Deep learning for Image Classification

Álvaro González Jiménez

Trabajo fin de Grado

Supervisado por Dr. Isabel A. Nepomuceno Chamorro

Universidad de Sevilla

January 2019

Publicado en January 2019 por
Álvaro González Jiménez

https://github.com/alvarogonjim
alvgonjim1@alum.us.es

Yo, D. Álvaro González Jiménez con NIF número 53585603L,

DECLARO

mi autoría del trabajo que se presenta en la memoria de este trabajo fin de grado que tiene por título:

Deep learning for Image Classification

Lo cual firmo,

Fdo. D. Álvaro González Jiménez
en la Universidad de Sevilla
11/01/2019

Tu dedicatoria aquí

Agradecimientos

No olvides añadir una nota de agradecimiento a quienes hayan contribuido emocionalmente al proyecto fin de Grado.

Contents

List of Figures

List of Tables

Parte I

Introducción

Introduction

In this chapter, we will explore a little bit about the Machine Learning world, the algortihms that people have been used for years to solve this and other types of problems. A brief summary about the current frameworks that we can use to solve this problem, theirs pros and cons. Finally we are going to explain the problems that we are going to solve.

1.1 Artificial Inteligence History

Deep Learning (DL) has revolutionized industry after industry. People thinks that Deep Learning and Artificial Inteligence (AI) are synonyms, but these words are totally different. We can define the AI as *the automation of intellectual tasks normally performed by humans* The AI started in the beginning of the 1957 when John McCarthy held the first academic conference on the subject, the AI machines were able to solve problems that were difficult for humans to solve.

One of the most important AI machines in the history is the Enigma machine built by Alan Turin at the end of World War II. This machine could crack in hours entire conversations and text made by the Nazis.

In the early years of AI, a lot of researchers believed that AI could be achieved by hard coding rules. The kind of AI is called symbolic AI and was useful in solving well-defined, logical problems but it was incapable of solving complex problems such as image recognition, object detection, object segmentation, language translation, and natural-languag-understanding.

In order to understand the relationship among AI, ML and DL let's visualize them as concentric circles.

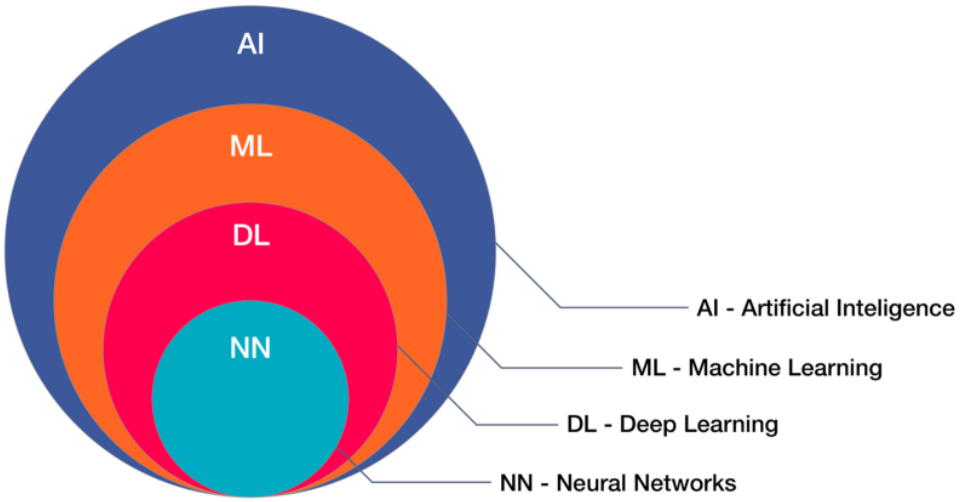

Figure 1.1: Representation AI, ML and DL [1]

The idea that came first (AI) and the largest one. Then machine learning (ML) which blossomed later), and finally DL which is driving today's AI explosion (fitting inside both).

1.1.1 Machine Learning

Machine Learning (ML) is a sub-field of AI and has become very popular in the last 10 years. AI has a lot of other sub-fields aside from Machine Learning, ML systems are built by showing lots of examples, unlike symbolic AI where we hard code rules to build the system.

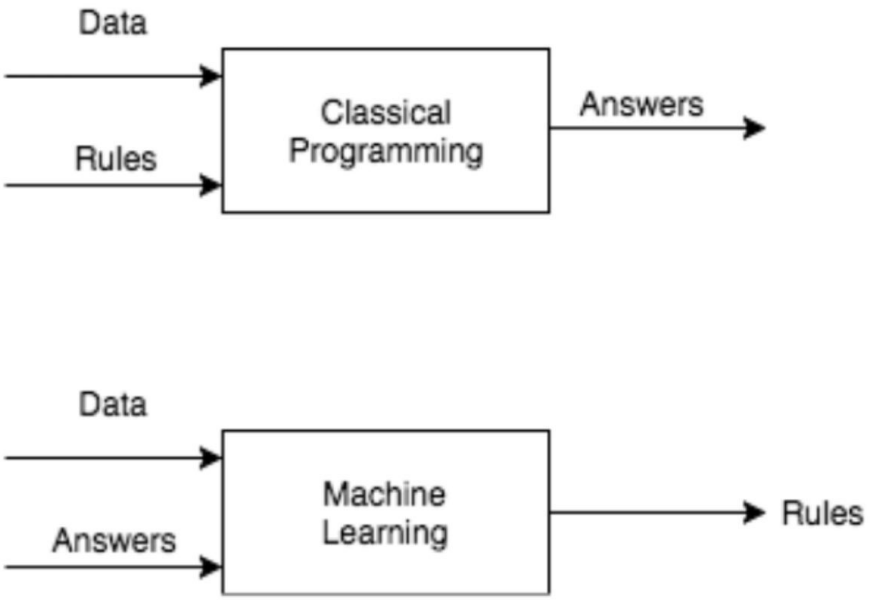

Figure 1.2: ML vs Traditional programming

At a high level, machine learning systems took at tons of data and come up with rules to predict outcomes for unseen data. Some examples of machine learning systems in the real life are:

- Google Photos uses a specific form of machine learning for grouping photos.
- Recommendations systems which are family of ML algorithms. We can see this kind of recommendations systems in many famous apps like Spotify (music), Netflix (movies) and Amazon (products).

1.1.2 Deep Learning

Deep Learning is a subfield of ML, that uses specific algorithms in order to predict for example whether an image contains a face or not. It extracts features suach as the first layer detecting edges, the second layer detecting shapes such as noses and eyes and the final layer detecting face shapes or more complex structures.

The use of DL has grown tremendously in the last few years with the rise of GPUs, big data, cloud providers (Amazon Web Services) and frameworks such as Torch, Tensorflow or Pytorch.

1.2 State of the art

1.2.1 Neuronal network

A neuronal network is a mathematical model that try to represents how our mind works. The first mathematical model was presented in 1943 called McCulloch-Pitts this model could do several simple tasks. [2]

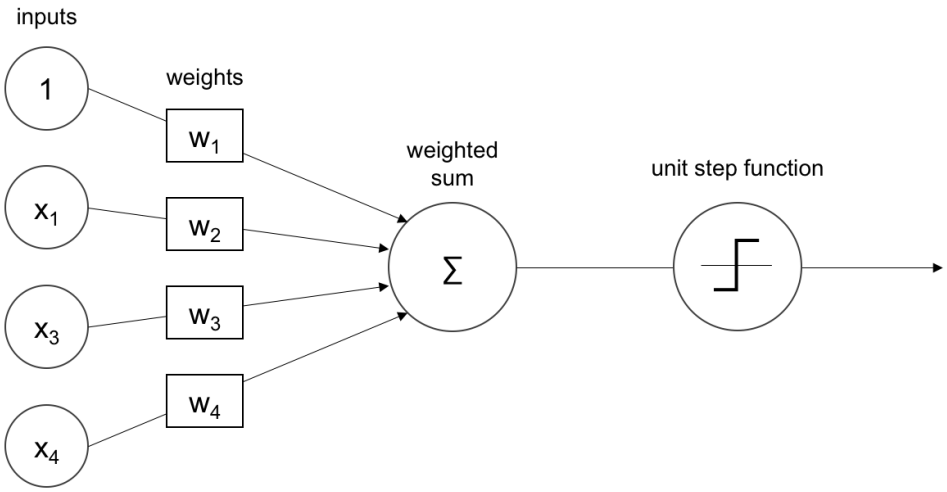

Figure 1.3: Example of a perceptron [4]

This picture represents a perceptron wich has a set of inputs that are going to be multiplied by their weights. Then the perceptron will permform a weighted summmation to produce an output. [3] Finally the output of the perceptron can be passed through an activation function or transfer function.[4] The process that the weight of the inputs changes in order to improve the output of the perceptron is called trainning.

An Artificial Neuronal Netrowk (ANN) is a collection of perceptrons and activation functions. The perceptrons are connected to form hidden layers or units. The hidden units form the nonlinear basis that maps the input layers to output layers in a lower-dimensional space, which is also called artificial neural networks. ANN is a map from input to output. The map is computed by weighted addition of the inputs with biases. The values of weight and bias values along with the architecture are called model.

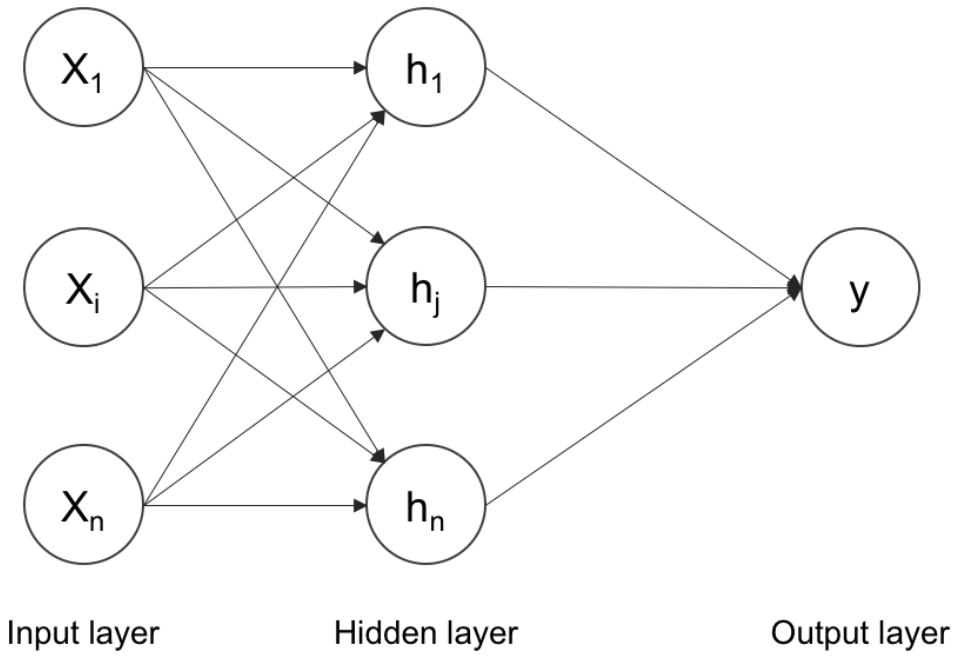

Figure 1.4: Example of an Artificial Neuronal Network (ANN) [4]

The ANN contains several parameters to optimize. The procedure of updating the weights is called backpropagation. The weights are updated from backward based on the error calculated. The procedure to minimize the error is called optimization.

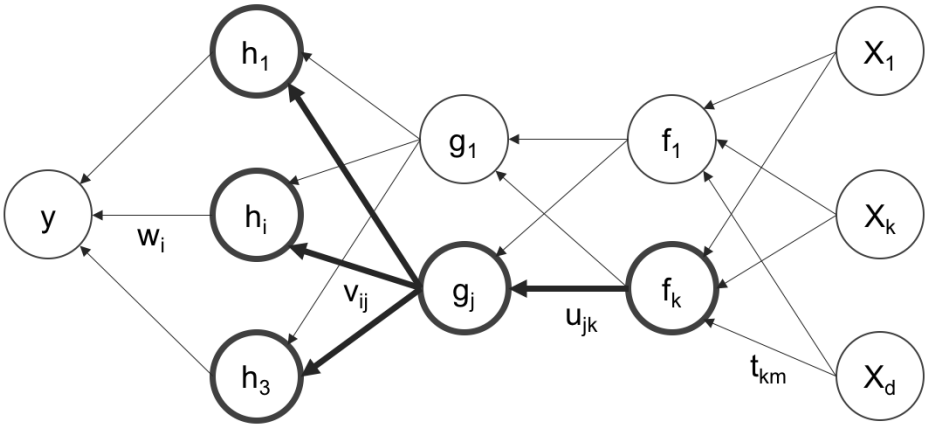

Figure 1.5: Example of backpropagation procedure [4]

A few different ANN models are availables among them the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). They have made some revolutionary improvements in the data analysis field.

1.2.2 Unsupervised Pretrained Networks

Autoencoders

Autoencoder is an architecture for unsupervised learning. The main porpouse ot this architecture is to reduce the dimensionality of the dataset and it learns directly from the input data. The structure is very simple, the input layer followed by the bottleneck where the encoder and decoder operation are done and finally the output layer that contains the same number of units as the input layer does.[20]

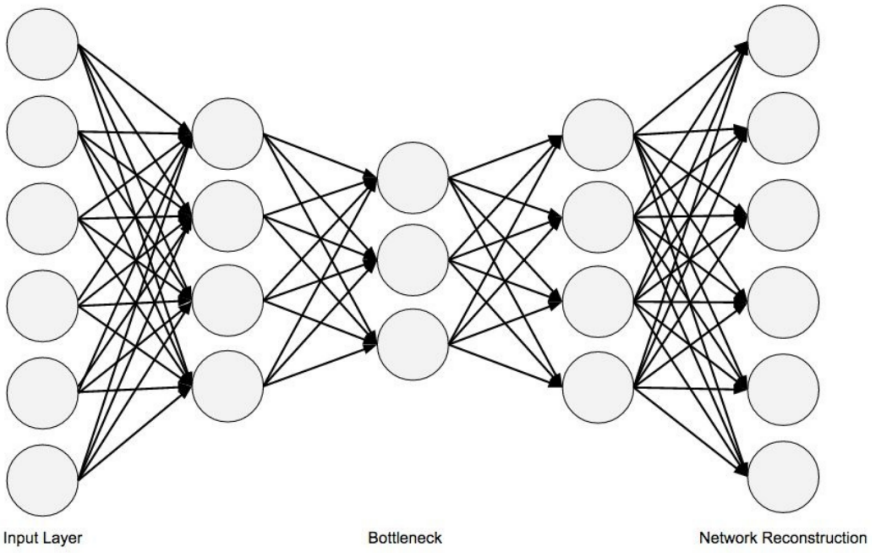

Figure 1.6: Example of Autoencoder [20]

Deep Belief Networks

Deep Belief Networks (DBNs) are composed of layer of Restricted Boltzmann Machines (RBMs). In unsupervised learning we use RBMs to extract high-level features from the input data. We discovered that if we let to RBMs to extract this high-level features it progressively can combine nonlinear functions to extract more complex data, to understand better how it works we can see an example with MNIST digits.[20]

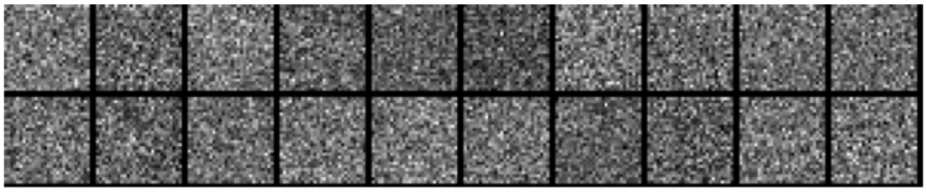

Figure 1.7: Activation render at the beginning of training [20]

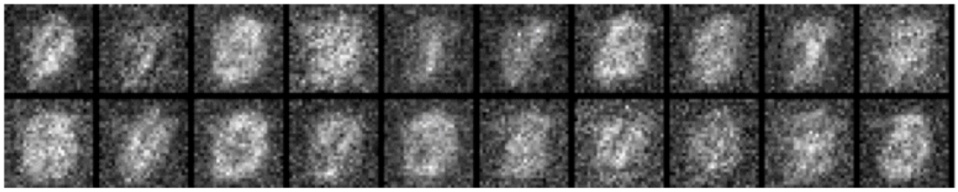

Figure 1.8: Features emerge in later activation render [20]

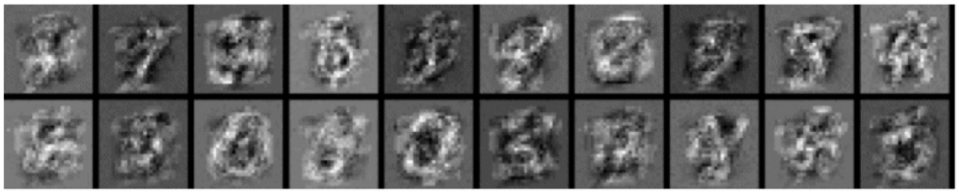

Figure 1.9: Portions of MNIST digits emerge towards end of training [20]

Generative Adversarial Networks

The Generative Adversarial Networks is an example of unsupervised learning architecture where we train two models in parallel. Nowdays GANs architecture become really famous to create new images based on other images[20].

Figure 1.10: Example of a GAN with The Mona Lisa painting [21]

GAN is composed by two players, one player (the generator) generates samples that are similar to the trainning data without having access to that data and the second player (the discriminator) examine that samples and determine that data is real or fake.[21]

1.2.3 Reinforcement Learning

Reinforcement Learning is a subfield of machine learning which addresses the problem of automatic learning of optimal decisions over time. This is a general and common problem studied in many scientific and engineering fields.[8]

Reinforcement Learning (RL) is the third camp and lays somewhere in between full supervision and a complete lack of predefined labels. On the one hand, it uses many well-established methods of supervised learning such as deep neural networks for function approximation, stochastic gradient descent, and backpropagation, to learn data representation. On the other hand, it usually applies them in a different way. We need an agent that takes actions in some environment.

A good example of this is a robot mouse in a maze, Its environment is a maze with food at some points and electricity at others. The robot mouse can take actions such as turn left/right and move forward. Finally, at every moment it can observe the full state of the maze to make a decision about the actions it may take.

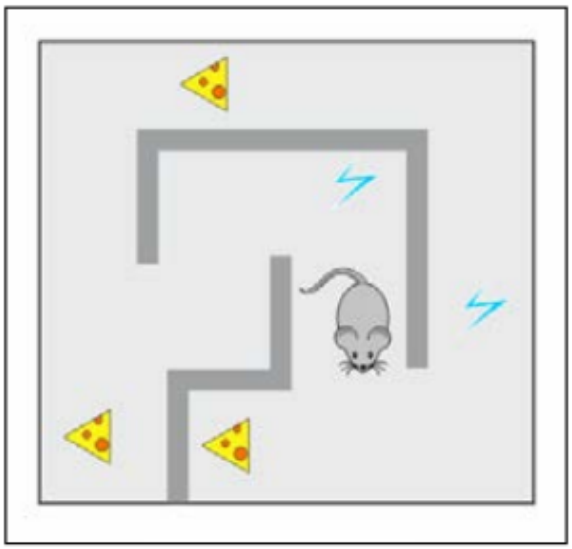

Figure 1.11: Example of the Robot Mouse scenario [8]

It is trying to find as much food as possible, while avoiding an electric shock whenever possible.These food and electricity signals stand as a reward given to the agent by the environment as additional feedback about the agent's actions.

1.2.4 Convolutional Neuronal Network (CNN)

If we want to use a ANN for images the size of the network will be very large because of the number of neurons and will results an overfitting. Moreover in the case that we want to use big resolution images the size of the network will be huge [4]

The convolutional neuronal network (CNN) is an advance ANN, which allows

the network to extract local as well as global features from the data, enhancing the decision-making procedure of the network[5]. Furthermore a CNN that typically has convolutional layers interspersed with pooling (or sub-sampling) layers and then followed by fully connected layers as in a standard multi-layer neural network.[6]

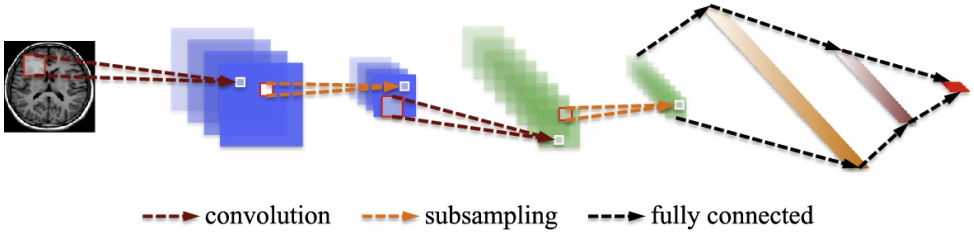

Figure 1.12: An architecture of a convolutional neural network

Convolution

A convolution is defined as a mathematical operation describing a rule for how to merge two sets of information. [7]

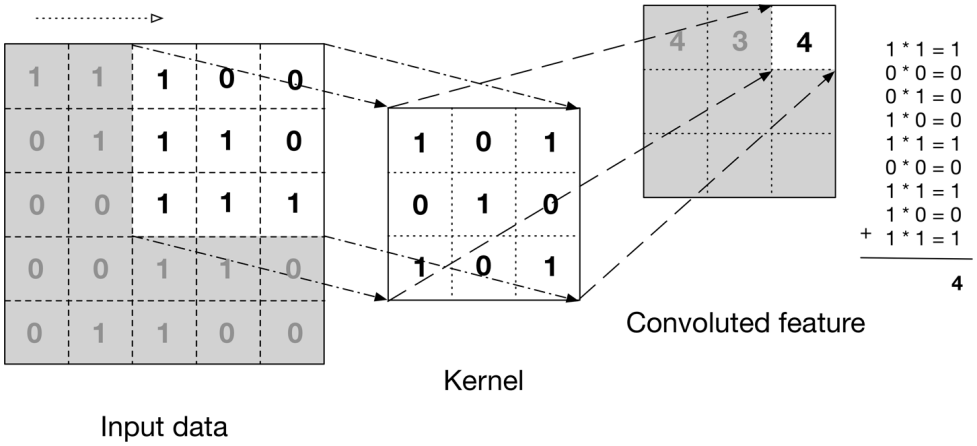

Figure 1.13: The convolution operation [4]

The figure ilustrate how the kernel is slid across the input data to produce the convoluted feature (output) data.

Pooling

Usually after the convolution layer we will include a pooling layer to reduce the spatial size [7] (height and width) for the next layer. The most common pooling operation is max-pooling that reduces the size by a factor of n^2 [6] but exists other pooling

operations like average-pooling, winner-takes-all pooling [9] and stochastic pooling [10].

Fully Connected Layers

We use this layer to compute class scores that we'll use as output of the network. The dimension of the final output will be [1 x 1 x N] where N is the number of classes [7]. For example in our case the number of classes will be 2 (carcinogenic or not).

1.3 SOFTWARE

There are many frameworks to resolve this problem. We are going to explain some for Python and compare them in order to choose the best framework to resolve our problem.

1.3.1 Tensorflow

Tensorflow was created by the IA Google Team. They followed the same structure that Theano library, the engine was written in C/C++ to increase the speed [12] It support the CPU and GPU operations and you can run with multiple GPU to optimize your model [14]. Furthermore this library include a visualization of your model called Tensorboard so you are able to see how is trainning your model for debugging and optimization. Tensorflow is the most famous library for Machine Learning (ML) works but, it's hard to learn and understand how it works in the beginning if we don't have any other experience in ML.

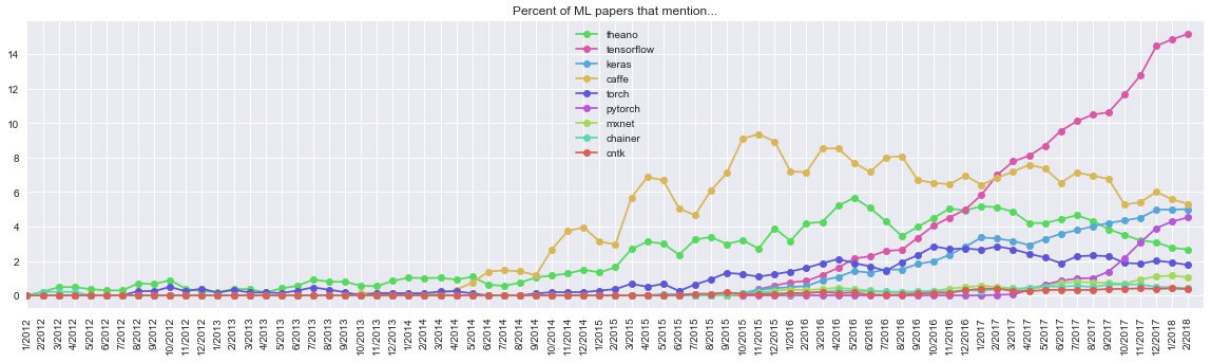

Figure 1.14: Percent of mentions in ML papers [11]

1.3.2 Theano

Maintained by Montréal University group [13] they were the pioneered to use a computational graph that will be use in Tensorflow project [12]. The big cons is for large models Theano will take long time to compute it, also it doesn't support multiple GPU. The error messages can be unhelpful so it will be hard for debugging tasks.

1.3.3 Keras

Keras is an easy-to-use Python library [13] that sits atop Tensorflow and Theano so it took the advantages of both. It has an intuitive and simple API so you can write a model with a few lines of code. Keras is not really flexible and has some problems to use the multi-gpu.

1.3.4 Pytorch

The Python version of Torch called Pytorch is an open source project of Facebook created in January 2017. PyTorch has quickly become the favorite among machine-learning researchers, because it allows certain complex architectures to be built easily. [12]. Furthermore the modularity of Pytorch because it's easy to pull someone's code and use luarocks to install required packages [13].

1.3.5 H2O

It's an open source software platform which core is coded in Java [18] H20 is feature-rich, ease of use and it's known for it's R and Spark integration but it provides support to other languages. [19] H20 counts with other plugins to create an easy app after train your model (Shinny App).

1.3.6 DL4J

DL4J is a JVM-based, industry-focused, commercially supported, distributed deep-learning framework [12]. Python has many scientific environments to work with Deep Learning like Numpy or Theano but DL4J with Java and Scala has several advantages, Java and Scala are inherently faster than Python. Anything written in Python by itself, disregarding its reliance on Cython, will be slower. Furthermore the license under DL4J is Apache 2.0 License so anyone is free to make and patents derivate works based

on Apache 2.0-licensed code.

1.4 Evaluate the model

There are many metrics and ways to evaluate the model and check the accuracy. Depends of the scenario we have to use a specific metric or other, there is not exist a metric for all the possibles scenarios. The objective of this section is to explain most of the famous metrics for a classification problem.

1.4.1 Confusion Matrix

This metric is the simplest because it's very easy to use and understand. Each column of the matrix represents the number of predictions for each class, while each row represents the instances in the actual class. One of the benefits of confusion matrices is that they make it easier to see if the system is confusing two classes.

The associated values of a confusion matrix are:

1. **True Positives (TP):** test result is one that detects the condition when the condition is present. Sick people correctly identified as sick.

2. **True Negative (TN):** test result is one that does not detect the condition when the condition is absent. Healthy people correctly identified as healthy

3. **False Positive (FP):** test result is one that detects the condition when the condition is absent. Healthy people incorrectly identified as sick

4. **False Negative (FP):** test result is one that does not detect the condition when the condition is present. Sick people incorrectly identified as healthy

Let's see an example of the confusion matrix to understand how it works. The scenario is a dog vs cat classification and the results are the following:

		Actual Class	
		Cat	Dog
Predicted Class	Cat	5	2
	Dog	3	3

Sensitive

Measures the ability of a test to detect the condition when the condition is present.

$$Sensitive = TP/(TP + FN)$$

In our cat classificator the sensitive is:

$$Sensitive = 5/(5 + 3) = 0.625$$

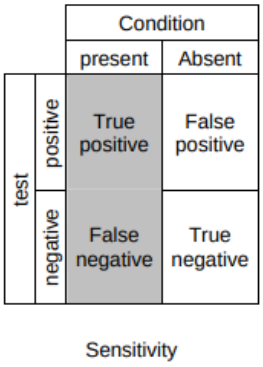

Figure 1.15: Calculate the Sensitive

Specificity

Measures the ability of a test to correctly exclude the condition (not detect the condition) when the condition is absent.

$$Specificity = TN/(TN + FP)$$

In our cat classificator the specificity is:

$$Specificity = 3/(3 + 2) = 0.6$$

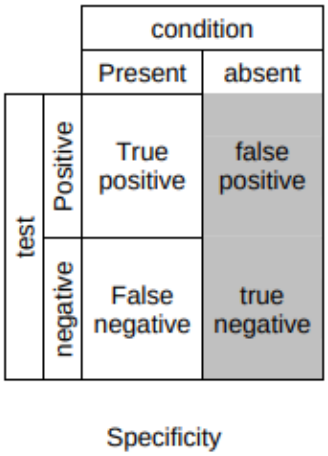

Figure 1.16: Calculate the Specificity

Predictive value positive

Measures the proportion of positives that correspond to the presence of the condition.

$$PVP = TP/(TP + FP)$$

In our cat classificator the predictive value positive is:

$$PVP = 5/(5 + 2) = 0.714$$

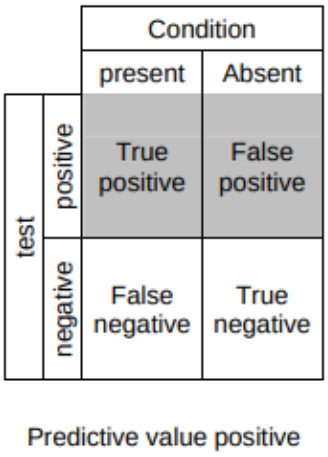

Figure 1.17: Calculate the Predictive Value Positive

Predictive value negative

Measures is the proportion of negatives that correspond to the absence of the condition.

$$PVN = TN/(TN + FN)$$

In our cat classificator the predictive value negative is:

$$PVN = 3/(3 + 3) = 0.5$$

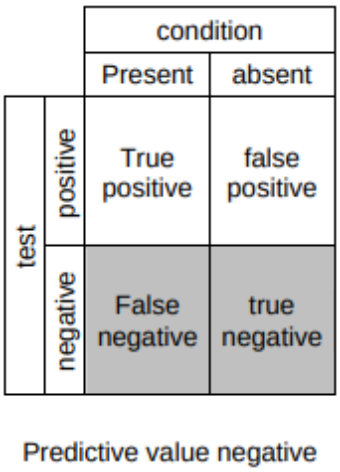

Figure 1.18: Calculate the Predictive Value Negative

1.4.2 The Area Under an ROC Curve

The area under the ROC curve (AUC) is a measure of how well a parameter can distinguish between two diagnostic groups (diseased/normal).

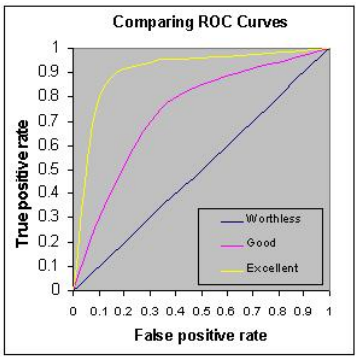

Figure 1.19: Examples of differents ROC curves [16]

It's a famous metric in the health scenario, the accuracy is measured by the area under the ROC curve. An area of 1 represents a perfect test; an area of .5 represents a worthless test [16]. A rough guide for classifying the accuracy of a diagnostic test is the traditional academic point system:

1. **.90 - 1** Excellent (A)
2. **.80 - .90** Good (B)
3. **.70 - .80** Fair (C)
4. **.60 - .70** Poor (D)
5. **.50 - .60** Fail (F)

1.5 ARCHITECTURES

Neural network can be compared with lego blocks, where you can build almost any simplex to complex structures. Actually, most of the famous architectures have been discovered during the Imagenet challenge, this challenge evaluates algorithms for object detection and image classification at large scale[17].

Depends of the task that we want to solve we have to use a type of architecture or other. The main types of tasks that computer vision can be categorised in are as follows:

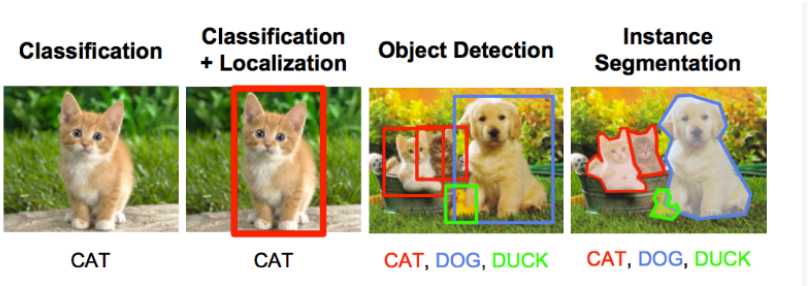

Figure 1.20: Examples of the main types of task in computer vision [17]

1. **Object Recognition / Classification.** In object recognition, the goal is to identify the class givin a raw image.
2. **Classification + Localisation.** - In this one we have to identify the class and find the location of that object in the raw image.

3. **Object Detection.** The task is to identify where in the image does the objects lies in.

4. **Image Segmentation.** It's a bit sophisticated task, where the objective is to map each pixel to its rightful class.

1.5.1 AlexNet

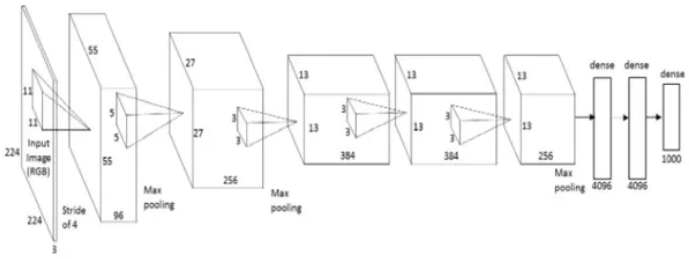

Figure 1.21: AlexNet Architecture

It's the first deep architecture created by Geoffrey Hinton and his colleagues. When broken down, AlexNet seems like a simple architecture with convolutional and pooling layers one on top of the other, followed by fully connected layers at the top. This is a very simple architecture, which was conceptualised way back in 1980s.

1.5.2 VGG Net

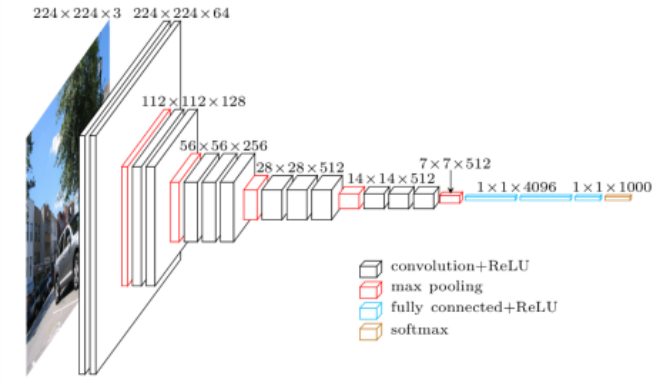

Figure 1.22: VGG Net Architecture

This architecture was introduced by the researchers at Visual Graphics Group at Oxford (hence the name VGG). The architecture is characterized by it's pyramidal

shape, where the bottom layers which are closer to the image are wide, whereas the top layers are deep.

1.5.3 ResNet

Residual Networks (ResNet in short) consists of multiple subsequent residual modules, which are the basic building block of ResNet architecture. A representation of residual module is as follows.

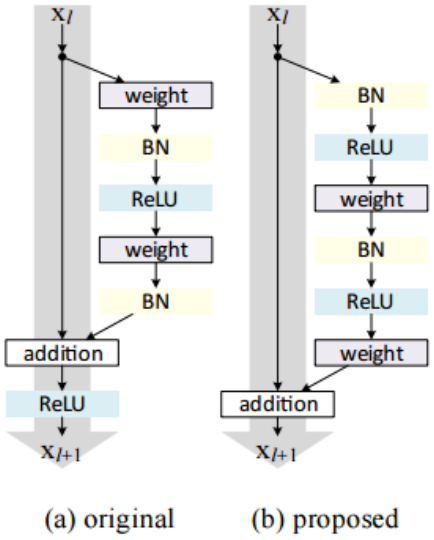

Figure 1.23: ResNet Architecture

The main advantage of ResNet is that hundreds, even thousands of these residual layers can be used to create a network and then trained. This is a bit different from usual sequential networks, where you see that there is reduced performance upgrades as you increase the number of layers.

Context

In this chapter we are going to explain the motivation, the main problem that we are going to solve. The development of the solution to the problem will be detailed, the results, plannification and costs for this solution.

2.1 Motivation

During my last year of degree I was wondering what I was going to work on my thesis. I knew I wanted to work on something related to health because I had always been interested in creating a product that would have an impact on society.

In my internship I was in one of the most important research centres in France in the field of artificial intelligence. There I realized the potential of this technology that was emerging during these years and that I could do something related to health and that I had not worked previously during my studies.

The results expected at the end of this work will not have the same quality and precision work and studies recintes in the market (accuracy of 80%-90%) because we do not have the same resources as them. The main objective is to achieve a scalable prototype whose precision and quality will increase in the future.

2.2 Problem

The problem that we are going to solve is a binary classification, we'll receive an image and we will detect if the image has skin cancer or not. The objetives of this project is create a model based on deep learning algorithms to solve this problem, use different techniques to improve the accuracy and compare the obtained results with other algorithms or other works (papers, studies, etc). At the end we are going to expose the model in a REST API so we can use in the future for other projects, for example:

1. Create an app to take photos of skin moles and detect if there are skin cancer or not on it.
2. Monitor and collect more data for future studies (areas, age, gender affected by skin cancer...)

2.3 ML Methodology

This project is a machine learning project that is highly iteractive. It has some similarities as a software project.

ML project structure	
Data collection and labeling	Create labeling and validate quality of the data
Model exploration	Start with a simple model and improve it
Testing and evaluation	Revisit model evaluation metric
Model deployment	Expose model as REST API,
Ongoing model maintenance	Retrain model with new data to prevent model staleness

Table 2.1: Machine Learning project structure

2.4 Data collection and labeling

During this period we have to determine the feasibility of the project, in our case it's not a hard task to acquire the data, the ISIC institute exposed in 2018 more than 20.000 skin lesiono images during a challenge. https://challenge2018.isic-archive.com/ Furthermore exist other famous datasets like *PH2Dataset* from the Oporto Universiy which the number of images is much lower but we can use some Data Augmentation techniques to improve the results.

2.4.1 ISIC Dataset

Parte II

Appendices

Bibliography

[1] Pippa Biddle,Diamond Inc *https://bit.ly/2rkhRUY* , June 2017 (pages v, 4).

[2] Fernando Sancho Caparrini *http://www.cs.us.es/ fsancho/?e=72* 23 of April 2017. (page 6).

[3] Sagar Sharma *https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53* 9 of September 2017. (page 6).

[4] Rajalingappaa Shanmugamani *Deep learning for computer vision: expert techniques to train advanced neural networks using TensorFlow and Keras* 2018. (pages v, 6, 7, 8, 11, 12).

[5] Abdullah-Al Nahid, Mohamad Ali Mehrabi, and Yinan Kong*Histopathological Breast Cancer Image Classification by Deep Neural Network Techniques Guided by Local Clustering* 7 March 2018.School of Engineering, Macquarie University, Sydney, Australia. (page 12).

[6] S. Kevin Zhou, Hayit Greenspan, Dinggang Shen *Deep learning for medical image analysis* 2017. (page 12).

[7] Josh Patterson and Adam Gibson *Getting started with deep learning* 2018. (pages 12, 13).

[8] Maxim Lapan *Deep Reinforcement Learning Hands-On/*, June 2018 (pages v, 10, 11).

[9] R.K. Srivastava, J. Masci, S. Kazerounian, F. Gomez, J. Schmidhuber *Advances in Neural Information Processing Systems.* 2013. (page 13).

[10] M.D. Zeiler, R. Fergus*Stochastic pooling for regularization of deep convolutional neural networks.* 2013. (page 13).

[11] Andrej Karpathy *https://twitter.com/karpathy/status/972295865187512320?lang=es* March 2018 (pages v, 13).

[12] *https://skymind.ai/wiki/comparison -frameworks-dl4j-tensorflow-pytorch* (pages 13, 14).

[13] Vicky Kalogeiton Stéphane Lathuilière, Pauline Luc Thomas, Lucas Konstantin Shmelkov *https://project.inria.fr/deeplearning/files/2016/05/DLFrameworks.pdf* 25 of January 2017 (page 14).

[14] Tensorflow API *https://www.tensorflow.org* 25 of January 2017 (page 13).

[15] Andrej Karpathy *https://twitter.com/karpathy/status/972295865187512320*

[16] The Area Under an ROC Curve definition *http://gim.unmc.edu/dxtests/roc3.htm* (pages v, 18, 19).

[17] Advanced Deep Learning Architectures Data *https://www.analyticsvidhya.com/blog/2017/08/10-advanced-deep-learning-architectures-data-scientists/* (pages v, 19).

[18] Srijan Agarwal *https://opensourceforu.com/2017/01/introduction-h2o-relation-deep-learning/*, 21 of January 2018 (page 14).

[19] Oksana Kutkina, Stefan Feuerriegel *http://www.rblog.uni-freiburg.de/2017/02/07/deep-learning-in-r/*, 7 of March 2016 (page 14).

[20] Josh Patterson, Adam Gibson *Deep Learning: A practitioner's approach/*, August 2017 (pages v, 8, 9, 10).

[21] Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, Marian Mazzone *CAN: Creative Adversarial Networks Generating "Art" by Learning About Styles and Deviating from Style Norms/*, August 2017 (pages v, 10).