

Presentation Analysis – Artificial intelligence in autonomous vehicles

Sergej Lamert 00727245

Department of Applied Computer Sciences, Technische Hochschule
Deggendorf

AI-7/IAS-7: Presenting in English

Neal O Donoghue, M. A.

22.01.2021

Contents

1. Introduction.....	2
2. Main Part	2
2.1 Transcript	2
2.2 Analysis	4
3. Conclusion.....	6
4. References	6

1. Introduction

Artificial Intelligence gets more and more important in our lives every year. It's widely used, whether in data analysis in social media or in groundbreaking research like DeepMind's AlphaFold. Since A.I. is getting so important, we want to give a glimpse of two of the most important concepts in A. I., namely neural networks and reinforcement learning, with autonomous vehicles as an example and how these concepts are interconnected with each other.

2. Main Part

2.1 Transcript

Now that you've heard Kevin's explanation of the most fundamental building block of autonomous vehicles – the sensorics, and Erik's explanation of the next building block: how neural networks use the generated data to detect people and streets. Now I want to lay the final building block on our tower and show how we can use this new information for our car to make decisions. But before I go into detail on how an autonomous vehicle makes decisions, we need to realize that the topic of reinforcement learning is like a web, it interconnects many scientific fields as this quote from David silver, a scientist at Google's DeepMind shows:

"For many [...] different fields of endeavour there is a branch of that field which is trying to study the same problem [as] reinforcement learning [...] – the science of decision making."

One of these many fields that conduct research on decision making is psychology. I'm sure that the words 'Dog' and 'Psychology', must ring a bell in some of you. I am talking about one of the most famous experiments in conditioning: Pavlov's dog. The psychologist

Ivan Pavlov conducted some experiments with dogs in the topic of conditioning. Now imagine this: Pavlov takes one of these golden retrievers (as seen in the presentation) and puts a nice juicy steak in front of the dog and the dog starts to salivate. Sometime after that he takes a bell and rings it. It's logical that there is no salivation. Now here comes the twist: if Pavlov gives said steak to the dog and simultaneously rings the bell, the dog gets conditioned to salivate on the mere stimulus of hearing the bell so that he associates "bell ring equals to food".

Some scientists took the results of Pavlov's experiments and asked: How can we do this in a computer program? For this they thought: You can put an agent (our car) in an environment (street) and the agent explores and interacts with the environment. In practical cases this is often done via simulation since you don't want a 'unexperienced' car to roam around freely in real life and wreak havoc. So basically, you simulate your daily driving experience, with people walking over streets and so on. The car navigates through the streets with specific actions: driver forward, backward, right, left etc. And while the car does that, the environment changes, either by itself or by the car. And now if we get back to conditioning: whenever the car does something, we don't want it to do, like hitting other cars, we "punish" it and "praise" it for doing something good. You probably think now: "How are we supposed to punish or praise the car, it's not a human, we can't hit it with a hammer and hope that it learns something from that". The answer to this is: We give him abstract "plus points" if the car does something good and "minus points" if not. This is usually called a 'reward', but we'll get back to this later.

Let's go one step further and look at the concept of how our autonomous car can learn from its exploration and interaction with the environment. For this, MIT offers an interesting "playground", where you can test your own RL algorithms, it's called "Deep Traffic". As you can see, it simulates a highway, autobahn or a dense street, which your car navigates through. Now let's dive deeper and define some terms, so that we all have the same common ground. In this picture you can see our cherry red car, this is our agent, all the white cars are just bots, they do random things to annoy you, just like drivers in real life. We can view the Environment as a 2D Grid with X and Y coordinates where every car, including ours has its specific coordinates. We can define by ourselves how often the grid should be updated with new positions.

An action is, as I already mentioned before, the things that our car can do. In our case it's what every car does: accelerate, decelerate, go left, go right. It seems that the MIT had to save some money since we can't drive backwards, but that's okay. As I said, a computer doesn't feel it if we punish it, but we can simulate it: we can introduce a positive reward for doing something good, meaning coming closer to a goal, and negative reward, doing

something that is the opposite like crashing into a car, or a person. The last point is the termination condition: When should the car stop? Unless it isn't in a chase with the police, we can define that he should stop after the reached 1000 reward points, meaning it passed 1000 cars.

How does the car actually learn things? First of all, RL algorithms are generally iterative, that means you'll have many, many, many sessions. One session is basically playing one round of the game from start to finish. For example, one session of chess goes from the start, until you blunder and shamefully lose. If we relate this to our example, one session goes from the start until the car crashes or reaches the goal. For every Action that the car takes, it saves all the parameters from above into a matrix. That means it saves: Where was I? What did I do? Where did I get? How did it work out? And since the algorithm saves everything it does, it can look up this Matrix at any time and so to speak 'look at the past' and make decision based on what happened back then, for example if the car was in a state A and took action B which was going to the left, and that resulted in a crash, the car will notice this in a future session and try other actions instead.

To conclude, you have seen, that autonomous vehicles are quite complex, we hope that we have given you a glimpse on the different levels an autonomous vehicle works on, the "vision" via sensorics, the use of this data to perceive the environment and objects in it, and last but not last, how a car can be trained to navigate through this digitally perceived world. I hope you enjoyed our little ride, have a nice day.

2.2 Analysis

The start is a recap of what already has been said by my teammates. With the 'building block' metaphor I tried to create a mental image in which the sensorics are the base, on top of that base comes the neural network and on top of that comes reinforcement learning. Since there are 3 'building blocks' the rule of three is also satisfied.

In my next step I create an analogy between Pavlov's dog in Psychology and the concept of reinforcement learning. For this I use the mental image of a web to show the complexity of the field of decision making and emphasize it with the quote of David Silver. After that I get to Pavlov's dog and introduce it with the metaphor of a ringing bell, "... must ring a bell in some of you". I used this exact phrasing, because I hoped that it may work subconsciously (people may associate a "bell" and "dogs" directly with Pavlov, despite the fact, that he used a metronome rather than a bell). In my next step I explain the experiment with use of the VAKOG system and show a representative picture of the experiment. I use

(both mental and physical) pictures of food and the sound of a bell to show how a dog can be conditioned.

To complete this section, I use the rhetorical question “How can we do this in a computer program?” to transit to the basic concepts of reinforcement learning. For this I show a diagram on how a reinforcement learning agent works in its most fundamental way. I link this diagram to the analogy of driving a car and give examples on how the concept transfers to a self-driving car. To emphasize, that the exploration of the environment is mostly simulated, I use the image of a “unexperienced” car wreaking havoc in real life.

At the end of this part I humanize the self-driving car with the metaphors of “praising” and “punishing”. I used these words because on the first glance they seem paradox to the listener because it’s obviously impossible to punish a car because it’s not a human being. Continuing, I put emphasize on this paradox with a rhetorical question in combination with a joke: “hit it with a hammer and hope that it learns something from that”. In my last step I resolve the paradox by answering the rhetorical question from above. I state that the praise and punishment will be abstract concepts that are called “reward”.

After I have shown the basics concepts of reinforcement learning with the diagram, I show a short clip of MIT’s “DeepTraffic”. The fact that it’s not a picture but a video (which looks like a game) should get the attention of the audience back to a higher level. I use the metaphor of a “playground” on it to make it more vivid. Then I begin to describe the road of the simulation with known terms: “highway, autobahn, just a dense street” (which is also compliant to the rule of three). In my next step I want to connect the concepts of RL from before with the concrete example of “DeepTraffic”. Since I wanted to keep it as simple as possible, I only used 5 technical terms: “Agent, Environment, Actions...”. Then I explain every term with real life examples that are most certainly known in my audience (since everyone in my audience is supposed to know how a car drives and what a coordinate system is). I try to make the explanation as lively as possible with mental images of a “cherry red car” or the feeling of bumping into another car. To describe the white cars, the “bots” I use humour: “they do random things to annoy you...”. While I describe the actions that the car can take, I joke about the fact, that in this simulation it can’t drive backward. For this I use the sentence: “...MIT had to save some money...”. To explain the termination conditions of the simulation I use humour again by creating an image of a police chase.

Before I get to my last point in the presentation, I ask another rhetorical question: “But How does the car actually work? I do this for a multitude of reasons, the first one being as a transition to the algorithm. The second reason is, that it might be already in some people heads, since all I did before was just the build-up for the upcoming part. For my final part I explain how a car learns by itself, for this I use again the example of MIT’s “DeepTraffic”. To

do this I use several methods. Firstly, I use repetition and the rule of three to emphasize that it takes many sessions: “many, many, many sessions”. Then I go on to explain what a session is, for this I use the analogy of a game of chess combined with humour: “... until you blunder and shamefully lose”. Afterwards I transfer the concept of a session the simulation, “until the car crashes or reaches the goal”. Secondly, I draw on the picture to emphasize what I speak, for example when I say “turn to the left” I draw an arrow to the left. Thirdly, I humanize the car and the algorithm with questions that the algorithm might ask itself, for example: “what happens if I take this action?”.

Lastly comes the conclusion, where I recap the whole presentation and how everything is building upon each other and finish it with the phrase “I hope you enjoyed our little ride”, which is a reference that our whole topic was about cars and how they can ride by themselves.

3. Conclusion

Concluding this transcript, I hope I was able to deliver a good presentation on how reinforcement algorithms work in principle and that this subject highly interesting since it deals with things that we do every day: making decisions. I found this topic so appealing because it combines both theoretical mathematical principles and practical results which can be applied in the real world as we have shown with the example of autonomous vehicles.

4. References

[1] DeepMind. (2015, May 13). RL Course by David Silver - Lecture 1: Introduction to Reinforcement Learning [Video].

Youtube:

https://www.youtube.com/watch?v=2pWv7GOvuf0&list=PLqYmG7hTraZBiG_XpjnPrSNw-1XQaM_gB&index=2&t=0s&ab_channel=DeepMind

[2] Parmar, B. (2019, Sep. 13). Pavlov's Dogs and The Importance of Environment in Learning

Medium: <https://medium.com/history-of-education-timeline/pavlovs-dogs-and-the-importance-of-environment-in-learning-b1cefd40d527>

[3] Surma, G. (2019, Jan 24). Deeo Traffic - DQN Tuning for Traffic Navigation.

Toward Data Science: <https://towardsdatascience.com/deeptraffic-dqn-tuning-for-traffic-navigation-75-01-mph-solution-23087e2411cf>