DRIVES

Development and Research on Innovative Vocational Education Skills

U3 DEEP LEARNING AND NEURAL NETWORKS

U3.E5 CONVOLUTIONAL NEURAL NETWORK (CNN) AND RECURRENT NEURAL NETWORK (RNN)

Artificial Intelligence Technician

May 2021, Version 1

Co-funded by the Erasmus+ Programme of the European Union

The Development and Research on Innovative Vocational Educational Skills project (DRIVES) is co-funded by the Erasmus+ Programme of the European Union under the agreement 591988-EPP-1-2017-1-CZ-EPPKA2-SSA-B. The European Commission support for the production of this publication does not constitute endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

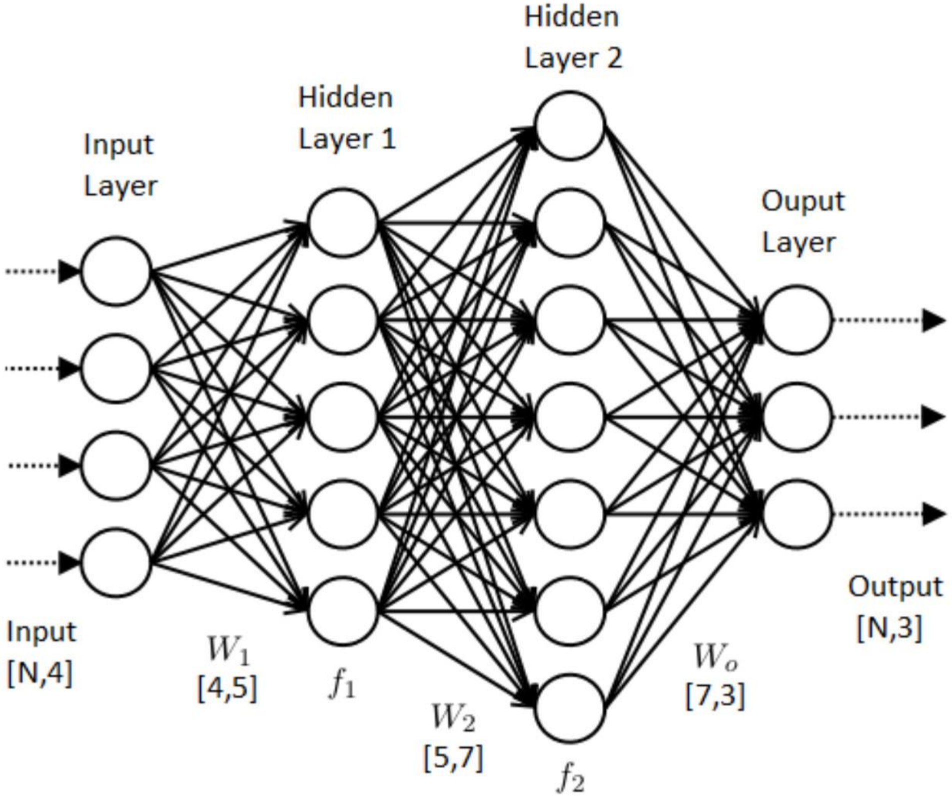
LEARNING OBJECTIVES

The student is able to

AIT.U3.E5.PC1	Define CNN and RNN.
AIT.U3.E5.PC2	Understand the differences between CNNs and RNNs.
AIT.U3.E5.PC3	Know the different use cases of CNNs and RNNs.
AIT.U3.E5.PC4	Select the architecture that best fits a specific problem or situation.
AIT.U3.E5.PC5	Implement RNN and CNN with TensorFlow.

Do you know how deep learning recognizes na object in na image?

Using Convolutional Neural Networks (CNN)



Cat

Bird

Dog

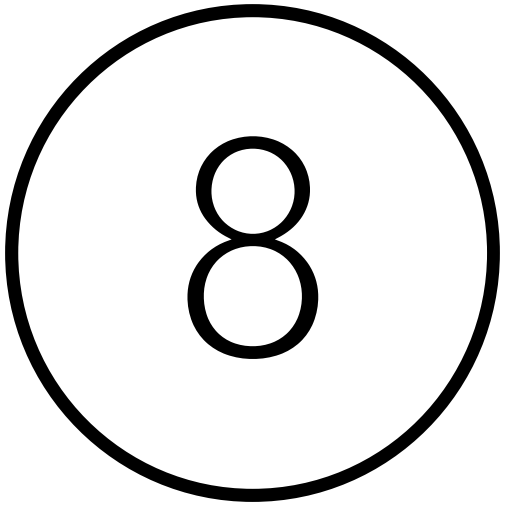
Definition

A convolutional neural network (CNN) is ….

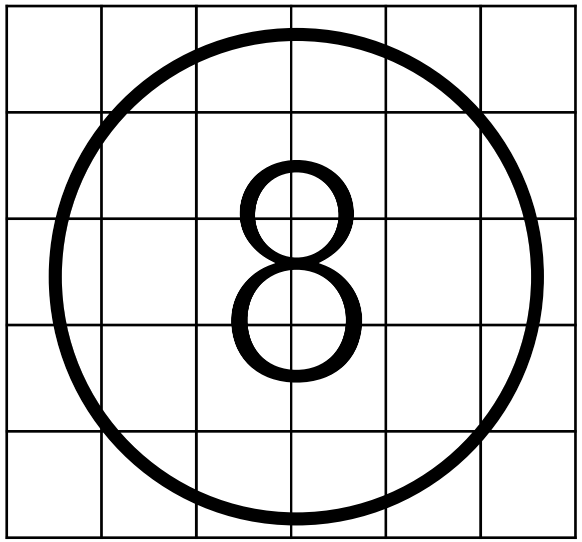
- a feed forward type of artificial neural network used in image recognition and processing;
- designed to take advantage of a picture (2D);
- very used in computer vision, specially in image classification;
- excellent in sequent data analysis such as natural language processing (NLP);
- a specific type of ANN that uses perceptrons, a machine learning unit algorithm, for supervised learning, to analyze data

DRIVES
Development and Research on Innovative Vocational Education Skills

Convolution Operation is the basis of CNN



Real Image of the digit 8



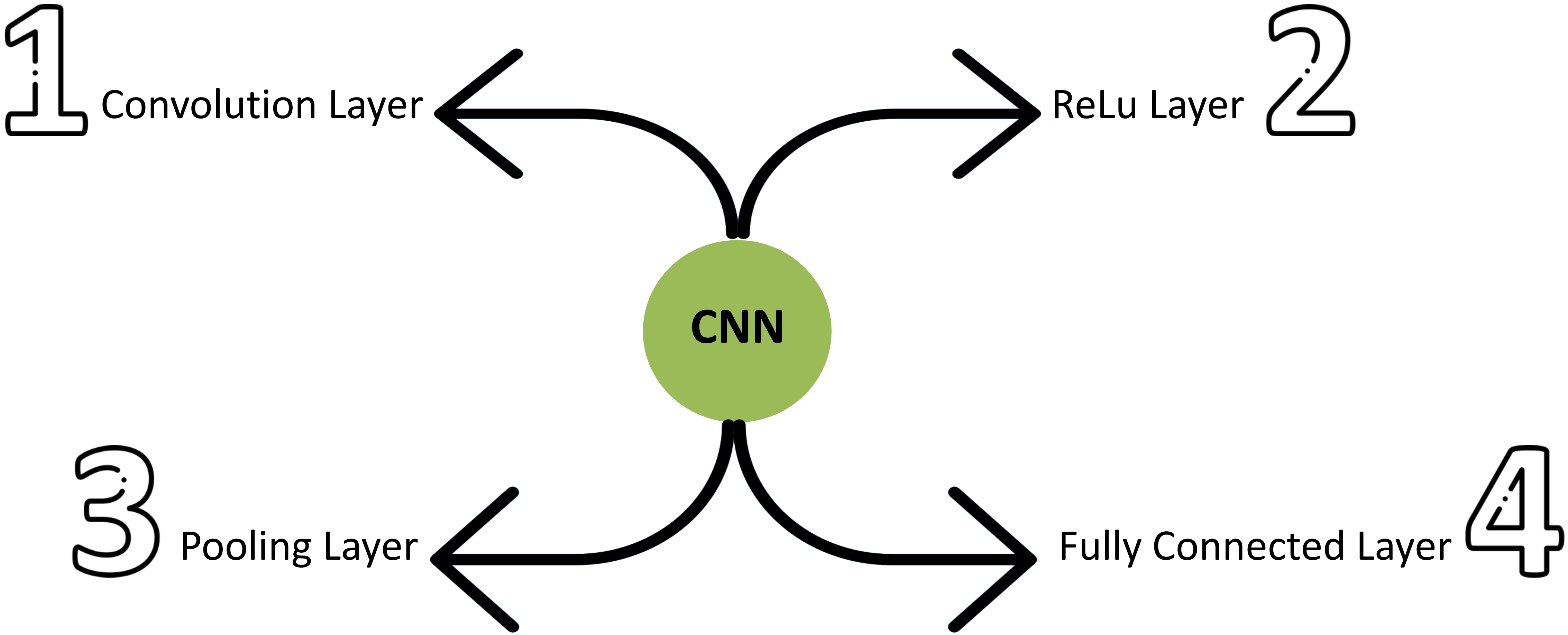
Digit 8 represented in the form of na array

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

Digit 8 represented in the form of pixels (0/1)

In CNN, every image is represented in the form of arrays of pixel values

LAYERS IN CONVOLUTIONAL NEURAL NETWORK



CONVOLUTION LAYER

STEP1: Submit the image into a convolutional Layer, applying filters.

Considering 5 * 5 image:

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

Image Pixels

1	0	1
0	1	0
1	0	1

Filter

4	3	2
2	4	3
2	3	4

Convolved Feature

Sliding the filter matrix over the image and computing the doi product to detec patterns

STEP 2: Once the feature maps are extracted, move the feature into a ReLu Layer (Normalization Layer).

1. Performs element wise operatios;
2. Sets all negative pixels to 0;
3. Introduces non-linearity to the network;
4. The output is a rectified feature map;

4	3	2
2	4	3
2	3	4

DRIVES

Development and Research on Innovative Vocational Education Skills

STEP 3: The rectified feature map goes trough a pooling layer. Pooling is a down-sampling operation that reduces the dimention of the feature map.

4	3	2
2	4	3
2	3	4

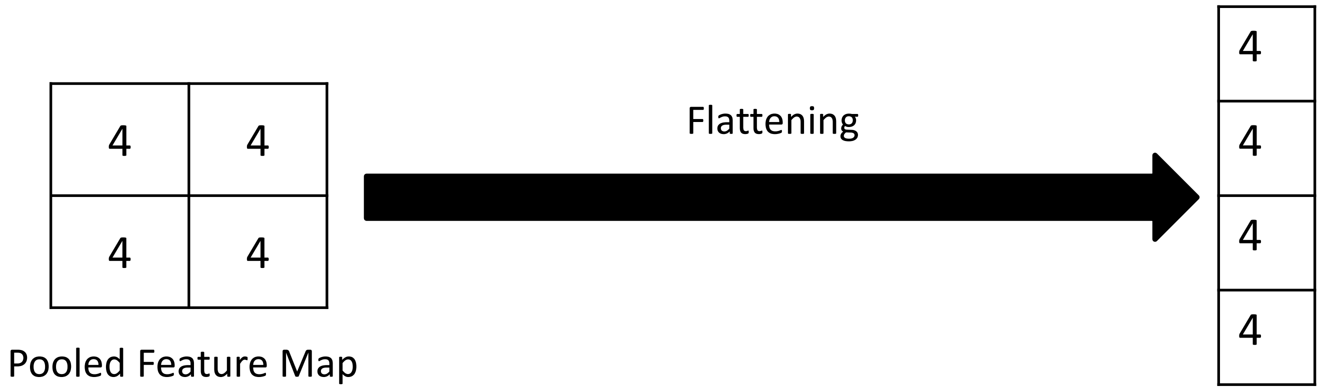
Rectified Feature Map

Max Pooling with 2x2 filters stride 2

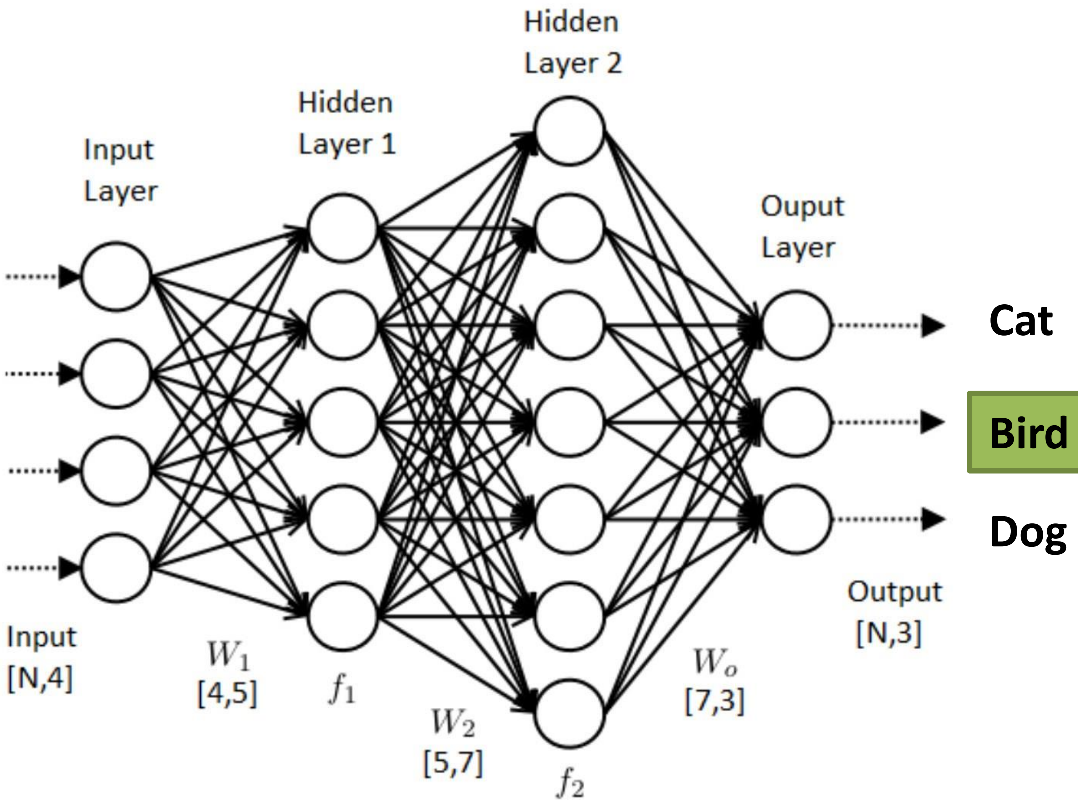
4	4
4	4

Polling Layer uses different filters to identify different parts of na image (edges, corners, body, eyes, ...)

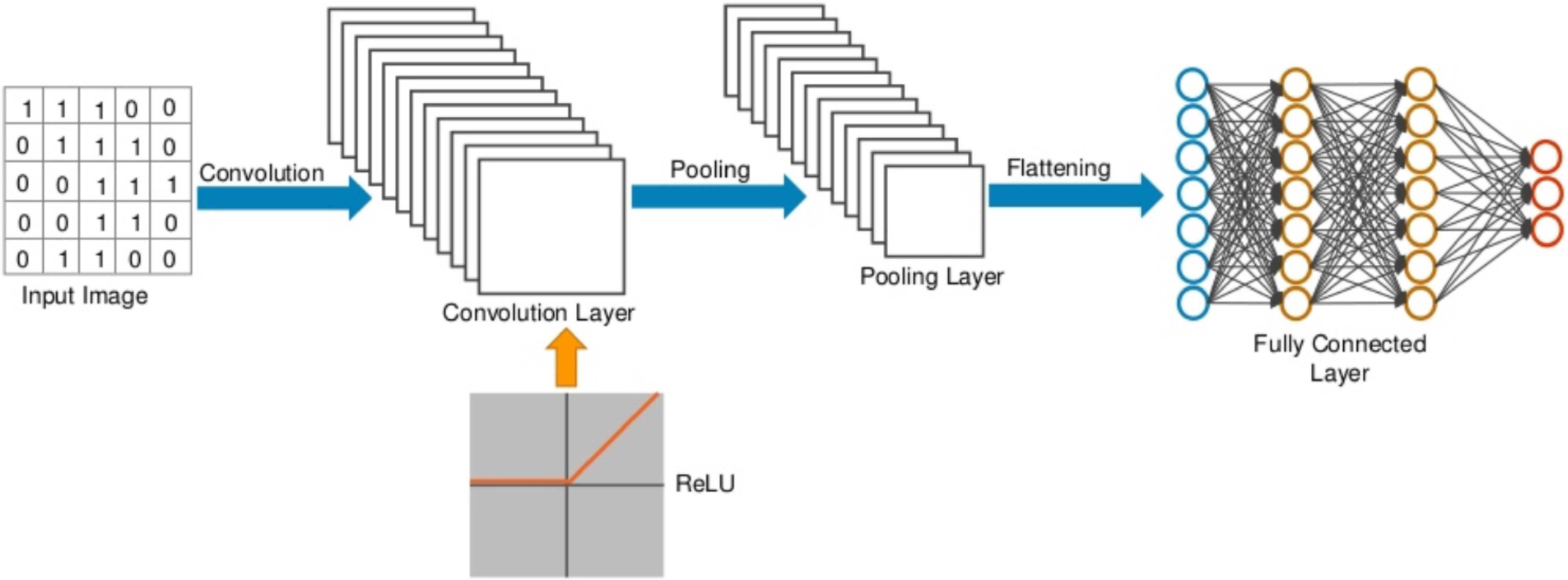
STEP 4: Flattening is the process of conversion of all the resultant 2D arrays from pooled feature map into a single continuous linear vector.



STEP 5 : The flattened matrix from the pooing layer is fed as input to the Fully Connected Layer to classify the image



SUMMARY



1 1 1 0 0
0 1 1 1 0
0 0 1 1 1
0 0 1 1 0
0 1 1 0 0
Input Image
Convolution
Convolution Layer
ReLU
Pooling
Pooling Layer
Flattening
Fully Connected Layer

Advantages

01 Very High accuracy in image recognition problems.

02 Automatically detects the important features without any human supervision.

03 Weight sharing.

Disadvantages

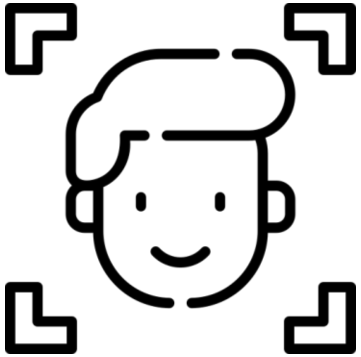
01 Do not encode the position and orientation of object.

02 Lack of ability to be spatially invariant to the input data.

03 Lots of training data is required.

CNN USE CASES

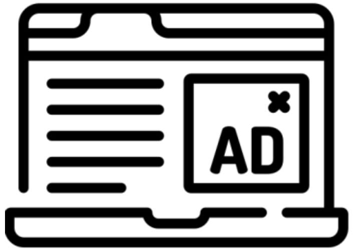
Decoding Facial Recognition



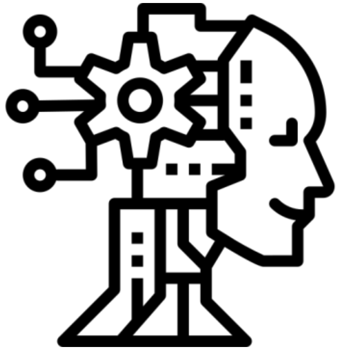
Document Analysis

Understanding Climate

Advertising



Robots that can mimic human behavior



Autonomous cars

RNN USE CASES

Customer support

Project Documentation

Chatbots

DRIVES
Development and Research on Innovative Vocational Education Skills

Definition

Recurrent Neural Network is ...

- a type of neural network that contains loops so that information can be stored in the network;
- commonly used in speech recognition and natural language processing (NLP);
- is a class of artificial neural networks in which the connections between nodes form a directed graph over a time sequence;

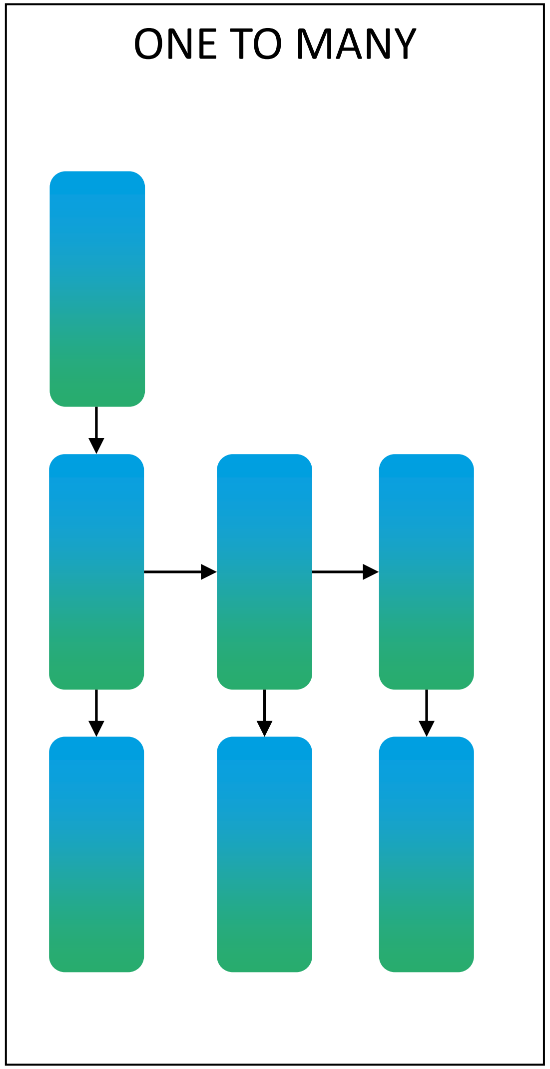
ONE TO ONE

Single Input

Single Output

This type or RNN is known as the Vanila Neural Network.

It is mostly used for regular machine learning problems



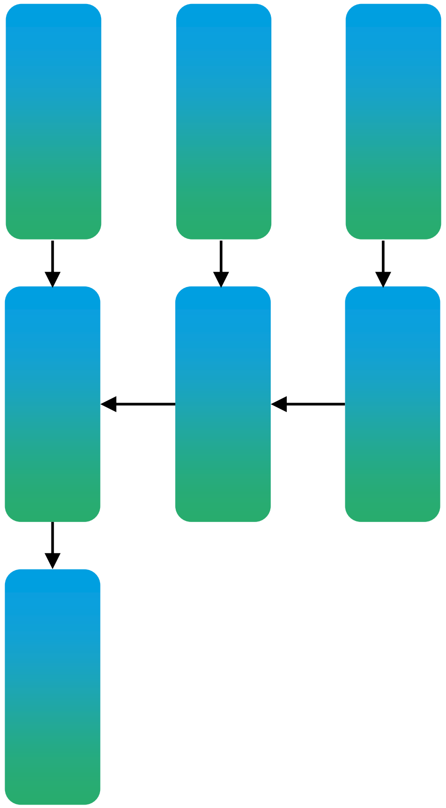
Single Input

Multiple Outputs

RNN, one to many type, generates sequence of outputs.

It is mostly used in Image Captioning

MANY TO ONE

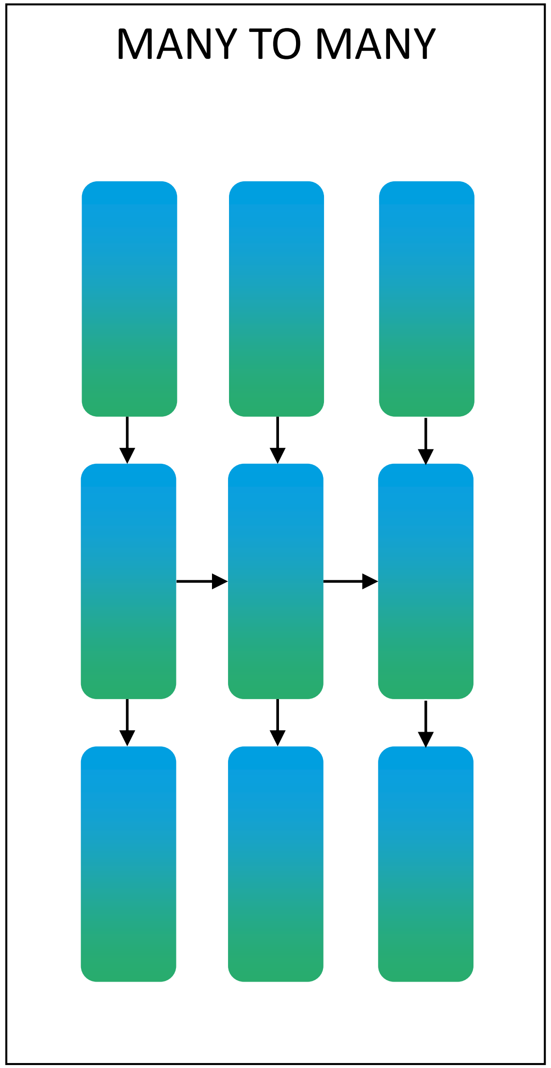


Multiple Inputs

Single Output

This type or RNN starts with a sequence of inputs.

It is very used in the analysis of sentiment.



Multiple Inputs

Many to Many type or RNN takes a sequence of inputs and generates another sequence of outputs. It is very used in Machine Translation

Multiple Outputs

Advantages

01 Remembers each and every information through time. It is useful in time series prediction only because of the feature to remember previous inputs as well.

02 Can be used with convolutional layers to extend the effective pixel neighborhood.

DRIVES
Development and Research on Innovative Vocational Education Skills

Disadvantages

01 Gradient vanishing and exploding problems.

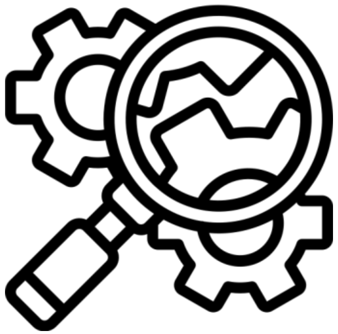
02 Training an RNN is a very difficult task.

03 It cannot process very long sequences if using tanh or relu as an activation function.

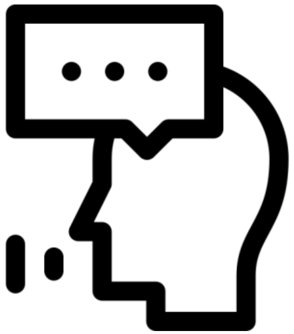
CNN VS RNN

CNN	RNN
It is suitable for spatial data such as images.	It is s suitable for temporal data(sequential data).
More powerfull than RNN	Includes less feature compatibility than CNN.
Takes fixed size inputs and generates fixed size outputs.	Can handle arbitrary input/output lengths.
Feed-forward artificial neural network with variations of multilayer perceptrons designed to use minimal amounts of preprocessing.	Unlike feed-forward neural networks can use their internal memory to process arbitrary sequences of inputs.
Uses connectivity pattern between the neurons.	Uses time-series information, what a user spoke last will impact what he/she will speak next.
Ideal for images and video processing.	Ideal for text and speech analysis.

Search Engines



Speech-to-text applications



eCommerce

Stock Price Forecasting

Ad Fraud, Spam Detection, Bot Detection

Market Research

TENSORFLOW

TENSOR

Multi-Dimensional Array

FLOW

Graph of Operations

- Introduced by Google
- Released on February 2017
- Allows interface and training entirely on browser
- Developed in C++
- Requires 2 hardware components : CPU (Central Processing Unit) and GPU (Graphics Processing Unit)

Tensors can be vizualized as:

Scalar	Vectors	Matrix	ND-tensors
10	$\begin{bmatrix} 18 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 18 & -4 \\ 2 & 7 \end{bmatrix}$	$\begin{bmatrix} [18\ 1] & [0.5\ 1] \\ [7\ 13] & [-4\ 14] \\ [0\ \ 0] & [2\ -0.7] \end{bmatrix}$
Number	Simple array	2D array	Multidimensional array, more than 2D

DRIVES
Development and Research on Innovative Vocational Education Skills

YOU WILL NEED A DATASET OF IMAGES

Suggestion: https://arxiv.org/abs/1708.07747 (Novel Image Dataset for ML Algorithms)

STEP 1: Import the require modules

STEP 2: Import the data

STEP 3: Analyse the data

STEP 4: Data Preprocessing

STEP 5: Define the Deep Learning Network

STEP 6: Compile the model

STEP 7: Fit the model

STEP 8: Evaluate the model

STEP 9: Make Predictions

DRIVES
Development and Research on Innovative Vocational Education Skills

STEP 1: Import the require modules

```
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.examples.tutorials.mnist import input_data
import os
```

STEP 2: Import the data

```
data = input_data.read_data_sets('directory',one_hot=True,source_url='')
```

one_hot=True -> converts the categorical class labels to binary vectors.

STEP 3: Analyse the data

Analyse what the images in the dataset look like: Do they need rescale? Are they in the require shape?

DRIVES
Development and Research on Innovative Vocational Education Skills

STEP 4: Data Preprocessing

1. Vizualize images to confirm that they are between 0 and 1

```
data.train.images[0][200:]
np.max(data.train.images[0])
np.min(data.train.images[0])
```

2. Reshape your data so that is Tensorflow expected input shape for its Deep Learning Model
   (<number of images>, <image x_dim>, <image y_dim>, <number of channels>)

```
train_X = data.train.images.reshape(-1, 28, 28, 1)
test_X = data.test.images.reshape(-1,28,28,1)
train_X.shape, test_X.shape
```

((55000, 28, 28, 1), (10000, 28, 28, 1))

3. Define the train and test set

```
train_y = data.train.labels
test_y = data.test.labels
```

STEP 5: Define the Deep Learning Network

1. Define Training Interations: **training_iters** (number of times you train the network), **learning_iters** (factor that is multiplied with the weights), batch_size (number of images that will go through the network each time. Should be a power of 2)

```
training_iters = 10
learning_rate = 0.001
batch_size = 128
```

2. Define Network parameters: number of inputs, number of classes (number of class labels)

```
n_input = 28
n_classes = 10
```

3. Define Placeholders

```
x = tf.placeholder("float", [None, 28,28,1])
y = tf.placeholder("float", [None, n_classes])
```

4. Create the Layers (It should always be defined the convolution and max-pooling functions)

- conv2d() function has 4 arguments: input x, weights W, bias b, and strides.
- max-pooling function has the input x and a kernel size k.

```
def conv2d(x, W, b, strides=1):
x = tf.nn.conv2d(x, W, strides=[1, strides, strides, 1], padding='SAME')
x = tf.nn.bias_add(x, b) return tf.nn.relu(x)
def maxpool2d(x, k=2):
return tf.nn.max_pool(x, ksize=[1, k, k, 1], strides=[1, k, k, 1],padding='SAME')
```

5. Define Weights and biases variables

```
weights = {
'wc1': tf.get_variable('W0', shape=(3,3,1,32),initializer=tf.contrib.layers.xavier_initializer()),
…,
 'out': tf.get_variable('W6', shape=(128,n_classes),initializer=tf.contrib.layers.xavier_initializer()), }
biases = {
'bc1': tf.get_variable('B0', shape=(32), initializer=tf.contrib.layers.xavier_initializer()),
…,
 'out': tf.get_variable('B4', shape=(10), initializer=tf.contrib.layers.xavier_initializer()), }
```

YOU WILL NEED A DATASET

STEP 1: Import the require modules

STEP 2: Import/Generate the data

STEP 3: Define the placeholder for the data

STEP 4: Define the Recurrent Network

STEP 5: Compile the model

STEP 6: Calculate the loss

STEP 7: Vizualize the training

STEP 8: Training and Testing

DRIVES
Development and Research on Innovative Vocational Education Skills

- Deep learning recognizes an object in an image using CNN
- A convolutional neural network (CNN) as many equivalent definitions. But the most common one is that CNN is a feed forward type of artificial neural network used in image recognition and processing
- In CNN, every image is represented in the form of arrays of pixel values
- There are 4 layer in a CNN: Convolution, ReLu, Pooling and Fully Connected
- CNN can be used in: Face Recognition, Document Analysis, Marketing, Autonomous Cars, among others
- RNN is a type of neural network that contains loops so that information can be stored in the network
- There are 4 types of RNN: one to one, one to many, many to one, many to many
- RNN can be used in: Search Engines, e-Commerce, Stock Price Forecasting, among others
- Tensorflow is the most used tool that allows interface and training entirely on browser in order to implement a CNN or RNN

REFERENCES

- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (pp. 1-6). Ieee.
- O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.
- Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. *arXiv preprint arXiv:1702.01923*.
- Chua, L. O., & Roska, T. (1993). The CNN paradigm. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 40(3), 147-156.

- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 International Conference on Engineering and Technology (ICET) (pp. 1-6). Ieee.

- Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into imaging, 9(4), 611-629.
- Zhu, Y., Zhao, J. K., Wang, Y. N., & Zheng, B. B. (2016). A review of human action recognition based on deep learning. Acta automatica sinica, 42(6), 848-857.
- Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 132306.

DRIVES
Development and Research on Innovative Vocational Education Skills

- Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In Eleventh annual conference of the international speech communication association.
- Dyer, C., Kuncoro, A., Ballesteros, M., & Smith, N. A. (2016). Recurrent neural network grammars. arXiv preprint arXiv:1602.07776.
- Medsker, L. R., & Jain, L. C. (2001). Recurrent neural networks. Design and Applications, 5.
- KP, S. (2019). RNNSecureNet: Recurrent neural networks for Cyber security use-cases. arXiv preprint arXiv:1901.04281.
- Rodriguez, P., Wiles, J., & Elman, J. L. (1999). A recurrent neural network that learns to count. Connection Science, 11(1), 5-40.
- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16) (pp. 265-283).
- Goldsborough, P. (2016). A tour of tensorflow. arXiv preprint arXiv:1610.01178.
- Zaccone, G. (2016). Getting started with TensorFlow. Packt Publishing Ltd.
- Shukla, N., & Fricklas, K. (2018). Machine learning with TensorFlow. Greenwich: Manning.

Regina Sousa

- PhD student in Biomedical Engineering
- Research Collaborator of the Algoritmi Research Center

0000-0002-2988-196X

Diana Ferreira

- PhD student in Biomedical Engineering
- Research Collaborator of the Algoritmi Research Center

0000-0003-2326-2153

José Machado

- Associate Professor with Habilitation at the University of Minho
- Integrated Researcher of the Algoritmi Research Center

0000-0003-4121-6169

REFERENCE TO AUTHORS

António Abelha

- Assistant Professor at the University of Minho
- Integrated Researcher of the Algoritmi Research Center

0000-0001-6457-0756

Victor Alves

- Assistant Professor at the University of Minho
- Integrated Researcher of the Algoritmi Research Center

0000-0003-1819-7051

REFERENCE TO AUTHORS

This Training Material has been certified according to the rules of **ECQA – European Certification and Qualification Association.**

The Training Material was developed within the international job role committee **“Artificial Intelligence Technician”:**

UMINHO – University of Minho (https://www.uminho.pt/PT)

The development of the training material was partly funded by the EU under Blueprint Project DRIVES.

Thank you for your attention

DRIVES project is project under **The Blueprint for Sectoral Cooperation on Skills in Automotive Sector**, as part of New Skills Agenda.

The aim of the Blueprint is **to support an overall sectoral strategy and to develop concrete actions to address short and medium term skills needs.**

Follow DRIVES project at:

More information at:

www.project-drives.eu

Co-funded by the Erasmus+ Programme of the European Union

The Development and Research on Innovative Vocational Educational Skills project (DRIVES) is co-funded by the Erasmus+ Programme of the European Union under the agreement 591988-EPP-1-2017-1-CZ-EPPKA2-SSA-B. The European Commission support for the production of this publication does not constitute endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.