Review of machine learning techniques and their applications in robotics.

AUTHOR *

* *Tiago Trocoli (Master's Student)*
RA: 226078
E-mail: t226078@dac.unicamp.br

***Resumo* – This article gives a brief overview of machine learning techniques, types, examples of their applications in robotic area. The first section gives the defnition of machine learning and its importance in robotics fields. The second section describes supervised learning, unsupervised and reinforcement learning along with robotic applications. The third section gives a mathematical overview of two famous algorithm: multilayer perceptron and support vector machine with their application in robotic area. This review concludes that although artificial intelligence has been improved a lot in robotics it still far from its potential to help society that has been brilliantly shown in sci-fi movies, where robotics are humanoid that helps police, know how to talk, walk, run and so on.**

***Palavras-chave* – support vector machine, neural networks, supervised learning, unsupervides learning, reinforcement learning, application in robotics.**

I. INTRODUCTION

Machine learning (ML) is a type of algorithm that performs a task without being explicitly programmed [1]. To do so, it trains a model, that is, a mathematical function that will recognize patterns in data. Data could be images, signals, audio, texts or any vector of number that could, for instance, represent a social-economic data of a country. Moreover, ML is divided into three classes as shown in figure 1, in which each of one will be briefly explained. ML is also a hot area that has been applied in various fields of knowledge, among them is robotics in which this article focus on.

In robotics field, researchers from Carnegie Mellon University are building a systems capable of finding and assessing faults in silicon wafers using convolution neural networks [2]. A collaboration between researcher from UC Berkeyley Center for Automation and Learning for Medical and Robotics and physicians led to creation of Smart Tissue Autonomous Robot (STAR) [3]. According to them star has making surgeons with a better precision and reliability than the best human surgeons [4]. In 2014, MIT's Lab for Information and Decision Systems build build multi-agent learning using robots [5]. In this system, robots collaborated to build a better and more inclusive learning model than could be done with one robot.

II. MACHINE LEARNING TYPES

Supervised learning is a type of machine learning algorithm that learns a function which maps data to their class. A model, that is, a mathematical function is developed based on labelled training data consisting of a set of training examples. That model is expected to separate data based on their classes. A class is the "type"of the data, for example, we could have data composed of images from bananas, apples and melons. These three fruits are the data classes that could be classified automatically by training a machine learning algorithm. The type of function is usually provided by the algorithm. For example, in the classical Support Vector Machine is a linear function and in Logistic Regression could be polynomial function. The learning phase is composed of adjusting the model parameters so to better maps data into their respective classes.

Supervised learning algorithms are numerous. For example, Support vector machines that classifies data based on a linear function, that is, a hyperplane that separates two classes of data. Logistic regression, that uses polynomial function as a model. Neural Network inspired by nature and is a famous nonlinear classifier widely used in robotics. The K-nearest-neighborhood that is based on data similarities to classify them. Linear discriminant analysis that uses a linear combination of features that characterizes or separates two or more classes of objects or events.

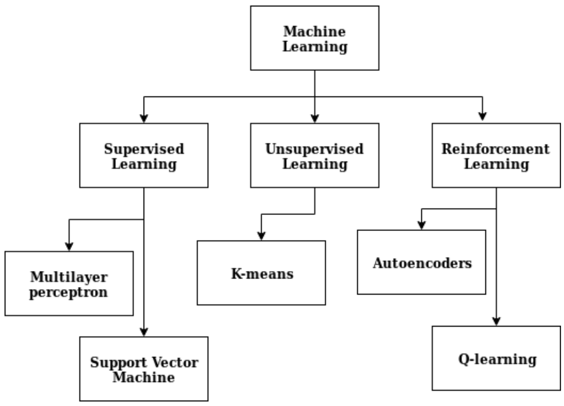

Figura 1. Machine Learning areas.

Different from supervised learning that requires data classes (or labels), in unsupervised learning the data are unlabeled. Thus, these methods tries to find a way to group data. In clustering analysis, data are grouped according to a similarity measurement. For example, in k-means, shown in figure 2, the algorithms finds centroids in which each data belongs to. These centroids are created according to similarities among data in a region of hyperspace. After creating centorids, any new data can be classified by applying a defined similarity measurement such Euclidean distance.

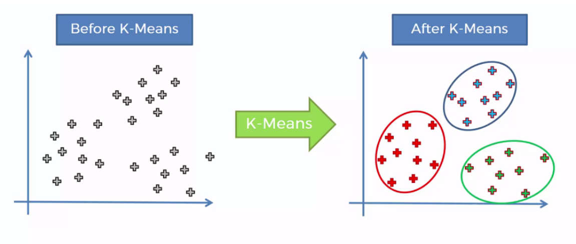

Figura 2. K-means methods in action.

Reinforcement learning (RL) is another fields of Machine Learning in which software agents need to take action so to maximize its performance. Thus, a simple reward feedback is required for the agent to learn its behaviour. This reward feedback is a cost function that need to be maximized. Moreover, RL methods rely on dynamic programming that is based on a predefined exact mathematical model that maximize its rewards. However, RL doesn't have an exact model and it generally can find local optimal solution. For example, Q-learning is a famous RL method that doesn't require to have a model of the environment. It tries to find the best policy of Markov chain process that maximize a cost function.

III. METHODS

A. Artificial Neural Network

Artificial neural network is a class of algorithm inspired by neural networks. In computer science it is frequently represented as a graph made up of layers in which each of them is composed of a column of nodes. Each column of nodes (layer) is connected by one another through edges. The first column is where the input (data) is inserted into ANN, called input layer. The output layer is where the results from ANN operations is outputted. Hidden layers are layers between the input and output ones, where ANN operations take place. Figure 3 shows a graph representation of ANN.

In robotics, ANNs are used in image classification, object detection and control including computer numerical control.. However, they also can be used in function approximation, regression analysis, including time series prediction, fitness approximation, signal processing and modeling [6]. ANN can also be used with adaptive filters such as Kalman filter, common applied in robotic area.

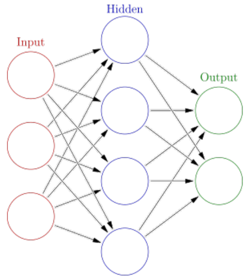

Figura 3. Artificial Neural Network Topology.

Examples of artificial neural network in robotics are numerous. Zhang and Yuan [7] developed a multi-sensor data fusion algorithm based on Fuzzy Neural Networks applied to wall following behavior. They showed that this method is practical and effective. Hexi Li, Jihua Li and Xinle Han [8] overcame a practical robot vision problem by developing three types of artificial neural networks corresponding to shape, color and texture of images with Dempster-Shafer theory to improve readability of robot vision. Bugeja, Fabri and Camilleri [9] proposed dual adaptive neural control for the dynamic control of nonholonomic mobile robots. They attested the effectiveness of this method by applying it in trajectory-tracking problem which resulted in a major improvement in tracking performance, despite the plant uncertainty and unmodeled dynamics.

1) Mathematical Model: ANN is based on a series of operations in hidden layers. Basically, each neuron has an activation function $h(\cdot)$ (its output) given an input $\mathbf{x} \in \mathbb{R}^n$, which is a linear combination of neurons' output of the previous layer, as shown in figure III-A1.

$$y = h(z) \text{ where } z = \sum_{i=1}^{m} w_i y_i + w_0, \tag{1}$$

in which a more compact way is to define $\mathbf{w} = [w_0, w_1, ..., w_p]^T$ and $\mathbf{y} = [1, y_1, y_2, ..., y_p]^T$, thus $y = h(\mathbf{w}^T\mathbf{y})$. However, in a multilayer neural network each neuron of a hidden layer as a slightly different equation given below:

$$y_i^l = h(z_i^l) \text{ where } z_i^l = \sum_{k=1}^{m^l} w_{i,k}^l y_k^{l-1} + w_{i,0}^l, \tag{2}$$

in which m^l is the number of neurons in layer $l-1$, y_i^l is the output of neuron i of layer l and $w_{i,k}^l$ is the weight of neuron i of layer l from neuron k in layer $l-1$.

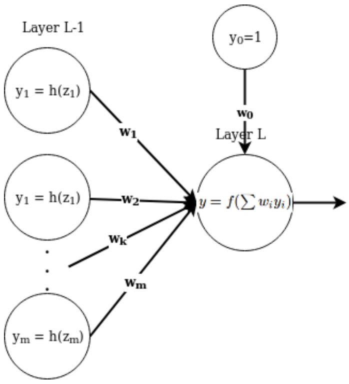

Figura 4. Artificial Neural Network Activation Function.

Many different forms of activation function exist in literature, as shown in figure 5. May be the most widely known is sigmoid function in which it generates a s-shape curve. Its codomain is $]0, 1[$, and as the input goes to positive side, the results closes to 1, otherwise, it closes to 0. When input is 0, results 0.5. Another one, is the hyperboloic tangent function, that also generates a s-shape curve but maps its inputs into values $] - 1, 1[$. Rectified linear units also referred to as ReLUs, utilize a specialized ramp function for the activation. A threshold function is a function which generates the output 1 if the input exceeds a certain value. Otherwise, this function takes the value 0.

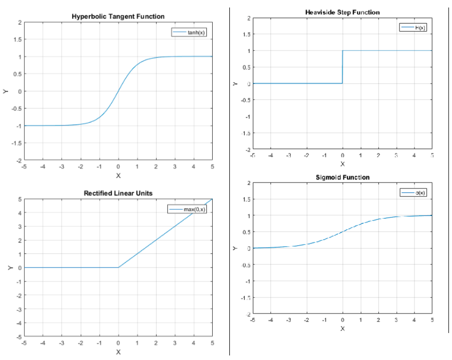

Figura 5. Artificial Neural Network Activation Function.

Until now, we know the equation of a single neuron, so this section derives a general ANN algorithm. Supposing a list of data $(\mathbf{x}_i, s_i)$, $i = 1, 2, ..., n$, which $\mathbf{x}_i$ is the data and s_i is its class. The number of class is p. Let's define an artificial neural network with L hidden layers with m^l neurons for each layer. ANN tries to adjust its weights to better classify data. So, data are inserted into ANN which maps each of them into a class. Let's define $\hat{s}_i$ the estimation of data i that ANN mapped into. The ANN needs to adjust its weights so that the errors $(s_i - \hat{s}_i)^2$ is the minimum possible, in other words

$$\text{minimize} \sum_{i=1}^{n} (s_i - \hat{s}_i)^2, \tag{3}$$

but we already know that $\hat{s}_i$ is the output, that is, $\hat{s}_i = h(z_i^L)$. Thus, all we need to do is to apply a nonlinear optimization algorithm to find the best weights possible. By "best"weights, we mean the ones that minimizes the problem 3.

Algorithm 1 Artificial Neural Network

Require: X = matrix which each row is a data. $h(\cdot)$ = activation function. m^l = number of neurons in layer l.

Let $Y^0 = X$ para $k = 1, 2, ..., m^1$.
for each hidden layer $l = 1, 2, ...L$ **do**
  for each neuron i in layer l **do**
    $\mathbf{z}_i^l = Y^{l-1}\mathbf{w}_i^l$
    $\mathbf{y}_i^l = h(\mathbf{z}_i^l)$.
  end for
end for
Let W be matrix of all weights.
$\mathbf{W} = \underset{\mathbf{W}}{\text{argmin}} \sum_{i=1}^{n} (s_i - h(\mathbf{z}_i^L))^2$

B. *Support Vector Machine*

The Support Vector Machine (SVM) is a well-known machine learning method that has been applied to a wide range of application in which robot is included. The first version of SVM was invented by Vladimir N. Vapnik and Alexey Ya. Chervonenkis in 1963. However, Vapnik suggested a nonlinear classifier version by applying the kernel trick to maximum-margin hyperplanes. The current version of SVM was proposed by Corinna Cortes and Vapnik in 1993 but published in 1995.

This section mentions papers about SVM applied in robotics. Schuurman and Ubbens [10] experimentally shows that SVM in vision-based obstacle detection demonstrates to be a high accuracy method. In the experiment, a single camera is mounted on the front of a mobile robot and an SVM is trained to classify obstacles as they are encountered by the robot. Nishikanto, Nusrat and Saiful [11] develop a SVM for real time vision based Human Robot Interaction. They use SVM in robot vision to detect classify facial expression and gender. For the prediction of any expression, facial images are taken in real time and provided the facial landmarks data to SVM. The robot classifies expression such as sad, angry, smile, surprise and normal with high accurate.

1) Mathematical Formulation: Suppose a data set is composed of two classes (C_1 and C_2), in which $\mathbf{x} \in \mathbb{R}^n$. The objective is to find a linear separetor, that, which is a hyperplane of the form

$$H(\mathbf{w}, \beta) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{w}^T\mathbf{x} + \beta = 0\},$$

for which the two classes are in hyperplane opposites sides, that is:

$$\mathbf{w}^T\mathbf{x} + \beta < 0, \; \forall\mathbf{x} \in C_1$$

$$\mathbf{w}^T\mathbf{x} + \beta > 0, \; \forall\mathbf{x} \in C_2$$

If we suppose the problem is *linearly separable*, meaning that the two set of points can be separated by a hyperplane, we could find a hyperplane that separates the two sets and is the farthest as possible from all points. In other words, we want to find a separator that is most distant to its closest point (margin) as shown in figure 6. To compute the margin, we need to have a formula for the distance between a point and a hyperplane. Thus, given an equation of hyperplane $H(\mathbf{a}, b)$, where $\mathbf{a} \neq 0, \mathbf{a} \in \mathbb{R}^n$ and $b \in \mathbb{R}$. The the distance between a point $\mathbf{y} \in \mathbb{R}^n$ and hyperplane is given by

$$d(\mathbf{y}, H(\mathbf{a}, b)) = \frac{|\mathbf{a}^T\mathbf{y} - b|}{||\mathbf{a}||}.$$

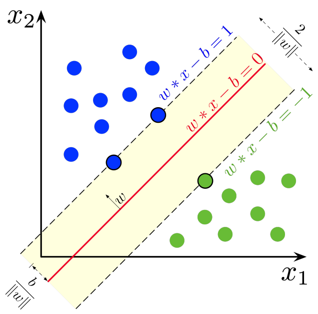

Figura 6. Margins of Support Vector Machine

Therefore, we conclude the margin correspond to a hyperplane $H(\mathbf{w}, -\beta)$, $\mathbf{w} \neq 0$ is

$$\min_{i=1,2,\ldots,n} \frac{|\mathbf{w}^T\mathbf{x}_i + \beta|}{||\mathbf{w}||}.$$

Thus, the problem that we consider is therefore

$$\begin{aligned} &\max_{(\mathbf{w},-\beta)} \Big(\min_{i=1,2,\ldots,n} \frac{|\mathbf{w}^T\mathbf{x}_i + \beta|}{||\mathbf{w}||}\Big) \\ &s.t.\ \mathbf{w}^T\mathbf{x} + \beta < 0,\ \forall \mathbf{x} \in C_1, \\ &\qquad \mathbf{w}^T\mathbf{x} + \beta > 0,\ \forall \mathbf{x} \in C_2, \end{aligned} \tag{4}$$

in other words, we want to find a hyperplane that has the maximum margin. However, is is bad model since its not convex and cannot be easily solved. So, we want to find a convex model of the problem. Note that if (w, β) is an optimal solution, then so is any nonzero multiplier of it, that is, $(\alpha w, \alpha\beta)$ for $\alpha \neq 0$. We can therefore decides that

$$\min_{i=1,2,\ldots,n} |\mathbf{w}^T\mathbf{x}_i + \beta| = 1, \tag{5}$$

so the problem becomes

$$\begin{aligned} &\max \frac{1}{||w||} \\ &s.t. \min_{i=1,2,\ldots,n} |\mathbf{w}^T\mathbf{x}_i + \beta| = 1 \\ &\qquad \mathbf{w}^T\mathbf{x} + \beta < 0,\ \forall \mathbf{x} \in C_1, \\ &\qquad \mathbf{w}^T\mathbf{x} + \beta > 0,\ \forall \mathbf{x} \in C_2. \end{aligned} \tag{6}$$

The maximization of $1/||w||$ is the same of minimization $(1/2)||w||^2$. The combination of the first equality and the two inequality constraints implies that a valid reformulation is

$$\begin{aligned} &\min \frac{1}{2}||w||^2 \\ &s.t.\ \mathbf{w}^T\mathbf{x} + \beta < -1,\ \forall \mathbf{x} \in C_1, \\ &\qquad \mathbf{w}^T\mathbf{x} + \beta > 1,\ \forall \mathbf{x} \in C_2. \end{aligned} \tag{7}$$

We removed equality constraint 5 since any feasible solution satisfies $\min_{i=1,2,\ldots,n} |\mathbf{w}^T\mathbf{x}_i + \beta| >= 1$ resulting in the well known model of Support Vector Machine.

This section was about the standard support vector machine version, however, other approaches were invented based on kernel tricks that satisfy the Mercer Kernels theorem [12], such as, Polynomial learning machine and Radial-basis-function network. Other versions includes Bayesian SVM and least-squares support-vector machine for regression problems [13].

IV. Conclusion

In this survey shows the mathematical aspects behind the two famous machine learning methods: neural network and support vector machine. Most of mathematical formulation derived from the support vector machine section was taken from the book Introduction to Nonlinear Optimization, by Amir Beck [14]. We concludes that although artificial intelligence has been improved a lot in robotics it still far from its potential to help society that has been brilliantly shown in sci-fi movies, where robotics are humanoid that helps police, know how to talk, walk, run and so on.

Referências

[1] Wikipedia contributors, "Machine learning," 2019, [Online; accessed 01-December-2019]. [Online]. Available: https://en.wikipedia.org/wiki/Machine_learning 1

[2] Cal-MR, "The robotics institute," 2019, [Online; accessed 02-December-2019]. [Online]. Available: https://www.ri.cmu.edu/ 1

[3] ——, "Center for automation and learning for medical robotics," 2019, [Online; accessed 02-December-2019]. [Online]. Available: http://cal-mr.berkeley.edu/ 1

[4] Will Knight, "Nimble-fingered robot outperforms the best human surgeons," 2019, [Online; accessed 02-December-2019]. [Online]. Available: https://www.technologyreview.com/s/601378/nimble-fingered-robot-outperforms-the-best-human-surgeons/ 1

[5] MIT News Office, "Collaborative learning - for robots," 2019, [Online; accessed 02-December-2019]. [Online]. Available: http://news.mit.edu/2014/collaborative-learning-for-robots-0625 1

[6] Wikipedia contributors, "Neural networks," 2019, [Online; accessed 01-December-2019]. [Online]. Available: https://en.wikipedia.org/wiki/Neural_network 2

[7] Z. Yi and L. Yuan, "Application of fuzzy neural networks in data fusion for mobile robot wall-following," in *2008 7th World Congress on Intelligent Control and Automation*. IEEE, 2008, pp. 6580–6583. 2

[8] H. Li, J. Li, and X. Han, "Robot vision model based on multi-neural network fusion," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*. IEEE, 2019, pp. 2571–2577. 2

[9] M. K. Bugeja, S. G. Fabri, and L. Camilleri, "Dual adaptive dynamic control of mobile robots using neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 129–141, 2008. 2

[10] T. W. Ubbens and D. C. Schuurman, "Vision-based obstacle detection using a support vector machine," in *2009 Canadian Conference on Electrical and Computer Engineering*. IEEE, 2009, pp. 459–462. 3

[11] N. S. Simul, N. M. Ara, and M. S. Islam, "A support vector machine approach for real time vision based human robot interaction," in *2016 19th International Conference on Computer and Information Technology (ICCIT)*. IEEE, 2016, pp. 496–500. 3

[12] Wikipedia contributors, "Mercer's theorem," 2019, [Online; accessed 01-December-2019]. [Online]. Available: https://en.wikipedia.org/wiki/Mercer's_theorem 4

[13] ——, "Support-vector machine," 2019, [Online; accessed 01-December-2019]. [Online]. Available: https://en.wikipedia.org/wiki/Support-vector_machine 4

[14] A. Beck, *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*. Siam, 2014, vol. 19. 4