AI Unplugged

Unplugging Artificial Intelligence

Activities and teaching material on artificial intelligence

Annabel Lindner
Stefan Seegerer

Preface

AI (Artificial Intelligence) is becoming a topic of increasing social importance. Political reactions like the publication of the AI strategy of the German Government in late 2018 are one indicator for that. But more importantly, we are already interacting with AI systems as if it were the most natural thing in the world, for example, when using language assistants such as Siri or Alexa. Nevertheless, according to surveys, over 50% of Germans do not know what artificial intelligence is.

To address this issue, we have put together a collection of Unplugged Activities related to the topic of AI. Unplugged Activities provide approaches that help learners of all ages to experience the ideas and concepts of computer science actively and do without the use of a computer.

This brochure contains five activities you can use to teach ideas and concepts of artificial intelligence to learners of all ages.

Nowadays, AI is primarily realized through machine learning, but artificial intelligence is far more than that: AI is not only about technical aspects, but also raises questions of social relevance. This brochure shows possibilities, how these topics can be discussed with children and adults.

If you have any questions, comments or remarks about this material, please do not hesitate to contact us at aiunplugged@digi4all.de.

Some activities require additional material.
Print templates can be found here:

https://aiunplugged.org

Contents

Classification with Decision Trees

The Good-Monkey-Bad-Monkey Game

Target group

Primary School Level, Secondary School Level

That's what it's all about

How does a computer make decisions independently? How does a computer decide whether a person is athletic, should acquire a loan, etc.? Such classification processes are a frequent application of AI. In this activity, students have the opportunity to create their own classification model using a decision tree. In the end, the best one of the students' models is selected for further classification tasks.

These ideas are behind it

- AI classifies data based on patterns.
- AI uses the classification model that best fits the given data.
- Classification models are not perfect.
- Certain combinations of characteristics indicate a certain category.

What you need

- Monkey cards (preparation: cutting the template, alternatively use digital version)
- Blackboard with magnets or pinboard

Here's how it works

Students examine how a series of sample elements (training data) belongs to a category. To do this, they develop criteria that can be used to classify new elements. Subsequently, the resulting models are tested with new examples (test data) and the accuracy of the prediction is determined.

Context

We are animal keepers in a zoo and responsible for feeding the monkeys. The monkeys look very cute, but we have to be careful because some monkeys are biting. We already know whether the monkeys in the zoo bite. However, new monkeys will be joining the group soon and we need to consider how to find out which new monkeys bite and which don't - preferably without getting too close to their teeth.

Activity Description

Depending on the target group, you choose the elementary game version with 20 picture cards (blue) or the advanced version with 40 picture cards (blue and green). These 20 or 40 monkeys are all animals of the zoo, i.e. we already know if they are going to bite. They are split into training and test data. Based on the training data, we think of criteria that determine whether the monkeys bite and check their reliability based on test data. The training data is - subdivided into the two categories biting and not biting - pinned on the board. The test data is not revealed at first. You can think of rules by which to distinguish the monkeys yourself or use one of the proposals below (using reduced subsets is also possible). The rules that apply in the examples are illustrated with decision trees. First, make your students aware of the details they could focus on by illustrating the procedure with an example. For example, compare the monkey cards 01 to 04 and 05 to 08. In this example, the shape of the mouth is an indication for biting monkeys, but not the eyes (Fig. 1). Alternatively, with older students, you can also use the simple version of the game (Version 1) to demonstrate the rules and the necessary procedures.

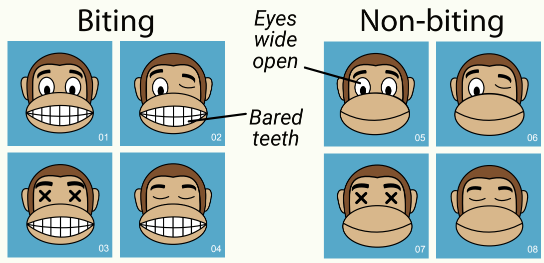
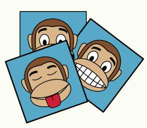


Fig. 1: In this simple example all monkeys with bared teeth are biting.

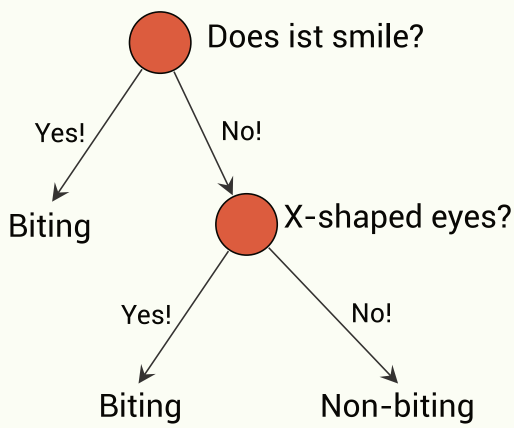
The students form teams of two and use the training data to develop criteria for distinguishing biting from non-biting monkeys. These must be clearly noted so that they can be applied to new examples by another team afterwards. A possibility to record the criteria is a decision tree. It should be the goal that the existence or absence of a particular characteristic permits a clear assignment to one of the groups. The use of decision trees is optional, alternatively, it is also possible to

Version 1 (blue)

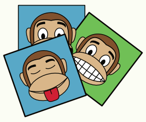


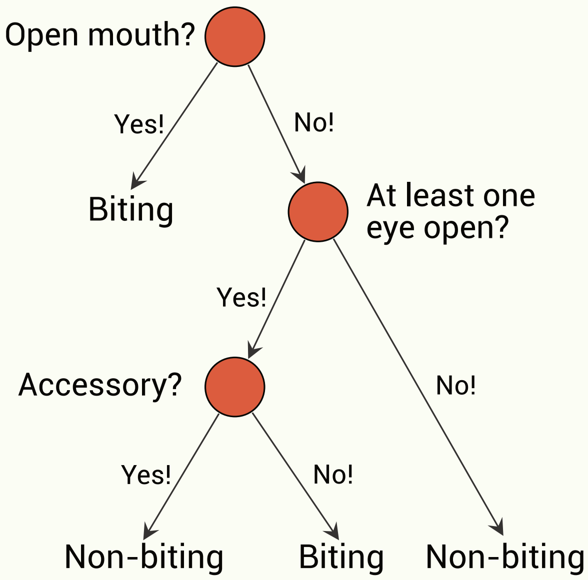
Version 1 training data
biting: 6, 7, 8, 15
non-biting: 1, 2, 4, 9, 12, 14, 17, 18

Version 1 test data
biting: 3, 5, 11, 19
non-biting; 10, 13, 16, 20



Version 2 (blue & green)





Version 2 training data
biting: 1, 2, 5, 9, 10, 14, 15, 16, 17, 28, 33, 35, 36
non-biting: 4, 7, 12, 19, 22, 23, 24, 25, 30, 32, 37, 38, 39, 40

Version 2 test data
biting: 6, 13, 18, 34
non-biting; 3, 8, 11, 20, 21, 26, 27, 29, 31

explicitly write down decision rules. At the end of the training phase, the criteria formulated are exchanged with another team. Now, the students are shown the pictures of the remaining monkeys (test data) one after the other. For each image, the teams decide whether the monkey will bite or not using the scheme of rules developed by their classmates. Each team notes down their decisions. After showing all monkeys, it is evaluated which team has assessed best the biting behaviour of the monkeys. It comes to the students' attention that many classification models categorize most monkeys correctly, but that it is difficult to properly classify all animals. For us as animal keepers it is therefore clever to use the most successful model when feeding the new monkeys, even if it doesn't guarantee that we will never get bitten.

In the advanced version, image no. 21 (see Fig. 2) can be used to illustrate the problems of an AI system when the characteristic value of an element differs significantly from the training data. We have no experience with the characteristics of image no. 21, because this monkey has a new, unknown mouth shape. Accordingly, an appropriate assignment of the monkey is not possible. In practice, the behaviour of an AI system is very difficult to predict for this case. Instead of image no. 21 you can also use the image of a different animal to emphasize the different characteristics of the new element. Subsequently, these examples can be applied to reality: a bank does not grant a credit to a specific customer unexpectedly, the self-propelled car recognizes leaves on the road as a dangerous situation and mistakenly slams on the brakes. In these situations, the AI system can also be dangerous, if it is not comprehensible, how these decisions were made.

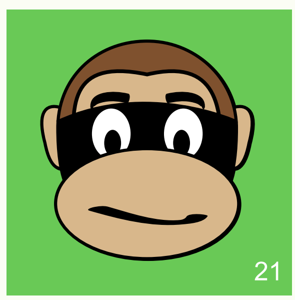


Fig. 2: For monkey 21, no explicit criteria can be derived from the data.

Background

Category formation is made possible by recognizing repetitive patterns in individual elements. But how do these aspects relate to artificial intelligence?

In so-called supervised learning, the AI system observes a series of input and output pairs (training data) and learns how they relate to each other as well as which patterns are typical for which category. This knowledge is then used to classify new elements into the categories. Test data, whose categories we know, but the AI system doesn't, is used to determine the quality of the learned classification model.

The same principle is used for neural networks and other AI applications. This procedure can lead to various problems, because no model is perfect. Depending on the training data, the classification model can overweight or neglect certain characteristics of the training data so that no general statements and, thus, no correct classification of unknown elements is possible. A lot of training data can help to reduce these effects, but does not always lead to more accurate results, since too much training data can also result in overfitting. In this case, the AI system learns the training data "by heart" and is no longer able to generalize to new data.

It makes sense to address these aspects of machine learning as part of the activity. When applying their rules in the test phase, let the students explain which characteristics they used to classify the monkeys. This will illustrate that the students have created different sets of rules. Point out that a classification model is unlikely to be 100% accurate and that the model that best classifies the test data will be chosen in the end. Have students describe their own "learning process" and then compare it to that of a computer.

Fig. 3: Training data split into two categories

#deeplearning

Image recognition with neural networks

Target group

Secondary School Level

That's what it's all about

How can a computer "recognize" things? How does a computer decide whether a photo shows a cat? How can it distinguish buildings from people? Recognizing objects based on their shape or their appearance is very easy for people. For the computer, which, for example, can be used in a self-propelled car to recognize the objects in its environment, this represents a complex task. In this activity, pupils have the opportunity to find out, how computers recognize the content of images.

These ideas are behind it

- Neural networks assign inputs to specific outputs: Raw data, such as images, are classified, for example, by assigning terms to the objects in the image.
- Neural networks consist of different abstraction layers that can identify increasingly complex features.
- The classes of objects to be recognized must already be known to the AI system.

What you need

- Photo cards of houses, cats and cars for each group

Here's how it works

The students recreate the image recognition process of a (simplified) neural network. They take on the roles of the different layers within such a network. They extract features from a photograph and classify the image. They recognize the limits of the system and consider which modifications to the network are necessary to achieve better results with their network.

Context

As humans, we rely heavily on what we see. If we see a cat, we know immediately that it is a cat, if we see a dog, we recognize it at once. A computer, on the other hand, can't detect this as easily, but it can learn it - just like we did in infancy. The computer is shown a lot of pictures of dogs, but also of other animals. With this information, it learns which patterns in an image can be used to distinguish a dog from a cat. Properly trained, the computer cannot only automatically label images, but also detect skin cancer or - built into a car - react to obstacles on the road.

Activity Description

Start by discussing how a computer might recognize the contents of an image. Answers will often refer to defined rules or a comparison with an image database, but nowadays computers do it differently. Divide students into groups of three, each group receives a stack of photo cards. In every group there are three rolls, each representing one layer of a neural network (see Fig. 4). The tasks of the roles are the following:

A picks an image from the stack of photo cards (B and C should not see the image!), creates two different sketches of it (30 seconds each) and passes them on to B. Thereby, it is important that C does not see the sketches.

B receives the sketches from A and checks whether square shapes, triangular shapes or round shapes are included. Then B passes the collected information on to C.

C evaluates the information received using the following table and announces whether the original picture is a house, a car or a cat.

	Rectangular shape?	Triangular shape?	Round shape?
House	Yes	Yes	No
Car	Yes	No	Yes
Cat	No	Yes	Yes

Finally, A determines whether the solution is correct.

Let the students try the game in different roles. It is also possible to have a role played by two students or to explicitly assign each role to three students, who each represent a neuron (i.e. a knot in the layer) and no longer a whole layer.

After a short trial, hand out some images to the students that either do not fit into the categories that the net can recognize or have features that do not allow a clear classification. For example, a picture of a dog is not correctly recognized by the net - simply because the net does not know the category "dog". Based on this finding, students are now considering how the net can be changed and expanded to

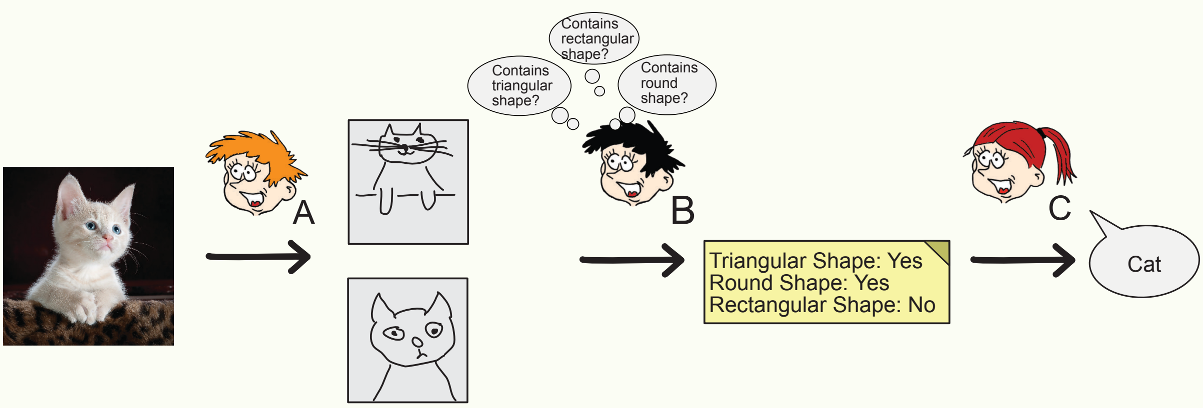


Abb. 4: The students' roles

recognize dogs or other objects in the future. First of all, a new output category must be introduced. At the same time, however, the number of characteristics identified by the net is no longer sufficient. Consequently, either further characteristics must be added that allow the categories to be distinguished, or several characteristics must be combined to form a more complex pattern. This merging ultimately involves the addition of further layers in the neural network. Fig. 6, for example, can be used to help students understand how to combine simple features into more complex patterns.

Background

When it is difficult to translate a given problem into logical rules, artificial neural networks are often used in problem-solving. Typical examples include the understanding of texts or the recognition of objects in images. The design idea for artificial neural networks originates from neurobiology and is based on the structure of the human brain. Analogous to a human nerve cell, which processes various stimuli and transmits an impulse, an artificial neuron also deals with various inputs and can transmit a signal. A weight is assigned to the input edges, i.e. they have a varying influence on the output of the neuron. An artificial neuron is thus somewhat similar to a human neuron, but functions more like a simple calculator: it multiplies the edge weights and input values, adds them up and passes on a result.

Just like in the human nervous system, many artificial neurons are connected and form a network in this way. The neurons are organized in layers. Depending on the complexity of a problem, a net can consist of two or more layers. In the initial situation of this activity, there are three layers. If a net has further layers between the input and the output layers this is called deep learning.

In practice, image recognition and classification usually work by means of so-called convolutional neural networks, which are specialized in recognizing patterns and are therefore very suitable for classifying images. This type of neural network is characterized by the fact that it uses so-called convolutions to extract characteristics and patterns from input data. Nowadays, these networks can classify images faster than humans.

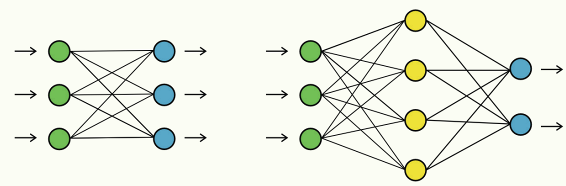


Fig. 5: Example for artificial neuronal networks - left: simple network, right: "deeper" network

How exactly does this work? Digital photos are composed of small color elements - pixels - arranged in a grid. Each pixel has a certain color value. For the computer, photos are - as opposed to humans - only numerical values. Initially, networks for image recognition try to recognize simple features. For this purpose, filters are placed over the image. This is similar to what we do in photo editing programs, for example, when we use a high pass filter (see Fig. 7). Ultimately, this is a mathematical calculation, which captures several pixels and calculates a new pixel.

Fig. 7: Applying a filter

Depending on the filter, it can be

Fig. 6: Neural nets identify increasily complex features

detected, for instance, that pixels with similar brightness values can be brought together to form edges. On a subsequent level, features such as horizontal and vertical lines, circles or corners are extracted. Common image processing programs such as Gimp allow entering such filters as a matrix, a good possibility to explore their effects yourself.

In the game, the drawings made by the students also serve as filters, as they extract the central elements of the object depicted in the photo. The sketches are then used to identify geometric shapes in the pictures. However, using only three features is a great simplification compared to a real neural network, which has several million neurons in a multitude of layers.

On the first levels of a neuronal network, there is a multitude of simple and rather geometric filters. These patterns are then (again by the application of filters) combined to more complex patterns. On "deeper levels" not only corners and edges can be recognized, but also parts of objects, like eyes or fur, and finally even complete objects, like dogs or cats. While being processed, (superfluous) information is repeatedly discarded, since, for example, the exact position of a diagonal line in an image is of little interest for the recognition of an object in many cases. In the end, a probability value indicates how likely an image can be assigned to a specific category.

However, a neural network cannot easily recognize the content of any image. Rather, the application frame is very limited: The neural net must first be "trained" with a very large number of images (several thousand). It learns which characteristics are decisive for images belonging to a certain category. Thus, the neural network can only correctly classify images whose category it already knows. For example, a net that is supposed to distinguish between dogs and cats cannot recognize other animals, but rather sorts them into one of the two familiar categories. However, a trained neural net can fulfil its task much faster than humans could ever do. Therefore, image recognition methods are already being used e.g. in self-propelled cars to detect various objects in road traffic (oncoming traffic, pedestrians, etc.) or in skin cancer detection.

Reinforcement Learning

Beat the Crocodile

Target group

Primary School Level, Secondary School Level

That's what it's all about

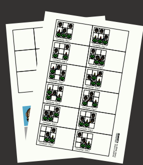
We are familiar with computers that can play chess and beat human players with superior ease. The Chinese board game Go, on the other hand, has long been considered so complex that only humans can master it - until Google used AlphaGo to frighten professional human players. In this activity, we see how computers can learn strategies for games, even though they only know the rules of the game.

These ideas are behind it

- Computers can learn by "reward" and "punishment".
- Computers evaluate the benefits of random actions based on reward and punishment.
- Computers learn strategies or action sequences by striving for maximum reward.

What you need

- Per pair of students: 1 "mini chess" field, 3 monkey and 3 crocodile cards, 1 overview of possible moves
- Colourful sweets (e.g. chocolate tokens) or paper tokens to evaluate the moves in 4 different colours (yellow, red, orange, blue; approx. 20 per colour)



Here's how it works

Two students each play a game of "mini chess" against each other. One student assumes the role of a "paper" computer. At first, the computer selects its moves randomly, but gradually learns with a candy token system which moves help it to win and which ones end in defeat. Using the strategy that develops this way, the computer gets better and better over time.

This activity is based on an idea of CS4Fun.
(http://www.cs4fn.org/machinelearning/sweetlearningcomputer.php)

Context

How do humans learn to play board or video games? Maybe we watch others playing or try out how certain actions or moves influence the game. The more frequently we win, the better we get in a game. We develop strategies to determine which moves are most successful in certain game situations. In the same way, a computer learns to play games.

Activity Description

The game follows simple chess rules: Each piece moves like a pawn, i.e. it can only move straight ahead and can only beat opposing pieces diagonally. A student takes over the monkeys and acts as a human player. Another student assumes the role of the computer in the form of crocodiles.

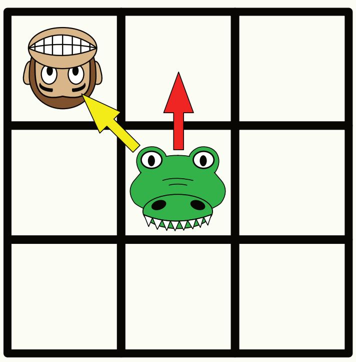


Fig. 8: Possible movements of a game piece

One side has won if it manages...

- to lead a piece to the other end of the playing field.
- to beat all opposing pieces.
- to ensure that the opponent cannot make any more moves in the next round.

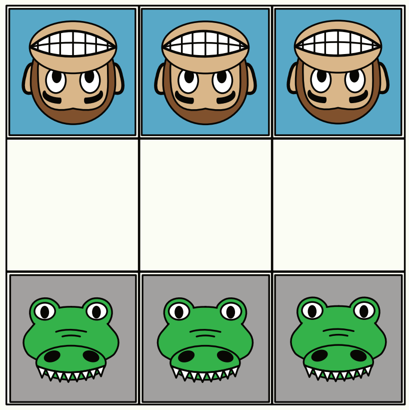


Fig. 9: Board before the game starts

As preparation, printouts of the computer's move options are spread out in front of the player taking over the crocodiles. Then, chocolate tokens (or other colored tokens) are distributed onto these moves. For each coloured arrow, you place a corresponding colour token in the area to the right of each game situation (see Fig. 11).

The human player starts. He or she can move freely according to the rules of the game. Then it's the crocodiles' turn. The player compares the current playing field with the possible moves and selects the appropriate playing situation from the given alternatives. For quicker orientation, the turn each playing situation belongs to is indicated. In the first round only the two possibilities for turn 1 have to be considered, in the second round the 10 moves for turn 2 and in round 3 the 7 moves for turn 3. Symmetric game situations are not listed twice. The crocodile player then closes his or her eyes and randomly picks one of the tokens placed next to the respective game situation and shifts it to the depicted board. The colour of the token determines which move is made and the player moves the piece according to the arrow of the same colour. If, for example, a red token is drawn, the crocodile is moved following the red arrow.

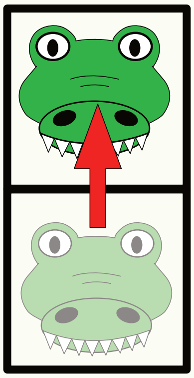


Fig. 10: Crocodile is moved along the red arrow

This procedure is repeated until the winner of the round has been determined. Before a new round is played, the computer adjusts its strategy as follows:

- Crocodiles have won: An additional token in the colour of the last winning turn is placed on the square of that turn.
- Monkeys have won: The chocolate token that determined the last move of the crocodile player is removed. The player of the monkey may eat it.

In addition, all tokens are placed to the right of the playing field again.

Optional: In order to simplify the rules, you can also neglect adding additional tokens when the crocodiles win.

Background

At first, the computer will hardly have a chance to win because it chooses its actions randomly (by picking a token with its eyes closed). The more games the computer finishes, the better it gets: it "learns" which moves help to win and which moves should be avoided because they ended in defeat in the past. In this way, the computer's strategy is refined gradually.

Since the computer is punished for losing and rewarded for winning, we also speak of reinforcement learning - learning through reward and punishment:

- Punishment = Taking away a piece of candy in a match that led to defeat.
- Reinforcement = Adding a piece of candy to a turn that led to victory.

This procedure "sorts out" those moves that resulted in defeat, so that at some point only "good" moves remain. In practice, strategies that do not lead to success would not be eliminated immediately, but only the probability of their occurrence would be reduced. In this way, the AI system gradually learns which strategy is most suitable in which situations, but does not immediately exclude individual strategies that have not led to success in every case. Although this procedure is simplified in the game by immediately removing moves that have led to defeat, it can never happen that all possible moves are eliminated for a game situation. For each situation there is at least one possible action that does not lead to an immediate defeat.

In this way, computers can learn to win a game simply by knowing the rules of the game or its possible inputs. For example, if a computer learns to play the video game Super Mario, it will initially only hit the keys randomly. This could lead to the computer stopping for minutes or running into the same opponent several times. It analyzes the objects or pixels in the image and reacts with inputs. Its goal is to maximize the points scored in the game which act as a reward. The further the computer can move to the right, the greater the positive gain. Over time, it will learn, for example, that jumping increases its reward if an opponent is immediately to his right, as it advances further in the level by jumping over the opponent. In this way, an AI system's performance improves bit by bit in a game, whereby the system always tries to maximize its reward (or more exactly: a certain function).

As part of the decontextualisation, have students analyse how the computer's behaviour develops. It should become clear that the computer comes to an efficient game strategy by assessing purely random actions. Afterwards, for example, a video about the game Super Mario (see website) can be used to show how reinforcement learning takes place in a neural network. Let the students reflect on the limits of the strategies learned by the system. You can combine this activity very well with the Back to the Roots: Crocodile Chess and Classic AI-activity to highlight the contrast between learning systems and traditional AI applications such as rule-based systems.

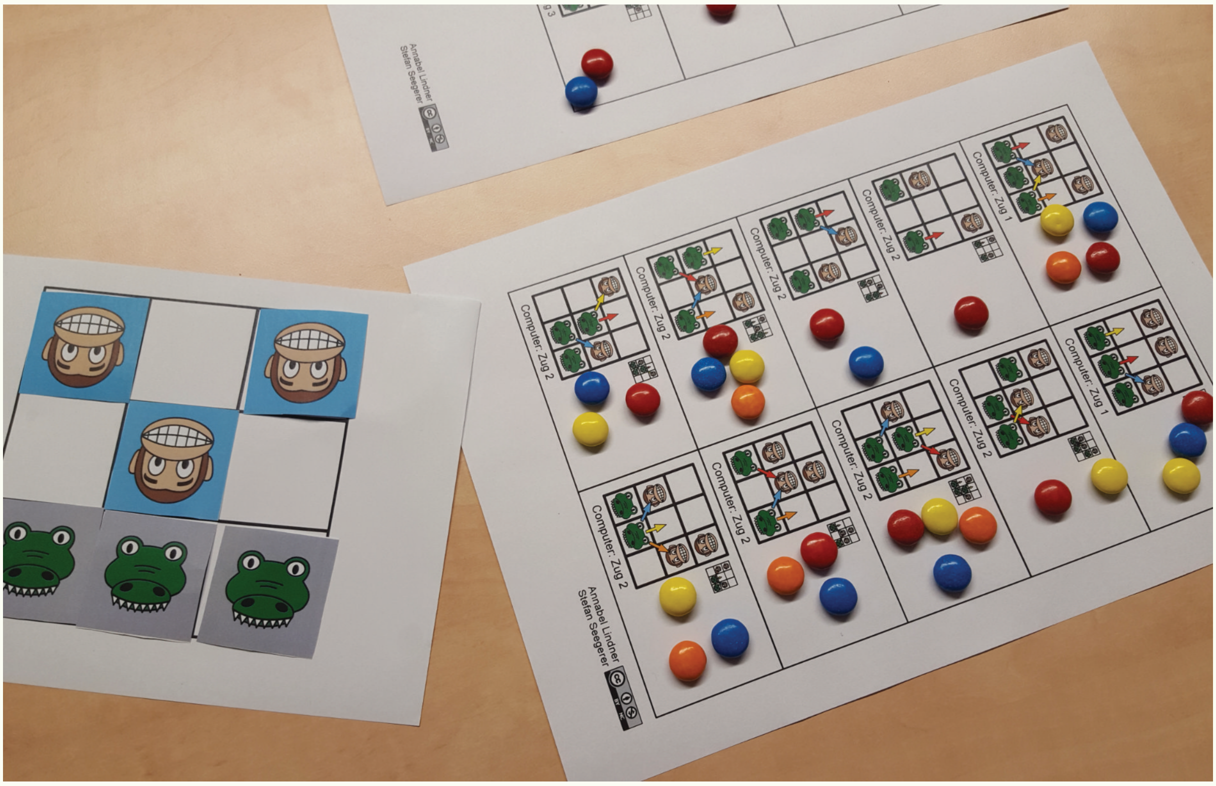


Fig. 11: Game setup: The distribution of the chocolate tokens shows the strategies learned

Back to the Roots

Crocodile Chess and Classic AI

Target group

Primary School Level, Secondary School Level

That's what it's all about

The previous exercises deal extensively with learning AI systems. But that's not all AI has to offer: The origins of AI are in logic and the idea to formalize knowledge by using a mathematical description making it available for machines this way. The differences between learning AI and traditional approaches and the limits of these systems are shown in this activity. For this purpose, the preceding *Reinforcement Learning* activity is performed with an expert system and thus illustrates the very different approaches.

These ideas are behind it

- Knowledge must be representable in a formal way in order to be processed automatically.
- Expert systems can combine rules and facts to generate new knowledge.
- Such AI systems do not make independent decisions but work according to the rules of logic.
- AI systems have processing mechanisms for automatically deducing information from existing knowledge.

What you need

- Per pair of students: 1 "mini chess" field, 3 monkey and 3 crocodile cards, 1 overview of rules for the next move

Here's how it works

Just as in the Reinforcement Learning activity, two students play a game of "mini chess" against each other. One student assumes the role of a "paper" computer. If, as recommended, this activity is combined with the previous *Reinforcement Learning* version, it is a good idea to exchange roles. Instead of randomly choosing its moves, however, the computer now works according to predefined rules, which are made available as copies.

Context

How can a computer be programmed to play board or video games? Computers can only "understand" the rules of a game and act accordingly if the rules are presented in such a way that a computer can process them. Knowledge must therefore be formally represented to be available for machine processing (e.g. by mathematical terms). In this case, the computer can evaluate it with the help of logic and derive its actions from it. AI systems are therefore not really intelligent, but skillfully use different possibilities to derive their behaviour from the knowledge available.

Activity Description

The game follows simple chess rules and has the same basic conditions as described in the *Reinforcement Learning* activity: Each piece moves like a pawn, i.e. it can only move straight ahead and only hit opponent pieces diagonally. One student takes over the monkeys and acts as a human player. Another student assumes the role of the computer in the form of the crocodiles.

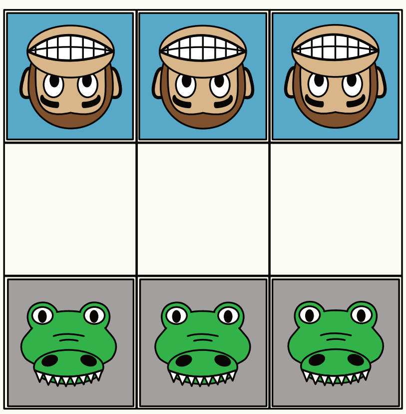


Fig. 12: Board before the game starts

One side has won if it manages...

- to lead a piece to the other end of the playing field.
- to beat all opposing pieces.
- to ensure that the opponent cannot make any more moves in the next round.

In preparation, the player taking over the crocodiles is given a printout of the rules for the computer's moves. These replace the move options and tokens from the Reinforcement Learning activity. The human player starts. He or she can move freely according to the rules of the game. Then it's the crocodiles' turn. The player compares the current game situation with the “rule table” and selects the appropriate scenario from all 10 options, symmetrical situations are not listed twice. Then he or she makes the move that the rule demands.

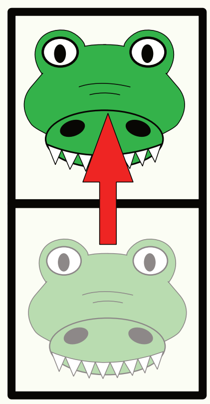


Fig. 13: Crocodile is moved alongside the red arrow

This procedure is repeated until a winner is determined. Several rounds can be played to check whether the computer is always able to win with the help of its rules.

Afterwards, present the extended version of the crocodile chess game with 4x4 fields (see website) to the students. Highlight the fact that the rules available to the computer are no longer sufficient. Let the students draw a comparison to the mini-chess variant in the Reinforcement Learning activity, in which the computer is learning: even here the "knowledge" of the computer is no longer sufficient. This is where both systems reach their limits.

The students are now thinking about what procedures are necessary to adapt the computer's "knowledge" to the 4x4 version of the game. They realize that the rule-based computer has to be adjusted manually by humans by adding new rules about the best move. In comparison, the learning system can learn the best behaviour for the 4x4 field in the same way as it did in the previous activity, i.e. by assessing random behaviour, as soon as all new possible moves have been added. The learning system thus needs a new training phase in which the knowledge for the extended game is implicitly acquired, while the new rules must be explicitly added to the rule-based expert system. Here, the human being has the task of first determining the best move for each game situation from all possible moves and then formalizing all rules completely and in a comprehensive way. This is not necessary for the learning system. This explicit formal representation of rules that control the system's actions is, however, hardly, if at all, possible for complex and multi-step problems. In this context, learning AI systems - in comparison to expert systems - offer the great advantage that they can determine such procedures "independently". Moreover, they can even detect correlations in data that cannot be identified by humans, e.g. because the consideration of many thousands of data records is necessary to find them.

Background

Logic and knowledge processing play an important role in many areas of computer science and are furthermore core topics of artificial intelligence. Since the natural language is ambiguous and too diverse to be an appropriate medium for making knowledge accessible for machines, the question about the best possible representation for machines is of decisive importance since the beginnings of AI.

In this context, approaches of this traditional form of AI rely on symbolic knowledge representation, i.e. the explicit representation of knowledge in computer systems, for example with the help of logic. This enables the unambiguous, uniform and precise representation of knowledge, which is necessary for the processing with a computer. Such representation methods are used, for example, in rule-based expert systems, which still play a role in commercial applications today. In such systems, logical statements that represent facts and knowledge about rules are automatically used to draw conclusions about how the computer has to act. In this activity, the facts correspond to the current game situation and the rule knowledge to the instructions which move has to be made.

The factual basis represents valid statements. A set of rules represented in "if ... then" form forms the rule base. Formally, this "if ... then" form can be expressed, for example, by propositional logic. A control system (inference engine) selects suitable rules based on the facts, evaluates them and acts accordingly. In complex expert systems, the conclusions drawn from rules can also serve as input facts for further rules and thus contribute to the expansion of the factual basis. In this activity, the task of the control system is taken over by the student who plays the role of the computer. With the help of the move specifications, which represent the rule base, he or she has to deduce the next move from the current game situation.

In a rule-based system, this procedure is called data-driven or forward chaining, because it is tried to achieve a yet unknown goal on the basis of facts. In contrast to this, there is backward chaining, which attempts to prove a hypothesis. In the game we are playing "backwards". Starting with a game situation in which the computer has won, we are trying to deduce the moves necessary to get there.

Even machine learning, which is the dominant method in artificial intelligence today and has already replaced expert systems and other traditional AI applications in many areas, cannot do without knowledge representation. In technologies such as neural networks, however, this is done implicitly, so that one speaks of sub-symbolic systems in this case: A systematic behaviour is trained and thus a kind of implicit knowledge about underlying correlations is acquired. However, it is difficult to gain insight into the concrete solution processes in these networks, since general rules underlying the data are only indirectly represented in the neural network, for example in edge weights and activation thresholds of the neurons. More about neural networks can be found in the activity #deeplearning. CS4FN also offers another activity that models an expert system for noughts and crosses in an unplugged way.

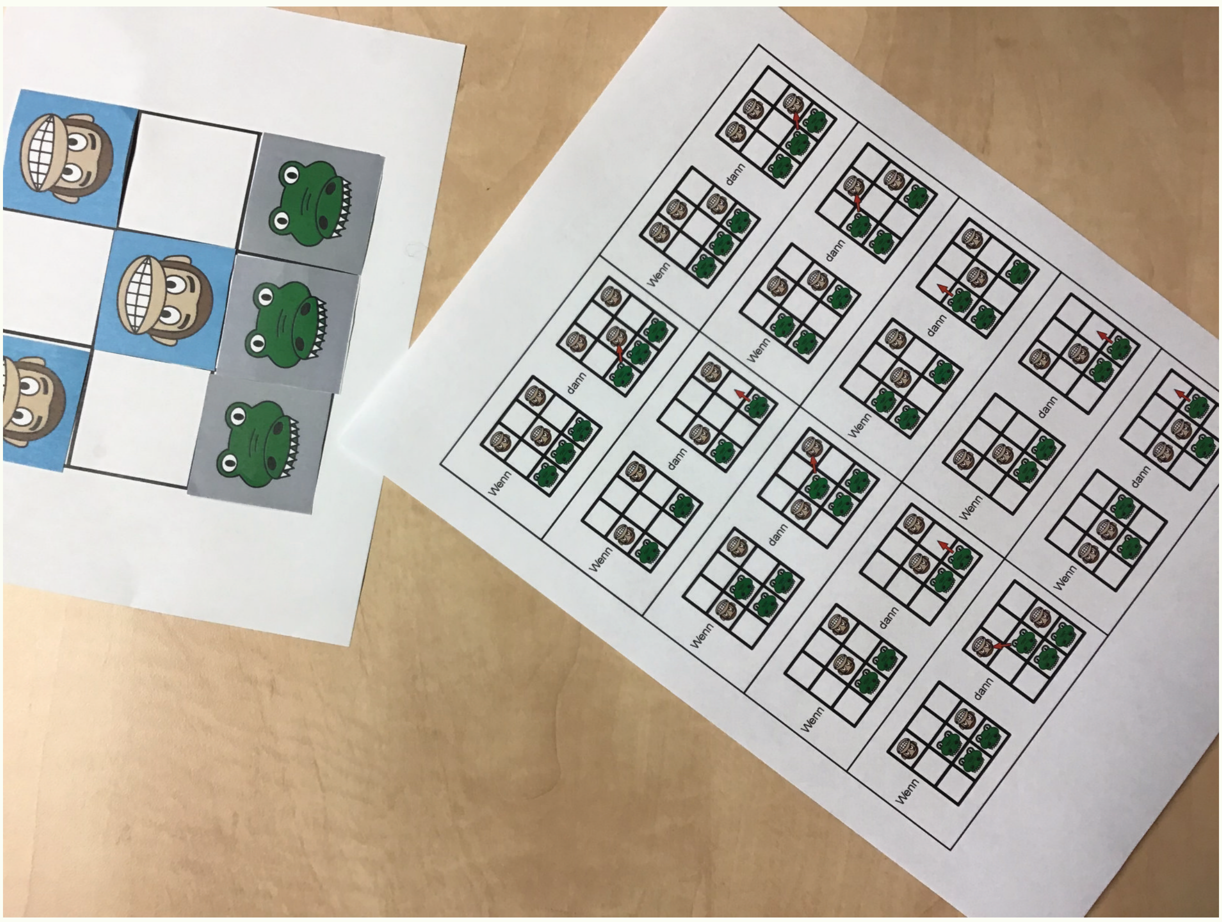


Fig. 14: Game setup: The rules for the person taking over the computer are clearly defined.

The Turing Test

“And oh! I am glad that nobody knew I'm a computer!”

Target group

Secondary School Level

That's what it's all about

How does a machine have to behave in order to be considered intelligent? What exactly does artificial intelligence mean? Researchers have been working on these questions since the beginnings of artificial Intelligence. With the Turing test, Alan Turing came up with an idea on how to determine whether a machine is intelligent in 1950.This activity reenacts the Turing test with students and aims to stimulate discussion about whether computers can actually show something like human intelligence. It also reveals how easy it is to be misled by a machine through carefully chosen examples of "intelligence".

These ideas are behind it

- Intelligent systems use certain strategies to imitate human behaviour.
- Special methods are needed to evaluate the intelligence of machines.
- The definition of (artificial) intelligence is not clear.

What you need

- Worksheets/slides with given Turing test questions for the whole class
- A copy of the answers to the Turing test questions
- 4 voluntary students in the roles of computer (1x), human (1x) and runners (2x)

Here's how it works

In this activity, students play a question-and-answer game in which they try to distinguish a computer from a human being by asking questions and analysing the answers. One student assumes the role of a computer, another one simply reacts as a human being. They are questioned by their classmates and the class has to determine who represents which role based on their answers.

Context

For centuries, philosophers have disputed whether a machine is capable of human intelligence or whether the human brain is perhaps just a very good machine. Some people think that artificial intelligence is an absurd idea, others believe that we will eventually develop machines that are as intelligent as we are. Artificial intelligence has a lot of potential, but on the other hand, the idea of intelligent machines also fuels fears.

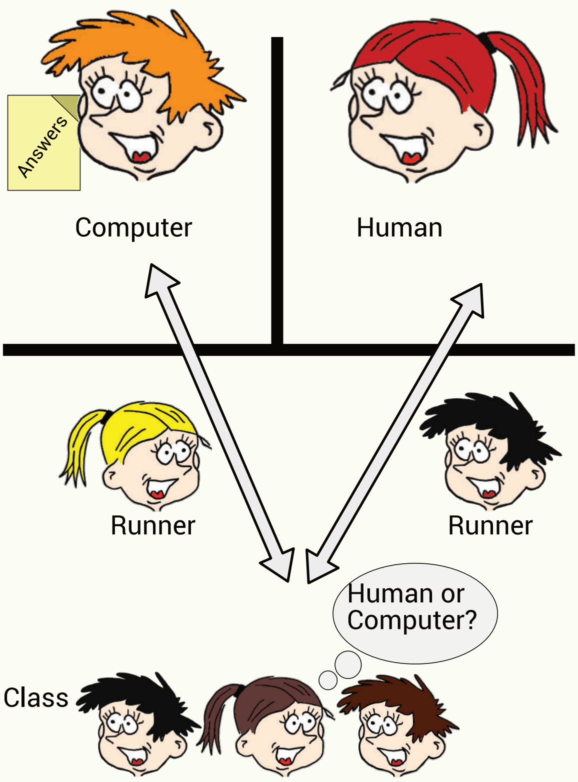


Fig. 15: Design of the Turing test

Activity Description

Before starting the game, discuss with the students whether they consider computers to be intelligent or assume that computers will ever be intelligent. Ask them how to decide whether a computer is intelligent and briefly introduce the Turing test, which is simulated in the activity.

To prepare the activity, four volunteers are selected to take on the roles of a computer and a human being (see Figure 15). In addition, there are two runners who ensure the fair course of the game and are equipped with a piece of paper and a pen to note down the answers. The roles of 'human' and 'computer' are secretly assigned by the teacher before these two students leave the classroom and head into two separate rooms (alternatively you can use partition walls, but make sure that the students do not see each other). The student assuming the role of the computer receives a copy of the answers to the Turing test questions. Each of the runners is responsible for one role, which one is also kept secret.

Now, the class has to find out which student has assumed the role of the computer. To do this, they select one question per round from the worksheet distributed, which is to be asked to the computer and the person. After a question has been chosen, the students should explain why they consider this question suitable for distinguishing the computer from the human being. This argumentation is the central element of the task, as the class reflects on how the answers of a person and an "intelligent" computer might differ.

Next, the runners pose the question to their classmates in the other rooms and the answers are brought back to the class. The human being is obliged to answer the question briefly and honestly - in other words, to give a human answer. The computer, on the other hand, selects the appropriate answer from the worksheet. If the instructions are written in italics, the computer has to work out an answer itself (e.g. the current time). In transmitting the answers given, the runners must be particularly careful not to reveal with whom they are interacting.

The class now discusses which answer is likely to come from a computer. Repeat the process with a few more questions, if possible until the class can make a clear decision about who the computer is. If the class cannot reliably distinguish between human and computer, the computer has passed the Turing test.

Background

Although no current computer program disposes of anything like general intelligence, the question of whether computers are basically capable of it is still unanswered. This is mainly due to the fact that the very definition of intelligence is controversially discussed.

Against this background, the British mathematician Alan Turing proposed a method for determining the intelligence of a machine without needing an exact definition of intelligence in 1950. This so-called Turing test lets the computer demonstrate its "intelligence". The scenario of the test is similar to the activity described above: A questioner interacts both with a person and a computer via chat. If he or she cannot reliably distinguish between the two, the computer has passed the Turing test. Since communication takes place via chat, the computer cannot reveal itself through physical characteristics, such as voice pitch. A well-known example of such an interaction system is the chatbot Eliza. The answers given by a student in the role of the computer are not unlike those given by an "intelligent" computer program. Some of the answers will very quickly expose the computer: a human will hardly be able to give the root of 2 to 20 digits. Other questions, in which the computer always uses a certain answer pattern, will reveal it only after some time. For example, answers to "Do you like XY?"-questions are not conspicuous when viewed independently. However, if you combine several questions of this type, it becomes clear that the computer works formulaically to generate answers from the questions. The answers can also show that the computer has misinterpreted a question, although this could also happen to a human being. Many answers are vague and further inquiry would make clear that the computer did not really understand the content of the question. Moreover, it is often safer for the computer to answer with "I don't know" (e.g. to the question about the root of 2). This feigns human traits, but can also lead to unmasking if this tactic is used too often or with too simple questions. Delayed and erroneous answers, for example to arithmetic problems, can also mislead the questioner for longer. Computers are thus able to feign their ability to talk, for example by formulaic answers, mirroring the statements of the interlocutor, reactions to keywords, the use of idioms and the resumption of topics, but this is only a facade that is easy to see through.

Further Ideas

Links and details about these activities can be found on our website.

Face Recognition

Our front door can distinguish us from the postman, our photo management software automatically tags our friends: face recognition is a common application of AI. In doing so, the technology should be as flexible as to recognize us even in winter with a cap and in summer with sunglasses. This activity conveys this principle through cartoon characters.

Monkey, Sherlock Monkey

How can knowledge be represented in such a way that a computer can "understand" it and draw logical conclusions from it? Logic and formal knowledge representation are of great importance here! AI systems are therefore not really "intelligent", but cleverly use different possibilities to represent knowledge. This kind of knowledge representation can also be mapped in logic puzzles: Corresponding puzzles require the combination of different facts according to certain rules, in order to then find a solution.

Brain-in-a-Bag

In this activity, students simulate the functioning of a neural net themselves with cords and toilet paper rolls. The final net is then able to play a game.

Unsupervised Learning

In addition to Supervised and Reinforcement Learning, there are also so-called Unsupervised Learning procedures: computers learn without previously known target values and without rewards. From a set of data points alone, categories (e.g. customers with high purchasing potential in webshops) or anomalies (e.g. suspicious activities on web servers) can be identified. Use chalk to draw a grid of coordinates (e.g. in the schoolyard) and ask your students to position themselves appropriately in the grid using the two axes. Depending on the axes selected, not only clusters but also outliers or anomalies can be identified.

Imprint

Editor:
Professorship for Computer Science Education
Friedrich-Alexander-Universität Erlangen-Nürnberg
Martensstraße 3
91058 Erlangen
https://aiunplugged.org

Editing and design:
Annabel Lindner, Stefan Seegerer