Machine Learning for Botball

Aleksandar Jevtic
Höhere Technische
Bundeslehranstalt Saalfelden
5760 Saalfelden, Austria
Email: jevtic.aleksandar@gmx.at

Bernhard Vorhofer
Höhere Technische
Bundeslehranstalt Saalfelden
5760 Saalfelden, Austria
Email: bvorhofer@email.com

***Abstract*—Machine learning, in particular using artificial neural networks, is a good way of programming robots to be flexible and adaptive enough to get used to particular settings. Mars missions, the theme of the Botball settings, could be a good possibility to take advantages of machine learning algorithms. We found that it could even be possible to use those algorithms in the Botball competition by loading libraries on to the controller, using e.g. motor feedback as an input to a *programming by demonstration* algorithm or employing *computer vision* algorithms to improve the capabilities of the camera.**

I. Introduction

Artificial intelligence gained more and more attention in the last decades, as modern kinds of artificial intelligence are starting to outperform humans in some tasks. One of the key aspects of developing those intelligent programs is the ability to learn from experience.

Machine learning is a field of computer science, which tries to create those systems for machines or programs that are able to generalize from experience and learn to perform a specific task. There are several approaches to doing this, in this paper we will mention some of the more relevant ones and mainly focus on artificial neural networks.

Lastly we will try to analyze, if these systems are applicable in a Botball robot.

II. Artificial neural network structure

The structures of artificial neural networks are inspired by biological neural networks such as nervous systems and especially brains. [1] The final goal is to use the learning ability of biological neural networks in a digitalized system.

A. The artificial neuron

Neural networks consist of many simple processors, called neurons [3], which have a certain amount of inputs and (often one) outputs. Those neurons are connected to other neurons in a network consisting of different layers.

The neuron takes all the incoming inputs and can be activated by those. In its simplest form, the neuron calculates the sum of the inputs. Its transfer or activation function then determines the output. Depending on the type of network and its application, that function can be linear, a step function or even sigmoid. [4]

Another important factor especially for the training of such networks is that all the inputs for the neuron are differently

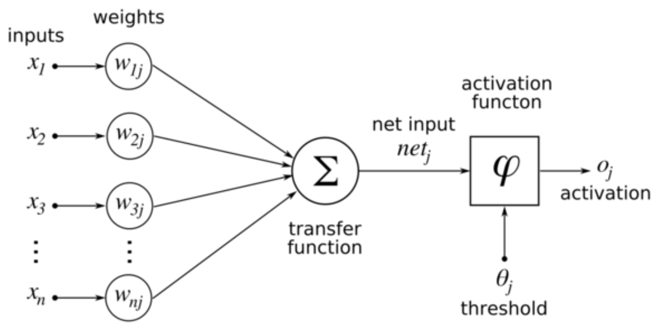

Fig. 1. Artificial neuron structure (https://upload.wikimedia.org/wikipedia/commons/6/60/ArtificialNeuronModel_english.png)

weighted, mathematically seen by a factor, the weight of the input. [4]

B. Basic structure

The previously described neurons are connected to other neurons in a network consisting of different layers. [2]

In the first layer, the so called input layer, the nodes only input is the networks inputs, those can e.g. be coming from a sensor. In the last layer, the output layer, the networks outputs are calculated. In between there are the hidden layers, where most of the complex intelligence lies. Normally the user has no access to those hidden layers. It is possible to place as many hidden layers as you want, with more hidden layers you are able to generalize more complex tasks.

The problem with too many hidden layers and too many nodes is that the more complex the network gets, the more time it requires to be properly trained. [5]

Feedforward ANNs

In Feedforward artificial neural networks the neurons outputs can only be connected to inputs of the next layer. That means that the information can only be transported in one direction. These networks are commonly used in pattern recognition systems. [1]

Recurrent ANNs

Neurons in recurrent ANN systems are able to feed their output back into previous layers of the network. By doing that the network becomes much more flexible, as it now is able to use its internal memory created by those feedbacks to process arbitrary sequences of inputs. Because of this

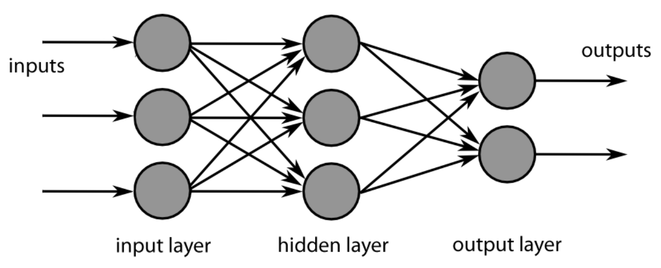

Fig. 2. A feed forward neural network (http://technobium.com/wordpress/wp-content/uploads/2015/04/MultiLayerNeuralNetwork.png)

behaviour these networks are commonly used in voice or handwriting recognition systems. [7]

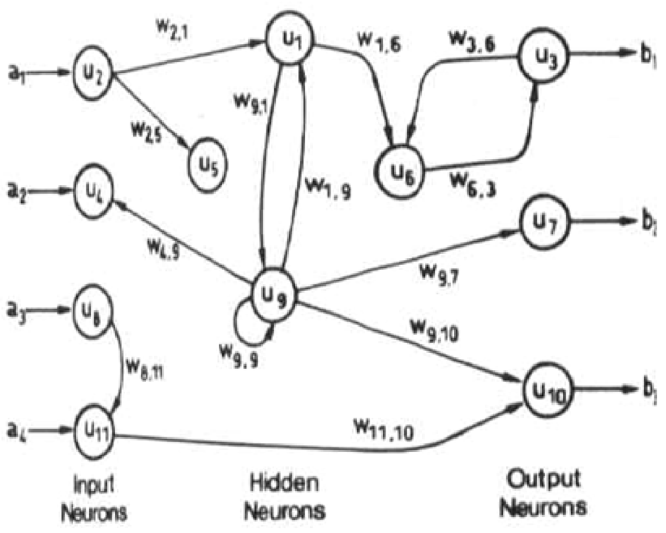

Fig. 3. A recurrent neural network [1]

C. *Training of networks*

Once a network is properly designed for an application, it is ready for training. To train a network, the weights in the structure of the network are initially set to random values and then adjusted by the learning algorithm. [6]

You can diffentiate between these kinds of paradigms for training:

- Supervised learning
- Unsupervised learning

1) Supervised Learning

Learning with supervision is working with training samples i.e. a specific set of inputs to which the desired outputs are known. When those inputs are fed to the system, a training algorithm calculates the so called cost the every node creates. [2]

When the network tries to calculate the outputs, they are compared to the provided desired outputs. Then the cost function calculates the error every node with its current weights is creating, this error function is usually then squared to get the MSE (Mean squared error). Then the goal of the learning algorithm is to minimize the MSE by adjusting the weights. [2]

2) Unsupervised Learning

When an ANN is trained without supervision, it is given unlabeled examples. This means that, in contrast to supervised learning, the network works with sets of inputs without knowing what the desired outputs are. [3] That means that you cannot calculate error or cost functions

Rather than work with error functions unsupervised algorithms are looking for a hidden structure in those sets of data, and tries to cluster them in classes, dependent of their statistical properties.

III. Applications

The possibilities for application of machine learning are very diverse and are often found in areas where they were not expected initially. Machine learning is employed when traditional programming techniques are not capable of performing tasks that require a form of artificial intelligence. We have analyzed several fields machine learning has already been successfully implemented in that are related to robotics, three of which are described below.

A. *Computer vision*

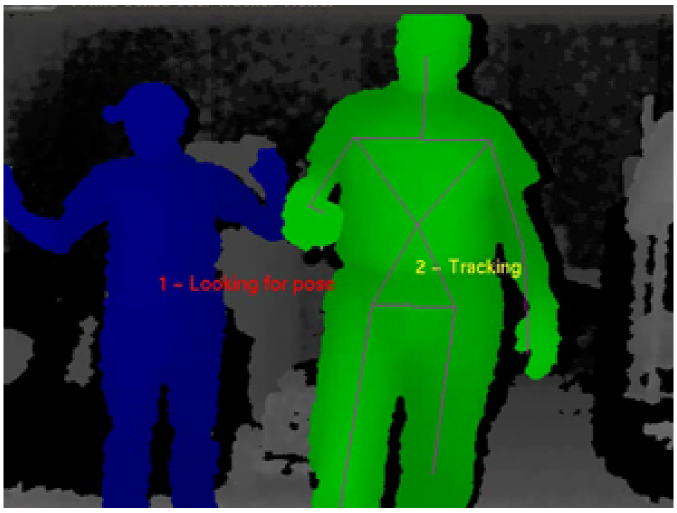

Fig. 4. Microsoft's Kinect uses image and depth data for skeletal tracking and other tasks (http://discover.umn.edu/sites/default/files/umnews/ur_multimedia_305288.jpg)

One of many fields of application for machine learning is computer vision. It includes methods for extracting information from images and processing that information in order to understand certain aspects of the image. In the majority of cases, this technology is employed to make decisions based on data gathered from images. A computer vision algorithm takes one or several images in the form of two-dimensional arrays of numbers (one for each color channel; usually red, green and blue) as an input and produces some form of output. This principle does not only apply to color values, other imaging technologies can be employed as well, such as data from depth sensors or thermal imaging. The output of a computer vision algorithm might be a number, for example the number of cars in the image, a boolean value, for example wether a production part is within specification or not, or some other data format that can be processed further.

Many computer vision systems rely on machine learning to accomplish the otherwise very difficult task of *understanding* an image. In contrast to traditional programming, a machine learning algorithm must be trained using datasets that are often specifically labeled for training purposes. When looking at applications like computer vision where the amount of training data needed can be enormous, it becomes clear that machine learning is closely related to the concepts of *big data* and *data mining*.

Machine learning for computer vision can be unsupervised or supervised. The problem with supervised learning for this application is the of images that needs to be labeled by humans. This can be avoided by using unsupervised learning, but these methods might not yield the same quality of results as supervised methods. There are also rather new concepts like ”semi-supervised learning” and ”self-supervised learning” that aim to solve this problem. [8]

We believe that computer vision is definitely one of the fields of application for machine learning that are most relevant to robotics. The ability for a robot to *see* raises a seemingly endless amount of possibilities that increase the device’s autonomy and awareness of its environment.

Artificial neural networks could be used in the Botball controller in order to improve its vision capabilities. This would probably require a third-party library to be installed on the system, one example for such a piece of software that we found during our research is *FANN* (Fast Artificial Neural Network Library), which can be used in many different programming languages including (but not limited to) C, C++, Python and Javascript; while C and C++ would presumably be the most relevant for the Wallaby controller. The library is easy to use, cross-platform compatible and open-source. It supports various learning- and training algorithms and is well documented, more information can be found in [12]. Unfortunately, we were not able to produce any significant results in object recognition using FANN yet, but we found a paper on spam image recognition using FANN [13] that looks like a good starting point for anyone interested in using FANN for image recognition or similar tasks.

B. Speech recognition

Over the last few years, there have been significant advances in speech recognition technology. Not only were the systems themselves improved, but were also more widely implemented in consumer products like smartphones or personal computers. There are countless possible applications for speech recognition, incluidng control of devices, transcription of recorded speech and real-time transcription (dictation)[9].

The vast majority of speech recognition systems utilize *hidden Markov models*. These are statistical models that are also employed in other forms of pattern recognition such as gesture recognition. [10] By introducing the concept of machine learning to hidden markov models, the model parameters can be altered according to the learning algorithm so the results can be greatly improved by training.

For the Botball competition, this application of machine learning is (currently) not very relevant, but it is certainly a very interesting field of study.

C. Programming by demonstration

Fig. 5. An industrial robot is *trained* by an operator (https://www.youtube.com/watch?v=lXSho9loGYU)

Programming by Demonstration (PBD) allows users to alter the behavior of an application by providing it with examples of what they expect it to do.

The major advantage over traditional programming is the fact that little to no special programming skills are required in order to instruct the software. The user gets to use the interface they are familiar with already, which makes the whole process simpler and more efficient. [11]

A good example for a programming by demonstration that can be found in many applications already is described in [11] on page two. The authors state that macro-recording implemented in many computer programs is a ”powerful but degenerate form of programming by demonstration” because in most cases only the raw keystrokes are recorded and can be played back later. Recorded macros might be suitable for specific scenarios, but fail to adjust to different situations.

A true PBD-based approach would not only record keystrokes but generate a program that resembles the user’s intentions as well as possible.

PBD allows robots to perform complex tasks that might not pose a problem to a human, but are very hard to program in a traditional manner. It has the potential to make programming more intuitive and efficient, which might be an interesting aspect to add to the Botball experience. Because of the versatility of the previously mentioned FANN library [12], it could also be utilized for the purpose of programming by demonstration.

IV. Conclusion

After our research into the topic of machine learning, we consider it a very interesting field we will most likely see astounding advances in the near future. We conclude that it could be possible to utilize machine learning methods in the Botball competition using libraries that are already available online, many of them being open-source software. The implementation of such features would certainly require a large amount of effort but in our opinion, the positive aspects outweigh the negative.

References

[1] Christos Stergiou, Dimitros Siganos: *Neural Networks*. https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html (accessed on 2016/03/18)

[2] Unknown Author: *Using neural nets to recognize handwritten digits*. http://neuralnetworksanddeeplearning.com/chap1.html (accessed on 2016/03/18)

[3] Jürgen Schmidhuber: *Deep Learning in Neural Networks: An Overview*. 2014.

[4] Martin Anthony: *Discrete Mathematics of Neural Networks: Selected Topics (Monographs on Discrete Mathematics and Applications)*. Society for Industrial and Applied Mathematics. 1987.

[5] Devendra K. Chaturvedi: *Soft Computing. Techniques and its Applications in Electrical Engineering*. 2008.

[6] Steven W. Smith: *The Scientist and Engineer's Guide to Digital Signal Processing*. 1st Edition. California Technical Pub. 1997.

[7] Alex Graves: *Generating Sequences with Recurrent Neural Networks*. 2014.

[8] Carl Doersch, Abhinav Gupta, Alexei A. Efros: *The Application of Hidden Markov Models in Speech Recognition*. Foundations and Trends in Signal Processing. vol. 1. no. 3. 2007.

[9] Mark Gales, Steve Young: *The Application of Hidden Markov Models in Speech Recognition*. Foundations and Trends in Signal Processing. vol. 1. no. 3. 2007.

[10] Thad Starner, Alex Pentland: *Real-Time Sign Language Recognition from Video Using Hidden Markov Models*. M.I.T Media Laboratory Preceptual computing Section Technical Report. no. 375. 1995.

[11] Tessa Lau, Steven A. Wolfman: *Programming by demonstration*. October 1995.

[12] Steffen Nissen: *Implementation of a Fast Artificial Neural Network Library (FANN)*. October 2003.

[13] Jason R. Bowling, Priscilla Hope, Kathy J. Liszka: *Spam Image Identification Using an Artificial Neural Network*. October 2003

List of Figures