Karlsruher Institut für Technologie

Karlsruhe Institute of Technology
Communications Engineering Lab

TBS

Master's Thesis

Nicolas Cuervo-Benavides

Advisor	:	Dr.-Ing. Holger Jäkel
Supervisor	:	MSc. Felix Wunsch

Start date	:	17th May 2017
End date	:	6th December 2017

Declaration

With this statement I declare that I have independently completed the above master's thesis. The thoughts taken directly or indirectly from external sources are properly marked as such. This thesis was not previously submitted to another academic institution and has also not yet been published.

Karlsruhe, 06.12.2017

Nicolas Cuervo-Benavides

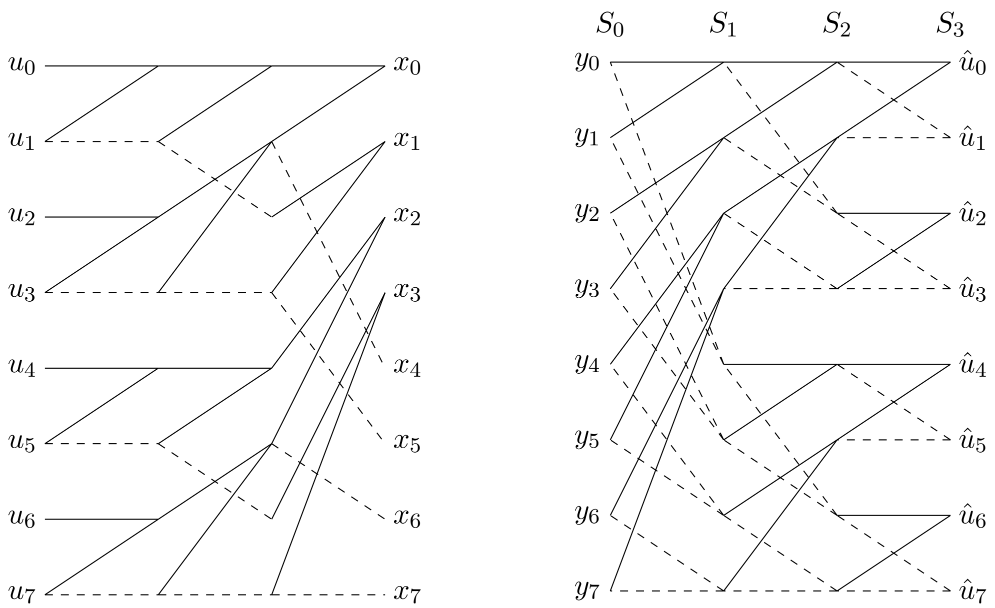
Abstract

This thesis collects the fundamentals of machine learning and applies them in a determined, state-of-art, communications scenario. CEL thesis rules require it to be about 3-5 pages. It is a summary of what you do in your thesis. Use around 5 pictures and outline whatever you did. And now a few lines of information.

This is a todo example

Polar codes are the first codes to asymptotically achieve channel capacity with low complexity encoders and decoders. They were first introduced by Erdal Arikan in 2009 [Jon05]. Channel coding has always been a challenging task because it draws a lot of resources, especially in software implementations. Software Radio is getting more prominent because it offers several advantages among which are higher flexibility and better maintainability. Future radio systems are aimed at being run on virtualized servers instead of dedicated hardware in base stations [Jon05]. Polar codes may be a promising candidate for future radio systems if they can be implemented efficiently in software.

In this thesis the theory behind polar codes and a polar code implementation in GNU Radio is presented. This implementation is then evaluated regarding parameterization options and their impact on error correction performance. The evaluation includes a comparison to state-of-the-art Low-Density Parity-Check (LDPC) codes.



(a) 8 bit polar encoder **(b)** 8 bit polar decoder

Figure 0.1.: Polar code encoding and decoding

The polar encoder is shown in Fig. 0.1a.

Contents

1. Introduction

Back in 1999, Joseph Mitola III coined the term Cognitive Radio (CR)[MM99] as a way to enhance the Software-Defined Radio (SDR) capabilities by the means of a dynamic model that, based on human intervention, improved the flexibility of devices by making them fully configurable and capable of adapting to the communication system's needs, suitable to react to the changes it the surrounding environment. A formal definition for the CR concept provided at [Hay05] encloses the term nicely by describing it as a wireless system that is *intelligent and aware of its surroundings*, whilst being able to learn, adapt and react to changes in the environment, by modifying its operation parameters such as the transmission power, the modulation scheme and its carrier frequency in real-time. Analogously, Jondral [Jon05] adopts the short definition for CR as "an SDR that additionally senses its environment, tracks changes, and possibly reacts upon its findings", becoming an autonomous unit with the potential of using the spectrum efficiently.

CR systems are intended to be immerse in a network, where it interacts with other systems that could be cognitive or non-cognitive radios. According to [GAMS], CR is grouped under three paradigms: underlay, overlay and interweave. The *Underlay Paradigm* allows the CR system to operate under acceptable levels of interference, determined by an interference threshold. Here, the CR is commonly called a Secundary User (SU), providing priority to the other systems in the network which it should not significantly interfere, known also as Primary User (PU). In the *Overlay Paradigm*, the cognitive transmitter knows information about the other transmitters in the network, such as their codebooks and modulation schemes. In addition, this model assumes that message that is being transmitted is known by the CR when transmission by a non-cognitive system is initiated. This provides the cognitive system with multiple choices on how to use this information: for instance, it can be used to mitigate or completely cancel a possible interference happening in the network during transmission. Additionally, the cognitive system could also retransmit this message to other non-cognitive systems in the network, acting as a relay and, effectively, assist increasing the Signal-to-Noise-Ratio (SNR) of the non-cognitive system to a level equivalent to the possible decrease due to CR transmissions. The *Interweave Paradigm*, or opportunistic communication, identifies temporary space-time-frequency gaps where it can intelligently allocate its transmission, increasing the available resource utilization and minimizing the interference with other active users. Hybrid schemes are also actively being developed [WN07] [KWS+15] [WKM+17], where characteristics from different paradigms are combined in order to achieve an effective use of the available communication resources.

The main characteristic required to apply any of the aforementioned paradigms is awareness, being it in regard of location, spectrum, time, etc. Awareness is achieved by the means of *the cognition cycle* [MM99], which can be seen in Fig. 1.1, which enfolds the way the CR parses the stimuli from the outside world in order to plan accordingly the proper reactions. This cognition cycle revolves around the following concepts

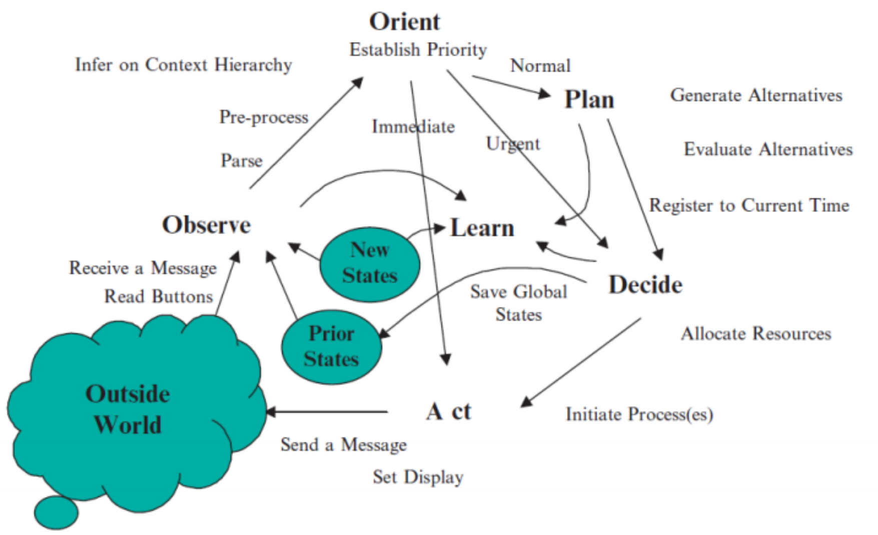


Figure 1.1.: The cognition Cycle[MM99]

- Observation: the CR receives any signals from the external world, which can contain any type of information that the system can use in its favor and the favor of a better use of its resources.
- Orientation: The CR determines the priority from the received signal as well as the type of reaction based on it.
- Planning: results from a normal-level priority, where a plan is generated and the sequence of actions to be taken are established.
- Decision: selects among the plan candidates the best proposal and allocates the necessary resources for its carrying-out.
- Acting: initiates the decided processes.
- learning: is an integration of observations and decisions, based on past and current states that are compared with expectations. When an expectation is met, the system achieves effectiveness. When not, observations are recorded and kept for further learning.

These aspects of CR come in handy when trying to solve one of the current major issues of communication systems: Spectrum Scarcity. The access to radio spectrum is highly regulated by government agencies such as the U.K. Offices of Communications (OFCOM), the Federal Coomunications Commision (USA) (FCC) and the International Telecommunications Union (ITU), and its access has been historically granted to the highest bidder on so-called *Spectrum Auctions* [Jon05] [SW14]. Therefore, the seek of new technologies that allow a more efficient access to the spectrum is paramount. In an effort to find effective solutions for this increasing issue, the Institute of Electrical and Electronics Engineers (IEEE) created a Standards Committee back in 2005 which, in association with the IEEE Communications Society (ComSoc) and the IEEE Electromagnetic Compatibility Society (EMC) dealt with the generation of standards for dynamic spectrum management. This committee was dissolved between 2007 and 2010 and, after organizational restructuring, the functions of standardization and spectrum management was handed to the Standards Coordinating Committee 41 (SCC41) - Dynamic Spectrum Access Networks (DySpan) [IEE15]. As part of this efforts to motivate state-of-art research in this regards, DySpan has organized since 2007 the *IEEE International Symposium on Dynamic Spectrum Access Networks* [Com]. Additionally, DySpan has embolden the healthy competition since 2015 by introducing the *Spectrum Challenge*, consisting on inviting team worldwide to solve a problem related with dynamic access to the spectrum and 5G implementations. The participating teams are given a set of requirements and limitations, but are encouraged to push this limits with creativity and innovation. The Karlsruhe Institute of Technology (KIT), represented by the Communications Engineering Lab (CEL), has taken part in these competitions achieving outstanding results, being awarded with the *Subjective Winner* award on 2015 [KWS+15] and the *Best Overall Solution* on 2017 [WKM+17]. This thesis utilizes the setup used at the 2017 spectrum challenge as base testbed. Fig. 1.2 shows the main characteristics of this setup.

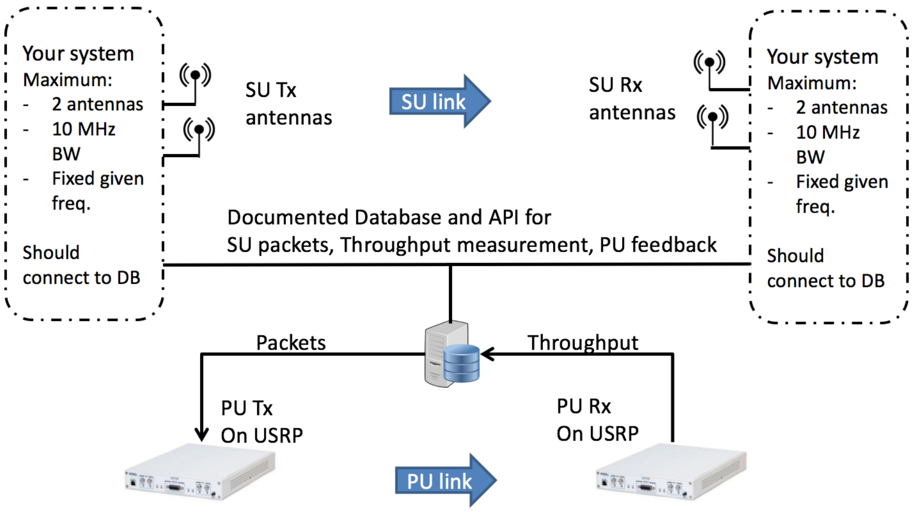


Figure 1.2.: The DySpan Spectrum Challenge Setup [Cha]

By using this configuration and keeping the hardware and overall physical considerations (such as Bandwidth (BW), number of antennas and central frequency), the idea of the challenge was to achieve the maximum throughput between the proposed SU systems, while interfering as little as possible with the existing PU. The competition consisted of two phases: during phase one the situational awareness of the proposed CR system is put under test, as it need to correctly identify the set of PU transmission parameters:

- Bandwidth and Carrier Frequency: along the 10MHz of maximum BW divided in four subchannels of 2.5MHz each, it needs to be detected if the PU is using one, two or four channels for its transmission. Effectively, it is needed to determine which frequencies are being used (identify the frequency hopping pattern) and when are they being used.

- Packed length: the PU transmitter sends packets in a bursty fashion to the corresponding receiver using packets of 100 or 1000 bytes.

- Inter arrival time between packets: the time between a packet transmission might vary from a situation to another. This times could be deterministic for some scenarios, as well as stochastic for others following a Poisson distribution. Correctly identifying the length of the packet, as well as the inter packet time of the current situation, allows to effective opportunistic access to the spectrum.

With this characteristics, a set of 10 different scenarios is built, whose parameters are depicted in Table 1.1

Scenario	**Description**
0	Single random channel, deterministic interpacket delay of 5ms
1	Single random channel, deterministic interpacket delay of 10ms
2	Two random channel hopping, deterministic interpacket delay of 5 ms
3	Four random channel hopping, deterministic interpacket delay of 10 ms
4	Two random channel hopping, deterministic interpacket delay of 5ms
5	Four synchronous channels, deterministic interpacket delay of 5ms
6	Four synchronous channels back-to-back, deterministic interpacket delay of 2ms
7	Four asynchronous random channels, Poisson distributed interpacket delay with mean of 20ms
8	Four asynchronous random channels, Poisson distributed interpacket delay with mean of 10ms
9	Four asynchronous random channels, Poisson distributed interpacket delay with mean of 5ms

Table 1.1.: Scenario description

The second phase of the competition regards the benchmark of the performance of the proposed SU implementation, where aspects such as innovation of the used waveform, machine learning algorithms used, and opportunistic access to the spectrum were considered. The proposed solutions, including the one proposed by CEL, can be found at [WKM+17], [PST+17], [PSK+17] and [LMH+17], where a high level of innovation and state-of-the-art research is compiled.
Being clear that *awareness* has been a primordial characteristic of CR since the conception of the concept until now, understanding that this is an area that invites to further research and considering the uprising research in the field of Artificial Intelligence (AI) algorithms, this work focuses on the learning aspect of CR, using the setup from Fig. 1.2 in order to effectively identify the scenarios described at Table 1.1. Previous research on this field covers aspects such as modulation recognition [OCC16a][OCC16b], resource allocation [ZMJ16], autoencoding and optimization of MIMO systems [OEC17], dynamic spectrum management [Hay05] and context awareness [PSK+17][WPR+17].
The outline of this thesis is as follows: an introduction to AI, focused on Machine Learning (ML) and Deep Learning (DL), alongside the most used algorithms used in academic and industrial fields is presented in chapter 2. General techniques to avoid phenomena such as underfitting and overfitting of the ML are, as well, portrayed. Chapter 3 describes the details of the testbed set up, the measurement of the data and the implementation of the machine learning models. The evaluation of the learning models is then presented in 4

with metrics of performance. The models are put into a live implementation, where the performance of the algorithms is put into test by classifying the scenarios of Table 1.1 in real-time - This is presented in chapter 5. Lastly, the conclusions and future work are summarized in chapter 6.

2. Artificial Intelligence

2.1. Overview

Intelligence as a concept has been a topic of exhausting research in fields such as neurology, philosophy, neuroscience, neurobiology, datascience, among others. The Oxford dictionary defines intelligence as *"the ability to acquire and apply knowledge and skills"* [Oxfa]. The first part of this definition applies to what is known as "learning", which is according to the accepted definition of the term as well [Oxfb], and that supports, from the etymology, the importance of the process of learning on intelligence.

Jeff Hawkins, a dedicated neuroscientist and author, has approached the subject from the engineering and medical flanks, analysing the structure of the brain and having the perspective of the possibilities of replicating artificially the most sophisticated type of intelligence found on Earth: the human. In his book *On Intelligence* [HB04], he captures his findings after inspecting the brain cortex and making a parallel between humans and machines. According to Jeff, *"it is the ability to make predictions about the future that is the crux of intelligence"*, and these predictions are based on the experiences from which the intelligent being has learnt, making decisions that lead it to the best possible known result. In order to create artificially a so-called *intelligent agent*, scientist have put extensive effort first on trying to replicate the known intelligence [Bro91][RPP07][Haw], taking the approach of generating a machine that is human-like and that behaves like one, being able to observe its surroundings, learn from stimulus that come from the real world, adapt to changes in those surroundings, plan accordingly to foreseeable process (therefore, make predictions), make decisions and act appropriately. These are the characteristics that Mitola [MM99] described in the cognition cycle for Cognitive Radio (CR), which can be applied to any intelligent agent and, consequently, motivate the further research of Artificial Intelligence (AI).

AI, however, encircles a variety of disciplines that are in themselves a complete course of research, as it can be seen in Fig 2.1. This work focuses only in the top branch: machine learning. However, given the slight differences regarding implementation, a separate section will be dedicated solely to deep learning.

2.2. Machine Learning

Machine Learning (ML) encloses the process of taking a data set that represents any phenomena and learning from it. Any type of being that is capable of learning from previous experiences is showing a kind of intelligence, as it interiorizes the stimulus/data and reacts accordingly when it presents itself again. The vast majority of living beings have this capacity, being the humans who have the lead on its effectiveness. Identifying objects, speaking languages, and reacting to any sensorial stimulus is a result of a successful learning process.

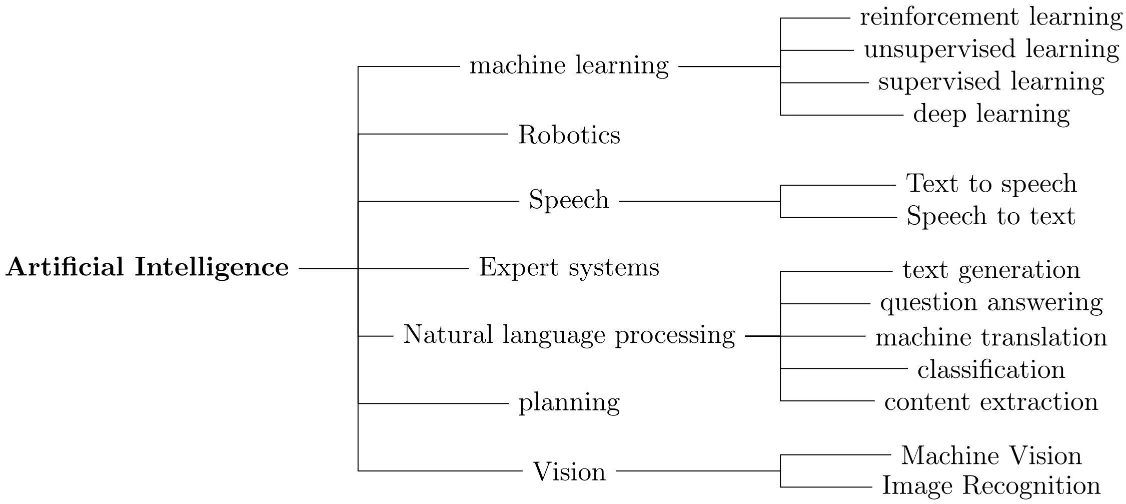


Figure 2.1.: Artificial Intelligence

Generally speaking, learning from data is done when no there is no analytic solution to an encountered situation, but there is enough data to adapt to the it, generating an empirical solution to a problem that cannot be mathematically a-priori described, but that follows a specific pattern[AMMIL10]. Just as humans do, the idea of machine learning is to generate intelligent agents computationally - teach computers to learn. The idea is as follows: a machine learning algorithm is given a set of data from which it can extract specific information that tells it the specifics about the data. With enough information, the computer is able to make predictions about other data in a different point of time if this data presents the same characteristics.
Although there is no specific mathematic representation of the specific problem to solve, many ML algorithms relay heavily on mathematic definitions and optimization theory. Further information regarding ML algorithms can be found in section 2.2.8. Yet is this versatility provided by the fact of not needing to pin down the specific analytic description of the problem which has impulsed this methodology into several fields of knowledge, being nowadays applied to solve problems such as financial forecasting[BM01], medical diagnosis[Kon01], entertainment[BL07] and communications systems (such as this thesis), among others. Examples of everyday problems that are suitable for ML implementation are:

- Ranking links and clicks for a better web search engine and advertisements.
- Custom user recommendations based on purchases/rents/views.
- Prediction of markets and stock exchange.
- Dating sites with reevaluation of algorithms based on successful matches.
- Financial fraud detection.
- Supply chain optimization

- Biotechnology research acceleration by sequencing and screening of DNA and protein/compound structures.

- National security based on enormous surveillance data.

2.2.1. The learning problem

Learning from data is definitely a hot topic, which can be seen from the increasing amount of research and application that has been handed over this theory and methodology. Additionally, it is noticeable how the term has been capturing the mainstream interest and is somewhat heard-of, as it can be seen in the Fig. 2.2, where this trend over the past few years is clear. At this point, it is preeminent to clarify what is the purpose of ML, and when it plays an important role. Although ML has shown to perform outstandingly into solving many problems, it is not intended to move aside the many and well designed analytic solutions for many of the scientific existing problems, but to come in handy when that analytic solution does not describe completely the problem or does not exist. In his book [AMMIL10], Prof. Yaser nicely states that although many problems can be solved effectively using a learning approach or an analytic approach, the point of learning is not to compare itself and overcome the performance over the mathematical description of existing problems, but to be a complementary tool for scientist in their eagerness to solve complex problems without being stuck when facing the lack of a complete description of it.

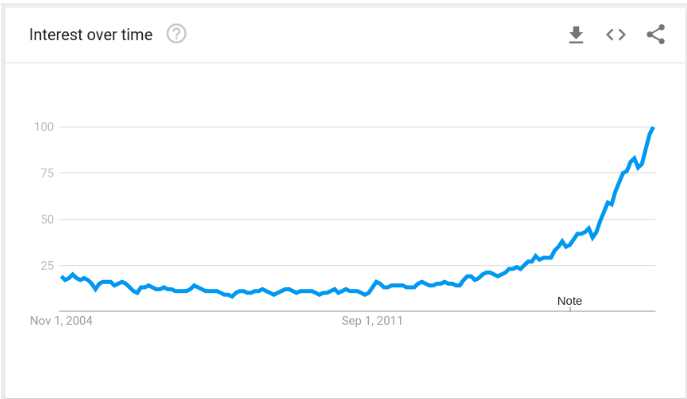


Figure 2.2.: 'Machine learning' Google search trend [Goo17]

The learning problem is summarized in the Fig 2.3. The learning algorithm $\mathcal{A}$ receives data of any form, and its solely purpose is to identify mechanisms that describe that dataset $\mathcal{D}$ closely. This dataset is defined as input-output samples for the supervised learning, input-weights samples for reinforcement learning and only inputs for the unsupervised learning. Further information regarding supervised, reinforcement and unsupervised learning can be found in section 2.2.2. For the sake of the explanation, lets take the supervised learning case, where the dataset $\mathcal{D}$ includes samples of the form $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_N, y_N)$, where x is the input that belongs to the input space $\mathcal{X}$, y is the corresponding output such that $y_n = f(x_n)$, and belongs to the output space $\mathcal{Y}$. Now, the learning algorithm $\mathcal{A}$ needs to find that function $f(x)$. For this, it counts with an hypothesis set $\mathcal{H}$, which are the mathematical representations that the algorithm uses as tools to accomplish his purpose. From $\mathcal{H}$ the algorithm takes one hypothesis $g : \mathcal{X} \rightarrow \mathcal{Y}$ that approximates f. After a g has been selected, the process estimates how alike the outputs from $g(x)$ are to $f(x)$, and feedbacks an error measure $E(g, f)$. This process is repeated iteratively until an hypothesis produces an acceptable minimum error.

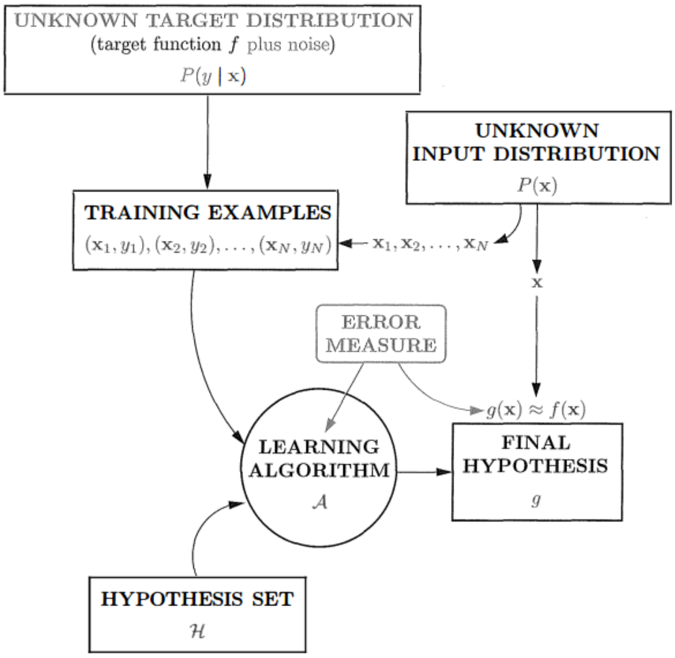


Figure 2.3.: The learning problem [AMMIL10]

Now, it is imperative to determine quantitatively what exactly *acceptable* entitles. Different applications can have different tolerances to error, and this affects directly the hypothesis } that is chosen at the end of the learning process. This means that this is an end-user parameter that has to be set as a requirement for the whole system.
After taking many different samples from $\mathcal{X}$, and reducing the error, we should take into account the samples that *we did not take*, meaning the samples that are not in our input set, and that probably behave similar to the $\mathcal{X}$ set - i.e. we should be able to identify similar samples in order to be able to predict. Therefore, a probability distribution is added to both the input samples and the final hypothesis in order to infer from $\mathcal{D}$ the behaviour of samples that are not in $\mathcal{D}$. Let us assume a binary classification problem, where $\mathcal{D}$ contains two classes A and B in a possibly infinite number of them. From a random sample pick the probability that the input sample is of the A type will be denoted as μ and, consequently, the probability that the input sample is of the B type is 1 - μ. The value of μ is unknown and will continue to be unknown in the process. From a random pick of N samples from $\mathcal{D}$, there is a proportion of ν samples of the A type, and we intend to determine how ν relates to μ. In statistical jargon, we want to determine how our sample relates to the whole population.

From any point of view, the larger the sample that contributes to ν, the closer the relation it has with μ, but this relation can be quantified using the *Hoeffding Inequality* [Hoe63], which states that for any sample size N,

$$\mathbb{P}[|\nu - \mu| > \epsilon] \leq 2e^{-2\epsilon^2 N} \quad \text{for any} \quad \epsilon > 0$$

Where $\mathbb{P}[\cdot]$ denotes the probability with respect to the chosen sample, and ϵ can be any positive value chosen by the data scientist, and represents the tolerance of error. This inequality says, simply put, that as the sample size N grows, it is *exponentially unlikely* that the realization ν deviates from μ by more than the tolerance ϵ. It can be clearly seen that the only the size of the sample N affects this bound. Consequently, to achieve a small tolerance ϵ, a large N has to be used.

Additionally, it is important to take into account the intrinsic noise of the systems, which can come from the nature of the input samples, i.e. the samples are not product of a deterministic system, or are immersed into some stochastic variation that can be modeled by noise, and that implies that by having the same input into the system *it is probable* get a different output. This entails a change in the labels of the model. That is, instead of having $y = f(x)$, we take y as a random variable resulting from a probability distribution $P(y|x)$. Accordingly, the input data points are therefore generated by the joint distribution $P(x, y) = P(x)P(y|x)$. With this description of the model, our target function (what $\mathcal{A}$ wants to learn) becomes $P(y|x)$, while $P(x)$ quantifies the importance of the input x in our learning accuracy.

2.2.2. Types of learning

There are three types of learning: supervised-, unsupervised-, and reinforcement learning. Each of them has specific characteristics, which are explained in the following subsections.

2.2.2.1. Supervised Learning

Is the type of learning where, in addition to the input dataset, the explicit correct outputs for those given inputs are given to the ML algorithm for training. There are two types of supervised learning:

- **Classification:** its main goal is to predict a *class label* from a determined set of choices. If the number of choices is two, the model corresponds to a *binary classification.* As it has only two options, it is suitable for problems whose expected answer is of the form "yes/no", "present/not present", "valid/invalid". For a greater number of classes the model corresponds to a *multiclass classification.* Examples of classification are:
  - Determining whether an email is spam or not constitutes a binary classification problem.
  - Identifying the zipcode from handwritten digits on an envelope is a multiclass classification problem.
  - Determine whether a tumor is benign based on size and shape data constitutes a binary classification problem
- **Regression:** its purpose is to predict a continuous behaviour, such as a trend, or a floating-number, and it is this continuity what sets it apart from the classification models. Examples of regression are:

– Predict the value of the stock market
– Determine the expected amount of crops yield from a plantation based on data such as previous yields, weather history, etc.

2.2.2.2. Unsupervised Learning

Unlike supervised learning, here the algorithms are feed with data but not with the expected outputs, which make this type of learning suitable for solving problems to which the output is unknown. The model is then in charge of extracting knowledge from the input data all by itself, without no further instructions. There are mainly two types of unsupervised learning that can be found in the literature [MG16], which are the *unsupervised transformations* and *clustering*.

- **Unsupervised transformations:** are models in charge of creating new representations of the input data, so that it becomes easier to understand and/or to use than the original data. This functionality is used, for example, to reduce the dimensionality of data that consists of several features. In such situation, the model transforms the data into a representation that summarizes the input with fewer features. Another important use of this type of models is finding the overall representation of the input data, such as the topic of a full text, or the sentiment in a short comment.
- **Clustering:** this kind of models group the data into determinate groups that share similar characteristics. This is used, for example, to generate suggestions based on previous purchases/views, or to group pictures from a directory with several images to the ones that contain certain people (and suggest tagging the names on them).

As the models do not know beforehand what type of information they are intended to learn, one of the tasks of the data scientist is to assess that the model is indeed learning something useful. This creates the opportunity for this sort of models to be used for the same data scientist to help them identify certain characteristics of the data that were not obvious for the *human-eye*, and certainly get a different perspective of the data from the ML model point-of-view.

2.2.2.3. Reinforcement Learning

This type of learning has a different approach to the previous two descriptions. Just like unsupervised learning, the model does not receive the expected outputs for the inputs it is given but, in contrast, it receives some output *possible* output along with a weight that states how good of an output it is. The idea behind this is that the model is then penalized when it provides a solution that is not according with the possible output, and is rewarded when it is. Then, the model uses this penalizations and rewards to adjust the type of outputs it generates, and so it eventually learns the correct behaviour for the situations it has been immerse into. This type of learning is similar to the way humans learn, in ways such as being penalized with pain when taking a sip of very hot coffee, or rewarded with winning a game of chess.

In that same manner, reinforcement learning comes in handy in teaching an intelligent agent how to play a game, where it is presented to a plethora of options (which makes it difficult to be modeled as a supervised learning problem) and has to choose the one that brings it near to victory. The most recent example is AlphaGo [Fu17], an intelligent agent capable of winning the world Champion on a Go game. A similar approach has been followed by IBM's with the Deep Blue chess-playing machine [Hsu99].

2.2.3. Training Models

2.2.4. Testing Models

2.2.5. Model Evaluation

2.2.5.1. Overfitting

2.2.5.2. Underfitting

2.2.6. Data preprocessing

The main requirement of ML is that data is the main requirement. In the same way, it is necessary to ensure that the data is valid and that information can be extracted from it.

2.2.6.1. Data transformation and scaling

2.2.7. Feature Engineering

UNSUPERVISED LEARNING CAN BE USED FOR FEAUTURE EXTRACTION andreas page 154 (PCA)

2.2.8. Machine Learning algorithms

2.2.8.1. K-nearest Neighbors

2.2.8.2. Support Vector Machines

2.2.8.3. Binary trees

2.3. Deep Learning

2.3.1. Neural Networks

2.3.2. Convolutional Neural Networks

2.4. Optimization of Cost Functions

3. Testbed Implementation

3.1. Software Defined Radio approach

3.1.1. GNURadio

3.1.2. Universal Software Radio Peripheral

3.2. Machine Learning models in Python and Jupyter

3.2.1. scikit-learn

3.2.2. keras

3.3. Data set Generation

3.3.1. Measure Campaign

3.3.2. Feature Engineering

3.3.3. Spectrograms generation

4. Evaluation and Results

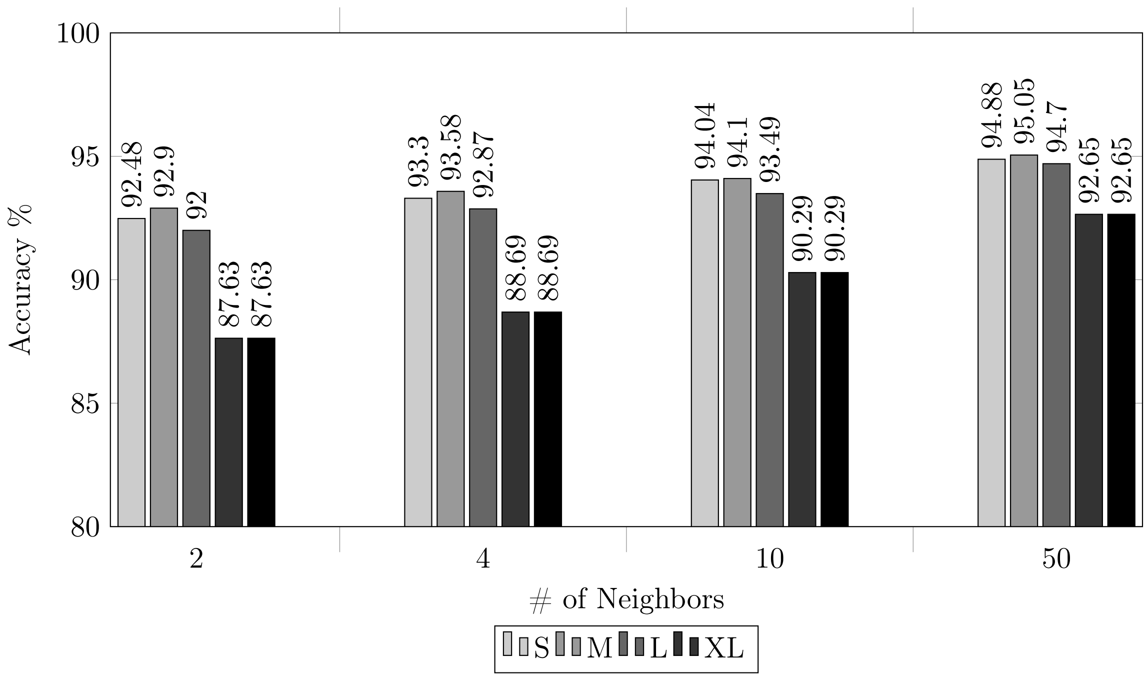


Figure 4.1.: Artificial Intelligence

4.1. Scenario Classification

4.2. Performance metrics

4.3. Dyspan setup comparison

5. Live implementation in GNURadio

6. Conclusion

So you made it! This is the last part of your thesis. Tell everyone what happened. You did something... and you could show that ... followed.
In the end make a personal statement. Why would one consider this thesis to be useful?
A pattern exists.
We cannot pin it down mathematically.
A data set.

A. Abbreviations

AI Artificial Intelligence

BW Bandwidth

CEL Communications Engineering Lab

ComSoc IEEE Communications Society

CR Cognitive Radio

DL Deep Learning

DySpan Dynamic Spectrum Access Networks

EMC IEEE Electromagnetic Compatibility Society

FCC Federal Coomunications Commision (USA)

LDPC Low-Density Parity-Check

ITU International Telecommunications Union

IEEE Institute of Electrical and Electronics Engineers

KIT Karlsruhe Institute of Technology

ML Machine Learning

OFCOM U.K. Offices of Communications

PU Primary User

SCC41 Standards Coordinating Committee 41

SDR Software-Defined Radio

SNR Signal-to-Noise-Ratio

SU Secundary User

Bibliography

[AMMIL10] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning From Data*, volume 21. AMLbook.com, 2010.

[BL07] James Bennett and Stan Lanning. The Netflix Prize. *KDD Cup and Workshop*, pages 3–6, 2007.

[BM01] Indranil Bose and Radha K. Mahapatra. Business data mining - A machine learning perspective. *Information and Management*, 39(3):211–225, dec 2001.

[Bro91] Rodney A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47(1-3):139–159, 1991.

[Cha] Dyspan Challenge. IEEE DySPAN | IEEE DySPAN.

[Com] Comsoc. ComSoc Conference Search | IEEE Communications Society.

[Fu17] Michael C. Fu. Alphago and Monte Carlo tree search: The simulation optimization perspective. *Proceedings - Winter Simulation Conference*, pages 659–670, 2017.

[GAMS] Andrea Goldsmith, Syed Ali Jafar, Ivana Mari, and Sudhir Srinivasa. Breaking Spectrum Gridlock with Cognitive Radios : An Information Theoretic Perspective.

[Goo17] Google Inc. Machine Learning – Explore – Google Trends, 2017.

[Haw] Jeff Hawkins. What Intelligente Machines Need to Learn From the Neocortex.

[Hay05] Simon Haykin. Cognitive Radio : Brain-Empowered. 23(2):201–220, 2005.

[HB04] Hawkins Jeff and Sandra Blakeslee. *On Intelligence*. Henry Holt and Co, 2004.

[Hoe63] Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, mar 1963.

[Hsu99] Feng Hsiung Hsu. IBM's Deep Blue chess grandmaster chips. *IEEE Micro*, 19(2):70–81, 1999.

[IEE15] IEEE DySPAN. IEEE DySPAN Standards Committee (DySPAN-SC), 2015.

[Jon05] Friedrich K Jondral. Software-Defined Radio—Basics and Evolution to Cognitive Radio. *EURASIP Journal on Wireless Communications and Networking*, 3:275–283, 2005.

[Kon01] Igor Kononenko. Machine learning for medical diagnosis: History, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89–109, aug 2001.

[KWS+15] Ankit Kaushik, Felix Wunsch, Andrej Sagainov, Nicolas Cuervo, Johannes Demel, Sebastian Koslowski, J Holger, and Friedrich Jondral. Spectrum Sharing for 5G Wireless Systems. (1):6–7, 2015.

[LMH+17] Alex Lackpour, Sean Mason, Chase Hamilton, David Tigreros, Matthew Giovannucci, Matthew Marcou, Yuqiao Liu, Marko Jacovic, and Kapil R. Dandekar. Design and implementation of the Secondary User-Enhanced Spectrum Sharing (SUESS) radio. In *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 1–2. IEEE, mar 2017.

[MG16] Andreas Mueller and Sarah Guido. *Introduction to Machine Learning with Python*, volume 53. 2016.

[MM99] Joseph Mitola and Gerald Q. Maguire. Cognitive radio: making software radios more personal. *IEEE Personal Communications*, 6(4):13–18, 1999.

[OCC16a] Timothy J. O'Shea, Johnathan Corgan, and T. Charles Clancy. Convolutional radio modulation recognition networks. *Communications in Computer and Information Science*, 629:213–226, 2016.

[OCC16b] Timothy J. O'Shea, Johnathan Corgan, and T. Charles Clancy. Unsupervised Representation Learning of Structured Radio Communication Signals. apr 2016.

[OEC17] Timothy J. O'Shea, Tugba Erpek, and T. Charles Clancy. Deep Learning Based MIMO Communications. pages 1–9, 2017.

[Oxfa] Oxford. intelligence | Definition of intelligence in English by Oxford Dictionaries.

[Oxfb] Oxford. learn | Definition of learn in English by Oxford Dictionaries.

[PSK+17] Francisco Paisana, Ahmed Selim, Maicon Kist, Pedro Alvarez, Justin Tallon, Christian Bluemm, Andre Puschmann, and Luiz Dasilva. Context-Aware Cognitive Radio Using Deep Learning. *IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN), Spectrum Challenge*, pages 1–2, 2017.

[PST+17] Stefanos Papadakis, Manolis Surligas, Kostis Triantafyllakis, George Vardakis, Athanasios Gkiolias, Nikolaos Karamolegos, and Antonis Makrogiannakis. An agile OFDM cognitive radio engine. In *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 1–2. IEEE, mar 2017.

[RPP07] Kyle B. Reed, James Patton, and Michael Peshkin. Replicating human-human physical interaction. *Proceedings - IEEE International Conference on Robotics and Automation*, (April):3615–3620, 2007.

[SW14] G. Staple and K. Werbach. The end of spectrum scarcity. *IEEE Spectrum*, 41(3):48–52, mar 2014.

[WKM+17] Felix Wunsch, Sebastian Koslowski, Sebastian Muller, Nicolas Cuervo, and Friedrich K. Jondral. A cognitive overlay system based on FBMC. *2017 IEEE International Symposium on Dynamic Spectrum Access Networks, DySPAN 2017*, pages 6–7, 2017.

[WN07] Zhiqiang Wu and Bala Natarajan. Interference tolerant agile cognitive radio maximize channel capacity of cognitive radio. In *2007 4th Annual IEEE Consumer Communications and Networking Conference, CCNC 2007*, pages 1027–1031. IEEE, jan 2007.

[WPR+17] Felix Wunsch, Francisco Paisana, Sreeraj Rajendran, Ahmed Selim, Pedro Alvarez, Sebastian Muller, Sebastian Koslowski, Bertold Van den Bergh, and Sofie Pollin. DySPAN Spectrum Challenge: situational awareness and opportunistic spectrum access benchmarked. *IEEE Transactions on Cognitive Communications and Networking*, pages 1–1, 2017.

[ZMJ16] Alessio Zappone, Bho Matthiesen, and Eduard A. Jorswieck. Energy-efficient MIMO overlay communications for device-To-device and cognitive radio systems. In *2016 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2016*, pages 115–120. IEEE, apr 2016.