilakukan testing dengan menggunakan $x_1 = 1$ dan $x_2 = 0$ (sebagai contoh, digunakan data yang sama dengan data training), maka $net = 2 * 1 + 3 * 0 - 4 = -2$.

- $y = \text{sign}(net) = \text{sign}(-2) = -1$.

SUMMARY

- Artificial Neural Network is a computational model that is inspired by the way biological neural networks in the human brain process information.

REFERENCES

- S. J. Russell and P. Borvig, *Artificial Intelligence: A Modern Approach (4th Edition)*, Prentice Hall International, 2020.
  - Chapter 19. Learning from Examples
- J. Han and M. Kamber, "*Data Mining: Concepts and Techniques (3rd Edition)*," Elsevier, 2012.
- Neural Networks and Deep Learning, http://neuralnetworksanddeeplearning.com/index.html (Accessed on November 27th, 2018).
- CS231n Convolutional Neural Networks for Visual Recognition, http://cs231n.github.io/neural-networks-1/, (Accessed on November 27th, 2018).
- A Basic Introduction To Neural Networks, http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html (Accessed on November 27th, 2018).
- A Gentle Introduction To Neural Networks Series — Part 1, https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc (Accessed on November 27th, 2018).

eof

Faculty of Informatics and Electrical Engineering