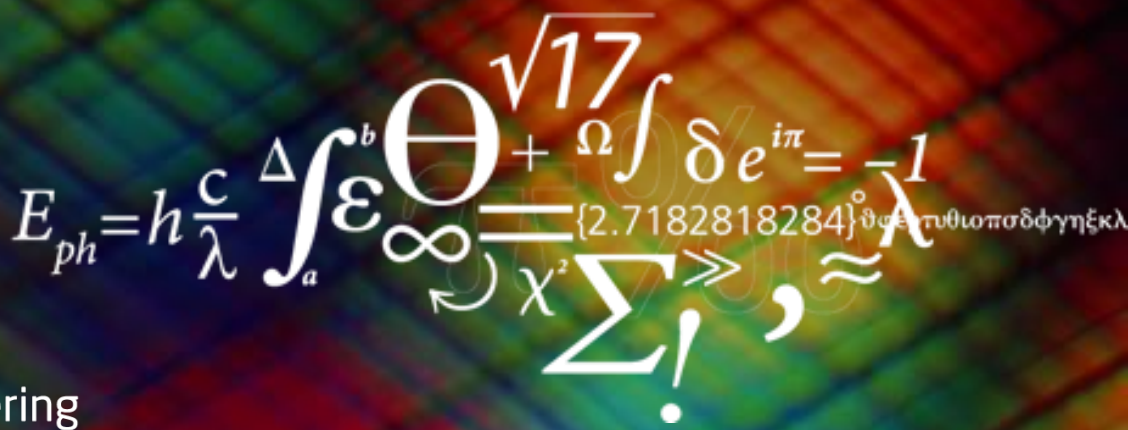
Technical University of Denmark

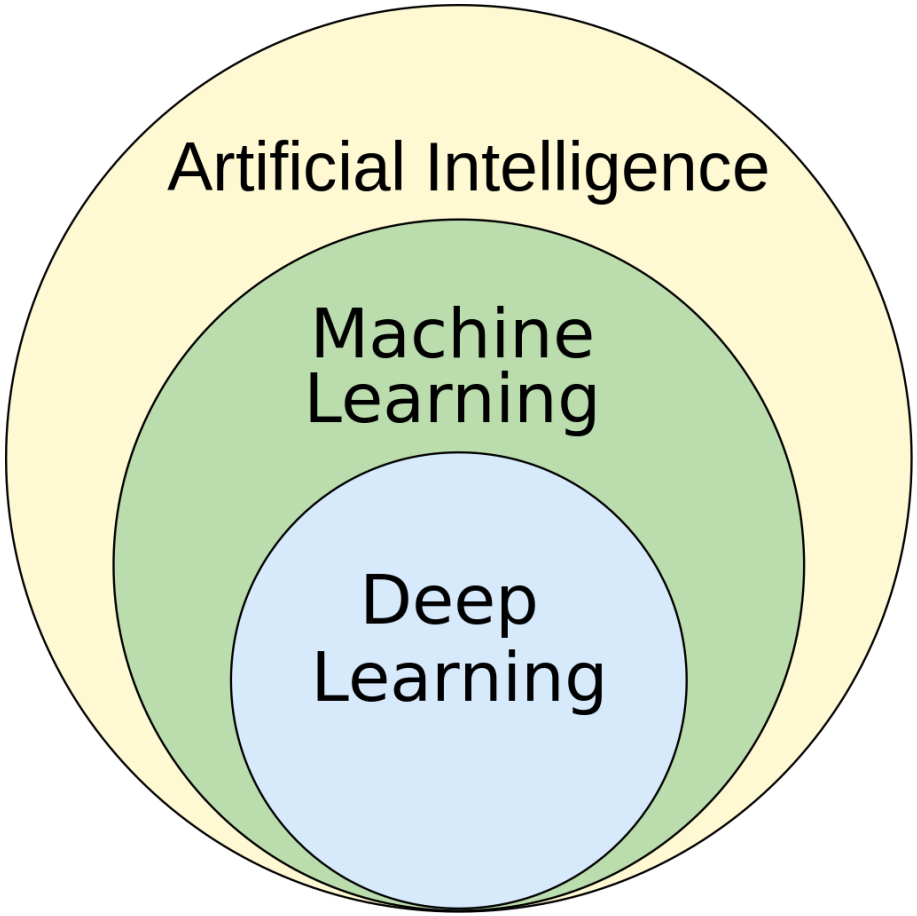
# Computational Imaging and Spectroscopy: Deep learning for imaging : introduction

Thierry SOREZE
DTU July 2024

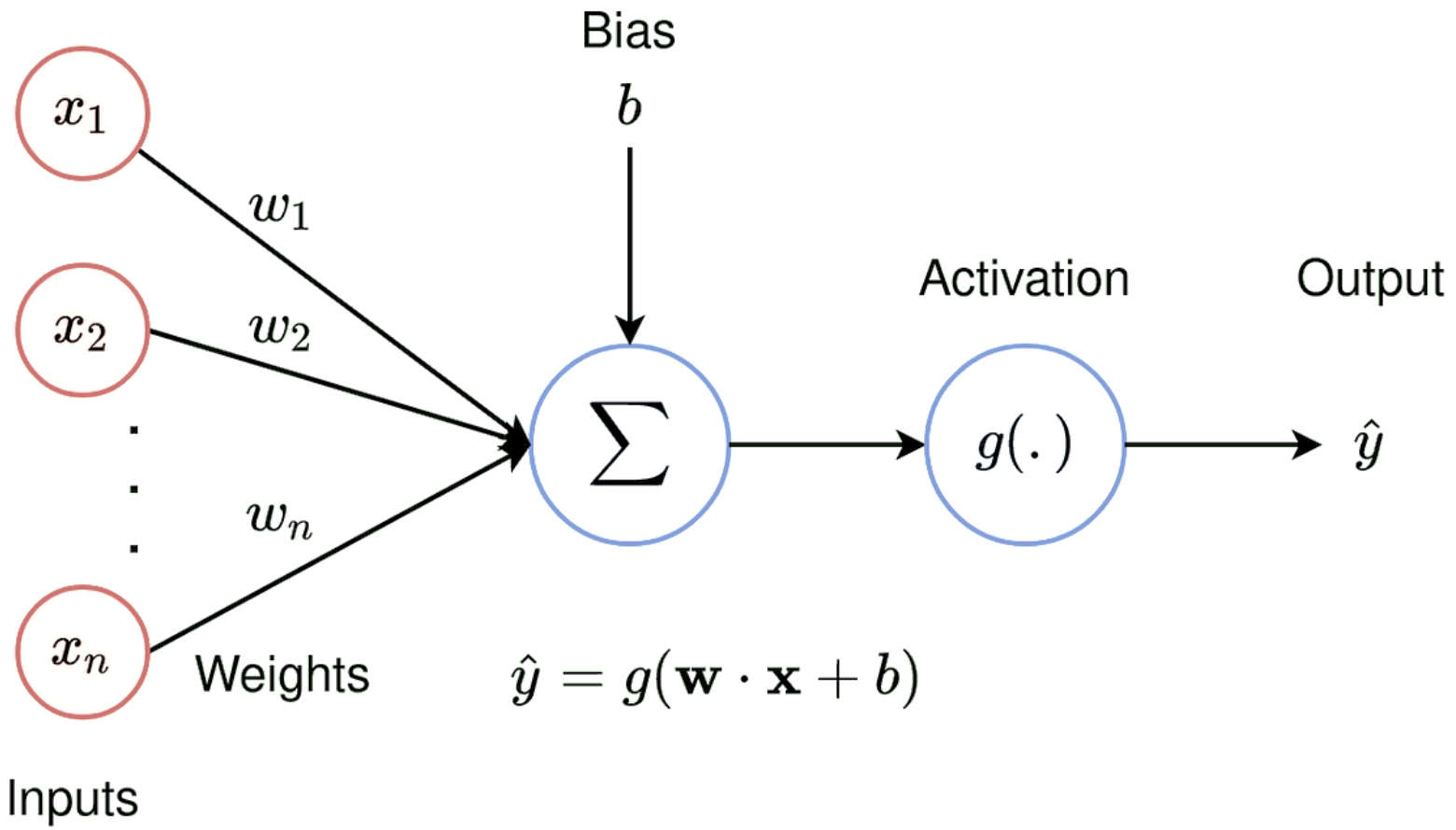



DTU Fotonik
Department of Photonics Engineering

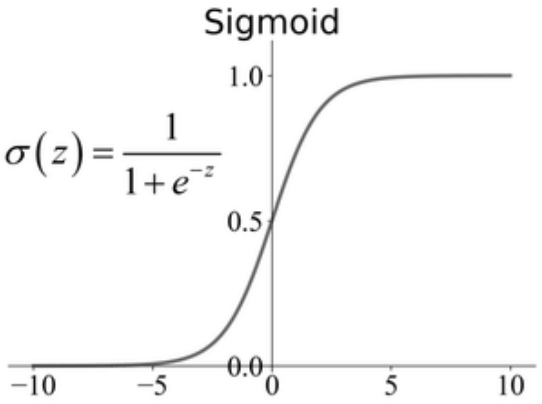
# Artificial intelligence



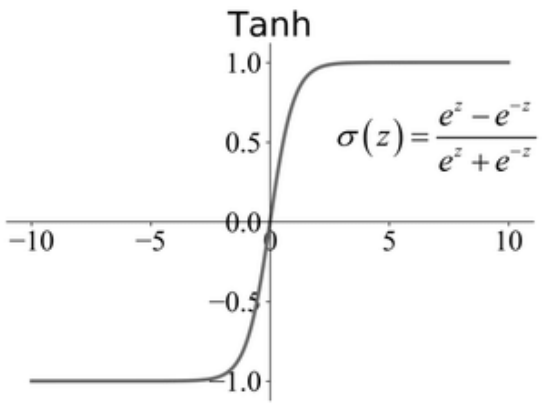
# Artificial Neural Network: The neuro model



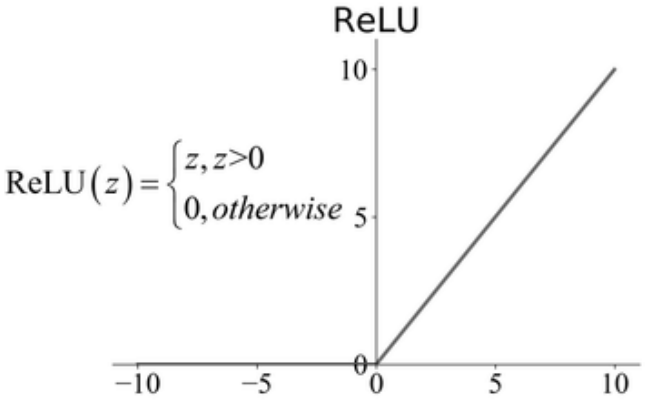
# Artificial Neural Network: Activation functions



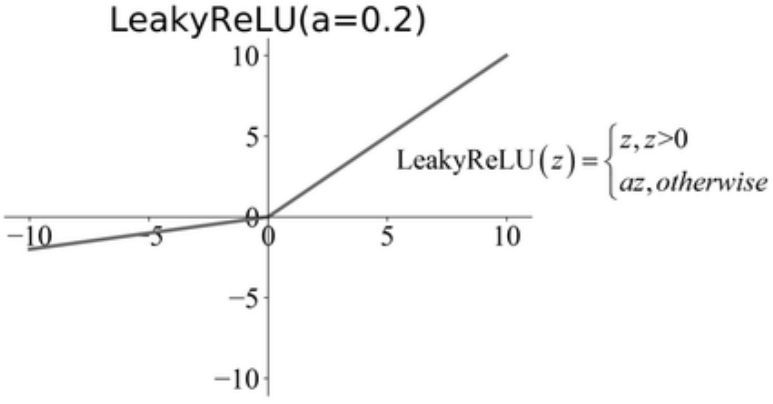
(a)



(b)

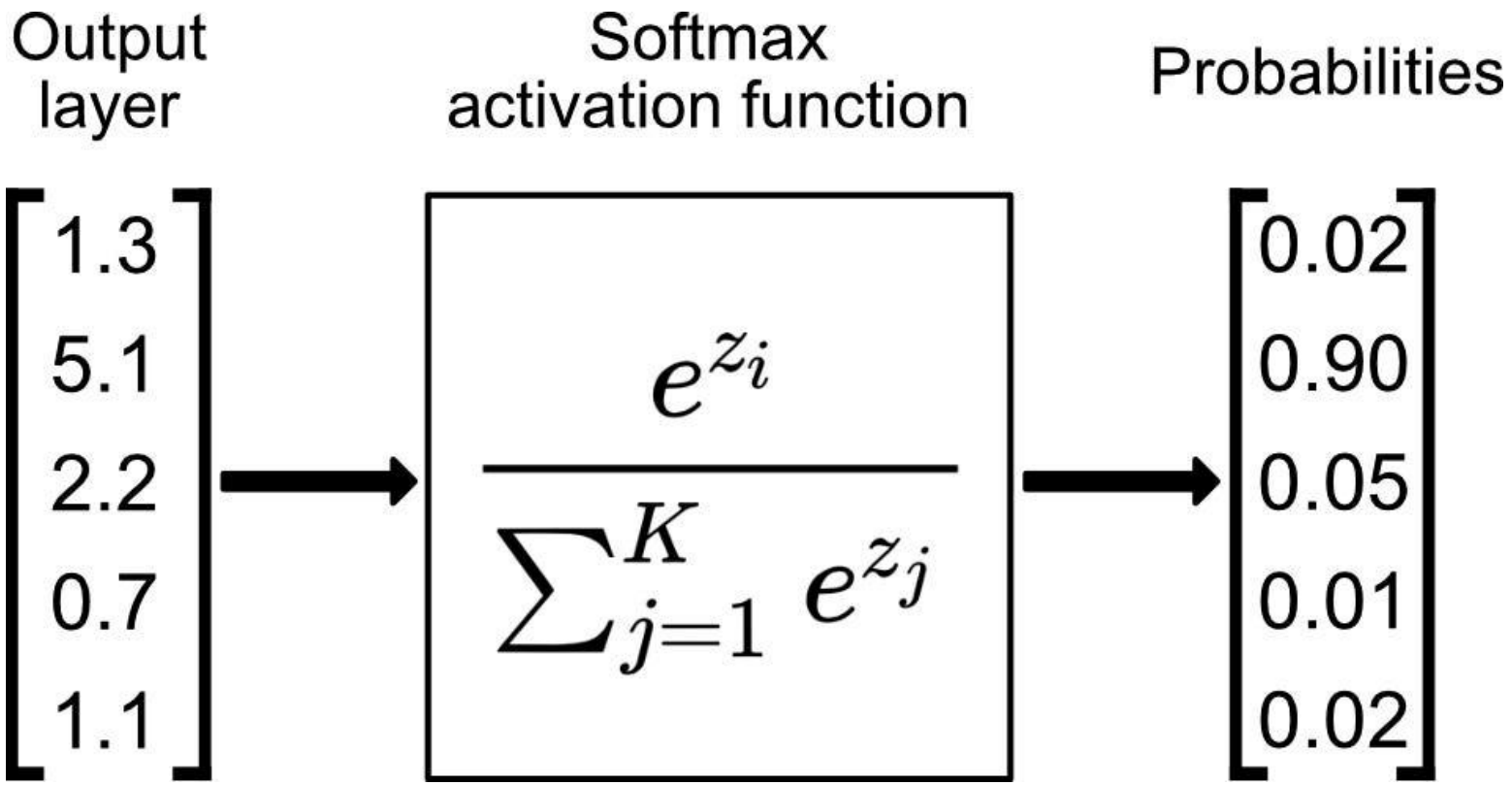


(c)

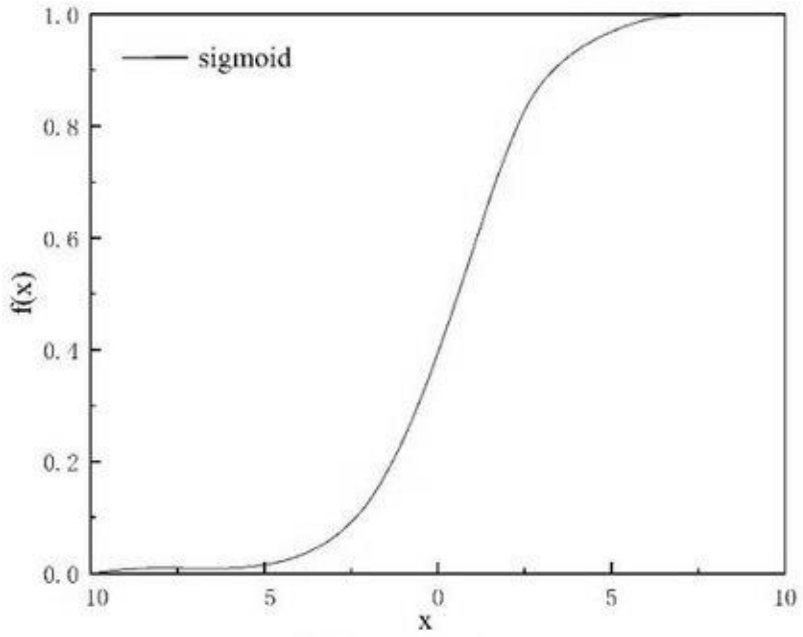


(d)

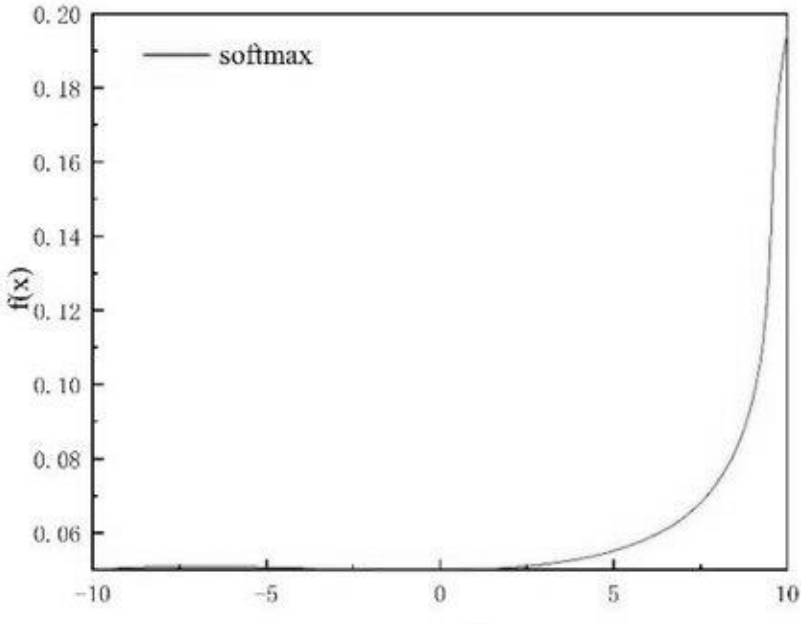
# Artificial Neural Network: Activation functions



# Artificial Neural Network: Activation functions

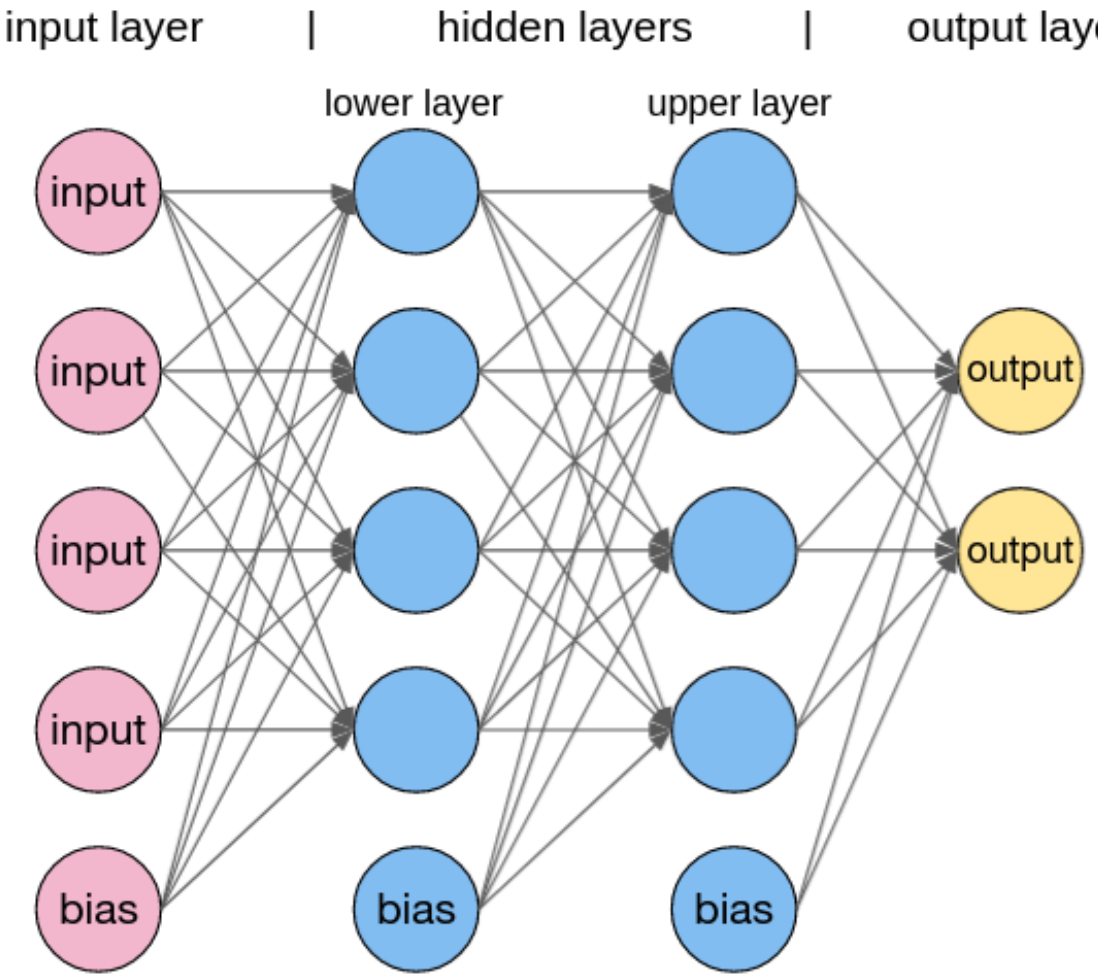


(a)Sigmoid function

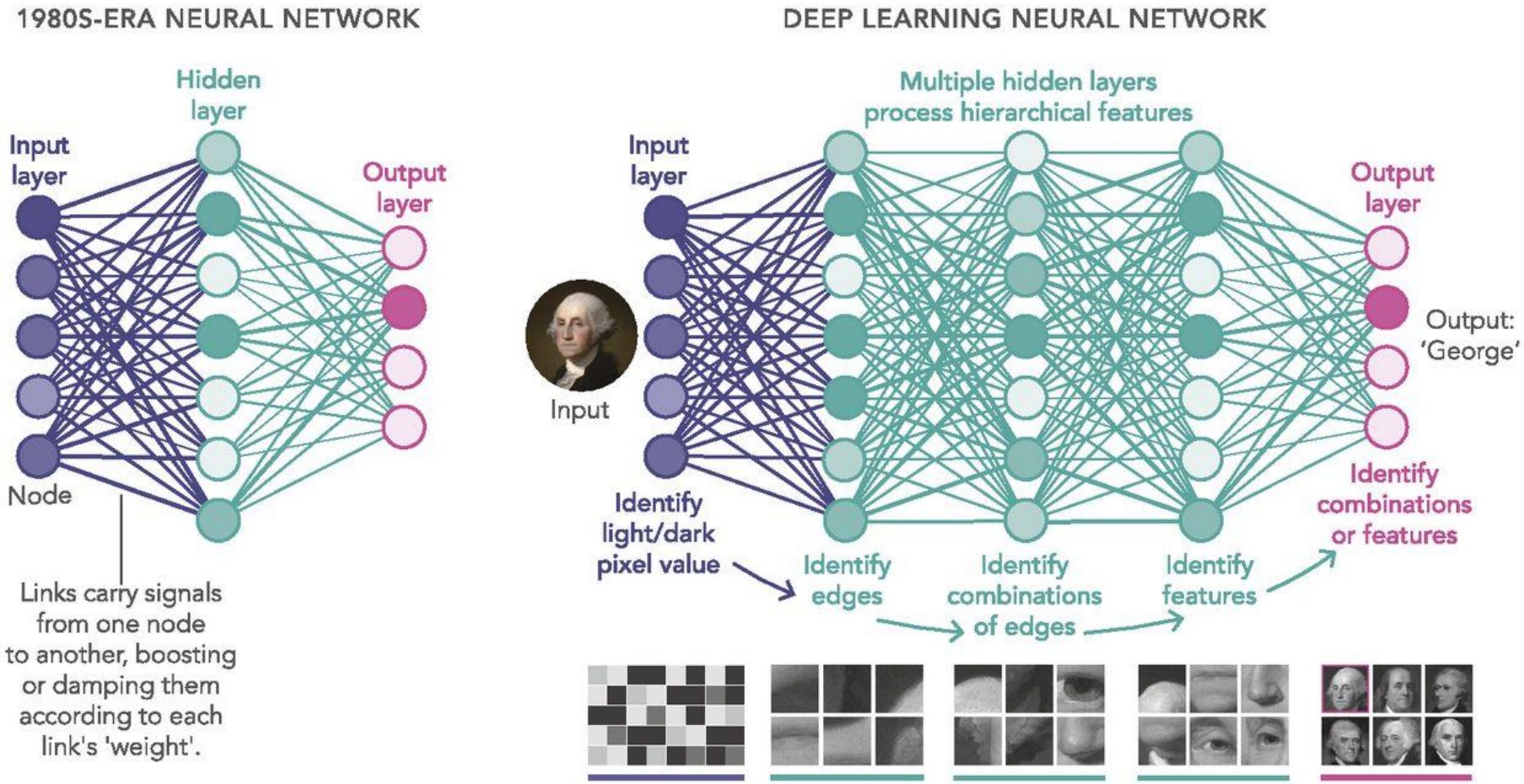


(b)Softmax function

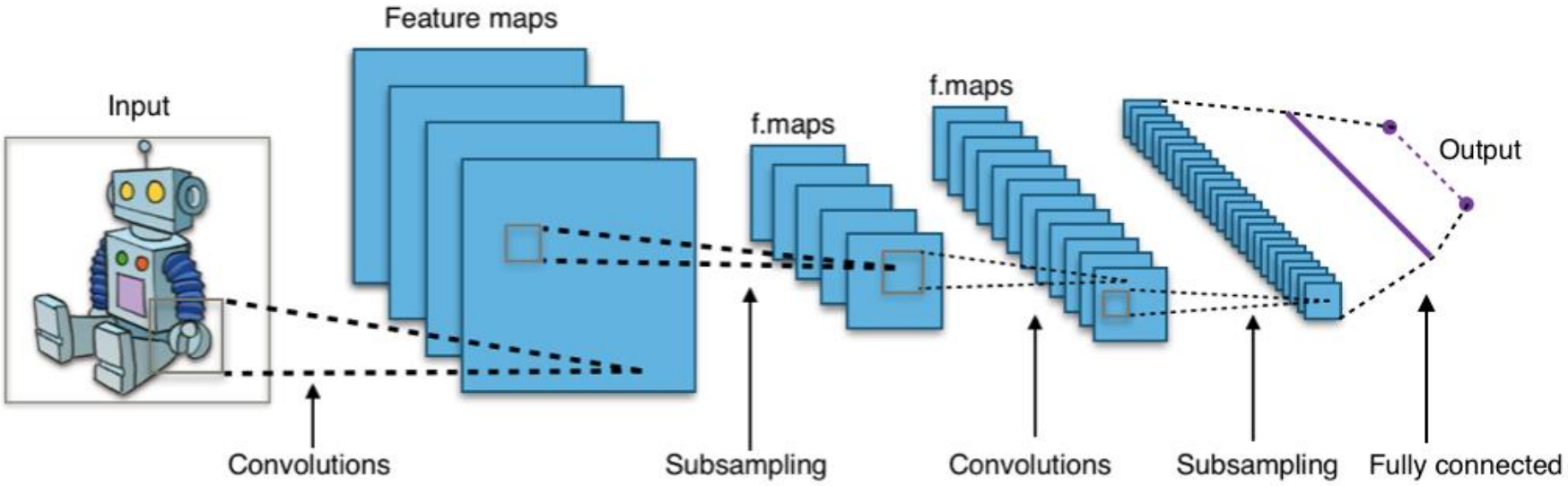
# Artificial Neural Network: Multilayer perceptron



# Artificial Neural Network: Deep Neural Networks



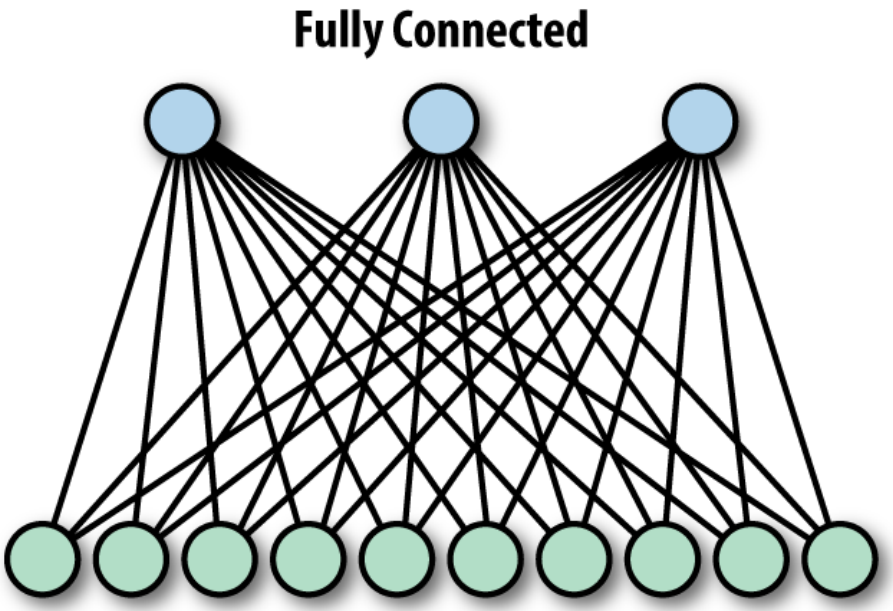
# Deep Neural Network: CNNs

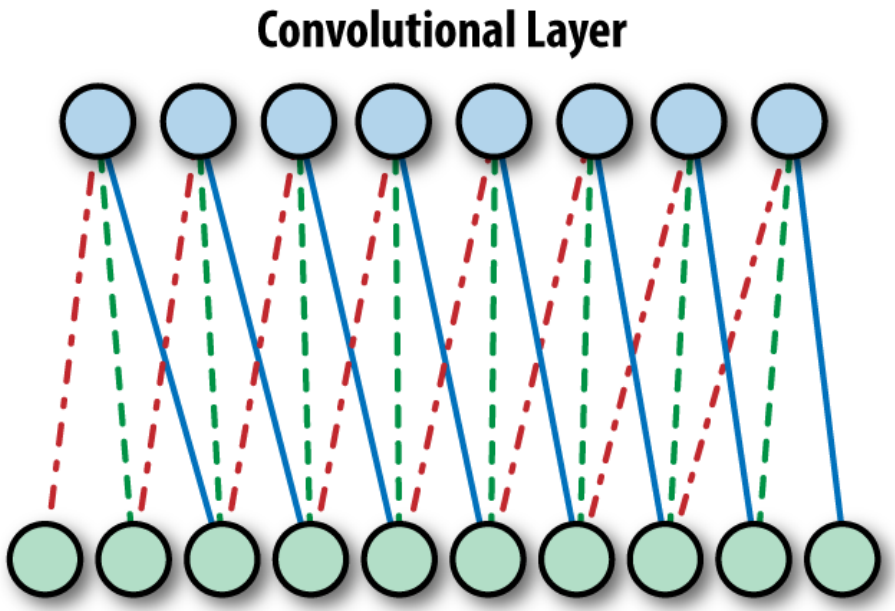


# Deep Neural Network: CNNs

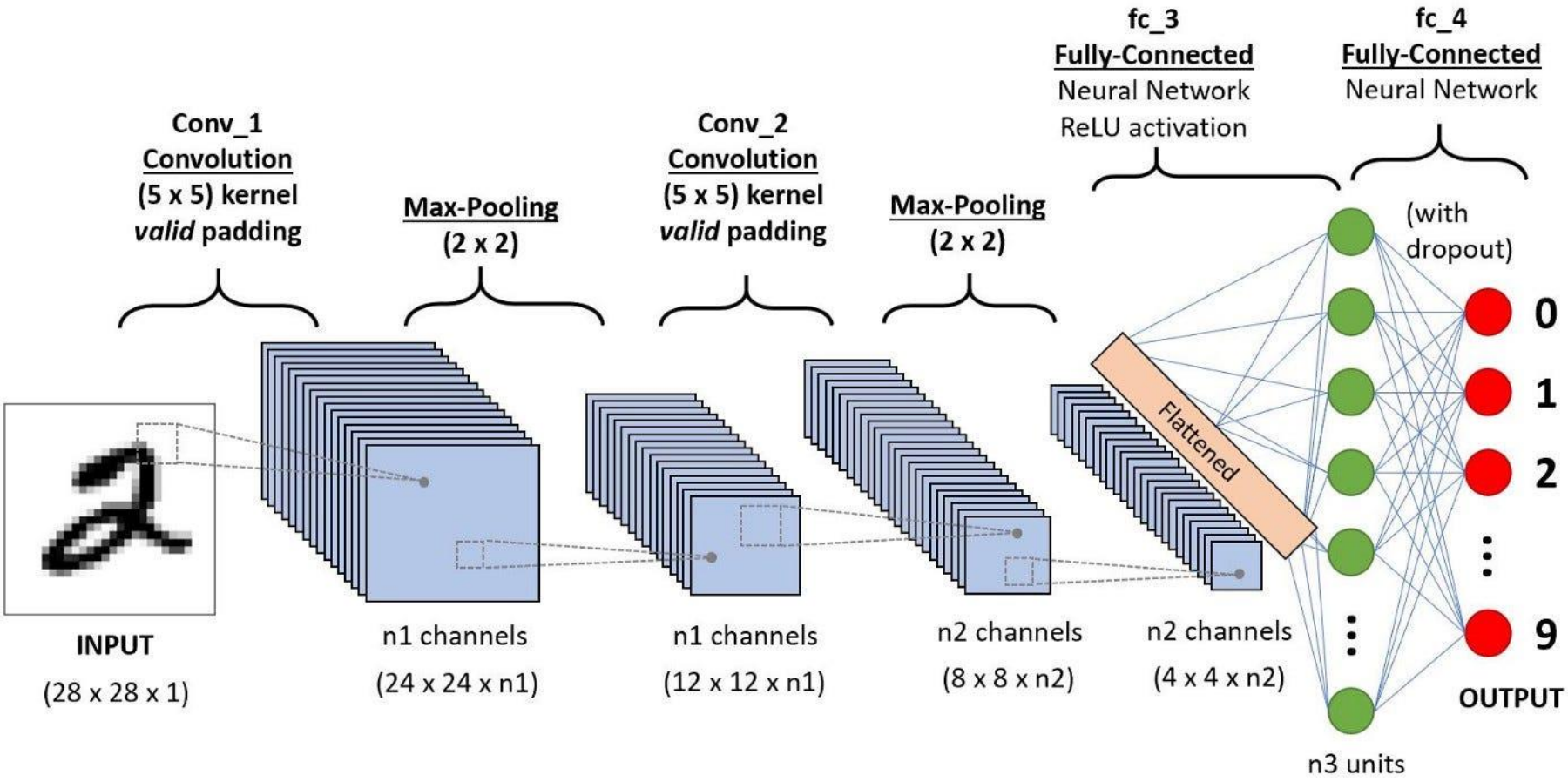
- Convolutional Layers
- Pooling layers
- Flattening layers
- Dense layers

# Deep Neural Network: CNNs

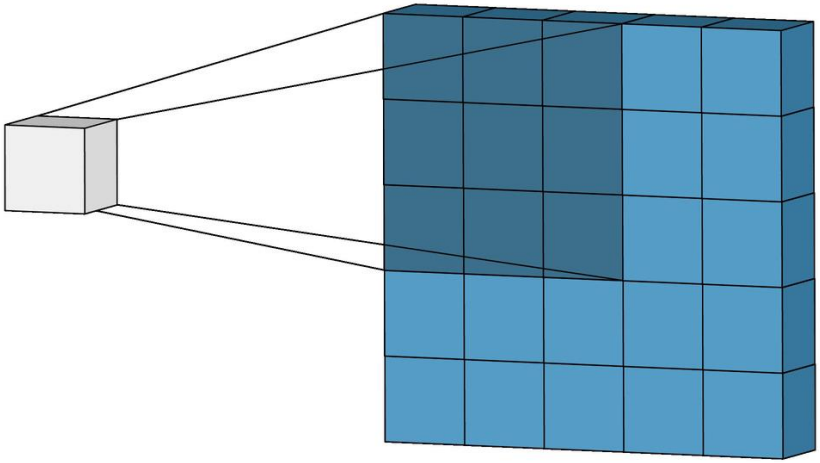


# Deep Neural Network: CNNs



# Deep Neural Network : Convolutions

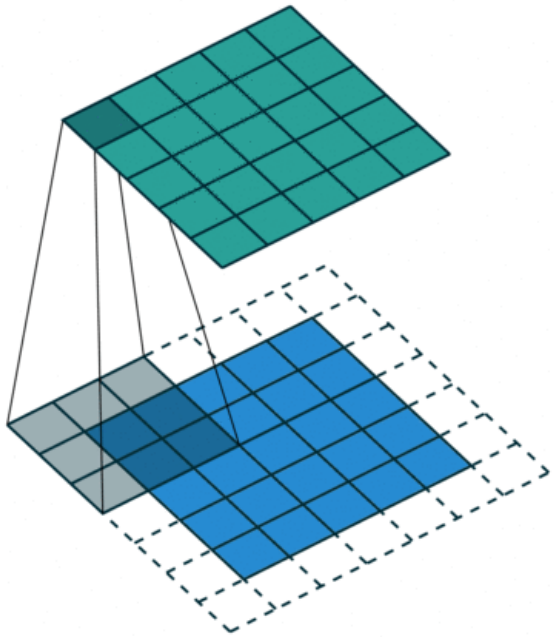


| | | | | |
|---|---|---|---|---|
| $3_0$ | $3_1$ | $2_2$ | 1 | 0 |
| $0_2$ | $0_2$ | $1_0$ | 3 | 1 |
| $3_0$ | $1_1$ | $2_2$ | 2 | 3 |
| 2 | 0 | 0 | 2 | 2 |
| 2 | 0 | 0 | 0 | 1 |

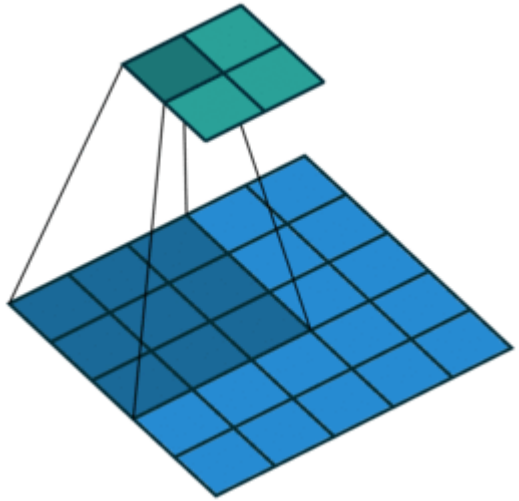
| | | |
|---|---|---|
| 12.0 | 12.0 | 17.0 |
| 10.0 | 17.0 | 19.0 |
| 9.0 | 6.0 | 14.0 |

# Deep Neural Network : Stride and padding

Padding

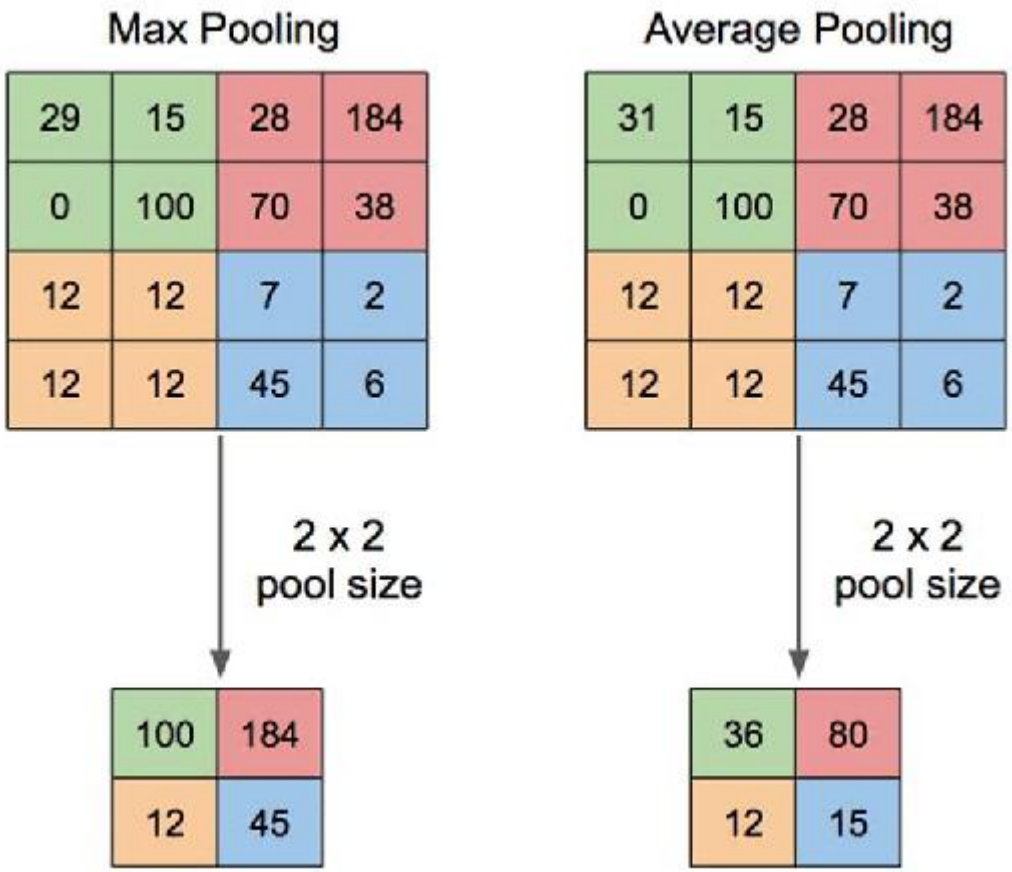


Stride

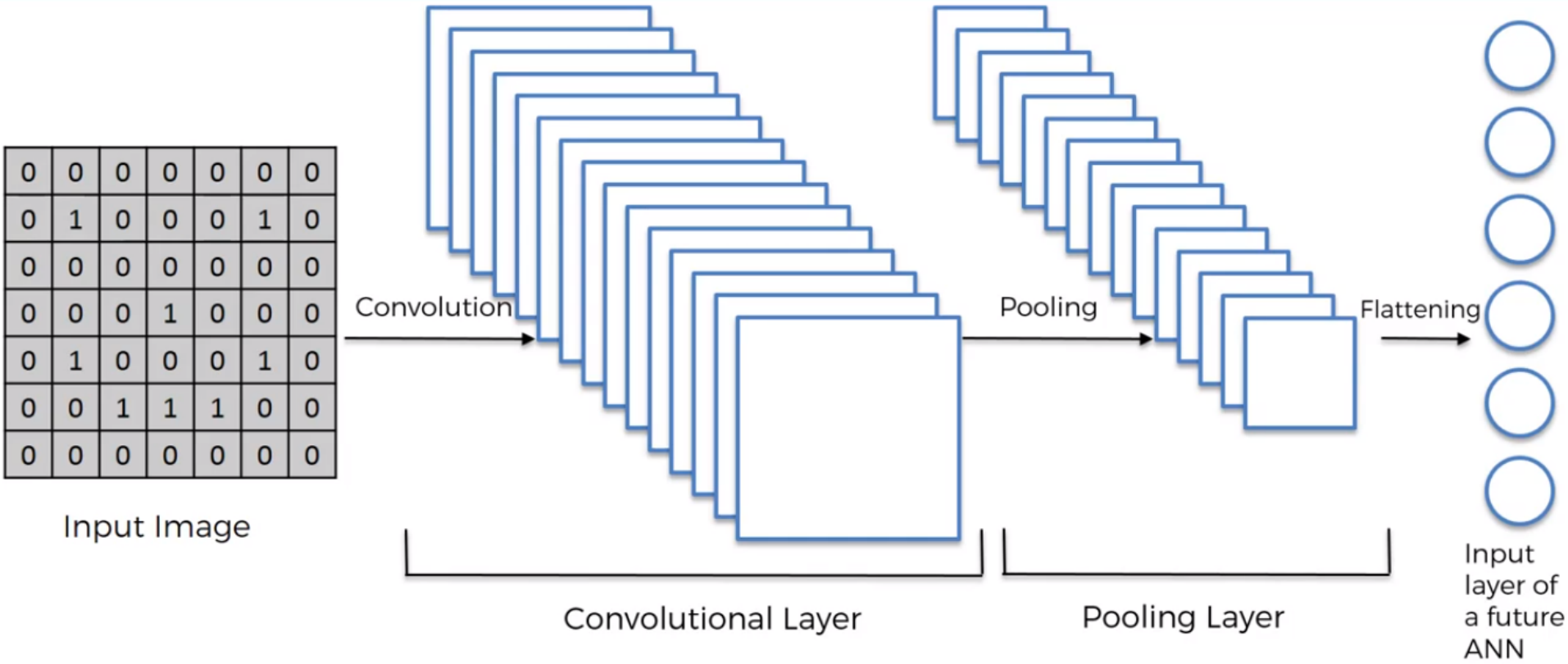


$$\text{Output size} = \frac{\text{Input size} + 2 \times \text{Padding} - \text{Kernel size}}{\text{Stride}} + 1$$

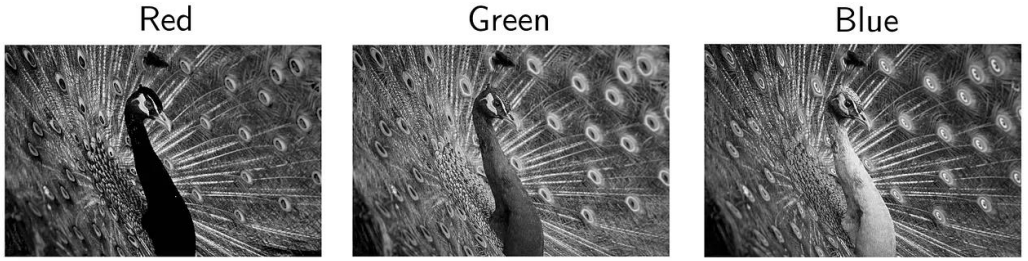
# Deep Neural Network : Pooling

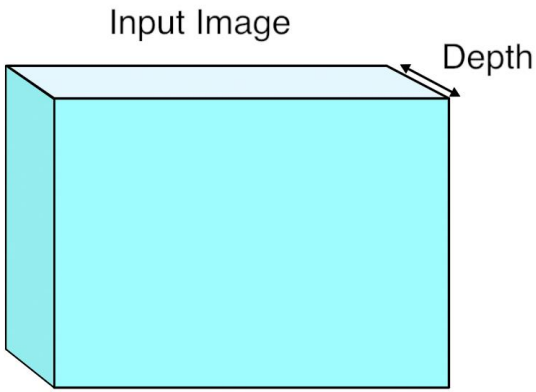


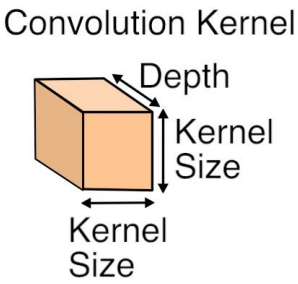
# Deep Neural Network : flattening

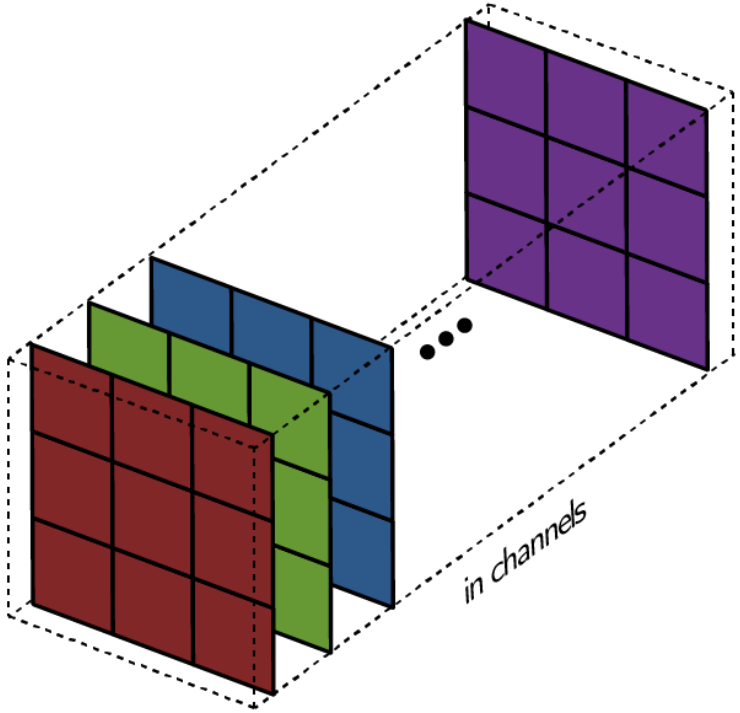


# Deep Neural Network : RGB images and CNNs

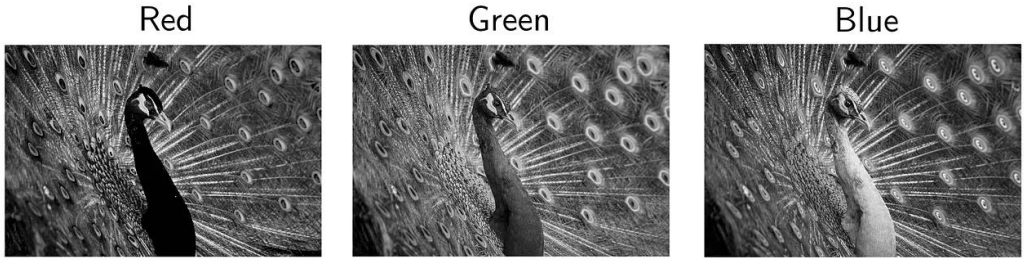


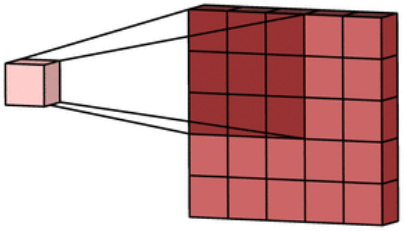



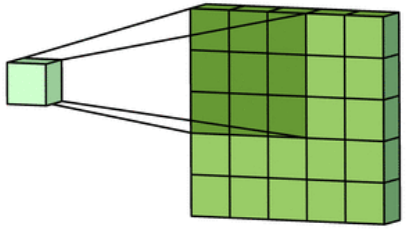



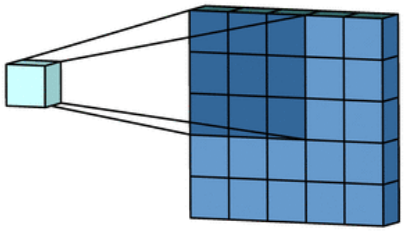



# Deep Neural Network : RGB images and CNNs

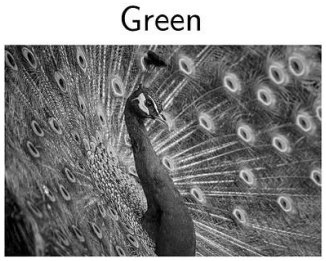


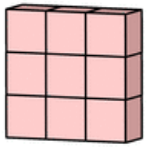


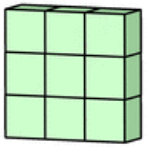


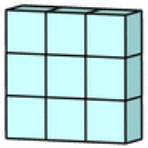
# Deep Neural Network : RGB images and CNNs

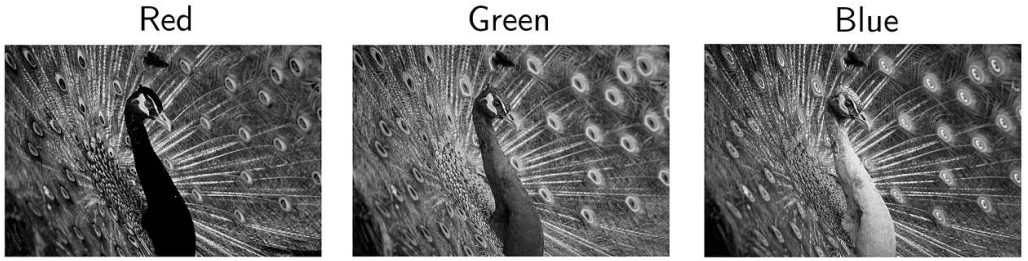


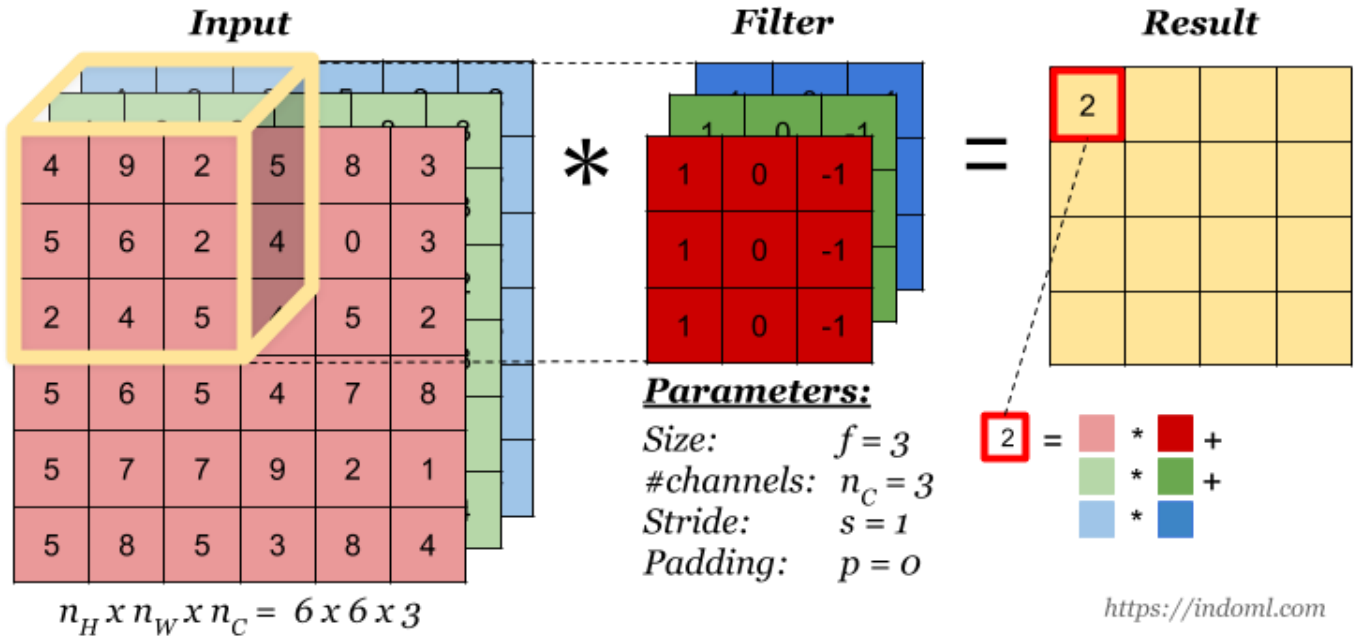




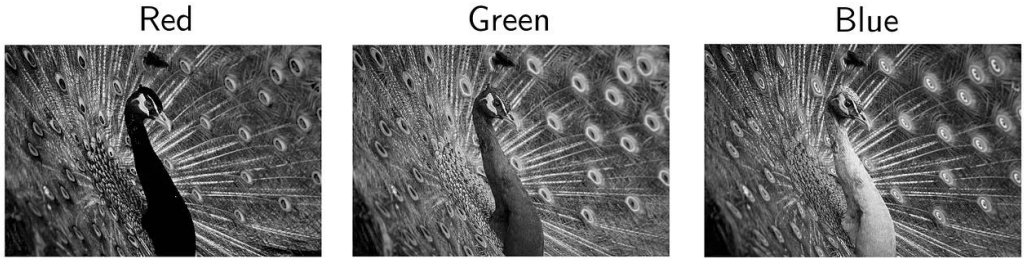


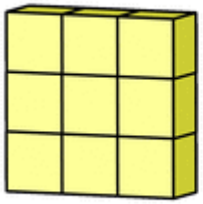
# Deep Neural Network : RGB images and CNNs



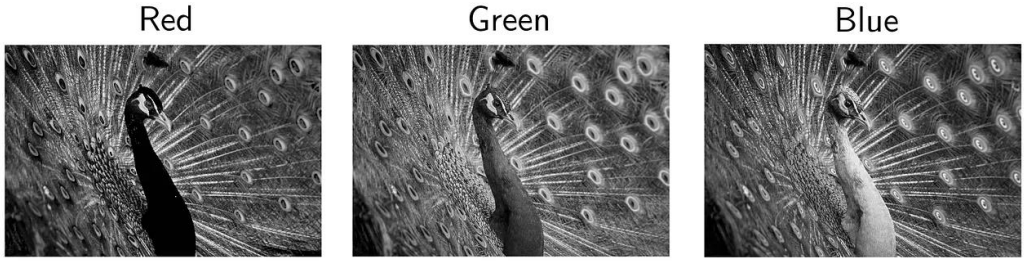



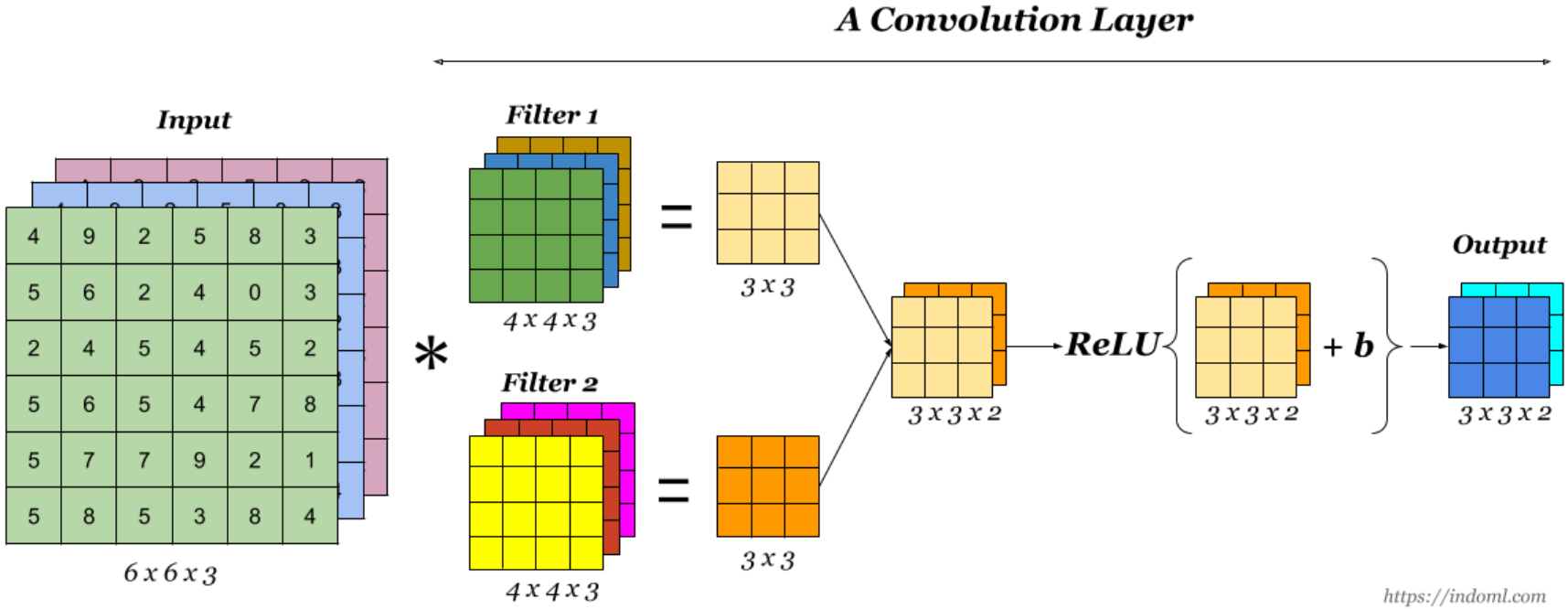
# Deep Neural Network : RGB images and CNNs



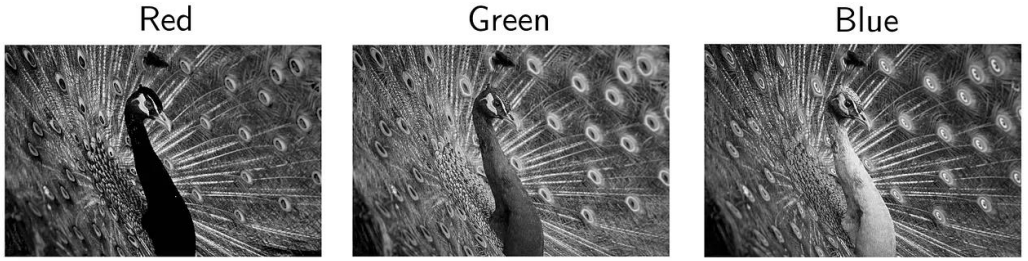



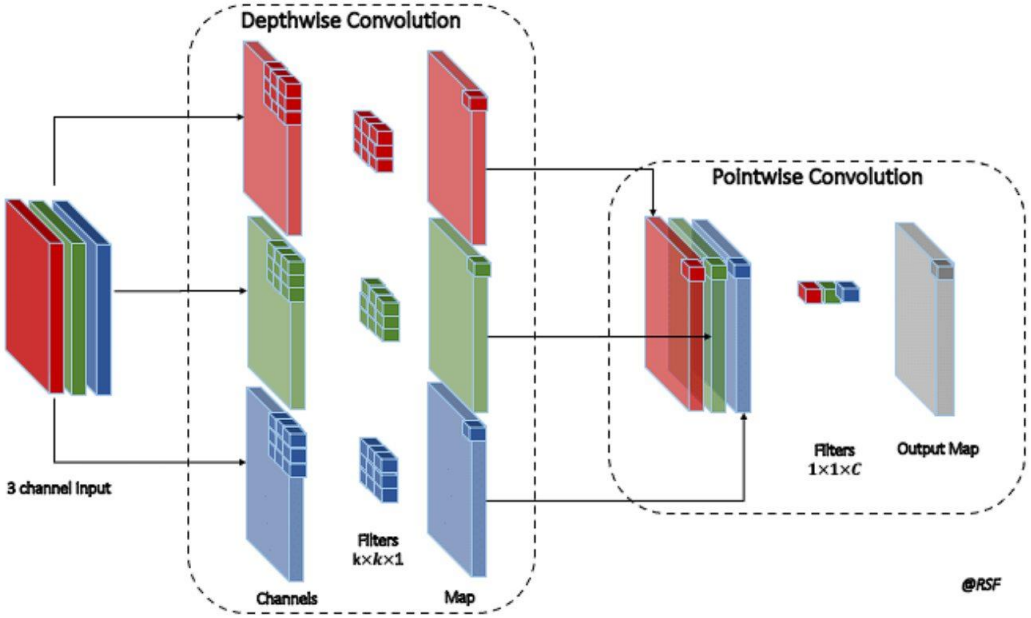
# Deep Neural Network : RGB images and CNNs



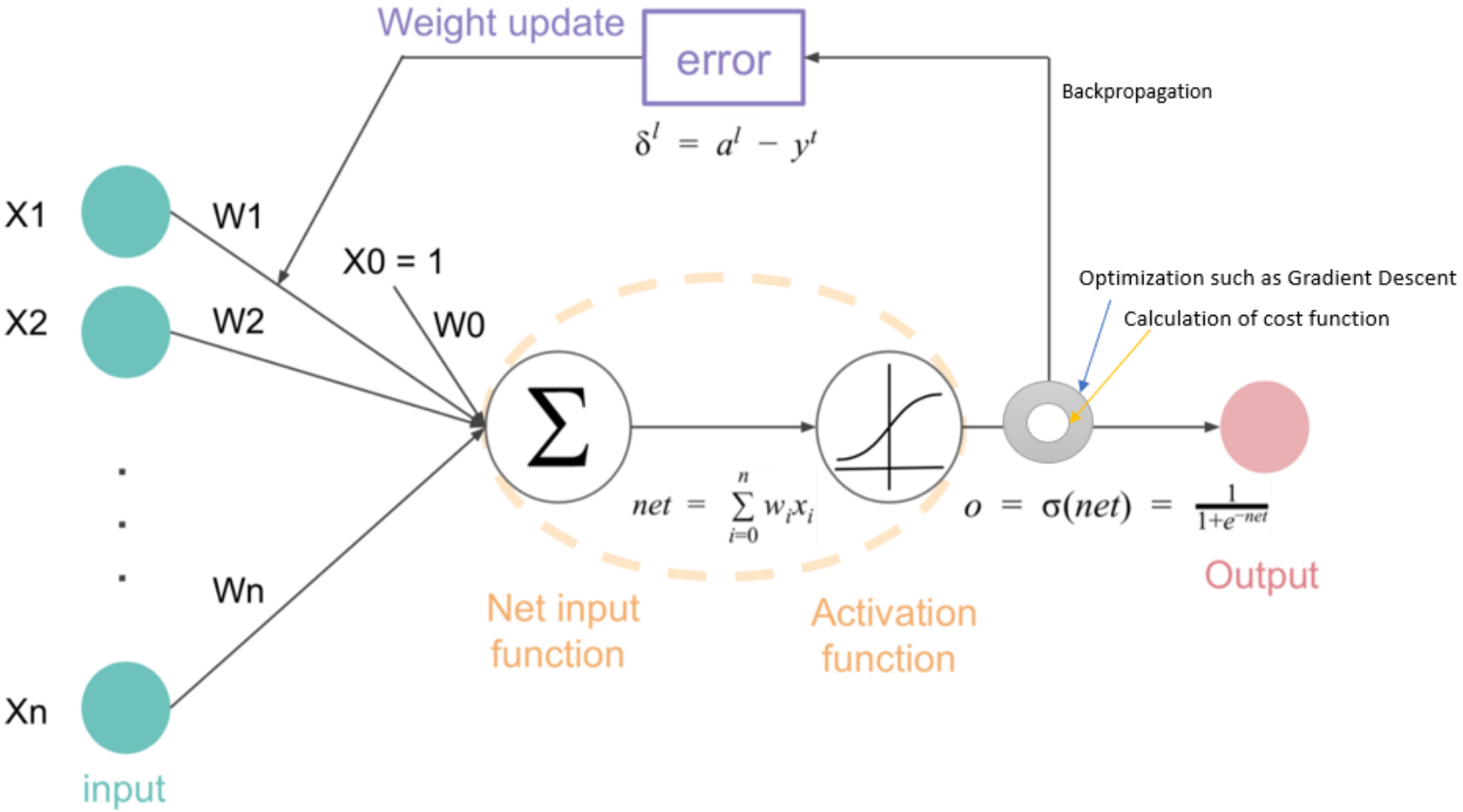



# Deep Neural Network : Dwise and Pwise Conv

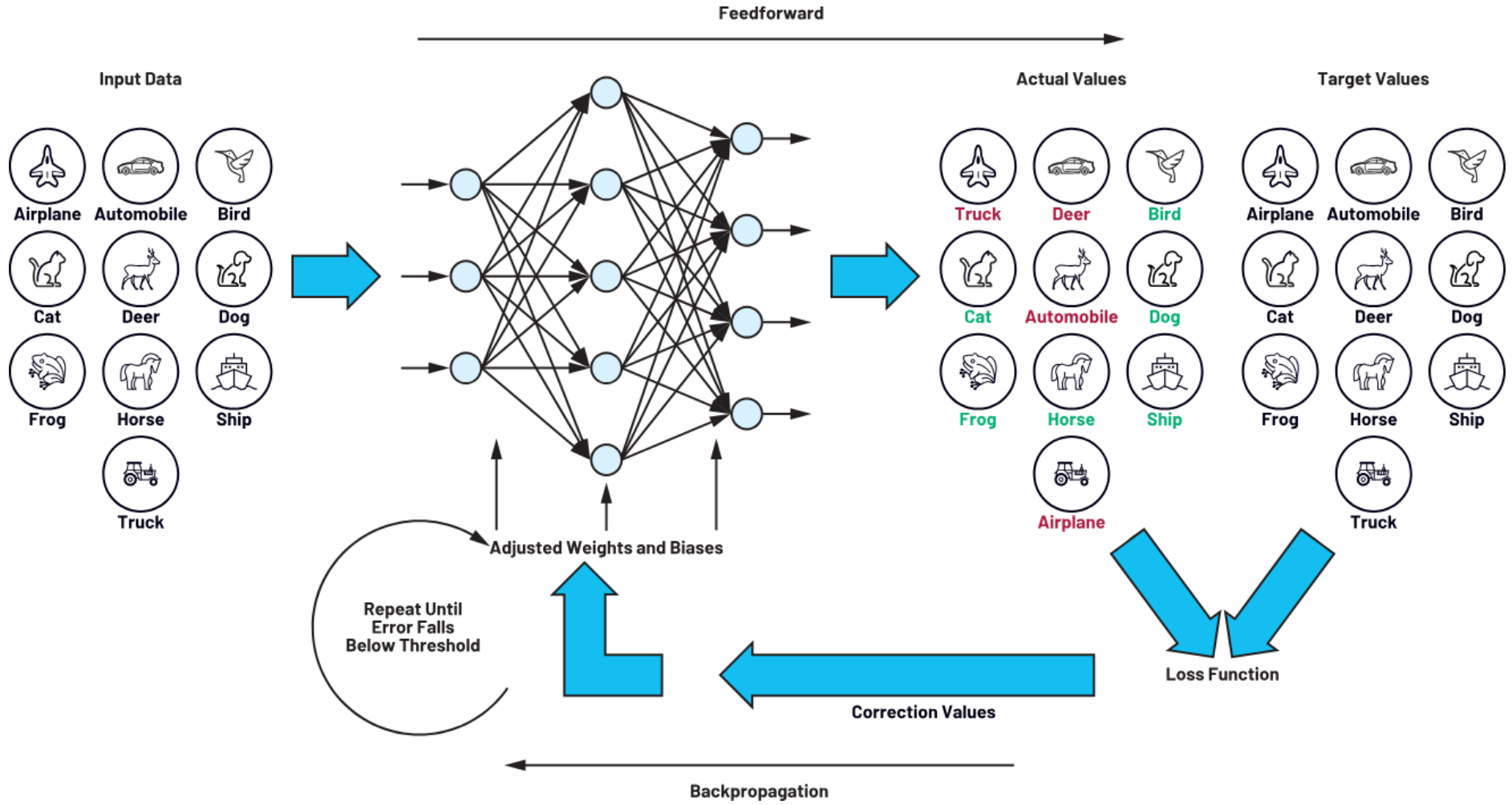


# Training: Backpropagation



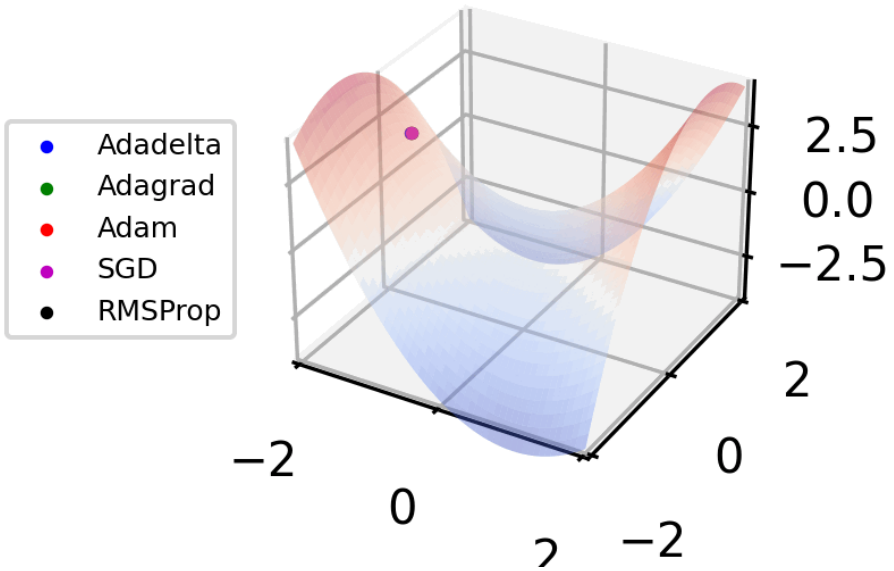
# Training: Backpropagation



# Training: Loss functions

| Task | Error type | Loss function | Note |
|---|---|---|---|
| Regression | Mean-squared error | $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$ | Easy to learn but sensitive to outliers (MSE, L2 loss) |
| | Mean absolute error | $\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$ | Robust to outliers but not differentiable (MAE, L1 loss) |
| Classification | Cross entropy = Log loss | $-\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] =$ | Quantify the difference between two probability |

# Training: ADAM Optimizer




$$m_w^{(t+1)} \leftarrow \beta_1 m_w^{(t)} + (1-\beta_1)\,\nabla_w L^{(t)}$$

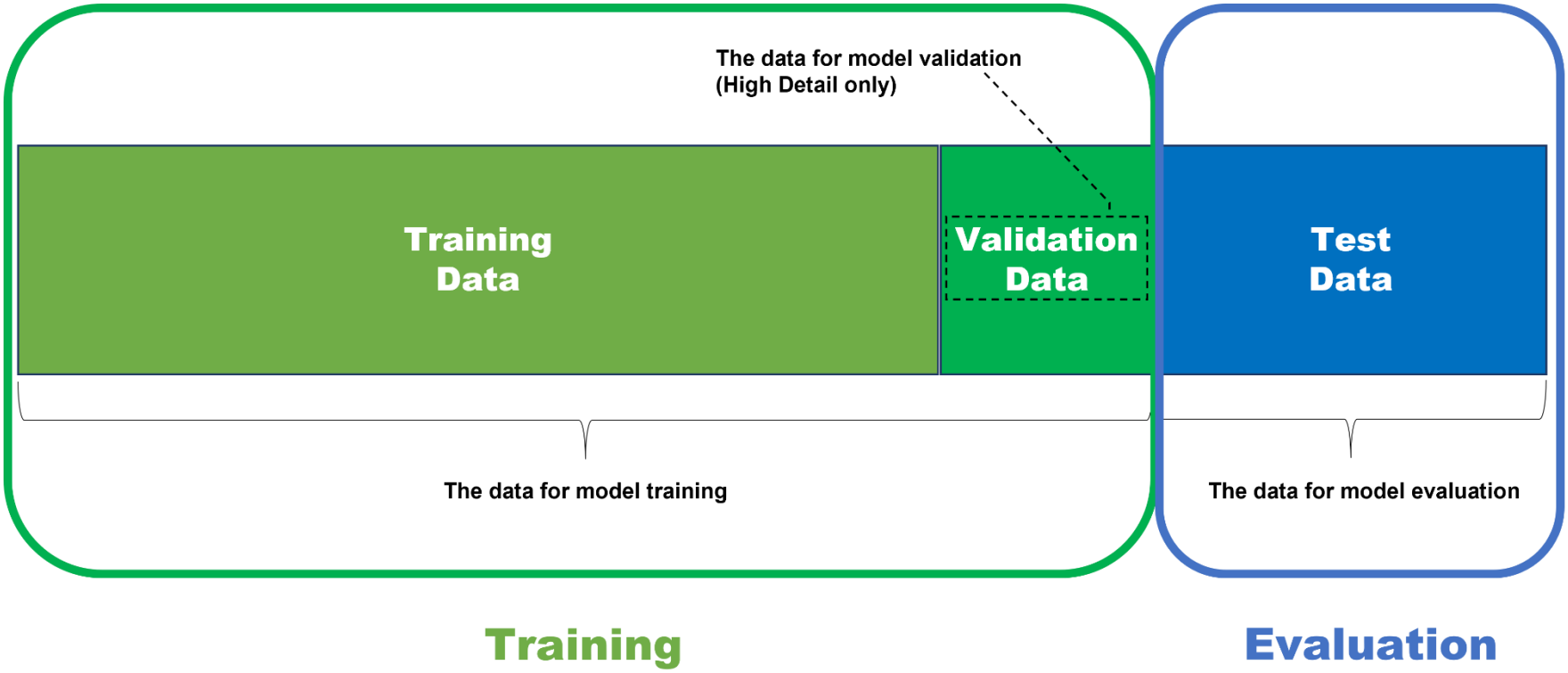
$$v_w^{(t+1)} \leftarrow \beta_2 v_w^{(t)} + (1-\beta_2)\left(\nabla_w L^{(t)}\right)^2$$

$$\hat{m}_w = \frac{m_w^{(t+1)}}{1-\beta_1^t}$$

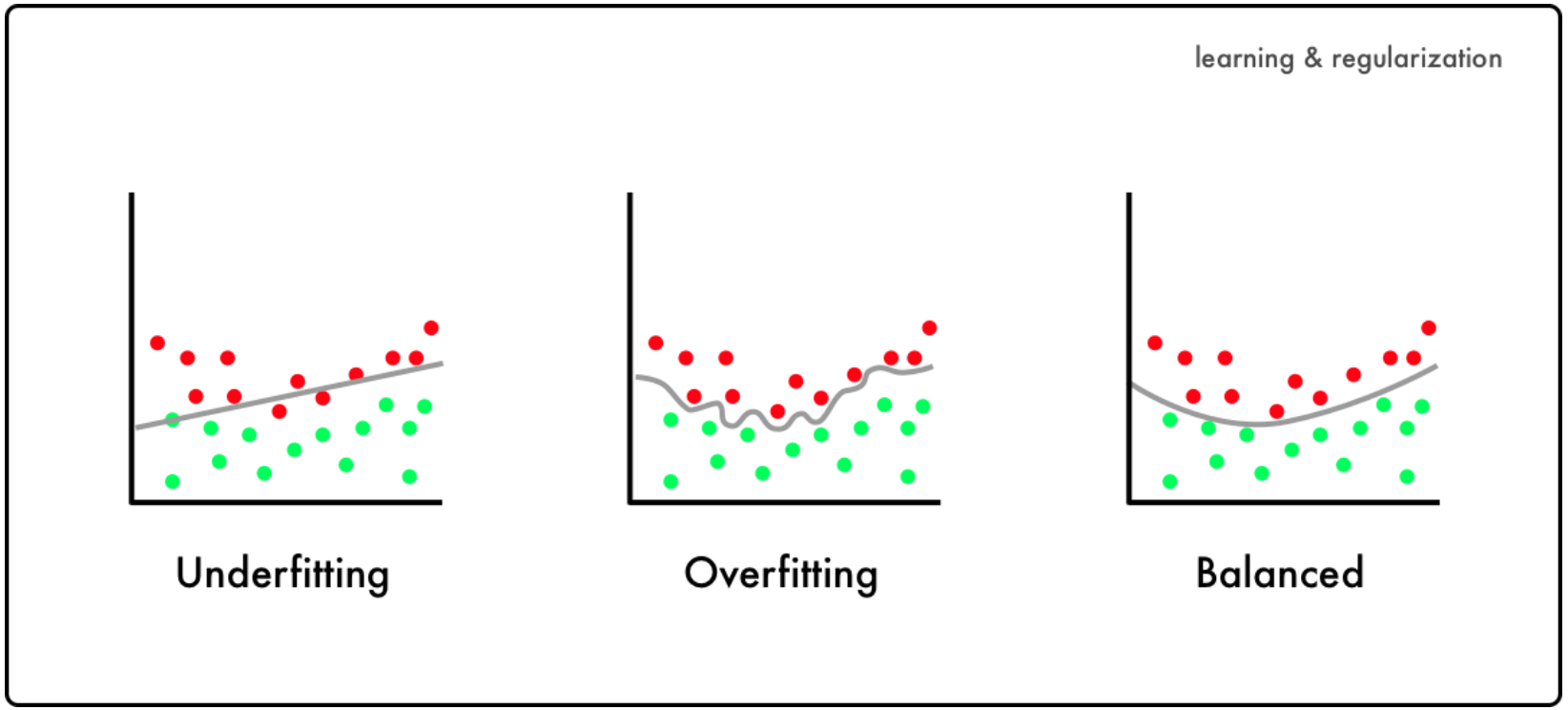
$$\hat{v}_w = \frac{v_w^{(t+1)}}{1-\beta_2^t}$$

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \frac{\hat{m}_w}{\sqrt{\hat{v}_w} + \epsilon}$$

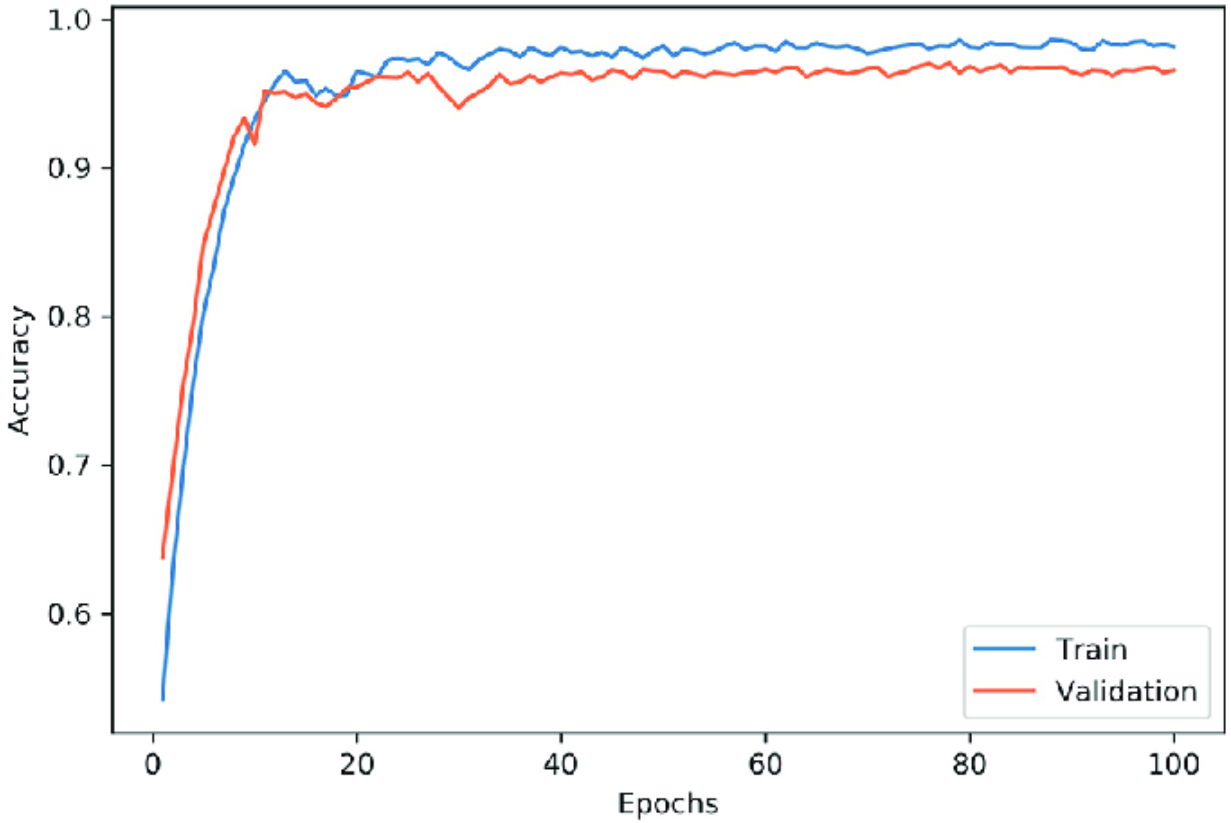
# Training: data split



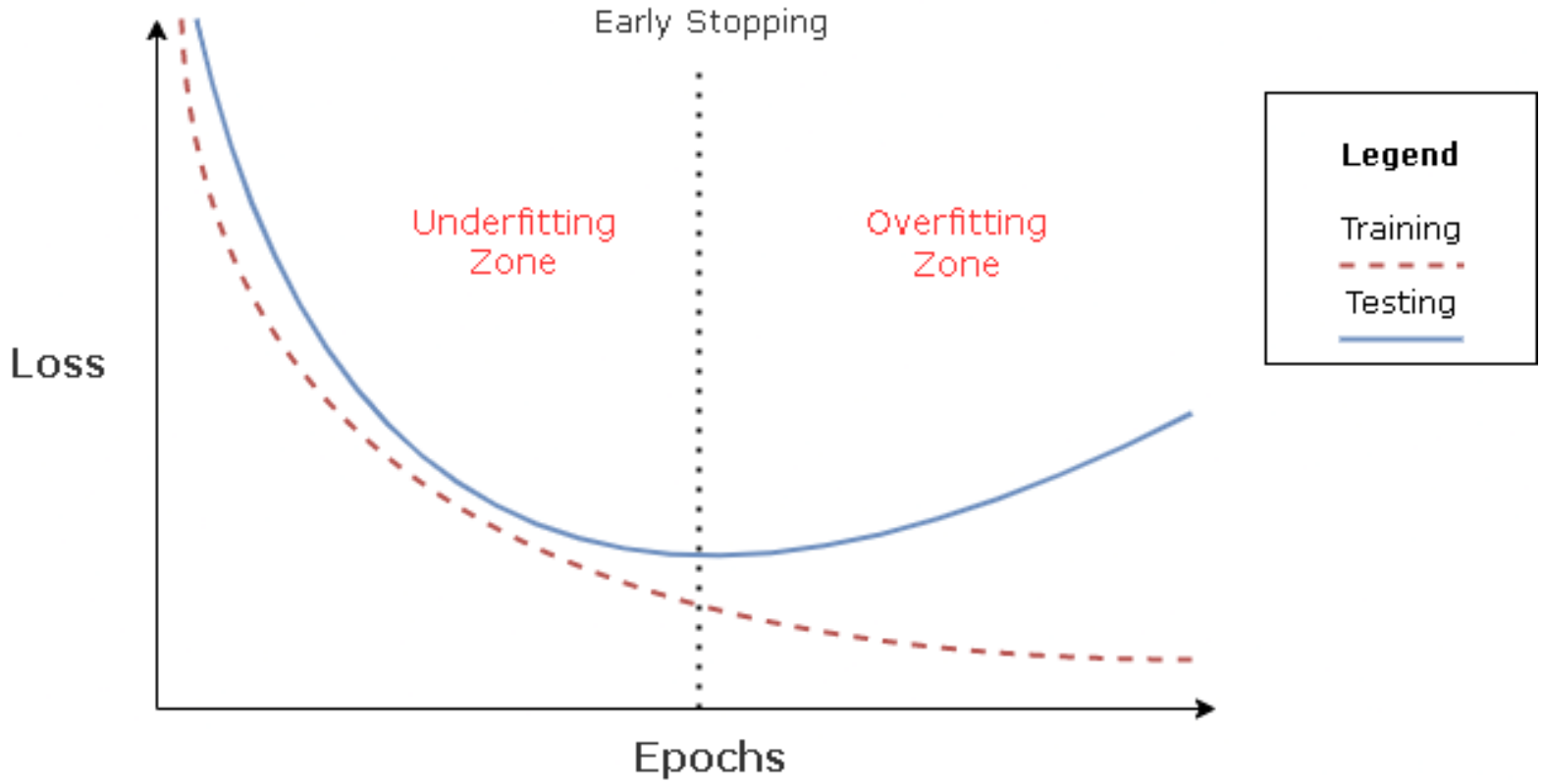
# Training: Overfitting



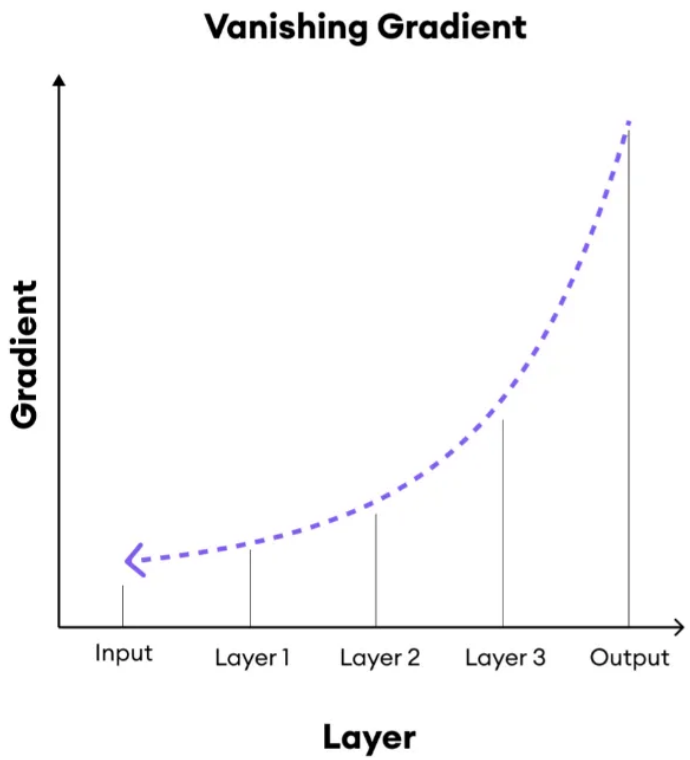
# Training: Overfitting

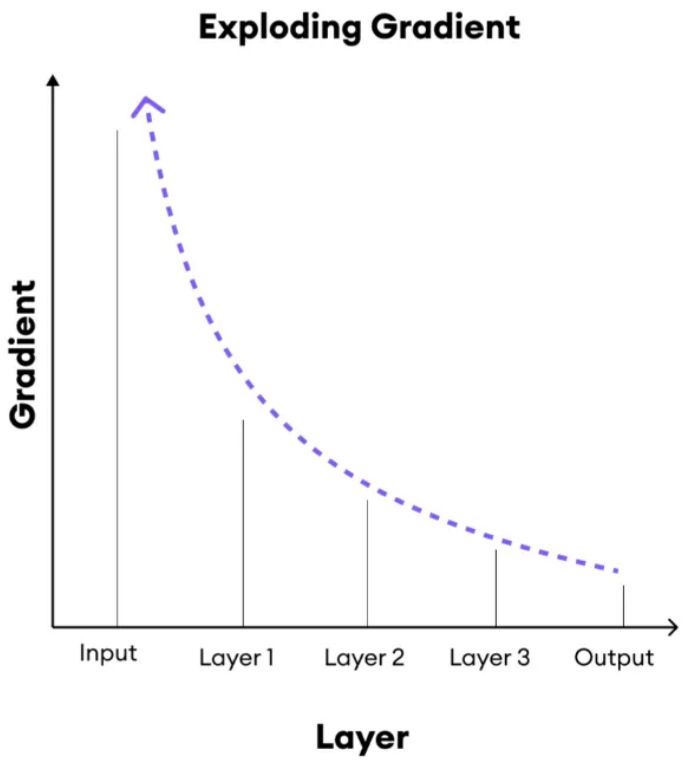


# Training: Overfitting

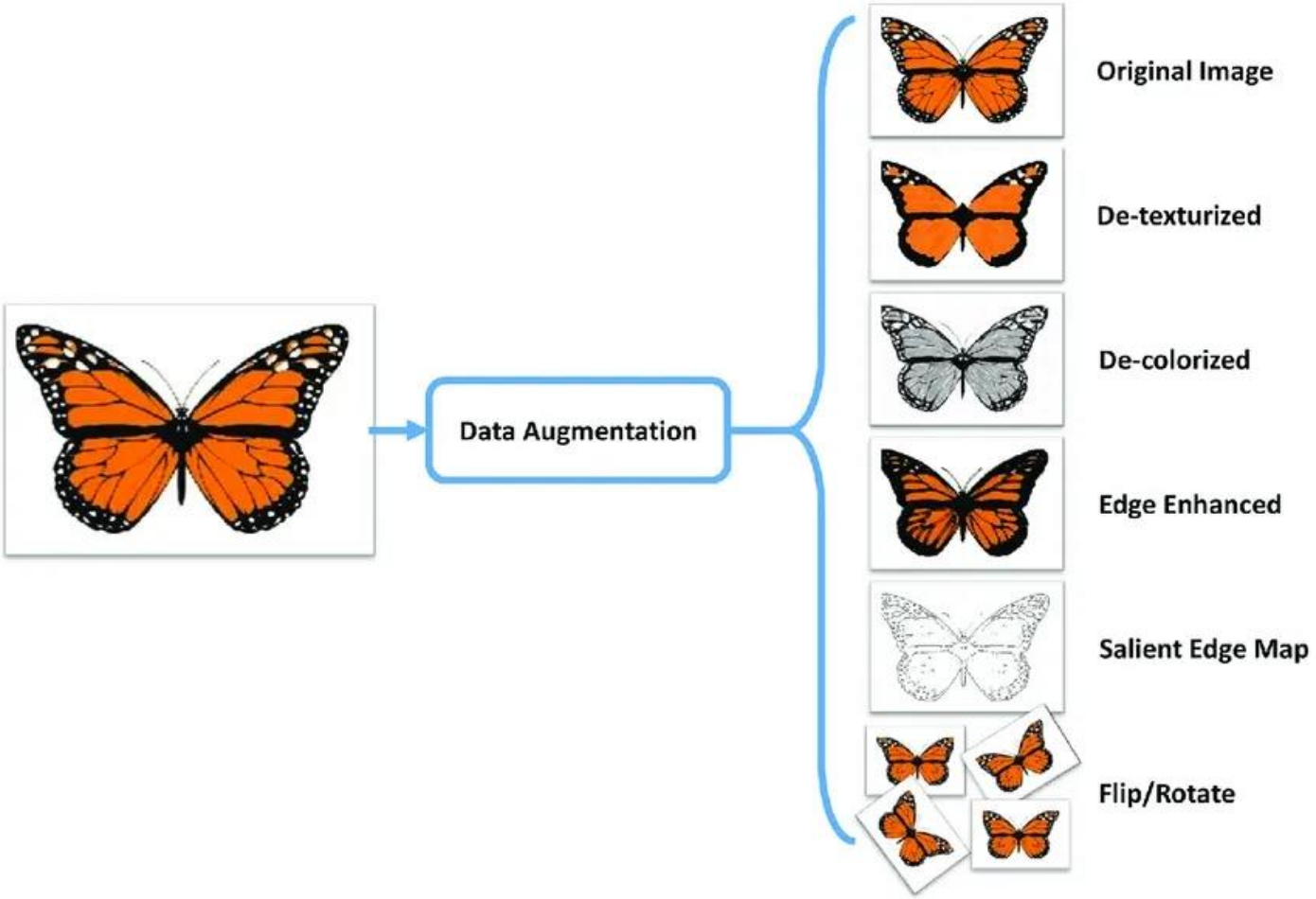


# Training: Vanishing and Exploding gradients

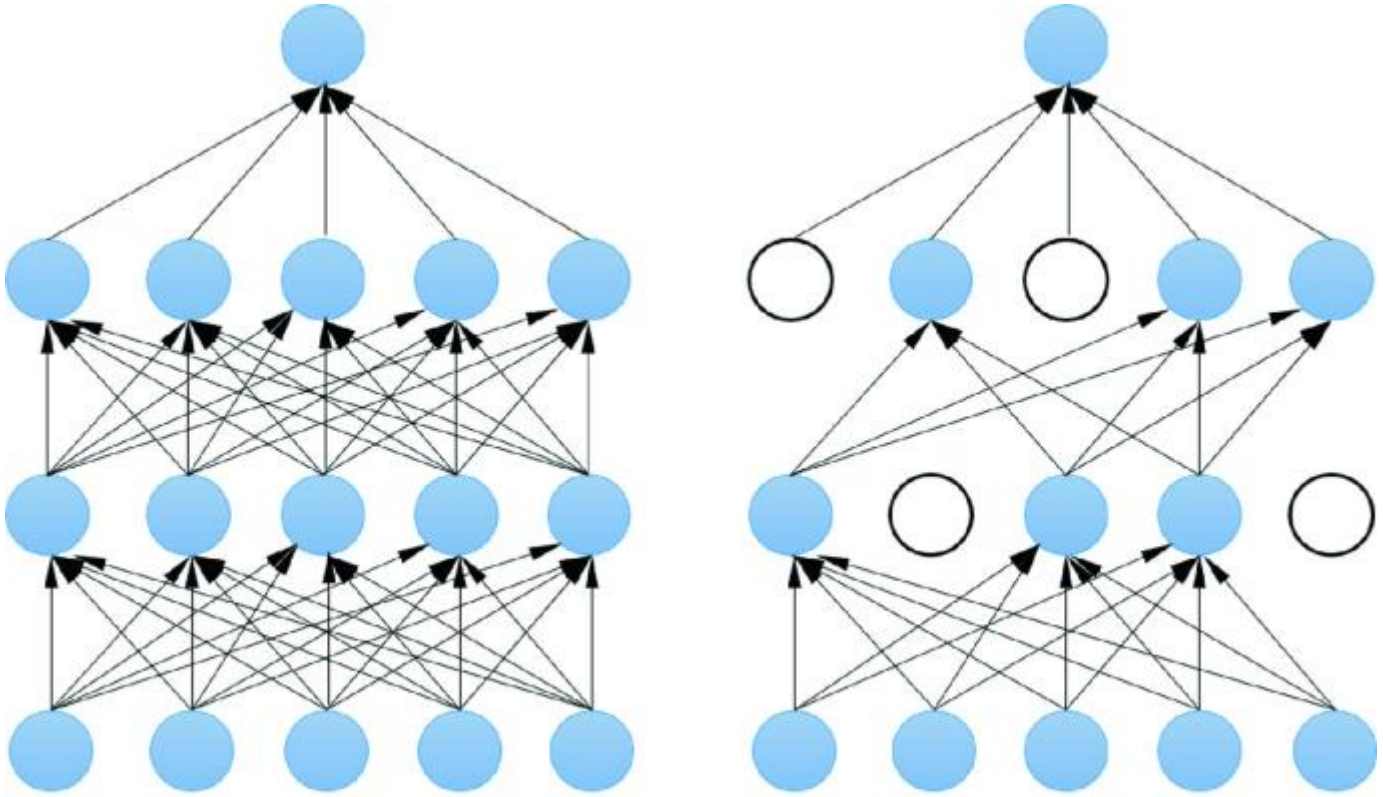


# Training: Data augmentation



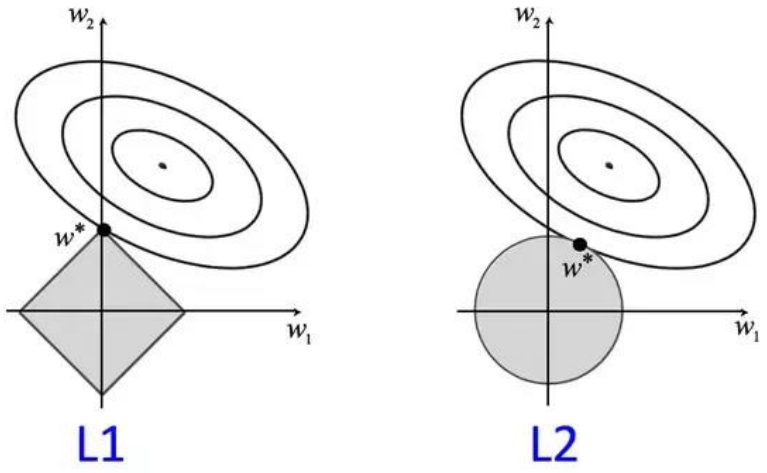
# Training: Dropout

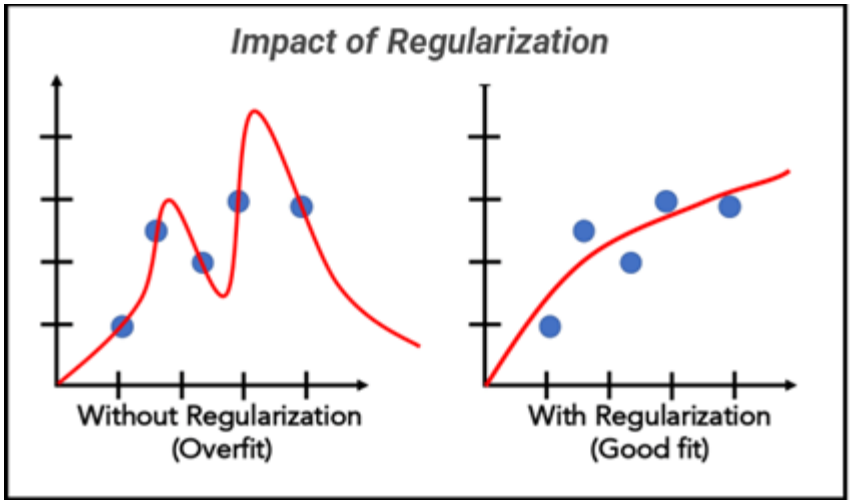


(a) A standard deep neural network.

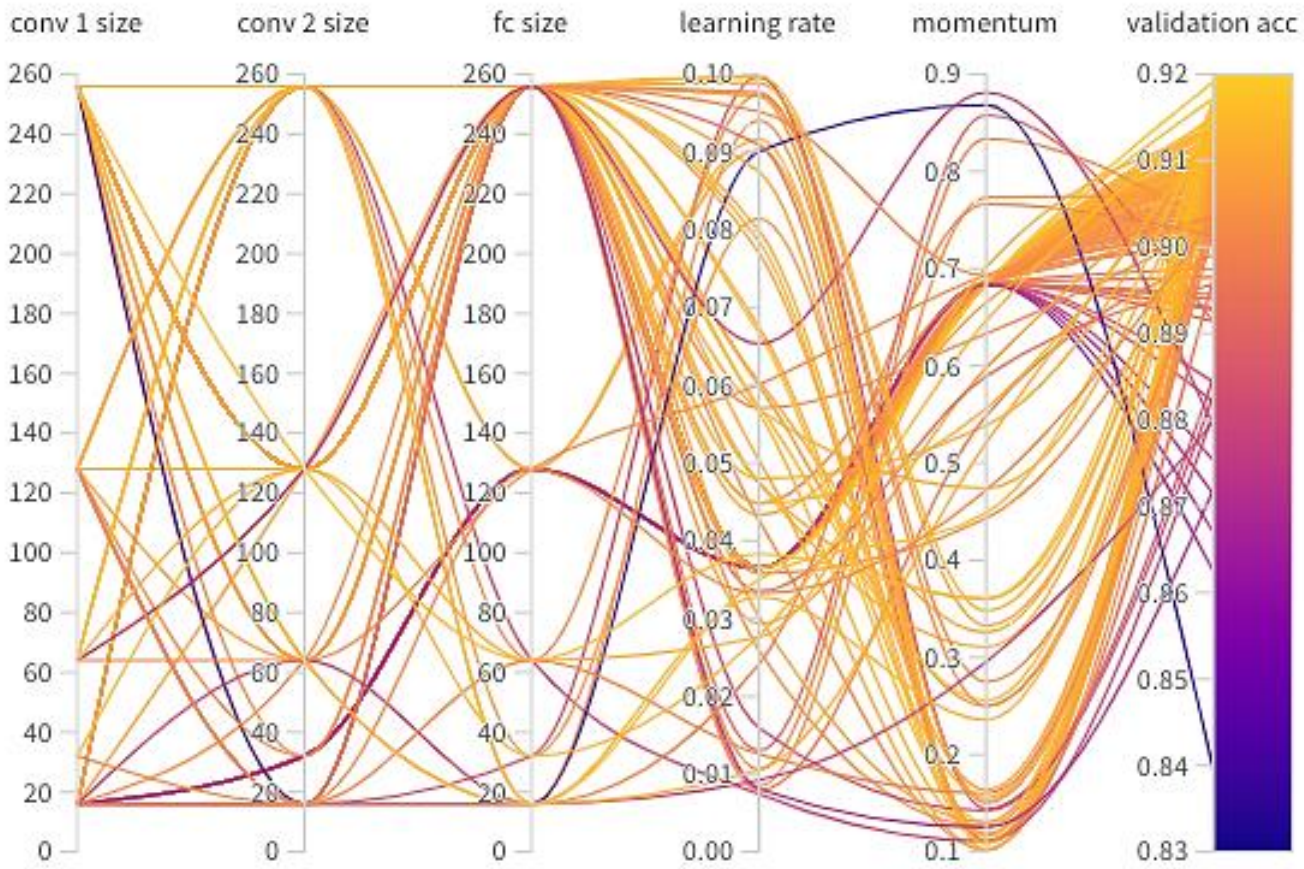
(b) After applying dropout.

# Training: Regularization



# Training: Hyperparameters search



# Hands on: Online step by step tutorials

**Deep learning tutorial:**

- https://www.kaggle.com/learn/intro-to-deep-learning

**Hands on:**

- https://www.tensorflow.org/tutorials/images/cnn
- https://machinelearningmastery.com/building-a-convolutional-neural-network-in-pytorch/

# Online resources for going further

https://medium.com/thedeephub/convolutional-neural-networks-a-comprehensive-guide-5cc0b5eae175

https://viso.ai/deep-learning/convolution-operations/

https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c

https://towardsdatascience.com/multilayer-perceptron-explained-with-a-real-life-example-and-python-code-sentiment-analysis-cb408ee93141