NOKIA

An introduction to Deep Learning

Mathieu Goutay
Nokia Bell Labs France

INSA Lyon

Follow the course at

mgoutay.github.io

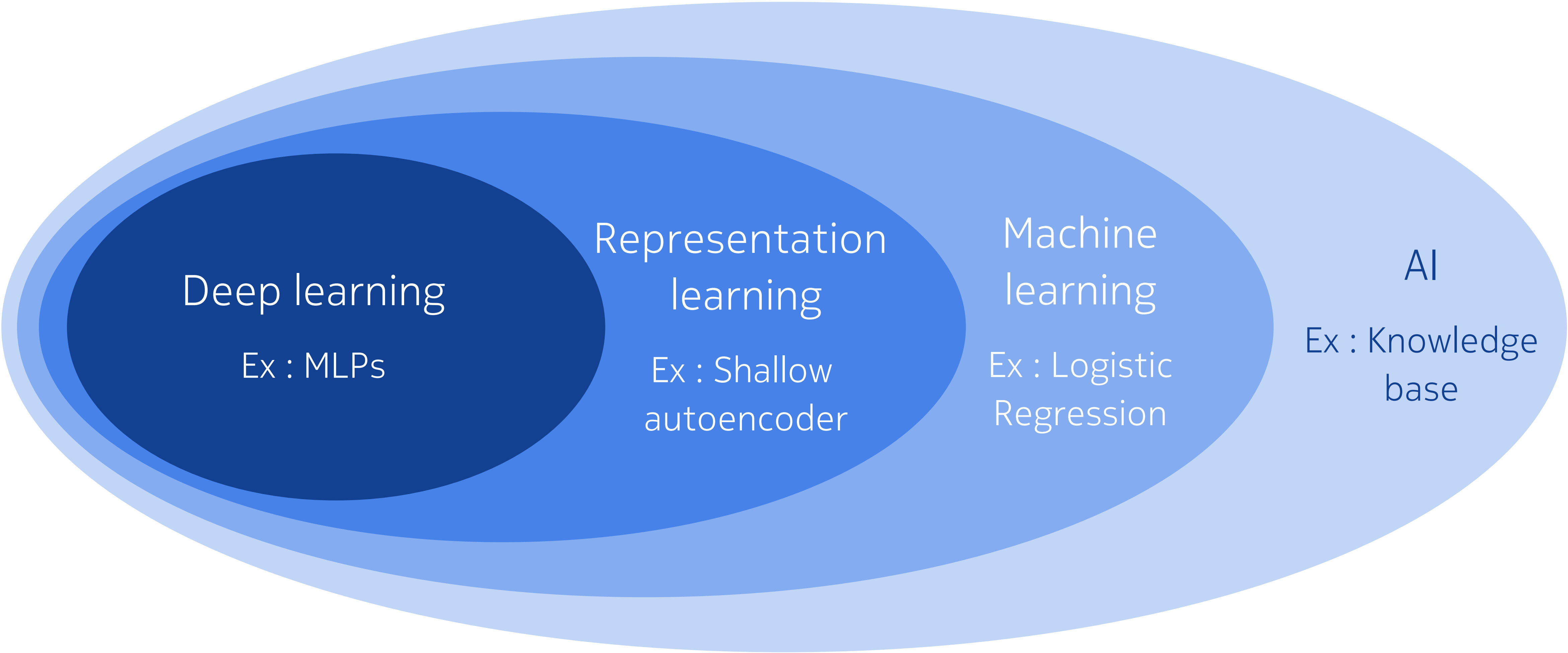
→Blog → Machine Learning Course

Deep Learning

Lesson

1. Introduction to Deep Learning : slides are available here
2. Tensorflow for beginners - Tensorflow for experts : notebook is available here

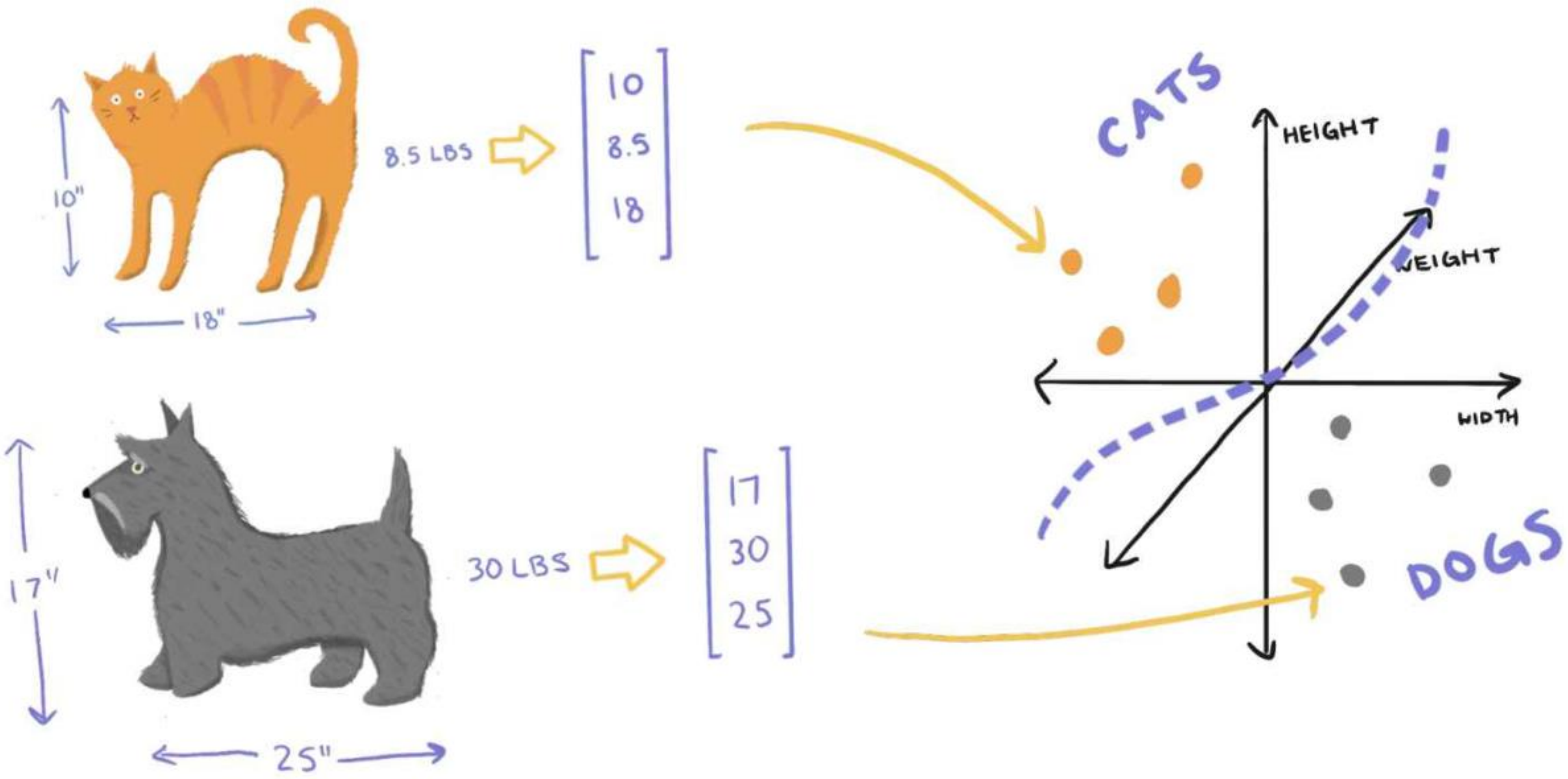
1. Introduction to Deep Learning



1. Introduction to Deep Learning

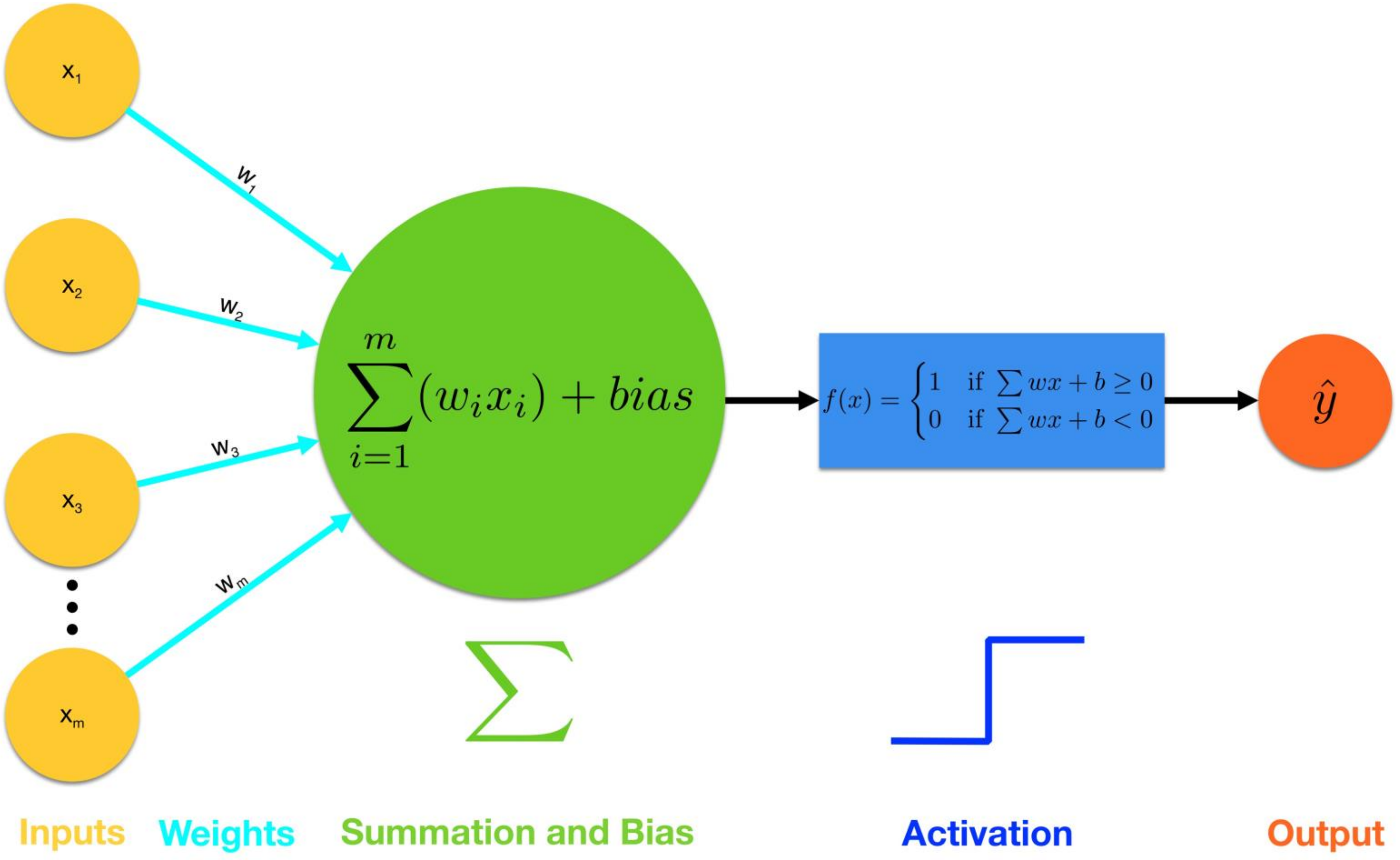
We usually have:

- A set of **Features** : height, weight, width
- A set of **Labels :** Cats, Dogs



1. Introduction to Deep Learning

The basic element is a **Neuron**



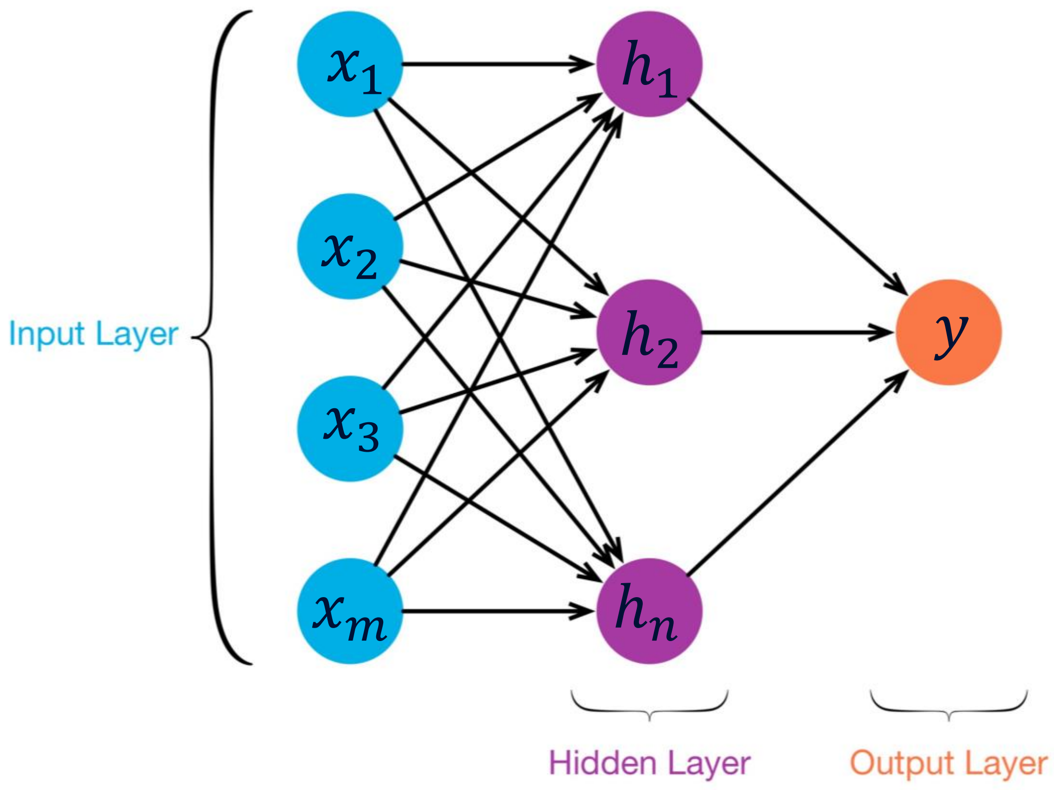
$$\hat{y} = f(w^T x + b)$$

https://towardsdatascience.com/introducing-deep-learning-and-neural-networks-deep-learning-for-rookies-1-bd68f9cf5883

NOKIA

1. Introduction to Deep Learning

Neural Network with 1 hidden **dense layer**



$$y = f(W_o(f(W_h x + b_h) + b_o)$$

https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f

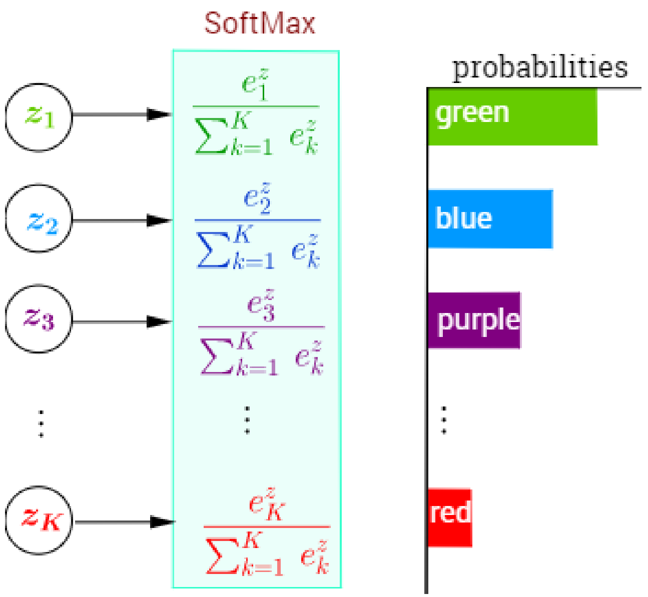
1. Introduction to Deep Learning

Other types of layers:

- **SoftMax:**

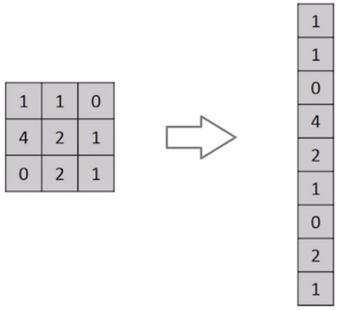
  Takes a vector $\boldsymbol{x}$ and applies the function $\mathbf{y}_\mathrm{i} = \frac{e^{x_i}}{\sum_k e^{x_k}}$

  Outputs probabilities (all $\boldsymbol{y}_i > 0$ and $\sum \boldsymbol{y}_i = 1$)



- **Flatten :**

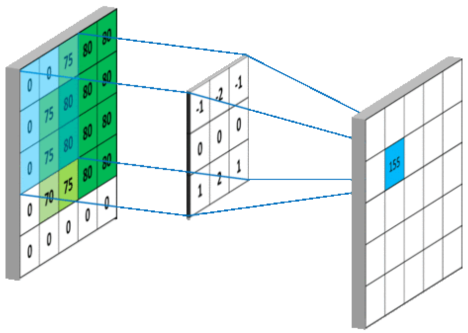
  Converts the inputs into a dimension [batch_size, -1]



- **2D Convolutional :**

  Multiplies the 2D inputs by N kernels creating N 2D outputs

  Parameters : # of kernels N, kernel size, …

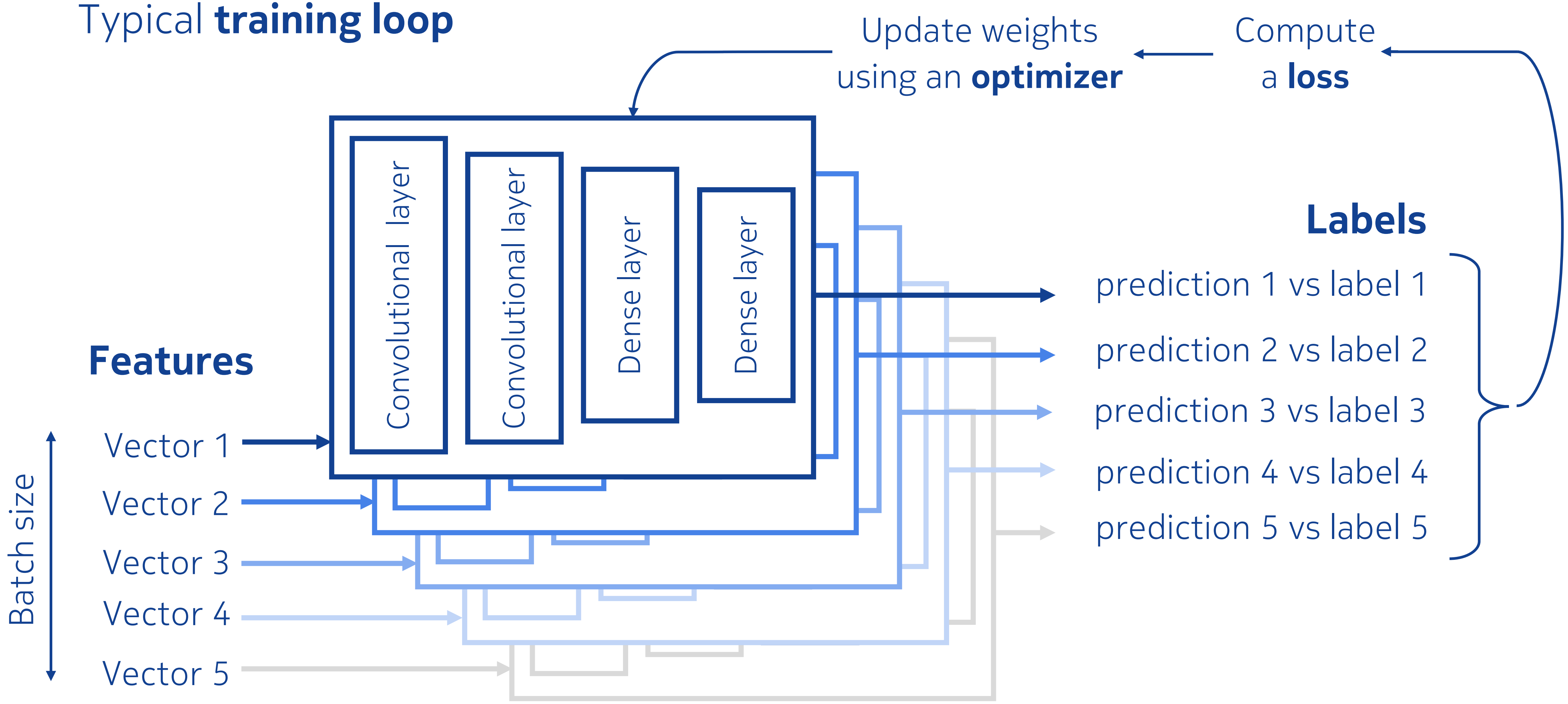


- **Etc.**

  https://mlnotebook.github.io/post/CNN1/ NOKIA

1. Introduction to Deep Learning

Typical **training loop**



Multiple copy of the **same** Neural Network

1. Introduction to Deep Learning

The **loss function** depends on the task

(for a batch: compute forward path → compute **loss** & gradients → apply optimizer)

- For a regression problem: Mean Squared Error, etc.

$$\mathrm{MSE} = \frac{1}{B_s}\sum_{i=1}^{B_s}(y_i - \hat{y}_i)^2$$

- For a classification problem : Cross Entropy, etc.

Cross-entropy is a measure of the difference between two probability distributions p and $\hat{p}$

$$\mathrm{CE} = \frac{1}{B_s}\sum_{i=1}^{B_s}\sum_{j=1}^{C} - p_{ij}\ \log(\hat{p}_{ij})$$

1. Introduction to Deep Learning

Training is done by Stochastic Gradient Descent (**SGD**) or a variant

(for a batch: compute forward path → compute loss & gradients → apply **optimizer**)

SGD updates the weights in the negative gradient direction

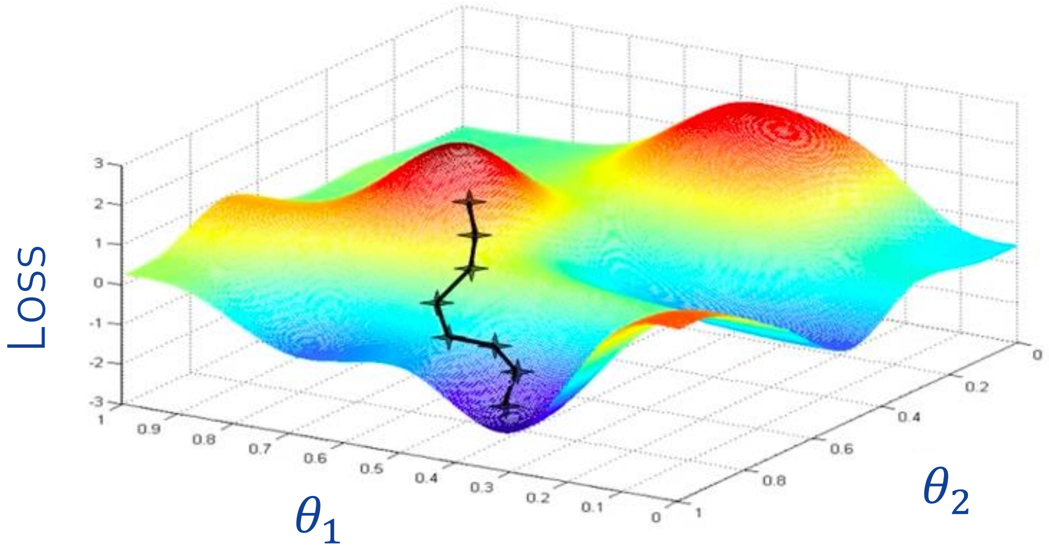
$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t; \hat{y}, y)$$

Learning rate

Gradient of the Loss function

Predictions

Labels



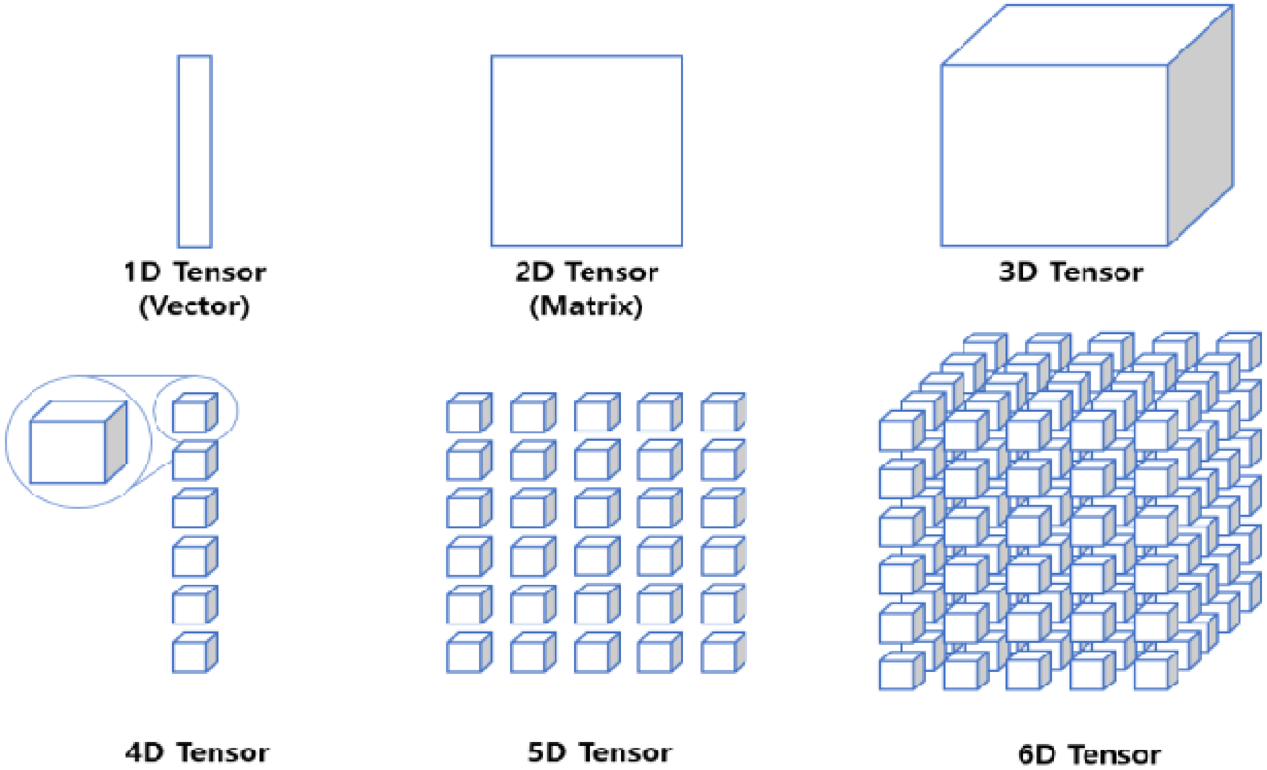
The most used variant is **Adam**:

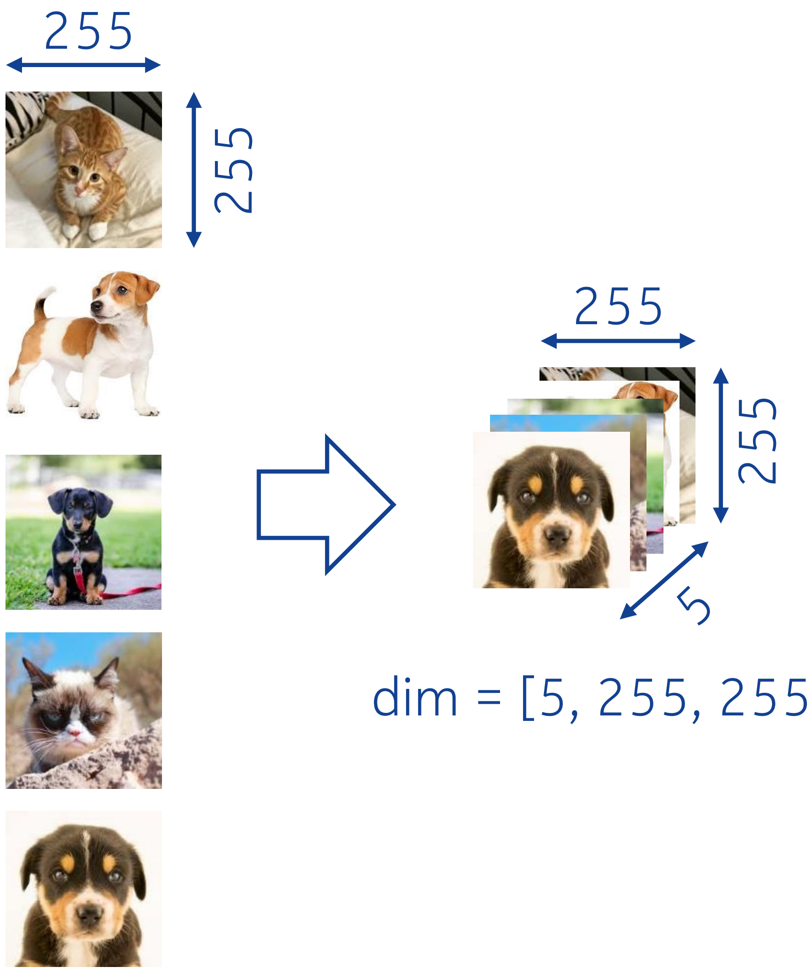
- Individual adaptive learning rate for each parameters
- Exponential moving average of gradients
- Computationally efficient

2. Tensorflow for beginners

A **Tensor** is a N-dimensional Matrix

The first dimension is usually the batch size





dim = [5, 255, 255]

NOKIA

2. Tensorflow for beginners

Tensorflow 2.0 is **very pythonic**

Lots of equivalent functions between Numpy & Tensorflow

Numpy

```
import numpy as np

a = np.array([[2, 2], [2, 2]], dtype=np.int32)
b = np.array([[3, 3], [4, 4]], dtype=np.int32)
c = a*b

print(c)
```

```
[[6 6]
 [8 8]]
```

Tensorflow

```
import tensorflow as tf

a = tf.constant([[2, 2], [2, 2]], dtype=tf.int32)
b = tf.constant([[3, 3], [4, 4]], dtype=tf.int32)
c = a*b

print(c)
```

```
tf.Tensor(
[[6 6]
 [8 8]], shape=(2, 2), dtype=int32)
```

Supported TF data types : bool, string, int8/16/32/64, float32/64, complex64/128, etc.

NOKIA

2. Tensorflow for beginners

The parameters of a NN are created as **Variables**

```
cst = tf.constant([[2, 2], [2, 2]]) # cst is a fixed Tensor
var = tf.Variable([[2, 2], [2, 2]]) # var will be updated during training

print('cst:', cst, '\n')
print('var:', var, '\n')
```

```
cst: tf.Tensor(
[[2 2]
 [2 2]], shape=(2, 2), dtype=int32)

var: <tf.Variable 'Variable:0' shape=(2, 2) dtype=int32, numpy=
array([[2, 2],
       [2, 2]], dtype=int32)>
```


NOKIA

2. Tensorflow for beginners

Keras is a high-level neural networks API

Give access to pre-made Layers

```
from tensorflow import keras
from tensorflow.keras.layers import Conv2D, Flatten, Dense, Softmax
```


For the following :

Deep Learning

Lesson

1. Introduction to Deep Learning : slides are available **here**
2. Tensorflow for beginners - Tensorflow for experts : notebook is available **here**

