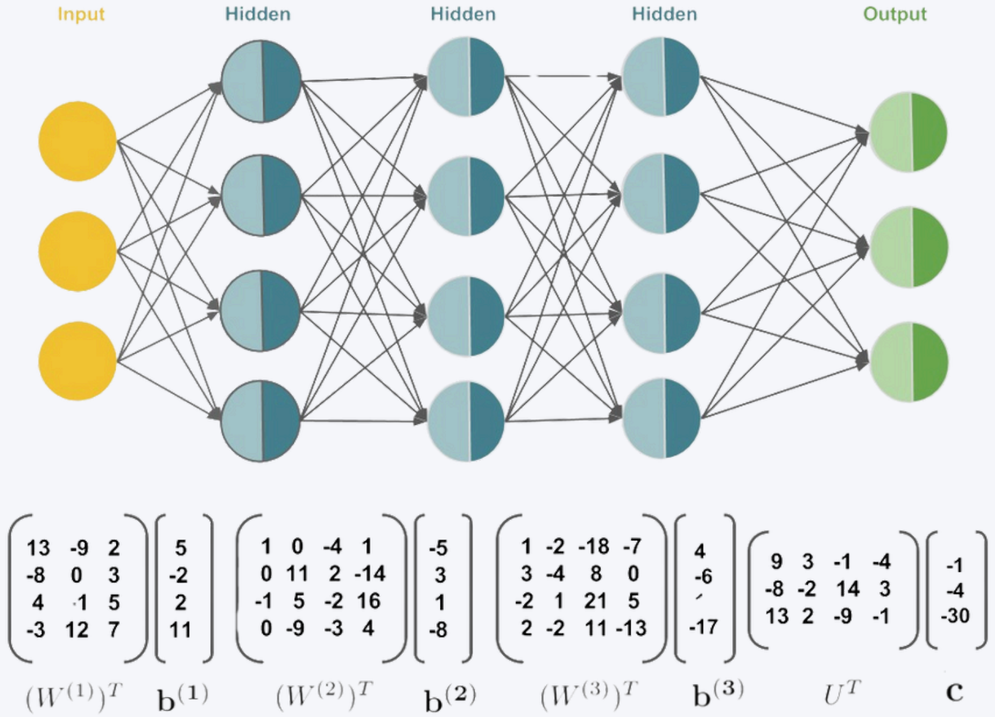
Multi-Layer Feed forward Neural Networks

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

أكاديمية طويق
Tuwaiq Academy

- We will now extend the model class once again, such that we allow an arbitrary amount l of **hidden layers.**

- The general term for this model class is **(multi-layer) feedforward networks** (inputs are passed through the network from left to right, no feedback-loops are allowed)



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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

أكاديمية طويق
Tuwaiq Academy

Feedforward Neural Networks

Non-Linear Activations

- Without non-linear activations, networks can only learn linear decision boundaries.

Layer Components

- Each hidden layer has:
  - Weight matrix (W_i)
  - Bias (b_i)
  - Activation (z_i)

Activation Calculation

  - For hidden layer i, the activation is:
  - $z^{(0)} = x$ (input data).

$$z^{(i)} = \sigma^{(i)}(\phi^{(i)}) = \sigma^{(i)}((W^{(i)})^T z^{(i-1)} + b^{(i)})$$

Chain Structure of the Network

- The network can be described as $f(x) = \tau \circ \phi^{(l)} \circ \sigma^{(l)} \circ \phi^{(l-1)} \circ \sigma^{(l-1)} \cdots \circ \phi^{(1)} \circ \sigma^{(1)}(x)$
- τ and φ are for the output layer.

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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

أكاديمية طويق
Tuwaiq Academy

Feedforward Neural Networks

Non-Linear Activations

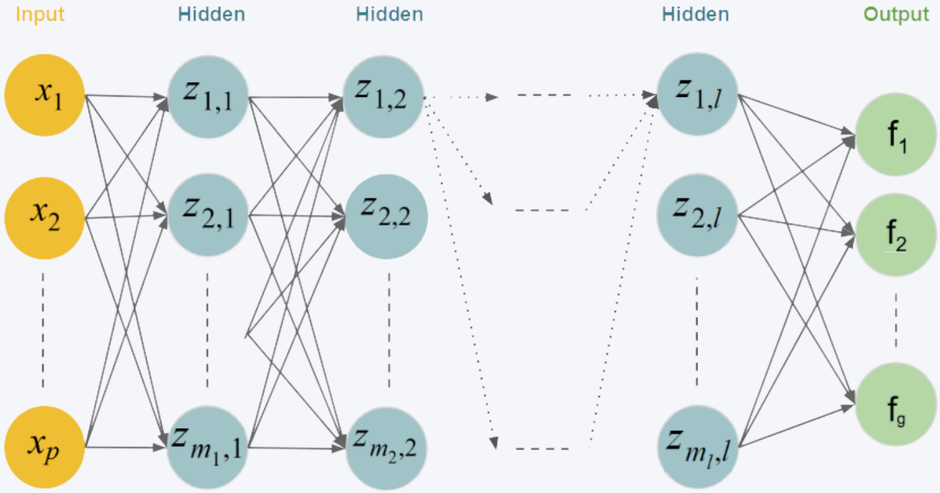
- Without non-linear activations, networks can only learn linear decision boundaries.

Layer Components

- Each hidden layer has:
  - Weight matrix (W_i)
  - Bias (b_i)
  - Activation (z_i)

Activation Calculation

- For hidden layer i, the activation is: $z^{(i)} = \sigma^{(i)}(\phi^{(i)}) = \sigma^{(i)}((W^{(i)})^T z^{(i-1)} + b^{(i)})$
- $z^{(0)} = x$ (input data).



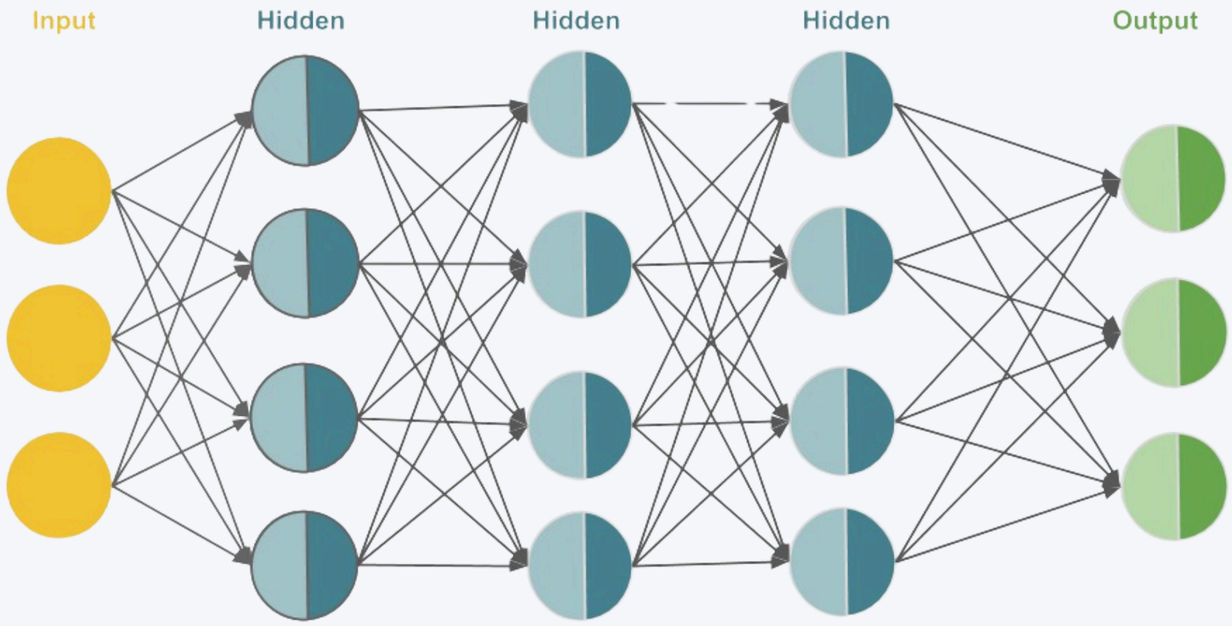
Chain Structure of the Network

- The network can be described as $f(x) = \tau \circ \phi^{(l)} \circ \sigma^{(l)} \circ \phi^{(l-1)} \circ \sigma^{(l-1)} \cdots \circ \phi^{(1)} \circ \sigma^{(1)}(x)$
- τ and φ are for the output layer.

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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

Feedforward Neural Networks | Example



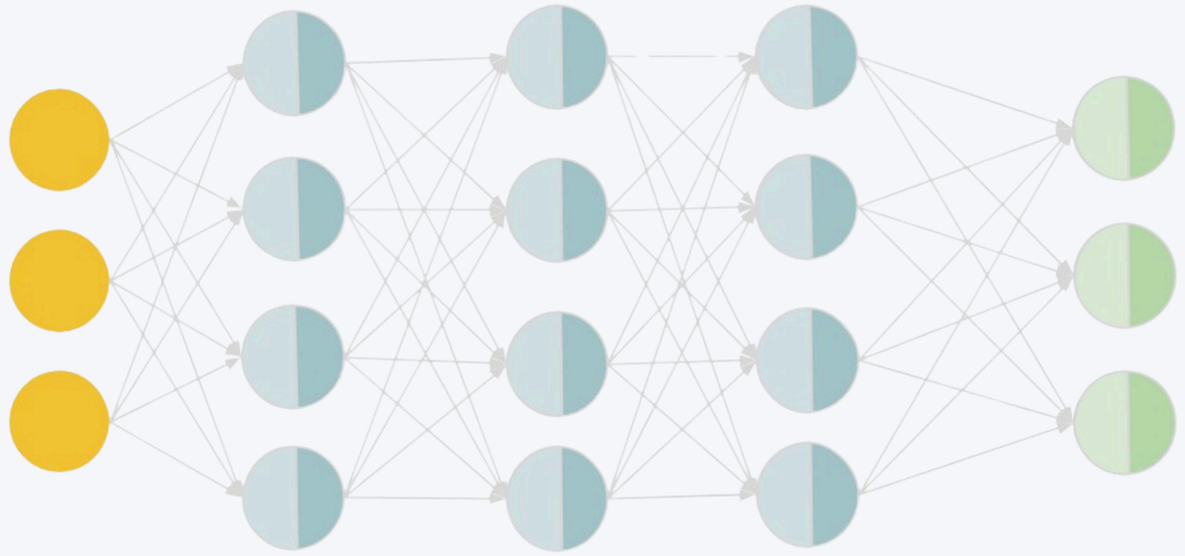
$$\begin{pmatrix} 13 & -9 & 2 \\ -8 & 0 & 3 \\ 4 & -1 & 5 \\ -3 & 12 & 7 \end{pmatrix} \begin{pmatrix} 5 \\ -2 \\ 2 \\ 11 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & -4 & 1 \\ 0 & 11 & 2 & -14 \\ -1 & 5 & -2 & 16 \\ 0 & -9 & -3 & 4 \end{pmatrix} \begin{pmatrix} -5 \\ 3 \\ 1 \\ -8 \end{pmatrix} \quad \begin{pmatrix} 1 & -2 & -18 & -7 \\ 3 & -4 & 8 & 0 \\ -2 & 1 & 21 & 5 \\ 2 & -2 & 11 & -13 \end{pmatrix} \begin{pmatrix} 4 \\ -6 \\ \cdot \\ -17 \end{pmatrix} \quad \begin{pmatrix} 9 & 3 & -1 & -4 \\ -8 & -2 & 14 & 3 \\ 13 & 2 & -9 & -1 \end{pmatrix} \begin{pmatrix} -1 \\ -4 \\ -30 \end{pmatrix}$$

$$(W^{(1)})^T \quad \mathbf{b}^{(1)} \quad (W^{(2)})^T \quad \mathbf{b}^{(2)} \quad (W^{(3)})^T \quad \mathbf{b}^{(3)} \quad U^T \quad \mathbf{c}$$

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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

Feedforward Neural Networks | Example

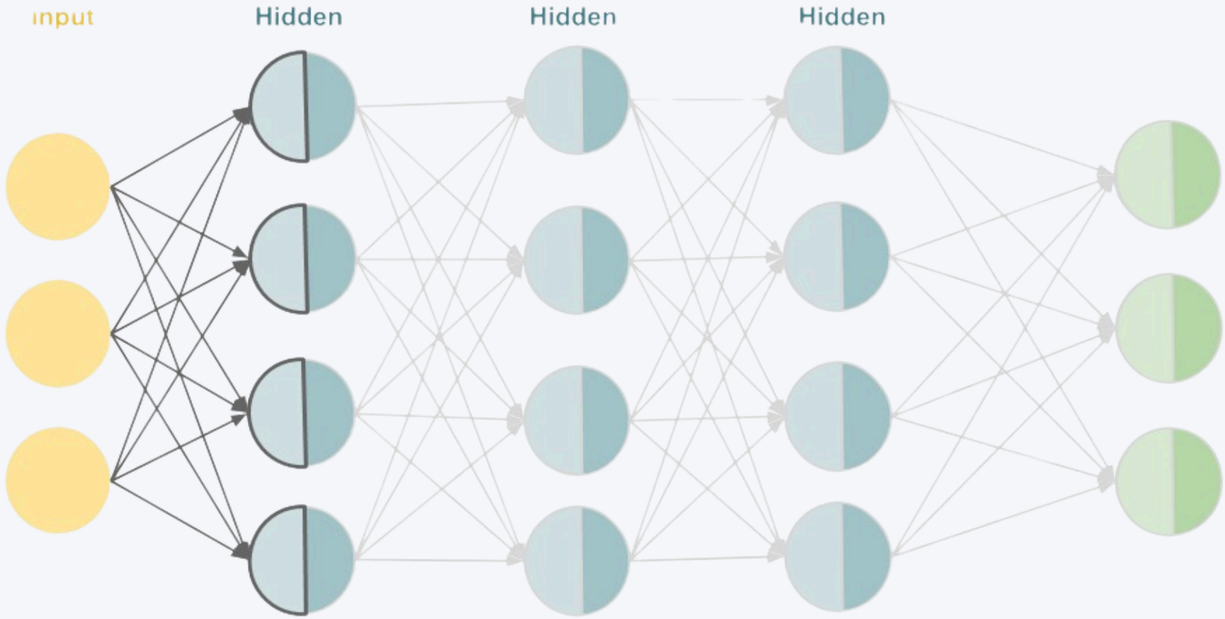


x

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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

Feedforward Neural Networks | Example



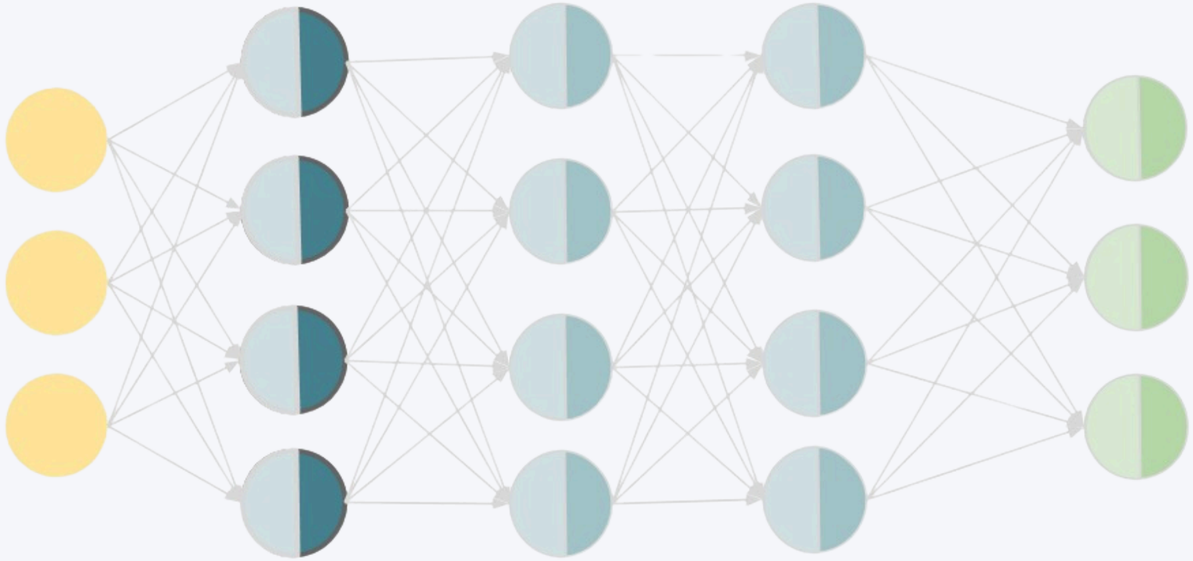
$$\begin{matrix}7\\1\\-4\end{matrix}\begin{pmatrix}7*13 + 1*(-9) + (-4)*2 + 5\\7*(-8) + 1*0 + (-4)*3 + (-2)\\7*4 + 1*(-1) + (-4)*5 + 2\\7*(-3) + 1*12 + (-4)*7 + 11\end{pmatrix}$$

$$\mathbf{x} \quad \mathbf{z}_{in}^{(1)} = W^{(1)T}\mathbf{x} + \mathbf{b}^{(1)}$$

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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

Feedforward Neural Networks | Example

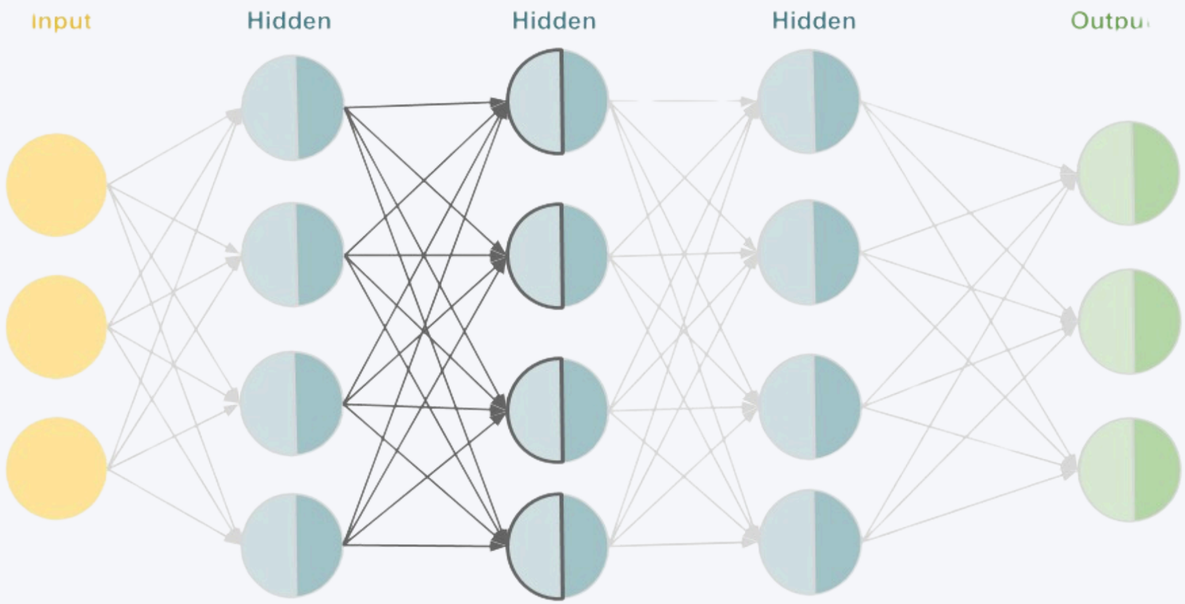


$\mathbf{x} \quad \mathbf{z}_{in}^{(1)} \quad \mathbf{z}^{(1)} = \mathbf{z}_{out}^{(1)} = \sigma(\mathbf{z}_{in}^{(1)})$

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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

Feedforward Neural Networks | Example



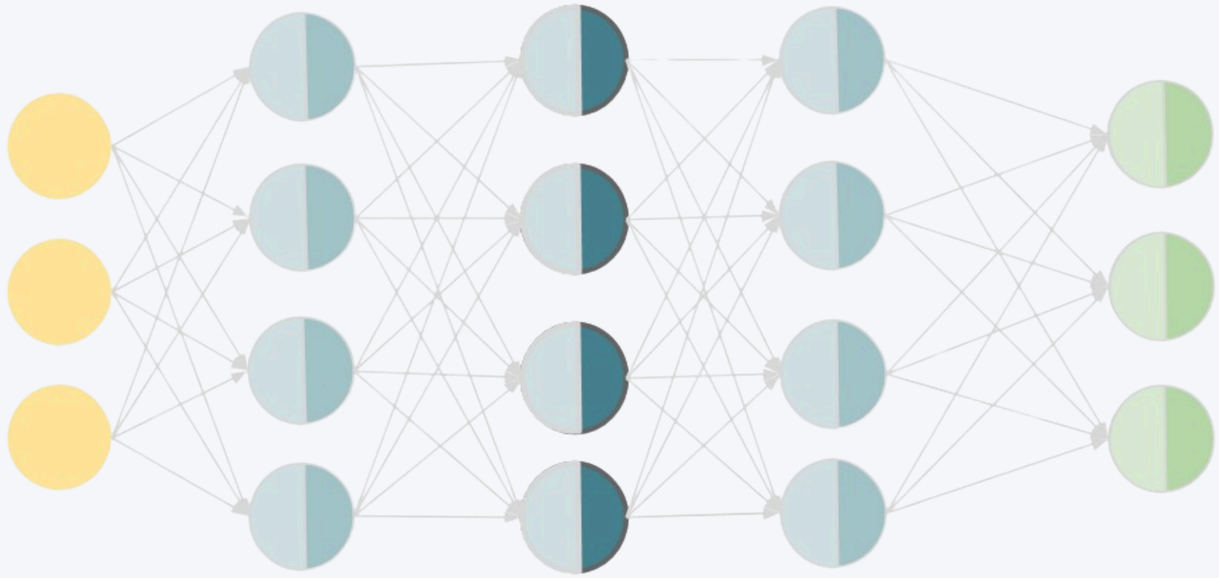
$$\mathbf{x} = \begin{bmatrix} 7 \\ 1 \\ -4 \end{bmatrix} \quad \mathbf{z}_{in}^{(1)} = \begin{bmatrix} 79 \\ -70 \\ 9 \\ -26 \end{bmatrix} \quad \mathbf{z}^{(1)} = \begin{bmatrix} 79 \\ 0 \\ 9 \\ 0 \end{bmatrix} \quad \mathbf{z}_{in}^{(2)} = W^{(2)T}\mathbf{z}^{(1)} + \mathbf{b}^{(2)}$$

$$\begin{pmatrix} 79*1 + 0*0 + 9*(-4) + 0*1 + (-5) \\ 79*0 + 0*11 + 9*2 + 0*(-14) + 3 \\ 79*(-1) + 0*5 + 9*(-2) + 0*16 + 1 \\ 79*0 + 0*(-9) + 9*(-3) + 0*4 + (-8) \end{pmatrix}$$

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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

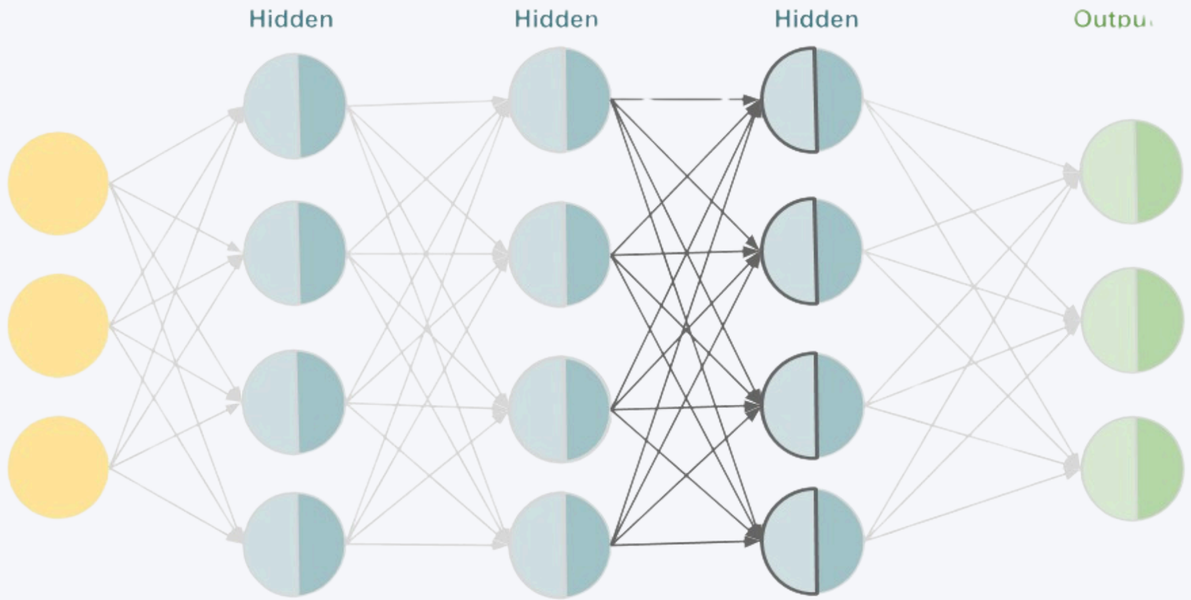
Feedforward Neural Networks | Example



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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

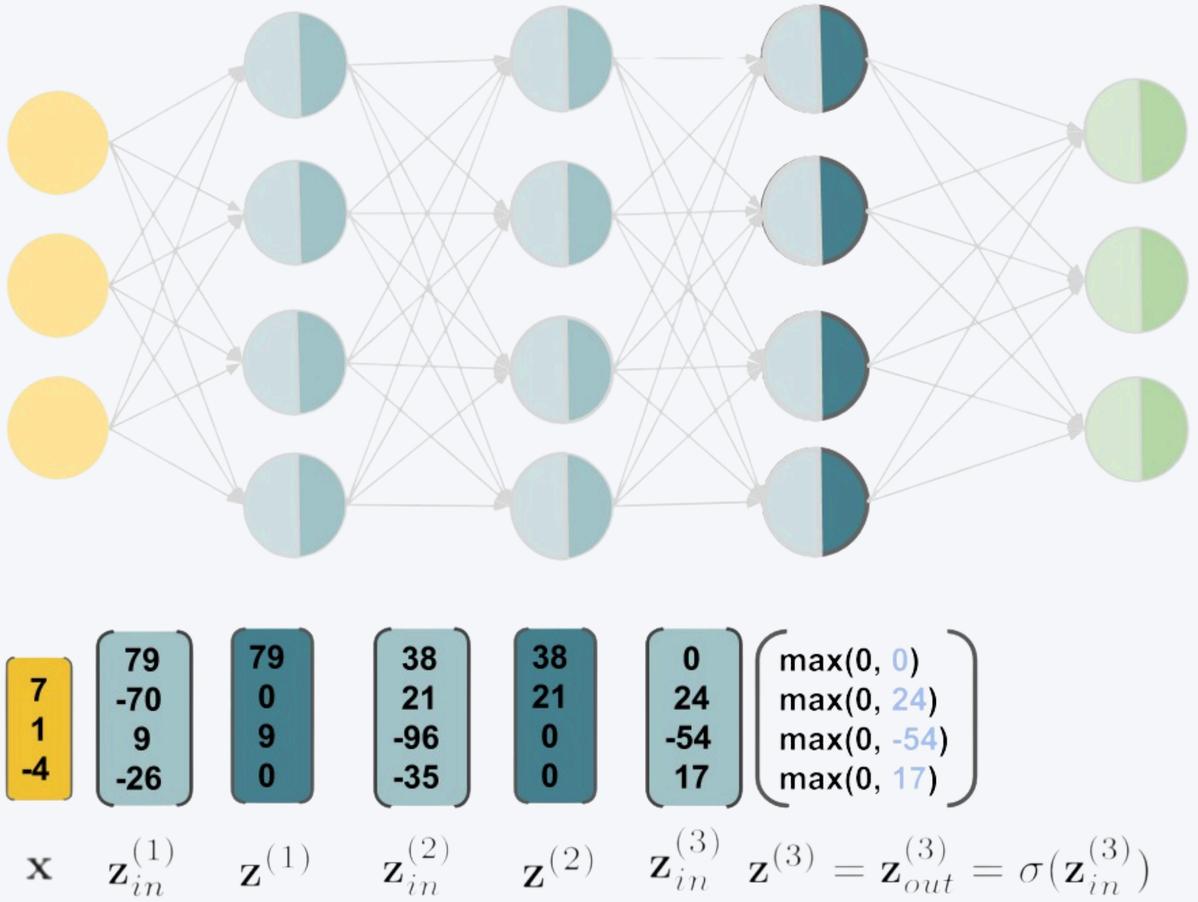
Feedforward Neural Networks | Example



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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

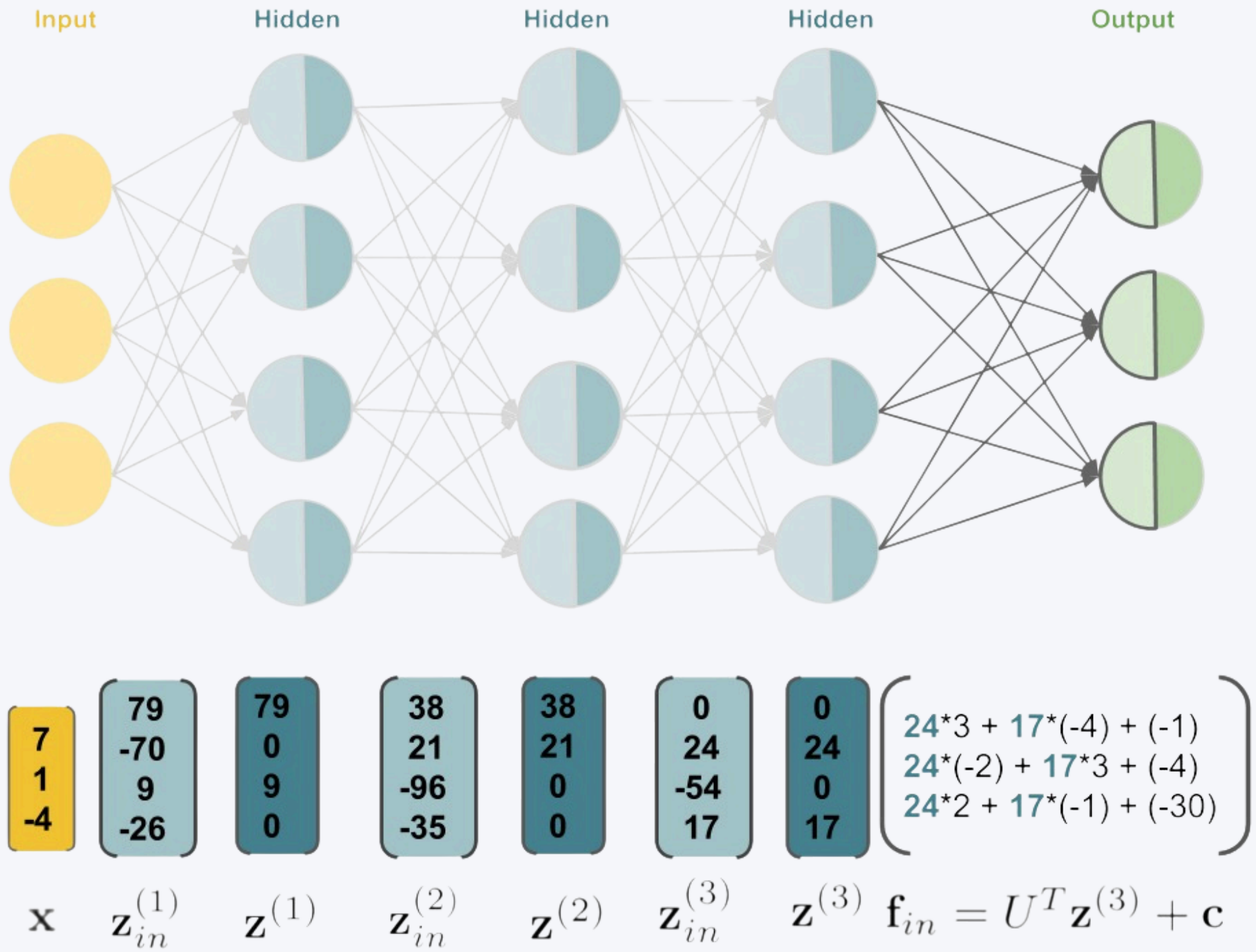
Feedforward Neural Networks | Example



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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

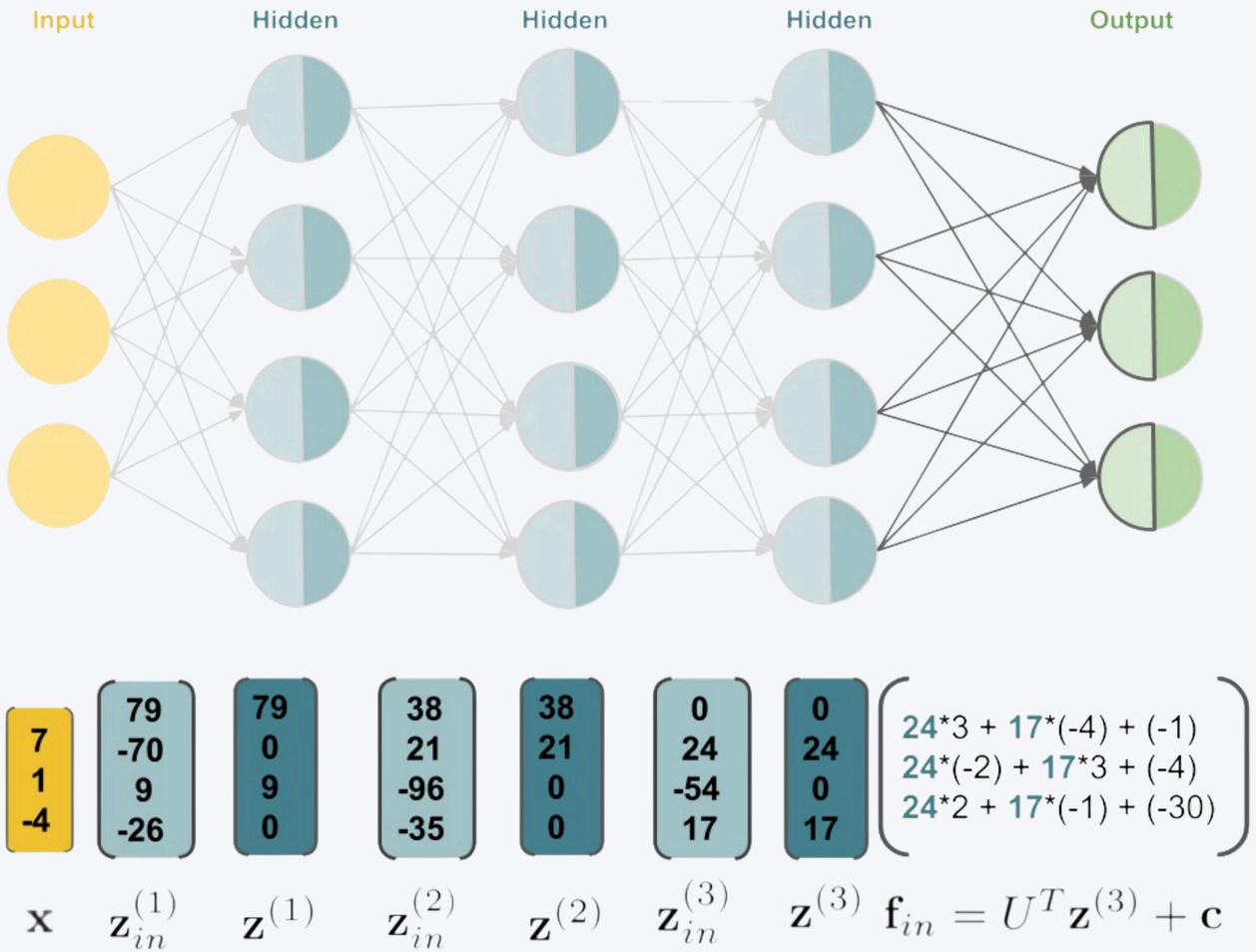
Feedforward Neural Networks | Example



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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

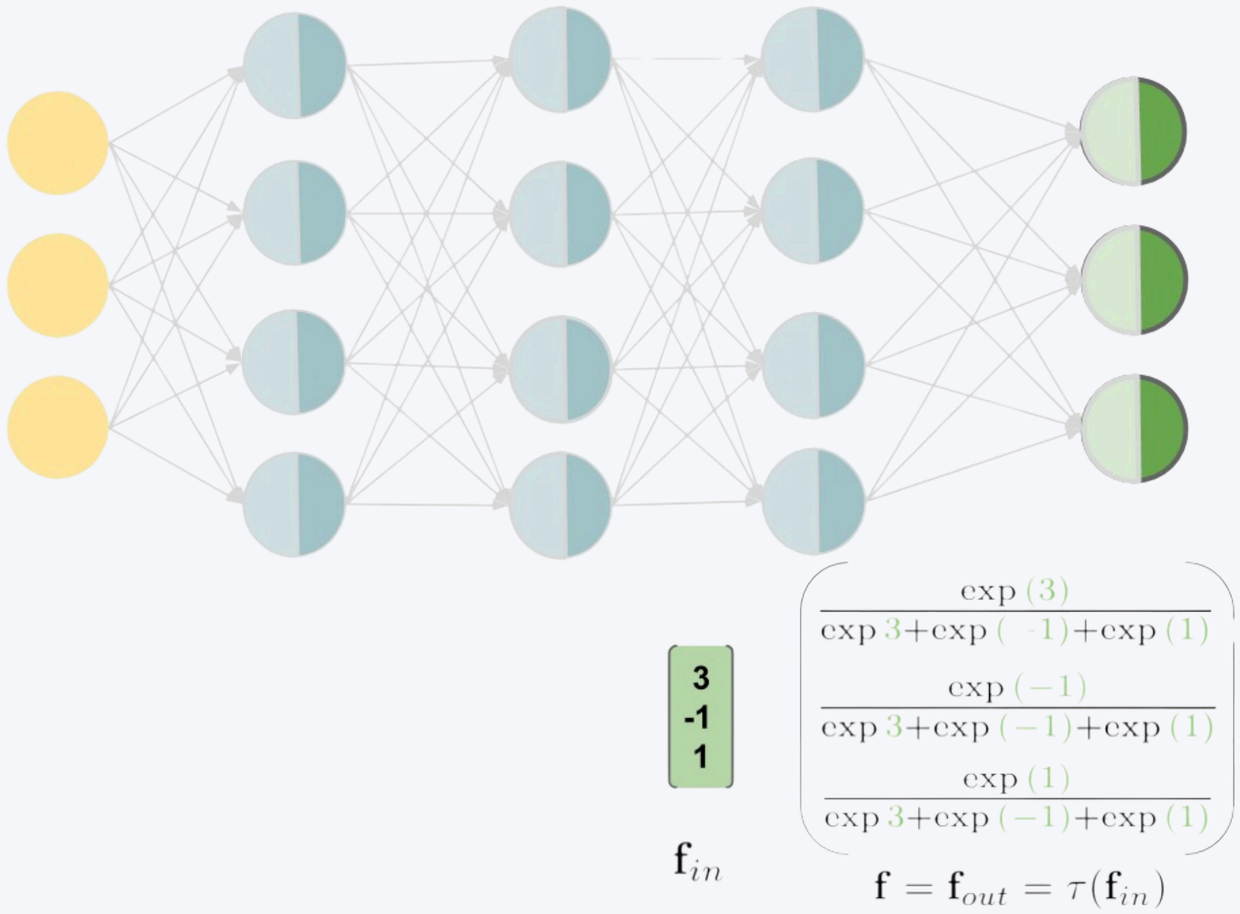
Feedforward Neural Networks | Example



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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

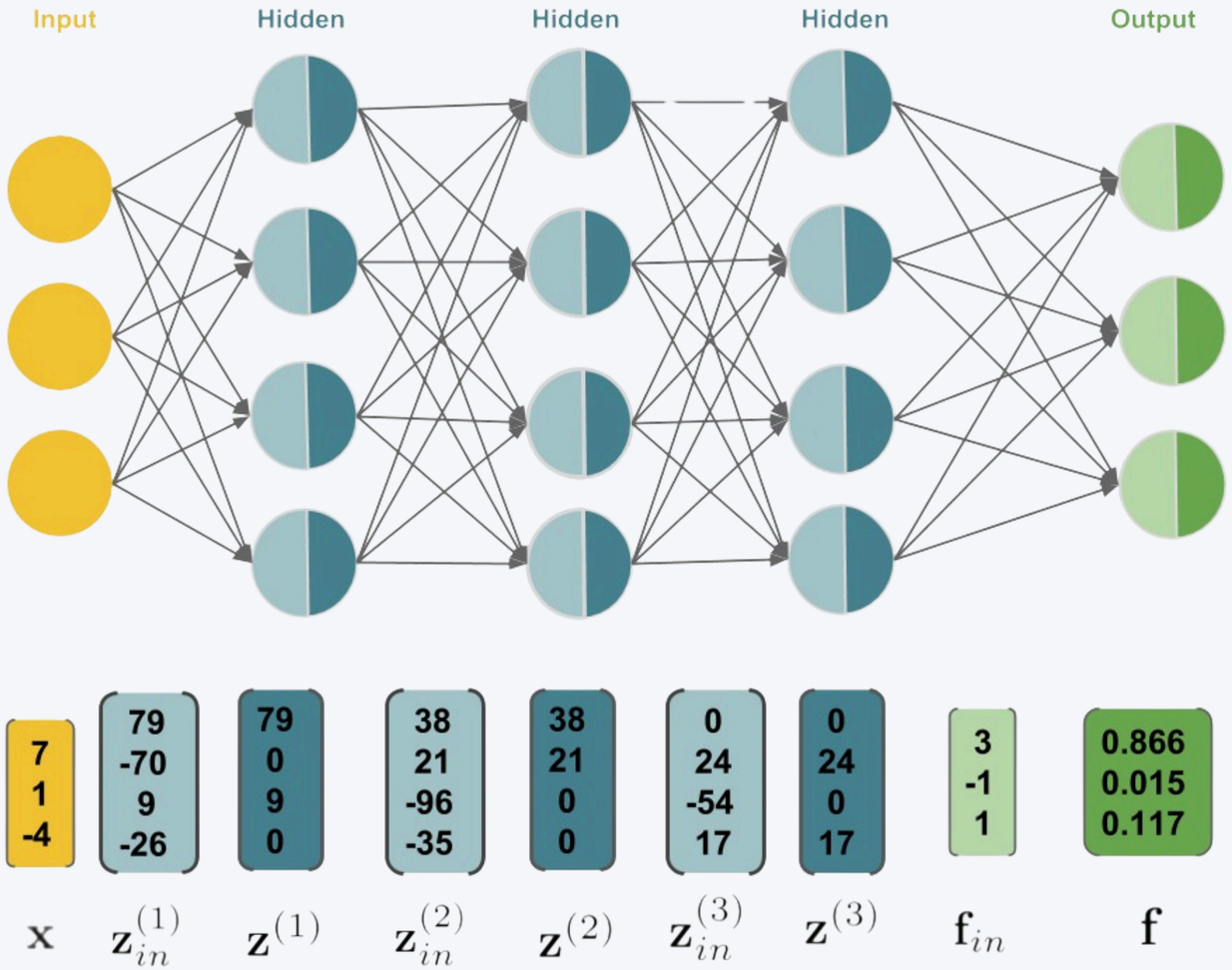
Feedforward Neural Networks | Example



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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

Feedforward Neural Networks | Example



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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

WHY ADD MORE LAYERS?

Multiple layers allow for the **extraction of more and more** abstract representations.

Each layer in a feed-forward neural network adds its own degree of non-linearity to the model.

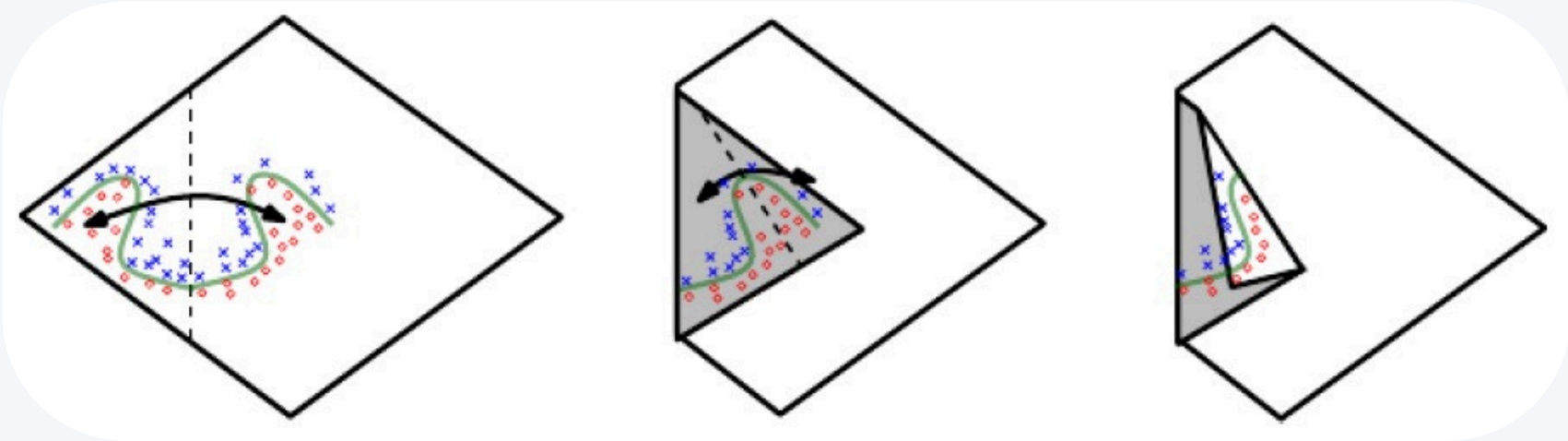


Figure: An intuitive, geometric explanation of the exponential advantage of deeper networks formally (Montúfar et al., 2014).

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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

DEEP NEURAL NETWORKS

Neural networks today can have hundreds of hidden layers.

The greater the number of layers, the "deeper" the network.

Historically DNNs were very challenging to train and not popular until the late ’00s for several reasons:

- The use of sigmoid activations (e.g., logistic sigmoid and tanh) significantly slowed down training due to a phenomenon known as “vanishing gradients”. **The introduction of the ReLU activation largely solved this problem.**
- Training DNNs on CPUs was too slow to be practical. **Switching over to GPUs** cut down training time by more than an order of magnitude.
- **When dataset sizes are small**, other models (such as SVMs) and techniques (such as feature engineering) often outperform them.

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.

MULTI-LAYER FEED FORWARD NEURAL NETWORKS

DEEP NEURAL NETWORKS

The availability of large datasets and novel architectures that are capable of handling even complex tensor-shaped data (e.g. CNNs for image data), **faster hardware, and better optimization and regularization methods made it feasible to successfully implement deep neural networks.**

An increase in depth often translates to an increase in performance on a given task. State-of-the-art neural networks, however, are much more sophisticated than the simple architectures we have encountered so far.

The term **"deep learning"** encompasses all of these developments and refers to the field as a whole.

Montúfarr, G., Pascanu, R., Cho, K., & Bengio, Y. (2014). Linear Regions of Deep Neural Networks.

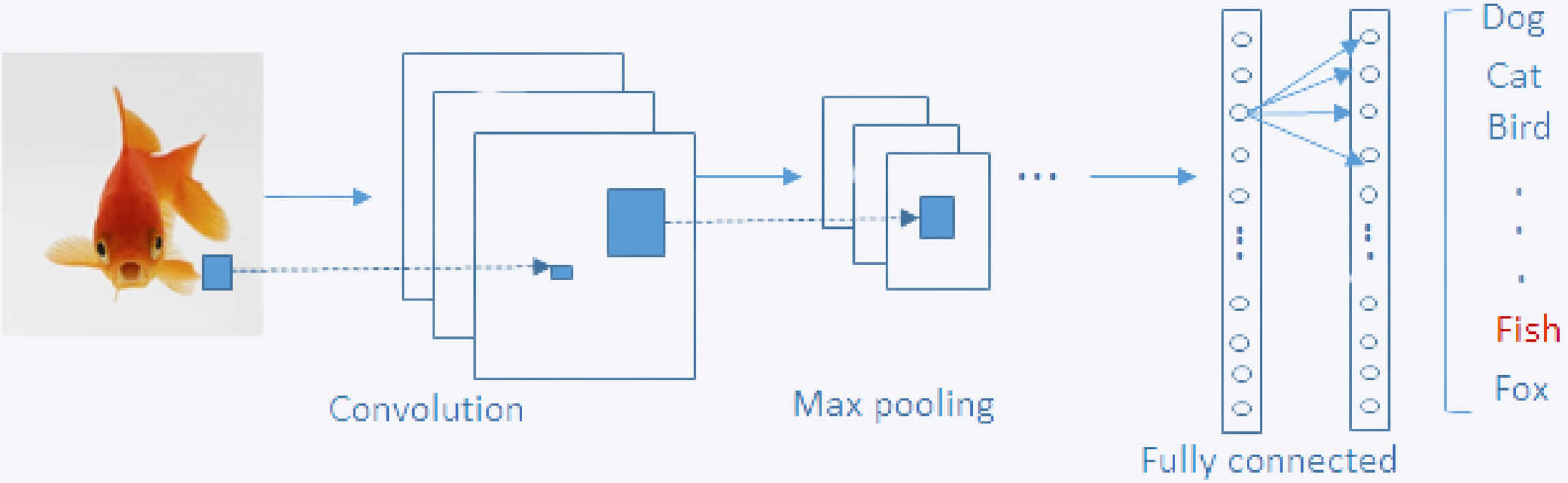
Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine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

CONVOLUTIONAL NEURAL NETWORKS

CONVOLUTIONAL NEURAL NETWORKS

أكاديمية طويق
Tuwaiq Academy

CONVOLUTIONAL NEURAL 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.

CONVOLUTIONAL NEURAL NETWORKS

أكاديمية طويق
Tuwaiq Academy

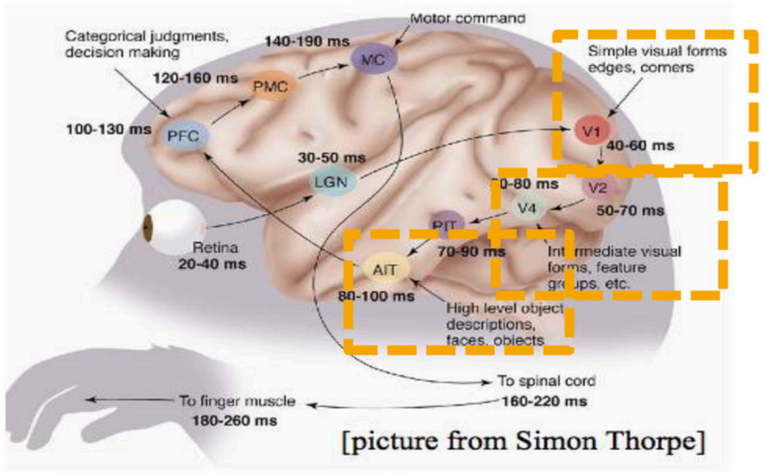
CONVOLUTIONAL NEURAL NETWORKS

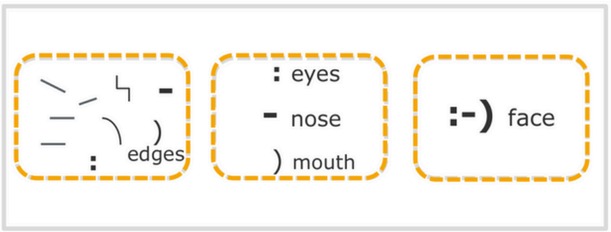
What Are CNNs?

- CNNs are a type of neural network designed to process visual data (like images).
- They are inspired by how the human brain processes visual information, particularly the visual cortex.

How Do CNNs Work?

- CNNs detect patterns in images, starting with simple features (like edges or corners) and building up to more complex features (like faces or objects).
- This happens in a hierarchical way, similar to how the brain processes vision.





Brain Analogy :

- **Visual Cortex Layers:**
  - The brain processes visual inputs in stages:
    - Simple features: Edges and corners (early stages).
    - Intermediate features: Shapes or outlines.
    - Complex objects: Faces or objects (later stages).
- **CNN Mimics This Process:**
  - Early layers of CNNs detect edges and textures (like the brain detecting edges).
  - Middle layers find shapes or parts of objects.
  - Final layers identify the whole object, like a face.

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.

CONVOLUTIONAL NEURAL NETWORKS

أكاديمية طويق
Tuwaiq Academy

Key Features of CNNs

- **CNNs** are made up of building blocks and components that automatically extract important features from data.
- They excel in **extracting spatial patterns** (like shapes, textures, and positions) from input data.
- Though originally designed for **images,** CNNs are also used in other domains like:
    - Natural Language Processing (NLP)
    - Audio Processing
    - Time-Series Analysis

Applications of CNNs in Computer Vision

Image Classification:

- Identifying the category or label of an image (e.g., cat, dog, car).

Object Detection/Localization:

- Finding and identifying objects in an image (e.g., detecting a car and its position in the image).

Semantic Segmentation:

- Dividing an image into regions and labeling each region (e.g., labeling every pixel as "sky," "tree," or "road").

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.

CONVOLUTIONAL NEURAL NETWORKS

CNNS - WHAT FOR?

A convolutional neural network is used to map raw pixels from a single front-facing camera directly into steering commands. The system learns to drive in traffic, on local roads, with or without lane markings as well as on highways.

Figure: All Tesla cars being produced now have full self-driving hardware

(Source: Tesla website)

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.

CONVOLUTIONAL NEURAL NETWORKS

CNNS - WHAT FOR?

computer vision (Zhang et al. (2016)). Given a grayscale photo as the input (top row), this network solves the problem of hallucinating a plausible color version of the photo (bottom row, i.e. the prediction of the network).

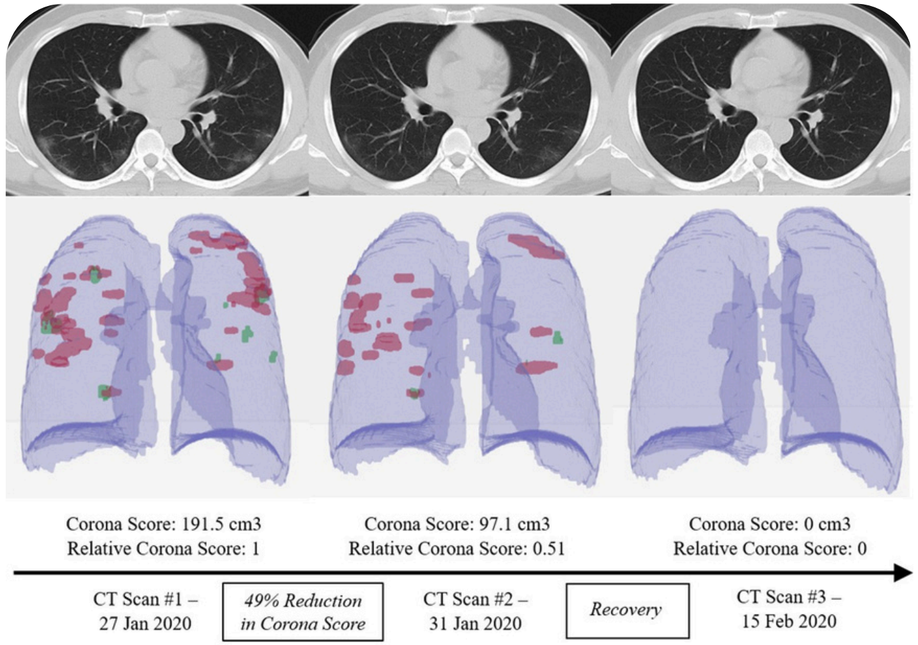
Figure: Image Colorization is another interesting application of CNN in

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.

CONVOLUTIONAL NEURAL NETWORKS

CNNS - WHAT FOR?

CNN for personalized medicine Examples: Tracking, diagnosis and localization of Covid-19 patients. CNN based method (RADLogists) for personalized Covid-19 detection: three CT scans from a single Corona virus patient diagnosed by RADLogists.



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.

CONVOLUTIONAL NEURAL NETWORKS

أكاديمية طويق
Tuwaiq Academy

CNNS - A FIRST GLIMPSE

Input Layer
- Takes input data, e.g., an image, audio, or other spatial data.

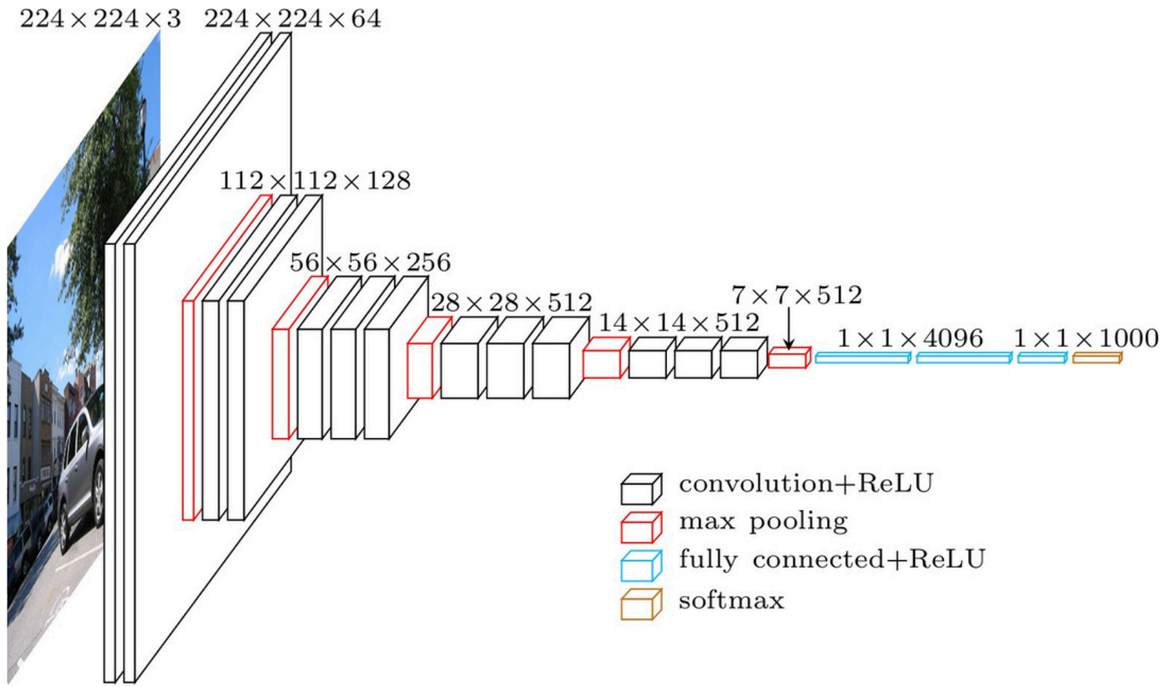
Convolution Layers
- Extract feature maps by applying filters to detect patterns like edges, corners, or textures.

Pooling Layers
- Reduce dimensionality of feature maps while keeping important information.
- Helps to filter meaningful features and make computations faster.

Fully Connected Layers
- Connect all features from previous layers.
- Combines extracted features for final decision-making.

Output Neurons
- Generate final predictions.

Softmax Layer
- Converts raw output values into probability scores, making it easy to interpret predictions.



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.

CONVOLUTIONAL OPERATION

CONVOLUTIONAL NEURAL NETWORKS

CONVOLUTIONAL OPERATION

FILTERS TO EXTRACT FEATURES

- Filters are widely applied in Computer Vision (CV) since the 70’s.
- One prominent example: Sobel-Filter. It detects edges in images.

What Are Edges?

- Edges happen where pixel intensity changes quickly (e.g., between light and dark areas).

How to Detect Edges?

- Approximate the gradient (rate of intensity change) for each pixel.

Sobel Gradient in the X-Direction:

Sobel filter approximates the gradient Gx in the x-dimension using this formula:**$G_x = S_x * A$**

- **A:** Original image.
- **Sx:** Sobel filter matrix.
- $*$: Convolution (a special operation, not standard multiplication).

Applying the Sobel filter detects edges by calculating intensity changes across the image in specific directions (x or y).

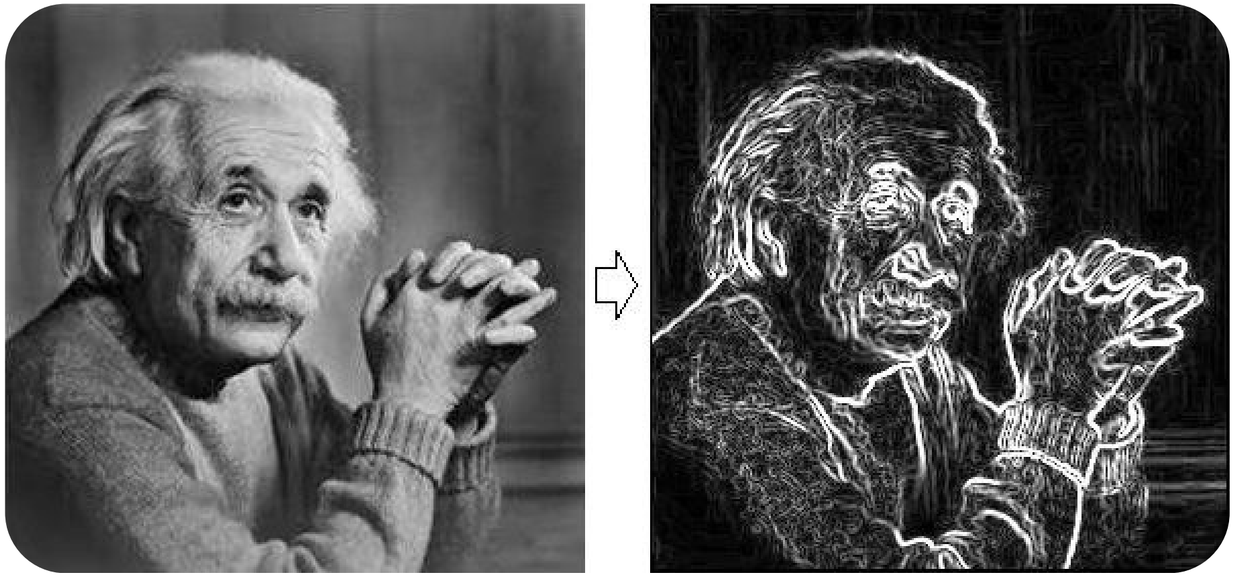


Figure: Sobel-filtered image.

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.

CONVOLUTIONAL NEURAL NETWORKS

FILTERS TO EXTRACT FEATURES:

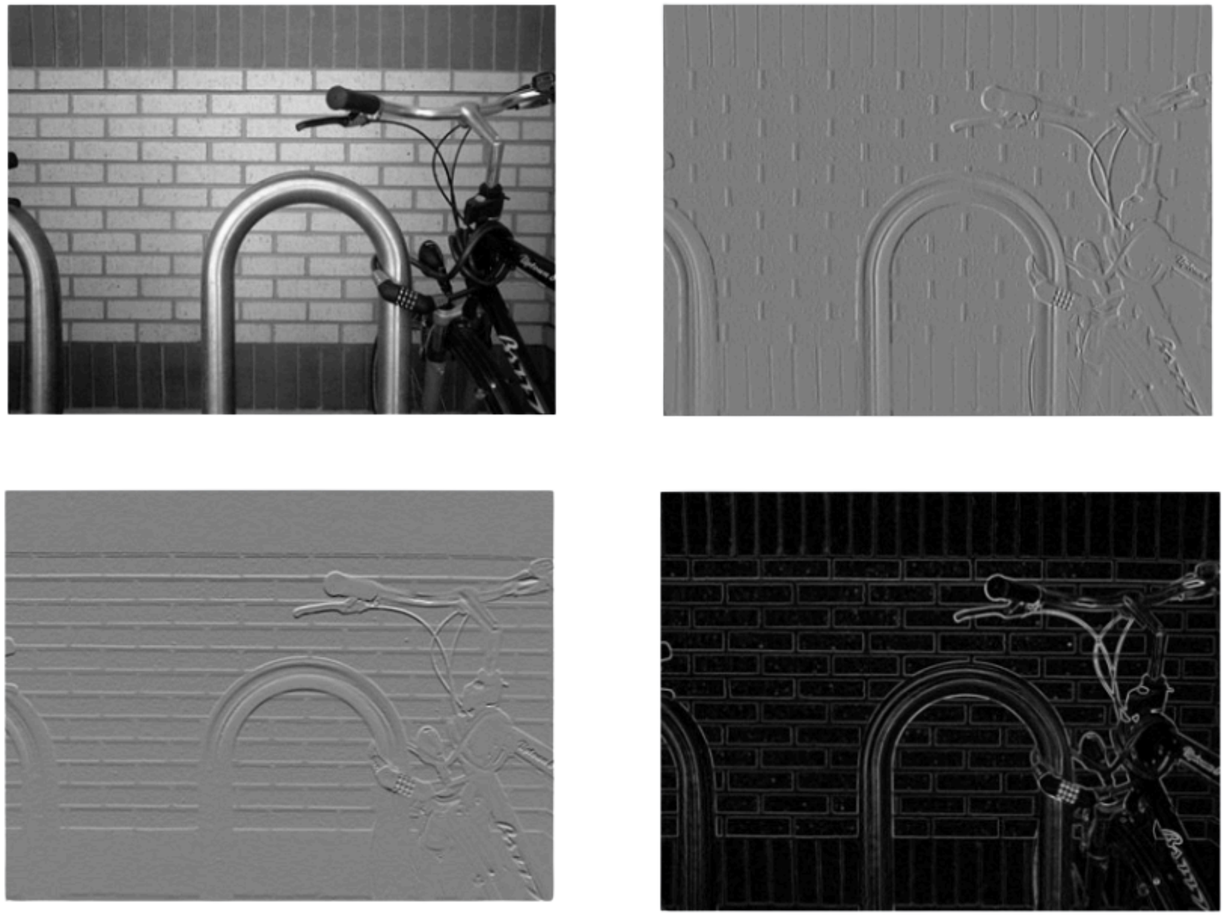


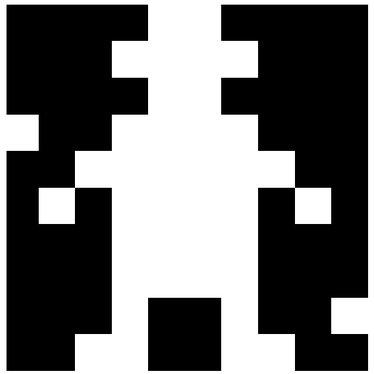
Figure: Sobel filtered images. Outputs are normalized in each case.

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.

CONVOLUTIONAL NEURAL NETWORKS

FILTERS TO EXTRACT FEATURES

How to represent a digital image?



Width

0	0	0	0	255	255	0	0	0	0
0	0	0	255	255	255	255	0	0	0
0	0	0	0	255	255	0	0	0	0
255	0	0	255	255	255	255	0	0	0
0	0	255	255	255	255	255	255	0	0
0	255	0	255	255	255	255	0	255	0
0	0	0	255	255	255	255	0	0	0
0	0	0	255	255	255	255	0	0	0
0	0	0	255	0	0	255	0	0	255
0	0	255	255	0	0	255	255	0	0

length

Basically as an array of integers.

- Digital image is just numbers organized in an array that computers can process, with each number representing how bright or colorful a pixel is.

For simplicity, the image is converted to black and white (grayscale):

- **Black = 0 (low intensity).**
- **White = 255 (high intensity).**

- Every pixel is assigned a value between 0 and 255, representing brightness.

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.

CONVOLUTIONAL NEURAL NETWORKS

أكاديمية طويق
Tuwaiq Academy

FILTERS TO EXTRACT FEATURES

S_x: enables us to to detect vertical edges!

- The **Sobel Operator Sx** is used as a filter to detect vertical edges in the image.
- The filter slides (convolves) across the image matrix.

Sobel-Operator

$$S_X = \begin{bmatrix} -1 & 0 & +1 \\ -2 & \boxed{0} & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

0	0	0	0	255	255	0	0	0	0
0	0	0	255	255	255	255	0	0	0
0	0	0	0	255	255	0	0	0	0
255	0	0	255	255	255	255	0	0	0
0	0	255	255	255	255	255	255	0	0
0	255	0	255	255	255	255	0	255	0
0	0	0	255	255	255	255	0	0	0
0	0	0	255	255	255	255	0	0	0
0	0	0	255	0	0	255	0	0	255
0	0	255	255	0	0	255	255	0	0

- At each position, it performs:
  - Element-wise multiplication between the filter and the image region.
  - Summation of the results.
- The **output matrix** highlights vertical edges by assigning high values where edges occur.

Sobel-Operator

$$S_X = \begin{bmatrix} -1 & 0 & +1 \\ -2 & \boxed{0} & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

0	0	0	0	255	255	0	0	0	0
0	0	0	255	255	255	255	0	0	0
0	0	0	0	255	255	0	0	0	0
255	0	0	255	255	255	255	0	0	0
0	0	255	255	255	255	255	255	0	0
0	255	0	255	255	255	255	0	255	0
0	0	0	255	255	255	255	0	0	0
0	0	0	255	255	255	255	0	0	0
0	0	0	255	0	0	255	0	0	255
0	0	255	255	0	0	255	255	0	0

- A small section of the image matrix is shown being processed with Sx, resulting in a new value in the feature map.

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.

CONVOLUTIONAL NEURAL NETWORKS

أكاديمية طويق
Tuwaiq Academy

FILTERS TO EXTRACT FEATURES

0	510	1020	510	-510	-1020	-510	0
-255	510	1020	510	-510	-1020	-510	0
-255	765	765	255	-255	-765	-765	-255
255	765	510	0	0	-510	-765	-510
255	510	765	0	0	-765	-510	-255
0	765	1020	0	0	-1020	-765	0
0	1020	765	-255	255	-765	-1020	255
255	1020	0	-765	765	0	-1020	255

Applying the Sobel-Operator to every location in the input yields the **feature map**.

- The filter is applied to every possible position in the input matrix.
- A new matrix (feature map) is produced where:
  - High values correspond to detected edges.
  - Low values indicate areas with no edges.

Normalization: The feature map is scaled or normalized to better highlight the patterns.

Normalized feature map reveals vertical edges.
Note the dimensional reduction compared to the dummy image.

128	191	255	191	64	0	64	128
96	191	255	191	64	0	64	128
96	223	223	159	96	32	32	96
159	223	191	128	128	64	32	64
159	191	223	128	128	32	64	96
128	223	255	128	128	0	32	128
128	255	223	96	159	32	0	159
159	255	128	32	223	128	0	159

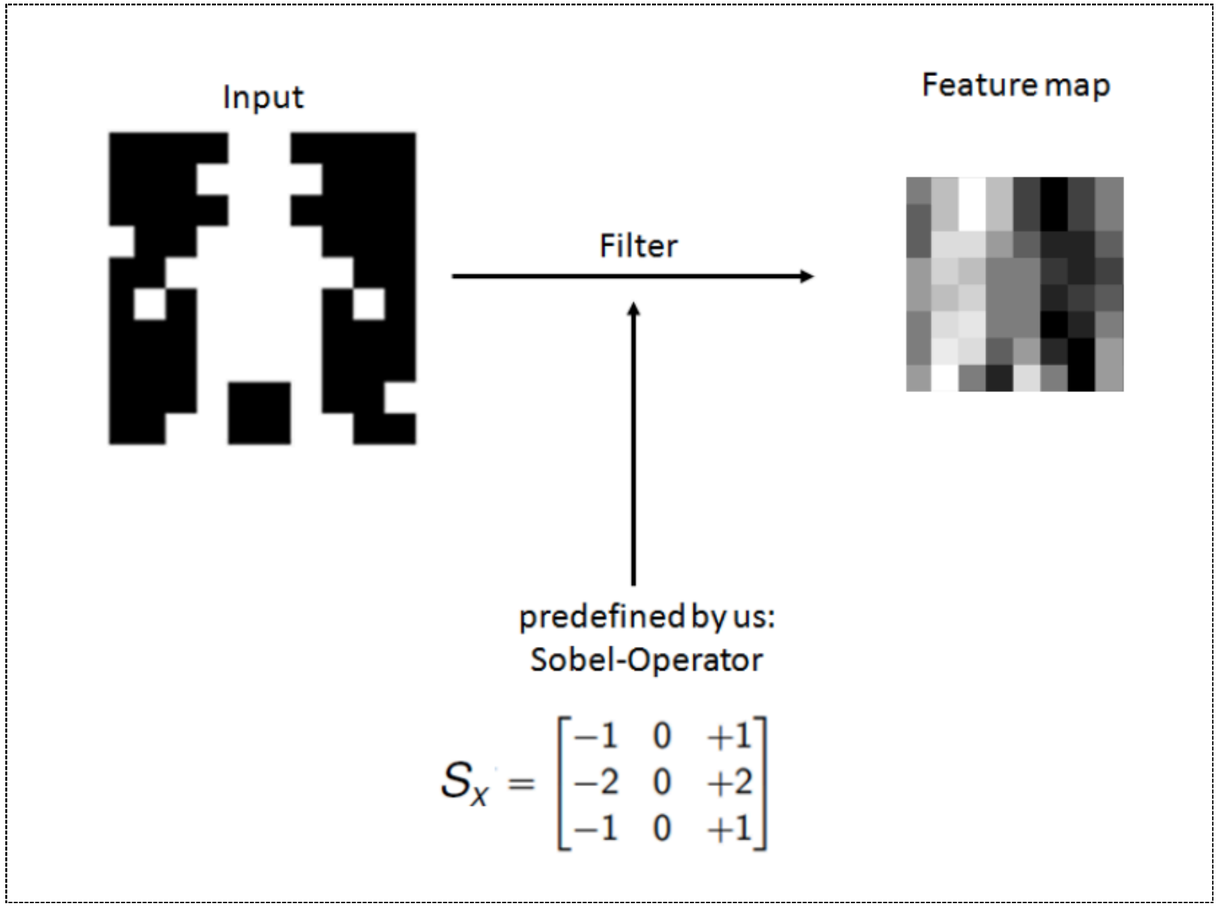
Bothmann, L., Strickroth, S., Casalicchio, G., Rügamer, D., Lindauer, M., Scheipl, F. & Bischl, B.. (2023). Developing Open Source Educational Resources for Machine Learning and Data Science. Proceedings of the Third Teaching Machine Learning and Artificial Intelligence Workshop, in Proceedings of Machine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

CONVOLUTIONAL NEURAL NETWORKS

FILTERS TO EXTRACT FEATURES

- **Filters** like the Sobel operator are applied to input images in CNNs to create feature maps, enabling the network to detect meaningful patterns.
- **Input:** The original black-and-white image.
- **Filter**: The predefined Sobel operator Sx.
- **Output**: The resulting feature map highlights vertical edges in the image.
- **Purpose**: These feature maps are used as the foundation for deeper layers of a CNN to detect more complex features (e.g., shapes, objects).

The convolution process transforms raw image data into feature maps that are essential for tasks like object detection and classification.



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.