AI from Concrete to Abstract
Demystifying Artificial Intelligence to the General Public

Rubens Lacerda Queiroz¹ · Fábio Ferrentini Sampaio²,³ · Cabral Lima¹,⁴ · Priscila Machado Vieira Lima⁵,⁶

Abstract

Artificial Intelligence has been adopted in a wide range of domains. This shows the imperative need to develop means to endow common people with a minimum understanding of what AI means. Combining visual programming and WiSARD weightless artificial neural networks, this article presents a new approach to enable general people (including children) to achieve this goal. The main strategy adopted by is to promote a demystification of Artificial Intelligence via practical activities related to the development of “learning machines”, as well as through the observation of their learning process. Thus, it is possible to provide subjects with skills that contributes to making them insightful actors in debates and decisions involving the adoption of Artificial Intelligence mechanisms. Currently, existing approaches to the teaching of basic AI concepts through programming treat “machine intelligence” as an external element/module. After being trained, that external module is coupled to the main application being developed by the learners. In the approach herein presented, the both training and classification tasks are blocks that compose the main program, just as the other programming constructs. As a beneficial side effect of this approach, the difference between a program capable of learning from data and a conventional computer program becomes more evident to the student. In addition, the simplicity of the WiSARD weightless neural network model enables easy visualization and understanding of training and classification tasks internal realization.

Keywords Artificial Intelligence · Teaching AI · Demystifying AI · AI and Block-Based Programming · WiSARD

1. Introduction

In a recent article published by the Brazilian Academy of Sciences, Almeida (2018) states that:

The preparation of a strategy for the advancement of artificial intelligence should start with some choices, such as: Which areas of application of Artificial Intelligence can generate the most economic growth and employment? How can artificial intelligence be applied to improve the quality of life of the Brazilian population? How to minimize the possible adverse effects of new technologies?

Yogeshwar (2018) states that “we need a culture where progress is the result of a reflection process of society, not the exclusive result of engineering and investors”. He added, considering technological advances and their impacts on society: “most politicians do not understand what is happening, they are literally ignorant”.

The perception of each person as an agent responsible for the future of technological development was also observed by Medina (2004): “studies of the past […] reveal that human agency, not technological determinism, has governed the path of history and laid the groundwork for our current challenges.”

An important question is: How would people, in general, and politicians, in particular, make choices, take decisions, set the course of our history “for the better” without understanding a minimum of the matter concerning their decisions, such as Artificial Intelligence? Decisions that concern not only to economic perspectives but also to the moral and ethical aspects of this technology.

As an example of the decision-making power of the general public concerning the future of AI, we can mention the recent public consultation launched by the Brazilian Government for the development of a “Brazilian Artificial Intelligence Strategy”. The purpose of this consultation is “to submit to any citizen's contributions a set of questions that will direct a policy that enhances the benefits of AI in Brazil and the solution of concrete problems,” (Caputo 2019). This Brazilian government initiative follows the
example of AI consultations carried out by other countries and international organizations. In 2017, the interest with the future of Artificial Intelligence led the European Parliament to hold a public consultation specifically on the future of robotics and Artificial Intelligence. “[...] The public consultation included two separate questionnaires, adapted to their audience: one for the general public [...] and one for specialists.” (EUROPARL 2017).

One of the difficulties to aid people in getting initial skills in Artificial Intelligence is the complexity of techniques used to develop AI Systems (Sakulkueakulsuk et al. 2018). As Richard Feynman said: “What I cannot create, I do not understand.” (Caltech 1988).

Visual programming has been used as a great tool for programming learners. The mixing of visual programming and educational robotics has also presented positive results (Queiroz et al. 2019; De Luca et al. 2018; Chaudhary et al. 2016). An interesting intersection between these two approaches is the construction of knowledge from concrete references. This kind of construction process increases the audience of the desired learning due to the reduction of abstract reasoning (see section 3.2). In the context of Artificial Intelligence, it is also possible to build some basic understanding based on establishing clear relationships of some aspects of this field with the tangible (see sections 3 and 5).

People today are exposed to a sort of AI environment and experience their potentials. However, the processes that allow this experience are not easily observable. Is the computer making inferences or the knowledge it is showing was explicitly "told" to it? How does it learn? In the current proposals to work on understanding Artificial Intelligence with general people, which include block programming tasks, the construction of the “Intelligent System” is divided into two distinct processes. First, the learner uses an AI platform, such as IBM Watson, to train a machine learning model. Second, the student builds a block program that uses the trained model to classify new data. The result of this classification can then be used in the program for decision making. In this way, the “intelligent part of the system” is handled as a pre-existing “entity”. As a result, the machine’s intelligence, and the construction of its learning process, are held in the world of abstractions. However, the higher-level tasks required for a machine to be able to learn from data can be included as commands into the program being developed. Thus, people can easily create a system that, step by step:

- Gives the machine the ability to learn
- Request some data for it to learn
- Solicit a label that tells it the meaning of this data
- Asks for data to classify

Including these AI tasks as block program commands, we take machine intelligence from the world of abstractions and bring it closer to the universe of the manipulable. In addition, we also make easily visible the essential difference between a program that uses machine intelligence from one that does not (in a connectionist approach3): the ability to learn from the data.

The analyses of the machine’s learning ability can also be carried out through observable aspects of the developed systems. Turing (1950) brings this idea through the “Imitation Game” proposed in his seminal paper “Computing Machinery and Intelligence” (see section 3.4). The agent-based approach (Russel and Norvig 2010) and the learning process defined by Mitchel (1997) are also tools that contribute to understanding machine intelligence from concrete towards the abstract (see section 3.3). Besides, using WiSARD weightless artificial neural network (WANN) (Aleksander et al. 1984), it is possible to replicate, through unplugged activities4, the processes of training (learning) and classification (application of what was learned) internally performed by the machine (see section 4.2).

The possibility of building a basic understanding of Artificial Intelligence from concrete references presents an opportunity to bring some knowledge about this field to a wider variety of people, from distinct ages and backgrounds. In this sense, this research presents an approach for the demystification of AI and the awakening of a conscious debate around this field, based on the following foundations:

- Knowledge construction from concrete towards the abstract.
- The inclusion of data acquisition, training, and classification tasks, (fundamental for Artificial Intelligence in a connectionist approach), as elements of a “traditional” computer program. That is, as part of a process that takes inputs, processes those inputs, and produces outputs that can be presented to the user or used to achieve a desired goal (Wing 2011).
- The appropriation of the concepts proposed by Alan Turing through the Imitation Game as a tool for the perception of the presence of Artificial Intelligence from observable aspects of AI.
- The adoption of a machine learning model that makes it possible to easily unravel the “magic” behind the observed learning process.

In these foundations, the student can realize (from the concrete to the abstract) that: (i) the machine learns the way

---

1. “When a programming language’s (semantically significant) syntax includes visual expressions, the programming language is a visual programming language.” (Burnett 2002, pp. 77).
2. https://www.ibm.com/watson
3. Connectionist AI is based on neurons simulation and it learns from examples presented to it (Diederich 2010).
4. Activities based on the Computer Science Unplugged technique, used to teach computer science concepts without a computer (Bell et al. 2009).
that humans program it to learn; (ii) learns with what we want it to learn; (iii) does what we allow it to do, through the programs we create. That is: humans are responsible for all that Artificial Intelligence can do for or against us.

2. Related work

In a bibliographic review conducted in 2017 by the authors of this article (Queiroz et al. 2017), we found only one paper dedicated to the study of how teaching AI in basic education. In that only work, storytelling, computer science unplugged, and educational robotics are used to develop activities to teach Artificial Intelligence from kindergarten to high school. Some AI topics covered through these activities are: graphs, data structures, sorting algorithms, problem-solving by search, intelligent agents, automata classic planning, and machine learning (Kandlhofer et al. 2016). The other pieces of work are dedicated to strategies on teaching AI at the university level, and to make some machine learning engines accessible to professionals of other areas, such as artists. In these works, the use of educational robotics and the agent-based approach where the most common strategies adopted.

Kandlhofer et al. (2016) identified in the scientific literature that teaching basic concepts and techniques of AI at the school level was quite rare. This statement still seems to be valid once the amount of scientific research in this field remains not much expressive. A new search conducted by the authors of this paper in 2019 brought few other studies and approaches. This search covered scientific databases and Google. The results are presented in the next section.

2.1. Scientific research

The bibliographic survey about teaching AI to the General Public conducted by the authors of this paper on “Periódicos Capes”5 and Google Scholar brought little scientific research on this theme. We present below those closely related to the purposes of this research.

Hitron et al. (2018) carried out a research with children between 10-12 years old to observe if the subjects could identify two basic machine learning concepts: data labeling and data evaluation. The experiment was carried out using a tennis-like movement recognition device. The observations pointed out that the subjects could understand the desired concepts and extrapolate this understanding to other kinds of applications using machine learning techniques.

In a second work carried out by the same research group, they presented a proposal of opening machine learning black boxes to children aged 10-13 (Hitron et al. 2019). The study was focused on classification tasks, considered by the authors as being less complex and more common in real-world applications than other kinds of machine learning problems. In this proposal, the process of learning gestures was presented and performed in 2 steps. In the first step, the learners feed the model with positive and negative examples of the class of gestures been learned. In the second step, the students evaluate the accuracy of the trained model toward the recognition of new examples. The interface, specially designed for the research, is composed of one button that changes the views between the training phase and the recognition phase. The user can toggle from one view to the other to retrain his/her model and observe the new results obtained in the recognition phase. The feature extraction and model selection phases were not included in the process to be considered too much complex.

Sakulkueakulsuk et al. (2018), present an approach to introduce AI to middle school students in Thailand, a country with 49 percent of labor employed in the agriculture sector. The approach integrates Machine Learning, gamification, and social context in STEM Education. Using a platform called Rapidminer6, the students had to train different models with features such as texture and color of mango, and observe which of the models would better predict, for example, the flavor of the mango based on its external features.

Drugu (2018) explores how children from 7 to 14 years old develop a better understanding of AI concepts and change their perception of smart systems using Cognimates.me7. The platform enables people, through the internet, to train machine-learning models to learn images, sounds, words, sentiments, among others, and use these models into programs developed in Scratch8.

2.2. “World Wild Web” solutions

Artificial Intelligence has shown fast advances in many areas. One of the factors that contribute to this growth in the adoption of AI solutions is the recent advance of the technologies that use Deep Learning (a connectionist AI approach). With this, an increasing number of sites and courses that aim to teach or explain AI to children or laypeople have arisen. As examples, we can mention AI in

---

5 “Periódicos Capes” is A Brazilian search engine that searches for papers in international scientific bases
6 Rapidminer is UI-based software that allows the user to build machine learning models, such as Decision Tree, Neural Network, and k-Nearest Neighbor, via graphical user interface
7 http://cognimates.me/home/
8 https://scratch.mit.edu/
It is relevant to mention that the fact that these solutions were found on our search on Google (and not on scientific databases) does not mean they have not been developed from scientific studies.

NVIDIA\textsuperscript{12} designed AI in Schools\textsuperscript{9} for helping teachers to demystify Artificial Intelligence to their students. The program adopts a Deep Learning approach with a set of lessons. Among other activities, the learner can use a free access web platform to train a machine learning model to classify images.

Teens in AI\textsuperscript{10} is a program that aims to expose young people, aged 12-18, to AI technologies developed and deployed for social good. Some of the topics covered by the project are AI, machine learning, and data science. The program combines a set of activities such as hackathons, boot camps, and accelerators with expert mentoring.

Make Learning for Kids\textsuperscript{11} presents a web platform to work AI concepts with children. Using IBM Watson, children can train machine learning models with text, image, numbers, and sounds. The learners can then use the previously trained models for building smart systems using Scratch.

In addition to the examples presented here, we can find many other sites, magazine articles, YouTube videos, and blogs dedicated to presenting Artificial Intelligence to the general public. This fact points to AI as a theme of great interest to society, which highlights the importance of conducting scientific research on this subject. It is also possible to observe the relevance of studies on demystifying AI to the general public in the recent initiative of AAAI and Computer Science Teachers Association (CSTA) in forming a group to develop national guidelines for teaching K-12 students about Artificial Intelligence (AAAI 2018).

A quite common strategy in the approaches presented here is to perform a machine learning model training and, after that, to evaluate the quality of the performed classifications. Two of the approaches include the building of block programs using the previously trained models. These approaches require an internet connection and do not include the model training as part of the programming environment. None of them presents a study about the impact of the learner's cognitive maturity on the desired learning.

3. Concepts involved

The approach presented in this article was inspired in a Kit for Teaching Computer Programming to children called DuinoBlocks4Kids (DB4K) (see section 4.1). As suggested by Design Science Research (Hevner 2007, Pimentel et al. 2019; Wieringa 2014) the development of DB4K was guided by some Theoretical Conjectures, and the created artifact was used to validate these conjectures. AI from concrete to abstract was developed over the same fundamental theoretical conjecture validated by DB4K: the construction of knowledge based on observable aspects of the processes to be understood is an effective approach to learn and exercise some fundamental computer science abstract concepts. We describe in this section the theoretical bases used to build and support this fundamental conjecture concerning the learning of some fundamentals of AI.

3.1. Constructivism, Constructionism and knowledge building

Piaget's Constructivist Theory (2003) shows that the increase of knowledge is built from the interaction of the subject with the physical environment. Papert's Constructionist Theory (1993a) added to the Piagetian Theory the idea that the construction of knowledge takes place more effectively when the learner consciously engages in the construction of something tangible. Papert saw in the computer a tool that could expand the possibilities of children's creation and, consequently, learning. The reason is that the computer allows people to develop projects with a higher degree of complexity than those they would be able to build using only the "physical world".

Within this context, Papert (1993a) created Logo, a software that allows users, through lines of code, to move a “turtle". The turtle is a cybernetic animal that can be either a virtual object (presented on a computer screen) or a manipulable physical object. This turtle leaves a trail (a drawn line) while it walks, allowing the user to have immediate feedback on the commands given by him/her to the computer. The knowledge construction is established by the user's reflection on the results of the commands he/she gave to the computer from the observation of the graphic elements the turtle draws.

These constructivist and Constructionist principles can be adopted to work in the development of a baseline understanding of AI with the general public. Using Block Programming people can build “learning machines” and interact with the machines they have built. As a result, the understanding of the machine learning process can be built from the observation of the behaviors presented by the developed systems (see section 5). Besides, with WiSARD WANN, learners can also visually observe and replicate, through unplugged activities, the internal process performed by the machine to learn (see section 4.2 and 5.4). Each of these activities can then be followed by discussions about AI fundamentals, reflections of this field.

\textsuperscript{9} http://aimsschools.com/
\textsuperscript{10} http://teensinai.com
\textsuperscript{11} https://machinelearningforkids.co.uk/
\textsuperscript{12} https://www.nvidia.com/
in society, the responsibility of each citizen about the future of Artificial Intelligence, among others.

### 3.2. Cognitive maturity and the power of abstraction

Jean Piaget distinguishes four general stages in cognitive development, namely: sensorimotor (0-2 years old), preoperational (2-7/8 years old), concrete operational (7/8 -11/12 years old), and formal operational (11-12 years old to adulthood) (Moreira 1999).

All individuals go through all these stages or periods in this sequence, but the beginning and end of each depend on the individual’s biological characteristics and educational, social factors. Therefore, the division into these age groups is a reference, not a rigid norm. (Furtado et al. 1999, pp. 102).

The concrete operational stage starts around seven/eight years old. It is characterized as a transition phase between action and more general logical structures, such as classification, ranking, ordering, and grouping (Souza and Wechsler 2014). The subject becomes able to reconstruct on the plane of representation what he/she had already built on the plane of action” (Souza and Wechsler 2014, pp. 144). In other words, subjects in this period begin to perform operations mentally and not only through physical actions as happened in the preoperational stage (Moyses 2002).

In this stage, the subject always resorts to concrete objects present or already known to perform the operations (hence the concrete operational designation). The achievement of concrete operations towards the absent is quite limited. That is, in the concrete operational stage, knowledge is built not from a definition, but a situation, from what is perceptible. The subject needs to compare what is being learned with what is already known or is being physically perceived (Furtado et al. 1999; Moreira 1999; Souza and Wechsler 2014; Pedrozo 2014).

The dominant kind of abstraction in this stage is, therefore, the Empirical (or simple). It consists in the construction of reasoning from the abstraction of objects belonging to the subject’s universe. Empirical abstraction holds on physical objects or material aspects of the action itself (Piaget et al. 1980). For subjects in the concrete operational stage, the use of empirical abstraction is, in general, a routine task. Hypothetical-deductive thinking, for which it would be necessary to construct abstractions from hypotheses (reflexive abstraction), tends not to appear during this period (Lister 2011).

As mentioned before, Piaget’s periods of cognitive development have age groups in which they most commonly occur. However, biological, educational, and social characteristics may cause subjects of the same age to achieve distinct cognitive maturities (Furtado et al. 1999). Besides, the same person may concurrently have characteristic cognitive traits of different developmental periods. For this reason, even adult individuals may not have the capacity for abstraction fully developed (Kramer 2007). Tests conducted on adult populations indicate that only about 30% of adults achieve formal operational skills, which includes the ability to build abstractions from hypotheses. Most adults remain in a transitory stage between concrete and formal operations (Kuhn 1977), and need concrete references for building their understandings.

Besides, according to the neo-Piagetian Theory, regardless of their age, people’s power of abstraction on a specific domain increases along with that person’s experience concerning that domain. “Thus, a person who is a novice in one domain (e.g. chess) will exhibit less abstract forms of reasoning than that same person will exhibit in a domain where he is expert (e.g. calculus)” (Lister 2011, pp. 10).

[…] when facing the need to cope meaningfully with concepts that are too abstract for them, CS13 students tend to reduce the level of abstraction in order to make these abstract concepts meaningful and mentally accessible […] by dealing with specific examples instead of with a whole set defined in general terms. (Hazzan 2008, pp. 40).

Thus, the adoption of approaches that require lesser power of abstraction for the subject to understand AI concepts becomes useful for people of different ages, not just for children. That is, build the intended understandings based on observable aspects of the studied processes ends up making this learning accessible to more people.

### 3.3. Intelligent agents and the learning process

Russell and Norvig (2010, pp. 35) present a pretty simple definition of an agent: “an agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators”. This concept enables a clear establishment of a relationship with elements of everyday life, such as the human being interacting with the environment through his/her body and mind. A robot may have cameras and ultrasonic devices as sensors and motors as actuators. A computer program uses the keyboard and files as sensors, and the screen, speaker and printer as actuators.

The agent concept is used as a tool to analyze systems. This analysis can be performed by observing the actions performed by the agent actuators as a response to the inputs received by its sensors. This approach can be used to

---

13 Computer Science
reduce the power of abstraction needed to differentiate an intelligent system from a conventional computer program because it is based on observable elements.

An intelligent agent learns from what it perceives from the environment. An interesting tool to be used to note the intelligent agent’s learning process is the concept of learning brought by Mitchel (1997, pp. 2):

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

This definition can be applied to a variety of learning tasks, for example, the handwriting recognition learning problem, in which we can identify:

- Task T: recognizing and classifying handwritten words within images
- Performance measure P: percent of words correctly classified
- Training experience E: a database of handwritten words with given classifications (Mitchel 1997, pp. 3).

By applying this concept, the learning process is perceived by the observation of the agent’s behavior. The difference between an intelligent system and a conventional computer program can be analyzed based on observable aspects of intelligence. This approach is in line with the ideas brought by Turing (1950) in his article “Computer Machinery and Intelligence”, covered in the next section.

3.4. Perception of intelligence

Discussing the answer to the question “Can machines think?”, Turing (1950) presents a game called “The Imitation Game”. In short, the original game works as follows: three people, one man (A), one woman (B) and one interrogator (C) participate in the game. Separate from the couple, now baptized as X and Y, the interrogator (C) can ask them any question. X and Y must answer the questions through typed papers to prevent the tone of the voice or the form of the writing from helping to identify who is who. The Interrogator's goal is to find out if X is the man (A) and Y is the woman (B) or vice versa. During this process, A must try to induce C to lose the game by pretending to be B, and B must help C by attempting to show that she is B.

Turing proposes a modification in the game. In the new version, a machine would take the place of A and try to deceive the interrogator by pretending to be human. Meanwhile, B (now male or female) would try to help the interrogator on distinguishing who is the machine and who is the man “behind the wall”.

What will happen when a machine takes the part of A in this game? Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? These questions replace our original, ‘Can machines think?’ (Turing 1950, pp. 424).

Looking for answers to these questions, Turing promotes an in-depth discussion on themes surrounding the possibility of the existence of intelligent machines, including aspects of different areas such as philosophy, mathematics, biology, religion, and psychology. But somehow, in today's universe of Artificial Intelligence, the ideas and concepts brought by Turing were reduced to a test named “The Turing Test”. Based on the Imitation Game, the test aims to determine if a machine is intelligent or not. However, the value of the Turing Test as a benchmark for Artificial Intelligence, as the goal of AI, is widely criticized by AI researchers and experts. According to Stuart Russell, almost nobody in AI is working on passing the Turing Test, except maybe as a hobby, and people working on passing the test would not be described as mainstream AI researchers (Prado 2015). Marvin Minsky called the Loebner Prize, the world’s oldest Turing Test competition, “obnoxious and stupid” (Dormehl 2017). For Marcus (2014), in terms of practical significance for Artificial Intelligence, passing the Turing Test means little.

The value of the Imitation Game is not in its use as a test to verify if a machine can think like a human being but in understanding Artificial Intelligence. Through the proposal of the Imitation Game, Turing gives us the idea that what matters in our understanding of whether a machine is “thinking” is if it behaves intellectually as a human would behave. It would not matter what the machine is doing to act that way, neither if the term “thinking” is the most appropriate to the internal processes the machine is performing to produce the observed outputs. What matters is that we have the perception that the machine is thinking. In other words, it is all a matter of intelligence perception from behavior, as emphasized by Stuart Russell:

[The Imitation Game] […] was designed as a thought experiment to explain to people […] that the possibility of intelligent machines did not depend on achieving consciousness […], an argument about the importance of behavior in judging intelligence…(Prado 2015)

This approach supports the idea of developing a baseline understanding of AI from the observation of the differences between the behaviors performed by “intelligent machines” and those presented by conventional computer programs.

4. Technologies involved

In this section we present the two main technologies that support the AI from concrete to abstract approach:
DuinoBlocks4Kids (Queiroz et al. 2019) and WiSARD (Aleksander et al. 1984).

DB4K is a didactic kit for computer programming learning via educational robotics based on the use of free technology, low-cost components, and recyclable materials. The kit consists of a block-based visual programming environment, a set of educational robotics materials, and a collection of activities. Its use is focused on exercising some Computational Thinking skills in elementary school children (7 years old onwards).

WiSARD is a lightweight weightless artificial neural network model. It has a very visual and simple learning and recognition process. The model was originally designed for image recognition but currently finds application in various domains.

4.1. DuinoBlocks4Kids

![DuinoBlocks4Kids interface](https://ginape.nce.ufrj.br/LIVRE/paginas/db4k/db4k.html)

DuinoBlocks4Kids Kit was developed in 2016 as part of Queiroz (2017) Master's Thesis. The Kit was built on constructivist assumptions (see section 3.1), especially the understanding that children aged 7 to 12 are in the concrete operational stage (Moreira 1999). In this stage, as described in section 3.2, the subject resorts to concrete objects present or already experienced to perform mental operations. Another theory that guided the development of the kit is Papert's Constructionism combined Piaget's Constructivist theory to the use of the computer in education (see section 3.1).

The DuinoBlocks4Kids visual block programming environment (Figure 1) is a programming environment for Arduino15 boards developed using Blockly16 and Ardublockly17. Through the DB4K available blocks, it is possible to control most common robotics devices used in educational robotics classes with Arduino. There are also blocks responsible for the program flux control, such as Repetition and Decision structures.

DB4K blocks embody a fewer abstract semantic than that commonly found in visual environments for Arduino boards programming. Each block makes explicit, through textual and iconic languages, the device that it controls and the result to be observed in that device when the block is used in the program. Besides, hardware-related details such as pinouts and voltage level values are suppressed. For example, to light up an LED, block programming environments for Arduino usually use the block: “set digital pin number 'n' to Hight/Low”. DB4K uses the block “Turn on the LED”, and the LED to be turned on is indicated by its color (Figure 2). That is, the device controlled by the block is an LED, and the expected result of using that block to control this device is that it lights up.

![Figure 2: “Turn LED on” block](https://developers.google.com/blockly/)

The parameters used in the blocks were also simplified. For example, in the case of the block "Spin DC Motor", instead of having to enter a numeric value between 0 and 255 to be applied to a given digital pin, the child chooses one of 3 predefined speeds within their universe of understanding (Figure 3).

![Figure 3: “Spin DC Motor” block](https://ardublockly.embeddedblog.com/index.html)

The kit is also composed of a set of robotics materials. One of these materials is a plastic box, named "The Little Magic Box" (Figure 4) which has all devices programmable by DB4k already connected to an Arduino Board. Some PET bottle robots were also developed, such as The Robot Bat and the Robot Fish (Figure 5). These robots are employed in activities with narratives to contextualize the use of the robotics devices previously worked in class. The use of these materials, along with the block programming environment, enables children to perform the debugging process. Through this process, the learners can, on their own, discover, verify, and correct possible errors in the logic of the developed programs. A particularly important task as it enables children to have more autonomy in their learning.

The DB4K kit was used in a study carried out with seven children (five boys and two girls) living in a low-income community in Rio de Janeiro, Brazil. The students had no previous computer programming experience and were enrolled in public schools. Four of them were from the 4th grade and three from the 3rd grade. The workshop was

---

14 These are adapted images. The original DB4K interface has texts in Portuguese. [http://ginape.nce.ufrj.br/LIVRE/paginas/db4k/db4k.html](http://ginape.nce.ufrj.br/LIVRE/paginas/db4k/db4k.html)

15 Arduino is a low-cost, open-source electronic prototyping platform that is simple to use for any student, including children. ([http://www.arduino.cc/](http://www.arduino.cc/))

16 [https://developers.google.com/blockly/](https://developers.google.com/blockly/)

17 [https://ardublockly.embeddedblog.com/index.html](https://ardublockly.embeddedblog.com/index.html)
composed of 14 meetings of 90 minutes each. The results of the workshop pointed out the feasibility of working the following skills of Computational Thinking with children from seven years onwards by using the DB4K kit: Ability to perform abstractions (more specifically empirical abstraction), understanding control flows, debugging and systematic error detection, iterative thinking, use of conditional logic, and structured problem decomposition.

![Image: The Little Magic Box](image)

Figure 4: The Little Magic Box

![Image: PET bottle Robot Bat and Robot Fish](image)

Figure 5: PET bottle Robot Bat and Robot Fish

DB4K presents itself as an appropriate choice for the development of a block-based programming environment to demystify AI to the general public. This is because it was built respecting the power for abstraction and other cognitive abilities of subjects in the concrete operational stage. As mentioned in section 3.2, reducing the level of abstraction needed to understand contents related to domains with which you are not familiar makes this understanding more accessible not only to children but also to a significant portion of the adult audience.

4.2. WiSARD

In this section, we will present concepts and mechanisms of WiSARD that enable the visualization and reproduction of its training and classification processes through unplugged activities. Additional details of WiSARD artificial weightless neural networks, as well as the differentiation between weightless and conventional (weighted) artificial neural networks, can be found in Carneiro (2012) and França et al. (2014). The motivation for using WiSARD as the machine-learning engine for AI demystification to the general public came from research on the use of this model in image recognition performed around the paper “Playing with Robots Using Your Brain” (Queiroz et al. 2018).

WiSARD (Wilkie, Stonham, and Aleksander's Recognition Device) is a machine learning device that has its training and classification process based on writing and reading data at a computer memory called RAM (Random-access Memory). We can understand the computer memory as a set of “little boxes” where each one can be used to store only a 0 or 1 digit. The entries for WiSARD, i.e., what it will learn, must be in binary format as in the examples of black-and-white images that will be presented in this section. As mentioned earlier, WiSARD was originally developed for the image recognition task, the same type of task adopted by the approach presented in this paper. However, different techniques can be used to binarize non-binary inputs. These inputs can then be used to train a WiSARD network. As a result, it is possible to adopt this model to develop solutions in different domains.

The WiSARD is essentially composed of the following elements:

- **Neurons**: RAM memories.
- **Discriminators**: A set of neurons that hold knowledge about a class of data. For example, a letter, an animal species, the meaning of a sound, and so on.
- **Mapping**: element used to indicate in which position from which neuron should occur a writing (training phase) or reading (classification phase).
- **Adders**: Element employed to indicate the degree of similarity of input data with data WiSARD has already learned.

As mentioned before, the input must be in binary format to be used for teaching WiSARD. As a first example, we will use the learning of a letter E pattern. This pattern will be represented by a black and white image of 3 by 5 pixels (Figure 6). A black pixel corresponds to the number 1, and a white pixel corresponds to the number 0. This entry for WiSARD already in binary format is called Retina. So, in our example, we have a Retina of 15 pixels that can be black (equal to 1) or white (equal to 0).

![Image: Letter E in 3x5 pixels format](image)

Figure 6: Letter E in 3x5 pixels format

The first step the WiSARD needs to perform is to divide the input pixels into sets of pixel sequences named Tuples. The WiSARD will use these Tuples to map the

---

18 A pixel is the smallest unit of a digital image.

19 A tuple is a finite ordered sequence of elements.
input image into Neurons formed by RAM memories (Figure 8). The number of pixels that will compose each Tuple must be a divisor of the total number of Retinal pixels. In this example, we will use Tuples of 3 pixels. To simplify the process visualization, we will identify each of the image's pixels by their coordinates, as in a Battleship game (Figure 7). The sequence of pixels that will be part of each Tuple is chosen at random. For the example presented here, the WiSARD will use the following Tuples: (A4, B2, C1), (A1, C4, A5), (C3, A2, B4), (B3, C5, A3) and (C2, B1, B5).

![Figure 7: Retinal pixels coordinates](image)

The number of elements in each Tuple determines the size of the WiSARD Neurons. How does it work? A binary value, named address, identifies each RAM position. Thus, the number of addressable positions in a neuron will be equal to 2 raised to the power of the number of digits that compose these addresses. In this example, each tuple is composed of 3 elements. So, 8 (2^3) positions can be addressed, corresponding to the addresses: 000, 001, 010, 011, 100, 101, 110, 111 (Figure 8).

![Figure 8: A WiSARD Neuron with 8 positions](image)

Each Tuple in the mapping indicates an address on one of the Discriminator's neurons. A Discriminator is formed by the set of neurons that will hold information about a specific class of data to be learned by the WiSARD (which in this example are letters). In this case, as we have 5 Tuples, we will have Discriminators formed by 5 Neurons (Figure 9). With that, we have everything we need to teach a letter to our WiSARD: A Retina, a Mapping, and a Discriminator formed by a set of Neurons.

The Training Phase consists of writing a number 1 in the memory positions of each neuron indicated by the addresses formed by the Tuples. At the beginning of Training, all positions of all Neurons are filled with the value 0. So, based on the five Tuples presented above, to learn the letter E the WiSARD will write a number 1 at the position indicated by the pixels A1-C4-A5 = 101, and so on to the last neuron. At the end of this process, we have a sample of the letter E already learned by WiSARD (Figure 9).

![Figure 9: WiSARD learning a sample of letter E](image)

We can then teach a second letter E sample, which must have the same Retina as the first. The Mapping is also the same. If an addressed position already has a number 1 written in it, it remains that way. After the second training, the Discriminator of the letter E is as shown in Figure 10.

![Figure 10: WiSARD learning a second sample of letter E](image)

Now we can check if our WiSARD can recognize a letter E that it has not learned yet, i.e., generalize from what it already knows. In the Classification (or Recognition) Phase, WiSARD checks what value is written at the accessed position of a neuron. That is, WiSARD performs a reading at the accessed address instead of writing. If the value read is the number 1, the WiSARD registers that that neuron had an output =1. Otherwise, if there is a 0 at the

---

20 Actually, the draw is pseudo-random, because computers are not usually able to generate truly random numbers.
accessed position, the WiSARD registers that the neuron had an output = 0 (Figure 11). Note that if we present one of the two letters E already trained for WiSARD to recognize, all neurons will display output 1. That happens because the neuron positions to be accessed will be the same as in the Training Phase. So, let's test the classification of a letter E not taught to the WiSARD (Figure 11).

To perform the classification, WiSARD uses an Adder. The Adder provides the measure of similarity. It indicates how much the input is similar to the class of data a Discriminator was trained to recognize. Summing all outputs 1 resulting from the classification process presented in Figure 11, we get a result = 4. Four of the five neurons had output = 1, which means that the pattern presented for the WiSARD is quite similar to the set of letters E taught to the WiSARD.

Now the WiSARD can be tested to distinguish a letter E from a T. In the Classification Phase, the WiSARD maps the pattern presented in its input to all existing Discriminators. Figure 14 shows that, at the end of this process, the Adder of the letter E Discriminator presents a higher value than Adder of the letter T Discriminator. This result indicates that the pattern presented to the WiSARD is more like an E than a T.

With this, the WiSARD will inform us that the exemplar presented to it is a letter E. In Figure 15, we have the same classification process now performed for a letter T not yet taught to the WiSARD. This time, the letter T Adder gives a higher result, and the WiSARD reports that
the letter pattern presented to it is probably a T. The examples presented here demonstrate primary operating mechanisms of the WiSARD. The same processes can be used with larger Retinas containing much more complex images.

Figure 15: A letter T Classification

In addition to allowing easy observation and replication of its training and classification processes, WiSARD enables the visualization of the representation of the acquired knowledge. This is possible because WiSARD allows you to extract Mental Images built from the information recorded in its neurons and present them in the form of pictures (França et al. 2014). These mental images show the "conception" that WiSARD has built from each of the "objects" it has learned.

The basic idea behind the process of building a mental image is to perform the reverse process performed in the training phase. The WiSARD accesses the positions of the neurons whose content is not 0 and checks the sequence of pixels of the Tuple that generated this address. Next, the WiSARD writes 1 (black) in the Retina's points corresponding to the digits 1 of the address, and 0 (white) in the Retina's points corresponding to the digits 0.

Figure 16 shows this process performed for the first Neuron of a Discriminator trained with a single image. The position of the Neuron where writing occurred is 101. The Tuple used to map this Address is A4-B2-C1. So, we have A4 = 1, B2 = 0 and C1 = 1. As a result, the WiSARD writes 1 (black) at position A4 of the Retina, 0 (white) at position B2 and 1 (black) at position C1. Figure 17 shows the same process being performed for the second Neuron. Figure 18 shows the sequence of this process for the last 3 Neurons. At the end, we see that the knowledge stored in the Discriminator is the shape of a letter H.

![Figure 16: Retrieving the data recorded in the first Neuron](image1)

![Figure 17: Retrieving the data recorded in the second Neuron](image2)

![Figure 18: Retrieving the data from the last three Neurons](image3)

21 The WiSARD can be trained to write the number of times each neuron position was accessed for writing, instead of just writing 1s on each accessed address. Thus, to construct the Mental Image from Discriminators trained with a set of images, WiSARD may ignore, or give "less weight", to the "less in common pixels" among the images used for Training (França et al. 2014).
We can observe that all the processes presented here are simple, very visual, and can be replicated through unplugged activities. For this reason, they require a low level of abstraction to be understood. This makes WiSARD a suitable model for presenting an example of machine learning process to the general public. As will be described in section 5, the understanding of these processes works as a base for building a series of debates that contribute to the demystification of AI, and to developing a sharper perception of its impact on today’s society. Furthermore, the “lightweight” of WiSARD allows its easy integration into a block-based programming environment embedded in a low-cost computer. In this way, the benefits of this learning can be carried to more people once the system can be used by low-income or physically isolated communities with hard access to internet services.

5. BlockWiSARD methodology

Classic literacy enables people to read and understand new text, instead of learning a text just by heart (Sklar and Parsons 2002). The same applies to AI literacy: It allows people to understand the techniques and concepts behind AI products and services instead of just learning how to use certain technologies or current applications (Kandlhofer et al. 2016, pp. 2).

One of the obstacles to help the general public to develop a basic understanding of Artificial Intelligence is “the complexity of AI and the prior technical background knowledge required in order to understand AI” (Sakulkueakulsuk et al. 2018, pp. 1006). BlockWiSARD was developed to overcome this initial barrier. It is a visual programming environment based on snap blocks that makes use of the WiSARD WANN to enable general people to develop systems with some learning capability. Figure 19 Shows de concepts and technologies involved in the development of BlockWiSARD.

![Figure 19: Concepts and technologies involved in the development of BlockWiSARD.](image)

DuinoBlocks4Kids (see section 4.1) was developed based on Constructivists and Constructionist fundamentals (see section 3.1). Allying these principals with visual programming and educational robotics DB4K decreases the abstraction power needed to learn basic programming concepts (see section 3.2). Using DB4K, the learner can program systems that perceive the environment through sensors and perform actions over this environment using actuators. Using WiSARD (see section 4.2), BlockWiSARD adds to DB4K the possibility of building systems able to learn. The Intelligent Agent approach and the Learning Process Concept (see section 3.3) make it easily observable the difference between a computer system capable of learning from a conventional computer system (in a connectionist approach). Finally, the verification of the existence or not of intelligence in the developed systems is built from the learner’s perception of the behaviors presented by these systems (see section 3.4). The four elements added for the development of BlockWiSARD have characteristics that help in reducing the power of abstraction required for people to learn in the same way as the components of DB4K.

5.1. BlockWiSARD environment

![Figure 20: BlockWiSARD interface overview](image)

The essential idea of BlockWiSARD is simplicity. The programming environment provides a little set of low complexity components with a semantics based on observable aspects of the concepts to be addressed. The environment’s main window (Figure 20) has two essential elements: the toolbox (Figure 21), where the user picks the blocks, and the workspace (Figure 22), where the user places the blocks to create the programs.

The workspace (Figure 22) has a set of controls for zooming the block program in and out, a trash can, and scrollbars. These are original elements provided by Blockly22. Additionally, we have an orange button on the top of the workspace responsible for “running” the Python23 code generated by the block program, and a red button used to stop the running program. The main

---

22 Blockly (https://developers.google.com/blockly) is the library used to develop BlockWiSARD.
23 Python is a programming language https://www.python.org/. BlockWisard converts the block program created by the user into a program in that programming language. This process is transparent to the user.
window also has interface elements to save a block program, to load a saved block program, to save the python code generated by the block program and to open, in python IDLE\textsuperscript{24}, the python program corresponding to the block program present on the workspace. At the right side the environment has an area to display, when wanted, the python code corresponding to the block program. The Toolbox (Figure 21) is organized in 4 groups of blocks: WiSARD, Controls + Conditions, Sensors, Actuators (LEDs, Motors, Displays and Audio), and Robot.

The Control Blocks group (Figure 23) is responsible for the program flux control. The Conditional Repetition Structure and the Decision Structures uses the blocks presented in the Conditions Group (Figure 24) as the Conditioning factors. In turn, the conditioning blocks use the values obtained by the sensors as reference values for verifying the satisfaction of the imposed conditions. These values can be acquired directly through the sensor group blocks or from the “Recognize Shot Image” block present in the WiSARD group.

The Sensors Blocks Group (Figure 25) has a block to acquire distance values, blocks for reading the state (on/off) of a pushbutton and switches, one block to take pictures with the webcam, one block to get data from the keyboard, and two blocks for data acquisition regarding luminosity. The pink blocks display the data read by the light sensor, the distance sensor, and the keyboard. They might be used in combination with the blocks: Write on the computer screen, Write on the LCD, and Speak (Figures 28 and 29).

The Actuators Blocks have four groups dedicated to them: (LEDs (Figure 26), Motors (Figure 27), Display (Figure 28) and Audio (Figure 29). Using these blocks, the user can control the devices most commonly used in educational robotics activities, write on the computer

\textsuperscript{24} IDLE is Python’s Integrated Development and Learning Environment. (https://docs.python.org/3/library/idle.html).
screen, play some sound effects, and use text-to-speech functionality.

The last group is the Robot Blocks group (Figure 30). These blocks are used to control the movements of a pair of wheels attached to a chassis.

The WiSARD group is responsible for giving the systems developed with BlockWiSARD the ability to learn. The next section presents a detailing of this group.
5.2. Including AI training and classification into blocks programming

In a conventional computer program, the programmer “says” to the computer the meaning of all data it is going to use to make decisions. For this reason, we know exactly how it will “act” when fed with certain data. Using AI (in a connectionist approach) is different. The computer builds the meaning of a set of data presented to it (with human help in a supervised learning approach) and then sets, by itself, the meaning of new data based on what it has learned before. As a result, we don’t know (a priori) what the computer will do (within a set of possible predetermined actions) when fed with certain data. This decision will depend on the meaning that the machine is going to determine for the data presented to it. A way to evidence this fundamental difference is to include in the coding process AI basic tasks such as training (learning something) and classification (recognizing something based on what has been learned before).

Another tool adopted in BlockWiSARD to highlight this distinction is the adoption of the Intelligent Agents approach (see section 3.3). The box with the question mark, present in Figure 31, represents the agent’s mechanism of decision. The agent uses this mechanism to say to its actuators what to do in response to the data acquired through its sensors. In DB4K (see section 4.1), this mechanism corresponds to the simple comparison of the input data with a fixed value pre-defined in the block program. With BlockWiSARD, the learner can easily create a program that turns the black box into an “intelligent box” capable of learning and to generalize from what has been learned. In this way, the learner can turn a non-intelligent agent into a pretty basic intelligent agent. An agent that learns and makes its decisions based on its interpretations of the meaning of data presented to it, not on values pre-determined on its code.

Tom Mitchell’s definition of the learning process was also adopted in BlockWiSARD. As described in section 3.3, we can say that an agent is learning when its performance (P) in performing a task (T) improves with experience (E). The observation of these variables helps to make the perception of the agent’s learning more explicit in a simple manner. In BlockWiSARD we have:

- Task T: To recognize pictures drawn in white papers.
- Performance measure P: The number of pictures correctly recognized. The learner can observe that through the actions performed by the agent in response to the classification process.
- Training experience E: a database of pictures with given classifications presented to the agent in a “batch process” and/or a set of pictures with given classifications presented by the user to the agent in real-time (online).

The WiSARD WANN is ideal for implementing activities around this concept because it can learn entirely online quite fast and with few examples. This characteristic helps to make easily observable, step-by-step, the gain in performance P on task T as training experience E increases.

To include the ability to learn in the programs developed with BlockWiSARD, we designed four blocks directly related to the process of training and classifying (Figure 32 a, b, c, and d). Combining this set of blocks with the blocks presented before, the learner can easily create systems that, step by step:

- Gives the machine the ability to learn
- Request some data for it to learn
- Solicit a label that tells it the meaning of this data
- Asks for data to classify

![Figure 31: An agent interacts with the environment through sensors and actuators (Russel and Norvig 2010)](image)

![Figure 32: WiSARD blocks](image)

The Block “Create WiSARD”, creates an instance of a WiSARD WANN. The WiSARD is presented to the user as a “kind of brain”. By using this block in the program, the student is providing the machine with the ability to learn. For maintaining the simplicity of the approach, this brain can learn only one kind of thing: identifying black drawings on white background. The program can have only one brain. So, the environment allows only one “Create WiSARD” block to be placed on the workspace.
Once the machine is equipped with the learning capability, the user can include blocks in the program that enable the device to feed its brain with “things” to be learned. There are two blocks with this functionality. The first indicates a folder in the machine with pictures to be learned. The second gives the machine the ability to learn from pictures taken in real-time with a webcam. In both cases, the program needs to inform the machine what the picture is. For this purpose, these blocks have a parameter to indicate a label (a name) for each image. Once created, the labels are saved and can be picked from a drop-down list. The next block is used to give the machine the ability to recognize an image captured with its webcam: the “Recognize Shot Image” block. Finally, we have a block that shows the mental image of a trained class of images (see sections 4.1 and 5.4).

The environment performs some automatic checks to ensure that, for example, a Learning block is not positioned before a “Create WiSARD” block, and that a “Recognize Shot Image” block is not used before a “Shot Image” block. In this type of situation, improperly used blocks are disabled and display a warning indicating the problem (Figure 33). This is useful to guide the student’s first steps in building his/her learning machine.

![Figure 33: Example of a disabled block](image)

Combining the Blocks Write in the Computer Screen, Read Text from the keyboard, and Take a Shot with the Cam, with the WiSARD and the Control Blocks, it’s possible to build a learning machine from scratch that interacts with the user using only the computer (See section 5.3). By adding the Robot Blocks, together with the blocks of the other sensors and actuators, you can create a basic intelligent robotic agent in a quite simple way.

### 5.3. Building a learning machine

In the approaches presented in section 2, the “Machine Learning black box” is opened only up to the following point: model training with a set of labeled data and learning verification by presenting new data to be classified. BlockWiSARD enables the learner to open the box a little wider and build the learning and classification process rather than just learning to use it.

Figure 34 presents a program created with BlockWiSARD that transforms a computer equipped with a webcam into a pretty basic learning machine.

![Figure 34: A simple learning machine program build with BlockWiSARD](image)
Through this program the computer can learn online and identify (generalizing from what has been learned) two different classes of images. The learner uses the blocks responsible for the training and classification process the same way as he uses the other program blocks to read input data, process this data, and present the results obtained from this processing or use them for decision-making.

When running this program, the computer interacts with the user asking him/her for pictures to learn or recognize. If the user asks the computer to identify any picture before teaching something to it, the computer tells it doesn’t know what that image is. Teaching the computer, for example, only one exemplar of a five petals flower and a five points star (Figure 35) it can generalize from what it has learned and correctly identify the images presented in Figure 36. When presented to the image in Figure 37 the computer confuses the flower with a star. But, if in the same program flow, the user teaches the machine that Figure 37 is a flower, the image will be correctly identified in the next recognition attempt. Through activities like this, the learning process, as described by Tom Mitchel (see section 3.3), can be easily observed and discussed. And the understanding of this process becomes more effective, according to Papert Constructionism (see section 3.1), because the observer was the builder of the observed system.

By connecting an Arduino board to the computer, the learner can develop systems that control actuators and acquire data from sensors connected to the board. Using a wheeled robot equipped with a Raspberry Pi25, a webcam, a speaker and some other sensors and actuators, one can build a smart robotic agent that learns in batch and/or online, and, for example, moves around the environment guided by signs with pictures that indicate the movements it should make (Figure 38).

Figure 38: Wheeled robot programmable with BlockWiSARD

Working with a little bit more sophisticated systems like these, the discussion about the existence or not of intelligence in the behaviors presented by the built and programmed devices can be extended. The learners can work on observing the difference between the actions performed by the machine from the reading and direct use of data from some sensors (distance sensor, light sensor, keyboard, buttons, and switches, for example) of those behaviors resulting from a learning process.

With this background, it is possible to bring the students to two relevant Artificial Intelligence questions related to two of the four categories of AI definitions presented by Russell and Norvig (Figure 39): Are these machines acting rationally? Are they acting like a human? From these initial questions, it is possible to broaden the debate with some other inquiries, such as: Even if you think these machines are not acting rationally or humanely, it seems to you that they are doing things that would require intelligence to be done? What is the difference between this seemingly intelligent behavior from other behaviors presented by these machines? If the machine is doing something that would require a human to use his/her intelligence to perform, do you believe that it is thinking like a human? A set of questions that may help the learners to expand their understanding of AI from practice.

As we are going to see in the next section, WiSARD enables the learners to go further in their AI understanding after this first level of discussion. The work around WiSARD internal processes may suggest questions on the two other categories of AI definitions presented by Russell and Norvig (Figure 39): Are these machines thinking rationally? Are these machines thinking humanely?

25 Raspberry Pi (https://www.raspberrypi.org/) is a popular low-cost single-board computer provided with a set of General Purpose Input/Output (GPIO) pins where one can connect robotics devices.
5.4. Opening the black box a little bit more

Papert once said:
The reason you are not a mathematician might well be that you think that math has nothing to do with the body; you have kept your body out of it because it is supposed to be abstract, or perhaps a teacher scolded you for using your fingers to add numbers! This idea is not just metaphysics. It has inspired me to use the computer as a medium to allow children to put their bodies back into their mathematics (Papert 1993b, pp. 31).

The WiSARD weightless artificial neural network is an appropriate tool to “put people bodies” into AI learning. Through unplugged activities, learners can reproduce the internal training and classifying processes of WiSARD using their body and brain. These activities, along with the use of BlockWiSARD, allow further advance the demystification of AI starting from the concrete towards the abstract (Figure 40).

In other artificial neural network models, the understanding of the way the model is internally working for learning could present itself as being significantly complex. When using WiSARD, this understanding becomes relatively easy because of the simplicity of the model (see section 4.2). For this reason, the WiSARD makes it possible, quite easily and playfully, open the “Machine Learning black box” a little bit more. As seen in section 4.2, the learners can view and reproduce, step by step, the processes of tuples selection, training of different letter images (by writing number ones into the bits of the classifier neurons), and later classification of new image samples (by checking the contents of the neurons).

Another interesting tool to be used in this sense are the mental images produced by WiSARD. They allow students to perform two relevant kinds of activities. First, the learners can extract the knowledge “recorded” in a WiSARD Discriminator, as described in section 4.2. Second, they can put themselves in the place of the machine and see how they would classify a picture using the knowledge that the device has about that image. For example, Figure 41 and Figure 42 show a set of six dog faces and six flowers used to train a WiSARD with BlockWiSARD, and Figure 43 shows the mental images displayed by BlockWiSARD of the previously learned figures. These mental images allow learners to put themselves in the place of the WiSARD and check how they would rank other pictures comparing them directly with the mental images produced by the WiSARD. After that, learners can present the same pictures to the WiSARD and see if it will rate them the same way.

Turing (1950), talking about the argument of consciousness, said:

[...] the only way by which one could be sure that machine thinks is to be the machine and to feel oneself thinking [...] . Likewise, according to this view the only way to know that a man thinks is to be that particular man [...] . [...] [So] it is usual to have the polite convention that everyone thinks (Turing 1950, pp. 446)

WiSARD allows people to look inside the machine “brain” and see what the device is doing to recognize the pictures presented to it and can be perceived from us as if it was thinking. From these activities, it is possible to develop, based on very concrete references, a debate about human intelligence and machine intelligence, based on questions like: Is the machine thinking like a human? Is the machine thinking rationally? Is the machine thinking? If the machine is not thinking, what is it doing? Does it make any difference about the way AI can influence and transform our lives if what the machine is doing to act the way it is acting is to think or not? Does it make any difference if the machine is using or not the same kind of
intelligence that we use to do things that before AI only humans could do? And finally, from the ideas brought by Alan Turing in his 1950 seminal paper: What is imperative for us right now? To discuss the impacts of using AI techniques on our lives today, and our near future, or whether computers will someday be able to think and act exactly like humans?

Figure 41: Dog faces used to train the WiSARD

Figure 42: Flowers used to train the WiSARD

Figure 43: WiSARD mental images of a dog and a flower

6. Conclusion

The choices that will define the future of Artificial Intelligence are in the hands of each citizen. These choices include the ethical principles on which AI is built and the degree of autonomy to be given to intelligent systems, among others. Thus, it is imperative that people, in general, be aware of what Artificial Intelligence is. In this way, initiatives that help the general public to build a basic understanding of this domain are of great importance.

Inserted in this context, this article presented an approach that shows the possibility to demystify AI to the general public by constructing knowledge from concrete references toward abstract concepts. By including basic AI learning and classification tasks as components of a block program, the learner can directly establish the difference between a system that uses Artificial Intelligence from one that does not (in a connectionist approach). With the ability of WiSARD to learn from few examples the learners can more clearly observe machine’s performance gain in the recognition task as experience increases. Furthermore, the WiSARD model allows the opening of the “machine learning black box” beyond what other approaches do. This is possible because WiSARD has a simple learning and classification process that can be easily replicable with unplugged activities. Finally, WiSARD can be trained with no internet connection, which is useful for low-income or geographically isolated communities.

As seen in Section 5, these attributes allow general people, including children, to establish initial contact with the four most adopted Artificial Intelligence approaches: Act Humanly, Act Rationally, Think Humanly, and Think Rationally. The appropriation of these concepts becomes a fertile ground for the debate about Artificial Intelligence, raising questions like: where AI currently stands? Which directions we want AI to take? What aspects of AI are most important to discuss today? What impacts can AI have on our society? By looking for answers to these questions, learners can internalize the main ideas about AI approaches and develop their critical view about the uses of AI in our society. This way, decisions regarding what AI may do for or against us, can be rendered to every citizen.

This process can be carried out with the general public and with a more specialized audience, such as students in computer science and engineering courses. Therefore, we hope that the knowledge presented in this research can contribute to building a future in which the benefits of Artificial Intelligence for society are maximized, and its risks and possible harm can be mitigated as much as possible.

6.1. Future work

The robot’s physical appearance, its manner of movement, and its manner of expression convey personality traits to the person who interacts with it. This fundamentally influences the manner in which people engage the robot (Breazeal 2002, pp. 51).

Thus, to increase learner engagement in some of the learning processes presented here, a study of plastic and behavioral characteristics to be adopted when designing robots used in these processes would be of great value.
Besides, for the results of such a study to reach a wider public, it is relevant to identify aesthetic and behavioral aspects that can be incorporated in robots built from the use of cheap or recyclable materials, and low-cost robotics.

Another relevant aspect to be developed from this work is the inclusion, within the processes described here, of other WiSARD potentials such as text, speech, and facial expressions recognition. That may be an interesting way to show learners that the same machine learning model can be applied in various domains.

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

AAAI (2018) AAAI Launches K-12 Initiative AI Magazine 39:98. https://doi.org/10.1609/aimag.v39i2_2820
Alekksander I, Thomas W, Bowden P (1984) WiSARD - a radical step forward in image recognition Sensor review. https://doi.org/10.1108/eb007637
Almeida V (2018) Brasil precisa de estratégia para inteligência artificial. Academia Brasileira de Ciências. http://www.abc.org.br/2018/08/08/brasil-precisa-deestrategia-parainteligencia-artificial/. Accessed 25 March 2020
Bell T, Alexander J, Freeman I, Grimley M (2009) Computer science unplugged: School students doing real computing without computers. The New Zealand Journal of Applied Computing and Information Technology 13:20-29
Breazeal CL (2004) Designing sociable robots. MIT press, Massachusetts
Burnett MM (2002) Software Engineering For Visual Programming Languages. In: Shi-kuo C (ed) Handbook Of Software Engineering And Knowledge Engineering, vol 2. World Scientific, Singapore, pp 77-92
Caltech (1988) Richard Feynman's blackboard at time of his death. http://archives-de.library.caltech.edu/islandora/object/ct1%3A483. Accessed 25 March 2020
Caputo M (2019) MCTIC lança consulta pública para a Estratégia Brasileira de Inteligência Artificial. Academia Brasileira de Ciências. http://www.abc.org.br/2019/12/21/mctic-lanca-consulta-publica-para-aestrategia-brasileira-de-inteligencia-artificial/. Accessed 25 March 2020
Carneiro HCC (2012) A Função do Índice de Síntese das Linguagens na Classificação Gramatical com Redes Neurais Sem Peso. Universidade Federal do Rio de Janeiro
Chaudhary V, Agrawal V, Sureka P, Sureka (2016) A An experience report on teaching programming and computational thinking to elementary level children using lego robotics education kit. In: IEEE Eighth International Conference on Technology for Education (T4E). IEEE, pp 38-41 https://doi.org/10.1109/T4E.2016.016
De Luca G, Li Z, Mian S, Chen Y (2018) Visual programming language environment for different IoT and robotics platforms in computer science education CAAI Transactions on Intelligence Technology 3:119-130. https://doi.org/10.1049/trit.2018.0016
Diederich J, Gunay C, Hogan JM (2010) Recruitment Learning. Springer Berlin Heidelberg, Heidelberg
Dormehl L (2017) Thinking Machines: The Quest for Artificial Intelligence--and Where It's Taking Us Next. Penguin Publishing Group, New York
Druga S (2018) Growing up with AI: Cognimates: from coding to teaching machines. Massachusetts Institute of Technology
EUROPARL (2017) Summary of the public consultation on the future of robotics and artificial intelligence. European Parliament. https://www.europarl.europa.eu/committee/en/archives/8/juri/public-consultations/robotics. Accessed 25 March 2020
França F, De Gregorio M, Lima P, de Oliveira W (2014) Advances in weightless neural systems. In: Proceedings of the 22nd European Symposium on the Artificial Neural Networks, Computational Intelligence and Machine Learning
Furtado O, Bock AMB, de Lourdes Trassi Teixeira M (1999) Psicologias: uma introdução ao estudo de psicologia. Saraiva, São Paulo
Hazzan O (2008) Reflections on teaching abstraction and other soft ideas ACM SIGCSE Bulletin 40:40-43. https://doi.org/10.1145/1383602.1383631
Hevner AR (2007) A three cycle view of design science research Scandinavian journal of information systems 19:4
Hitron T, Orlev Y, Wald I, Shamir A, Erel H, Zuckerman O (2019) Can Children Understand Machine Learning Concepts? The Effect of Uncovering Black Boxes. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp 1-11. https://doi.org/10.1145/3290605.3300645
Hitron T, Wald I, Erel H, Zuckerman (2018) O Introducing children to machine learning concepts through hands-on experience. In: Proceedings of the 17th ACM Conference on Interaction Design and Children, pp 563-568. https://doi.org/10.1145/3202185.3210776
Kandlhofer M, Steinbauer G, Hirschmugl-Gaisch S, Huber P (2016) Artificial intelligence and computer science in education: From kindergarten to university. In: IEEE Frontiers in Education Conference (FIE). IEEE, pp 1-9. https://doi.org/10.1109/FIE.2016.7757570
Kandlhofer M, Steinbauer G, Hirschmugl-Gaisch S, Huber P (2016) Artificial intelligence and computer science in education: From kindergarten to university. In: IEEE Frontiers in Education Conference (FIE), IEEE, pp 1-9. https://doi.org/10.1109/FIE.2016.7757570
Kramer J (2007) Is abstraction the key to computing? Communications of the ACM 50:36-42. https://doi.org/10.1145/1232743.1232745

Kuhn D, Langer J, Kohlberg L, Haan NS (1977) The development of formal operations in logical and moral judgment Genetic Psychology Monographs 95:97-188

Lister R (2011) Concrete and other neo-piagetian forms of reasoning in the novice programmer. Conferences in Research and Practice in Information Technology Series 114:9-18

Marcus G (2014) What Comes After the Turing Test? The New Yorker. https://www.newyorker.com/tech/annals-of-technology/what-comes-after-the-turing-test. Accessed 25 March 2020

Medina EM (2004) Beyond the ballot box: computer science education and social responsibility SIGCSE Bull 36:7–10. https://doi.org/10.1145/1041624.1041626

Mitchell TM (1997) Machine Learning. McGraw-Hill, New York

Moreira MA (1999) Teorias de aprendizagem. E.P.U., São Paulo

Moyle’s JR (2002) Só brincar? O papel do brincar na educação infantil. Artmed, Porto Alegre

Papert S (1993a) Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, New York

Papert S (1993b) The Children’s Machine: Rethinking School In The Age Of The Computer. Basic Books, New York

Pedrozo MK (2014) As fases do Desenvolvimento Infantil parte 4: Estágio Operatório Concreto. Espaço Mediação. http://psicopedagogiacuritiba.com.br/fases-desenvolvimento-infantil-parte-4-estagio-operatorio-concreto/. Accessed 25 March 2020

Piaget J (2003) The Psychology of Intelligence. Routledge, Oxfordshire

Piaget J, Ackermann E, Entel A, Sinnott E (1980) Investigaciones sobre la abstracción reflexionante: La abstracción del orden y de las relaciones espaciales. vol 2. Huemul, Buenos Aires

Pimentel M, Filippo D, Santoro FM (2019) Design Science Research: fazendo pesquisas científicas rigorosas atreladas ao desenvolvimento de artefatos computacionais projetados para a educação. In: Jaques P, Pimentel M, Siqueira S, Bitencourt I (eds) Metodologia de Pesquisa em Informática na Educação: Concepção de Pesquisa, vol 1. SBC, Porto Alegre. https://metodologia.ceie-br.org/livro-1/. Accessed 25 March 2020

Prado GMD (2015) Researchers say the Turing Test is almost worthless. Business Insider. https://www.businessinsider.com/ai-researchers-aren’t-trying-to-pass-the-turing-test-2015-8. Accessed 25 March 2020

Queiroz RL (2017) DuinoBlocks4Kids: utilizando Tecnologia Livre e materiais de baixo custo para o exercício do Pensamento Computacional no Ensino Fundamental I por meio do aprendizado de programação aliado à Robótica Educacional. Universidade Federal do Rio de Janeiro

Queiroz RL, de Azeredo Coutinho IB, Xexêo GB, Lima PMV, Sampaio FF (2018) Playing with robots using your brain. In: 17th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames), IEEE, pp 197-1977. https://doi.org/10.1109/SBGAMES.2018.00031

Queiroz RL, Sampaio FF, Lima PMV (2017) Uma investigação sobre estratégias a serem adotadas para o aprendizado de Inteligência Artificial no Ensino Fundamental por meio da Robótica Educacional. In: ERSI-RJ, Rio de Janeiro.

Queiroz RL, Sampaio FF, Santos MPd (2019) DuinoBlocks4Kids: Using Free Technology and Low-Cost Materials for the Exercise of Computational Thinking in Primary School via Programming and Educational Robotics. Brazilian Journal of Computers in Education 27:167-197. http://dx.doi.org/10.5753/ribe.2019.27.02.167

Russell SJ, Norvig P (2010) Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River

Sakulkeaaksuk B, Witson S, Narmacjornwiwat P, Pataranapatpon P, Surareungchai W, Pataranapatpon P, Subsoontorn P (2018) Kids making AI: Integrating Machine Learning, Gamification, and Social Context in STEM Education. In: IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), 4-7, pp 1005-1010. https://doi.org/10.1109/TALE.2018.8615249

Sklar E, Parsons S (2002) RoboCopJunior: a vehicle for enhancing technical literacy. In: AAAI Mobile Robot Competition, pp 108-118. https://www.aaai.org/Papers/Workshops/2002/WS-02-18/WS02-18-016.pdf. Accessed 25 March 2020

Souza NM, Wechsler AM (2014) Reflexões sobre a teoria piagetiana: o estágio operatório concreto. Cadernos de Educação: Ensino e Sociedade 1:134-150. https://profamaurybarreto.files.wordpress.com/2014/03/piaget.pdf. Accessed 25 March 2020

Turing AM (1950) Computing machinery and intelligence. In: Epstein R, Roberts G, Beber G (eds) Parsing the Turing Test. Springer, Dordrecht, pp 23-65

Wieringa RJ (2014) Design Science Methodology for Information Systems and Software Engineering. Springer, Heidelberg

Wing JM (2006) Computational thinking Commun ACM 49:35–36. doi:https://doi.org/10.1145/1118178.1118215

Wing J (2011) Research notebook: Computational thinking—What and why? The Link Magazine 06:20-23. https://www.cs.cmu.edu/sites/default/files/11-399_The_Link_Newsletter-3.pdf. Accessed 25 March 2020

Yogeshwar R (2018) Os dados que você entrega acabarão por estrangular-lo no sistema. El País. https://brasil.elpais.com/brasil/2018/10/16/tecnologia/1539702580_437926.html. Accessed 25 March 2020

21