Universidade do Minho

Escola de Engenharia – Disciplina de Aprendizagem e Extração de Conhecimento

Trabalho Prático 1

# Sistemas de Aprendizagem

Leandro Oliveira Freitas (ID6902)

Outubro, 2017

# Reinforcement Learning

Leandro Oliveira Freitas[1]

Escola de Engenharia. Disciplina de Aprendizagem e Extração de Conhecimento.

Universidade do Minho, Braga, Portugal

[1]leanfrts@gmail.com

## 1. Descriptive characteristic

Reinforcement learning is a Machine Learning algorithm that aims to improve the performance of systems through the interaction between them and computing methods. (Sutton and Barto, 2012). The main idea is to reward a good result of processing procedures. With this the machine tries to improve its performance aiming to be rewarded. Instead of defining a sequence of methods to be processed, the system should try all available options in order to find the best way to solve a problem. Each path for the solution may have a value attached to it and with this it is possible to create ranks. Thus, every time the system finds a satisfactory solution, it is rewarded with numerical signals. This means that the better is the solution, the better will be the rewarding.

This creates an interaction where the system learns what are the best ways to deal with a problem. As long as the system gets rewards it will always try to find the best solutions, and consequently, the tendency is that its performance will be improved along the time. Depending on the modelling of the system, it also be able to find good solutions for future situation, not only for the current task that is being processed (Sutton and Barto, 2012).

Comparing this method with other machine learning alternatives, it may need a longer time to find the best solution for a problem. Depending on the problem that needs to be solved, the system may need a long time to try all possible alternatives and, consequently, it will need a longer time to learn.

In order to analyse all the options that can be used to solve a problem, the software agent uses a methodology called situation-action maps (Shapiro, et al, 2001), to specify Markov Decision Processes (MDP). The situation step refers to the identification of important aspects of the problem that will be solved. After this definition, the software agent starts to act, using the aspects as input. It must be able to interfere in the state of the fact, to provide a solution. The last part refers to the goals that the process will have. To be able to optimize a solution, the learning agent must know what it is heading to. The importance of this relays on the fact that it needs to know what is a solution in order to test all the possibilities to achieve it.

## 2. Learning capacity

According to Lee and Anderson (2017), the exchanging of knowledge can be applied between only two tasks when using reinforcement learning. Thus, one of the problems faced by these algorithms is the fact that it should use supervised learning approach (Lee and Anderson, 2017), i.e., the agent must be aware what is the output solution to test the options that lead to it. For instance, in context-aware computing, there several situations where the system does not know what should be the output. This happens due to its dynamicity, with states of entities changing fast in a small period of time. Context-aware systems use machine learning algorithms aiming to predict possible actions of users, based on what is known about his behaviour. However, as their behaviour may change inexpertly, it is not possible to provide a possible output for the system to learn. In these cases, an unsupervised approach is more adequate.

According to Sutton and Barto (2012), there are two main approaches to implement reinforcement learning: SARSA and Q-Learning. SARSA is used to build a hybrid learning system with queuing network model in Tesauro, et. al, (2006).

Besides the good results of the application of reinforcement learning to find feasible solutions for a problem, this approach presents a big disadvantage. Depending on the number of options of actions, the agent may take a long time to find the most suitable path, when applied to real world problems (Arabnejad, 2017).

## 3. Development tools

The evolution of hardware and software technologies is helping the proposal of new development tools for the consolidation of systems with a more improved artificial intelligence. Algorithms for machine learning are directly related to this scenario, and among the approaches, the reinforcement learning appears as a feasible alternative.

The Reinforcement-Learning-Toolkit 1.01 is collection of tool for the Python programming language and was developed by the Reinforcement Learning and Artificial Intelligence (RLAI) group. The focus is to help users with different goals (learners and teachers) to use this approach. The toolkit is multiplatform, available for Linux, Windows and Mac OS. The documentation and installation instructions are available in the website of the project.

An alternative to it is the OpenAI Gym Beta2, a toolkit for development and comparing reinforcement learning solutions, developed by the company OpenAI. The toolkit has a large range of environments. Currently it is available only for Python, however, according to the website of the project it is intended to expand the kit for other languages. Besides the instructions for installing the toolkit, the website also presents a good justification for why choosing reinforcement learning approach. The has a good description of available environments and the validations of them.

## 4. Market solutions

There are many companies that are adopting machine learning algorithms to improve their production and, consequently, their profit. The site chatbots magazine3 presents an article with several business areas that are being benefit from reinforcement learning.

In the manufacturing field, robots and reinforcement learning are being used to move objects from one place to another. They learn even when they do not succeed from it and with the knowledge obtained they improve the performance of their function.

Besides that, companies like Tesla, Google and Uber are investing in researches about self-driving vehicles, using reinforcement learning, among other approaches. Part of the process includes a supervised learning, where it is necessary to analyse all the possible next positions of the car and to find the best path to reach a destination.

## 5. Conclusion

There are a large range of Artificial Intelligence approaches, each of them may be more suitable for a specific situation. Machine learning algorithms are being used not only to automate a process, but also to improve it the best way possible. Supervised learning aims to help software agents to learn from a situation using inputs datasets to reason and an expected output as parameter.

Reinforcement learning is a sort of supervised learning and has as main objective to test all possible, and feasible, alternatives to reach a reach a goal. In exchange of this the software agent is rewarded, according to its performance. If it achieves a good result, it gets a positive feedback. If not, it gets a negative feedback. The more the agent is rewarded, the more it learns and, consequently, the best will be its performance. Thus, the acquired experience can help the agent to learn faster.

This paper presented an overview of reinforcement learning, describing its concepts and characteristics, learning capacities and, also, tools and solutions available nowadays. Though some limitations of its usage, like not being able to be applied in dynamic fields such as context-aware systems, reinforcement learning is a very good solution for automation of processes in a diversity of business fields.

[1] http://incompleteideas.net/rlai.cs.ualberta.ca/RLAI/RLtoolkit/RLtoolkit1.0.html
[2] https://blog.openai.com/openai-gym-beta/
[3] https://chatbotsmagazine.com/reinforcement-learning-and-its-practical-applications-8499e60cf751

## References


1. Richard S. Sutton and Andrew G. Barto, “Reinforcement Learning: An Introduction”, The MIT Press, 2nd edition, 2012.
2. D. Shapiro, P. Langley and R. Shachter, “Using Background Knowledge to Speed Reinforcement Learning in Physical Agents”. Proceedings of the Fifth International Conference on Autonomous Agents, 2001.
3. M. Lee and W. Anderson, “Can A Reinforcement Learning Agent Practice Before It Starts Learning?”. International Joint Conference on Neural Networks, Anchorage, USA. 2017.
4. G. Tesauro, N.K. Jong, R. Das, M.N. Bennani, “A hybrid reinforcement learning approach to autonomic resource allocation”, IEEE Intl Conference on Autonomic Computing, pp. 65-73, 2006.
5. H. Arabnejad, C. Pahl, P. Jamshidi and G. Estrada, “A Comparison of Reinforcement Learning Techniques for Fuzzy Cloud Auto-Scaling,” 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Madrid, 2017, pp. 64-73. doi: 10.1109/CCGRID.2017.15.

# Artificial Neural Networks

Leandro Oliveira Freitas[1]

Escola de Engenharia: Disciplina de Aprendizagem e Extração de Conhecimento

Universidade do Minho, Braga, Portugal

[1]leanfrts@gmail.com

## 1. Descriptive characteristic

Artificial Neural Networks (ANN) is an approach of machine learning in which the algorithms are applied to work using the same idea of the human brain. According to Staub et al. (2015) ANNs uses networks to acquire data and store it through artificial neurons. Systems developed with this technique uses artificial neurons to analyse samples from a dataset aiming to find the probability of each possible output be classified within one of the available classes. According to Haykins (1999), Artificial Neural Networks can be defined as a set of distributed processors composed by units capable of storing and providing knowledge to be used. The architecture of ANNs is represented in figure 1:

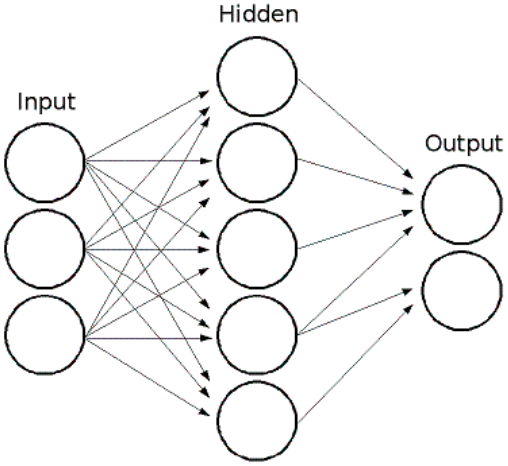


**Figure 1:** generic architecture of Artificial Neural Networks.

According to Staub et al. (2015) the architecture of ANNs is composed by the following layers:

- Input: refers to the set of data that needs to be analysed. This dataset can be organized in groups, which represents the neurons of the network. The number of input neurons must be the same of input data;
- Hidden: represents function for the processing of the input data;
- Output: represents the learning process and decision making. It creates an output and, as well as in the input layer, the number of neurons is the same of the number of output dataset.

Based on this generic architecture, several others were developed. For instance, the Perceptron architecture does not have the hidden layer, and two input neurons and only one output. On the other hand, Multilayer Perceptron can use several hidden layers. In the Recurrent architecture, the hidden layer may have self-connection and can be activated by others.

Considering the above architecture, figure 2 presents the nonlinear model of neurons of ANNs (Haykin, 1999):

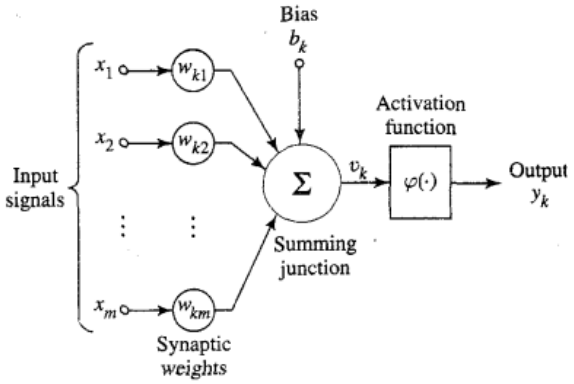


**Figure 2:** Nonlinear model of neurons (Haykin, 1999).

In this model the x1 to xm represents the input layer, the xk1 to xkm represents a connection component between the input and hidden layers, with the assign of weights for each data, the Summing Junction refers to the adjustment of the input net, depending of the Activation Function refers to a connection with the output layer, and yk represents the output neuron.

## 2. Learning capacity

Once its reference to human brains, Artificial Neural Networks must present several benefits when compared to other machine learning algorithms. ANNs have a considered processing power due to parallel processing structure and due to the learning capabilities (Haykin, 1999).

Following it is described some of the benefits of Artificial Neural Networks (Haykin, 1999; Staub et al. 2015):

- Nonlinearity: besides the linear features, it can assume nonlinear characteristics through interconnection of neurons that are nonlinear. It can be distributed all over the network;
- Input-output mapping: ANNs can be used through supervised learning approaches, where, it is provided an input set of data and an expected output. From this, the system trains the data until achieve a satisfactory result, considering the output;
- Adaptatively/ contextual information: the weights defined in an ANN can be adapted considering the changes in the domain where it is being applied. With this, the dataset can be retrained. Context information is easily handled by neural networks. This feature is very important when developing applications for context-aware systems, where any change in the environment can influence the behaviour of the system;
- Evidential response: ANNs can be used not only to provide a solution for a problem, but also, to analyse the reliability of the output information;
- Fault tolerance: the distributed nature of ANNs allows them to have a high level of fault tolerance, especially if implemented in hardware form. If the data from a neuron is compromised, the damage to the rest of the network can be minimized.

## 3. Development tools

There are several tools for the development and analysis of Artificial Neural Networks. For instance, SIMBRAIN[1] is a free tool, multiplatform, that has a visual component for analysis of processing tasks with ANNs. It has a good documentation with API. One of the main features of the tool is the possibility of integration with "world components" allowing the developer to perform more realistic tests and the ability of represent the state space of the network. Figure 3 presents a screenshot of the tool available on the site.

[1] http://www.simbrain.net/

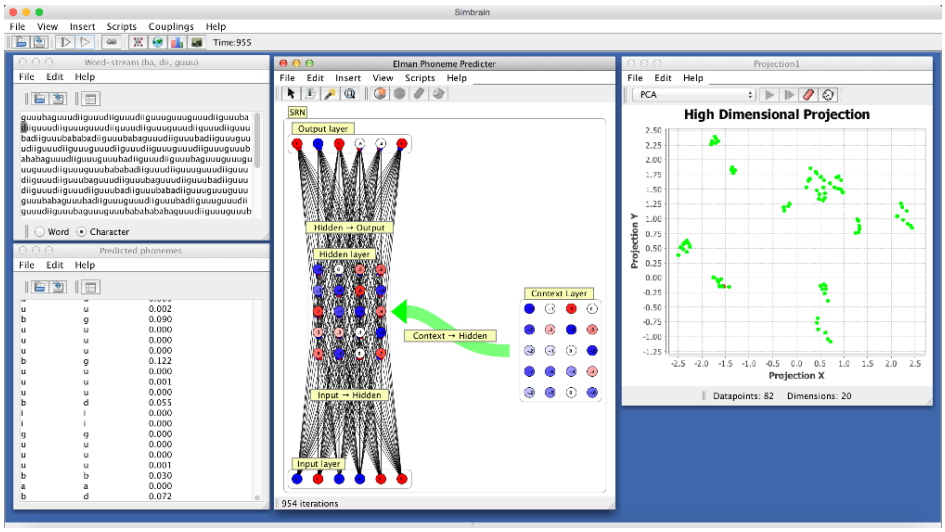


**Figure 3:** SIMBRAIN development tool. Retrieved from: (http://www.simbrain.net/Screenshots/Simbrain3/TextWorld.png).

Besides that, there are packages for the most used programming languages for development of ANNs algorithms. For instance, Neural Network Toolbox[2] for Matlab, Java Neuro Network Framework[3] for JAVA and Pybrain[4] for Python.

## 4. Market solutions

Due to the powerful characteristics, several companies, from different business areas are investing in researches with Artificial Neural Networks. A company named minds.ai[5] has developed several solutions with this machine learning technique. For instance, they created a software to automatize the process of tracking and identification of cars in footages for an automotive company. The main idea was to improve the job of humans to a level where they have only to verify the reliability of the labels.

The company GMDH Data Science[6] also has been using Artificial Neural Networks to develop solutions for its clients. The website of them has a list of the companies that are using their software. Among the features of the software is the analysis of data for prediction of sports, business and stock market.

## 5. Conclusion

Artificial Intelligence is becoming part of our life, gradually. Machine learning algorithms are to train or analyse a set of data, making computers learn and, consequently, improve their knowledge about a specific domain.

One of the most powerful machine learning approaches is the Artificial Neuro Networks and its related area of Deep Learning. The main idea is to simulate the human brain with a network composed by neurons capable of processing datasets by analysing synaptic with weights. The results of the used of this technique is proved to be reliable and is topic of research in the biggest Information Technology companies.

This paper presented an overview about Artificial Neural Networks, with some concepts, its generic and most used architectures, and also some of the benefits of it. Besides that, it presented some of the development tools and market solutions that use this techinique.

---

[2] https://www.mathworks.com/products/neural-network.html
[3] http://neuroph.sourceforge.net/
[4] http://pybrain.org/pages/features
[5] https://www.minds.ai/neural-networks
[6] https://gmdhsoftware.com/neural-network-software

## References


1. Haykin, S., “Neural Networks – A Comprehensive Foundation”, Prentice-Hall, New Jersey, 2nd Edition, 1999.
2. Staub S, Karaman E, Kaya S, Karapinar H, Güven E. “Artificial Neural Network and Agility”. World Conference on Technology, Innovation and Entrepreneurship. Procedia - Social and Behavioral Sciences 195 ( 2015 ) 1477 – 1485.

# Support Vector Machine

Leandro Oliveira Freitas[1]

Escola de Engenharia. Disciplina de Aprendizagem e Extração de Conhecimento.

Universidade do Minho, Braga, Portugal

[1]leanfrts@gmail.com

## 1. Descriptive characteristic

The main idea of Support Vector Machine algorithms is to divide two classes of points with a hyperline separation This line should have the same distance from the closest point in each of the classes. The power of generalization reasoning depends on the accuracy of the distance (Cristianini and Shawe-Taylor, 2000). Once having the maximum distance from the closest point to the hyperline it is possible to classify all points from the two classes with high safety (Cristianini and Shawe-Taylor, 2000).

The SVM technique uses supervised learning approach, where it is necessary to provide the input dataset (labels) and also an expected output. The output can be defined as classifiers, models or hypothesis. With these parameters the algorithm trains the input aiming to achieve the closest result as possible from the expected output. The input data refers to the domain in which it is needed to analyse and make predictions (Lorena and Carvalho, 2007).

SVM algorithms are able to deal with problems of overfitting through an explicit optimization of statistical bounds (Cristianini and Shawe-Taylor, 2000). This refers to the low rate of success by analysing a new set of data. The more data provided for analysis, the lower is the risk of overfitting. Besides that, the reduction of misclassification penalty also helps to minimize the chances of overfitting (Lorena and Carvalho, 2007).

Support Vector Machines is one of the most used technique of machine learning due to several reasons, including the high level of success and fast performance. Its results can be compared, in terms of performance, to other techniques like Artificial Neural Networks. SVMs are widely used in different fields of research, such as text categorization, image analysis and bioinformatics (Lorena and Carvalho, 2007).

## 2. Learning capacity

SVMs are used to train datasets through the theory of statistical learning and are able to analyse it and predict the class for new data (Lorena and Carvalho, 2007). One of the most practical uses of SVM is in the pattern recognition field due to the possibility of training data and finding the optimal solution in polynomial time (Cristianini and Shawe-Taylor, 2000). Induction principle of inference is able reason over a set of data and derive generic conclusions (Lorena and Carvalho, 2007). Also, it is widely applied in the training of datasets that aims an output with results beyond the "yes/ no" classification, using the Regression approach (Boswell, 2002).

SVM algorithms are modular, i.e., uses a dimensional space where is possible to separate the data, through the notion of "kernel induced feature space" (Boswell, 2002). Different from other machine learning algorithms, the probability of success of reasoning can be explicitly calculated with SVM algorithms (Boswell, 2002).

The classifier refers to a function (f) that considers the input data (x), calculates it and achieve a conclusion (y). Thus, its generic form can be represented by f(x,y) (Lorena and Carvalho, 2007).

SVM are allows the identification specific data inside the set that should be ignored during the training process. This data refers to imprecise or incorrect attributes. The analysis of them could lead to unsatisfactory results (Lorena and Carvalho, 2007).

## 3. Development tools

There are a wide range of development tools available to be used with Support Vector Machines algorithms. Following, some of them are described:

- PyMVPA[1] is a package for Python programming language that contains an extensible framework for statistical leaning analyses. The website contains instruction for installation, a tutorial, documentation and e-mail address for support. According to the site it has algorithms for implementation of classification, regression, feature selection and data import and export;
- Shogun Toolbox API[2] has a set of examples available in several programming languages like Python, Java, R and C#. It uses kernel and linear Support Vector Machine approaches to implement binary classification;
- The Modular Toolkit for Data Processing (MDP)[3] is a package with different machine learning algorithms, including supervised and unsupervised approaches, that has as main advantage to combine them as a pipeline to improve the performance of tasks;
- PyBrain[4] is a library with machine learning techniques for Python programming language. It contains a package, *svmunit*, for implementation of SVM algorithms. Different from MDP, the developers advise user to not combine it with other models from the project. The main idea of this package is the comparison of feed-forward methods of classification. Based on information from the site it is possible to conclude that it has restrictions when compare with other development tools;
- Machine Learning Py (mlpy)[5] is another tool for Python available for the two versions of the programming language with algorithms to implement supervised and unsupervised learning. For dealing with problems through SVM is allow the implementation of regression and classification algorithms;
- The scikit-learn[6] project provides support for every sort of input samples. The website contains a good documentation about the tools, including a brief explanation about SVM, with advantages and disadvantages of this technique.

Some of these implementations were analysed and compared each other by the site of *scikit-learn*. For the benchmark it was used two datasets as input, Madelon and Arcene. Pybrain and MDP did not appear in the result of the comparison. Figure 1 shows the result:

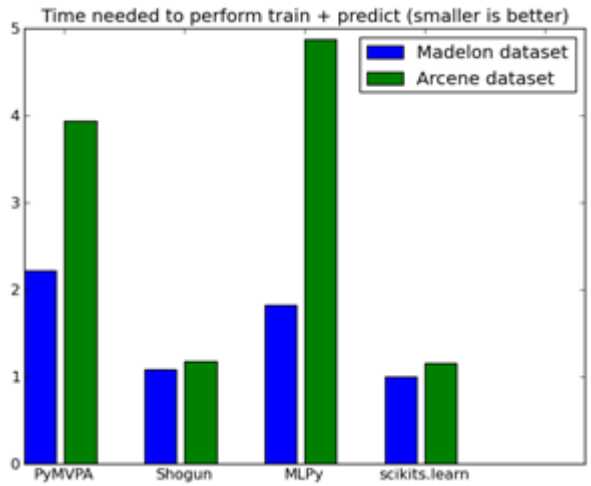


**Figure 1:** benchmark performed by scikit.learn. Adapted from: http://scikit-learn.sourceforge.net/ml-benchmarks/#support-vector-machines.

[1] http://www.pymvpa.org/index.html
[2] http://www.shogun-toolbox.org/examples/latest/index.html
[3] http://mdp-toolkit.sourceforge.net/api/mdp-module.html
[4] http://pybrain.org/docs/api/supervised/svm.html
[5] http://mlpy.sourceforge.net/
[6] http://scikit-learn.org/stable/modules/svm.html

## 4. Market solutions

Support Vector Machine is a machine learning technique with very good performance and proved to achieve good classification and prediction results, when compare to other approaches. Thus, companies from different business fields are using it in their solutions. FICO [7] is a company of analytic software with a wide range of solutions. It uses SVM algorithms in their software solutions, among other machine learning approaches. According to the site of the company, the main focus is the analysis of costumer behaviour and data analysis trying to maximize the profit, including management of risk and fraud, and profitable customer relationships, always aiming the minimization of costs.

TIBC Statistica[8] is another also an analytic software company that uses SVM in their products. According to the website of the company, their range of solutions vary from drag and drop softwares for Data Scientists without the need of dealing with programming, scripting in R and Python programming language, until business solutions like quality control and design of experiments for industrial fields, imbedded analysis and development of flexible workflow models to standardize processes. At last, the company also uses machine learning algorithms for the analyses of feasibility of deployment of Internet of Things devices.

## 5. Conclusion

Support Vector Machines are one of the most used algorithms of machine learning due to its good performance and results. It is widely used in different fields for classification of data and predictions, from scientific researches to business solutions. This approach needs as parameters a dataset as input and an expected output for the training. To apply this algorithm, it is necessary to create a hyperline dividing the dataset in two classes to be analysed. One of the main advantages is the prospect of manage the overfitting of the classification.

This paper presented an overview about Support Vector Machines, describing the their concepts, the main learning capabilities, some of the development tools available and a few market solutions.

## References


1. Boswell, D.: Introduction to Support Vector Machines. August, (2002).
2. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, (2000).
3. Lorena, A. C., Carvalho, A. C. P. L. F.: Uma Introdução às Support Vector Machines. Revista de Informática Teórica e Aplicada (RITA), 14(2), 43-67, (2007).


[7] http://www.fico.com
[8] http://statistica.io/