AI-based Audio Analysis of Music and Soundscapes

Deep Learning

Dr.-Ing. Jakob Abeßer
Fraunhofer IDMT

jakob.abesser@idmt.fraunhofer.de

Deep Learning
Outline

- Introduction
- Fully Connected Neural Networks
- Convolutional Neural Networks

Deep Learning
Introduction

- Artificial neural networks → mimic brain processing
    - Connected neurons
    - Weighted input summation
    - Non-linear processing

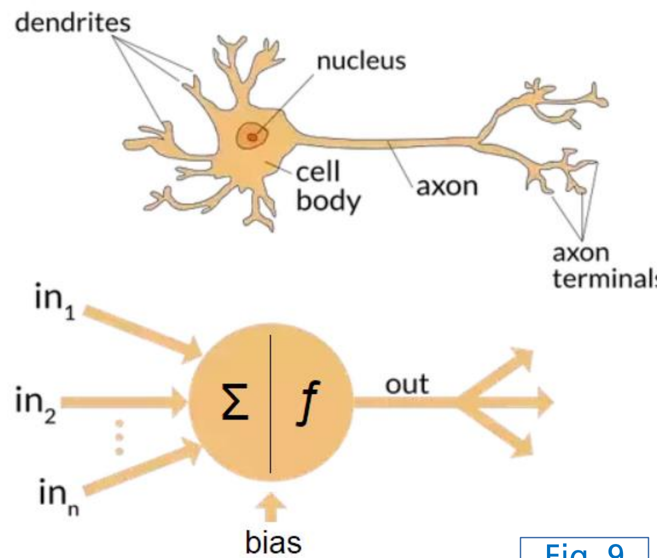


Fig. 9

Deep Learning
Introduction

- Artificial neural networks → mimic brain processing
  - Connected neurons
  - Weighted input summation
  - Non-linear processing
- Shallow networks

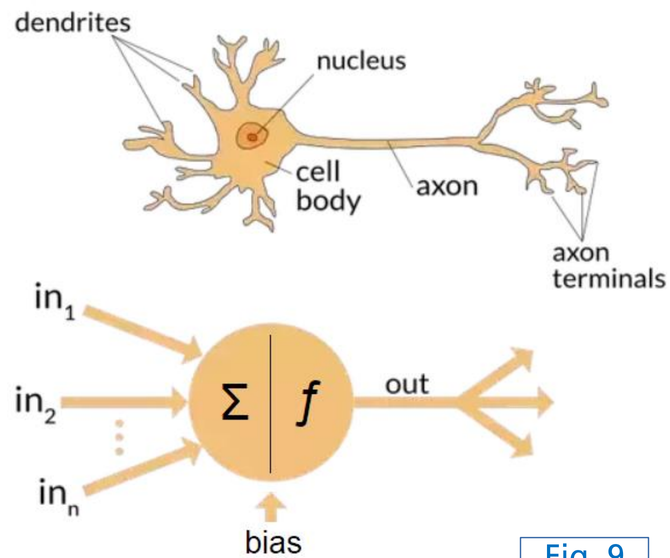


Fig. 9

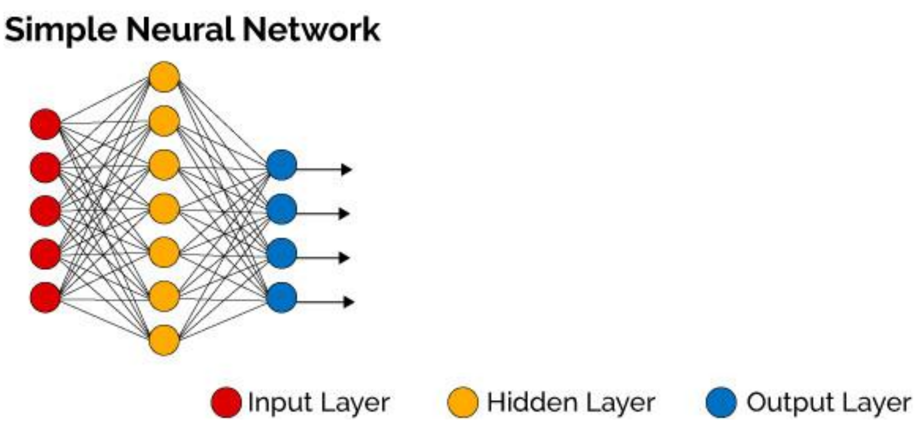


Fig. 10

Deep Learning
Introduction

- Artificial neural networks → mimic brain processing
    - Connected neurons
    - Weighted input summation
    - Non-linear processing
- Shallow networks → deep networks

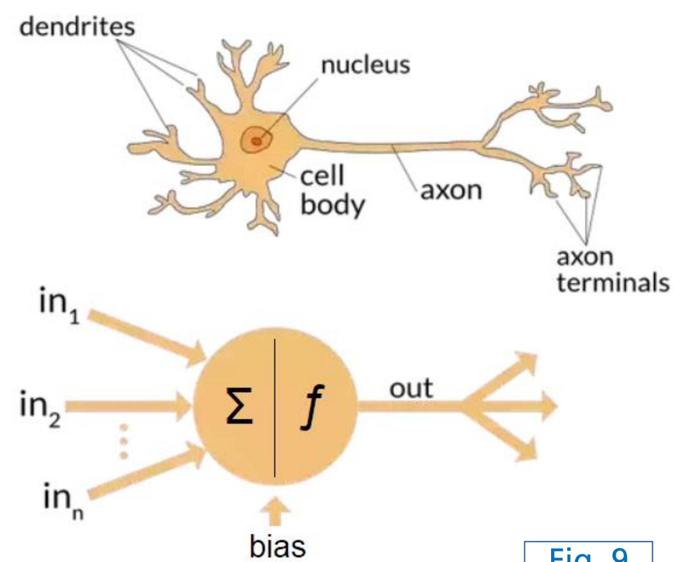


Fig. 9

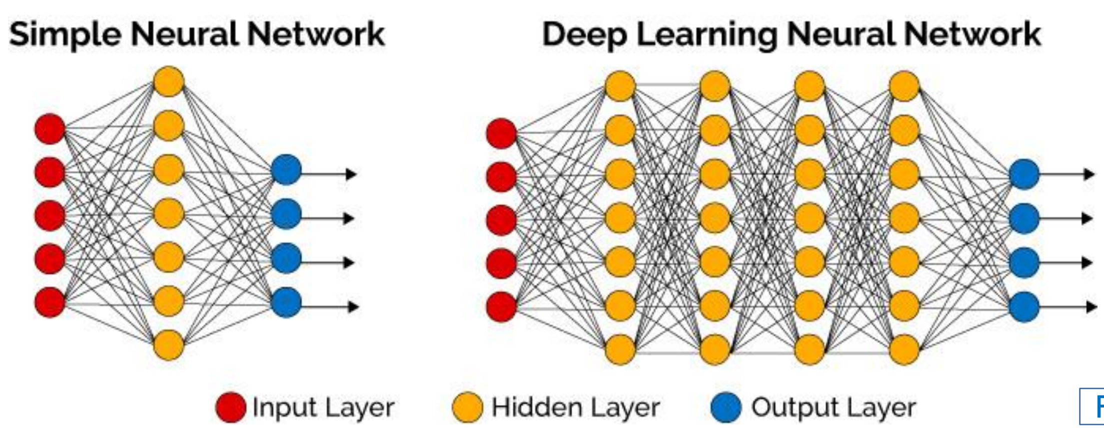
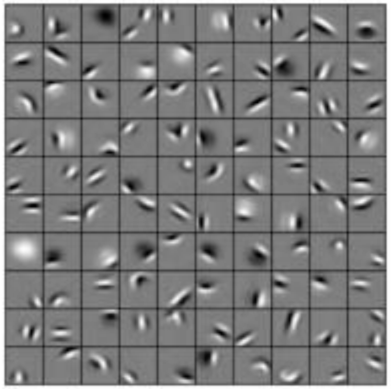


Fig. 10

Deep Learning
Introduction

- Hierarchical feature learning
  - Example (face recognition)



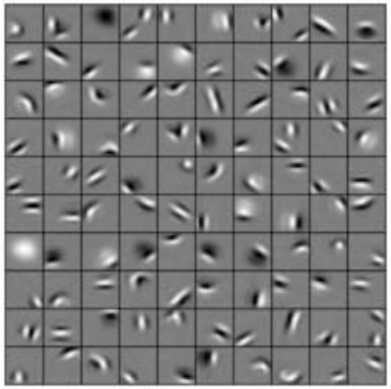
Edges, curves

Fig. 11

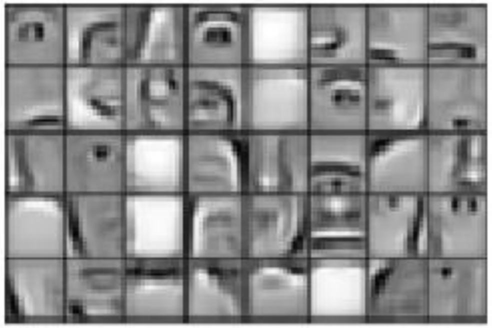
First layers → Final layers

Deep Learning
Introduction

- Hierarchical feature learning
  - Example (face recognition)



Edges, curves



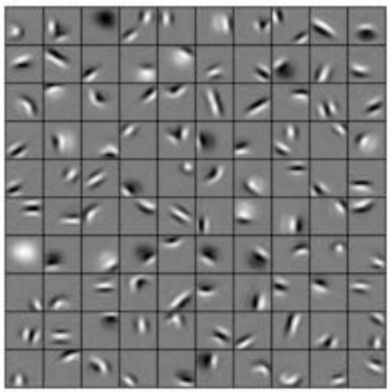
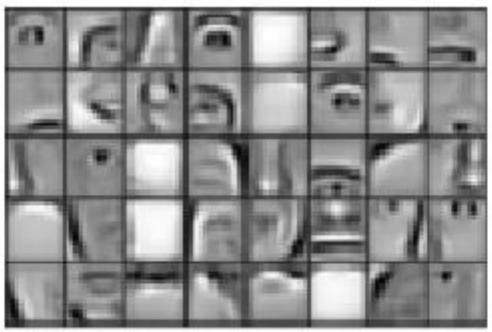
Shapes, object parts

Fig. 11

First layers → Final layers

Deep Learning
Introduction

- Hierarchical feature learning
  - Example (face recognition)


Edges, curves

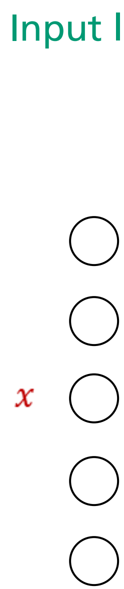
Shapes, object parts

Objects (faces)

Fig. 11

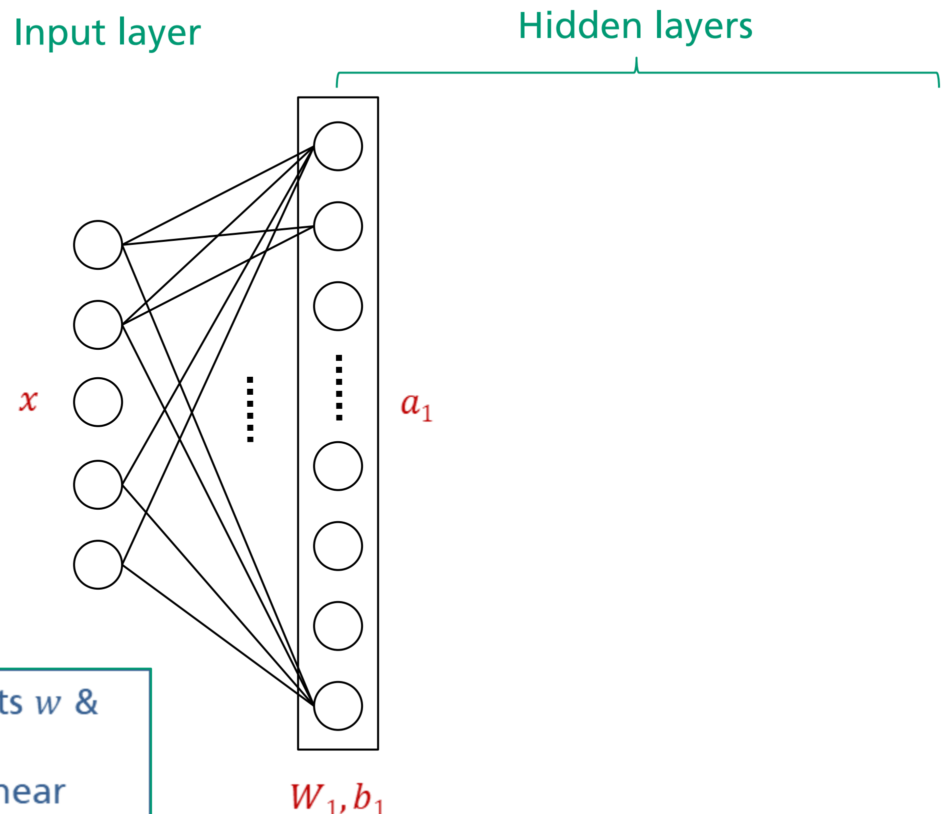
First layers → Final layers

Deep Learning
Fully-connected (Deep) Neural Networks



Own

Deep Learning
Fully-connected (Deep) Neural Networks

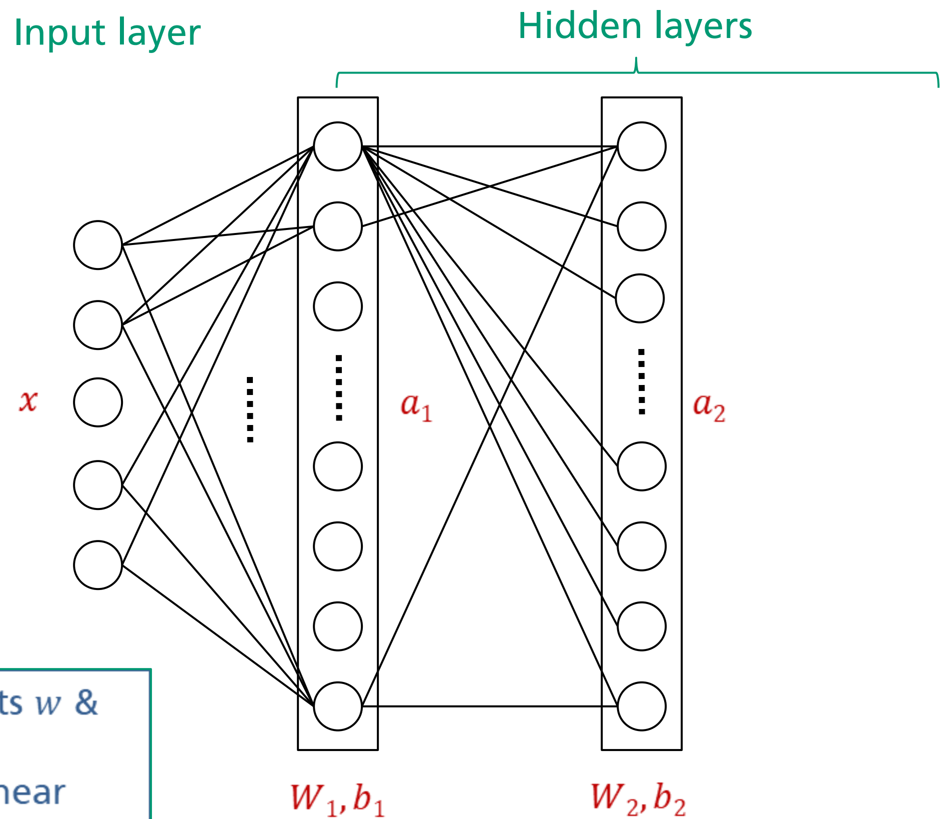


$z_1 = W_1 x + b_1$
$a_1 = g(z_1)$

1) Weights w & biases b
2) Non-linear activation function $g(z)$

Own

Deep Learning
Fully-connected (Deep) Neural Networks



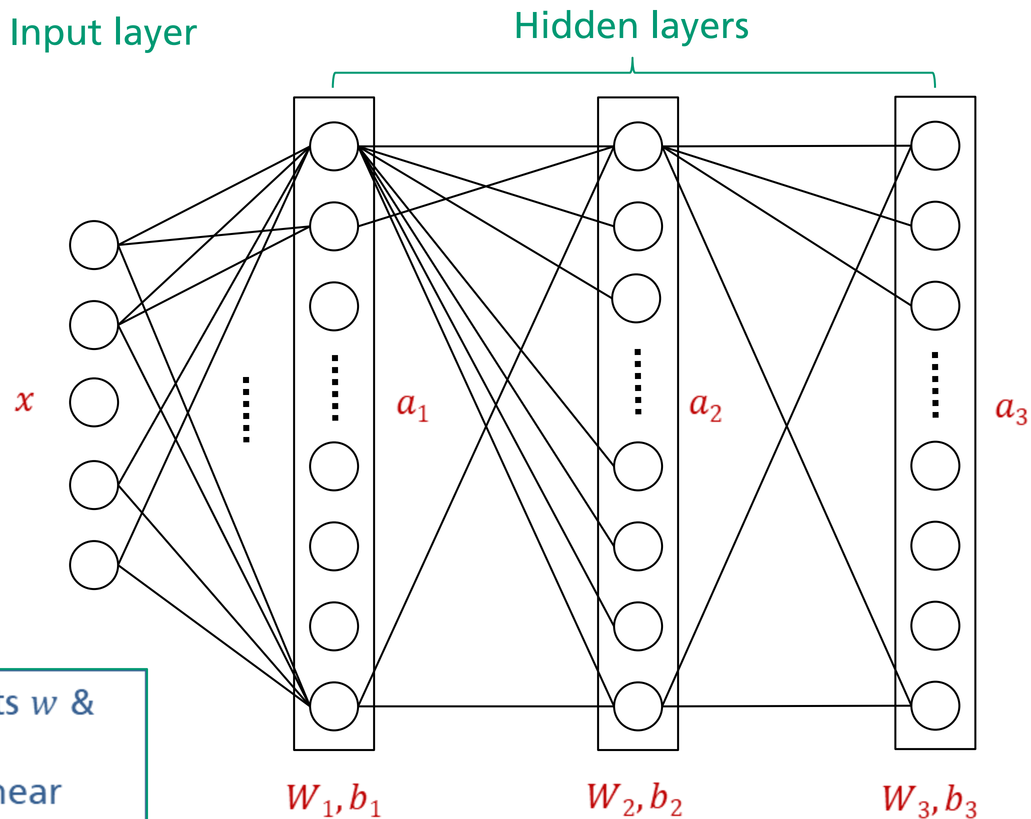
1) Weights w & biases b
2) Non-linear activation function $g(z)$

$z_1 = W_1 x + b_1$ $\quad$ $z_2 = W_2 a_1 + b_2$

$a_1 = g(z_1)$ $\quad$ $a_2 = g(z_2)$

Own

Deep Learning
Fully-connected (Deep) Neural Networks



1) Weights w & biases b
2) Non-linear activation function $g(z)$

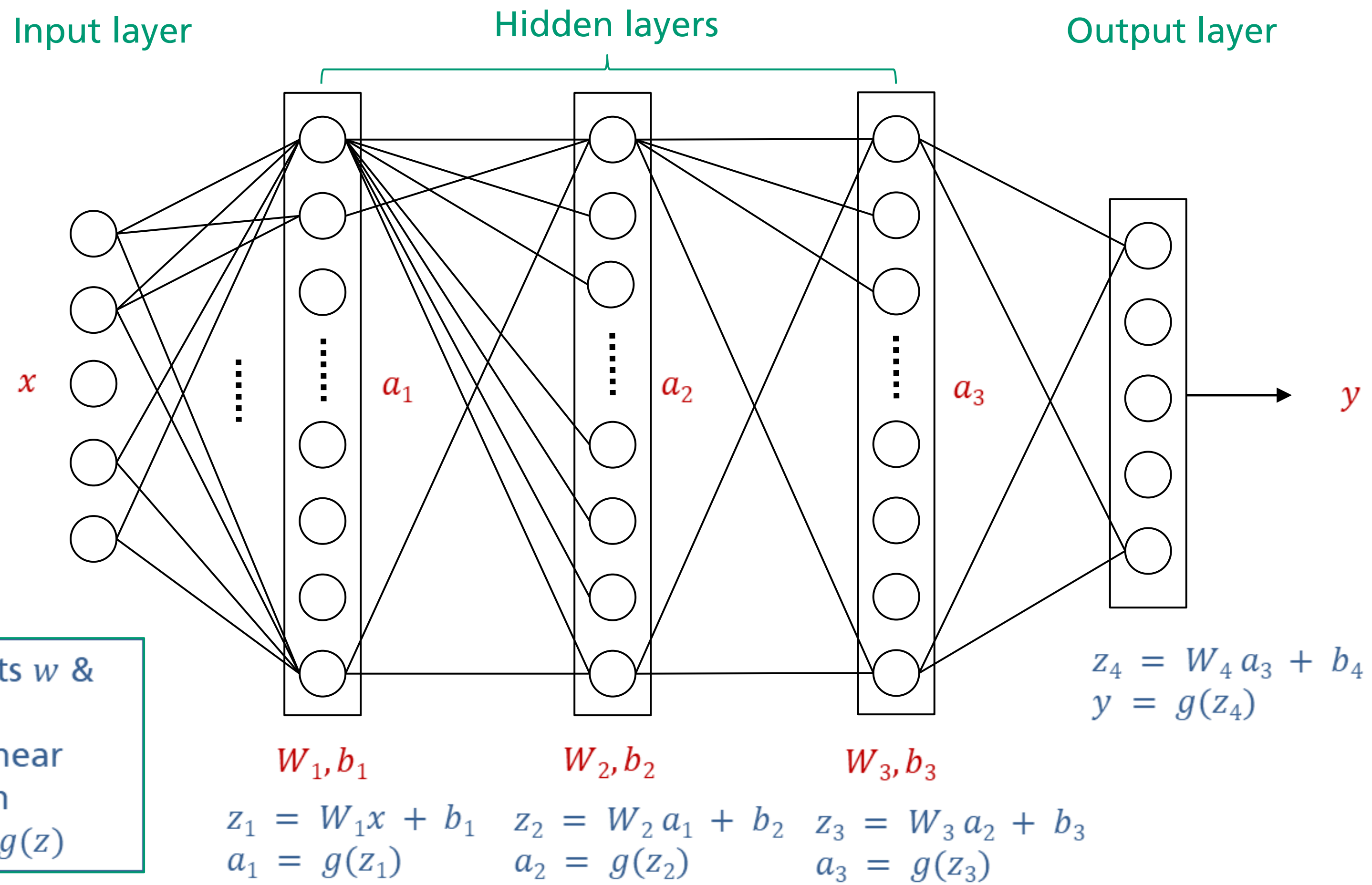
$z_1 = W_1 x + b_1$
$a_1 = g(z_1)$

$z_2 = W_2 a_1 + b_2$
$a_2 = g(z_2)$

$z_3 = W_3 a_2 + b_3$
$a_3 = g(z_3)$

Own

Deep Learning
Fully-connected (Deep) Neural Networks



1) Weights w & biases b
2) Non-linear activation function $g(z)$

Own

Deep Learning
Activation Functions

- Activation functions add non-linearity
- Make networks more powerful in (complex) pattern recognition
- Examples:

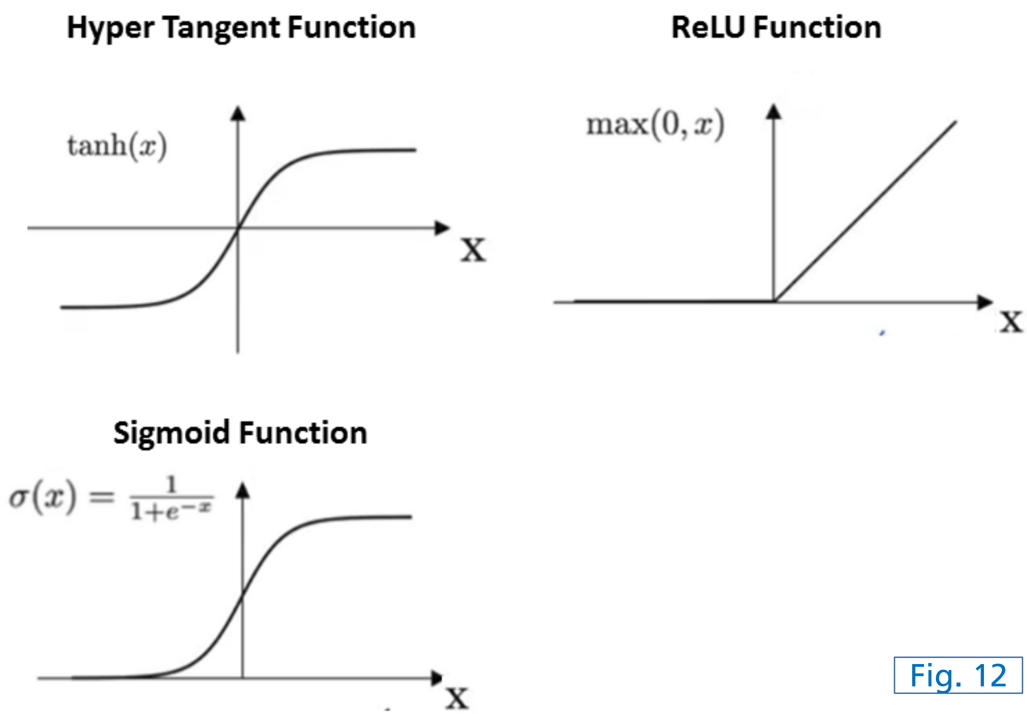


Fig. 12

Deep Learning
Training

- Overview

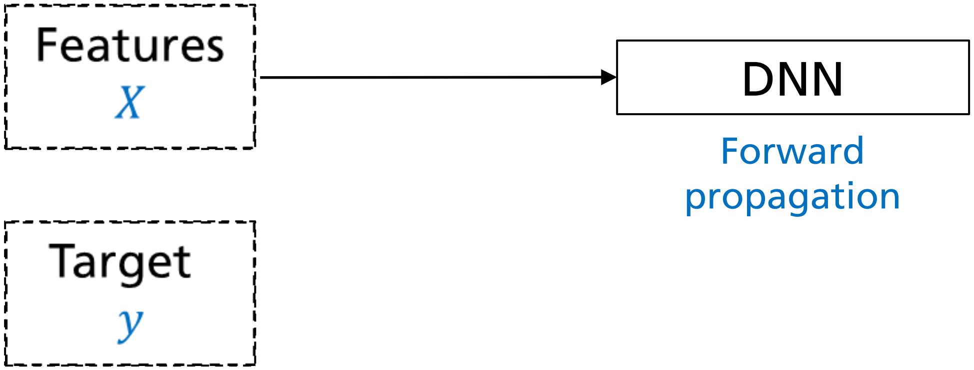
Features
X

Target
y

Own

Deep Learning
Training

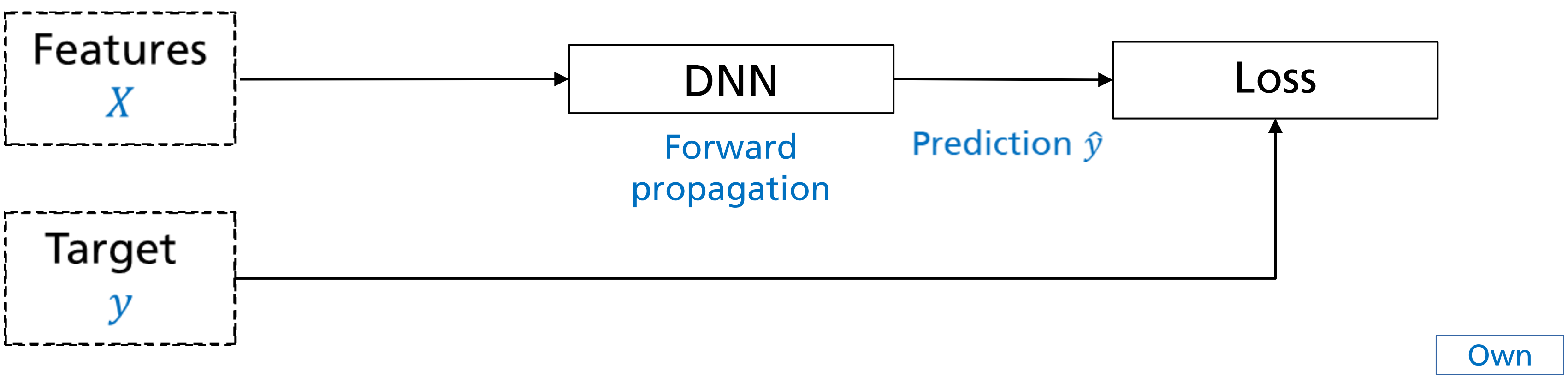
- Overview



Own

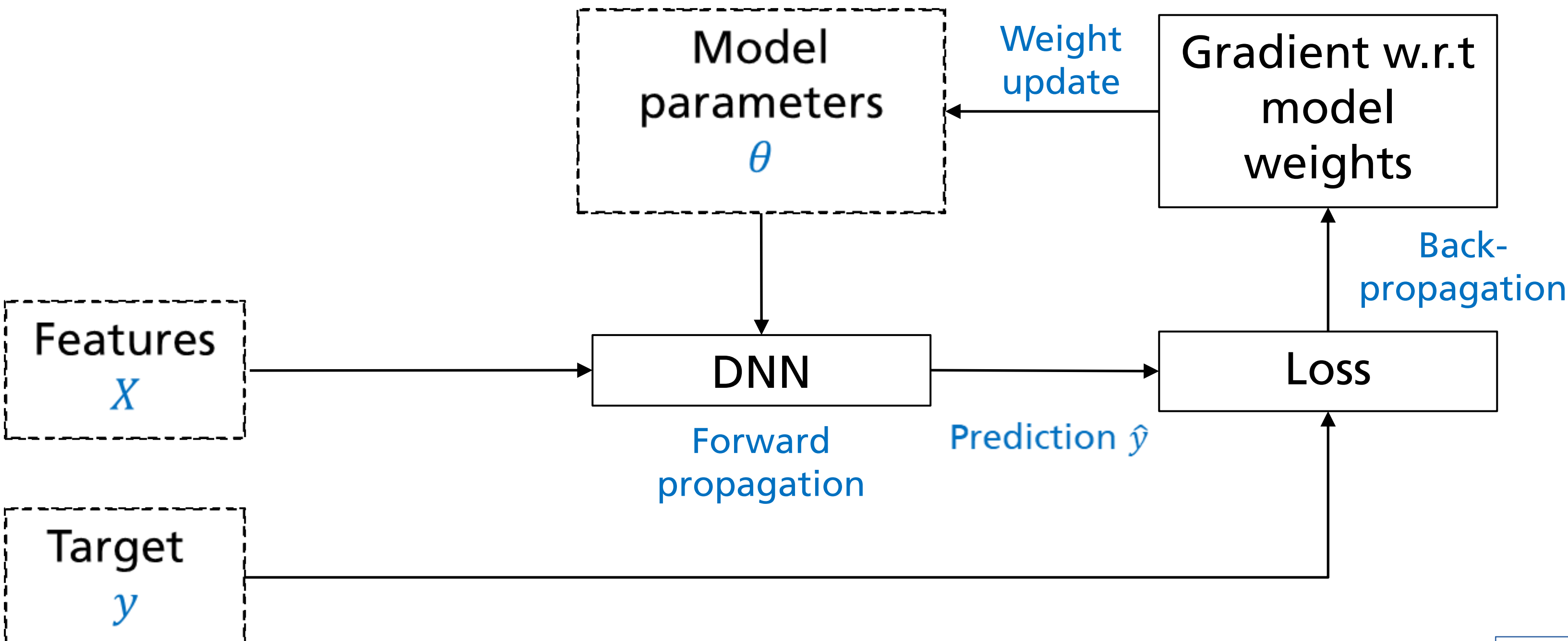
Deep Learning
Training

- Overview



Deep Learning
Training

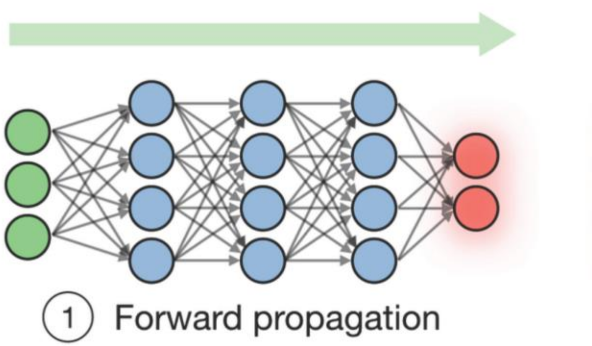
- Overview



Own

Deep Learning
Training

- Forward propagation → propagate batch of training data through the network → compute loss (compare to targets)

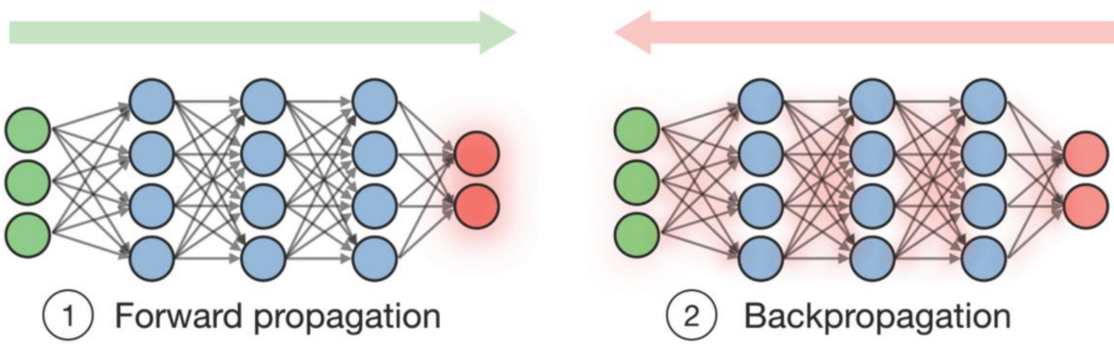


$$\boxed{L(z,y) = -\left[y\log(z) + (1-y)\log(1-z)\right]}$$

Fig. 20

Deep Learning
Training

- Forward propagation → propagate batch of training data through the network → compute loss (compare to targets)
- Backpropagation → backpropagate loss → compute gradients of loss w.r.t. weights



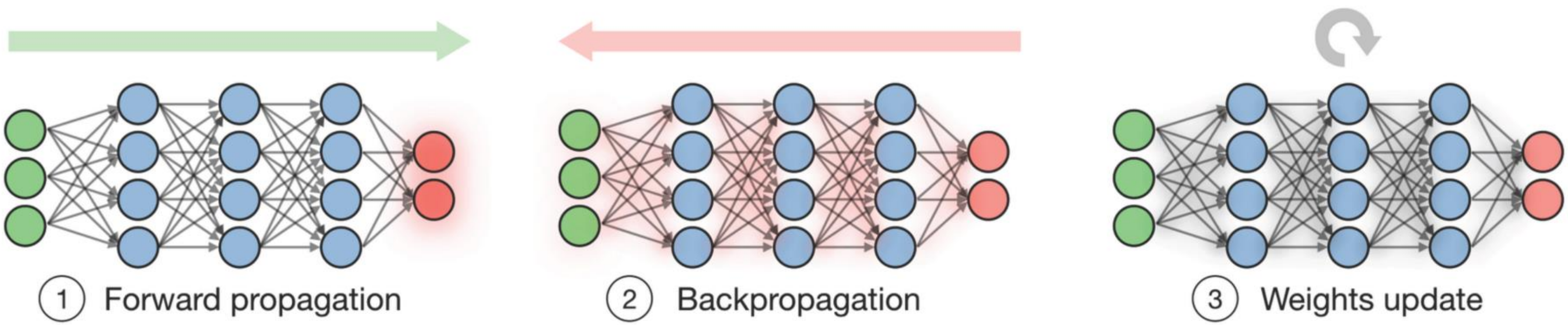
$$\boxed{L(z,y) = -\left[y\log(z) + (1-y)\log(1-z)\right]}$$

$$\boxed{\frac{\partial L(z,y)}{\partial w}}$$

Fig. 20

Deep Learning
Training

- Forward propagation → propagate batch of training data through the network → compute loss (compare to targets)
- Backpropagation → backpropagate loss → compute gradients of loss w.r.t. weights
- Weights update → use gradients & learning rate to update weights



$$\boxed{L(z,y) = -\left[y\log(z) + (1-y)\log(1-z)\right]}$$

$$\boxed{\frac{\partial L(z,y)}{\partial w}}$$

$$\boxed{w \longleftarrow w - \alpha\frac{\partial L(z,y)}{\partial w}}$$

Fig. 20

Deep Learning
Training

- Gradient descent
    - Move in opposite direction of gradient
    - Learning rate effects step size

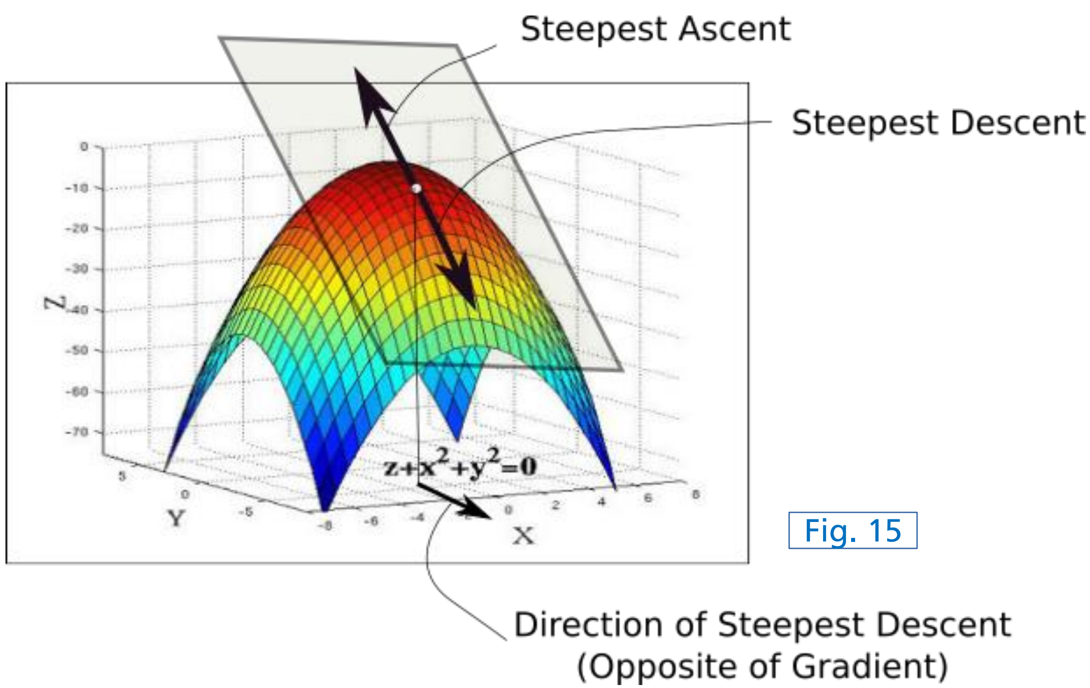


Fig. 15

Deep Learning
Training

- Loss contour
    - Goal → find global minima

Toy example

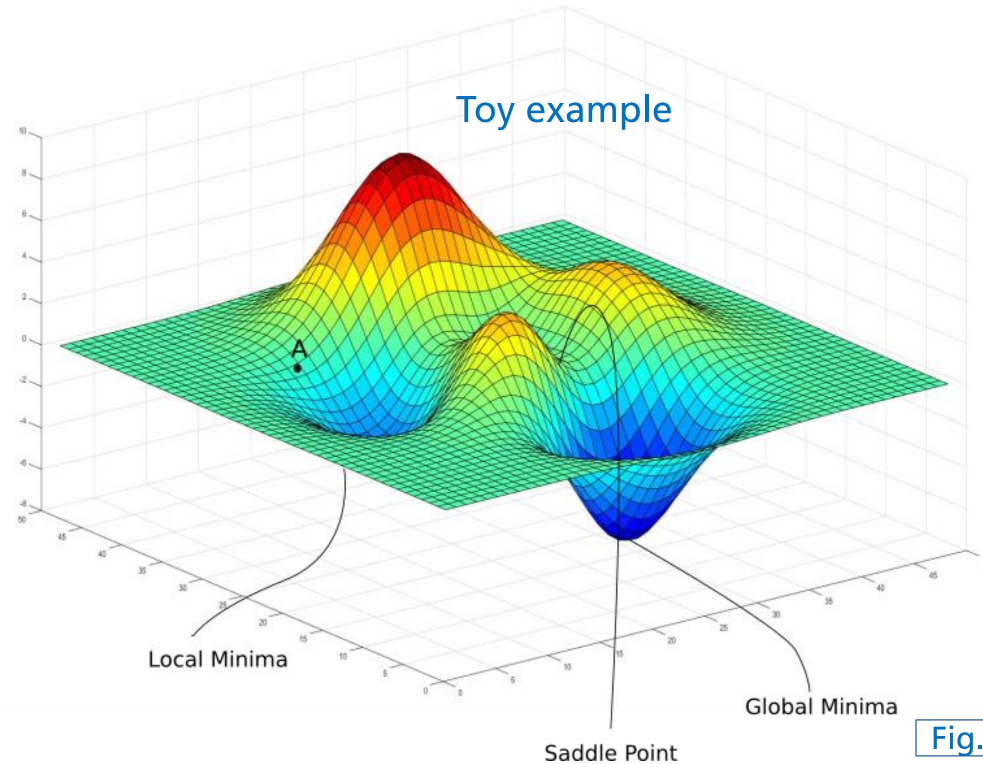


Fig. 13

Complex DNN

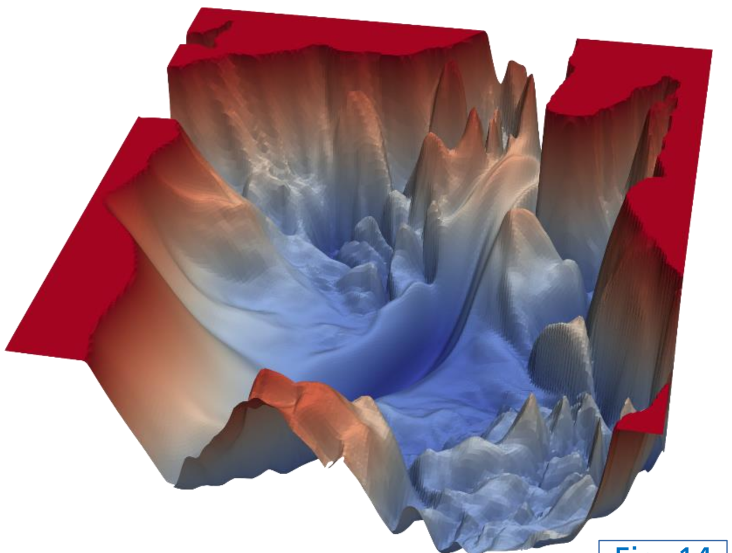


Fig. 14

Deep Learning
Playground

- A neural network playground!
    - https://playground.tensorflow.org

Deep Learning
Convolutional Neural Networks (CNN)

- Convolutional layers
    - “Convolution” → (local) dot-product between filter and input

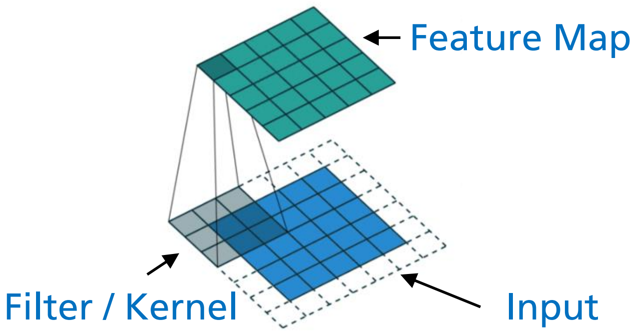


Fig. 16

Deep Learning
Convolutional Neural Networks (CNN)

- Convolutional layers
  - "Convolution" → (local) dot-product between filter and input
  - Shared weights (across input)

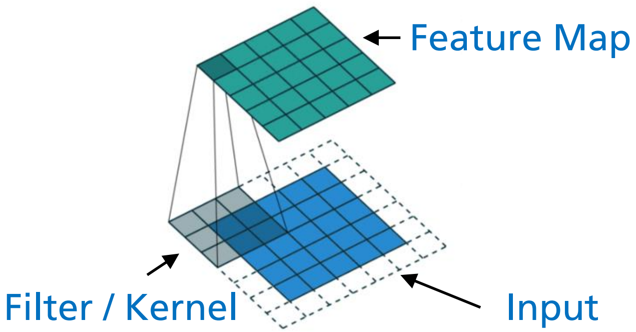


Fig. 16

Deep Learning
Convolutional Neural Networks (CNN)

- Convolutional layers
    - “Convolution” → (local) dot-product between filter and input
    - Shared weights (across input)
    - translation of input → translation of activations (equivariance)

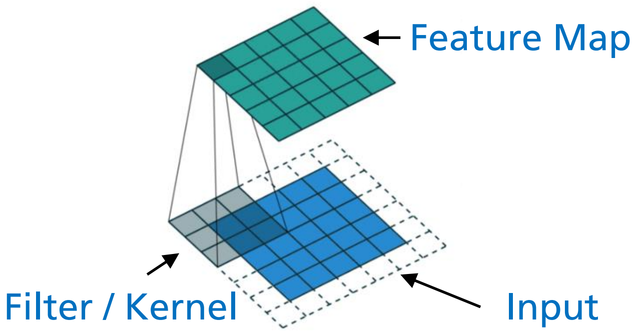


Fig. 16

Deep Learning
Convolutional Neural Networks (CNN)

- Convolutional layers
  - “Convolution” → (local) dot-product between filter and input
  - Shared weights (across input)
  - translation of input → translation of activations (equivariance)
- Pooling → local aggregation / down-sampling

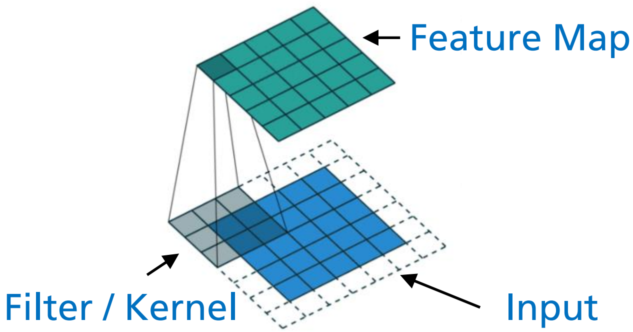


Fig. 16

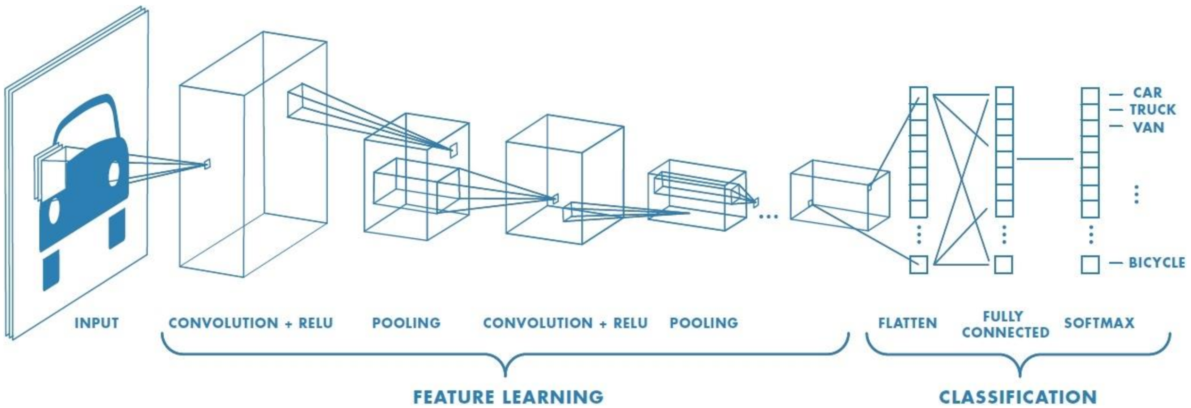


Fig. 17

Audio Processing
Programming Session

Fig. 2.1

References

Introducing Machine Learning. (2016). Retrieved from https://www.mathworks.com/content/dam/mathworks/tag-team/Objects/i/88174_92991v00_machine_learning_section1_ebook.pdf

S. Legg, M. Hutter (2007). Universal Intelligence: A Definition of Machine Intelligence. Minds & Machines. 17 (4): 391-444.

L. Samuel (1959). Some studies in machine learning using the game of checkers. IBM Journal of research and development. 3(3), 210-229

Srihari, S. N. (2020). *Forward Propagation and Backward Propagation (Deep Learning Lecture)*. Retrieved from https://cedar.buffalo.edu/~srihari/CSE676/6.5.0 Forward Backward.pdf

Virtanen, T., Plumbley, M. D., & Ellis, D. (Eds.). (2018). *Computational Analysis of Sound Scenes and Events*. Cham, Switzerland: Springer International Publishing.

Images

Fig. 1: [Machine Learning, 2016], p. 4, Fig. 2

Fig. 2: https://i0.wp.com/www.sthda.com/sthda/RDoc/figure/clustering/ partitioning-cluster-analysis-k-means-plot-4-groups-1.png

Fig. 3: https://i.stack.imgur.com/hsiIO.png (https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html)

Fig. 4: https://miro.medium.com/max/975/1*OyYyr9qY-w8RkaRh2TKo0w.png (reproduced)

Fig. 5: https://lilianweng.github.io/lil-log/assets/images/self-sup-lecun.png

Fig. 6: https://www.asimovinstitute.org/wp-content/uploads/2019/04/NeuralNetworkZoo20042019.png

Fig. 7: https://www.educative.io/api/edpresso/shot/6668977167138816/image/5033807687188480

Fig. 8: [Virtanen, 2018], p. 170, Fig. 6.7

Fig. 9: https://miro.medium.com/max/915/1*SJPacPhP4KDEB1AdhOFy_Q.png

Fig. 10: https://www.skampakis.com/wp-content/uploads/2018/03/simple_neural_network_vs_deep_learning.jpg

Fig. 11: https://pic4.zhimg.com/80/v2-057b248288a8af2f01272a956f862873_1440w.png

Fig. 12: https://blog.e-kursy.it/deeplearning4j-workshop/video/html/presentation_specific/img/4_activation_functions.png

Images

Fig. 13: https://blog.paperspace.com/content/images/2018/05/challenges-1.png

Fig. 14: https://www.cs.umd.edu/~tomg/img/landscapes/noshort.png

Fig. 15: https://blog.paperspace.com/content/images/2018/05/grad.png

Fig. 16: https://www.wandb.com/articles/intro-to-cnns-with-wandb

Fig. 17: https://www.freecodecamp.org/news/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050/

Fig. 18: https://wiki.tum.de/download/attachments/22578349/RNN1.png

Fig. 19: https://stanford.edu/~shervine/teaching/cs-230/illustrations/architecture-rnn-ltr.png

Fig. 20: [Srihari, 2020], p.8, (Fig. 1)

Images

Fig. 1:

References

[1] Sternberg, R. J. (2022). human intelligence. Encyclopedia Britannica. https://www.britannica.com/science/human-intelligence-psychology

[2] Gross, R., Psychology (2015). The Science of Mind and Behaviour, Hodder Education

[3] Legg, S., Hutter, M. (2007). Universal Intelligence: A Definition of Machine Intelligence. Minds & Machines 17, 391–444

[4] Russell, S., Norvig, P. (2016). Artificial Intelligence: A Modern Appoach, **PEV**, third ed.

[5] Koza, J. R., Bennett, F. H., Andre, D., Keane, M. A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. Artificial Intelligence in Design '96. Springer. pp. 151–170.

Audio

[Audio 1] https://freesound.org/people/xserra/sounds/196765/

[Audio 2] https://freesound.org/people/IliasFlou/sounds/498058/ (~0:00 – 0:05)

[Audio 3] https://freesound.org/people/danlucaz/sounds/517860/ (~0:00 – 0:05)

[Audio 4] https://freesound.org/people/IENBA/sounds/489398/ (~0:00 – 0:07)