Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

**Μεταπτυχιακή Διατριβή**

| | |
|---|---|
| Τίτλος Διατριβής | **Μια Ανασκόπηση της Βαθιάς Μάθησης:**<br>**Θεωρία, Μέθοδοι και Εφαρμογές**<br><br>**A Review of Deep Learning:**<br>**Theory, Methods & Applications** |
| Ονοματεπώνυμο Φοιτητή | **Ιωάννης Κοντούλης** |
| Πατρώνυμο | **Βασίλειος** |
| Αριθμός Μητρώου | **ΜΠΠΛ/11046** |
| Επιβλέπων | **Γεώργιος Τσιχριντζής, Καθηγητής** |

Ημερομηνία Παράδοσης **Οκτώβριος 2017**

## Τριμελής Εξεταστική Επιτροπή

| | | |
|---|---|---|
| (υπογραφή) | (υπογραφή) | (υπογραφή) |
| Όνομα Επώνυμο<br>Βαθμίδα | Όνομα Επώνυμο<br>Βαθμίδα | Όνομα Επώνυμο<br>Βαθμίδα |

## ABSTRACT

The presented thesis, entitled “A Review of Deep Learning: Theory, Methods & Applications”, is an extensive literature review of deep learning, with regard to the origins and the interdisciplinary nature of the field. Deep learning is a revolutionary machine learning approach, that constitutes a class of machine learning techniques, where many layers of information processing stages in hierarchical supervised architectures are exploited for unsupervised feature learning and for pattern analysis or classification. The origins and the motivations of deep learning are mainly found in artificial neural networks, as well as in other related scientific fields as artificial intelligence, cognitive neuroscience and signal processing among others. In recent years, systems based on deep learning techniques and algorithms have become extremely popular both in academia and in many industry sectors, due to the state of the art performance on numerous machine learning problems. This thesis investigates how the fundamental “deep” ideas have been defined and how research interests have shifted over the years. In this perspective the basic building blocks for building deep learning architectures are presented and analyzed.



## ΠΕΡΙΛΗΨΗ

Η παρούσα μεταπτυχιακή εργασία, η οποία τιτλοφορείται "Μια Ανασκόπηση της Βαθιάς Μάθησης: Θεωρία, Μέθοδοι και Εφαρμογές", αποτελεί μια εκτενή βιβλιογραφική ανασκόπηση αναφορικά με την προέλευση και τη διεπιστημονική φύση του εν λόγω πεδίου. Η βαθιά μάθηση αποτελεί μια επαναστατική προσέγγιση μηχανικής μάθησης, που συνιστά μια ειδική κατηγορία τεχνικών μηχανικής μάθησης, κατά την οποία πολλά επίπεδα επεξεργασίας πληροφοριών σε συστήματα ιεραρχικά

εποπτευόμενων τεχνικών αξιοποιούνται για τη μη εποπτευόμενη εκμάθηση χαρακτηριστικών, καθώς για ανάλυση προτύπων ή κατηγοριοποίηση. Οι καταβολές και τα κίνητρα της βαθιάς μάθησης εντοπίζονται κυρίως στα τεχνητά νευρωνικά δίκτυα, καθώς και σε άλλους συναφείς επιστημονικούς τομείς, όπως η τεχνητή νοημοσύνη, η γνωσιακή νευροεπιστήμη και η επεξεργασίας σήματος μεταξύ άλλων. Τα τελευταία χρόνια, τα συστήματα που βασίζονται σε τεχνικές και αλγόριθμους βαθιάς μάθησης έχουν γίνει ιδιαιτέρως δημοφιλή τόσο στην ακαδημαϊκή κοινότητα όσο και σε πολλούς κλάδους της βιομηχανίας, λόγω των εξαιρετικών επιδόσεων των μεθόδων αυτών σε πληθώρα προβλημάτων μηχανικής μάθησης. Η παρούσα εργασία διερευνά πως έχουν καθοριστεί οι “βαθιές” θεμελιώδεις ιδέες και πως έχει μετατοπιστεί το ερευνητικό ενδιαφέρον κατά τη διάρκεια του χρόνου. Σε αυτό το πλαίσιο παρουσιάζονται και αναλύονται τα βασικά δομικά στοιχεία για την οικοδόμηση αρχιτεκτονικών βαθιάς μάθησης.

For the completion of the current thesis, I would like to express my sincere gratitude to my advisor Professor, George Tsihrintzis. I would also like to thank Dr. Dionisios Sotiropoulos for his cooperation, advice and his valuable contribution to the successful completion of this study.

# Contents

# Machine Learning

## Introduction

Learning is a fundamental cognitive process, which can modify behavior as a result of experience, forming the conditions for long-term optimization through change. This change reflects gradual modifications in behavior, according to a given criterion, when a similar situation occurs (Lampropoulos & Tsihrintzis, 2015). In that context, people modify their behavior in such a way that makes them perform better in the future (Witten & Frank, 2005).

Machine learning is the field of computer science where the objective is to develop algorithms, computer applications, and systems that have the ability to learn, and accordingly to improve their performance (Lampropoulos & Tsihrintzis, 2015). This is how to get computers to program themselves based on experience plus some initial structure-knowledge (Mitchell, 2006).

## The Learning Process: Basic Principles

An explicit definition of learning process must meet the following conditions (Camastra & Vinciarelli, 2007):

- Acquisition of new declarative knowledge.
- Development of motor and cognitive skills through instruction or practice.
- Organization of new knowledge into general effective representations.
- Discovery of new facts and theories through observation and experimentation.

Most of the proposed learning definitions have as common the acquisition of knowledge or skills by study, instruction or experience (Lampropoulos & Tsihrintzis,

2015). Knowledge arises through the interaction with the environment, where this process results gradually in improving the behavior components, considering the following parameters (Russell & Norvig, 2010):

- The components to be improved.
- The prior knowledge the person (agent) already has.
- The kind of representations to be used.
- The available feedback to learn from.

The modeling process of information and knowledge, as well as the simulation of structures and mechanisms that make learning possible, introduce the concept of cognitive system. A cognitive system is a natural or artificial information processing system, including those systems with the capability of perception, learning, reasoning, decision making, communication and action. The concept of learning in a cognitive system is related with the capability of acquiring knowledge by interacting with the environment in which the system operates, and also the capability of improving the state it performs an operation, and consequently its performance through repetition (Vlahavas et al., 2006). Linear and nonlinear models, nonparametric models and support vector machines (SVM) are examples of typical learning systems (Russell & Norvig, 2010).

## The Field of Machine Learning

Machine learning is a subcategory of artificial intelligence, that intersects with several scientific fields such as statistics, mathematics, computer science among others. Under this scope, the learning process is based on examples, which may be related to data from observations, instructions, etc. Machine learning focuses on algorithms that enable computers to do things, taking advantage of the previous information-knowledge.

A prominent feature of machine learning is the processes automation with or without the least possible human effort. The classification and the regression constitute the two main categories of problems in the machine learning field. In the first case, the objective is to categorize data into predefined categories-classes, while in the second one the objective is to predict a real value. What is desirable in either case is the development of general purpose - efficient - algorithms (Schapire, 2008).

## Models and Patterns

In order to understand and represent the environment in a simplified and abstract fashion, we use models. A model is an artificial system that we build in order to simulate and understand a physical system. Key features of a model are the abstraction and the predictive power. Both of these features relate to the relationship of the model with the original (natural) system of reference. Abstraction is related to the selection of those features which are considered as essential, and therefore are incorporated into the model. The attributes that are considered as minor or coincidental or peripheral are omitted in order to simplify and comprehend the reference system. The selection of the essential features concerns both the construction and functionality of the model (Protopapas, 2004).

This methodology is a typical inductive example, where learning is defined as the acquisition or discovery of general relations - dependencies - rules from particular given examples (Vlahavas et al., 2006; Lampropoulos & Tsihrintzis, 2015). Respectively, the organization and correlation of experiences in a new kind of structure, recommends patterns. The construction of models or patterns from a dataset, within the context of a computational system, constitutes the framework of machine learning (Vlahavas et al., 2006).

## Research Lines

Machine learning refers to computational modeling of learning processes, based on the available learning theories. Within the field of machine learning, different approaches are distinguished.

The first approach is about the development and optimization of learning systems, which have the capacity to simulate accurately specific learning tasks (i.e., task-oriented studies). The second approach involves the design and construction of computational models of mind, in order to simulate the human learning processes. This research approach is part of a multidisciplinary effort, where scientists from different disciplines work together, in order to decipher the modus operandi of the human mind. The third approach studies the learning methods and algorithms independently of the application domain (Camastra & Vinciarelli, 2007).

## When A Machine Learns?

According to the learning definition, people learn by changing their behavior in a way that makes them perform better in the future. But how and under which conditions does a machine learn? Actually, this question could be answered under a philosophical perspective, but this is outside the scope of this study. In a human-centered approach, learning implies thinking and purpose. Something that learns has to do so intentionally. Therefore, how could we describe learning without purpose - intentionality?

A machine learns whenever it changes its structure, program, or data, based on the input it receives or in response to external stimuli - information, in such a manner that its expected future performance improves (Nilsson et al., 1998). This improvement implies adaptation to new circumstances, as well as pattern detection (Russell & Norvig, 2010). Consequently, a machine learns with respect to a particular task “T”,

performance metric “P”, and type of experience “E”, if the system reliably improves its performance “P” at task “T”, following experience “E” (Mitchell, 2006).

## Machine Learning: Cross-Section of Disciplines

Machine Learning is the natural outgrowth of the intersection of Computer Science and Statistics (Nilsson et al., 1998). Historically, the field of statistics has been more concerned with testing hypotheses, while the machine learning field has been more concerned with formulating the process of generalization as a search through possible hypotheses. Given the fact that while many machine learning techniques do not involve any searching at all, and statistics is far more than hypothesis testing, the above conceptualization is rather an oversimplification (Witten et al., 2011). Given the progress achieved during the last decades in statistics and machine learning in general, there have been emerged more than one convergence points. For example, in practice, standard statistical methods are applied in the phase of constructing and refining of an initial example set (e.g., visualization of data, selection of attributes, discarding outliers, etc.). In general, we can see that many statistical tests are used to validate machine learning models, as well as to evaluate machine learning algorithms (Witten et al., 2011).

Machine learning is undoubtedly an interdisciplinary scientific field, consisting of the cross-fertilization of different disciplines and research directions, theories and methods. The disciplines that have contributed to the renaissance of machine learning, cover a wide range of fields which are summarized in the following chapter.

### Brain models

An essential source of inspiration and motivation for various machine learning techniques is the study of the brain, namely the comprehension of biological neural networks. According to the biological paradigm, artificial neural networks, which are

brain-inspired computational models, consisting of non-linear elements (Nilsson et al., 1998), constitute powerful machine learning tools. Neural networks models achieve state of the art performance on many complex problems. This. approach according to which systems are capable of learning by transforming their internal structure, is called connectionism

### Psychological Models

The science of psychology investigates the relationship between brain function and behavior in the context of a precisely defined environment. This methodological framework studies humans performance in a series of learning tasks-problems using as main research tool the experimental procedure.

There are many different approaches that compose the landscape of psychology, depending on different research methodologies and tools. The cognitive approach focuses on the study of information processing in terms of mental-cognitive processes (e.g., memory, language skills, problem solving, etc.). This approach relies mainly on computational models of cognition, through which cognitive processes are being simulated.

### Artificial Intelligence

Artificial intelligence and machine learning are closely related, since both fields share the idea that systems (models) should be able to learn and adapt through experience. Artificial intelligence refers to a wide range of problems and explores issues such as the role of analogies in learning, how to use the experience of previous situations to deal with new situations, discover rules for expert systems, etc. (Nilsson et al., 1998).

Artificial Intelligence has been influenced by several scientific disciplines (e.g., Philosophy, Mathematics, Economics, Psychology and Neuroscience, Linguistics, Computer engineering, Control theory and cybernetics) (Russell & Norvig, 2010). The

birth of artificial intelligence dates back to the decade of 1950. Since then, many transitional periods have taken place. The main phases-periods of AI are summarized as follows:

- 1952-1969: The General Problem Solver (GPS) program was created. The GPS program was designed in order to imitate human problem-solving protocols. The success of GPS and subsequent similar modeling efforts of cognition led to the formulation of the physical symbol system hypothesis (Newell & Simon, 1976), according to which a physical symbol system has the necessary and sufficient means for general intelligent action (Russell & Norvig, 2010). The symbolic approach defines the nature of representations, as well as their relationship with the mechanisms that produce them. In this context, representations are considered as abstract symbols (Protopapas, 2004). Any system (human or machine) exhibits intelligence, is operating by manipulating data structures composed of symbols (Russell & Norvig, 2010).
- 1966-1973: In these early years of artificial intelligence, the focus of interest was on creating systems, that could play chess or prove mathematical theorems, or generally simulate complex skills and processes. Most of the algorithms were developed during this period were centered around simple syntactic manipulations. The basic algorithm strategy was to test different approaches until the best solution is found (Russell & Norvig, 2010).
- 1969-1979: The methods and strategies of this period were not expandable to larger scale problems. Towards this direction, emphasis was put on exploitation of domain-specific knowledge. This architecture can more effectively deal with problems concerning specific areas of expertise (Russell & Norvig, 2010).
- 1986-present: The progress and achievements of modern artificial intelligence has been largely due to the return of neural networks. The arrival of new algorithms and the evolution of existing ones, as for example the reinvention of

the back-propagation learning algorithm (first found in Bryson & Ho, 1969), gave a new momentum to the field (Russell & Norvig, 2010).

The first attempts of artificial intelligence have not yielded the expected results due to various reasons:

- Limited computational power of CPU's.
- Inefficient algorithms.
- The lack of a global vision.
- Explicit and realistic goals.

The true renaissance of artificial intelligence happened mainly during the last years. Approaches which are based on hidden Markov models (HMMs), Bayesian network formalism, and deep learning architectures are valuable tools in this direction

### Statistics

Statistical learning lies in finding desired dependence-relation for an infinite domain using a finite amount of given data (Lampropoulos & Tsihrintzis, 2015). The problem in this context is to find the optimal way to use samples drawn from unknown probability distributions, in order to decide from which distribution some new sample is drawn. The method refers to predicting the estimated value of an unknown function at a new point, given the values of this function at a set of sample points (Nilsson et al., 1998).

Usually, the number of the desired relations for a finite amount of given data is infinite. Given this fact, the interest shifts in choosing the most appropriate relation per case. To solve this problem, we use the principle of parsimony, or as is well known, the principle of Occam’s razor, according to which one should not increase the number of

entities unnecessarily or make further assumptions than are needed to explain anything. Therefore, when there are many suggested solutions for a problem, we have to choose the simplest one (Lampropoulos & Tsihrintzis, 2015).

### Adaptive Control Theory

Under the adaptive control theory perspective, the problem is approached at the base of estimating the unknown parameters of a process during its operation. The difficulty lies in that the parameters continuously change during operation, and the control process must track these changes. The main difference between adaptive controllers and linear controllers, regards the controller's ability for adjustment, in order to handle unknown model uncertainties (Cao et al., 2012).

## The Contribution of Machine Learning

The modeling of learning processes in the context of machine learning, creates new perspectives in terms of exploring both research and technical issues. Some of them are summarized in the following questions:

- How humans and animals learn, namely what are the possible learning processes that are responsible for the consolidation of memory and learning?
- Which is the optimal way to reach out difficult computational problems?
- How to develop automatic mechanisms for particular kind of problems?

### Engineering Reasons

The contribution of machine learning, beyond the above mentioned, includes important engineering reasons as well.

The first reason concerns the capability of learning systems to adjust their internal structure, by transforming their network elements. This adjustment - regularization, corresponds to the production of correct outputs for a large number of

sample inputs, constraining this way their input - output function. The ultimate purpose of this process is to approximate the relationship that is implicit in the examples (Nilsson et al., 1998).

The second reason regards the process of discovering patterns in large quantities of data. Data mining, namely the discovery of hidden regularities (or important relationships) in the growing volumes of data, constitutes a popular machine learning method. The basic condition for data mining is the partial or complete automation of procedures. The extraction of patterns through these techniques must be in line with some basic principles, regarding technical and economic issues. Another basic requirement, regarding data mining, is that the availability of data should be in substantial quantities (i.e., Big Data). The proposed techniques approach the issue of data mining in two ways: finding the appropriate techniques for finding and describing structural patterns in data, as well as explaining these data and making predictions from it (Witten et al., 2011).

The third reason concerns the continuous improvement of existing machine designs (i.e., make machines being adaptable in the continuously changing environment). Machine learning methods can be used for on-the-job improvement of existing machine designs (Nilsson et al., 1998).

Machine learning is also used in computing problems, when designing and programming explicit algorithms is infeasible (Lampropoulos & Tsihrintzis, 2015). Things are getting complicated when the case concerns large amounts of data. A key advantage of machine learning methods is the management of large volumes of data, enabling to manipulate knowledge and capture more of it than humans do manually (Nilsson et al., 1998; Mitchell, 2006). The main focus of machine learning is about what computational architectures and algorithms can be used, in order to handle the large amount of available data, namely to effectively capture, store, index, retrieve and merge these data.

## Types of Learning

The underlying learning strategies are based on two discrete entities, the teacher and the learner. These two entities constitute a two-way (dynamic) relationship, according to which the teacher has the necessary information and knowledge (i.e., the know how), in order to perform a given task in a given context, while the learner has to learn the necessary information and therefore to acquire the knowledge needed, in order to perform a specific task in a given context (Camastra & Vinciarelli, 2007; Lampropoulos & Tsihrintzis, 2015). The taxonomy of learning strategies is determined by the amount of inference performed by the learner on the information provided by the teacher. According to this assumption, the following learning types are defined: learning from instruction, learning by analogy and learning from examples.

### Learning from Instruction

Learning from instruction is one of the most common types of learning. Most educational systems around the world use this learning strategy as the main educational method in the curriculum process. This type of learning involves two entities (actors), the teacher and the learner. Teacher is responsible for organizing and transmitting information and knowledge to learner. Depending on the context, the teacher's role may be replaced by a textbook or any other equivalent teaching medium. The learner has to perform some inference, in order to transform the incoming information in such a way that the resulting knowledge is potentially usable. This phase is subdivided into individual transformation processes, where intermediate representations arise. These representations must be compatible with learner's cognitive system. The newly acquired information is incorporated in the learner's existing cognitive schemes-structures, by forming new records and/or integrating with prior knowledge (Camastra & Vinciarelli, 2007).

In conclusion, we can say that the learner is required to perform some inference, while the teacher is responsible for the organization and supervision of the entire learning process, ensuring the terms and conditions which incrementally increase learner's knowledge (Camastra & Vinciarelli, 2007; Lampropoulos & Tsihrintzis, 2015).

## Learning by Analogy

Learning by analogy refers to the learning process, that is based on finding similarities in the structure-core level between the pre-existing knowledge of the learner and the incoming information transmitted by the teacher. The acquisition of new knowledge and skills lies in the transformation and incremental processes of the existing knowledge, in a suitable form that is as similar as possible to the desired new concept or skill, on the basis of adaptation and utility in the new situation. An essential feature of learning by analogy is that much of the organization of information-knowledge, as well as the main part of inference procedures is performed almost exclusively by the learner (Camastra & Vinciarelli, 2007; Lampropoulos & Tsihrintzis, 2015).

In terms of processing steps, the presence of new information triggers an identification process, according to which a fact or skill analogous in relevant parameters should be retrieved from memory (Lampropoulos & Tsihrintzis, 2015). The next processing step is the transformation process of the retrieved knowledge, in a proper form, so as to be compatible and applicable to the new situation (Camastra & Vinciarelli, 2007).

## Learning from Examples

The learning from examples approach, refers to the category of learning problem, according to which from a collection of input-output pairs, a learner infer a function that predicts the output for new inputs (Russell & Norvig, 2010) (i.e., given an example

set of limited size, find a concise data description) (Camastra & Vinciarelli, 2007). This type of learning is a key viewpoint of machine learning. In particular, the learning process is analyzed as finding desired dependencies using a limited number of observations (Lampropoulos & Tsihrintzis, 2015).

Learning from examples uses methods and techniques applied in statistics. Statistical learning is about the formalization of relationships between variables in the data in the form of mathematical equations. Both machine learning and statistics attempt to answer the same question, namely how learning from data is possible. According to statistics, the formulation of a problem is the discovery and processing of the samples drawn from an unknown probability distribution. The purpose here is to find out from which distribution a new sample has been drawn (Lampropoulos & Tsihrintzis, 2015). The role of the learner lies in the inference of a general concept description that describe the examples, given a set of examples. In this case, the roles between teacher and student are reversed, since the amount of inference performed by the learner is much greater than in other types of learning. Hence, the inference processes performed by the learner, imply a higher degree of complexity, comparing to the above mentioned learning types (i.e. learning from instruction, learning by analogy) (Camastra & Vinciarelli, 2007; Lampropoulos & Tsihrintzis, 2015). The classification of learning techniques in the context of machine learning recommends three classes (i.e., supervised learning, unsupervised learning and reinforcement learning).

### Inductive & Deductive Learning

Focusing on the learning from examples approach, two basic subcategories are distinguished: the inductive and the deductive learning.

In inductive learning, the proceeding of inference, has as starting point one or more particular cases, in order to come up to a general case. Another formulation of this method, is to begin from a finite sample of data and to come up to a generalization

about a whole population. This kind of learning aims to search for a general function or rule, using exclusively input-output pairs. This inference process does not necessarily guarantee that the function describing this input-output relationship is always the correct one (Russell & Norvig, 2010). The key point in inductive learning is the extraction of general relations rules (dependencies), namely similar characteristics (common patterns) from a set of particular given examples, generalizing (potentially) for any future test set (Russell & Norvig, 2010; Lampropoulos & Tsihrintzis, 2015). Inductive learning is the most popular type of learning within the machine learning methods.

Deductive or analytical learning is based on existing facts and knowledge. The main goal is to extract new knowledge using previous knowledge. The deductive reasoning method uses a set of known relations-rules (dependencies) that match the training data (Lampropoulos & Tsihrintzis, 2015), going from a known general rule to a new rule that is logically entailed (Russell & Norvig, 2010). In conclusion, we can say that the new knowledge-inferences in deductive learning is in a sense implicit in the initial knowledge.

Depending on the type of feedback, learning from examples can be also be divided in three additional categories. This division is mostly based on how to use and exploit the training data. The purpose here is to find a general description (inference/ dependency relationship), that describe the data about a specific kind of problem (Lampropoulos & Tsihrintzis, 2015), namely to weigh the problem according to the data. These broad categories, which will be discussed in detail below, are the following: supervised learning, unsupervised learning and reinforcement learning.

### Supervised Learning

Supervised learning or learning from examples is the most common form of machine learning, whether it's deep or not (LeCun et al., 2015). The system learns a function or

concept based on the dataset, which is a description of a model. The supervised learning process can be summarized as follows (Vlahavas et al., 2006):

- The system must decode the incoming information (i.e., input data), forming an expression of the model, by learning inductively a function called target function (e.g., denoted by "c"). The target function is used to predict the value of a variable, called dependent variable or output variable, according to the values of a set of variables, called independent variables or input variables or characteristics.
- A set of instances (e.g., denoted by "X"), corresponds to the total of the different input values of the function, where every instance is described from a set of attributes or features. Accordingly, a subset of the set of instances for which we know the variable output value, corresponds to the training set (e.g., denoted by "D").
- The system then examines several alternative functions, so as to approximate the target function in a more precisely manner. These alternative functions are called hypotheses (e.g., denoted by "h"), while the total of all these possible hypotheses is denoted by "H" (Vlahavas et al., 2006).
- Inductive learning is based on the inductive learning hypothesis, according to which every hypothesis that has been found to closely approximate the target function for a sufficiently large set of examples, will also approach in the same manner the objective function for instances that have not previously examined. A hypothesized function is said to generalize when it guesses well on the testing set. Both mean-squared-error and the total number of errors are common measures (Nilsson et al., 1998).

Supervised learning includes classification, prediction and regression tasks, where the objects that are related to a specific concept are pairs of input-output patterns.

The system observes some example input - output pairs and learns a function that maps from input to output (Russell & Norvig, 2010). This assumption implies that data that belong to the same concept are already associated with target values (Lampropoulos & Tsihrintzis, 2015).

The main supervised learning techniques are the following: (Vlahavas et al., 2006):

- Concept Learning.
- Classification or Decision Trees.
- Rule Learning.
- Instance Based Learning.
- Bayes Learning.
- Linear Regression.
- Neural Networks.
- Support Vector Machines (SVM).

### Unsupervised learning

Unsupervised learning or learning from observation refers to formulation of a concise description of data by passively mapping or clustering data according to some order principles, where the system is required to discover correlations or groups in a dataset, in order to generate patterns. According to this type of learning, the data constitute only a set of objects, no label is available regarding the specific associated concept, while it is not known whether patterns will emerge (Vlahavas et al., 2006; Lampropoulos & Tsihrintzis, 2015). Consequently, there is only a training set of vectors without function values for them (Nilsson et al., 1998). The focus is on creating groups-clusters of similar objects according to a similarity criterion, and afterwards to infer a concept that is shared among these objects (Theodoridis & Koutroumbas, 1999).

Unsupervised learning methods are implemented in taxonomic problems, where the goal is to invent ways to classify data into meaningful categories (Nilsson et al., 1998). This class of methods focus on learning patterns in the input even though no explicit feedback is supplied. Detecting potentially useful clusters of input examples, is one of the most common tasks in unsupervised learning (Russell & Norvig, 2010).

Unsupervised learning deals with the issue of dimensionality reduction. The basic concept regarding the dimensions in a problem, is to preserve the initial information of data, while ensuring an optimal way of computational complexity. There are extensive references in the literature on this issue, which is commonly referred to as the curse of dimensionality. According to the curse of dimensionality, the number of variables in multivariate problems requires exponentially increasing the amount of computational resources (Lampropoulos & Tsihrintzis, 2015). High dimensionality of data is a fundamental barrier in many Computational science and engineering applications. One of the most characteristic problems, that is raised in the context of pattern classification tasks, is related to the learning complexity. In particular, the learning complexity grows exponentially with the linear increase in the dimensionality of the data.

In order to address the problem of dimensionality, some pre-processing steps on the raw data are required (Arel & Rose, 2010). Unsupervised learning techniques include algorithms that preserve the initial information of data, allowing a more efficient computation. This process is carried out by providing a transformation on the data, by computing an intermediate representation from high-dimensional to low-dimensional spaces. This dimensionality reduction scheme is often referred to as feature extraction (Arel & Rose, 2010; Lampropoulos & Tsihrintzis, 2015).

## Reinforcement Learning

Reinforcement Learning is inspired by phycological theories, and in particular by behaviorist psychology. In terms of neurological level, reinforcement learning is referred to the basal ganglia in subcortical nuclei of the mammal brain (Lampropoulos & Tsihrintzis, 2015). Additionally, this type of learning has been influenced by control theory (Camastra & Vinciarelli, 2007; Russell & Norvig, 2010; Cao et al., 2012). Under the control theory perspective, the problem is approached at the base of estimating the unknown parameters of a process during its operation (Cao et al., 2012) in a dynamic environment that results in state-action-reward triples as the data (Camastra & Vinciarelli, 2007).

In reinforcement learning the basic idea is related to the procedure of performing actions in order to achieve a goal. The basic concept in this type of learning is to yield an efficient method to develop goal-directed action strategies, through trial and error procedures. Reinforcement learning applies to that category of problems where there is not a priori explicit knowledge on the actions to be taken in order to perform a task. An agent (system) is allowed to automatically determine the ideal behaviour within a specific context, in order to maximize its performance, defined by the rewards that the agent get in a given state (Camastra & Vinciarelli, 2007; Lampropoulos & Tsihrintzis, 2015).

This type of learning differs from the ones mentioned previously (i.e., supervised learning, unsupervised learning), since the learning algorithm is not informed which actions to take into account regarding a given situation (Camastra & Vinciarelli, 2007). Actually, there are not taken into account input-output pairs at all. The feedback the learner receives is a kind of reward, which may not be provided in real time, i.e., immediately after the action is taken (Camastra & Vinciarelli, 2007; Lampropoulos & Tsihrintzis, 2015), while the concise description of data is the strategy that maximizes the reward. A major challenge of a reinforcement learning algorithm is

to find out a trade-off between exploration and exploitation. This is feasible either by choosing those actions that have been successfully tested in the past and proved to be effective in producing reward, or by discovering those actions that have not been tried in the past so as to explore thoroughly the state space (Camastra & Vinciarelli, 2007).

# Neural Networks

## Introduction

Neural networks are brain-inspired computational models that belong to the class of parametric non-linear models. In general, a neural network is a collection of units connected together. The properties of the network are determined by its topology and the properties of the neurons (Russell & Norvig, 2010). The study of neural network theory has received significant influences from many different scientific disciplines. In this perspective, it has drawn extensively on concepts from neuroscience and psychology, as well as from the tools of mathematics and computer science. Neural network models are based on the biological paradigm, as they use structures and processes that mimic those of the human brain. Consequently, these models give us insights into the cognitive functions, as an attempt to model the information processing capabilities of the brain.

Neurons are the base elements of the brain, that transmit impulses in the nervous system (Basegmez 2014). As an artificial neuron is defined a computational model, the parts of which can be matched with those of a biological neuron (Vlahavas et al., 2006). Artificial neural networks are algorithms inspired from the biological neurons and synaptic links (Hamed et al. 2013). They are data processing systems consisting of artificial neurons organized in structures similar to those of human brain, and can be considered as non-linear estimators, which depend on nonlinear approximations applied directly on the input space (Lampropoulos & Tsihrintzis, 2015). The architecture and the basic principles of operation of neural networks models, make them an important tool for dealing with many difficult problems. In many cases, neural networks are used as a black box, regarding the processing procedures that take place in the hidden layers of the network. In a typical neural network, a certain input produces a desired output,

but how the network achieves this result is left to a self-organizing process (Rojas, 1996).

Neural networks consist of non-linear elements, interconnected through adjustable weights, where computation is explained in terms of network interactions. Neural networks have been widely used as computational models of knowledge and cognition. They constitute one of the most popular and effective forms of learning systems, and play an important role in the machine learning field (Nilsson et al., 1998).

The comparison between a biological and an artificial neuron is evaluated in terms of an abstract fashion. According to this scheme, there are retained some functional properties of natural neural networks, which are considered to be important for computational properties, regarding the capability of implementing distributed processing, tolerance to noisy inputs and learning (Protopapas, 2004).

## Biological Neural Networks

A biological neural network is composed of a collection of neurons, which are the elementary building blocks of the network, interconnected with each other via axon terminals. The constituent parts of a typical biological neuron, are the body, which constitutes the core of the neuron, the dendrites, which collect information sent on by other neurons, and the axon, a primary output pathway by which neurons send information to other neurons.

Signals (input) that reach a neuron are defined as afferent inputs. The fact that the axon of the afferent neuron does not contact the dendrite of the receiving neuron, results in the creation of a gap between the neurons. These gaps between neurons are called synapses. The axons and dendrites are connected to each other via synapses (Protopapas, 2004; Kandel et al., 2006; Basegmez, 2014). Chemical procedures that take place among the synapses attain the acceleration or deceleration of the flow of electrical charges in the body of the neuron. Subsequently, neurotransmitters released

by the afferent neuron into the synapse are picked up by receptors in the receiving neuron's dendrites. The action of some neurotransmitters results in an excitatory effect on the receiving neuron, thus increasing the network activation. Vice versa, other neurotransmitters have an inhibitory effect, reducing the network activity of the receiving neuron. Both excitatory and inhibitory inputs can vary in strength depending on two things: the chemical composition of the neurotransmitters and the strength of the synapse (Gluck & Myers, 2001).

Cognitive functions, as they emerge from the underlying brain processes, suggest that the brain function is resulting from the ability of the synapses to constantly change their conductance. The electrical signals that enter the body of neurons operate in combination rather than individually. This property combined with the existence of a threshold trigger, defines the architecture and the basic principles of neural network operation. In particular, when the combined result exceed a threshold value (usually this value is predefined), the signal is propagated to the other neurons through the axon (Vlahavas et al., 2006).

Given the above, we conclude that biological neural networks are self-organizing systems and each individual neuron is also a delicate self-organizing structure capable of processing information in many different ways. This massive and hierarchical-network structure of the brain seems to be the basic precondition for the emergence of complex behavior and consciousness (Rojas, 1996).

## Artificial neuron

The modeling processes of a biological neuron are based on the choice of the fundamental elements of its basic architecture and function. According to this abstraction, an artificial neuron is considered to be a biological model. Unlike the electrical pulses detected in the brain, the input signals received by an artificial neuron are continuous variables. The variation of any such input signal varies depending on the

weight value. This corresponds to the role of the synapse in a biological neuron. This value receives a positive or negative sign respectively, simulating this way the accelerating or decelerating operation of the synapse. The body of the artificial neuron consists of two parts. The first one is the sum module, which adds the modulated input signals, in order to produce a quantity (i.e., "S"). The other part of the body is the activation function, which operate as a kind of filter and forms the final value of the output signal (Vlahavas et al., 2006).

The design and construction of artificial neural networks is based, as is true for any attempt to model a system, on an abstraction process, with regard to the physical system - network. The degree of complexity of a real biological neuron defines the context of artificial neural networks theory. According to this, we do not consider the whole complexity of the real biological neurons. The general practice used in order to model neural networks is to conceive each neuron as a function that produces numerical results at some points in time. Therefore, only some principles of biological neural networks are used, while each case determine the desired level of detail (Rojas, 1996).

A network structure (network) is composed by multiple artificial neurons, which are connected to each other. The type and structure of the connection depends on the architecture and the topology of each network. In the most common case, each neuron transmits its activation to all the following neurons it is connected with. The input received by each neuron is the result of the processing being performed by its weighted input connections. Specifically, the incoming input value gets multiplied by the weight associated with that connection. The final product of this process (i.e., the resulting signal) is combined with signals, respectively derived from other connected neurons. Finally, through the activation function, the activation of the neuron is performed or not (Gudi, 2014).

The essential properties of neurons, which are considered as elementary on network-level and are usually retained in the artificial neural models are the following (Protopapas, 2004):

- Each neuron receives signals and is activated or not to send signals to other neurons.
- The neuron transmits the activity of its inputs, meaning that it can accept a total of strong or weak inhibition or stimulation.
- The effect of stimulating a neuron on another is proportional to the connection strength between them.
- The learning process in a neural network is achieved through the changing in the synaptic weights.

## Artificial Neural Networks

Artificial neural networks theory, which is also called connectionism, parallel distributed processing (PDP), or neural computation, lies on the essential properties of biological neural networks, in terms of information processing and without considering the whole complexity of real biological neurons. Artificial neural networks, in general, are networks of interconnected units, which serve as model neurons, computing nonlinear functions of their input. Beyond the similarities with the biological model, an artificial neural network may be viewed as a statistical processor, which delivers probabilistic assumptions about data (Jordan et al., 2006). The main difference between artificial neural networks and conventional computer systems lies in the massive parallelism and redundancy processeses. These processes allow to address issues with regard to the unreliability of the individual computation units.

## Structure of Neural Networks

The functionality and flexibility of neural networks arises from the incorporation of several key features, such as adaptability, nonlinearity and arbitrary function mapping capacity. These features, in the context of neural network architecture, make them one of the most reliable solutions to resolve problems such as pattern and sequence recognition (classification), data processing (e.g., filtering, clustering, etc.) (Zorbas et al. 2015). The hierarchical multilayered structure of neural networks enable the transmission of information not only to the immediate neighbours but also to more distant units (Rojas, 1996).

An artificial neural network is actually a graph, with vertices (neurons, or units) and edges (connections) between vertices (Jordan et al., 2006). Such a network is composed of a system of interconnected artificial neurons, which can compute their activations based on the weights of their connections and the activations of their neighbours (Gudi, 2014). The processing units of a neural network are the nodes (or units), that constitute a simplified mathematical abstraction of the basic functionality of a neuron. The neurons are connected each other with directed connections (links). Each node in a network includes two types of connections, i.e., afferent and abductive connections. Furthermore, each node is characterized by a degree of stimulation or activation level. The degree of stimulation depends both on the particular node and on the sum of all the incoming connections.

The power of nodes on a neural network, as is also the case in a network of biological neurons, is not the result of individual neuronal activity but the result of the collective power that occurs when many nodes are interconnected in a network (Gluck & Myers, 2001). The relationship between the activity carried through the afferent connections and the consequent stimulation of a node is called activation function or integration function (Protopapas, 2004).

The function of each synapse is modeled by a modifiable weight, which is associated with each connection (Polk & Seifert, 2002). Each link is associated with a numeric weight, which determines the strength and sign of the connection and is characterized by a numeric value called associative weight (weight) (Protopapas, 2004; Russell & Norvig, 2010). Each node in the network is characterized by an activation level. This activation level can be considered as the probability that the node will generate output of its own, or as an approximation of the node's response strength on a scale from 0, where the node is not active, to 1, where the node is fully active (Gluck & Myers, 2001).

Each unit-node converts the pattern of incoming activities it receives into a single outgoing activity, which transmits to other units-nodes. The sequence of steps that make up this process is as follows: Initially, the node multiplies each incoming activity by the weight on the connection, and then adds together all these weighted inputs to get the total input. Subsequently, the node uses an activation function that transforms the resulting total input into the outgoing activity (Polk & Seifert, 2002).

### Activation Functions

The selection of the activation function is mainly based on their properties to simplify or enhance the overall behavior of the network (Gudi, 2014). The activation function meets two basic conditions. The first condition is related to the activation or not of a node. The node is active close to 1, given the correct inputs, and inactive close to 0, given erroneous inputs. According to the second condition, the activation should be non-linear, because otherwise the overall neural network will degenerate into a simple linear function, by significantly limiting the computing power of the network (Rojas, 1996).

There are several types of activation functions, with the following cases being the most typical (Polk & Seifert, 2002; Vlahavas et al., 2006):

- *Step function*. This function gives as an output result, usually 1, if and only if the value calculated by the adder is greater than a given threshold value (i.e., "T").
- *Sign function*. The resulting value of the output is positive or negative respectively, if the rate calculated by the adder is greater or lower than a given threshold value (i.e., "T").
- *Logistic function*. This type of function is part of a broader class of functions, called sigmoid functions. The comparative advantage of sigmoid functions lies in their ability to be continuous and differentiable in the entire range of input values, as well as in their ability to limit the output between 0 and 1 or between -1 and 1. Another property of sigmoid functions is that they are used as a filter, by suppressing the large values, while giving satisfactory output for small input values. The fact that the output varies continuously but not linearly as the input changes, ensures that the network can implement a nonlinear function. The structure and the functionality of sigmoid units is closer to the biological plausibility of real neurons (Polk & Seifert, 2002; Vlahavas et al., 2006).

## Neural Network Architectures

The efficiency of a neural network is directly related to its topology (i.e., architecture) and the learning algorithm that is used per case in order to find the weights of the network. Depending on the oroblem to be solved, network's nodes may be linked in different ways (Polk & Seifert, 2002), recommending different learning styles (Polk & Seifert, 2002).

The most common type of artificial neural network, in terms of its architecture and structure, consists of three groups or layers of units: the input layer, the hidden layer/layers and the output layer (Polk & Seifert, 2002; Protopapas, 2004):

- *Input layer - nodes*. These nodes are at the input layer of the network and represent the raw information that is fed into the network. Each input can be either excitatory or inhibitory, whilst is can also be characterized as strong or weak.
- *Output layer - nodes*. Output nodes are referred to those nodes from which we choose to take the result of the network processing.
- *Hidden layer(s) - nodes*. Between the input and output nodes there is a layer of intermediate nodes, the hidden nodes. These nodes neither accept external stimuli nor are they part of the network processing result. The activity of each hidden node is determined by the activities of the input nodes, as well as the weights on the connections between the input and hidden nodes (Polk & Seifert, 2002; Protopapas, 2004).

The most popular neural network architectures are the feedforward and recurrent neural networks, which are discussed in the next section.

## Feedforward Neural Networks

A Feedforward network or multi-layer perceptron (MLP) is the most widely used architecture for implementing neural network models. According to this type of architecture, a network is consisted of a sequence of layers, where each layer is composed of a number of units (nodes). Information flows from the bottom to the top (i.e., bottom-up approach). In other words, every node in the network receives input from upstream nodes and delivers output to downstream nodes. The connectivity of a feedforward network is complete in the sense that the units are arranged in layers, so that all units in successive layers are fully connected (Hamed et al. 2013).

Typically, feedforward networks have one input layer, one or more hidden layers, and an output layer. The connection between the input and output layers is

unidirectional and there is no connection between the neurons within a layer. This is equivalent to a directed acyclic graph, namely the units-nodes and connections of the network correspond to the nodes and edges of the graph (Russell & Norvig, 2010; Kriegeskorte, 2015).

When the number of hidden layers in a network increases, the network is called deep neural network. There is no clear limit or some formal definition according to which a network is defined as deep. In that sense, if there is more than one hidden layers, the network is called deep neural network (DNN). Each neuron in the hidden layer(s) calculates its output using the activation function, so the values are propagated from the input layer through hidden layer(s) to the output layer. Moreover, a basic feature of this type of network is that there are no loops (Basegmez, 2014).

Feedforward neural networks are universal function approximators, which calculates a static function mapping inputs to outputs. Therefore, the presentation of a stimulus (input) leads to the direct production of a response (output). These processes within the layers of the network represent a snapshot of the current input to the output (Protopapas, 2004). Actually, all the weights themselves define the internal state of the network (Russell & Norvig, 2010).

Both the training process and the mathematical description of the function of a neural network can be described as relatively simple. The drawback of this type of networks lies on the limited range of their computational capacity (Protopapas, 2004).

## Recurrent Neural Networks

Recurrent Neural Networks (RNN) are networks that are based on a special type of architecture, where units can be connected in cycles. A recurrent network is equivalent to a feedforward network, which is consisted of an infinite number of layers, where each layer is connected to the next one by the same weights matrix. The information flow within the network in a recurrent manner, generates ongoing dynamics, which

contribute to the processing of temporal sequences of inputs. This feature constitutes a comparative advantage over feedforward networks, since it can approximate any dynamical system, given a sufficient number of nodes (units) (Kriegeskorte, 2015).

The structure and architecture of recurrent networks is similar to that of biological neural networks, in which lateral and feedback connections are ubiquitous. Given these properties, recurrent networks are characterized as more realistic compared to feedforward network (Kriegeskorte, 2015). A recurrent network is also a powerful tool for modeling sequence data.

In a recurrent network there may be connections between the neurons within a layer, feeding its outputs back into its own inputs (Russell & Norvig, 2010), namely each hidden unit can interact with each other hidden unit (Kriegeskorte, 2015). These networks process the data stream successively, e.g., an input sequence one element at a time. According to this processing model, a history of the data is created, maintaining additional - implicit - information about the elements of the sequence in the form of a state vector (LeCun et al., 2015). The dynamic behavior of the system emerges from this sequential processing style. Particularly, the activation levels of the network form a dynamical system that may reach a stable state or exhibit oscillations or even chaotic behavior (Russell & Norvig, 2010). The stability of such a network - system depends on the relationship between inhibitory and excitatory connections (Protopapas, 2004).

Recurrent networks can support short-term memory (STM), since the response of the network to a given input depends on its initial state, which may depend on previous inputs (Russell & Norvig, 2010). This structure makes it possible to remember the context of sequential data (Basegmez, 2014).

The benefits of using recurrent models to solve complex computational problems, make these systems a worthwhile solution. Meanwhile, the use of these systems has several disadvantages. Problems are mostly related to the training phase, which may be quite difficult and problematic because of several instabilities (e.g., get

stuck in local minima, etc.). Also, the mathematical description of the functions used is often impossible due to high complexity of networks parameters (Protopapas, 2004). Indicatively, a common problem that occurs during the training process, is that the backpropagated gradients either grow or shrink at each time step, so over many time steps they typically explode or vanish (LeCun et al., 2015), an issue that will be extensively analyzed in the next section.

## Neural Networks: Learning and Training Processes

Learning process in terms of neural networks implies the adjustment of the weights of the connections between the neurons in the system. This adjustment of the weights in the synapses define the behavior of the individual neurons, as well as the overall behavior of the network. The training of a neural network is accomplished using sophisticated algorithms. Depending on the network's architecture, the algorithm that is appropriate for each case is chosen. In any case, regardless of the type of learning, i.e., whether it is supervised or unsupervised learning, the objective is to optimize the distribution of network weights. That means to bring the network into a state of equilibrium.

From time to time dozens of algorithms have been proposed in the international literature. Some of these algorithms laid the foundation for the evolution of artificial neurons and inspired the creation of new algorithms. Some of the most popular, efficient and commonly involved in machine learning tasks algorithms are the following (Basegmez, 2014; Lake et al., 2016):

- The Perceptron algorithm (Rosenblatt, 1958).
- Hebbian learning (Hebb, 1949).
- The BCM rule (Bienenstock et al., 1982).
- Backpropagation algorithm (Rumelhart et al., 1986).
- The wake-sleep algorithm (Hinton et al., 1995).

- Contrastive divergence (Hinton, 2002).

## The Perceptron Network

Perceptron network, or single-layer neural network is a network with all the inputs connected directly to the outputs (Russell & Norvig, 2010). Perceptrons are the basic neural network building blocks, which are used for training algorithms (Basegmez, 2014). Given the fact that each weight of the network affects only one of the outputs, we consider a perceptron network as corresponding to n separate networks, according to the n number of output units. In other words, what the perceptron networks actually does, is the decomposition into n separate learning problems, given an n-output problem. Consequently, it will be n separate training processes (Russell & Norvig, 2010). As we will see later, even if this particular process of decomposition works well for this type of network architecture, it fails in a multilayer network.

The basic training process of a three-layer neural network includes the following steps. The first step corresponds to the training phase, during which we feed the network with examples. These examples or as we usually call them "training data", consist of a pattern of activities for the input units together with the desired pattern of activities for the output units. Given some well-defined criteria, which are usually set at the beginning of the network training process, we then examine the degree of similarity between the desired output of the network and the actual output of the network. This comparison process leads to the determination of the error rate.

In order to achieve better results and produce a better approximation of the desired output in the network, we change slightly the weight of each connection, so as to succeed gradually an optimized result (Polk & Seifert, 2002). The weights should be adjusted gradually to reach the desired behavior, which is possible using different activation functions. This gradual adjustment will allow new neurons to produce values

between the range of 0 and 1. A major drawback of perceptron's networks is that they can only carry out linearly separable functions (Basegmez, 2014).

Depending on the type of activation function used per case, the training algorithm will be either the perceptron learning rule, i.e., $w_i \leftarrow w_i + \alpha\ (y - h_w\ (x)) \times x_i$, or the gradient descent rule for the logistic regression (Russell & Norvig, 2010).

## Gradient Descent Training Rule

The output produced by a neuron, whether it is a neuron belonging at the hidden layer/layers, or respectively a neuron belonging at the output layer, depends both on the weight of the connections and an additional numeric value, called bias. Bias is a value that is added to the sum calculated at each node, without counting the nodes belonging at the input layer, during the feedforward phase. The bias value is usually by default 1 and is used within a neural network architecture mostly for computational purposes (Protopapas, 2004). The expected goal during the training phase of a network is to find the minimum output error, i.e., the minimum difference between the desired output and the actual output of the network.

### Description

As discussed in the previous section, the training phase of a neural network is done by feeding it using training data. The assessment of whether the network has been adequately trained and can generalize the acquired knowledge into new examples, is done through comparing the actual activity of the output(s) units with the desired result. Subsequently, what needs to be done is the calculation of the error, which is defined as the square of the difference between the actual and the desired activities.

In order to reduce the error, we have to modify the weight of each connection in a sophisticated way. To perform this process, we first have to compute the error derivative for each weight. Calculating this quantity, i.e., the error derivative, we

perform the necessary change at each weight on the nodes of the network by an amount that is proportional to the rate at which the error changes as the weight is changed (Polk & Seifert, 2002), that is the derivative of the error with respect to the weight. This iterative reduction of the errors through small adjustments to the weights, is one of the most popular and effective training methods for training a neural network model and is known as gradient descent (Kriegeskorte, 2015).

Gradient descent learning algorithm is based on the idea of finding how much a slight change of the weights will reduce the output error. The initialization of weights at the beginning of the training process is done randomly. The adjustment of each weight is being calculated in proportion to the effect on the error. The gradient descent method guarantees that we are moving in that direction in weight space, along which the error descends most steeply (Kriegeskorte, 2015).

The gradient descent method is a local algorithm, which takes into account only the local neighborhood in the weight space. This local nature of the method is an important advantage compared to other training algorithms. The comparative advantage lies in the fact that with so many available directions to move in, gradient descent is unlikely to get stuck in local minima. Getting stuck in local minima means that the error increases in all directions and no further progress is possible (Russell & Norvig, 2010; Polk & Seifert, 2002; Kriegeskorte, 2015). All these properties of gradient descent algorithm make it a valuable tool for finding stable and effective solutions.

### Limitations

Meanwhile, the high dimensional training data, pose a series of challenges and obstacles to the algorithm's optimal operation. The increased dimensionality in the weight space in combination with the architecture of the network (e.g., connections driving the preceding layers), make the use of gradient base algorithm, at least in its simplest version, insufficient.

A classic problem that arises concerns the calculation derivative of the output error. The difficulty lies on the individual calculation for each training sample. The solutions that have been proposed to address the problem focus on variations of sampling.

### Stochastic Gradient Descent (SGD)

Stochastic gradient descent method (SGD) is a proposed solution, used to speed up the training process. The algorithm is based on the estimation of the true gradient, using small amount of samples, from which samples it then extracts the average (mean) (Basegmez, 2014).

The basic processes of the algorithm are described as follows (Polk & Seifert, 2002; LeCun et al., 2015):

- Initially, the input vector is presented only for a few examples.
- Then the outputs, as well as the errors, are calculated.
- The next step of the algorithm corresponds to the calculation of the average gradient for the selected examples and to the corresponding weights adjustment.
- These processes are performed iteratively for many small sets of examples, drawing from the total set of instances (training examples), until the average of the objective function stops decreasing (LeCun et al., 2015).

## Back-propagation Algorithm

The essential theoretical ideas regarding the back-propagation learning algorithm were first developed in 1969 by Bryson and Ho, while the algorithm was reinvented a few years later, in the mid-1980s by several different research groups (Rogers & McClelland, 2004; Russell & Norvig, 2010; LeCun et al., 2015). The back-propagation (or backpropagation) algorithm is an efficient process for calculating the error derivative for the weight (EW). The algorithm has been applied successfully towards a

wide range of problems with regard to different scientific fields (e.g., computer science, psychology, cognitive science, etc.). The back-propagation algorithm was mainly spread within the framework of Parallel Distributed Processing (PDP) (Russell & Norvig, 2010), and is still a valuable option for neural networks training.

### Description

Backpropagation algorithm is a gradient-descent method that makes iterative small adjustments to the weights of the nodes, in order to reduce the errors of the outputs (Rumelhart et al., 1986). This method constitute an inherently gradual learning process, which is based on the presentation of many training examples. The algorithm addresses the fundamental problem of discovering useful representations in data, which means that it can teach the hidden units of the network to produce interesting representations of complex input patterns (Polk & Seifert, 2002).

We can see the backpropagation algorithm as a practical application of the chain rule for derivatives. That is the computation of the gradient of an objective function with respect to the weights of a multilayer stack of modules. Under this perspective, the key point of the process is that the gradient of the objective function can be calculated with respect to the input of a module. The process works backwards from the gradient with respect to the output of that module or respectively the input of the subsequent module. This process can be applied repeatedly to propagate gradients through all modules, starting from the output layer to the input layer (LeCun et al., 2015).

As previously mentioned, a perceptron network (i.e., a simple form of neural network) decomposes into n separate learning problems for an n-output problem. However, this kind of decomposition is not possible if the network consists of many hidden levels. This is the case of a multilayer neural network. In this regard, the problem lies in the computational complexity involved in adding additional layers of hidden nodes. While the error calculation in the output layer is clearly evident, the

situation is more complicated in hidden layers. Having as a sole source of information the training data, we cannot estimate accurately the values of the nodes in the hidden layers (Rumelhart et al., 1986; Russell & Norvig, 2010).

In particular, if we add the gradient contributions from each layer during the weights' update, a multilayer network decomposes an n-output learning problem into n learning problems. In order to accomplish this decomposition process, we have to propagate the error from the output layer directly to the hidden layers. The backpropagation process emerges directly from a derivation of the overall error gradient. A basic requirement for updating the connections between the input units and the hidden units, is the definition of a quantity, which must be analogous to the error term for the output nodes (Russell & Norvig, 2010).

The basic processes of the backpropagation algorithm are described as follows (Russell & Norvig, 2010; Basegmez, 2014; Rogers & McClelland, 2004):

- The network is given the input and the neurons (nodes) are activated until the upper layer.
- The output error of the upper layer is calculated using the observed error of the layers below.
- Then, the error(s) is backpropagated, in order to calculate the error of the former layer.
- The gradient of the error function corresponds to the multiplication of the output error of the neuron and the activation function of the stimulating neuron in the former layer. This multiplication process is repeated for each training sample.

Based on the assumption that all the units of the network are linear, the algorithm computes each error derivative of the weights (EW), while computing the rate at which the error changes as the activity level (EA) of a unit (node) is changed.

Specifically, for the output nodes, the EA corresponds to the difference between the actual and the desired output.

When we want to compute the EA for a hidden unit, which belongs in the hidden layer just before the output layer (i.e., last hidden layer), we must take into account all the weights between that hidden unit(s) and the output units to which it is connected. Subsequently, we multiply those weights by the EAs of those output units and then add the products. After calculating all the EAs in the last hidden layer, we similarly follow the same procedure for the remaining layers of the network. The workflow (i.e., direction of processing) is evolving from layer to layer in a direction opposite to the way the activities propagate through the network. The EW is the product of the EA and constitute the outcome activity through the incoming connection. When we want to apply this method in a network composed of nonlinear units, the process is not significantly different. The differentiation, unlike a network consisting of linear units, lies on the processing step before applying the back-propagation. In that case, the EA must be converted into the rate at which the error changes as the total input received by a unit is changed (EI) (Polk & Seifert, 2002; Rogers & McClelland, 2004).

### Advantages of Back-propagation

Back-propagation algorithm is an important tool for training neural networks based models, consisting of multiple hidden layers and large amount of data. The technique of back propagation makes it possible to create useful representations. These representations, also called internal or intermediate representations, address several fundamental problems related to learning processes. In order to train a multilayer neural network, we use the global convergence technique, according to which the weights are gradually adjusted, so that to reduce the total output error. However, as it turns out, for many machine learning tasks, the global convergence technique is not taken as a prerequisite in order to achieve good performance (Polk & Seifert, 2002).

In terms of Bayesian learning, the backpropagation algorithm implements a Bayes optimal process in the sense that it learns connection weights that maximize the probability of the output given the input (Rogers & McClelland, 2004).

### Limitations

Back-propagation algorithm is an innovative method that combines many advantages. However, it also has some significant drawbacks, depending on the perspective and implementation context.

A major drawback concerns to the biological plausibility of the method. The term biological plausibility refers to the structure and mode of operation of the biological neural system, according to which the information flow does not look like the way the back-propagation algorithm works. Specifically, back-propagation activation seems to require that information must travel through the same connections in the reverse direction, namely from one layer to the previous one. The transmission of information backwards along the axon does not fit with realistic models of neuronal function (Polk & Seifert, 2002; Lake et al., 2016). However, this objection is actually under question, since the brain seems to have many pathways, namely from upper layers back to previous layers. Consequently, the brain could use these pathways in many ways to transmit the information required for learning (Polk & Seifert, 2002).

Another objection to back-propagation as a realistic model of learning has to do with the learning strategy it uses. A basic assumption of the method is that it presupposes feedback for each training instance, namely providing the correct output per case. This seems to contradict what is actually true in real learning conditions, since people learn without direct instructions of how to construct the internal representations of the received stimuli (Polk & Seifert, 2002).

In addition to the above drawbacks, the speed of the algorithm raises questions about its efficiency and computational completeness. The main argument lies in the fact

that the relationship between the learning time and the size of the network is proportional. This dependency shows that the bigger the network, the more time it takes to learn. What actually happens is that the time needed to compute the error derivatives for the weights on a given training example is proportional to the size of the network because the amount of computation is proportional to the number of weights (Gluck & Myers, 2001; Polk & Seifert, 2002). Assuming that bigger scale networks require more training data, it is concluded that the update rate of the weights is also rising. Therefore the learning time grows much faster than does the size of the network (Polk & Seifert, 2002).

# Deep Learning

## Introduction

Deep learning (DL), also known as deep structured learning, hierarchical learning, representation learning or feature learning among others, is a popular research area of machine learning research, that constitutes a class of machine learning techniques, where many layers of information processing stages in hierarchical supervised architectures are exploited for unsupervised feature learning and for pattern analysis or classification (Deng & Yu, 2014). Deep learning methods allow computational models, that are composed of multiple processing layers or stages of nonlinear information processing, to learn representations of data hierarchically with multiple levels of abstraction, by extracting complex relationships among the data (Deng, 2013; LeCun et al., 2015; SAP SE, 2015).

Deep learning techniques constitute a revolutionary machine learning approach that has been applied across a wide range of fields regarding classification, information retrieval, image recognition-retrieval, dimensionality reduction, natural language processing and robotics among others (SAP SE, 2015). While most of these problems are studied extensively in the light of classical machine learning methods (e.g., shallow neural network architectures, Support Vector Machines, etc.), during the last years, deep learning has demonstrated state of the art performance on many of these tasks (Hinton & Osindero, 2006; Vincent et al., 2010; SAP SE, 2015), by discovering intricate structures (patterns) in high-dimensional data (LeCun et al., 2015). Furthermore, the hierarchical nature of learning processes taking place in deep learning architectures, facilitates the processing of big data.

The key point behind the philosophy of deep architectures is the computation of hierarchical features or representations of the observational data (i.e., raw data), where

the higher-level features are defined from the lower-level ones (Deng & Yu, 2014). The features in a deep learning scheme are represented automatically from the data using a general purpose learning procedure (LeCun et al., 2015), drawing ideas and inspiration from biological and circuit complexity theories (SAP SE, 2015).

## A Brief History of Deep Learning

Deep learning is in the point of intersections among several disciplines involving the research areas of artificial neural networks (ANN), artificial intelligence (AI), cognitive science and neuroscience, computational neuroscience, graphical modeling, optimization, pattern recognition and signal processing (Deng & Yu, 2014; Poggio, 2016; Goodfellow et al., 2016). This interdisciplinary machine learning approach has drawn heavily on knowledge from the fields of statistics and applied math (Goodfellow et al., 2016). It also relies on the structure and functionality of the human brain - mind, that is believed to operate in a deep fashion. This architecture, composed of multiple layers of abstraction, allows the creation of multiple types of representation, which have simpler features at the lower levels and more complex features on the upper layers.

### The Origins of Deep Learning Approach

Historically, the concept of deep learning originated from artificial neural network research, within the field of artificial intelligence. The main idea is based on learning large neural network style models with multiple layers of representation, approaching the data in an hierarchical and abstract fashion (Poggio, 2016; Lake et al., 2016). The architecture of deep network models consists of multiple layers of hidden layers, which are capable of learning abstract features of the input by constructing internal representations. The ultimate goal of this iterative process is the creation of even more internal representations independent of the raw data (input), that have been derived from the integration of several patterns (Basegmez, 2014). Consequently, the purpose is

the construction of intermediate representations, through an hierarchical learning process, having as a result the emergence of complex and high level abstractions, driven from the raw data (Najafabadi et al., 2015). These methods have made great progress and have contributed significantly in solving problems, which for a long time remained intractable. (e.g., object recognition, speech recognition, control, etc.) (Schmidhuber, 2014; LeCun et al., 2015).

Although deep learning approach seems to count only a few years as a machine learning paradigm, its origins date back many years ago. Trying to briefly review the milestones of deep learning, we observe the influence of different scientific directions. Recently, deep learning has seen stunning growth in its popularity and usefulness, mainly because of the increased computational power, the availability of larger datasets, as well as of the available techniques to train deeper networks in a more efficient way. The core idea behind deep learning is based primarily on the composition of features at multiple levels under an hierarchical scheme (Goodfellow et al., 2016).

The key points of development of deep learning are summarized in three periods, reflecting different philosophical viewpoints and technological innovations. The field was not always known as deep learning, since it has been renamed several times. The first period of development of deep learning, known as cybernetics, covers the decades between 1940-1960. The second period, known as connectionism, refers to the decades between 1980-1990 (Goodfellow et al., 2016). In the year 2006, the interest in deep feedforward networks was revived and is now known as deep learning. In that context, specific unsupervised learning procedures enable the creation of feature detectors without the need of labelled data (LeCun et al., 2015).

## Deep Motivations

The fundamental assumption of deep learning theory refers to the computation of hierarchical representations of the available input data, where the higher-level features

emerge from lower-level ones. This is probably what the brain seems to do. Actually, brain seems to operate in a deep fashion, attempting to represent the received stimuli into meaningful internal representations (Bengio & LeCun, 2009).

The inspiration behind the hierarchical learning architecture of deep learning algorithms is reinforced by the observation that humans learn concepts in a sequential order (Rocki, 2016), and they organize their ideas and concepts hierarchically, through composition of simpler ideas. They first learn simpler concepts and then compose them in order to represent more abstract ones (Bengio & LeCun, 2009). In particular, the whole process of learning is based on the creation of simple patterns, that contribute to the formulation of more complex patterns in terms of those previously learned, through intermediate (abstract) representations (Rocki, 2016). This idea is coherent with human information processing mechanisms, regarding the processes of vision and audition.

These complex information processing systems indicate the need of deep architectures, in order the brain to manage the vast amount of information, by decomposing sensory stimuli, and accordingly extracting complex structures and constructing internal representations (Deng & Yu, 2014). This kind of architecture allows the function of complex cognitive processes - skills (e.g., observation, analysis, learning, decision making, etc.) These factors contributed to the creation of the deep learning approach, where the main point is the modeling of real world data in an hierarchical fashion, emulating the deep layered learning process and attributes of the primary sensory areas in the neocortex (Najafabadi et al., 2015).

### Brain and Neocortex

The roots of deep learning architectures are based in the science of the brain. One of the fundamental features of the brain is associated with its ability of performing parallel and distributed processing. The brain is an inherently parallel system, that is composed of multiple subsystems (modules), where there is no explicit distinction between

memory and processing functions, which are performed together, simultaneously and uninterruptedly (Protopapas, 2004; Kandel et al., 2006).

Basic findings from neuroscience have provided insight into the principles governing information representation in the mammalian brain, formulating the hypothesis that mental (cognitive) activity consists primarily in the electrochemical activity in networks of neurons. Anatomical studies in conjunction with functional and neuroimaging techniques show that the mammal brain is organized in a deep architecture, where each input stimulus is represented at multiple levels of abstraction, and each level corresponding to a different area of the cortex (Serre et al., 2007; Bengio, 2009; Bengio & LeCun, 2009). The brain handles huge amounts of information through multiple stages of transformation. Each of these stages corresponds to different (intermediate) representations, which are considered to be sparse and distributed. These distributed representations are important in order to untangle underlying causes of variation, and they can be significantly more noise resistant, comparing to dense representations (Ahmad & Hawkins, 2015), as well as much more expressive (Rocki, 2016).

A specialized and extremely complex brain area is the neocortex, that is found only in mammals and is believed to be the place from which the intelligence emerges. Neocortex, which is associated with many cognitive abilities, is layered and hierarchical. There is not universal convergence and unanimity in the scientific community regarding the general principles of its operation, due to the increased complexity of the brain cells network (Rocki, 2016). Neocortex handles functions such as sensing, motor control, as well as higher-level cognitive functions, such as attention, planning, navigation, execution etc. (Palm, 2012). This particular brain structure does not explicitly pre-process sensory signals, but defines the structure and the conditions for the propagation of those signals through a complex hierarchy of modules, that progressively learn to represent observations based on the regularities they exhibit (Lee

et al., 1998; Lee & Mumford, 2003). Extensive studies on neocortex reveal the presence of multiple networks, where the organization in columns reflects the local connectivity of the cerebral cortex. These findings demonstrate that all the parts of neocortex are involved and interact each other, constituting a network of parallel and distributed processing (Rogers & McClelland, 2004; Rocki, 2016).

The neocortex of human brain is a thin, extended, convoluted sheet of tissue with a surface area of about 2600 $cm^2$, and thickness 3-4 mm. It contains up to 28 x $10^9$ neurons. Cortical neurons are connected with each other and with cells in other parts of the brain by a vast number of synapses. The cortex is organized horizontally into six layers, where lower layers projecting to higher layers and higher layers projecting back to lower layers (Mountcastle, 1997; Kandel et al., 2006; ). The structure of neocortex is sparse and distributed, allowing the optimal information processing of the input stream. Neocortex cells are grouped into clusters of cells, depending on their functionality, and they act as feature detectors. A feature detector includes those cells that exhibit similar behavior, by being simultaneously active or inactive. For example, the cells belonging to each column in the neocortex that respond to sensory stimuli, regarding the representation of a certain muscle or a region of vision or sound or even a region of another modality, are excited simultaneously (Rocki, 2016). This functionality constitutes a lateral inhibition mechanism, according to which a column that is active will prohibit other nearby columns from becoming active simultaneously. This mechanism, that leads to sparse activity patterns, allows the existence of independent feature detectors in the cortex (Bell & Sejnowski, 1997).

The most representative example, regarding to the hierarchical nature of the brain, came from basic neuroscience work in visual cortex. In their research study, Hubel and Wiesel (Hubel & Wiesel, 1962) recorded simple and complex cells in the primary visual cortex, suggesting a model according to which simple, complex and hyper-complex cells constitute a series of layers of units supporting the construction of

complex features, while preserving high degree of invariance (Kriegeskorte, 2015). The mechanisms of the primate visual system (visual pathway) propagate the incoming information in a sequence of processing stages, corresponding to detection of edges, primitive shapes, that gradually lead to more complex visual shapes (Bengio, 2009).

Significant discoveries from the fields of visual neuroscience, neuroimaging and neurophysiology suggest that the hierarchical nature of visual perception is based on the division of the connections into three different types, namely feedforward, lateral, and feedback connections. This hierarchy is based on the fact that several neurons are separated from the input by a large number of synapses. This kind of structure allows for the creation of more complex representations of the input features (Levine, 2013; Kriegeskorte, 2015). In particular, as information flows from the primary visual area V1 to the extrastriate areas V2 and V4, as well to the inferior temporal cortex (IT cortex), the neurons respond gradually to increasingly abstract features, while preserving increased invariances to several aspects of the perceived image, such as rotation, viewpoint, lighting, background, etc. (Bar et al., 2001; Kandel et al., 2006).

In conclusion, the structure and functionality of neocortex consists of the following essential features (Palm, 2012):

- It is a highly repetitive structure built of a vast number of nearly identical columns (Mountcastle, 1997). This complex and interconnected brain structure is layered and hierarchical such that each layer is capable of learning more abstract concepts (Felleman & Van Essen, 1991).
- A major feature of neocortex relates to the capacity of plasticity. Actually, neocortex is plastic to such an extent that one neocortical area can take over another areas function (Buonomano & Merzenich, 1998).
- Several high level cognitive skills, related to the neocortex (e.g., sensory perception, spatial reasoning, language, motor control, etc.), constitute

procedures that are acquired in an evolutionarily manner, during the early stages of development (Palm, 2012).

According to the above attributes, some theories, most of them arising from the domain of neuroscience, suggest that the neocortex is built using a general purpose learning algorithm and a general purpose architecture, capable of solving many different and complicated tasks. This general purpose architecture seems to be adequate to explain intelligence, and therefore how consciousness emerges (Palm, 2012; Rocki, 2016), and at the same time it constitutes an argument in favor of the view that there are not specific-region algorithms. This theory is based on the assumption that the complexity of the neocortex is limited to a single learning module, while the expectation is that if we can understand and decode this module, we can replicate and apply it in a scale that leads to reliable and effective artificial intelligence.

Deep learning theory considers that much of the power of neocortex results from this deep and hierarchical structure. Consequently, any research efforts and implementations in the deep learning domain focus on these issues and attempt to unlock these powers by examining the effects of layered and hierarchical structures (Palm, 2012). The idea of a layered hierarchical computational brain model has emerged early in the scientific community within the fields of cognitive psychology and cognitive science. The Deep Learning field borrows a lot from these ideas and can be seen as trying to implement some of these ideas (Bengio, 2009).

### Reverse Engineering

The neural perspective on deep learning, which refers to the use of deep neural network architecture of multiple non-linear processing layers, is motivated by two general assumptions.

The first one is related to the understanding of the basic computational principles behind the brain. Having as a starting point the hypothesis that the brain seems to work in a deep learning mode, a conceptually realistic way to create artificial intelligence is via the reverse engineering perspective. Reverse engineering refers in general to the process of analyzing a subject system, in order to identify the system's components and their interrelationships. This approach gradually leads to the depiction - transformation of the system's representations in another form and/or at a higher level of abstraction (Chikofsky & Cross 1990). This is achieved by breaking up the potential solutions into multiple levels of abstraction and processing (Bengio & LeCun, 2009). The reasoning used here is about implementing the practice of decomposing a system in order to gain understanding about its function (Mikowski, 2013).

The second assumption refers to the promotion of the scientific questions that are related to the basic principles governing the structure and functioning of the brain and resulting cognition, contributing in this way greatly in many scientific fields (Goodfellow et al., 2016).

While the field of neuroscience has contributed greatly and in many ways to the emergence of deep learning and is regarded as a valuable source of information for the basic and applied scientific principles that govern the field, the modern era of deep learning exceeds the neuroscientific perspective, in favor of the recent generation of machine learning models (Poggio, 2016; Goodfellow et al., 2016). Besides the neural-based models, deep learning can also be implemented in different machine learning frameworks, regarding to a more general principle of learning multiple levels of composition and abstraction. Although there has been notable progress in the study of the brain, there is still lack of knowledge, on how to replicate proportionally the brain processes (algorithmic learning processes) in the field of machine learning. The reasons for this lack of progress in the study of the brain, are related mainly with technical issues, regarding the methods used for the imprinting (simulation) of the operation of

thousands (millions) of interconnected neurons simultaneously. TThe spatial and time localization restrictions of neuroimaging techniques, as for example the Functional Magnetic Resonance Imaging (fMRI), combined with the required computing power, that is necessary for conducting large-scale simulations of the brain function in real time, gradually shifted the interest of research in areas beyond the field of neurobiology. In other words, we don't have as this moment enough information about the brain, in order to use it as a main guide for exploring the deep learning perspective (Goodfellow et al., 2016).

Consistent with the above principles and findings, one can approach the deep learning in the sense of simulating the brain functions, by constructing more accurate models of how the brain actually/potentially works. This is an extremely interesting and promising challenge, which relates inherently to the scientific fields that study the brain and cognition related issues. The researchers that study the brain, as for example, psychologists, cognitive scientists, neurologists, neuroscientists, tend to use the deep learning methods and tools as the medium for their research.

In the case of computer science and in particular of artificial intelligence, this is not the objective. The main goal of deep learning is related with how to build computer systems, which manage to solve successfully complex and demanding tasks-projects, that require skills akin to these of human intelligence. Applied math fundamentals like information theory, linear algebra, probability theory, and numerical optimization inspire in a catalytic way the deep learning progress (Goodfellow et al., 2016). Furthermore, over the last years the increasing computing power and the availability of labeled large datasets, have contributed greatly to the development and performance of deep learning systems (Poggio, 2016).

## Cognitive Science

Cognitive science is the scientific interdisciplinary field which study the mind. This multidimensional perspective is the result of the intersection of many different scientific fields, including psychology, philosophy, artificial intelligence, neuroscience and linguistics among others. This wide range of fields makes it possible to formulate theoretical directions and ideas beyond the scope of each field individually. Consequently, cognitive science is not limited to the sum of the individual constituent components, but to their intersection or converging work on specific problems.

A fundamental principle of cognitive science is the idea of computation, according to which the mind is an information processor. This theoretical perspective (i.e., information processing) on what the mind is, assumes some form of mental representations and processes, on the basis of which information is processed. The concept of representation is central to almost all the theoretical approaches of cognitive science (Protopapas, 2004; Friedenberg & Silverman, 2006). The approach in which the concept of representation is not used, is that of the dynamic approach, which employs-implements dynamic systems.

### The Interdisciplinary Perspective: Methodological Approaches

The interdisciplinary perspective of cognitive science, is based on the co-existence of several individual disciplines. The different approaches that make up the cognitive field are the following (Friedenberg & Silverman, 2006):

- **The Philosophical Approach**: This branch of cognitive science does not contribute descriptions of observations or models of experimental results, but a priori arguments for the structure and function of the mind. The consequence of the philosophical approach to thinking of the mind as an information processor is the conception of consciousness as a symbolic processing (Protopapas, 2004). The primary method of philosophical inquiry is reasoning (deductive and

inductive reasoning). According to the first one, given an initial set of statements assumed to be true, some statements are extracted that logically must be correct. Correspondingly, according to inductive reasoning, some observations are made about specific instances in the world, some commonalities among them are extracted, which in turn lead to conclusions being drawn (Friedenberg & Silverman, 2006).

- **The Psychological Approach**: The method that psychologists use is based on the experimental method, according to which a working hypothesis is formulated about how internal mental phenomena and accordingly external behaviors arise, which is confirmed or not in the context of an experiment under a set of controlled conditions. The field of psychology includes several subdisciplines (e.h., functionalism, Gestaltists, Psychoanalytic psychology, etc.) (Friedenberg & Silverman, 2006).
- **The Cognitive Approach**: The emphasis is shifted to internal mental operations-processes. Mental functionality is described in terms of the information processing (i.e., representation and computation). The basic tools used are the experimental and computational modeling. Computational modeling methods use software based implementations, in order to describe several cognitive processes as memory, attention, pattern recognition, etc. This modeling method is also used in the artificial intelligence and network approaches (Friedenberg & Silverman, 2006).
- **The Neuroscience Approach**: Cognitive neuroscience studies the mental events in terms of underlying brain mechanisms (i.e., brain and endocrine system), providing a description of the cognitive/mental processes at the implementation level. At this level of analysis, the subject of study is the cell biology of individual neurons and of neuron-to-neuron synaptic transmission, the patterns of activity in local cell populations, and the interrelations of larger

brain areas. The great evolution that has been made recently in neuroimaging techniques (e.g., Positron Emission Tomography - PET, Computerized Axial Tomography - CAT, and functional Magnetic Resonance Imaging - fMRI) has contributed significantly in measuring the brain activity concurrently with the performance on a given task (e.g., pattern recognition, attention, memory, imagery, and problem solving) (Kandel et al., 2006; Friedenberg & Silverman, 2006).

- **The Network Approach**: A key building block of this approach is the collection of individual computing units, which are connected to each other and they form a network of interdependence and interaction between them (Friedenberg & Silverman, 2006).
- **The Evolutionary Approach**: In this perspective, the basic principles of selection theory are used to analyze and interpret the various cognitive processes. An example in this direction is the evolutionary psychology (Friedenberg & Silverman, 2006).
- **The Linguistic Approach**: In this perspective, the interest lies in linking the symbolic philosophy of mind and the classical theoretical linguistics, as expressed in Chomsky's theory. In both cases it is assumed a priori that the system (cognition and language) is a formalistic computational system, while any variations relate to the exact type of formalistic system or details of its operation (Protopapas, 2004). The methods used are the experiments as well as the construction of computational models (Friedenberg & Silverman, 2006).
- **The Artificial Intelligence Approach**: Artificial Intelligence (AI) is the field of computer science dealing with the design and implementation of programs that are capable of mimicking human cognitive abilities, thus displaying features that we usually attribute to human behavior. Such abilities are problem solving, vision through learning, learning and memory, drawing conclusions,

understanding natural language (NLP), etc. (Vlahavas et al., 2006). In this direction AI gives insights into the function of human mental operations (Friedenberg & Silverman, 2006).

- **The Robotics Approach**: The fields of robotics, brain and cognitive science respectively set up an interaction framework that is analyzed into three main streams of research: cognition, action and perception. The basic principles here are associated with learning and development, as well as the dynamics of knowledge acquisition in the framework of goal directed actions (Friedenberg & Silverman, 2006).

## Theoretical Approaches to Cognitive Science: The Classical & Connectionist View of Computation

The computational tradition of mathematical logic is expressed in cognitive science with the symbolic approach. Under this scope, symbolic representations participate into algorithmic syntactic processes, regardless of their content. The tradition of association and empiricism is expressed through the connectionist approach. Connectionism is based on the basic principles of Parallel Distributed Processing (PDP) (Russell & Norvig, 2010), and on flexible learning on networks of simple interconnected nodes. The dynamic approach incorporates the time dimension, introducing a continuous non-representational view based on mathematical formalism (Protopapas, 2004).

**The Classical (symbolic) Cognitive Science**: The essence of the hypothesis that the mind is a computer concerns the syntactic properties of symbols' representations, according to which to which these symbols interact and participate in several processes. Their semantic content is the product of the observer' s interpretation of the system that has access to the external environment of the mind and the relationship between the mind and its environment (Protopapas, 2004). Within this

view, knowledge is represented locally, in the form of symbols, while processing occurs in discrete stages (Friedenberg & Silverman, 2006).

**The Connectionist Cognitive Science**: Another approach, alternative to the symbolic one, is to study the mind by using models based on the brain architecture. Within this concept, some functional properties of natural neural networks are preserved, which are considered as important for the computational properties of the mind. These models are called artificial neural networks (ANN) or connectionist models (Protopapas, 2004). The operation of these models in the context of cognitive science focuses on understanding the cognition and relative cognitive processes, and not so much on the description of neuronal function (Protopapas, 2004).

In the view of connectionist approach, knowledge is not represented as in the traditional symbolic processing systems, where processing and storage are distinct functions. In contrast, knowledge is represented as a pattern of activation, which results from changes in the weights of the connections, that is distributed throughout a network. In connectionism processing occurs in parallel through the simultaneous activation of nodes (Friedenberg & Silverman, 2006).

## Learning Deep Architectures

The deep learning approach is differentiated significantly comparing to the previous generation of machine learning methods (e.g., shallow structured architectures), both in terms of structure and computational efficiency. In the deep learning framework, the features are distributed in different layers and are separated between blocks, stacking up the non-linear transformation layers. This spatial arrangement recommends an hierarchical architecture, and allows the construction of abstractions and complex representations from a huge amount of unsupervised data in an automated schema (Deng, 2013; Najafabadi et al., 2015).

# Deep Learning: A General Purpose Learning Procedure

## Shallow vs Deep Structured Architectures

Shallow structured architectures are typically consisted of just a few layers of nonlinear feature transformations. Systems based on shallow architectures have limited capabilities regarding modeling and representational power (Deng & Yu, 2014). applications involving natural signals has been reported to be problematic when applied on shallow architectures and can be inefficient (Bengio & LeCun, 2009; Deng & Yu, 2014; SAP SE, 2015). The hierarchical multi-level approach of deep learning mdthods achieve greater efficiency at extracting non-local and global relationships and patterns on the data, compared to shallow network architectures (Bengio & LeCun, 2009; Najafabadi et al., 2015).

The main distinction between the deep approach and other machine learning methods, related to unsufficiently deep architectures, is referred to the processes that take place in the hidden layers (nodes), the units where the internal representations are formed. Shallow classifiers require for their operation a suitable feature extractor. The suitability of feature extractors is defined as the capacity of forming representations, which are selected depending on the input stream-data (e.g., images), and therefore are important for discrimination processes. In addition, the produced representations must be invariant concerning the irrelevant aspects of the input stimuli (LeCun et al., 2015).

The procedure of choosing the right features for a given task is an extremely complex and sophisticated process. This problem becomes even more difficult, when the case concerns complicated problems and high dimensional data (Gudi, 2014). Conventional methods for discovering abstractions, namely higher-level concepts from lower level features, requires to ensure a large set of relevant hand labeled examples, and at the same time to define manually all the necessary abstractions (Bengio, 2009). The option of hand designing efficient and reliable feature extractors, requires a

considerable amount of domain expertise and advanced engineering skills (LeCun et al., 2015). Furthermore, defining manually the right features for a given task, implies interdependence relationships among the proposed solutions and the user's domain knowledge, while this process needs to be repeated for almost every different task (Gudi, 2014).

All these handcrafted processes can be avoided using a general purpose learning procedure, that exhibits similar characteristics to those of neocortex (Arel & Rose, 2010), focusing on learning hierarchical models of data (Palm, 2012). This is the objective in a deep learning framework, where the construction of feature detectors is accomplished using unsupervised learning procedures, without requiring labelled data (LeCun et al., 2015). The interest in this case is shifted to procedures and strategies, according to which we try to discover the structure in the data through a general purpose learning algorithm (Bengio et al., 2012; LeCun et al., 2015).

### Deep Architectures and automated feature learning

Deep learning methods are closely related to representation learning, where the objective is to discover the representations needed for classification or detection tasks based on unlabelled (raw) data, that are available in vast quantities (Palm, 2012). Deep learning algorithms are based on a hierarchical multi-level learning scheme, according to which a deep learning based solution is intuitive and principle-oriented. In that context, multiple lower level concepts, namely less abstract concepts and representations in the lower levels of network's hierarchy, compose progressively groups of concepts, that in turn lead to higher level concepts (i.e., more complex representations). Finally, these abstract concepts compose a complex representation, which is a highly non-linear function of the input data (Gudi, 2014; Deng & Yu, 2014, Najafabadi et al., 2015).

In a deep learning architecture, the extracted features are defined by the network itself (Gudi, 2014). To achieve this, each layer is pre-trained with an unsupervised learning algorithm, namely a nonlinear transformation of its input or the output of the previous layer, and accordingly provides a representation in its output. These processes are performed autonomously through a series of processing stages in consecutive layers (Hinton & Salakhutdinov, 2006; Najafabadi et al., 2015), deriving high-level representations directly from the raw data (Gudi, 2014). This automated way of learning features makes it possible to learn very complex functions, through the mapping process of the input data to the output directly from the data (LeCun et al., 2015; SAP SE, 2015). The benefit deriving, inter alia, from the use of these techniques, is that human intervention and mediation is minimized (Bengio, 2009).

The basic architecture and operating principles of a typical deep learning network are summarized as follows (Hinton & Osindero, 2006; Najafabadi et al., 2015):

- The input (raw) data is fed to the first layer.
- The first layer is then trained based on this data.
- The output from the first layer, which constitute the first level of transformed representation, is provided as input stream to the second layer.
- This process continues until the required number of iterations is reached, namely when the desired number of the consecutive layers is obtained. Each layer computes a non-linear transformation of the previous layer (Gudi, 2014), and propagates the extracted features of each layer to the next one, until the final output of the network is achieved (SAP SE, 2015). What is important here, is the reuse of the output of each layer of the network as input to the next layer. Learning each layer of feature detectors corresponds to modeling the activities of feature detectors in the layer below (Hinton & Osindero, 2006).
- This structure involves multiple levels of abstractions, where lower-level abstractions tend to be more related to segments of data, while higher-level

abstractions are more directly tied to meaningful concepts (SAP SE, 2015), yielding increasingly abstract associations between these concepts (Rocki, 2016).

Deep learning, as a branch of machine learning, refers to the algorithms that model complex patterns. The operation of these algorithms is based on the data that feed the multiple non-linear transformations, aiming each level to capture a different level of abstraction (Schmidhuber, 2014). Automatic learning of features in data, through this iterative reconstruction procedure of pre-training several layers, is a key aspect of deep learning, that leads progressively to more complex detectors, with higher representational power, contributing to the optimal initialization of networks weights. Under this perspective, a system is able perform complicated functions, that are high sensitive to details and at the same time insensitive to large and irrelevant variations (LeCun et al., 2015).

In conclusion, the ability of a network to learn more complex functions depends mainly on its depth. This automated method gives satisfactory solutions to many of the above mentioned drawbacks of shallow modelling techniques. The utility of learning features in this way (i.e., automatically) is extremely useful, especially in cases where the amount of data continue to grow (Bengio, 2009).

## Advantages of Deep Learning Methods

Deep learning is one of the basic methodologies that are widely used in the context of machine learning. Methods and algorithms based on deep learning techniques have been proved to be extremely successful in solving complex problems. Deep learning models are widely applied in the domains of computer vision, natural language processing, recommendation systems, biomedical informatics, etc. The superiority of deep techniques is primarily due to three key components, which are related to

efficiency, precision and flexibility. These properties are illustrated in the following principles (Bengio et al., 2006; SAP SE, 2015):

- Deep learning systems are capable of learning complex and highly varying functions.
- They analyze low, intermediate, and high level abstractions, with minimal human mediation.
- They process a very large set of examples, using mostly unlabeled data.
- They also exploit the synergies that take place across a wide number of tasks (Bengio et al., 2006).

### Why depth matters

One of the key features of deep learning networks is their ability to represent efficiently complex functions. This comparative advantage is a direct result of the basic structure and architecture of deep networks. This special structure is consisting of fewer units and weights, comparing to other machine learning methods as for example the shallow neural networks and the support vector machines (SVM) (Bengio, 2009). Algorithms based on deep learning architectures demonstrate state of the art performance on many machine learning tasks, as they exploit the inherent structure of the target function (Deng & Yu, 2014; Kriegeskorte, 2015). The use of a general purpose learning procedure (i.e., models consisting of multiple stages of nonlinear information processing units), is related to the fact that each module in the stack transforms its input to increase both the selectivity and the invariance of the representation (LeCun et al., 2015).

A significant difference between a shallow neural network with a single hidden layer and a deeper network respectively, concerns the reuse of computations within the network, namely using the output of each layer as input to the next layer, in a bottom-up fashion (feedforward neural networks). Every node in a shallow three-layer network

receives input from the upstream nodes and delivers output to downstream nodes. Respectively, maintaining the same number of units in a deeper network, the last node is able to compute any function the shallow network can compute. In a network with several hidden layers, the units of the single layer are distributed across multiple hidden layers. According to this configuration, the deep network could have the same connectivity from the input to the hidden units, and respectively from the hidden units to the output.

Under this perspective, the two networks seem equivalent. However, the reverse is not true. The deep network is permitted additional non zero weights from any given layer to higher layers, enabling the reuse of the results of previous computations, and therefore extending the deep network' s expressive power (Kriegeskorte, 2015).

## Popularity of Deep Learning

The term deep learning is relatively recent, and is extremely popular both in academia and in many sectors of industry. The popularity of deep learning is due to the state of the art performance on numerous machine learning tasks. Nevertheless, the basic characteristics of deep learning appeared relatively early in the historical trends of the machine learning engineering community, as well as in other related disciplines. The underlying mathematics as well several key points of deep learning theory, regarding the computation of hierarchical representations of the input data, have been known to the scientific community for many decades.

Many of the ideas and techniques that have contributed to the development of neural networks (shallow or deeper networks), as for example the parallel distributed processing theory (PDP), distributed representations and the backpropagation algorithm, arose in the context of cognitive science and artificial intelligence.

## Breakthrough Technologies

Deep learning models have grown in size over time, as computer hardware and software infrastructure for deep learning techniques has improved (Goodfellow et al., 2016). Consequently, the impressive achievements and progress that has been observed during the last decades in the deep learning field, are to a very large extent a consequence of much faster hardware, large amounts of data, as well improvements in processing methods.

The optimal performance of a neural network is primarily based on the training process-phase. A basic prerequisite for the successful training of a neural model is the availability of large amount of data. The availability of large volumes of exploitable data is a major issue with regard to deep learning. Several limitations regarding data collection and storage technologies did not allow previously the exploit of large datasets. But even when data became available in large quantities, other problems arose regarding algorithmic and computational issues.

The revival of neural networks is largely due to the available computational resources and the recent technological achievements. Progress in these areas made it possible to run much larger models. This progress is driven by faster computer systems, capable of performing calculations much faster and efficiently on large available datasets (Gudi, 2014). The aforementioned parameters allowed the construction of large-scale networks, which achieve better and more accurate results in more complex and demanding tasks (Goodfellow et al., 2016).

The success of deep learning models is also due to the increased chip processing abilities of general-purpose powerful graphical processing units (GPUs) (Deng & Yu, 2014; Gudi, 2014). The highly parallel structure of GPUs makes them more efficient than general purpose CPUs for calculating complex algorithms, where the processing of large blocks of data is done in parallel. Deep networks are inherently parallelizable and fit well in a parallel computational architecture. The mainstream methods used for

training deep learning models is via platforms using CPUs and multi-thread GPUs. This powerful combination is a key aspect for achieving state of the art performance on numerous machine learning tasks (Gudi, 2014), exploiting successfully large volumes of data (Deng & Yu, 2014; Goodfellow et al., 2016).

Other factors that undoubtedly contributed to the evolution of deep learning are the general progress in other methods and techniques of machine learning, as well as achievements of the signal-information processing research.

Deep Learning can be conceptualized as an optimization problem, where the objective is finding a function of the model error. Deep learning networks, trained with large datasets, surpass overfitting issues and show a testing error similar to the training error (Poggio, 2016).

Summarizing all those previously mentioned, i.e. the increase in model size over time due to the availability of faster CPUs and the advent of general purpose GPUs, the faster network connectivity and the better software infrastructure for parallel and distributed computing (Goodfellow et al., 2016), the use of deep learning systems is a new gold standard both in informatics research and industry. Systems based on deep learning architectures are capable of exploiting complex nonlinear functions, making effective use of both labeled and unlabeled data, in order to learn distributed and hierarchical feature representations (Deng, 2014; Goodfellow et al., 2016).

## Distributed representations

### The Origins of Distributed Representations

One of the key concepts of deep learning is that of distributed representation (Hinton, 1986). This type of representation sets the core of connectionist view, where knowledge is represented as a pattern of activation or weights that is distributed throughout a network (Friedenberg & Silverman, 2006). Connectionism is an approach to the study

of the mind, using models based on the architecture of the brain. This conception maintains some functional properties of natural neural networks, which are considered essential for the computational properties of the mind. The models used in this context are called artificial neural networks or connectionist models (Protopapas, 2004).

Distributed representations are fundamental to deep learning models. The term distributed representation, as expressed and introduced by Hinton in 1986, refers to the idea, according to which each input to a system should be represented by many features, and each feature should be involved in the representation of many possible inputs (Hinton, 1986; Goodfellow et al., 2016). Concepts can be represented by distributed patterns of activity in networks of neuron like units. The advantage resulting from such a structured architecture, is that it leads to automatic generalization.

### Integrating Distributed Representations into Deep Learning Systems

During the training phase of a network, the weights are changed in order to incorporate new knowledge about one or more concepts. Accordingly, these changes affect the knowledge associated with other concepts that are represented by similar activity patterns (Hinton, 1986). This type of representations that have these characteristic properties are called distributed representations, because their elements are not mutually exclusive and their many configurations correspond to the variations seen in the observed data (LeCun et al., 2015).

Distributed representations recognize patterns in data. This recognition process is based on training the networks in a supervised or unsupervised way, where the input pattern is represented by a set of features that are not mutually exclusive, and might even be statistically independent (Bengio, 2009; Gudi, 2014). Computational architectures integrating distributed representations make possible the transformation of the input data into abstract features, that are automatically captured and compactly represented across the hidden layers of the network (SAP SE, 2015). The compact

representation of each sample, through a large number of possible configurations of the abstract features, leads to enriched representations, where the number of possible configurations is exponentially related to the number of extracted abstract features (Najafabadi et al., 2015).

Data are generated through interactions processes among several factors, some of which are known, while several others are unknown. These interactions enable new combinations of the values of the learnt features beyond those seen during training (LeCun et al., 2015). What is achieved through these internal processes, is that when a data pattern is obtained through configurations of the learnt factors, additional data patterns, which previously were not known, can likely be described through new configurations of the already learnt factors and patterns (Bengio, 2009; Bengio et al., 2012).

Deep networks are efficient computational models, achieving state of the art performance on numerous machine learning tasks (Hinton & Osindero, 2006; Vincent et al., 2010). This is mainly due the multi-level architecture of these networks where the information to be processed is not located in just a single layer of the network, but it is distributed across multiple layers, allowing learning intermediate representations. Intermediate (abstract) representations, can be invariant to the local changes of the input data, separating the different sources of variations observed in data (Najafabadi et al., 2015) and produce generalizations that are based on different levels of abstraction, focusing on a small subset of a large number of features. One of the most important benefits of incorporating intermediate representations, is the fact that these representations can be shared across different problem areas. Knowledge resulting from this type of abstract learning is reusable, since new high-level features can be learned by combining lower-level intermediate features from a common pool of information (SAP SE, 2015).

## Big Data and Deep Learning

Big data, also known as massive data, is referred to the exponential growth and wide availability of digital data, that are difficult to be managed and analyzed using conventional software tools and technologies, such as the traditional machine learning techniques, which are based mainly on shallow structured architectures. The collection of these massive amounts of unlabeled data, often in the form of real-time streams, is coming from many different sources and is growing at astonishing rates, both in shape and size (Chen & Lin, 2014). The increasing data volume, which defines the scope of big data, is mainly due to powerful data collection sensors, sophisticated digitalization techniques and increased storage capabilities, in the form of large datasets, high dimensionality and complex data formats (Hammer et al., 2014).

Deep learning methods exploit massive amounts of unlabeled data (i.e., extraction of abstract representations from the raw data by using a hierarchical multi-level learning approach), which is the objective in the big data analytics era.

### Big Data Analytics

Big data analytics define a number of challenges related to computational and technical restrictions (e.g., limits of the typical storage hardware, processing and computing capacity of traditional data analysis techniques). (Elaraby et al., 2016). The amount of data collected on a daily basis is increasing rapidly, posing challenges for the machine learning community.

Given the above, big data analytics provide the appropriate tools and methods in order to analyze and extract hidden correlations and complex patterns from large scale data (Najafabadi et al., 2015; Elaraby et al., 2016).

The conventional machine learning and feature engineering algorithms have not been designed to handle massive amounts of raw data that are usually observed in big

data problems. Furthermore, it is not considered as trivial the fact that the big data systems incorporate inherent complexities (Najafabadi et al., 2015).

## Deep Learning: An Effective Big Data Analytic Tool

Big data involve data analysis from very large collections of data. Shallow structured architectures, consisting often just a single layer of hidden units, transform the input information into a structured feature space. This relatively simple structure is effective for well-constrained problems (Elaraby et al., 2016). Big data problems require specific processes, that provide automated extraction of complex data.

Deep learning networks, composed of feature detector units organized in multi layers (Lee et al., 2009), is a valuable tool for big data analytics. Systems based on deep learning architectures extract complex abstractions providing high-level data representations from unlabeled data. This class of problems is summarized on extracting hidden patterns from massive volumes of data, fast information retrieval, data indexing and tagging, and simplifying discriminative tasks (Elaraby et al., 2016).

Deep learning algorithms exhibit remarkable performance compared to traditional shallow learning architectures at extracting global patterns and relationships in the data. The multilevel structure at the hidden layers of deep learning systems, often resulting in thousands or even millions of free parameters, achieves efficient solutions (Bengio & LeCun, 2007). An important advantage of deep learning, is the capability of generalization in non-local and global ways, generating learning patterns and relationships beyond immediate neighbors in the data.

Computer industry giants like Apple, IBM, Microsoft, Google, Facebook, etc., in order to deal with huge amounts of data on a daily base, have focused on the research and development of deep learning technologies. Examples in this direction are Apple's Siri virtual personal assistant, IBM's brain-like computer (IBM Watson), Microsoft's

real-time translation (Microsoft Translator Speech API), as well as many other artificial intelligence projects and applications (Chen & Lin, 2014).

## Characteristics of Big Data: The Four V's

Big data analytics and deep learning are interrelated domains that relate on issues such as large scale models, heterogeneity, noisy labels, and non-stationary distribution, among many others. Big data analytic tools focus on the following (Najafabadi et al., 2015; Elaraby et al., 2016):

- Address issues related to volume and variety of large amounts of labeled or unlabeled data.
- Handle its incompleteness and noise properties, providing high level representations.
- Learn factors of data variation and make effective integration of different data formats.
- Provide online learning for handling fast coming of data streams.

The basic challenges posed by big data, can be described by the three V's model, introduced by Doug Laney (Laney, 2001). The three V's refer to volume (i.e., large scale of data), variety (i.e., different types of data) and velocity (i.e., speed of streaming data) (Katal et al., 2013; Chen & Lin, 2014). In addition, the three V's model has been extended to the four V's model, taking also into account the dimension of veracity (Hammer et al., 2014; Najafabadi et al., 2015).

### Deep Learning From High Volumes of Data

Data volume refers to the size of datasets and it depends on the number of data points, its dimensionality, or both (Hammer et al., 2014). High volumes of data pose high challenges to deep learning systems. There are two conditions to be met, according to

which the data size that is being processed cannot be limited, while the speed of processing remains constant (Elaraby et al., 2016). The properties that define the field of big data impose a special kind of architecture, according to which the input is consisted by a large number of examples and the output generates large varieties of class types. According to these properties they arise multiple attributes and high dimensionality, increasing geometrically the running-time complexity and model complexity (Chen & Lin, 2014).

Analyzing and processing big amounts of data involves new resources. The use of large amounts of data complicates the process of training a deep learning algorithm, using only a central processor. Algorithms training optimization is accomplished via distributed frameworks with parallelized data machines. The method that is extensively used to optimize training processes, maintaining at the same time the accuracy stable, is through utilizing clusters of CPU and/or GPU (Chen & Lin, 2014).

High volumes of data, often contain incomplete data and noisy labels (Katal et al., 2013; Chen & Lin, 2014). In fact, a large percentage of the data to be processed may not be labeled at all. Deep learning algorithms can inherently manage unlabeled data, by learning data distribution during training process without using label information. In addition, deep learning methods can also deal with noisy data, showing high tolerance to data messiness (Chen & Lin, 2014).

### Deep Learning For High Variety of Data

Data variety refers to heterogeneous data formats, collected by multiple sensors, derived from distributed data sources, different representation methods and different technologies (Hammer et al., 2014). Nowadays, data production is continuous and uninterrupted, contributing to the creation of many different data types. This results in increasingly diverse and complex data formats, coming from a variety of sources, presumably with different distributions (Chen & Lin, 2014 ; Elaraby et al., 2016). Deep

learning techniques combine data that show irregularities and dissimilarities in their source or structure, addressing successfully the several challenges (Elaraby et al., 2016).

But how can we actually deal with high variety of data, namely data coming from different sources with different distributions? The solution lies in integrating data through deep learning architectures. This integration process is achieved through learning data representations from each data source individually. As mentioned previously, one of the most important features of deep learning systems lies in the capability of representations learning. The discovery of intermediate representations in deep networks is achieved through unsupervised learning, maintaining multiple levels of abstraction, by extracting complex relationships among the data (Deng, 2013; Chen & Lin, 2014; LeCun et al., 2015).

Consequently, to address the integration data problem, we first have to learn data representations from each individual data source, and then to integrate the learned features at different levels in the network architecture (Chen & Lin, 2014).

### Deep Learning For High Velocity of Data

Data velocity refers to the speed of data accumulation. The production of data is an uninterrupted and continuous process, that poses high challenges for big data learning. A basic requirement for data processing of this type is their timely processing (Chen & Lin, 2014). While transfer rates can be limited, the same it' s not true for the requests, the number of which remains unlimited (Elaraby et al., 2016). These restrictions are pushing the traditional machine learning methods and systems to their limits.

A promising solution for managing this mass amount of information in such a high velocity, is online learning approaches. According to this type of learning, the system learns one instance at a time. As soon as the true label of each instance is available, it can be used for refining the model (Shwartz, 2012; Elaraby et al., 2016).

Stochastic gradient descent is a widely used method in the context of deep learning systems and can be successfully implemented to online learning as well. According to stochastic gradient descent method, one training example with the known label is used at a time to update the model parameters (Chen & Lin, 2014). In particular, the estimation of the true gradient is accomplished by taking small amount of samples and averaging them (Basegmez, 2014). This is a repetitive process, which is being conducted for many small sets of examples, drawing from the total training set until the average of the objective function stops decreasing (LeCun et al., 2015). An efficient way to accelerate the training process, instead of proceeding (processing) sequentially one example at a time, is to perform updates on a mini-batch basis (Scherer et al., 2010).

High velocity in data implies that data distribution is not stable but is changing over time. Non-stationary data tend to be separated into chunks, using data from a small time interval. The theoretical background behind this observation, is that data adjacent in time may be characterized by a high degree of correlation, and consequently follow the same distribution (Chien & Hsieh, 2013). The capability of handling the data as a stream, is a fundamental feature of deep learning algorithms (Chen & Lin, 2014).

### Deep Learning for High Veracity of Data

Data veracity refers to the biases, noise and abnormality in data, namely to the fact that data quality varies considerably for big data sources, making any intervention almost impossible (Hammer et al., 2014). Given the fact that veracity aspect deals with uncertain or imprecise data, the information that is derived from it is unreliable, and therefore not directly usable. Addressing veracity problems is achieved through the use of data quality strategies and tools as part of a big data infrastructure (Jewell et al., 2014).

In conclusion, we can say that the main problems related to big data analytics are the following (Chen & Lin, 2014):

- Performing parallel and distributed data processing.
- Implementing high dimensionality and data reduction (Hammer et al., 2014).
- Integrating heterogenous data.
- Tracking and analyzing data provenance in real time.
- Decision making (Chen & Lin, 2014).

## Deep Learning Challenges in Big Data Analytics

Deep learning is an important step toward big data analytics. This coexistence allows the extraction of distributed representations directly from unsupervised data, while keeping balanced the need for human intervention (Najafabadi et al., 2015). The basic properties of deep learning systems, which are crucial for big data analytics, are the following (Najafabadi et al., 2015):

- Deep learning systems inherently exploit the availability of massive amounts of data.
- The multiple intermediate-abstract representations enable the analysis of raw data presented in different formats and coming from different sources.
- The extraction from every new data type can minimize the need for input from human experts.

In order to exploit the potential of big data, many challenges need to be addressed, concerning the optimization of existing algorithms and the technical issues arising (Chen & Lin, 2014). Although there has been remarkable progress in the field of deep learning during the last years, there are still many things to be done with regard to big data. In particular, issues to be resolved are the following (Najafabadi et al., 2015):

- Handling streaming data of high dimensionality.

- Distributed/parallel computing (Elaraby et al., 2016).
- Data sampling for generating useful high-level abstractions.
- Domain adaptation, (i.e. suitable data distribution).
- Semi-supervised and active learning.

### Real-Time Non-Stationary Data

A basic feature of real-time non-stationary data regards the high rate of production, where the distribution of these data is varying over time.

In order to extract the optimal number of features in a dataset, most learning algorithms use the cross-validation method (Zhou et al., 2012). However, this is not the case for real-time non-stationary data. A proposed solution to address this problem is the online incremental learning (Chen & Lin, 2014; Najafabadi et al., 2015). Incremental learning is a widely used method in the field of machine learning, especially in those cases where the training samples become available one after another over time, in a constantly changing learning environment (Liu et al., 2008).

Incremental feature learning addresses successfully the problem of learning from large amounts of data. This is possible by starting with a slight set of initial features, and automatically adjusting the number of features as the training proceeds (Zhou et al., 2012). In that context, Zhou et al. (Zhou et al., 2012) propose an incremental feature learning algorithm, that is based on denoising autoencoders. The gist of this learning method lies on adding new features mappings to the existence feature set, and then merging them redundantly (Zhou et al., 2012; Elaraby et al., 2016). The progress made so far in deep learning algorithms on big data is important, but there is much more to be done in this direction (Elaraby et al., 2016).

### Data Parallelism

The complexity of a model is comprised of many parameters, including the large scale data input, high dimensionality attributes and great varieties of output (Najafabadi et al., 2015). Traditional computational systems, consisting of just a single central processor and a storage unit, are unable to support sufficiently complex models. The ever increasing complexity of computational models requires the adoption of a platform, consisting of distributed computational frameworks with parallelized machines.

Parallel implementation is based on the combination of multiple CPU's and GPU's in increasing speed of training, while maintaining the accuracy of learning algorithms (Dean et al., 2012; SAP SE, 2015). This parallel and distributed computational framework solve many of the problems associated with training deep machine learning algorithms.

### Multimodal Data

Big data analytics is directly related with multimodal learning, since the large amounts of data is usually collected from multi-modalities. Multimodal learning involves relating information from multiple sources, where each modality usually involves a different representational schema and alterative correlational structure (Elaraby et al., 2016).

Deep learning algorithms can handle heterogeneous data integration successfully, since they can learn variation factors in data and provide intermediate - abstract representations. However, the number of modalities that these algorithms can handle is limited. In fact, this number is limited to integrate data coming from just two modalities (Chen & Lin, 2014). Fusion and confliction are two additional problems that arise due to the high variability of the data. Systems based on deep learning architectures are unable to cope adequately with these problems (Elaraby et al., 2016).

The contribution of deep learning in the big data analytics domain is undoubtedly important. Although deep learning provides the required complex (distributed) representations for handling big data tasks, there are many more things to be done in order to fill existing needs and gaps. A key point of all these efforts is to find out how we can adapt the deep learning algorithms in order to support specific types of problems. These types of problems often involve, among others, streaming data analysis, high dimensionality, distributed computing, information retrieval, domain adaption etc. (Najafabadi et al., 2015).

## Building Blocks for Building Deep Learning Architectures

### Introduction

The concept of deep learning and artificial neural networks are interdependent. As mentioned previously, the origins and the basic ideas of deep learning spring up from the basic disciplines of neuroscience and cognitive science. In that context, the role and contribution of connectionism has undoubtedly been significant. In particular, the processing in connectionism occurs in parallel through the simultaneous activation of nodes at multiple levels of hierarchy, using intermediate (abstract) representations. In this section, we will focus on two basic neural network structures that define the building blocks for building deep neural networks architectures.

Neural networks use a variety of topologies and learning algorithms, depending on the requirements and specifications of the problem to be solved. Two of the most widespread and predominant architectural structures are the probabilistic and the direct encoding models. A typical example of the first case is the Restricted Boltzmann Machine (RBM), while Autoencoder (AE) is a representative example of direct encoding models. Both RBM and AE are unsupervised single layer learning algorithms, which are used to build deeper models (Najafabadi et al., 2015; Elaraby et al., 2016).

### Auto-Encoders (AEs)

Auto-encoders (AEs), also called autoassociators, are feedforward neural networks constructed of three layers (i.e., input, hidden and output layer), that copy its input to its output (Basegmez, 2014; Najafabadi et al., 2015; Goodfellow et al., 2016; Elaraby et al., 2016; Liu et al., 2016). An auto-encoder is an unsupervised single-layer learning algorithm (Elaraby et al., 2016; Liu et al., 2016), which try to capture the structure of the input data, in a manner that learn to produce a compressed encoded representation form the raw input (Basegmez, 2014; Elaraby et al., 2016). The basic idea of the algorithm is to reconstruct the input using compression, based on intermediate representations. Consequently, in a sense, the target output is the input itself (Basegmez, 2014; Najafabadi et al., 2015).

The network structure is composed of two modules: an encoder function $h = f(x)$ and a decoder respectively that produces a reconstruction $r = g(h)$ (Goodfellow et al., 2016). The approach is based on abstraction, namely the input (raw data) is converted into an intermediate representation, which is then transformed in its original form by the encoder function. What is being sought throughout the processing process is to approximate the identity function (Liu et al., 2016). Auto-encoders are unable to learn to copy perfectly. This is mainly due to the training process of the network (Goodfellow et al., 2016). This network behavior results from the fact that only data that resembles the training data are being processed. Unlike other methods, which use different learning algorithms, an auto-encoder is not based exclusively on the training data and their labels, but attempts to capture (recreate) the internal structure of the input streams. In that sense, the hidden layers of the network can be seen as feature detectors (Basegmez, 2014). According to this architecture and functionality, an auto-encoder is a powerful model that is capable of extracting useful features, continuously, during the propagation process, while filtering the useless information and finally learning useful properties about the data (Liu et al., 2016; Goodfellow et al., 2016).

An auto-encoder network structure contains the same number of nodes in the input and output layer. Network's training is done using several techniques, including stochastic gradient descent (mini-batch gradient descent following gradients computed by the backpropagation algorithm), as well as using recirculation, which however is not as efficient as the backpropagation algorithm (Najafabadi et al., 2015; Goodfellow et al., 2016).

The training process is carried out in two stages. At first, unsupervised learning is used for feature learning, while during the next phase, supervised learning is used for fine-tuning the network. Getting a closer look of the training process, we observe the following: (Liu et al., 2016):

- Feed-forward propagation is first performed for each input, in order to obtain the output value $\bar{x}$.
- In the next phase, squared errors are used to measure the deviation of $\bar{x}$ from the input value.
- Then, the error will be backpropagated through the layers of the network to update the weights (Elaraby et al., 2016; Liu et al., 2016).

### Restricted Boltzmann Machines (RBMs)

Restricted Boltzmann Machines (RBMs) are generative stochastic neural networks, namely probabilistic generative models, which learn a joint probability distribution over the input (raw data), without using data labels (Chen & Lin, 2014; Basegmez, 2014; Najafabadi et al., 2015). A RBM is a diversified version of the conventional Boltzmann machines (BMs). A Boltzmann machine can be seen as a non-multilayered neural network, where units are bi-directional and make stochastic decisions, taking per case the value 1 or 0 (Liu et al., 2016; Elaraby et al., 2016).

Actually, a RBM is a special type of Markov random fields, containing one visible and one hidden layer (Najafabadi et al., 2015; Liu et al., 2016). The connections

between the layers are undirected, namely there are not connections between the units withing the same layer. Hence, connections are exclusively between units (neurons) of different layers, enabling the propagation of the values across the layers in both directions (Basegmez, 2014; Najafabadi et al., 2015).

The training process of a RBM is done through a sequence of successive processes. At first, Gibbs sampling is applied on a random state in one layer. Once the states of the units in one layer are given, it is automatically updated all the units in the other layers. This update process is repeated until a threshold value is met (Liu et al., 2016). In conclusion, we can say that RMBs are capable of utilizing large amounts of unlabeled data for exploiting complex data structures (Chen & Lin, 2014).

## Deep Neural Network (DNN) Architectures

Artificial neural networks (ANNs) based on deep architectures, recommend a powerful computational framework for machine learning. There are several types of deep architectures used in ANNs. In this section we will discuss two of the most popular and effective architectures used to build deep neural networks. Most notable, among others, are the deep belief networks (DBNs) and the convolutional neural networks (CNNs).

### Deep Belief Networks (DBNs)

#### Introduction

Deep Belief Networks (DBNs) are probabilistic generative models, composed of multiple layers of stochastic, latent variables (Liu et al., 2016; Arel et al., 2009). The typical DBN architecture consists of multi layers of hidden, random variables, that can be used as feature detectors (Elaraby et al., 2016). This deep architecture exploits both labeled and unlabelled data, allowing the learning of feature representations (Chen & Lin, 2014). The DBN approach incorporates strategies of unsupervised pre-training and

supervised fine-tuning in order to construct the models (Chen & Lin, 2014; Elaraby et al., 2016). A DBN model is made up of several individual modules. In particular, the model is composed of a stack of Restricted Boltzmann Machines (RBMs), aligned on top of each other (Hinton & Osindero, 2006; Basegmez, 2014; Chen & Lin, 2014; Liu et al., 2016; Elaraby et al., 2016).

### The structure and the algorithm

The building blocks that compose a deep belief network (DBN) consist of a stack of RBMs. Each such structure has one visible layer at the input and hidden layers up to the output. The connections of the network are located between the two adjacent layers, while in the same layer there are no connected units (Chen & Lin, 2014). The visible layer of each RBM is connected directly to the hidden layer of the previous RBM. Network's connectivity is not uniform across the entire network and is differentiated at the two higher levels, which are non-directional and symmetric. This configuration at the upper layers of the network recommends an associative memory (Elaraby et al., 2016; Liu et al., 2016).

A DBN can be considered as a special form of the Bayesian probabilistic generative model. This kind of model is able to learn a joint probability distribution of the training data, without being based on labeled data (Elaraby et al., 2016; Liu et al., 2016). DBNs, due to their specific topology and functionality, are an excellent choice for dealing with highly nonlinear parameter estimation problems, especially when applied on tasks with unlabeled data, providing efficient unobserved initialization points (i.e., the initial weights of the network are learned from the structure of the raw data) (Deng, 2011; Liu et al., 2016).

The training process of a DBN is divided into two distinct phases. During the first training phase, which is also called pre-training stage, a bottom-up strategy of information processing is carried out. The learning method used here is unsupervised

learning, which serves for feature extraction. Subsequently, during the second training phase, which is also called fine-tuning stage, a top-down strategy of information processing is carried out. The learning method used is supervised learning. In particular, a supervised learning algorithm is executed, in order to optimize the parameters and the overall network performance (Liu et al., 2016). This dual nature of a DBN, i.e., the combination of both bottom-up and top-down information processing, characterizes it as a hybrid model.

Learning processes carried out in a DBN are analyzed in two phases, the pre-training and the fine-tuning (Basegmez, 2014; Chen & Lin, 2014; Elaraby et al., 2016; Liu et al., 2016):

- Initially, a pre-training process is carried out. The basic principles followed are those that apply on Restricted Boltzmann Machines training.
- Under this prism, the first layer processes the raw data. This processing results in an intermediate representation, which is used as input to the next layer.
- The second layer is fed with the output data of the first layer, used as training data.
- The second layer is also trained as a RBM, exploiting the training examples from the previous step.
- The aforementioned procedure is repeated for all the remaining layers, until all the weights of the network are initialized.
- Then, a last layer is added to the network, representing the preferred outputs. This fine-tuning phase implements a top-down strategy, optimizing network's performance.

The pre-training phase, which is a fundamental procedure of DBNs, can also be implemented through the use of alternative architectures (e.g., stacked denoising auto-encoders) (Chen & Lin, 2014). Random initialization of weights can have several

implications and performance issues (e.g., local optima and over-fitting problems). Initializing the weights with the techniques used in DBN's, leads to avoiding such problems (Chen & Lin, 2014; Liu et al., 2016).

## Convolutional Neural Networks

### Introduction

Convolutional Neural Network (CNN) is a special feedforward-backpropagate neural network, which constitutes a variation of multiple layer perceptrons (MLP) architectures (Basegmez, 2014). This kind of network is directly inspired by the biological processes taking place in the visual system (Hubel & Wiesel, 1962; Palm, 2012; Basegmez, 2014; LeCun et al., 2015). Extensive experimental studies conducted in the 1960s by Hubel and Wiesel (Hubel & Wiesel, 1962) on the mammalian visual cortex, revealed that this part of the brain is composed of specific cells with high specificity to patterns within a localized area (i.e., receptive fields). The spatial arrangement of these cells covers the entire visual field, and the activation is gradually propagated into higher level cells. Respectively, higher levels cells, which have greater receptive fields, receive input from these simple cells. This architecture allows the learning of translational-invariant representations (Arel et al., 2009; Palm, 2012; Basegmez, 2014; Najafabadi et al., 2015).

From a technical point of view, the origin of convolutional models is based on Neocognitron model, which resembles the structure of the mammalian visual system. Back in the 1980s, Neocognitron model introduced a powerful image processing tool (Fukushima, 1980). Beyond the common origin and similarities between the two models, there are also some fundamental differences, concerning architectural issues. A basic difference lies in the fact that CNN's use the backpropagation algorithm (Fukushima & Miyake, 1982; LeCun et al., 2015).

Convolutional neural network was introduced by Y. LeCun in the 1990s (LeCun et al., 1990), and is mostly used in image recognition tasks, as well as in other machine learning problems (e.g., face detection, document analysis, speech detection) (Arel & Rose, 2010; Palm, 2012; Goodfellow et al., 2016). CNN's can deal with several important aspects of natural signals such as local connections, shared weights, pooling, as well as the use of multiple successive layers of hierarchy, namely multiple layers of neuron groups (Basegmez, 2014; Chen & Lin, 2014; LeCun et al., 2015; Goodfellow et al., 2016).

### Basic Ideas

Conventional artificial neural networks (e.g., shallow feedforward neural networks) are a powerful and reliable machine learning tool, which however, due to inherent architectural and structural constraints, is not the optimal choice for several machine learning problems (e.g., time-series and image data problems). Convolutional neural networks operate directly on input data, minimizing the need for extensive data pre-processing (Arel & Rose, 2010; Basegmez, 2014). This property is very useful when dealing with two (or more) dimensional data as is the case of image recognition tasks. (Arel & Rose, 2010; Basegmez, 2014). While ANN's concatenate the two dimensional data into vectors, CNN's layers are associated directly to portions of the input (Basegmez, 2014).

Inspired by the architecture of the visual cortex pathway, which is being consisted of simple and complex cells recommending a hierarchical structure, in a typical convolutional neural network there are multiple levels of hierarchy. Some layers serve for feature representations, while others for classification processes (Chen & Lin, 2014). According to this functionality, a CNN is composed of convolution layers and sub-sampling or pooling layers. The convolution layers form a kind of intermediate representations called feature maps, where each such map is derived over feature maps

of former layers. The subsampling (pooling) layers reduce the sizes of proceeding layers, by downsampling the feature maps based on a stable coefficient (Palm, 2012; Chen & Lin, 2014).

CNN's are designed to take advantage of the properties of natural signals (LeCun et al., 2015). The procedure of modeling physical signal processing raises several challenges and requirements that need to be overcome. One of the most important challenges is to take into account the spatial dimensions properties of the input (Gudi, 2014). The standards that meet the above challenges and are embedded in CNN's are local connections, shared weights and pooling processes (LeCun et al., 2015).

The basic (binary) function implemented in neural networks is matrix multiplication. What differentiates CNN's both in terms of architecture and functionality is the implementation of a mathematical operation called convolution (Goodfellow et al., 2016). Convolution is a mathematical operation that captures the amount of overlap of one function as it is shifted over another function. The implementation of a convolution operation on two functions, results in the production of another function, which is a modified version of one of the original functions. This kind of linear operation, which is actually a filtering process, allows pattern detection in different parts of an array (LeCun et al., 2015).

The representations resulting by the convolution process are smaller than the input itself. This is due to the fact that convolutional networks use sparse weights. The sparse connectivity implies that only some of the output units interact with some of the input units (Goodfellow et al., 2016). Another property of convolutional networks is concerned with parameter sharing. According to this property, the same parameter is being used for multiple functions (Goodfellow et al., 2016). Parameter sharing scheme is based on the assumption that if one feature is useful to compute at some spatial location, then it should also be useful to compute at a different location. The benefits of

incorporating these properties results in minimizing memory requirements, and improving the total performance of the model. The main advantage of CNN's is the tolerance to the translation of the input data, meaning that if the input changes, then accordingly the output changes in the same manner (Basegmez, 2014; Goodfellow et al., 2016).

## References


1. Ahmad, S. & Hawkins, J. (2015). Properties of Sparse Distributed Representations and their Application to Hierarchical Temporal Memory. *CoRR*, abs/1503.07469.
2. Arel, I., Rose, D. C., & Coop, R. (2009). DeSTIN: A Scalable Deep Learning Architecture with Application to High-Dimensional Robust Pattern Recognition. In Proceedings of AAAI Fall Symposium: Biologically Inspired Cognitive Architectures.
3. Arel, I., Rose, D. C., & Karnowski, T. P. (2010). Deep Machine Learning - A New Frontier in Artificial Intelligence Research. *Comp. Intell. Mag.*, Vol. 5(4), pp. 13-18.
4. Bar, M., Tootell, R. B.H., Schacter, D. L., Greve, D. N., Fischl, B., Mendola, J. D., Rosen, B. R., & Dale, A. M. (2001). Cortical Mechanisms Specific to Explicit Visual Object Recognition. *Neuron*, Vol. 29, pp. 529–535.
5. Basegmez, E. (2014). The Next Generation Neural Networks: Deep Learning and Spiking Neural Networks. Technische Universitat Munchen.
6. Bell, A. J. & Sejnowski, T. J. (1997). The "independent components" of natural scenes are edge filters. *Vision Res.*, Vol. 37, No. 23, pp. 3327-3338.
7. Bengio, Y. (2009). Learning Deep Architectures for AI. Foundations and Trends in Machine Learning. Vol. 2, No. 1, pp. 1-127.
8. Bengio, Y. & LeCun, Y. (2009). Tutorial: Learning Deep Architectures. In Proceedings of the 26th Annual International Conference on Machine Learning.
9. Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, Vol. 2, No. 1, pp. 1-127.
10. Bengio, Y. & Lecun, Y. (2007). *Large-scale kernel machines. Scaling learning algorithms towards AI*. MIT Press, London: Cambridge, Massachusetts.

11. Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 35, No. 8, pp. 1798-1828.
12. Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy Layer-Wise Training of Deep Networks. In Proceedings of NIPS'06, pp. 153-160.
13. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer Science+Business Media, LLC, New York, USA.
14. Buonomano, D. V. & Merzenich, M. M. (1998). CORTICAL PLASTICITY: From Synapses to Maps. *Annu. Rev. Neurosci.*, 1998. 21:149-86.
15. Camastra, F. & Vinciarelli, A. (2007). Machine Learning for Audio, Image and Video Analysis: Theory and Applications. Springer, ISBN:1848000065 9781848000063.
16. Cao, C., Ma, L., & Xu, Y. (2012). Adaptive Control Theory and Applications. *Journal of Control Science and Engineering*, Vol. 2012, Article ID 827353, 2 pages.
17. Chen, X. W. & Lin, X. (2014). Big Data Deep Learning: Challenges and Perspectives. *IEEE Access*, Vol. 2, pp. 514-525.
18. Chien J. T. & Hsieh H. L. (2013). Nonstationary Source Separation Using Sequential and Variational Bayesian Learning. *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 24, No. 5, pp. 681-694.
19. Chikofsky, E. J. & Cross, J. H. (1990). Reverse engineering and design recovery: a taxonomy. *IEEE Software*, Vol. 7, No. 1, pp. 13-17.
20. Dean, J., Corrado, G. S., Monga, R., Chen, K., Devin, M., Le, Q. V., Mao, M. Z., Ranzato, M., Senior, A., Tucker, P., Yang, K., & Ng, A. Y. (2012). Large Scale Distributed Deep Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, NIPS'12, pp. 1223-1231.

21. Deng, L & Yu, D. (2014). Deep Learning: Methods and Applications. *Foundations and Trends in Signal Processing*, Vol. 7, No. 3-4, ISSN: 1932-8346, pp. 197-387.
22. Deng, L. (2011). An Overview of Deep-Structured Learning for Information Processing. In Proceedings of the APSIPA-ASC 2011.
23. Deng, L. (2012). Three Classes of Deep Learning Architectures and Their Applications: A Tutorial Survey. *APSIPA Transactions on Signal and Information Processing*.
24. Deng, L., Yu, D., & Platt, J. (2012). Scalable stacking and learning for building deep architectures. In Proceedings of ICASSP 2012.
25. Elaraby, N. M., Elmogy, M., & Barakat, S. (2016). Deep Learning: Effective Tool for Big Data Analytics. *International Journal of Computer Science Engineering (IJCSE)*, Vol. 5 No. 05, Sep 2016.
26. Felleman D. J. & Essen DC. V. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb Cortex*, 1(1), pp. 1-47.
27. Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, Vol. 36, 4, pp. 193-202.
28. Fukushima, K. & Miyake, S. (1982). Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, Vol. 15, pp. 455-469.
29. Gluck, M. A. & Myers, C. E. (2000). Gateway to Memory: An Introduction to Neural Network Models of the Hippocampus and Learning. MIT Press, Cambridge: 2000.
30. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning (Adaptive Computation and Machine Learning series) MIT Press, 2016.

31. Guanyu, Z., Kihyuk, S., & Honglak, L. (2012). *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics. Online Incremental Feature Learning with Denoising Autoencoders*. In Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, PMLR, Vol. 22, pp. 1453-1461.
32. Gudi, A. (2014). Recognizing Semantic Features in Faces using Deep Learning. Master's Thesis, Informatics Institute, Graduate Schools of Science, University of Amsterdam.
33. Hamed, M. G., Gianazza, D., Serrurier, M., & Durand, N. (2013). Statistical prediction of aircraft trajectory: regression methods vs point-mass model. In Proceedings of USA/Europe Air Traffic Management Research and Development Seminar, 2013.
34. Hammer, B., He, H., & Martinetz, T. (2014). Learning and modelling big data. In Proceedings of the ESANN 2014, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, ISBN 978-287419095-7.
35. Hinton, G. E. (1986). Learning distributed representations of concepts. In Proceedings of the Eighth Annual Conference of Cognitive Science Society, Amherst, Mass, 1986, pp. 1-12.
36. Hinton, G. E. & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, Vol. 313, Issue 5786, pp. 504-507.
37. Hinton, G.E., Osindero, S., & Teh, Y.W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18, pp. 1527–1554.
38. Hubel, D. H. & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex, *J Physiol.*, Vol.160(1), pp. 106-154.

39. Jewell. D., Barros, R. D., Diederichs, S., Duijvestijn, L. M., Hammersley, M., Hazra, A., Holban, C., Li, Y., Osaigbovo, O., Plach, A., Portilla, I., Saptarshi, M., Seera, H. P., Stahl, E., & Zolotow, C. (2014). IBM Form #: REDP-5070-00.
40. Kaminski, K. T. & Milkowski, M. (2013). *Regarding the Mind, Naturally: Naturalist Approaches to the Sciences of the Mental. Chapter one: Reverse Enginnering in Ccognitive Science*. Cambridge Scholars Publishing, 12 Back Chapman Street, Newcastle upon Tyne, NE6 2XX, UK.
41. Katal, A., Wazid, M., & Goudar, R. (2013). Big data: Issues, challenges, tools and Good practices. *IC3*, pp. 404-409, IEEE.
42. Kriegeskorte, N. (2015). Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing. *Annual Review of Vision Science*, Vol. 1, pp. 417-446.
43. Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2016). Building Machines That Learn and Think Like People. *CoRR*, Vol. abs/1604.00289.
44. Lampropoulos, A. S. & Tsihrintzis, G. A. (2015). Machine Learning Paradigms, Applications in Recommender Systems. Intelligent Systems Reference Library, Vol. 92, © Springer International Publishing Switzerland 2015.
45. Laney, D. (2001). Data Management: Controlling Data Volume, Velocity, and Variety. Copyright © 2001, META Group Inc.
46. LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Howard, W. E.,& Jackel, L. D. (1990). *Advances in Neural Information Processing Systems. Handwritten Digit Recognition with a Back-Propagation Network*. Morgan-Kaufmann, pp. 396-404.
47. LeCun, Y., Bengio, B., & Hinton, G. (2015). Deep learning. *Nature*, Vol. 521, pp. 436-444.

48. Lee, H., Roger, G., Rajesh, R., & Andrew, Y. Ng. (2009). *Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations*. In Proceedings of the 26th Annual International Conference on Machine Learning, ACM, Montreal, Quebec, Canada: 2009, ISBN: 978-1-60558-516-1, pp. 609-616.
49. Lee, T. S., & Mumford, D. (2003). Hierarchical Bayesian inference in the visual cortex. *Journal of the Optical Society of America*, Vol. 20, No. 7, July 2003.
50. Lee, T. S., Mumford, D., Romero, R., & Lamme V. A.F. (1998). The role of the primary visual cortex in higher level vision. Vision Research Vol. 38, pp. 2429-2454.
51. Levine, S. (2013). Exploring Deep and Recurrent Architectures for Optimal Control. *CoRR*, Vol. abs/1311.1761.
52. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, Vol. 234, pp. 11-26.
53. Liu, X., Zhang, G., Zhan, Y., & Zhu, E. (2008). *Frontiers in Algorithmics: Second Annual International Workshop. An Incremental Feature Learning Algorithm Based on Least Square Support Vector Machine*. In Proceedings of FAW 2008, Springer Berlin Heidelberg, Berlin, Heidelberg: 2008, pp. 330–338.
54. Mitchell, T. M. (2006). The Discipline of Machine Learning. CMU-ML-06-108.
55. Mountcastle, V. B. (1997). The columnar organization of the neocortex. *Brain*, 120 (4), pp. 701-722.
56. N. Zorbas, D. Zissis, K. Tserpes and D. Anagnostopoulos, "Predicting Object Trajectories from High-Speed Streaming Data," in Proceedings of IEEE Trustcom/BigDataSE/ISPA.

57. Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M.,Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, Vol. 2, No. 1.
58. Nilsson, N. J. (1996). Introduction to Machine Learning. An early draft of a proposed textbook. Copyright © 2005 Nils J. Nilsson.
59. Nilsson, N. J. (1998). Introduction to Machine Learning: An Early Draft of a Proposed Textbook. Copyright © 2005 Nils J. Nilsson.
60. Pallath, P. (2015). Embed Deep-Learning Techniques into Predictive Modeling, Using SAP Predictive Analytics for Complex Modeling. © 2015 SAP SE or an SAP affiliate company.
61. Palm, R. B. (2012). Prediction as a candidate for learning deep hierarchical models of data. Master's thesis, Technical University of Denmark, DTU Informatics.
62. Poggio, T. (2016). Deep Learning: mathematics and neuroscience. Views and Reviews, CBMM.
63. Polk, T. A. & Seifert, C. S. (2002). Cognitive Modeling. The MIT Press, © 2002 Massachusetts Institute of Technology.
64. Quinlan, J.R. (1990). Learning Logical Definitions from Relations. *Machine Learning*, Vol. 5, No. 3, 239-266.
65. Rocki, K. (2016). Towards Machine Intelligence. *CoRR*, Vol. abs/1603.08262.
66. Rogers, T. T. & McClelland J. L. (2004). Semantic cognition : a parallel distributed processing approach. The MIT Press, © Massachusetts Institute of Technology: 2004.
67. Rogers, T. T. & McClelland, J. L. (2004). *Semantic Cognition. A Parallel Distributed Processing Approach*. The MIT Press, Cambridge, Massachusetts London, England: 2004.

68. Rojas, R. (1996). Neural Networks A Systematic Introduction. Springer-Verlag, Berlin: 1996.
69. Russell, S. J. & Norvigm P. (2010). Artificial Intelligence: A Modern Approach, 3rd Edition. Pearson Education, Inc., Copyright © 2010, 2003, 1995 by Pearson Education, Inc.
70. Schapire, R. (2008). Computer Science 511 Theoretical Machine Learning. Princeton University, Computer Science Department, Lecture #1, February 4, 2008.
71. Scherer, D., M{\"u}ller, A., & Behnke, S. (2010). *Artificial Neural Networks. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition*. In Proceedings of ICANN 2010: 20th International Conference, Thessaloniki, Greece, September 15-18, 2010, pp. 92-101.Deep Learning in Neural Networks: An Overview.
72. Schmidhuber, J. (2014). Deep Learning in Neural Networks: An Overview. *CoRR*, Vol. abs/1404.7828, Technical Report.
73. Serre, T., Kreiman, G., Kouh, M., Cadieu, C., Knoblich, U., & Poggio, T. (2007). A quantitative theory of immediate visual recognition. *Progress in Brain Research*, Vol. 165, pp. 33-56.
74. Shai, S-S. (2012). Online Learning and Online Convex Optimization. *Foundations and Trends® in Machine Learning*. Vol. 4, pp. 107-194.
75. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P-A. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *J. Mach. Learn. Res.*, Vol. 11, pp .3371-3408.
76. Witten, I. H. & Frank, E. (2005). Data Mining, Practical Machine Learning Tools and Techniques, 2nd Edition, Copyright © 2005 by Elsevier Inc.

77. Witten, I. H., Frank, E., & Hall, M. A. (2011). Data Mining, Practical Machine Learning Tools and Techniques, 3rd Edition, Copyright © 2011 Elsevier Inc.
78. Wu, Y. & Razavim R. (2015). An Introduction to Deep Learning, Examining the Advantages of Hierarchical Learning. © 2015 SAP SE or an SAP affiliate company.