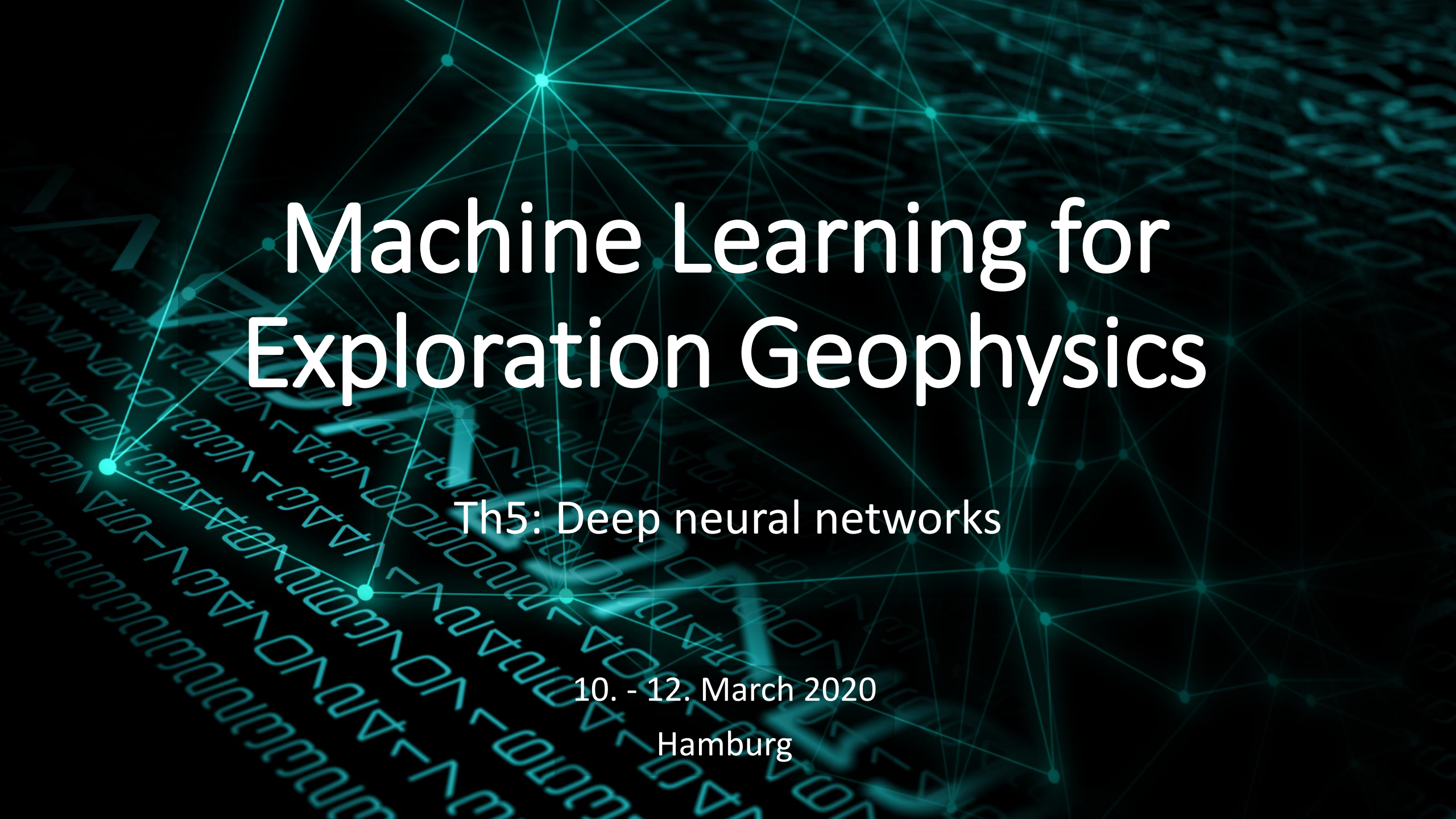


Machine Learning for Exploration Geophysics

Th5: Deep neural networks

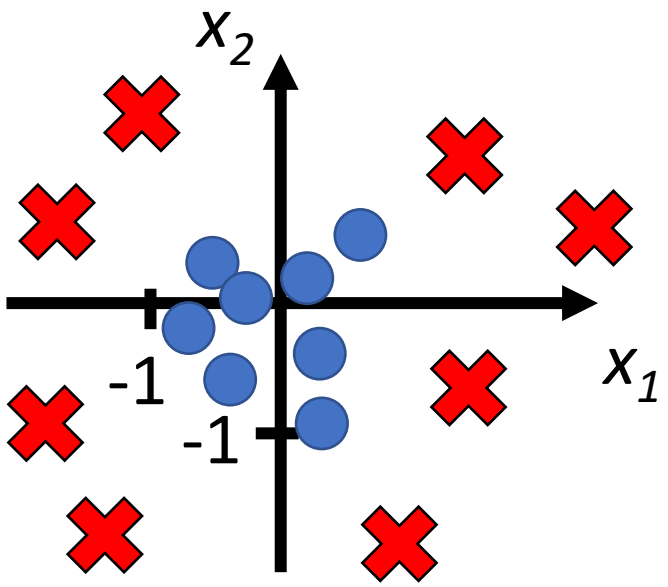
10. - 12. March 2020

Hamburg

Outline

- What is a Neural Network?
- Network architectures
- Activation functions
- Optimization algorithm
- Convolutional neural network

Non-linear Classification



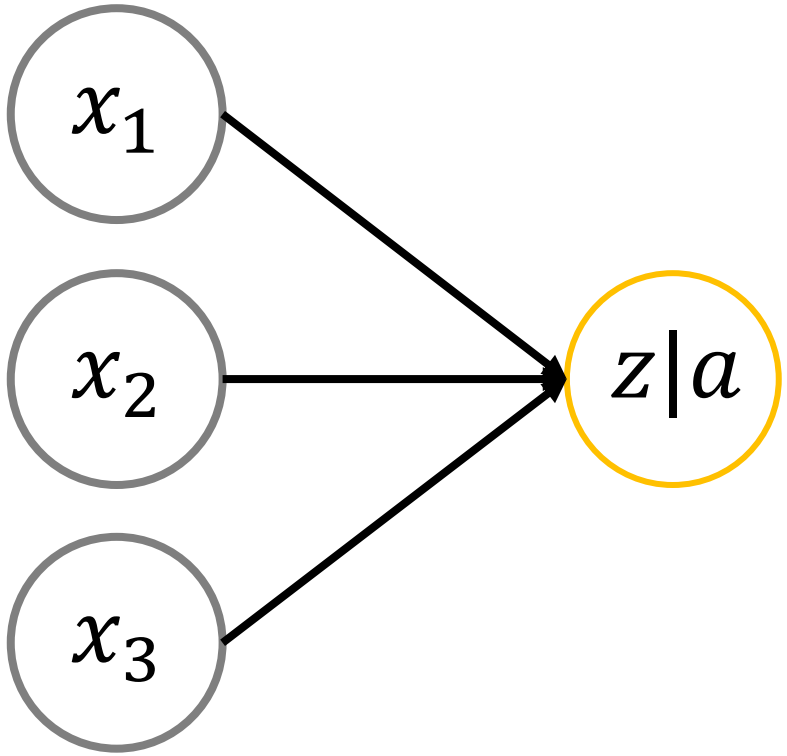
$$y_w(x) = \sigma(b + w_1x_1 + w_2x_2 + w_3x_3)$$

$$y_w(x) = \sigma\begin{pmatrix} b + w_1x_1 + w_2x_2 + w_3x_3 \\ w_4x_1^2 + w_5x_1x_2 + w_6x_2^2 + \cdots \end{pmatrix}$$

$x_1, x_2, \ldots x_{100}$

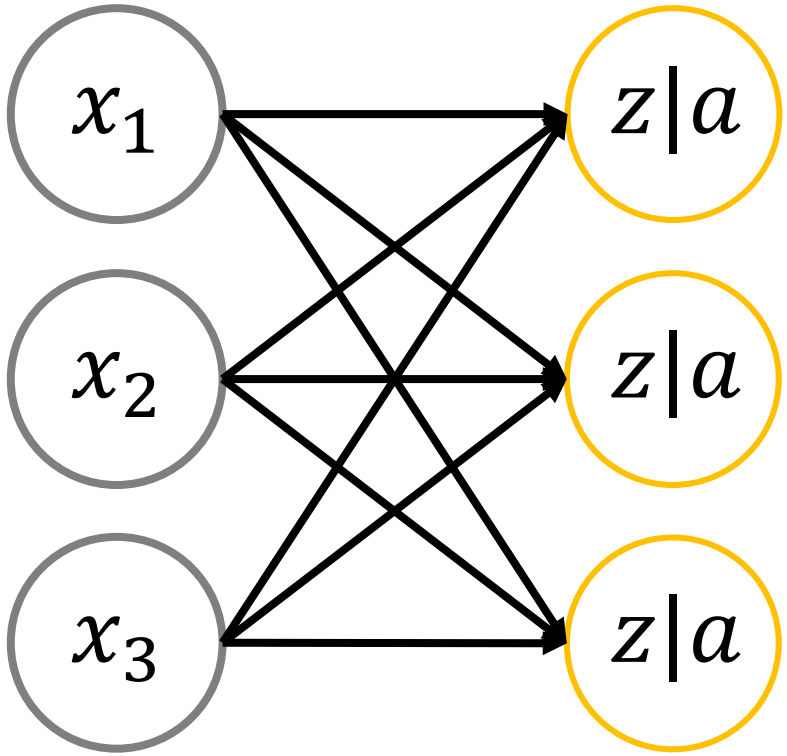
Non-linear Classification

$$y_w(x) = \sigma(b + w_1 x_1 + w_2 x_2 + w_3 x_3)$$



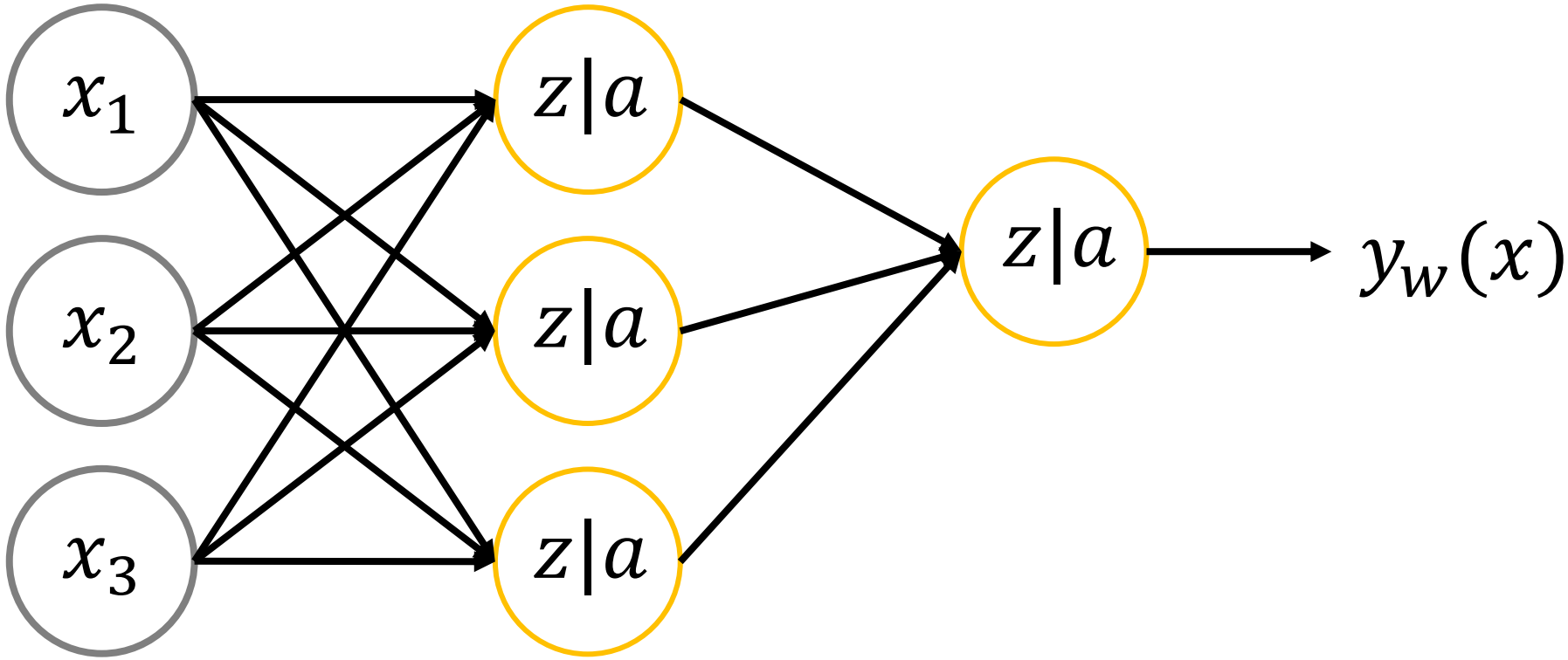
Non-linear Classification

$$y_w(x) = \sigma(b + w_1 x_1 + w_2 x_2 + w_3 x_3)$$

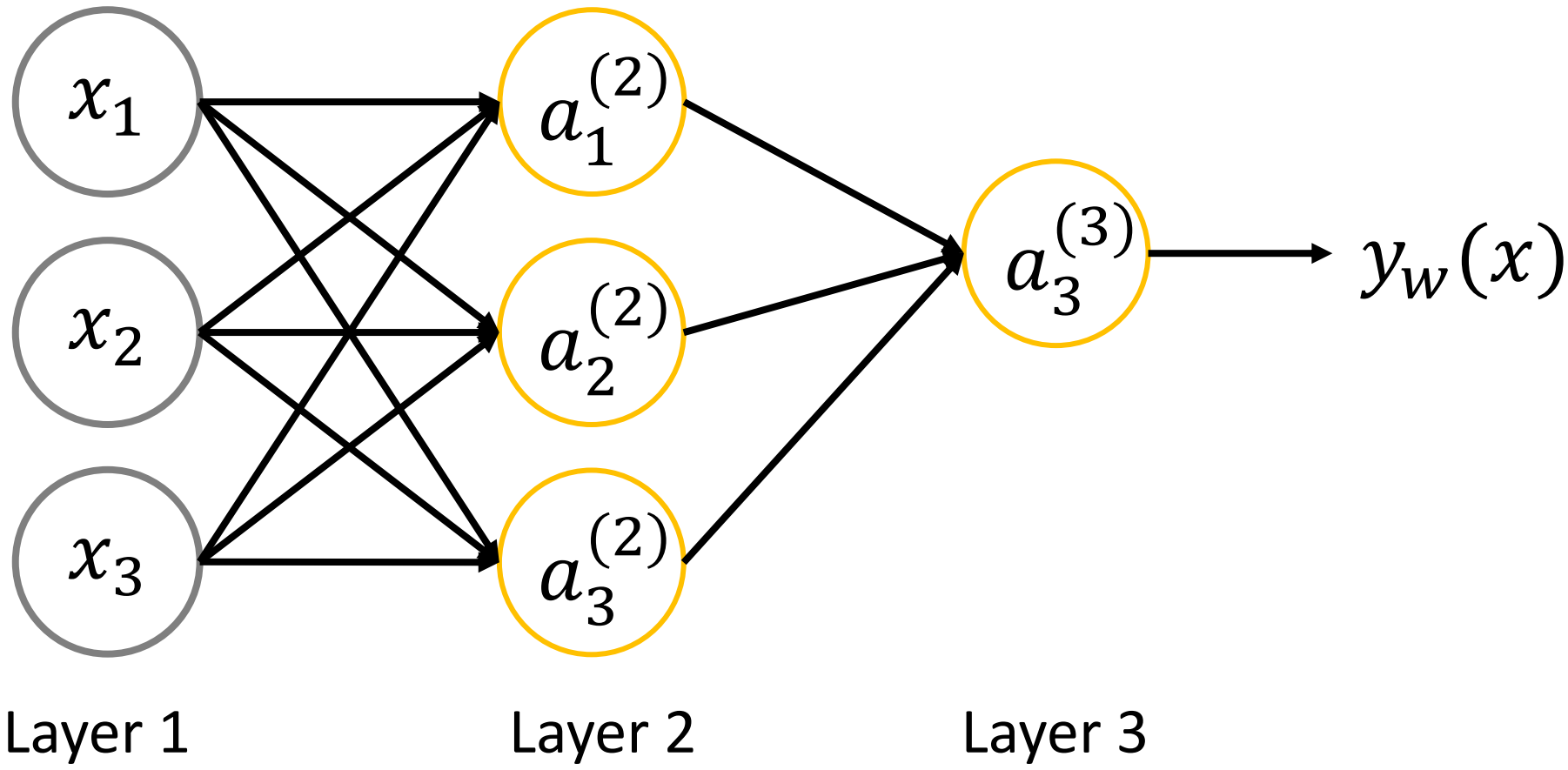


Non-linear Classification

$$y_w(x) = \sigma(b + w_1x_1 + w_2x_2 + w_3x_3)$$



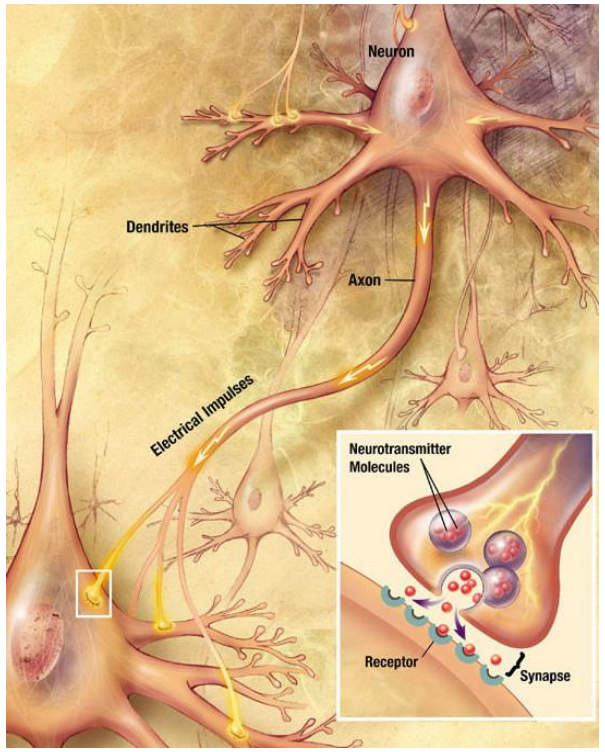
Neural Network

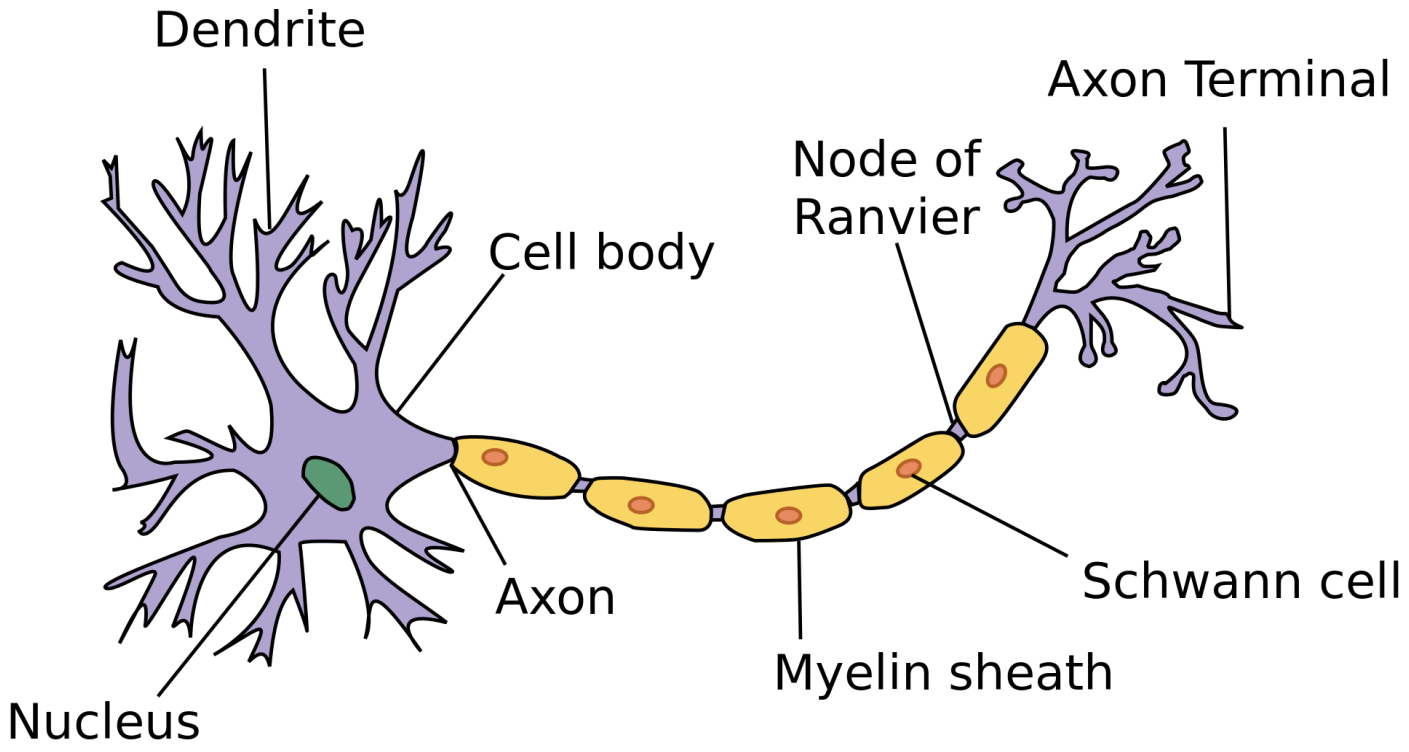


Neural Network

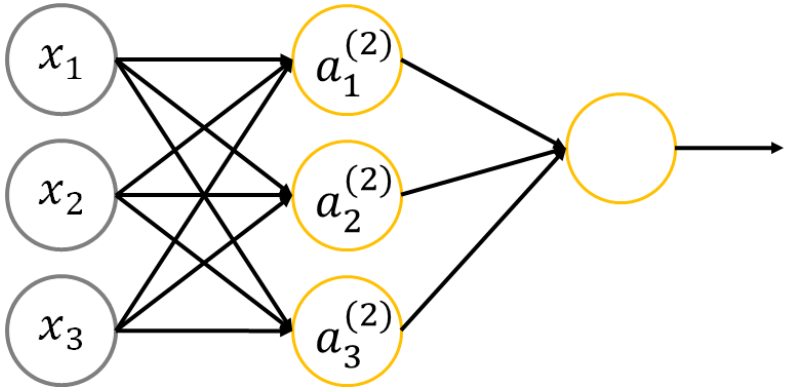
- Origins: Algorithms that try to mimic the brain
- Was very widely used in the 80s and early 90s; popularity diminished in late 90s
- Now: state of the art technique for many applications

Neurons in the brain





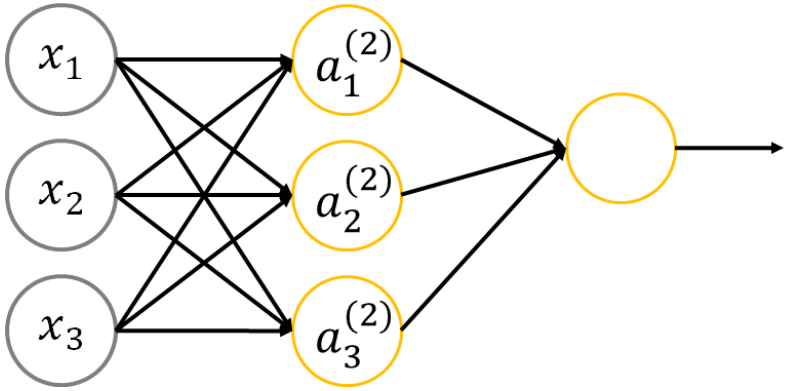
Neural Network



$$a_1^{(2)} = g\left(\mathrm{w}_{10}^{(1)} x_0 + \mathrm{w}_{11}^{(1)} x_1 + \mathrm{w}_{12}^{(1)} x_2 + \mathrm{w}_{13}^{(1)} x_3\right)$$
$$a_2^{(2)} = g\left(w_{20}^{(1)} x_0 + w_{21}^{(1)} x_1 + w_{22}^{(1)} x_2 + w_{23}^{(1)} x_3\right)$$
$$a_3^{(2)} = g\left(w_{30}^{(1)} x_0 + w_{31}^{(1)} x_1 + w_{32}^{(1)} x_2 + w_{33}^{(1)} x_3\right)$$

$$y_w(x) = a_1^{(3)} = g\left(\mathrm{w}_{10}^{(2)} a_0^{(2)} + \mathrm{w}_{11}^{(2)} a_1^{(2)} + \mathrm{w}_{12}^{(2)} a_2^{(2)} + \mathrm{w}_{13}^{(2)} a_3^{(2)}\right)$$

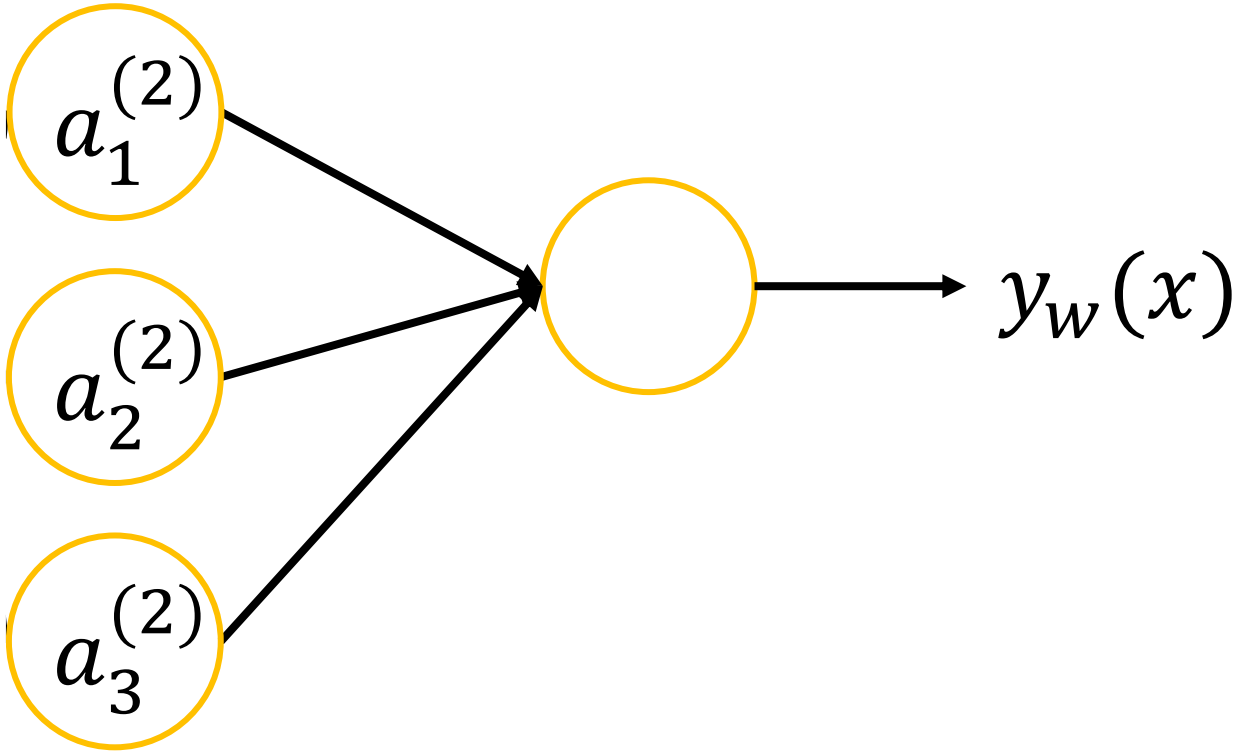
Neural Network



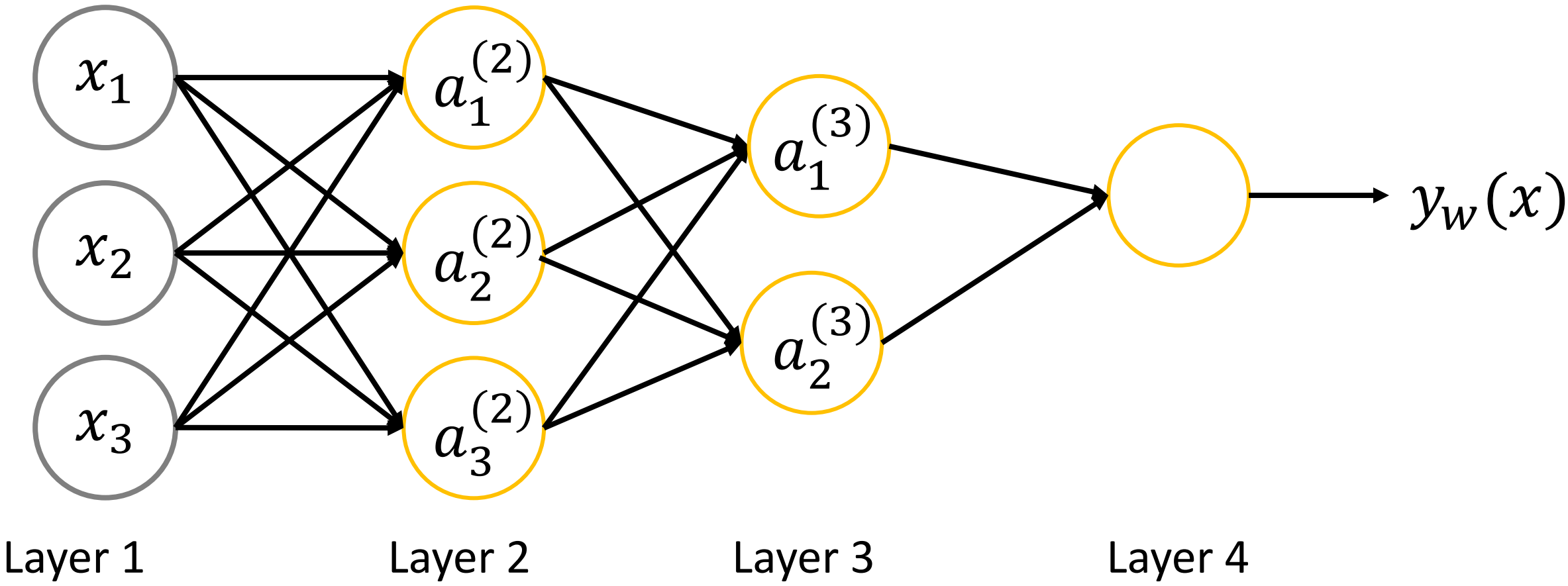
$$\boldsymbol{a}^{(2)} = g\left(W^{(1)}\boldsymbol{x}\right)$$

$$y_w(x) = a_1^{(3)} = g\left(\mathrm{W}^{(2)}\boldsymbol{a}^{(2)}\right)$$

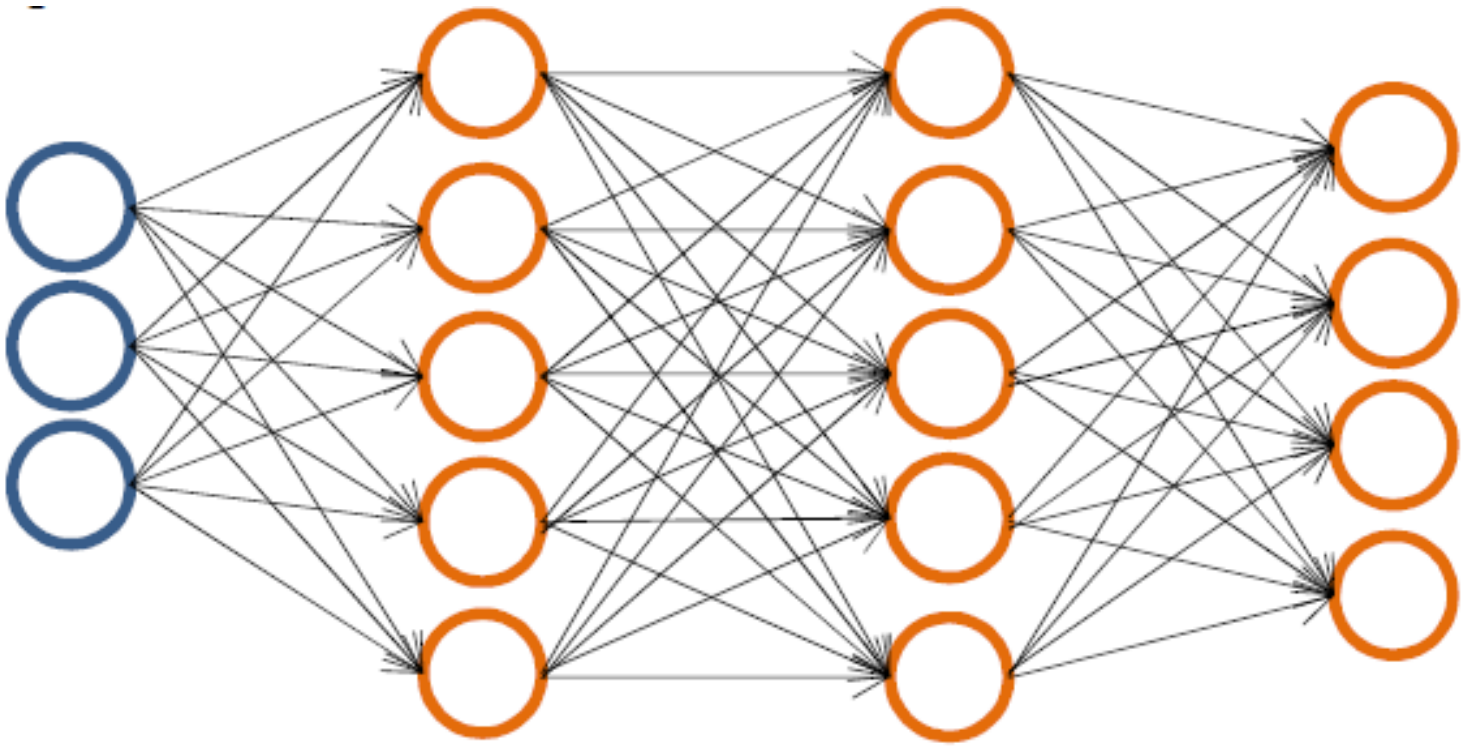
Neural Network learning its own features



Other network architectures



Multiclass classification



Reflection

Diffraction 1

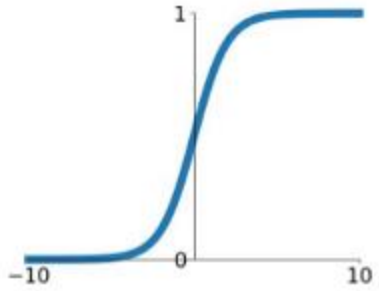
Diffraction 2

Noise

Activation functions

Sigmoid

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



Neural network example

- https://playground.tensorflow.org

Optimization algorithm

- Gradient descent
- Stochastic gradient descent
- Mini Batch Gradient Descent
- Momentum
- RMSprop
- Adam

$$W := W - \alpha\, dW$$

$$V_{dW} = \beta\, V_{dW} + (1-\beta) dW$$

$$W := W - \alpha\, dV_{dW}$$

$$S_{dW} = \beta\, S_{dW} + (1-\beta) dW^2$$

$$W := W - \alpha \frac{dW}{\sqrt{S_{dW}} + \epsilon}$$

Gradient descent

Want: $\min_{b,w} J(b,w)$

1. Start with a guess:

$$b = b_0, \qquad w = w_0$$

2. Calculate:

$$db = \frac{\partial J}{\partial b}, \qquad dw = \frac{\partial J}{\partial w}$$

3. Update:

$$b_{i+1} = b_i - \alpha db, \qquad w_{i+1} = w_i - \alpha dw$$

Optimization algorithm

- Gradient descent
- Stochastic gradient descent
- Mini Batch Gradient Descent
- Momentum
- RMSprop
- Adam

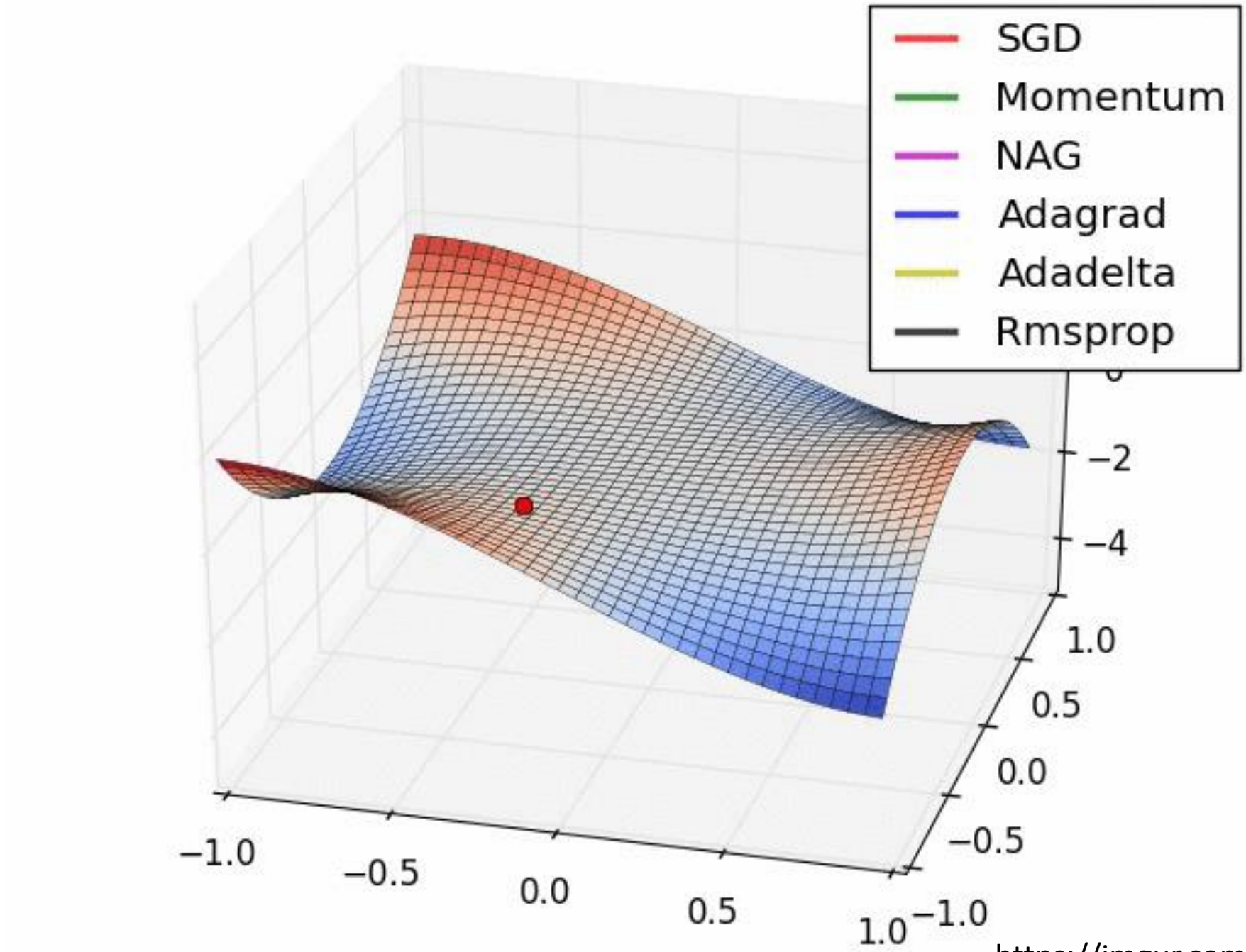
$$W := W - \alpha\, dW$$

$$V_{dW} = \beta\, V_{dW} + (1-\beta) dW$$

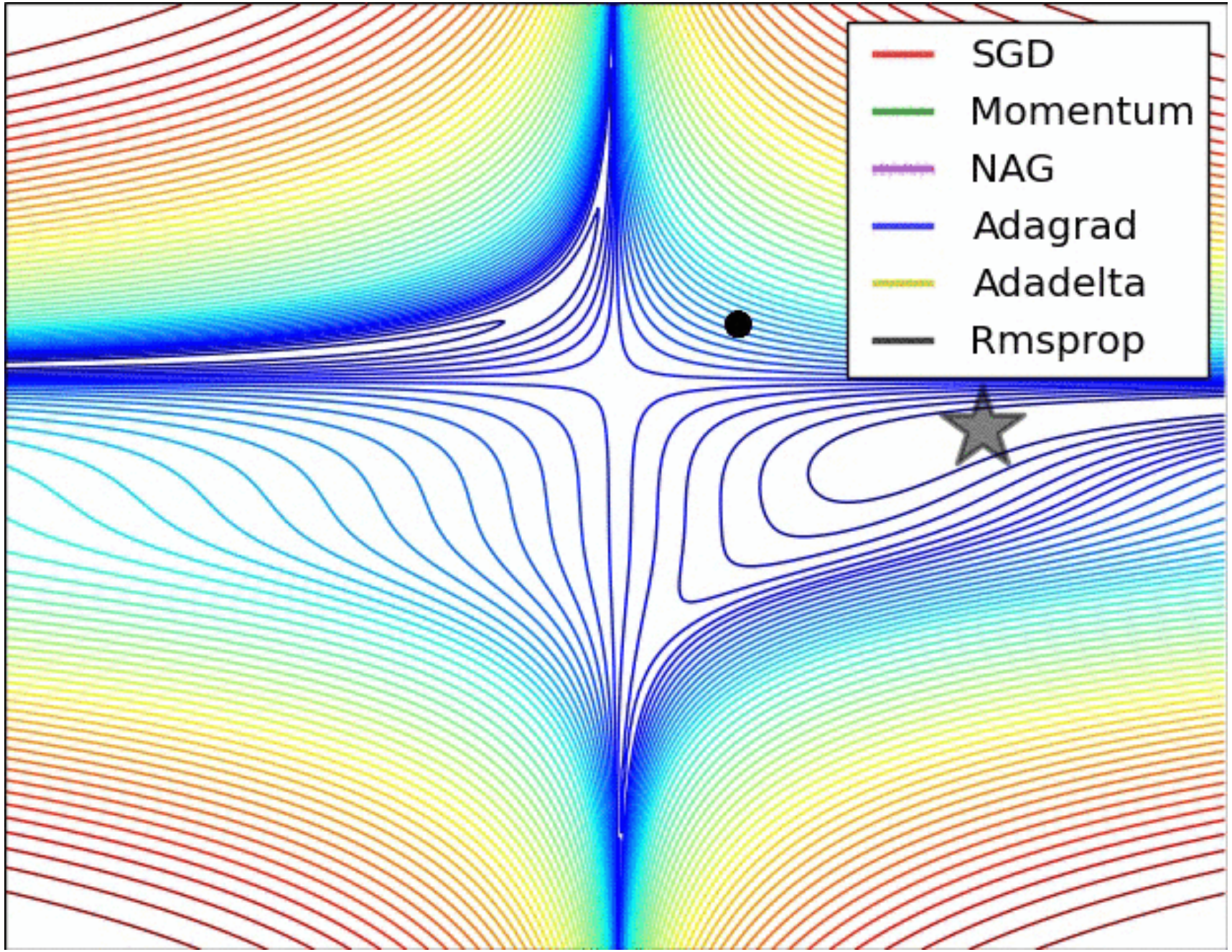
$$W := W - \alpha\, dV_{dW}$$

$$S_{dW} = \beta\, S_{dW} + (1-\beta) dW^2$$

$$W := W - \alpha \frac{dW}{\sqrt{S_{dW}} + \epsilon}$$



Source:
https://imgur.com/a/Hqolp#NKsFHJb



Source: https://imgur.com/a/Hqolp#NKsFHJb

Optimization

- https://deeplearning.ai/ai-notes/optimization/
- https://deeplearning.ai/ai-notes/initialization/

Convolutional neural network

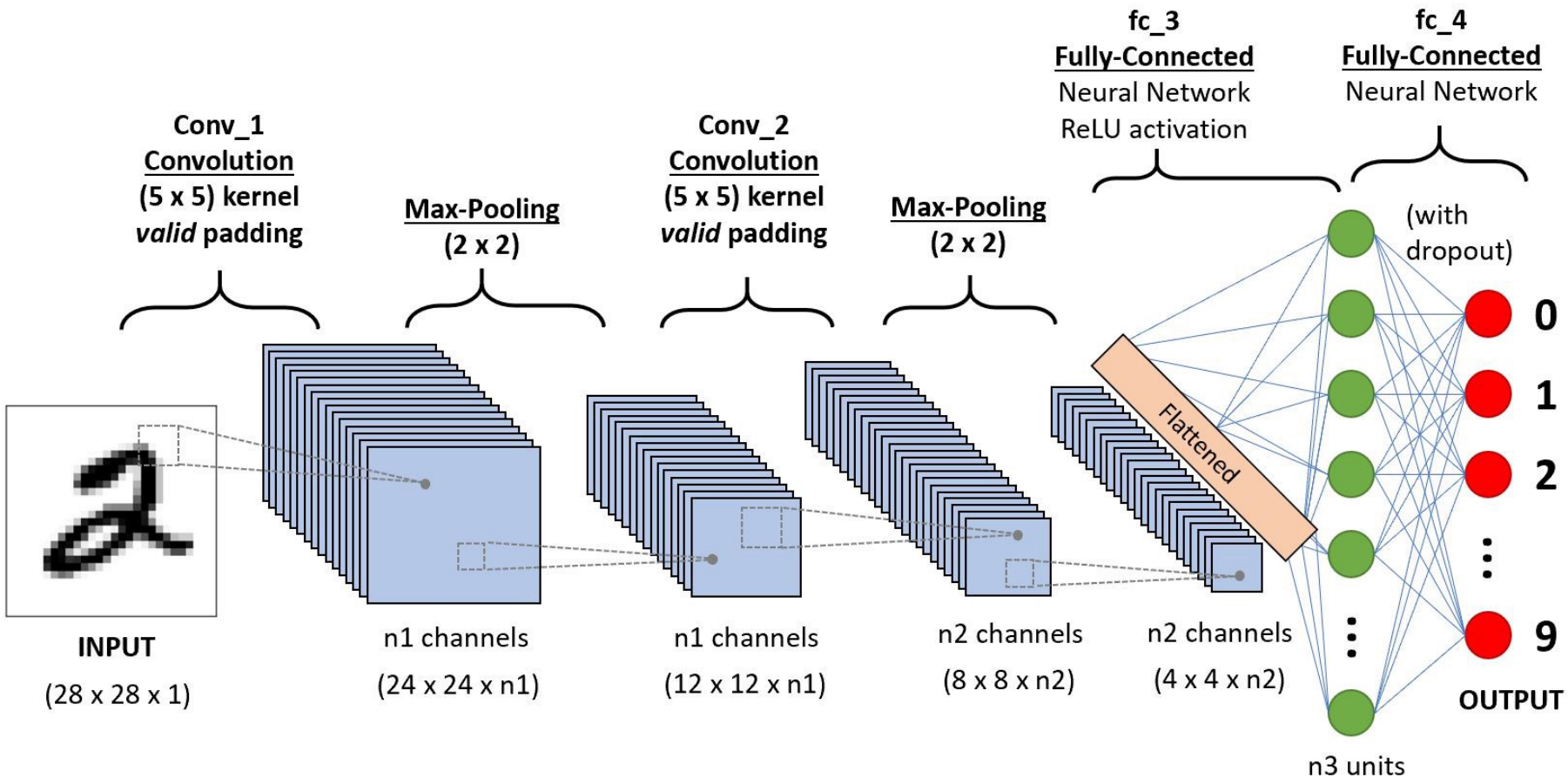


Figure from https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324