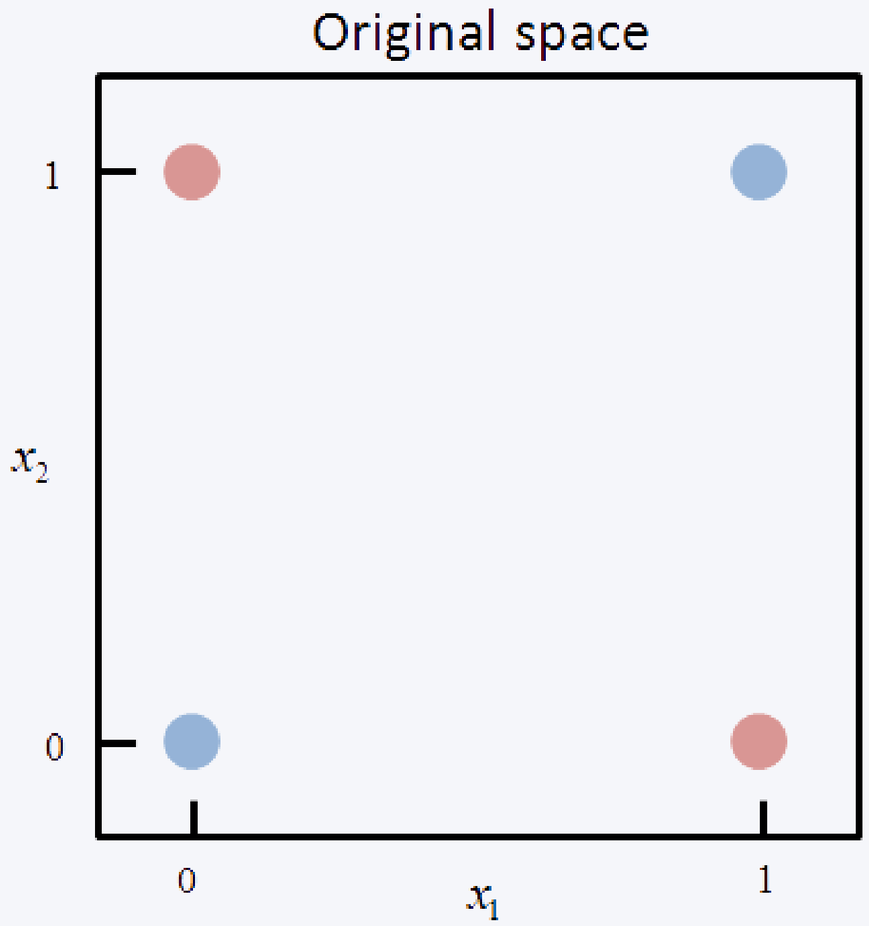
XOR-Problem

XOR-PROBLEM

أكاديمية طويق
Tuwaiq Academy

Example problem on a single neuron can not solve but a single hidden layer net can



Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

XOR-PROBLEM

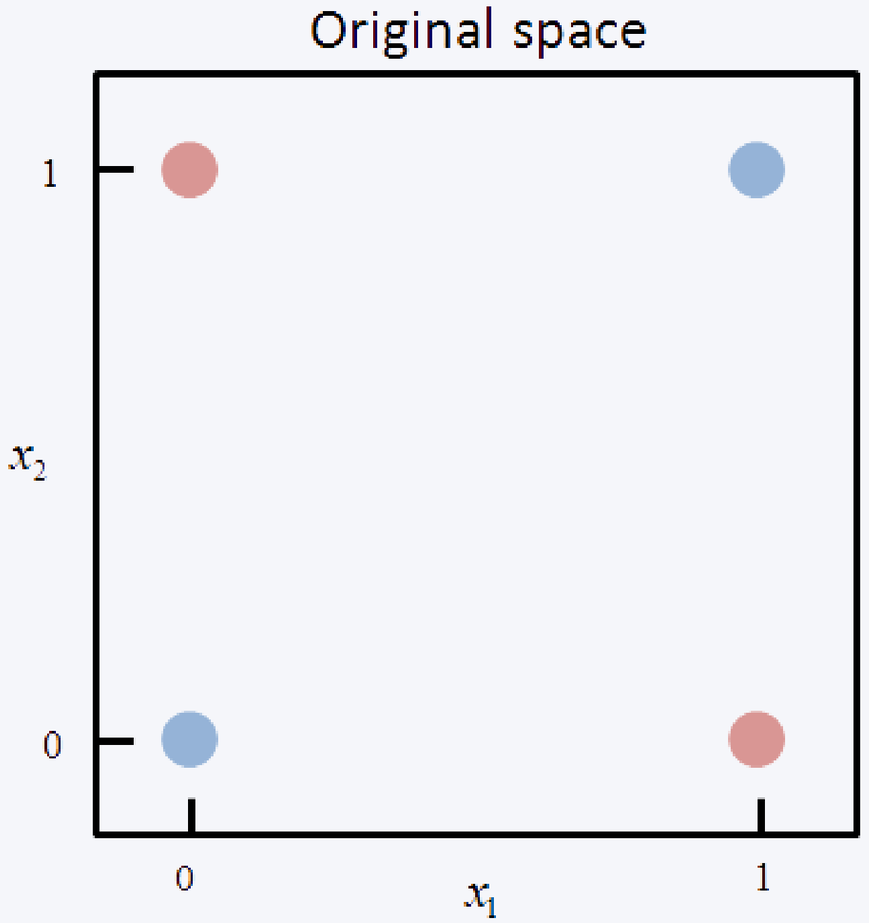
أكاديمية طويق
Tuwaiq Academy

Suppose we have four data points:

$$X = \{(0, 0)>, (0, 1)>, (1, 0)>, (1, 1)>\}$$

The XOR gate (exclusive or) returns true, when an odd number of inputs are true:

x_1	x_2	XOR = y
0	0	0
0	1	1
1	0	1
1	1	0



Can you learn the target function with a logistic regression model?

Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

XOR-PROBLEM

أكاديمية طويق
Tuwaiq Academy

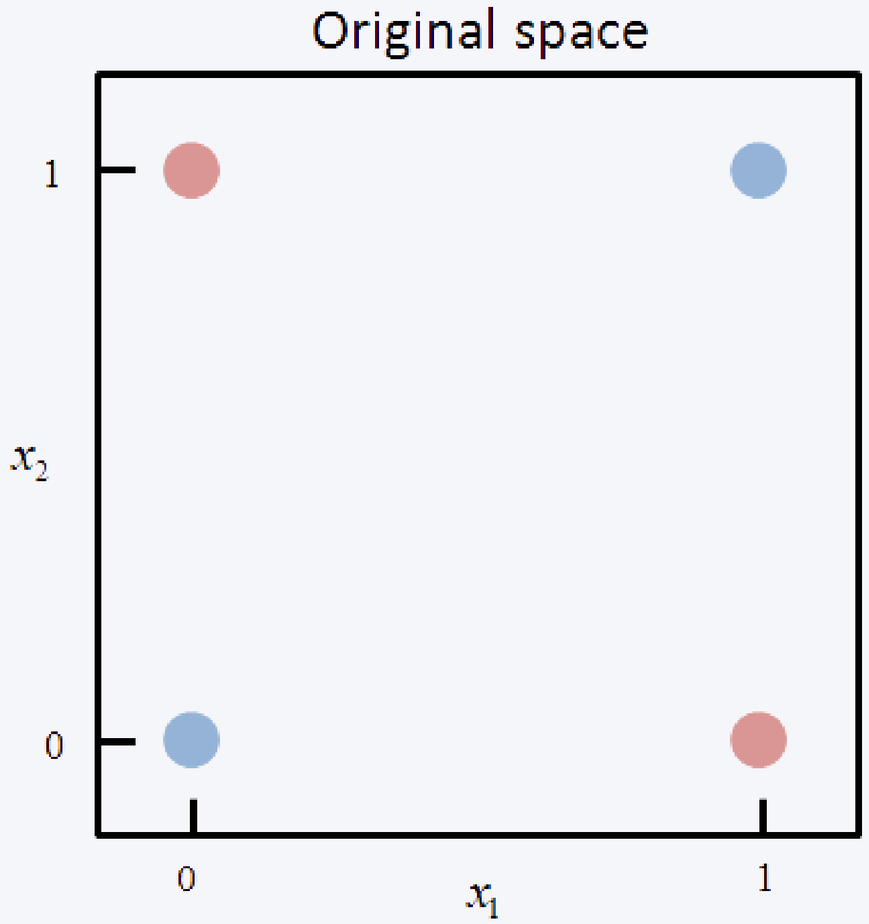
Answer : CAN’T , Why ?

XOR Problem Basics:

- XOR (Exclusive OR) outputs 1 when inputs are different (e.g., 0,1 or 1,0) and 0 when inputs are the same (e.g., 0,0 or 1,1).
- XOR is **not linearly separable**, meaning a straight line cannot separate the output classes.

Why Logistic Regression Fails:

- Logistic regression works by finding a straight line (or hyperplane) to separate data.
- Since XOR data is not linearly separable, logistic regression cannot classify it correctly.



Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

XOR-PROBLEM

أكاديمية طويق
Tuwaiq Academy

Consider the following model:

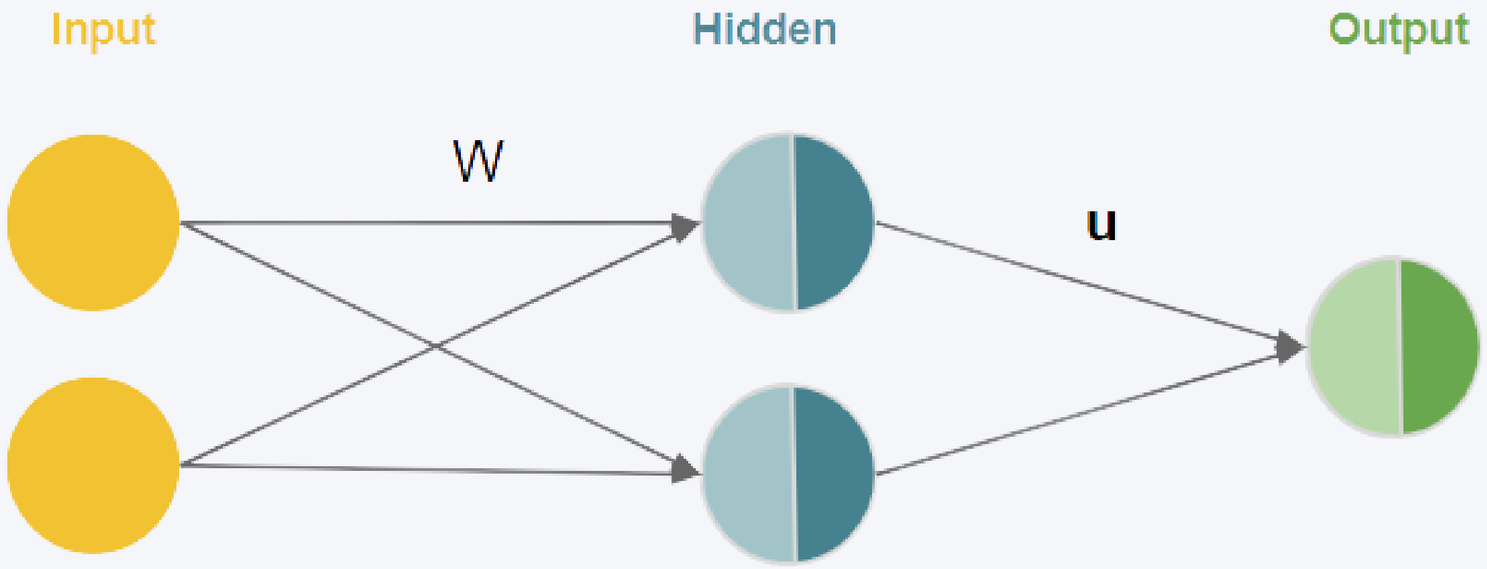


Figure: A neural network with two neurons in the hidden layer.
The matrix W describes the mapping from x to z. The vector u from z to y.

Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

XOR-PROBLEM

أكاديمية طويق
Tuwaiq Academy

Let’s solve XOR using a perceptron model with ReLU and step activation functions.

ReLU Activation Function:

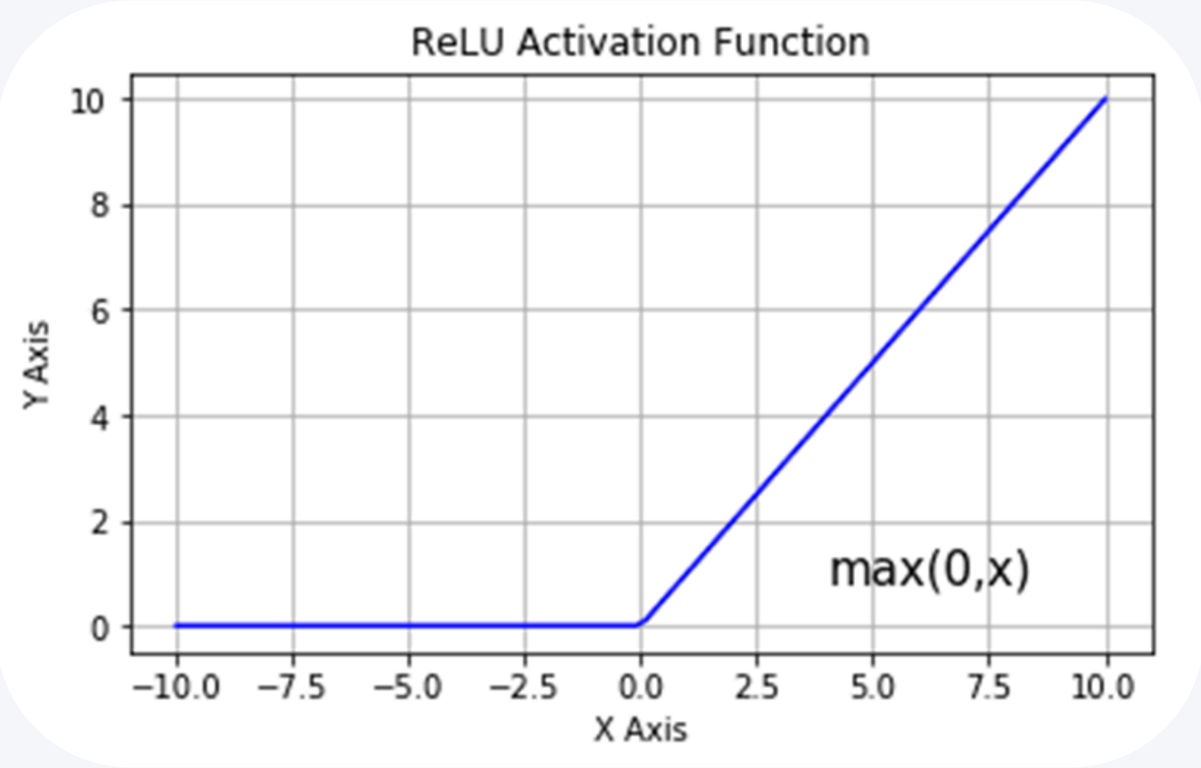
$$\sigma(z) = \max(0, z)$$

Outputs 0 if $z \leq 0$, otherwise outputs z.

Thresholding (Step Function):

$$\tau(z) = [z > 0]$$

Outputs 1 if $z > 0$, otherwise outputs 0.



Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

XOR-PROBLEM

أكاديمية طويق
Tuwaiq Academy

Let’s solve XOR using a perceptron model with ReLU and step activation functions.

Architecture of the Model:

Model Equation:

$$f(x|W,b,u,c) = \tau(u^\top \sigma(Wx+b)+c)$$

Input Layer:

- The input, x, is a 2D vector representing the XOR input values (x1,x2).

Weights (W):

- A weight matrix (W) is applied to the inputs to transform them into a new space.
- This transformation helps make the data linearly separable.

Bias (b):

- A bias vector is added to the weighted input to shift the transformation, improving flexibility.

ReLU Activation (σ):

- The ReLU function. σ(z)=max(0,z) is applied to the result, introducing non-linearity.
- Non-linearity is crucial for solving non-linear problems like XOR.

Output Layer:

- The transformed values are further processed using: **A weight vector (u) and bias (c).**
- The final result is passed through a thresholding function **(τ)** that outputs 0 or 1.

Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

XOR-PROBLEM

أكاديمية طويق
Tuwaiq Academy

Let’s solve XOR using a perceptron model with ReLU and step activation functions.

Input and Weight Matrices

- Input Matrix (X):
  - Rows represent data points:

$$X = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$$

- Weight Matrix (W):
  - Hidden layer transformation:

$$W = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix}$$

Bias and Transformation

- Bias Vector (b):

$$b = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

- Transformed Space ($Z = Wx + b$):
  - After applying weights and biases:

$$Z = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 2 \\ 2 & 1 \end{bmatrix}$$

Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

XOR-PROBLEM

أكاديمية طويق
Tuwaiq Academy

Let’s solve XOR using a perceptron model with ReLU and step activation functions.

Applying ReLU and Step Functions

- ReLU Transformation:

$$Z = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 2 \\ 2 & 1 \end{bmatrix}$$

$$\sigma(Z) = \max(0, Z)$$

  - Result:

$$\sigma(Z) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 2 \\ 2 & 1 \end{bmatrix}$$

- Step Function:

$$\tau(\sigma(Z)) = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

  - XOR is now perfectly classified!

Final Output Transformation

- Final Calculation:
  - Multiply activated values by $u = \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}$ and add bias $c = 0.5$.
  - Output:

$$f(x) = [u^\top \sigma(Z) + c]$$

Result:

$$f(x) = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

XOR-PROBLEM

أكاديمية طويق
Tuwaiq Academy

we learned:

- Non-Linearity is crucial for solving non-linear problems like XOR.

- Neural networks use layers and activation functions to:
  - Transform data into separable spaces.
  - Solve problems beyond the capability of linear models.

Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

أكاديمية طويق
Tuwaiq Academy

Let’s Code

Implementing XOR Problem with Keras

Single Hidden Layer NN

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Graphical Representation of Models:

- Simple functions/models (e.g., logistic regression) can be visually represented as graphs.
- This helps break down complex concepts into understandable pieces.

Building Blocks for Complex Functions:

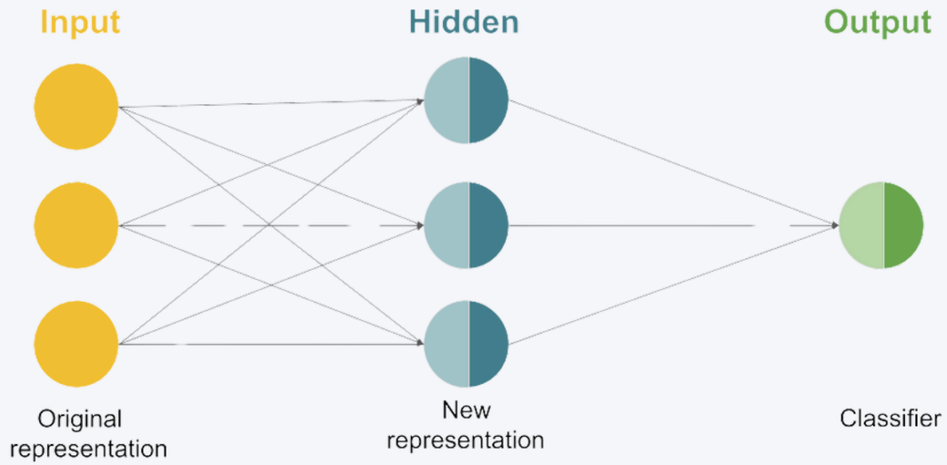
- Individual neurons act as basic units or "building blocks."
- Combining neurons creates networks capable of handling complex tasks.

Representation of Complex Hypotheses:

- Neural networks can model highly intricate patterns and relationships in data.
- They provide access to vast "hypothesis spaces" for learning.

Learning Common Patterns Efficiently:

- Neural networks enable us to focus on hypothesis spaces aligned with real-world data.
- This data-efficient learning approach mirrors common patterns found in nature and our universe.
- Example: Recognizing handwritten digits or identifying objects in images.

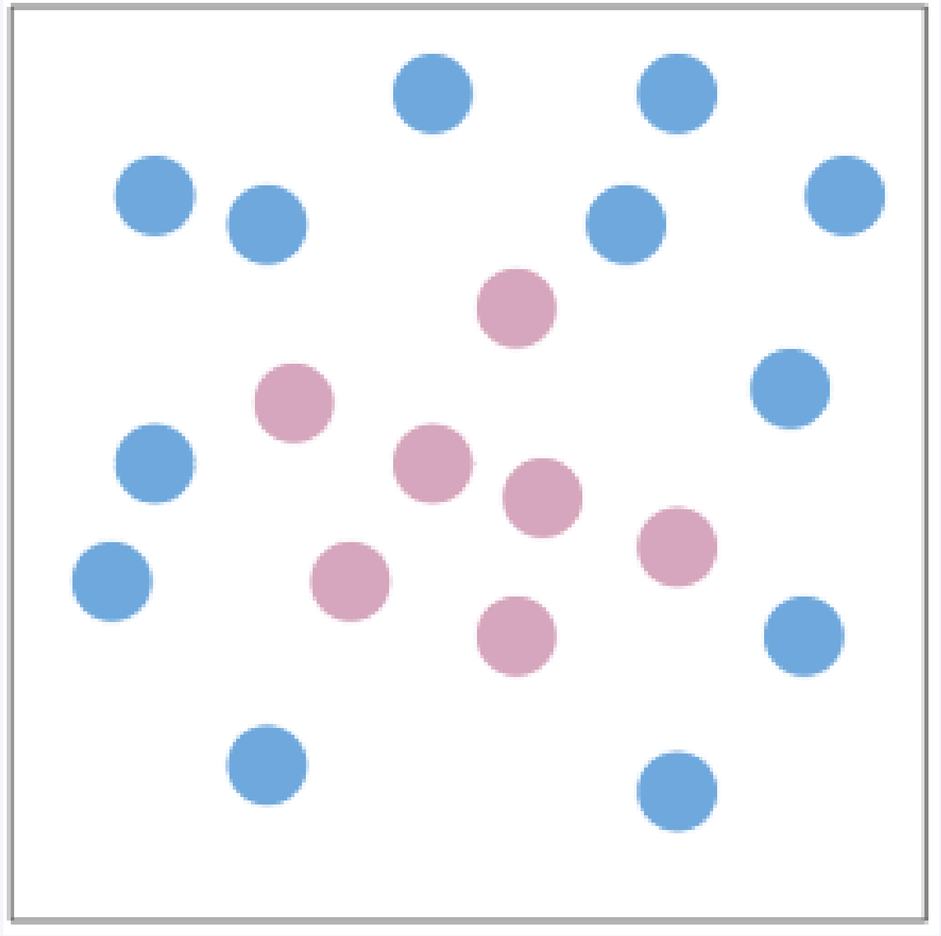


Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Can a single neuron perform binary classification of these points?



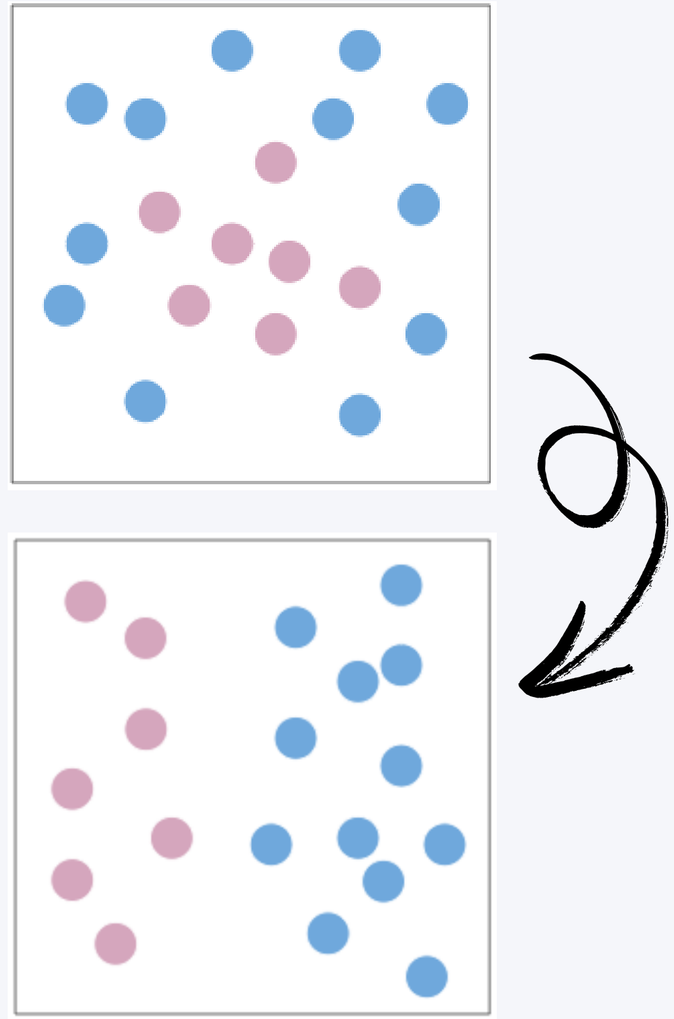
Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Can a single neuron perform binary classification of these points?

As a **single neuron** is restricted to learning **only linear decision boundaries**, its performance on the following task is quite poor



However, the **neuron can easily separate the classes if the original features are transformed** (e.g., from Cartesian to polar coordinates):

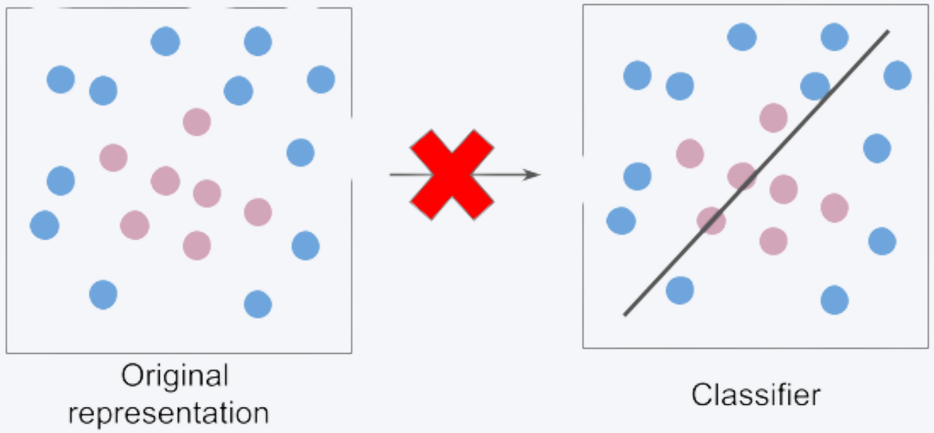
Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Can a single neuron perform binary classification of these points?

Instead of classifying the data in the original representation,



we classify it in a new feature space.

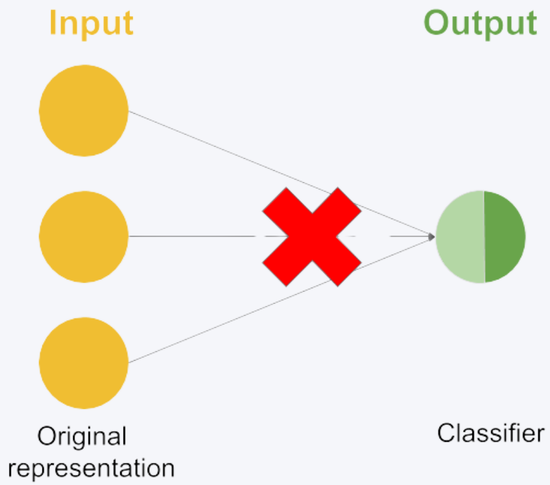
Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

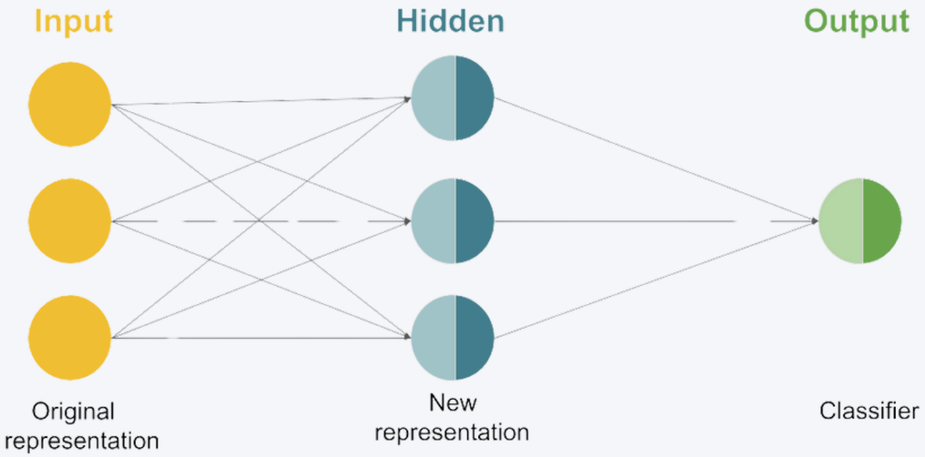
أكاديمية طويق
Tuwaiq Academy

Can a single neuron perform binary classification of these points?

So, Instead of a single neuron,



we use more complex networks.



Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

What is Feature Engineering?

- Before **deep learning**, domain experts manually designed features for tasks like image recognition and speech processing.

- This process is called feature engineering and is crucial for model performance.

Deep Learning Automates This Process:

- Instead of manually designing features, deep learning models learn features automatically from raw data.

- This automation is a core strength of deep learning.

Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Why is this Important?

- Feeding a classifier the "right" features is critical for success.
- Deep learning eliminates the dependency on manual expertise, enabling models to find the most effective representation on their own.

Examples:

- In image recognition, instead of manually extracting edges or textures, deep learning discovers these patterns in layers.
- In speech recognition, it learns phonemes and patterns directly from audio data without requiring hand-crafted features.

Key Insight:

- Deep learning replaces traditional feature engineering with automated representation learning, improving efficiency and adaptability.

Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

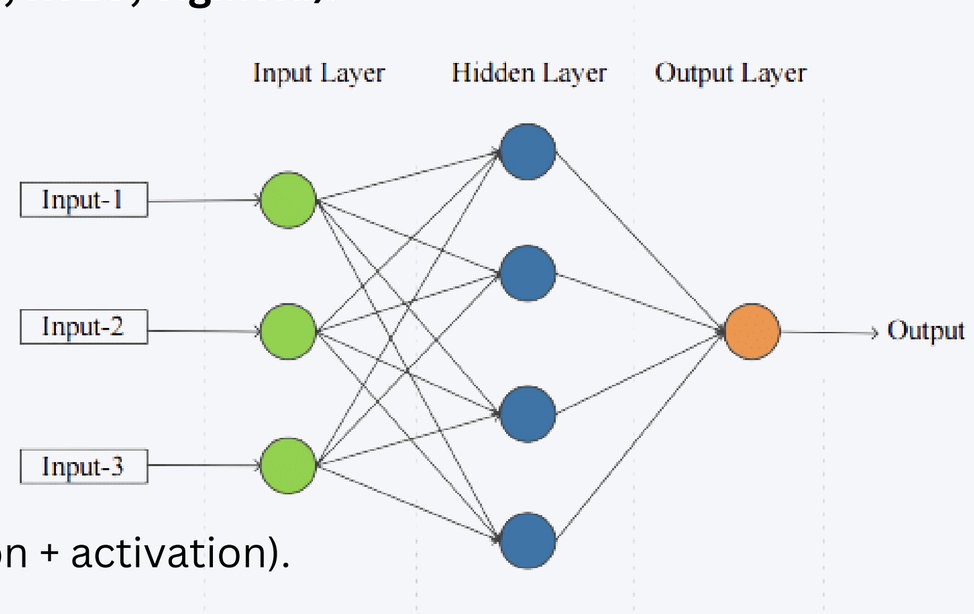
SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Single Hidden Layer Networks

Single Neurons Perform Two Steps:

- **Affine Transformation: Compute a weighted sum of inputs plus a bias.**
- **Activation: Apply a non-linear transformation to the weighted sum (e.g., ReLU, Sigmoid).**



Structure of a Single Hidden Layer Network:

- **Hidden Layer:**
  - Contains a set of neurons that process the input.
  - Each neuron performs the 2-step computation (affine transformation + activation).
- **Output Layer:**
  - Consists of one or more neurons that produce the network's final result.
  - These neurons also perform the same 2-step computation.

Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

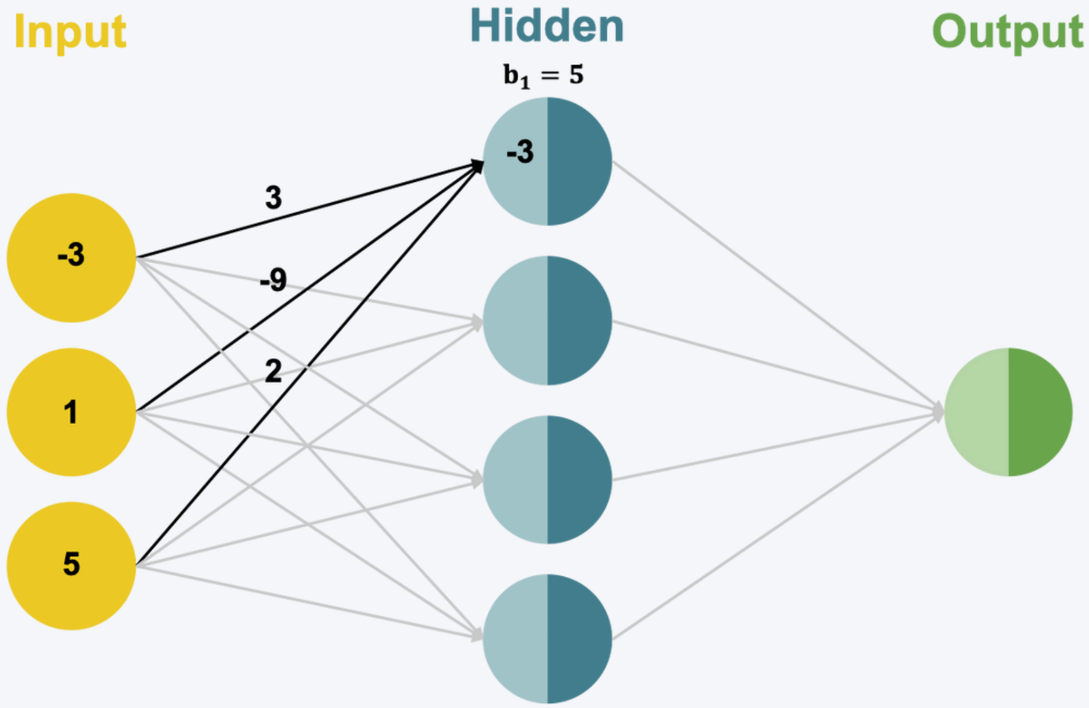
SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

SINGLE HIDDEN LAYER NETWORKS: EXAMPLE

Graphical Representation of Models:

As a single neuron is restricted to learning only linear decision boundaries, its performance on the following task is quite poor:



Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Example

Input to Hidden Layer:

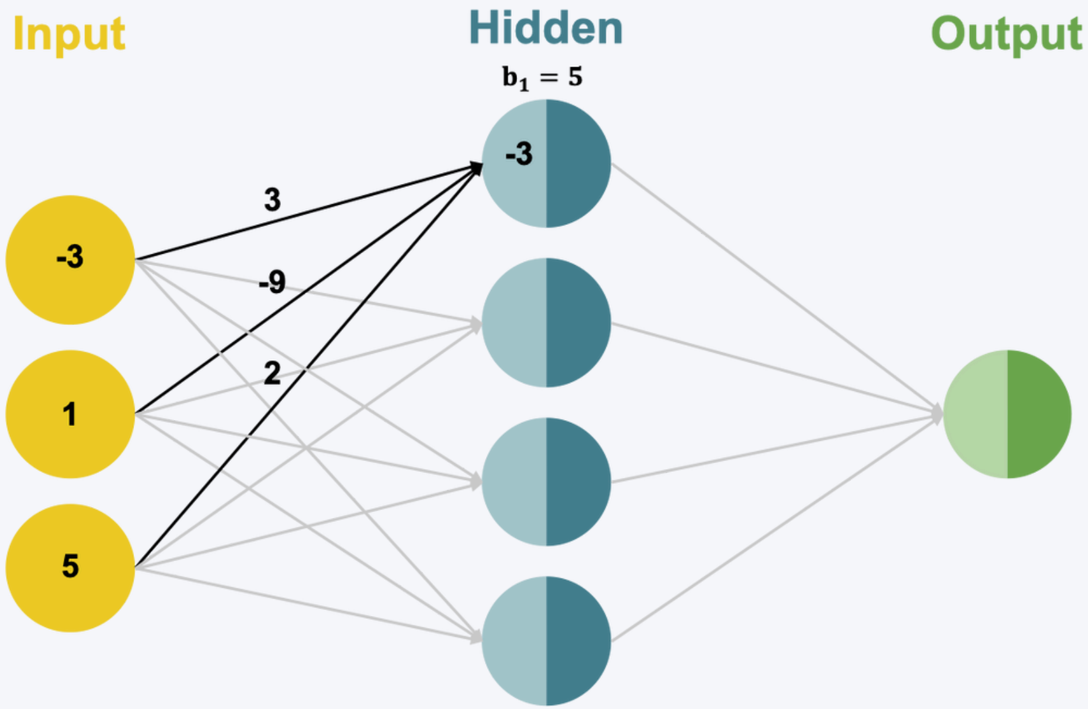
- Each input is multiplied by its respective weight (w), and a bias (b) is added.
- This operation is called an affine transformation:

$$z = w_{1,1}x_1 + w_{2,1}x_2 + w_{3,1}x_3 + b_1$$

Hidden Neuron Example (Calculations):

- **For one hidden neuron:**
  - Input values: $x_1 = -3, x_2 = 1, x_3 = 5$
  - Weights: $w_{1,1} = 3, w_{2,1} = -9, w_{3,1} = 2$
  - Bias: $b_1 = 5$

$$z = (3 \cdot -3) + (-9 \cdot 1) + (2 \cdot 5) + 5 = -9 - 9 + 10 + 5 = -3$$



Activation Function:

- After the affine transformation, **a non-linear activation function** (e.g., ReLU or Sigmoid) is applied to z.
- This introduces non-linearity to the network.

Output Layer:

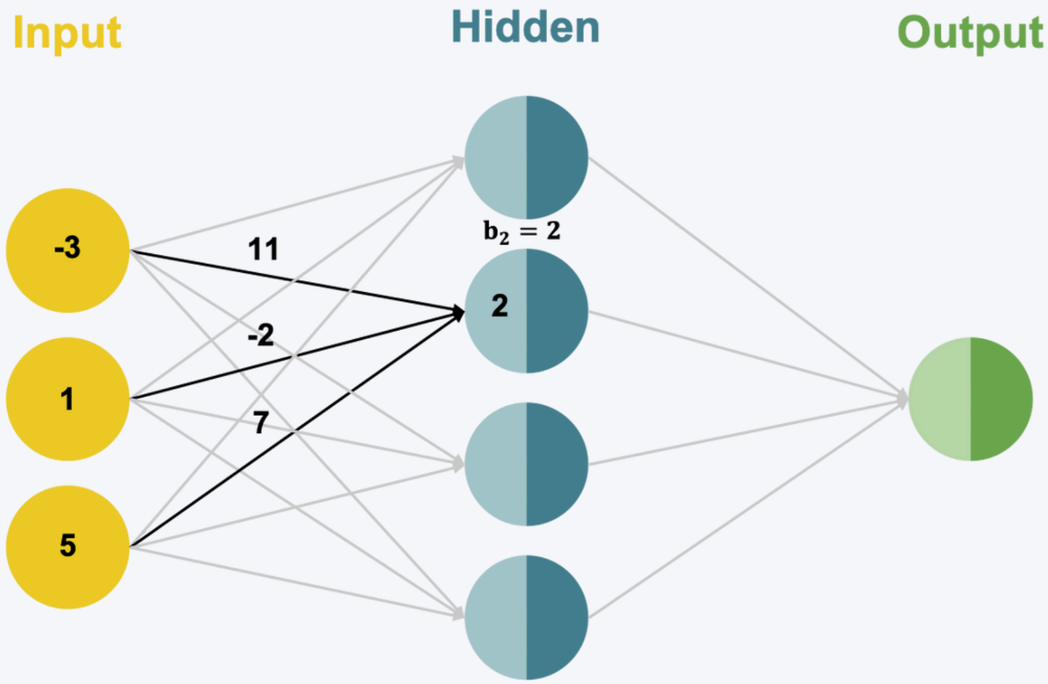
- Outputs from the hidden layer neurons are passed to the output layer, where a similar 2-step computation **(affine transformation + activation)** produces the final output.

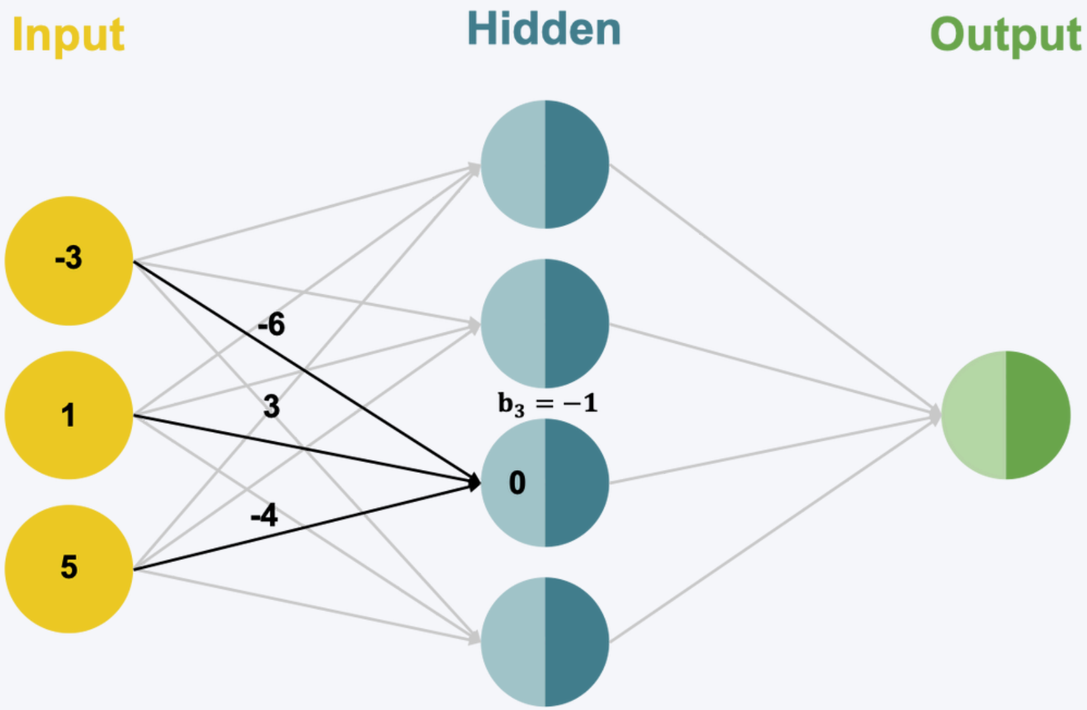
Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Example: Try to do it !



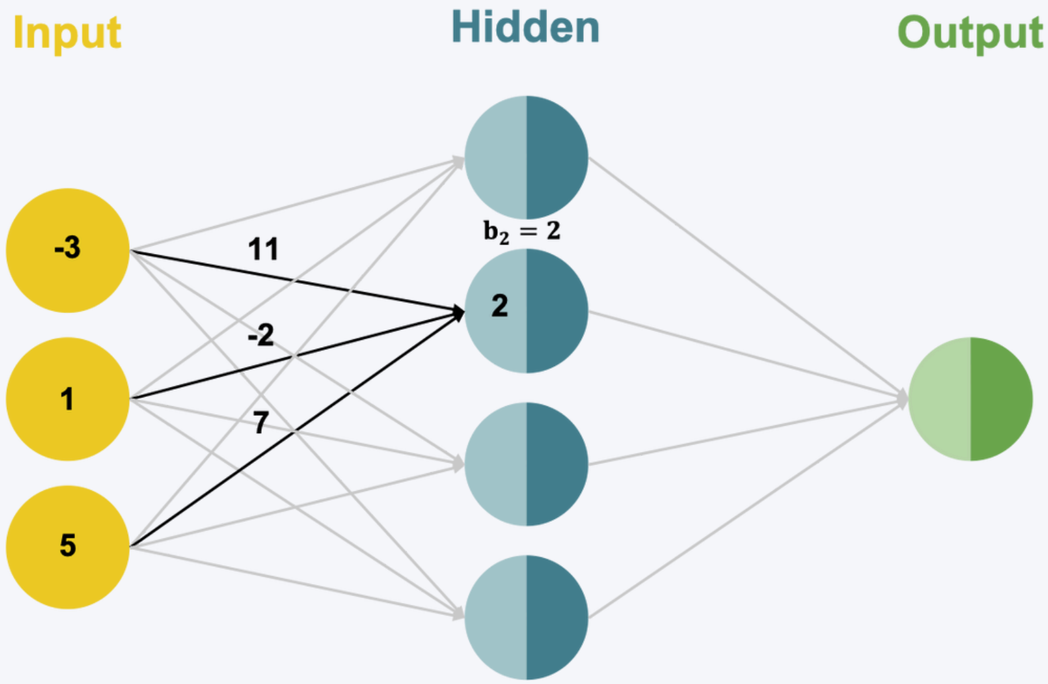


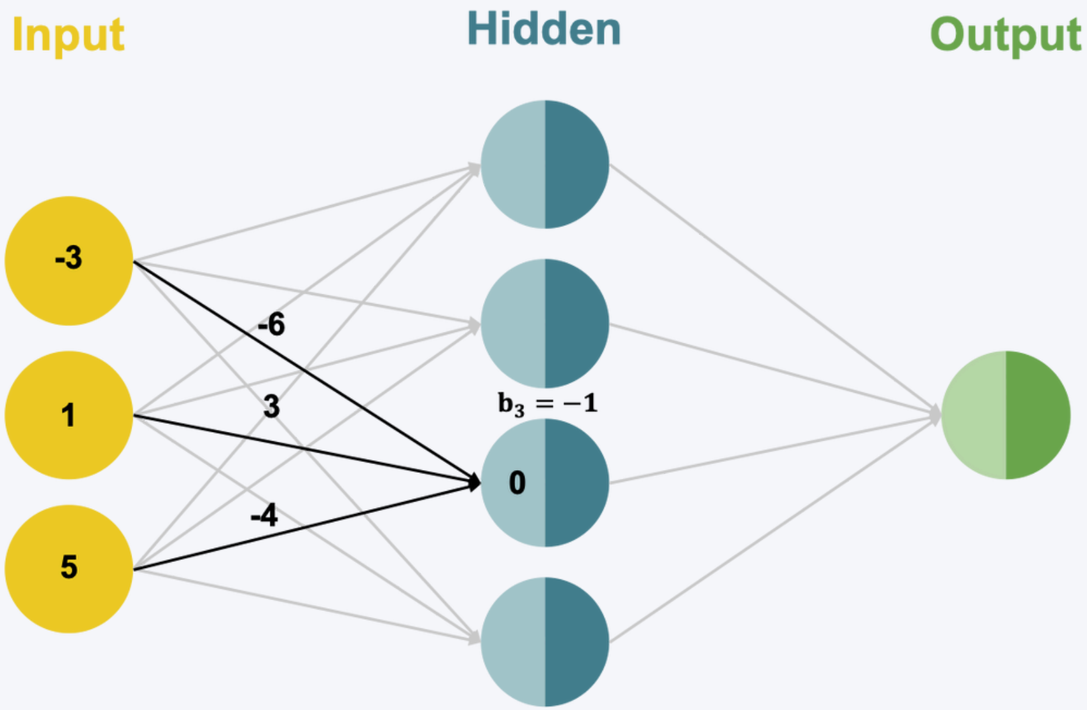
Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Example: Try to do it !



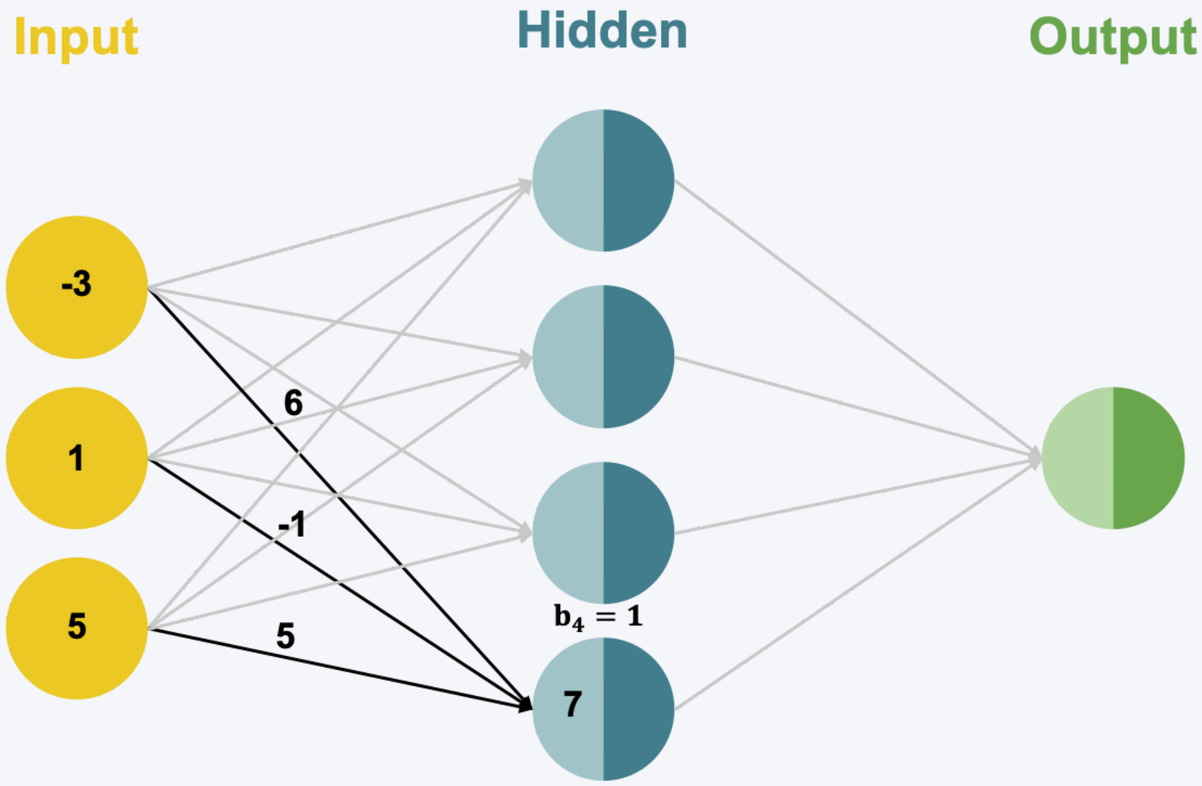


Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Example: Try to do it !

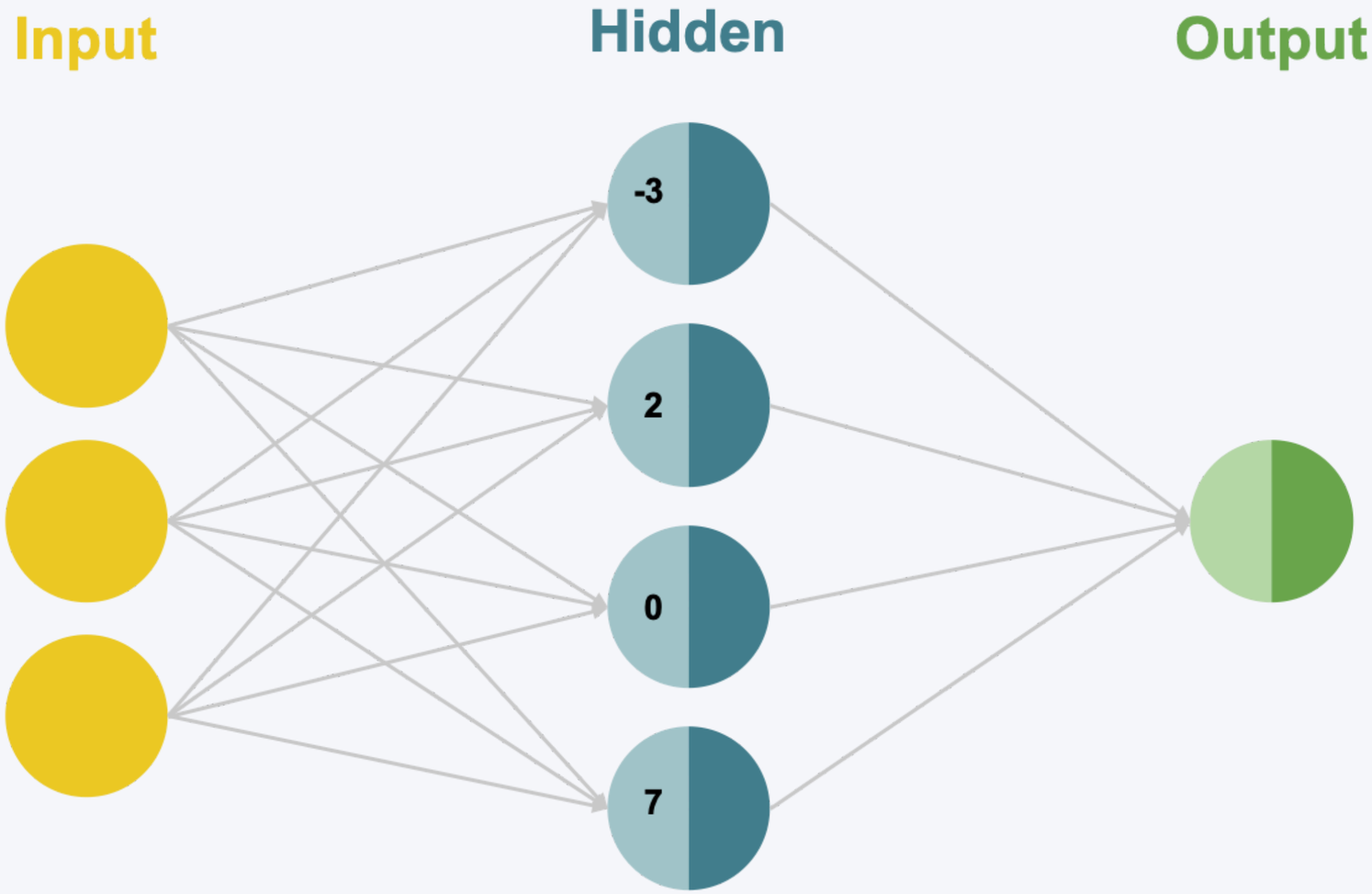


Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Example: We End Up With:

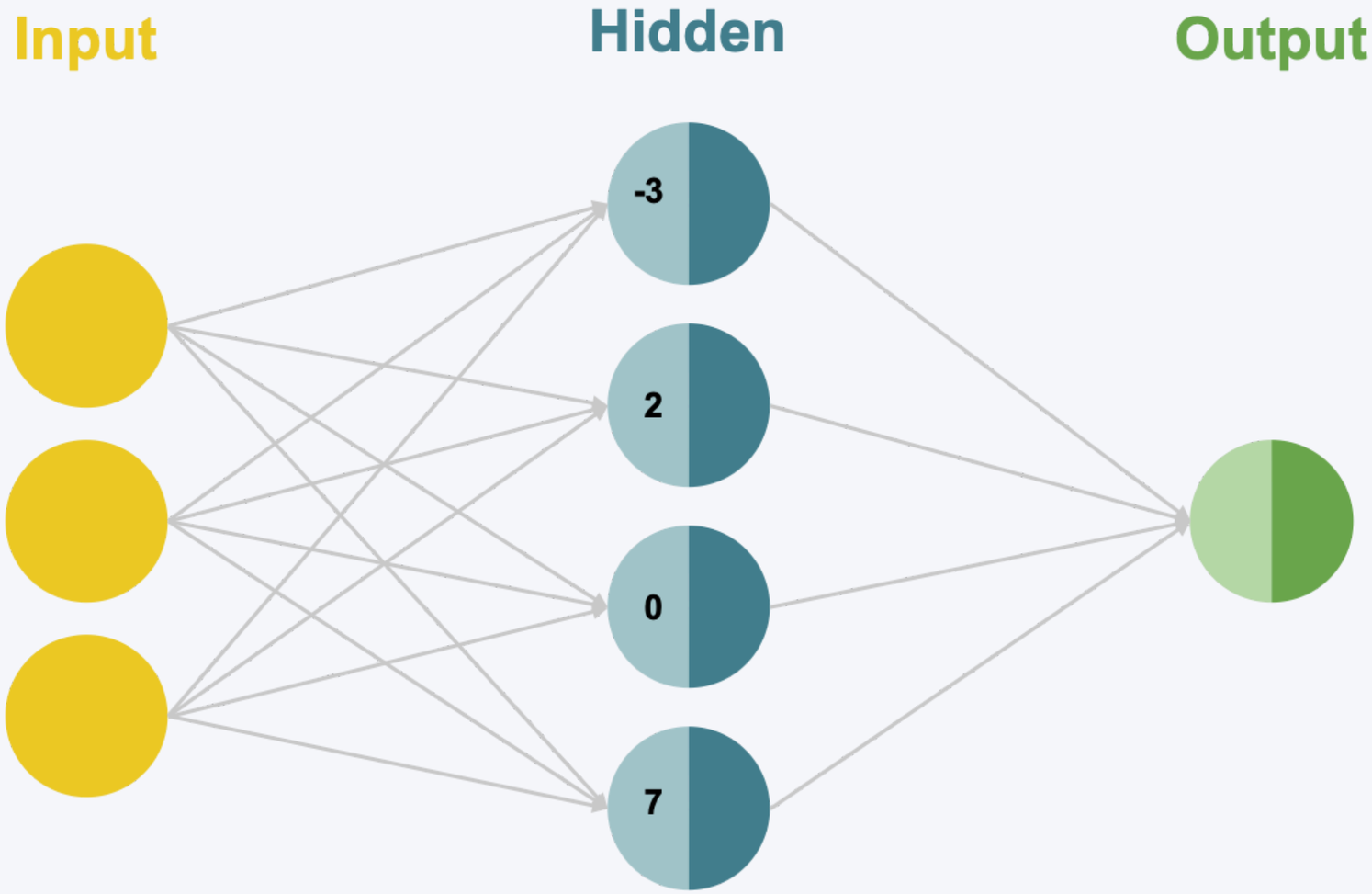


Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Example: We End Up With:



Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

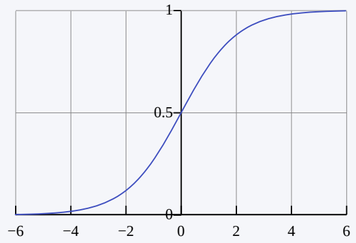
SINGLE HIDDEN LAYER NN

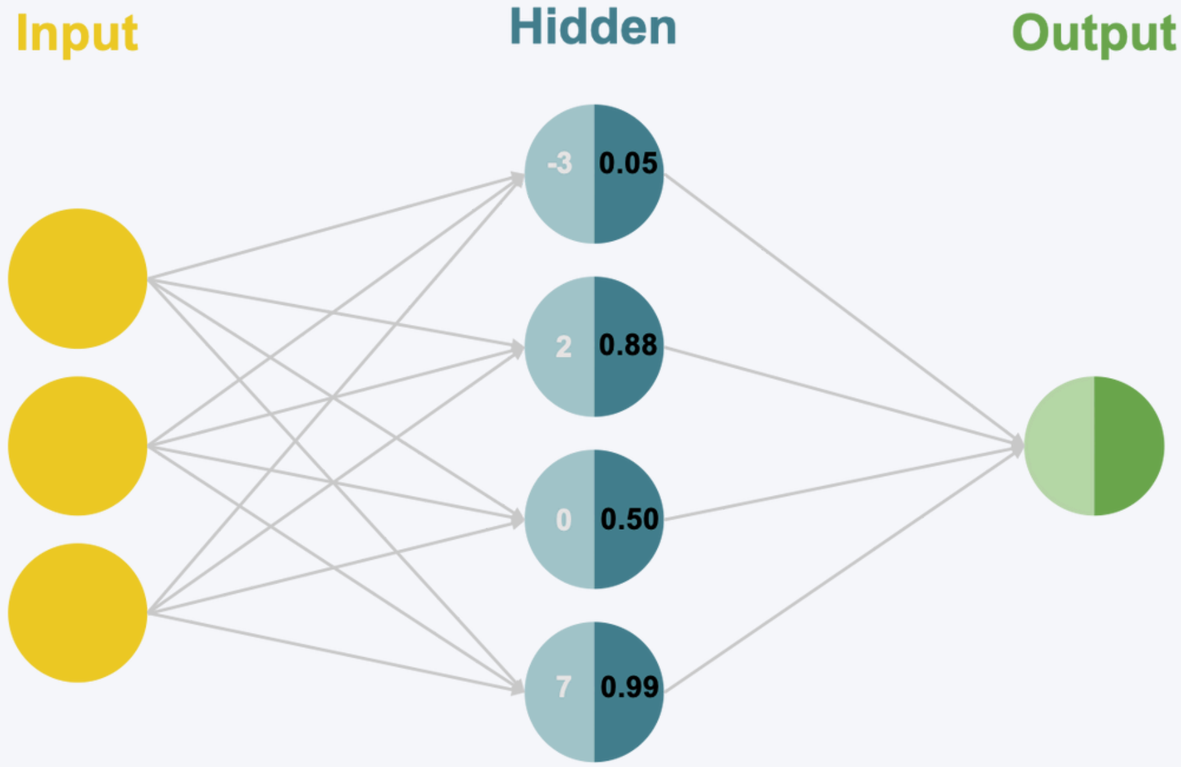
أكاديمية طويق
Tuwaiq Academy

Example: Applying Activation Function

Each hidden neuron performs a non-linear **activation** transformation on the weight sum:

$$z_{\text{out}} = \sigma(z_{\text{in}}) = \frac{1}{1+e^{-z_{\text{in}}}}$$





Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Example: Applying Activation Function

$$u_{\text{out}} = u_1 z_{1,\text{out}} + u_2 z_{2,\text{out}} + u_3 z_{3,\text{out}} + u_4 z_{4,\text{out}} + c$$

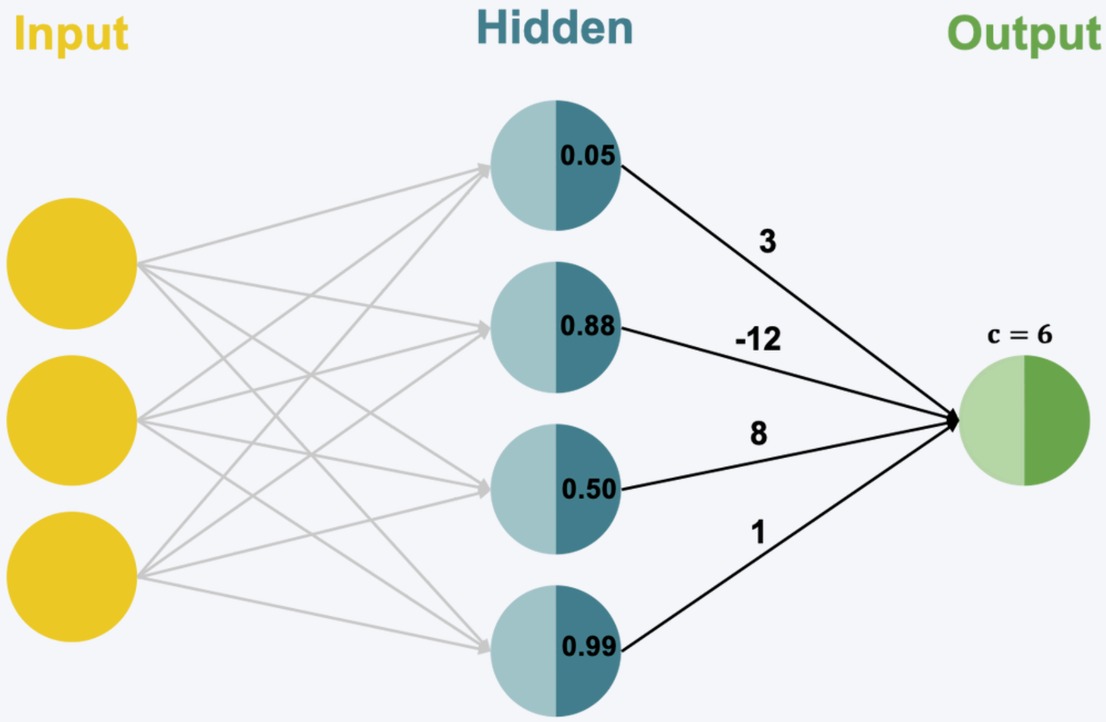
activation transformation on

- Weights (u1,u2,u3,u4) and Bias (c):
- These weights determine how much influence each hidden neuron has on the final output.

$$u_{\text{out}} = (3 \cdot 0.05) + (-12 \cdot 0.88) + (8 \cdot 0.50) + (1 \cdot 0.99) + 6$$

$$u_{\text{out}} = 0.15 - 10.56 + 4.00 + 0.99 + 6$$

$$u_{\text{out}} = 0.58$$



Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

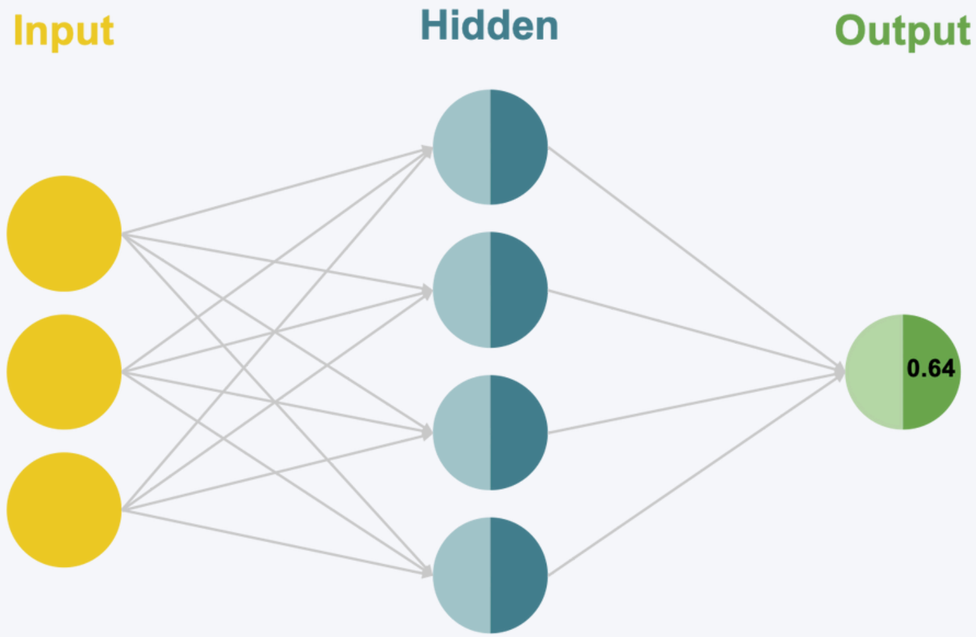
أكاديمية طويق
Tuwaiq Academy

Example: Applying Activation Function

Each hidden neuron performs a non-linear **activation** transformation on the weight sum:

$$f_{\text{out}} = \frac{1}{1 + e^{-0.57}} = \frac{1}{1 + 0.57}$$

$$f_{\text{out}} = \frac{1}{1.57} \approx 0.64$$



Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

أكاديمية طويق
Tuwaiq Academy

Activation Function

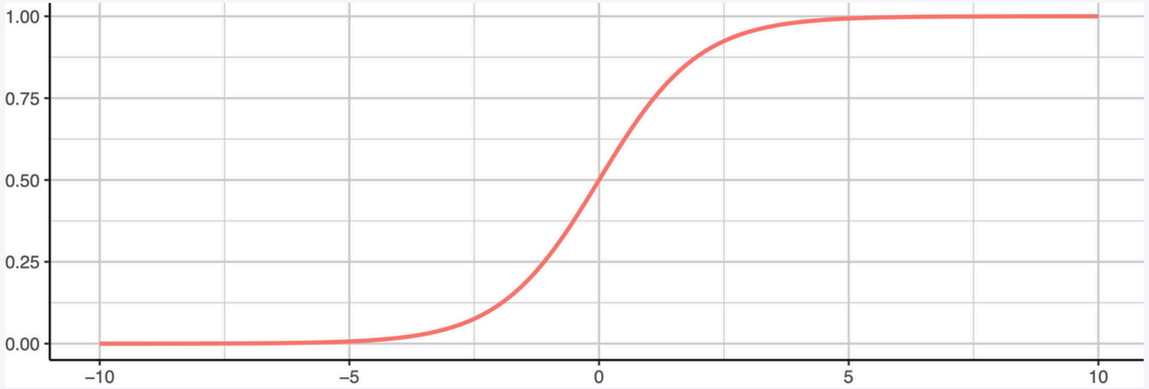
Sigmoid Function

- Formula:
- Range: (0, 1)

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

- When to Use:
  - Binary classification tasks (e.g., for the output layer).
  - Probabilistic outputs (e.g., the probability of a class).
-
- Pros:
  - Outputs are easily interpretable as probabilities.

- Cons:
  - Can suffer from vanishing gradients in deep networks.
  - Slower convergence during training.



Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.

SINGLE HIDDEN LAYER NN

Activation Function

ReLU (Rectified Linear Unit)

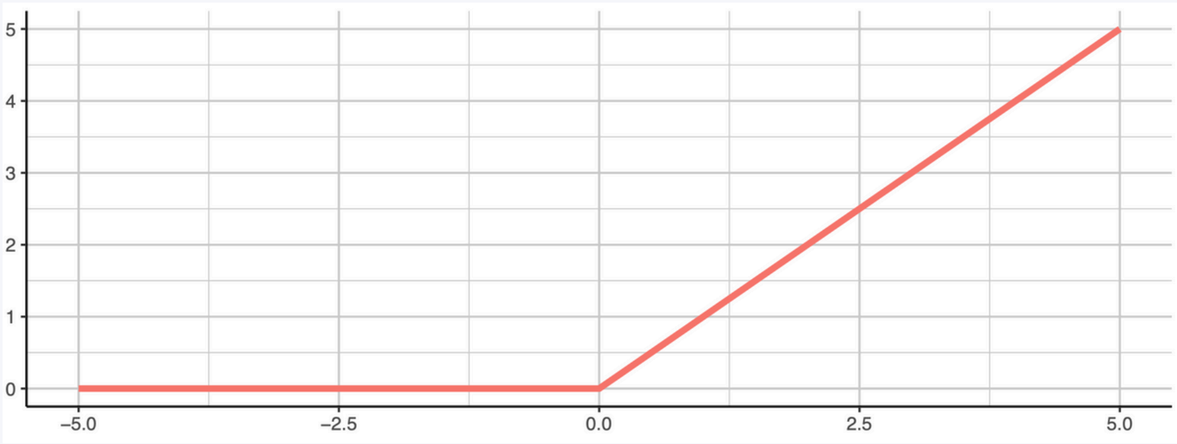
- Formula:

$$\mathrm{ReLU}(x) = \max(0, x)$$

- Range: [0, ∞)

When to Use:

- Hidden layers in most neural networks (e.g., image recognition, NLP).
- General-purpose activation for intermediate layers.



Pros:

- Computationally efficient.
- Avoids vanishing gradient problems.

Cons:

- Can suffer from dying neurons (outputs stuck at 0 for certain weights).

Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine Learning Research 207:1-6 Available from https://proceedings.mlr.press/v207/bothmann23a.html.