Deep Learning & Beyond

AI FUNDAMENTALS

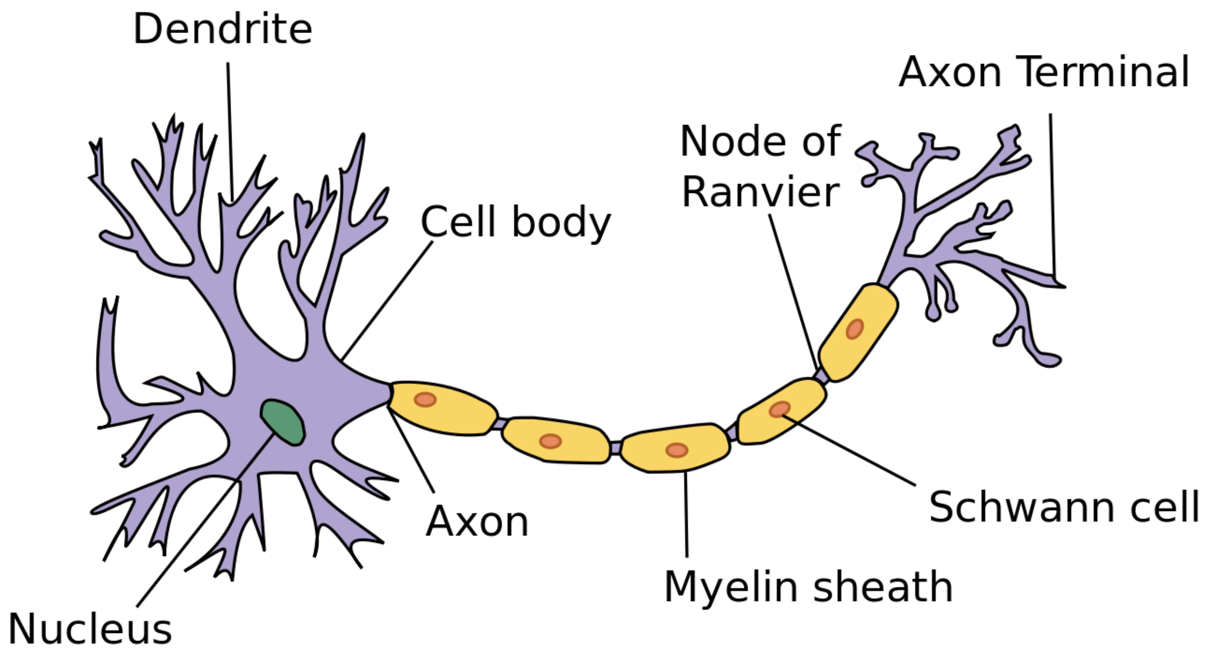
Nemanja Radojkovic
Senior Data Scientist

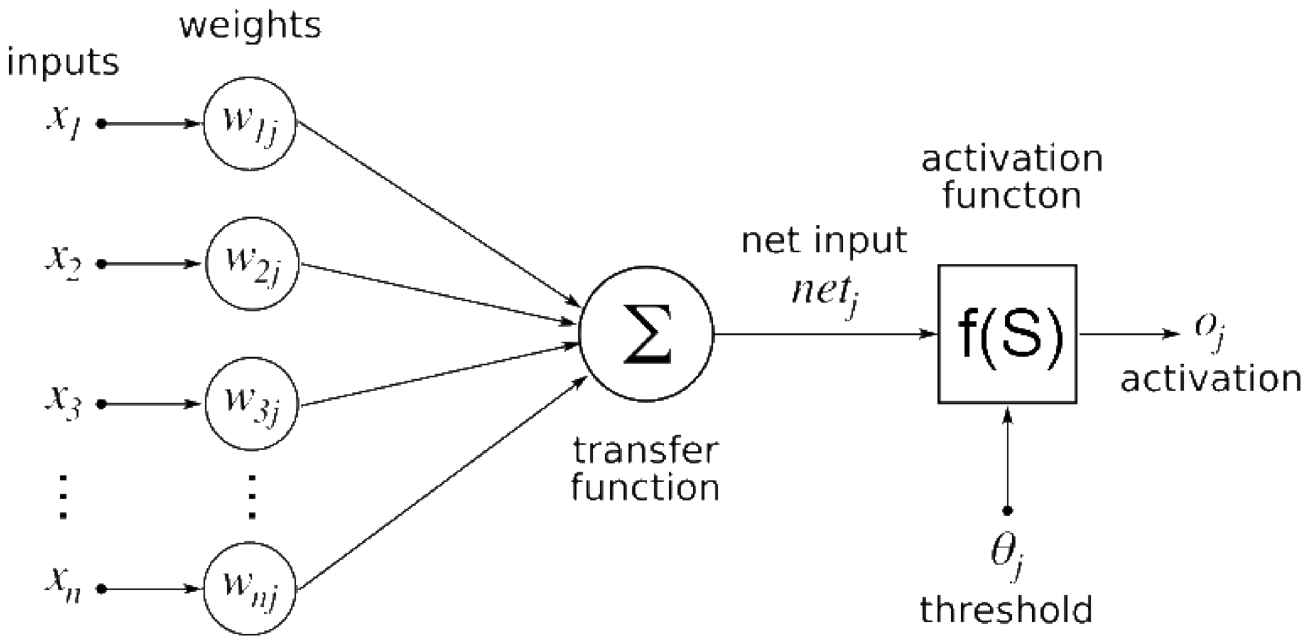
Brief history of Neural Networks

- **1958:** Artificial Neural Networks invented by psychologist Frank Rosenblatt, inspired by human perception processes.
- **1986:** Rumelhart, Williams and Hinton co-author a paper that popularizes the **backpropagation algorithm**.
- **2012:** a convolutional neural network (CNN) called AlexNet wins the ImageNet 2012 Challenge.

> *"Suddenly people started to pay attention, not just within the AI community but across the technology industry as a whole." ~ The Economist*

The building blocks





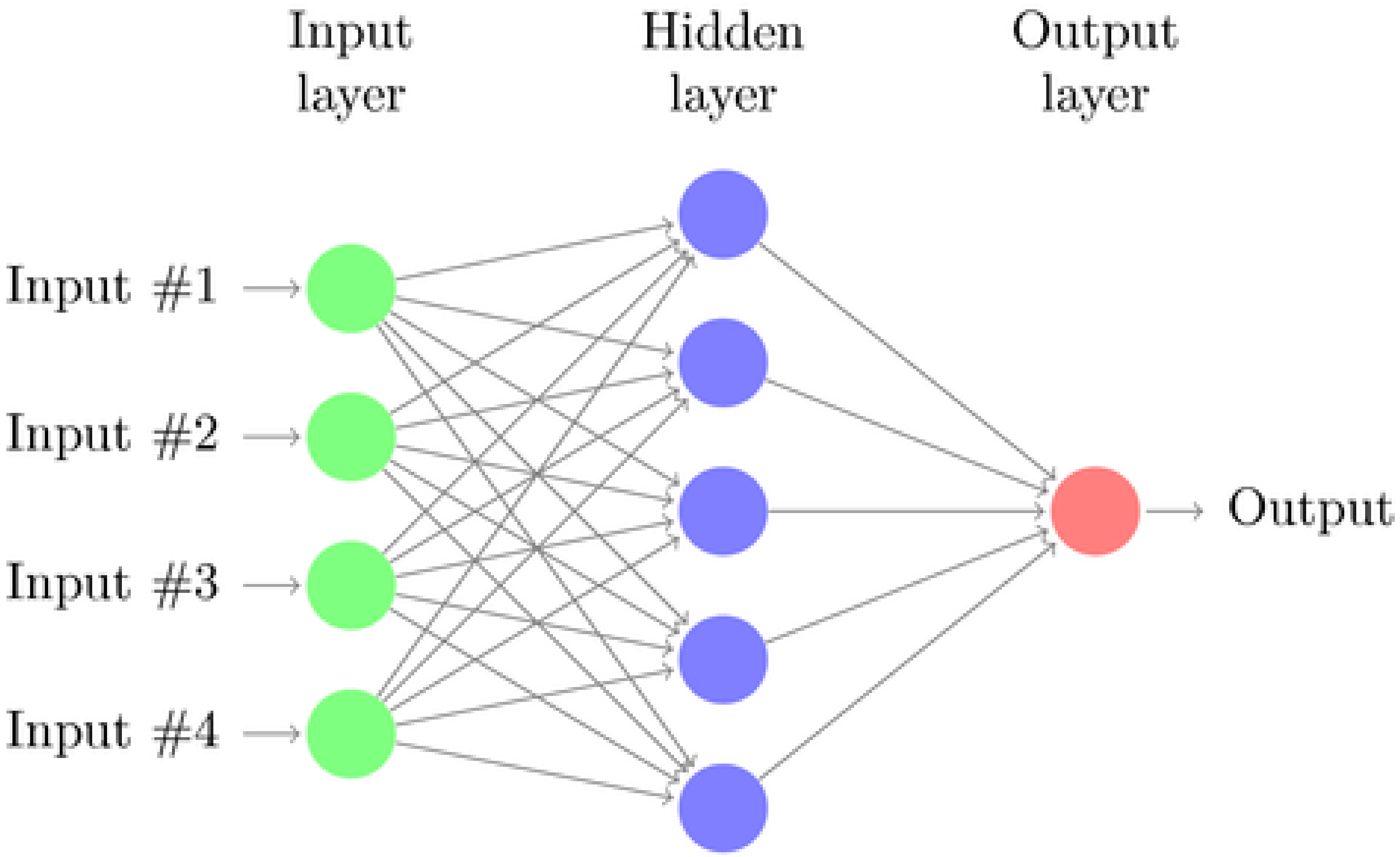
Human neuron

- Multiple dendrites (inbound signal paths)
- Nucleus (the processing unit)
- Single axon (outbound signal path)

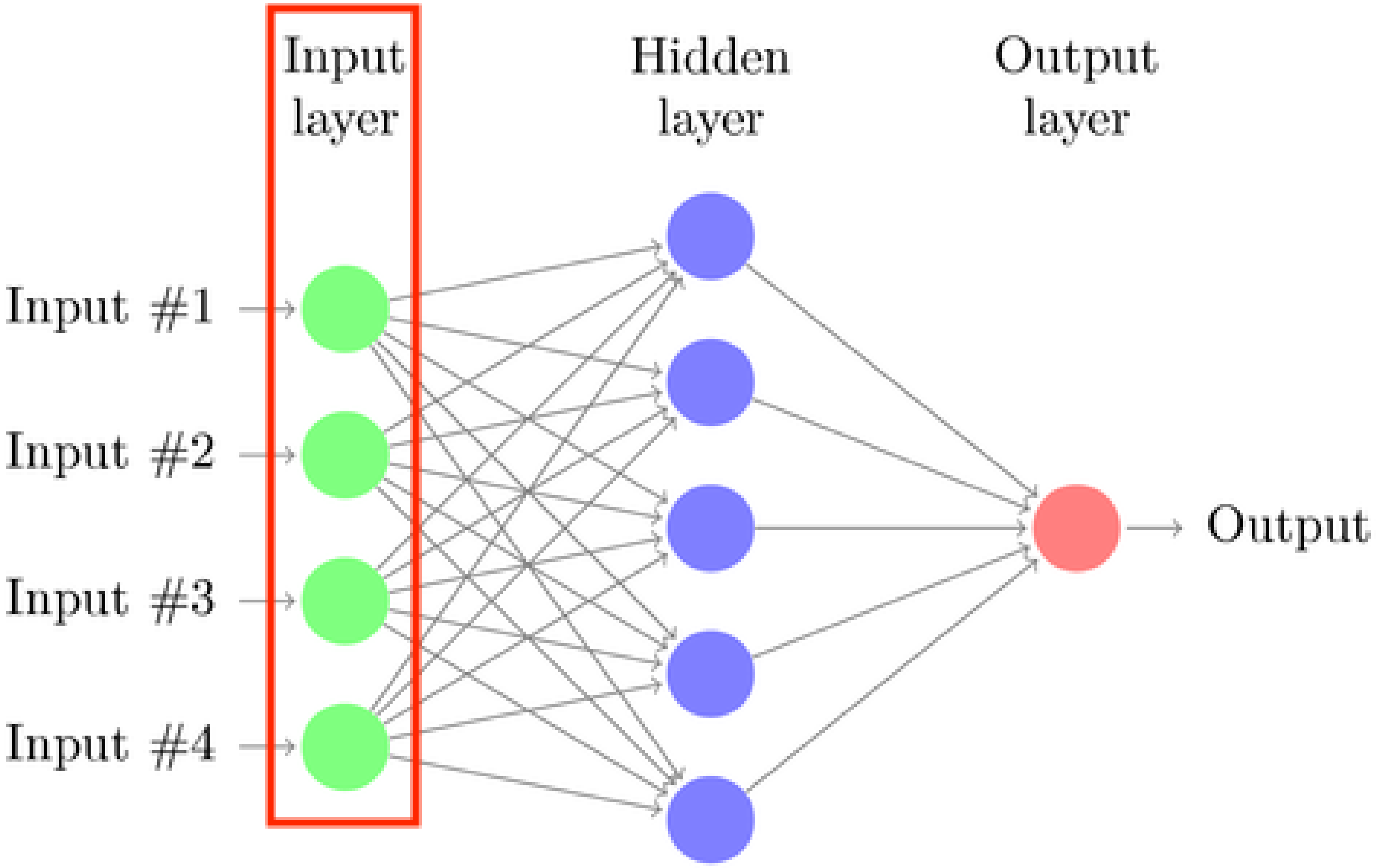
Artificial neuron

- Multiple inputs
- Transfer and activation functions
- Single output

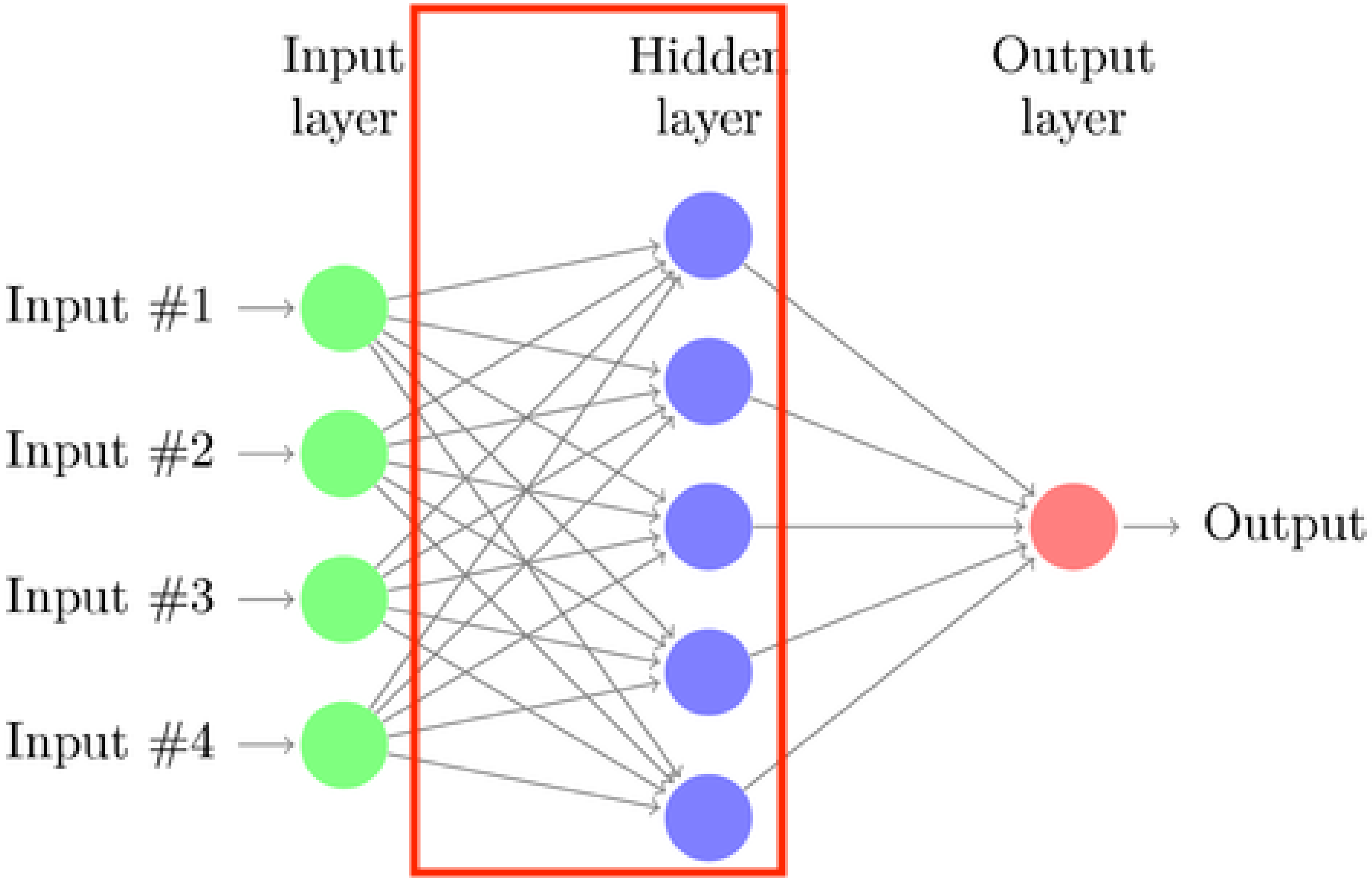
The basic network structure



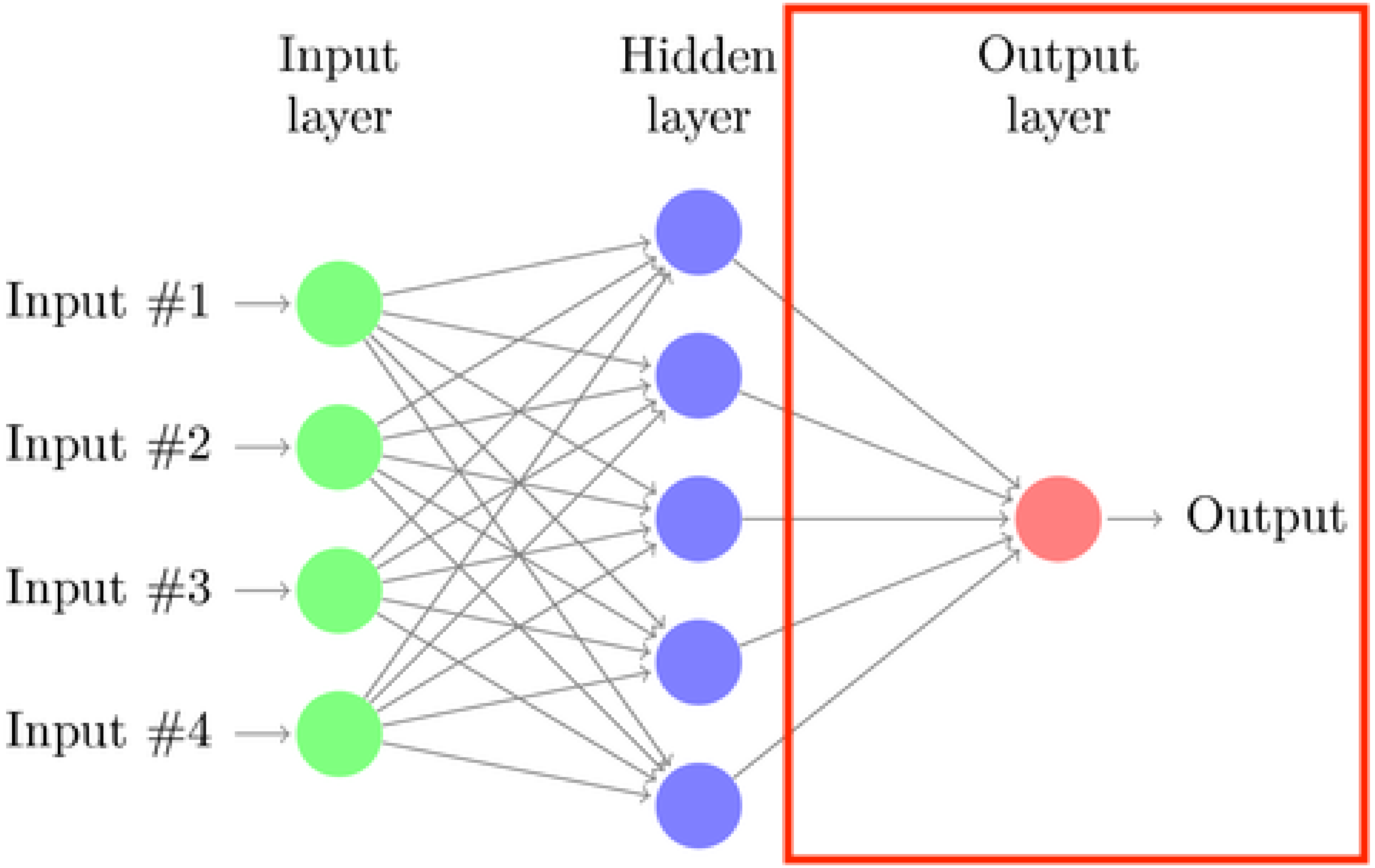
The basic network structure - input layer



The basic network structure - hidden layer



The basic network structure - output layer



How do we make them?

```
# Import the necessary objects from Tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense


# Initialize the sequential model
model = Sequential()


# Add the HIDDEN and OUTPUT layer, specify the input size and the activation function
model.add(Dense(units=32, input_dim=64, activation='relu')) # relu = REctified Linear Unit
model.add(Dense(units=3, activation='softmax'))


# Prepare the model for training (multi-class classification problem)
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Your turn!

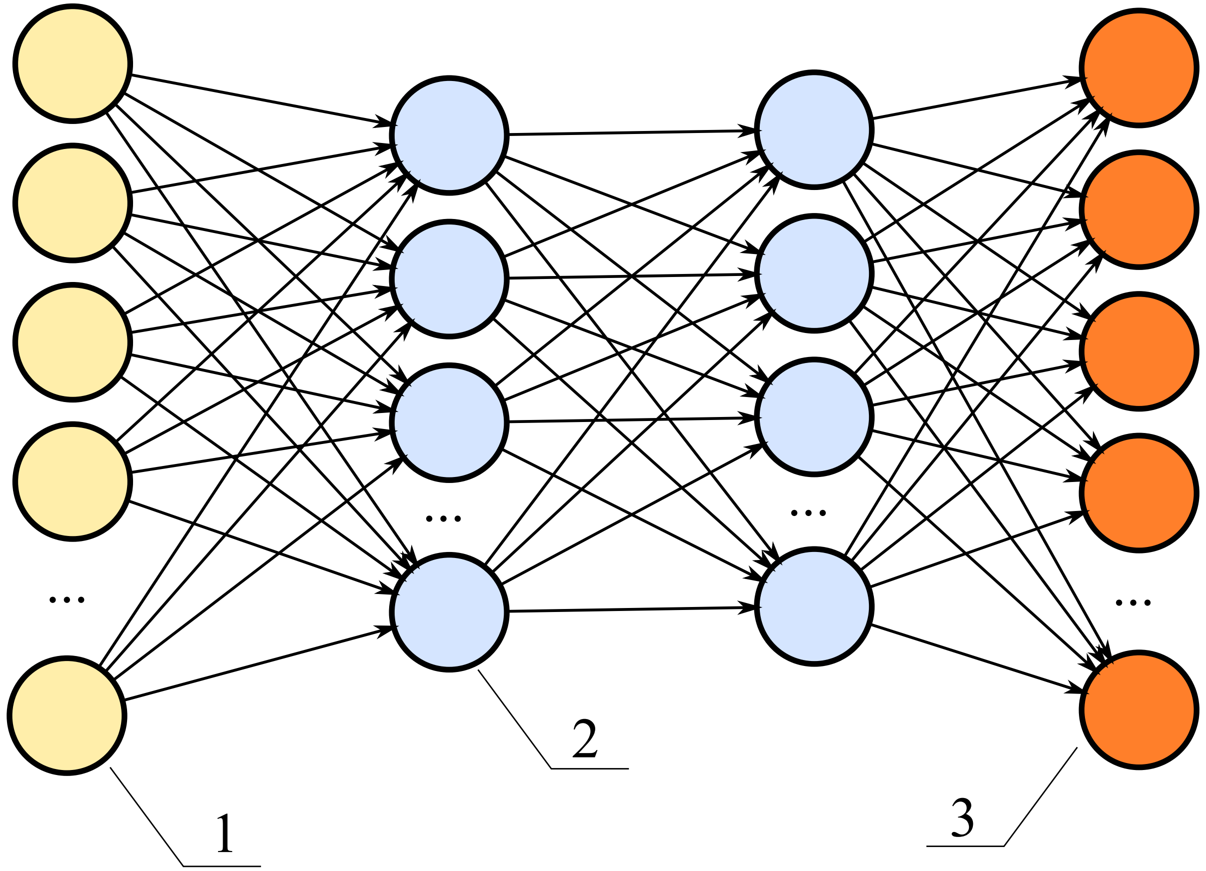
AI FUNDAMENTALS

Deep Learning

AI FUNDAMENTALS

Nemanja Radojkovic
Senior Data Scientist

Deep Neural Networks: what are they?



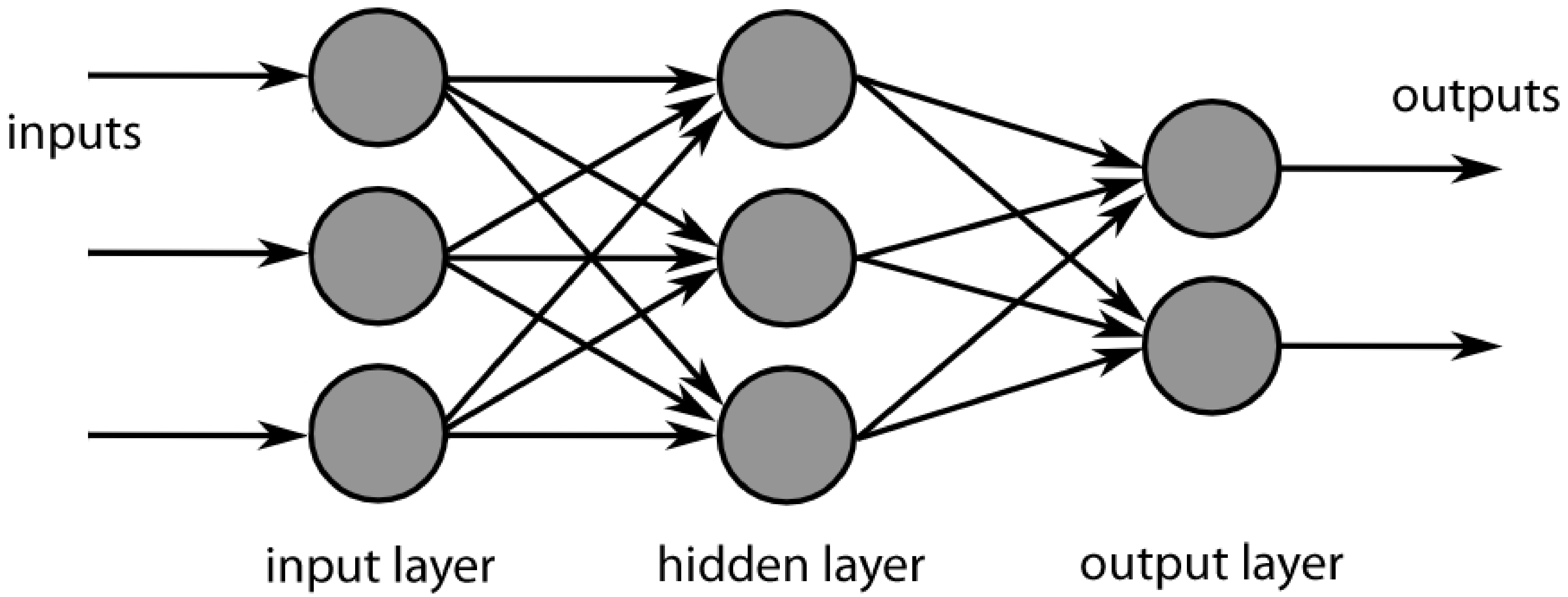
Shallow networks:

- 2-3 layers

Deep Neural Networks

- 4+ layers

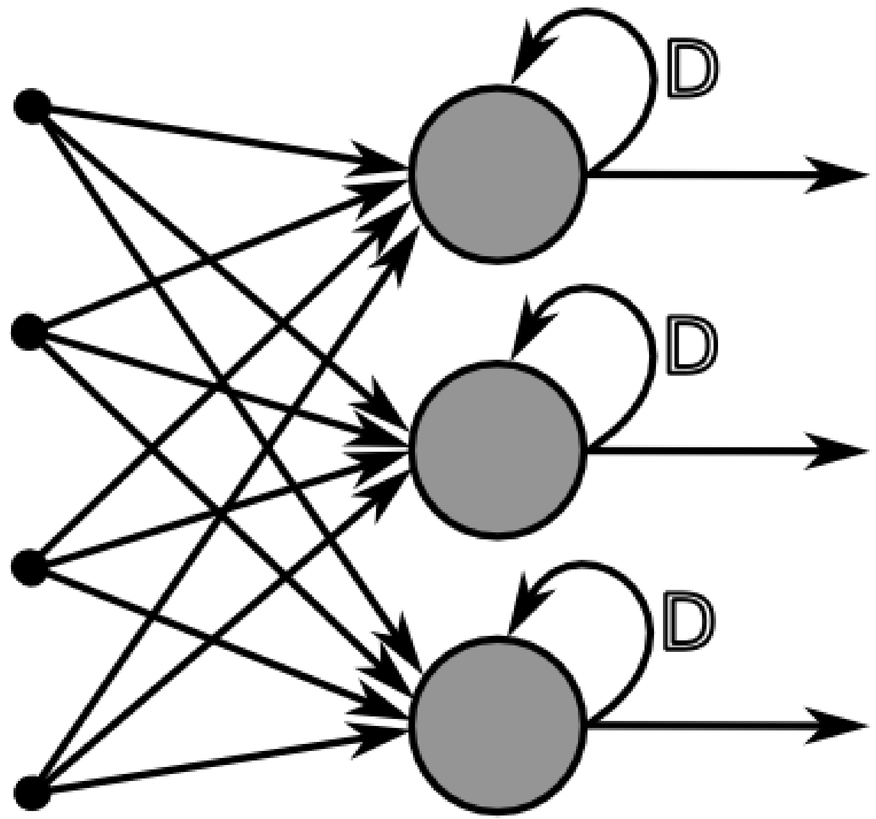
Types of DNNs: Feedforward



Applications: General purpose.

Weak spot: Images, text, time-series.

Types of DNNs: Recurrent

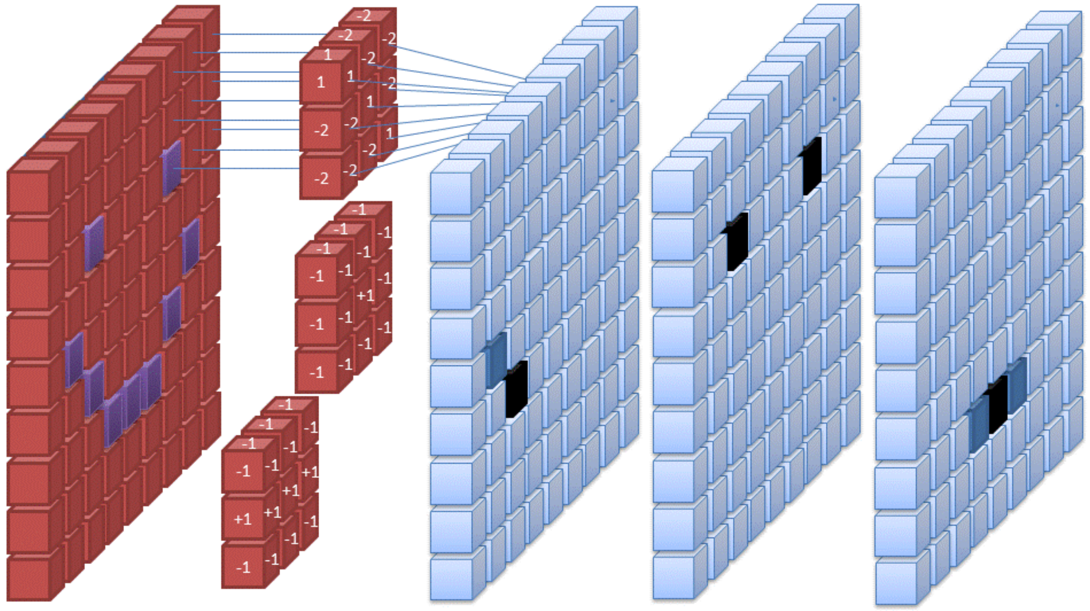


output layer

Applications:

- Speech
- Text

Types of DNNs: Convolutional



- Image/Video
- Text

Layers and layers

1. Dense: `tensorflow.keras.layers.Dense`
   - Single-dimensional feature extraction, signal transformation.
2. Convolutional: `tensorflow.keras.layers.Conv1D, Conv2D, ...`
   - Multi-dimensional, shift-invariant feature extraction, signal transformation.
3. Dropout: `tensorflow.keras.layers.Dropout`
   - Overfitting prevention by randomly turning off nodes.
4. Pooling/sub-sampling: `tensorflow.keras.layers.MaxPooling1D, MaxPooling2D, ...`
   - Overfitting prevention by sub-sampling.
5. Flattening: `tensorflow.keras.layers.Flatten`
   - Converting multi-dimensional to single-dimensional signals

Your first Deep Learning model

```python
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import (Dense, Conv2D, MaxPooling2D, Flatten)


# Initialize the model
model = Sequential()


# Create your 5-layer network (input specified implicitly with 1st layer)
model.add(Conv2D(64, kernel_size=3, activation='relu', input_shape=(28,28,1)))
model.add(MaxPooling2D(pool_size=(2, 2),strides=(2, 2)))
model.add(Flatten())
model.add(Dense(10, activation='softmax'))


# Set fitting hyper-parameters and compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Let's practice!

AI FUNDAMENTALS

Convolutional Neural Networks

AI FUNDAMENTALS

Nemanja Radojkovic
Senior Data Scientist

Convolution

> Mathematical operation describing how signals are transformed by passing through systems of different characteristics.

Inputs:

1. Input signal (video, audio...)
2. Transfer function of the processing system (lens, phone, tube...)

Result: The processed signal

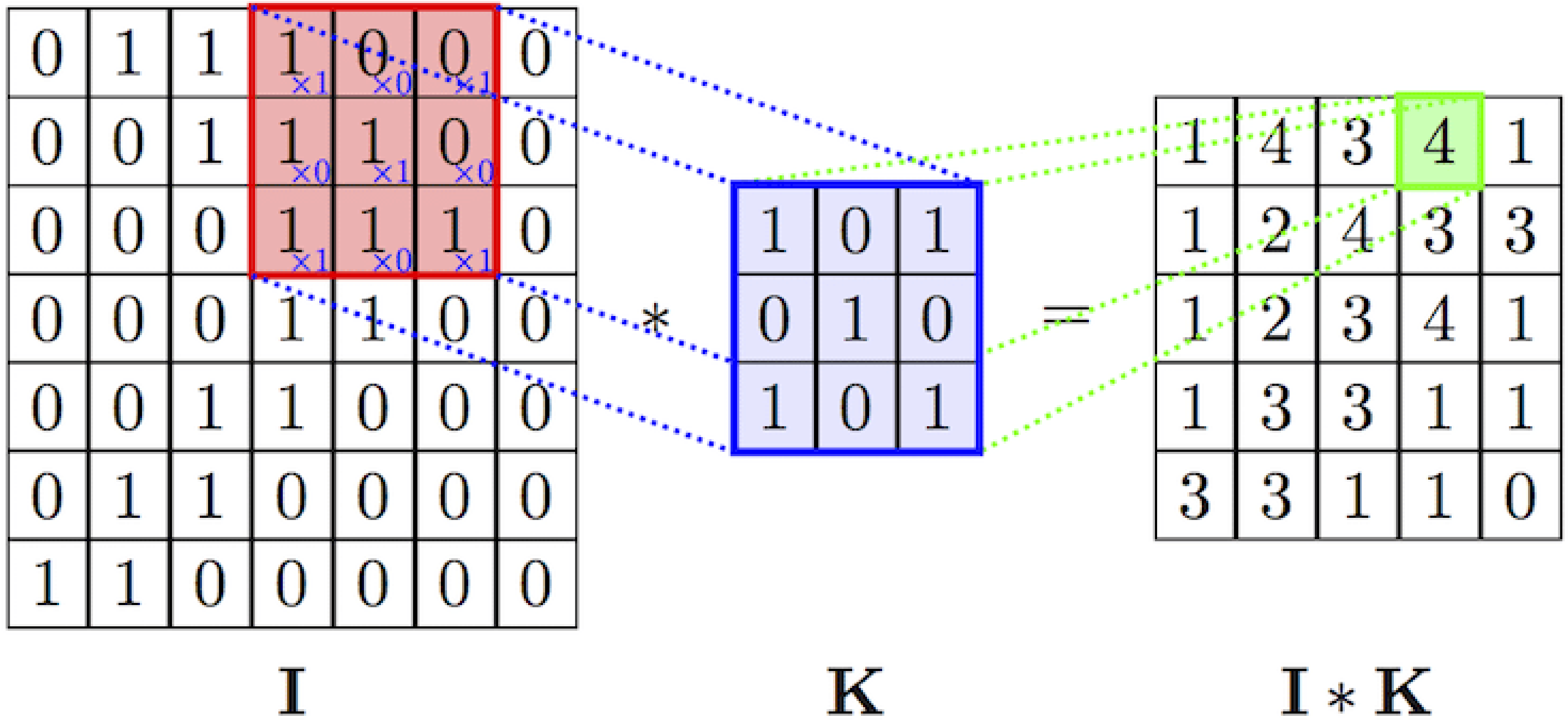
Example: Simulating the "telephone voice"

- *Convolution(raw audio, telephone system transfer function)*

Convolution on images: Kernels

Convolution ~ Filtering

Kernel = Filter ("lens")



0	1	1	1	0	0	0
0	0	1	1	1	0	0
0	0	0	1	1	1	0
0	0	0	1	1	0	0
0	0	1	1	0	0	0
0	1	1	0	0	0	0
1	1	0	0	0	0	0

$\mathbf{I}$

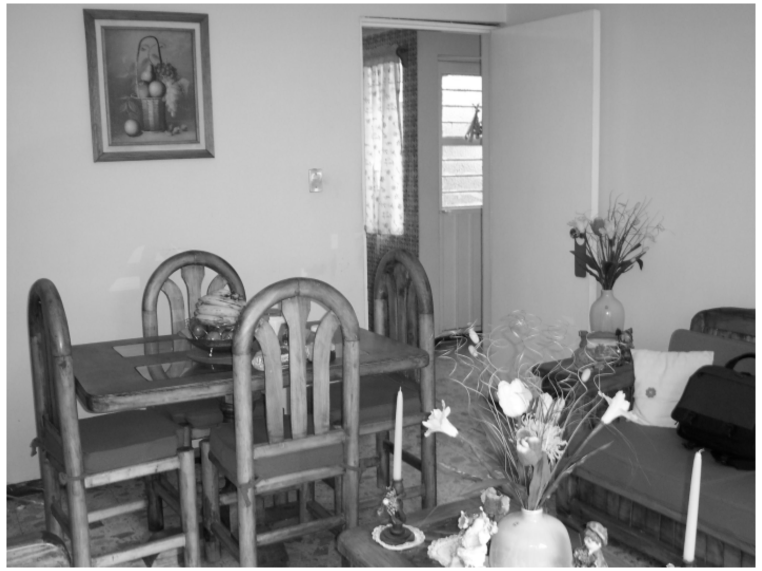
1	0	1
0	1	0
1	0	1

$\mathbf{K}$

1	4	3	4	1
1	2	4	3	3
1	2	3	4	1
1	3	3	1	1
3	3	1	1	0

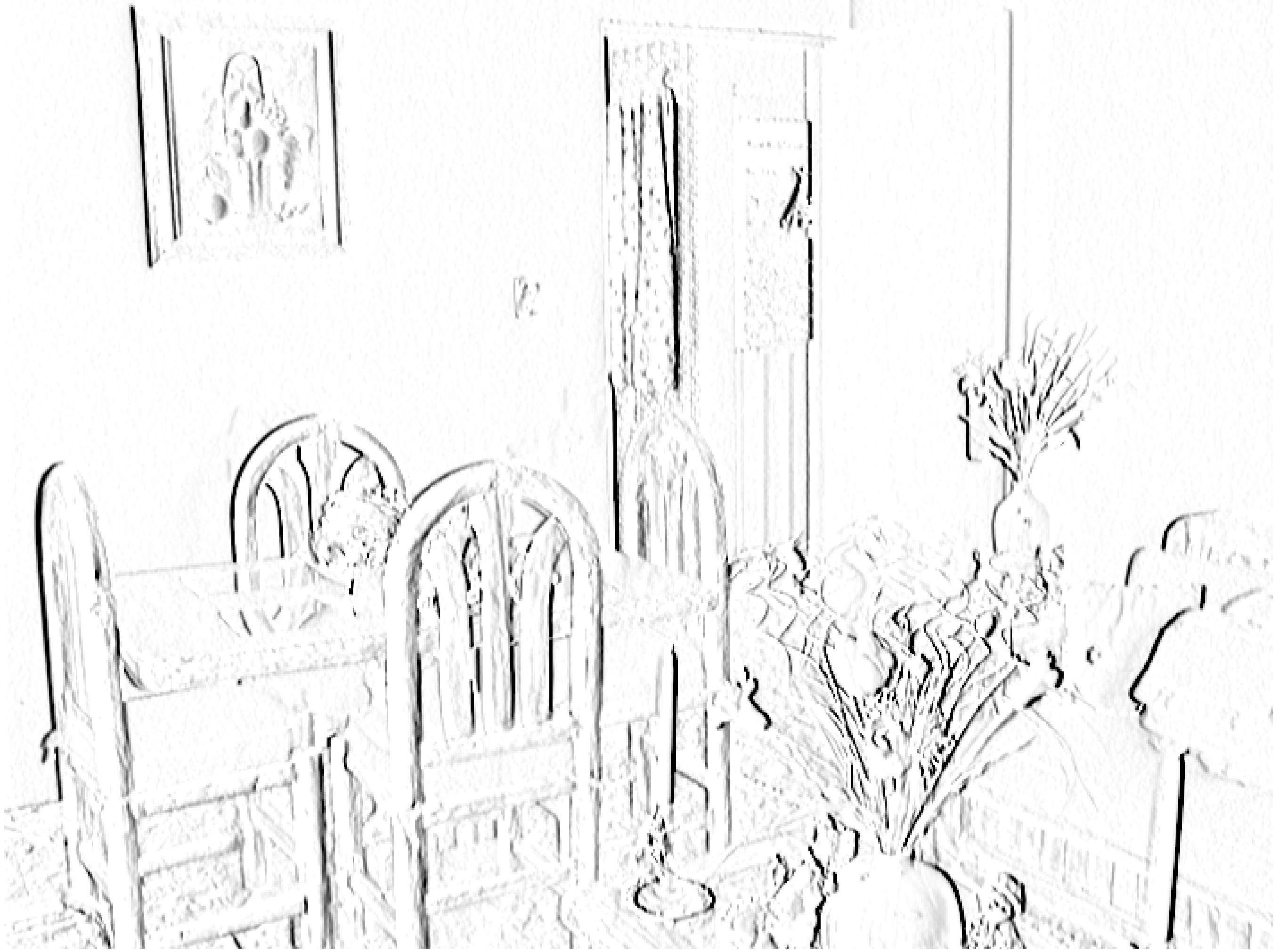
$\mathbf{I} * \mathbf{K}$

Example: Vertical edge detection



+

-1	-2	-1
0	0	0
-1	-2	-1



The beauty of it all

Traditional Computer Vision:

- Deterministic pre-processing and feature extraction, hard-coded by the Computer Vision engineer through hours and hours of experimentation with different approaches.

Computer Vision, the Deep Learning Way:

- Get tons of labelled images and let the algorithm find the optimal kernels on its own.
- Kernels == feature extractors.
- Downside: Very data "hungry"!

Let's practice!

AI FUNDAMENTALS

Congratulations!

AI FUNDAMENTALS

Nemanja Radojkovic
Senior Data Scientist

The journey has just begun!

- Data extraction
- Data wrangling
- Time series analysis
- ...

Have fun learning!

AI FUNDAMENTALS