

SDS PODCAST EPISODE 71 WITH HADELIN DE PONTEVES



Kirill: This is episode number 71 with your favourite Hadelin de Ponteves.

(background music plays)

Welcome to the SuperDataScience podcast. My name is Kirill Eremenko, data science coach and lifestyle entrepreneur. And each week we bring you inspiring people and ideas to help you build your successful career in data science. Thanks for being here today and now let's make the complex simple.

(background music plays)

Kirill: Hello.

Hadelin: Hi.

Kirill: So, guys, welcome. There's no intro for this one because Hadelin's right here.

Hadelin: Hi guys. So glad to be back.

Kirill: Yeah. We're sitting in Portugal in his apartment, and we're talking about AI today, right?

Hadelin: Yeah, so exciting. We just finished a course and now we're going to talk about AI because we really went deep into it, right?

Kirill: Yeah, for sure. And Hadelin's wearing his favourite baseball cap backwards.

Hadelin: Yes, that's right. I feel so much more comfortable!

Kirill: Yeah, and I'm wearing some funky glasses, so we're all set for today and the topic is Artificial Intelligence, and we're

going to try and make it very simple, very accessible, especially for those of you who are not into the topic. And the other thing I thought of, just before this I had a couple of guests talking about AI, and I thought listeners are going to be like, oh cool, one person about AI, second person about AI, third person about AI, and then we had a break. The previous episode was Caroline McColl about recruitment, which was good. And now they're going to listen to this, they're going to be like "About AI again!" But don't take it that way, AI is an important topic, right?

Hadelin: It's an important topic in data science. As we said, we just finished a course, so we want to end on a good note about AI.

Kirill: Yeah, yeah, we want to, while the knowledge is fresh, share everything with you, and plus we'll make it as accessible as possible so to give you an intuitive and general overview of where the world's going in the space of artificial intelligence and what we learned doing this course as well.

Hadelin: Absolutely. We learned a lot of stuff.

Kirill: Yeah, yeah, like you mentioned, it wasn't a normal course for us.

Hadelin: Yes, I was learning with the students. That's amazing. That was an amazing feeling. Yes, to be honest, this course was on a very recent topic, on state-of-the-art models, so I was reading the research papers, I was getting informed. I didn't know all the models first, because very simply, since they're very recent, I didn't study any of them during my Masters of Research in Machine Learning. So I was very glad to learn about all this and especially with the students, so yeah, that

was a great experience. It was a very new experience, but a great one.

Kirill: Yeah. And I like how you mentioned that it was kind of like doing a PhD for you.

Hadelin: Yes. Definitely.

Kirill: You did all the coding, right?

Hadelin: Exactly. And it was exactly like doing research. You know, I had an experience of research, it was working for a company making a recommender system and I spent my time reading some research papers. So studying the algorithms, trying to combine them, and here that was the same. I was reading the research papers to see which model we would implement to solve some very challenging problems. OK, we're playing with games, but these were not easy games for an AI. So that was very challenging. So I had to go deep into the research papers and understand all this. Sometimes it's not very clear, because you know you get not much resource on it. So yes, it was a great experience. It reminded me of research.

Kirill: Yeah, yeah, for sure. So tell us a bit more. You mentioned games. People who haven't taken, or even heard about, this course that we created, why games? What's the point of creating AI for games?

Hadelin: Yeah, so this is thanks to OpenAI Gym. OpenAI Gym is a website launched by entrepreneurs like Elon Musk and Peter Thiel, and basically they made such environments so that we can have some kind of benchmark for the AI models we implement. So that's an all-made environment, and we

basically only need to implement an AI to test it and see if it can solve the games. If there wasn't an OpenAI Gym website, we would need to make the games ourselves. We would need to make all the games to find the environment, maybe making some map, defining the action. Well here, thanks to OpenAI, we don't have to do all these things. Everything is already well-prepared and we have some code to pre-process the images so everything is already ready, and we just need to implement the AI, and that's so cool because we can just focus on what's most important.

Kirill: Yeah.

Hadelin: But having said that, there is one module in the course –

Kirill: Yeah, I was about to say, the self-driving car.

Hadelin: The self-driving car, yeah. There is no self-driving car on OpenAI Gym, because that's basically not a game, so we had to make this environment and it was quite heavy, heavy code.

Kirill: Especially if somebody goes into the course, if somebody listening jumps into the course, it's the first module. (Laughs)

Hadelin: It's the first module. (Laughs) We announced it as the last module because we thought it would be the most challenging one, but not at all. This is the most simple one, and we're going to explain why.

Kirill: And in all fairness, even though creating an environment is complex, you provide the code. So, if somebody doesn't want to go through the tedious process of creating the

environment, you can just use your code and then just build the car from there.

Hadelin: Absolutely. And besides, it is commented, so they can still understand the code if they want.

Kirill: Yeah, fair enough. Cool. But I want to return to my previous question. It is great that OpenAI Gym has these games available and so on, but tell us a bit more about why teaching AI to play games is a valuable skill. How can you use that in the real world?

Hadelin: Because basically, the models we implement to solve games are the same models we will implement to solve real world business problems or even any kind of general problems. That's what we say in the video, you know. We need an environment on which we can train the AI on and that's all the games, but then the models, if they work on the games, they're very likely to work on real world environments or real world problems by only changing a few parameters. So that's why this is not only to have fun, because of course it's fun to play some games with some AI, but also it's very useful for more real world and useful problems.

Kirill: Yeah, for sure. And that is very well backed by an example which I personally love and I've given a couple of times already, the Google example, how they use their AI. So the same team that built Alpha Go, they created an AI—we can only speculate that they used similar concepts—to optimize the electricity consumption in one of Google's warehouses. They saved like 40% on energy consumption.

Hadelin: Yes, that's amazing. We can do amazing things with AI. For example, the first module about the self-driving car, we

implement some AI and I'm like 90% sure that we could use this AI for this electricity consumption optimization problem by only changing a few parameters. Because we basically only need to change the actions, which is just a set of a couple of numbers, indexes, and we also need to change how the reward is decided, how we're going to attribute a good reward and bad reward. We only need to change this and then we don't have anything to change in the model. Maybe some parameters to tune it, but it's very similar. So, feel free to take the challenge if you want to. That would be a very good practice.

Kirill: Yeah. And imagine that 40% decrease in energy consumption result in like 15% cut in the electricity bill. Imagine cutting your electricity bill at home by 15% or being able to cut the electricity bill of your organization by 15% or doing something similar like water consumption, whatever. These optimization problems are huge and when you get AI to do them, they can really save costs and improve efficiency in business.

Hadelin: Yes. And there's another pretty exciting example, you know, the JP Morgan Bank. They made an AI that solved a specific task in one second whereas this task was solved by hundreds of lawyers and 3,000 hours or something like that. So, it basically solved a problem in one second that they usually solved in 3,000 hours. So that's another crazy example of what AI can do.

Kirill: That's pretty cool. I think I've heard of that one. That's pretty cool. Okay, so let's talk about reinforcement learning. The AI that we're working with, exploring in our course, is

reinforcement learning type of AI. Let's get started. What is the first step that we explain in this course?

Hadelin: That is Q-learning. Q-learning is at the base of reinforcement learning. Let's try to explain in a few words what it is. Basically, Q-learning consists of learning Q-values. That sort of represents how well an action does in a specific state. So, for example, you have your environment, you're going to play an action to reach the next state, and the Q-value represents how well this action did by being played in this state.

And then in Q-learning there's no neural network yet because basically you've just learn this Q-function inside an equation, which is the Bellman equation, and iteratively you come to a better Q-value each time, making better actions and therefore solving the game or whatever the goal you want to accomplish. That's Q-learning, that's at the base, and of course we go much further than this, higher level.

Kirill: Yeah. So, just on Q-learning, I wanted to highlight a very important point, which you said – iterative, right? Iteration is key here. AI cannot learn without having the ability to try things over and over and over again. So that's something you have to keep in mind and it's good in these artificial or virtual environments, because you can afford to try it as many times as you want. But then when you get to real life environments, when you have like self-driving cars in real life, that's why you see Google, when they release a self-driving car, it doesn't just go on the streets right away. It has to drive for hours and hours. Even if they think it's doing good, it should still keep driving and getting experience.

- Hadelin: Exactly, because it has to explore and practice. Exploration is another very important principle of reinforcement learning – exploration and exploitation.
- Kirill: Yeah, and we will get to that in a second as well. Actually, on that note, I also really like that quote from I think Elon Musk or Tesla, they said, “The more you drive, the more we learn.” Because the Tesla car has AI built into it, so they’re not only integrating certain rules on how it should drive. It’s actually learning from your driving. And that’s why Tesla, for instance, has a huge advantage, in my view, over companies like Google, or now Google self-driving cars are called Waymo, or other self-driving cars that are popping up here and there. Tesla has this whole user base of people who are driving and they’re learning from them driving, so it’s huge.
- Hadelin: They have the advantage of data. That’s an amazing and powerful advantage.
- Kirill: Yeah, for sure. And before we go on, just to kind of concrete this in, I think I may have touched on this in a couple of other podcasts with other guests, but I’d like for you to reiterate this. What’s the difference between artificial intelligence, like reinforcement learning, and some other machine learning algorithm, where things are like, you know, the heuristic type of algorithms where things are like pre-coded into it?
- Hadelin: It’s basically static vs. dynamic. For example, if we look at the other models we made in the other course, most of the time we only had to make one prediction. You know, we had a dataset, it learns the correlations on the dataset, or even some complex datasets like a bunch of images. And then we

give it a test of a couple of data or observations or images, and it makes some predictions. But it's static. And this time with AI, it's like we're making a prediction in real time. For example, when we play 'Breakout' or when we play 'Doom,' each time we have to predict the action we want to play in a dynamic way. That's the big difference with what happened before.

Kirill: I totally agree. And would you say that—again, going back to the previous year, the example of Alpha Go is AI, whereas the example of Deep Blue in 1997, I think, when Deep Blue beat Garry Kasparov in chess, that was static AI. Like, even though it was exploring stuff, it had predetermined rules on how to take on situations.

Hadelin: Exactly. Here everything is stochastic. We play our action based on what we call the Softmax method, which basically makes a distribution of probabilities of the different actions we can play. And then we take a random draw of these distributional probabilities and we play the action that most of the time has the highest probability, but thanks to the Softmax method, we still explore the other action. That's all stochastic, and as you just said, this is not deterministic anymore.

Kirill: Okay, gotcha. And now moving on to the next step. After that Q-learning, which is like module zero in our course, even though it sounds complex, it was the most basic thing that we covered off.

Hadelin: Yes, but it's important to understand it.

Kirill: Yes, of course. It lays the foundation for everything else. What was the next step?

Hadelin: The next step is therefore module one, that's the self-driving car, and in terms of AI and models, well, that's the first step we enter into the deep learning AI because we have a neural network. But that's where we start to try to build a brain like a human brain. And that's because in this first brain we made for the self-driving car – and by the way, I think we called the object 'brain' to represent the AI, that's because it's just a neural network – well, we just have some fully connected layers which are going to take as input an encoded vector describing what's happening in the environment, and we call it 'the state of the environment,' so that's the input state.

Then the signal goes into the neural network, only in fully connected layers, and then the neural network outputs prediction, which actually are the Q-values. That's why the Q-learning is somehow integrated into deep learning, so it outputs the Q-values, and then based on the Softmax technique that we just spoke of, we're going to output an action, it's going to play an action. That's very exciting because that's our first step of trying to make something that looks like a human brain, but only with fully connected layers. That's like with only neurons that have transmissions. But then you're going to see what's going to happen in module two and module three. We make something that looks more like a human brain and we'll discuss that soon, right?

Kirill: Yeah, in just a second. And on this I actually wanted to mention that—for those of you who are listening, we just want to get an overview of AI and we have no intention or no plans of getting deep into it right now – maybe in the future

some time – but something to take away from here is the concept of Q-learning. We've already touched on it and Hadelin explained what Q-learning is. It's about assessing the value of actions, potential choices that the AI has, possible actions that it can take in a certain state, and assessing how valuable each one of them is and then deciding which ones to take based on that.

Well, in this first module in the course that we just talked about, we are taking that concept of Q-learning, of Q-values, which is a normal concept which has nothing to do with neural networks or deep learning. That's very important to understand. It was developed in the middle of the previous century by Bellman, hence the Bellman equation, so it's very, very old. Neural networks didn't even exist then, they came around in like the 70s or 80s.

But now, in module one, we're combining the two. So we're taking this mechanism of learning through potential values and we are adding a brain which mimics the human brain to that. And what you get is kind of like an AI which you can think of as a child. Like when a child learns to walk, you don't give it a set of rules on how to walk and it's not coded into the DNA. It just basically tries to do this, tries to get up, tries to fall over, it just keeps trying different things and eventually the brain restructures itself in a way that it remembers which actions lead to the good results and basically what's happening is remembering which actions have the best Q-values, which are the most valuable actions to achieve this task of getting from the position of sitting down to where the mother is sitting, or to where the food is, or to where the cat is, or whatever the baby is facing. So

that's what we're combining synthetically in this first module when we're putting together neural networks and Q-learning principles.

Hadelin: Yes. And if some of you are wondering what is this Q-value exactly, how does the neural network know how to output such a Q-value, it's very important to understand that there is an important thing next to it, which is a target. So we have a target, there is a specific equation for the target, a specific computation, and the goal of the neural network is to get the Q-value close to the target, to minimize the difference between the predicted Q-value and the target Q-value. That's how it's knowing how to do the computations to compute the Q-values.

Kirill: For sure. That's a great overview of Q-values and hopefully, if you're going to take anything away from this podcast, now you know a bit more about Q-values and how they operate.

Hadelin: Yes.

Kirill: Let's move on to module two. What happened after the self-driving car?

Hadelin: Yeah. So, continuing on what we said about trying to make a brain that is close to a human brain, in the second module we implement an AI to take the challenge to beat 'Doom' and this time we add a specific feature to this brain, which are the eyes. We add them with what we call the convolutional neural networks and that's basically—instead of taking as input an encoded vector, it's literally going to take the images, exactly like a human playing a game, watching the game with his own eyes.

So this time we make a deep reinforcement learning model, because this time we add a convolutional neural network as the first layers of the brain. So we still had to fully connect the layers that come after the convolutional neural network, and basically we take it to the next level now with those eyes. That's where we enter into the deep reinforcement learning world, and that's because this time we are playing with eyes. And we get some really good results. Actually, the last practical tutorial is very exciting because—

Kirill: How surprised were we there.

Hadelin: Yeah, not only were we surprised, but very excited and happy with the results. You should look at our faces when we see the AI killing all these monsters.

Kirill: It's like a corridor—you know the game 'Doom,' guys, right? It's an old school game where you have to kill these monsters and collect weapons and go through levels. So there's like a corridor which you have to go through, there's like six monsters guarding it, and in the end is like this vest that you have to get.

Hadelin: Yes, the goal is to reach the vest.

Kirill: Reach the vest as fast as possible. But our AI just like picks up this gun or had this gun and it's like bam, bam, bam – just kills five monsters and gets the vest.

Hadelin: It kills all the monsters. No—

Kirill: Five.

Hadelin: Except one, yes.

Kirill: It's just crazy. It's just nuts.

- Hadelin: Yeah. You should really see the look on our faces when we found that.
- Kirill: Yeah. That tutorial, by the way, we specifically made it a free preview tutorial. So even if you're not inside the AI course, you can just head over onto it if you want to check this out and just check out this tutorial. The looks on our faces are worth seeing it.
- Hadelin: And besides, there's something that happens after this video of the AI killing all these monsters. We're not going to say what it is, but we are even more surprised to see it happen, right?
- Kirill: What was it?
- Hadelin: You know, runs...
- Kirill: Oh, yeah, and then just like—yeah, smashes the round completely. But yeah, that's what we learned about deep convolutional Q-learning, so that's pretty cool. Is there anything else we want to add on that?
- Hadelin: Well, basically the important thing is that we're adding some eyes to the neural network, so the AI has eyes and then it's the same principle. We just changed the way of getting the input. This time it's not an encoded vector, it's literally the input images.
- Kirill: Yeah, the input images which then go through a network and then they become like a vector anyway. So, the way we did the self-driving car is kind of like an easier implementation for the AI because it gets information about the state through coordinates and stuff, whereas here it's like you say, exactly the same as human would see.

- Hadelin: Yes, exactly. You're right, we have an encoded state that happens after the convolutional layers, but that's because whatever we implement, we still have to have this flattened vector that becomes the input of fully connected layers that will output a prediction.
- Kirill: Yeah, that's right. It's like when you see in real life with your eyes. You see an image, but in reality your brain is working with electrical signals, right?
- Hadelin: Yes, exactly.
- Kirill: Same thing in the AI.
- Hadelin: Yes. We can maybe add something about eligibility trace, remember?
- Kirill: Okay.
- Hadelin: So, we add a special feature in this module two. Actually, we tried it with a simple deep convolutional Q-learning model and we didn't manage to solve 'Doom' at the beginning, so then we added this algorithm which is an algorithm of research papers. Speaking of getting your head into the research papers, what I mean by that, there is this algorithm and this research paper that implements what we call eligibility trace, or n-step Q-learning. That basically means instead of getting the reward each time and getting the target at each step, we're going to get a cumulative target over ten steps, or even more steps, and a cumulative reward. So that basically changes the equation. Instead of having an equation at each step, we get some equations every ten steps.

Kirill: Gotcha. So that you guys don't zone out at this point in time when Hadelin starts talking about rewards and cumulative stuff, I'm going to add a bit. Is that cool?

Hadelin: Yeah, sure.

Kirill: So, when we talk about reward, because we haven't gone into this yet in the podcast and I want to clear this up, as your artificial intelligence progresses through an environment, whether it's through playing 'Doom', or driving a self-driving car, or playing another game, or optimizing electricity consumption, or whatever, whatever it is, it is constantly getting feedback from the environment. It's either getting good behaviour points or it's getting bad behaviour points. So it's either getting rewarded or it's getting punished. And based on that feedback, it can understand whether its actions are good or bad.

So if it gets a reward from taking a certain action – for instance, killing a monster in 'Doom' – it gets points and it can see those points. It realizes, "Oh, okay, cool, I should do more of that. It's good to kill monsters." Whereas if it gets a punishment – for instance, its health goes down when it doesn't kill a monster and the monster shoots it and it doesn't dodge the missile, then it realizes, "Oh, I should do less of that. I should not get in the way of monsters because my health goes down and that's a punishment."

It's very similar to real life. If we go back to the example of a baby learning to walk, when it's starting to learn to walk, when it falls over, it hurts itself and its nervous system — it's ridiculously surprising how the real world is so similar to AI — the nervous system of the baby gives it a bruise, at first

it goes purple so that's probably a signal to the parents first that something went wrong, but the nervous system of the baby gives it pain in the head. Pain is a concept that is generated inside our body. It's nothing more than an electrical signal in your brain telling you that you have pain. In reality, it doesn't exist. It's something that your brain made up for you to train you to not fall over more. So basically, when a baby falls over, it gets feedback from the environment, which is incorporated into the algorithm which is its brain, and it turns into pain, and the baby learns, "I should not do any more of that. That action which I did, putting my right foot on my bum or something while standing on my left foot leads to falling over, leads to pain, so I'm not going to do any more of that."

On the other hand, if it manages to make a step forward, it doesn't really get much reward. And this goes back to the eligibility trace. From one step forward, you don't get a lot of reward. From two steps forward, you don't get a lot of reward. But if you make ten steps forward and you catch the cat and pull its tail and it screams and you have fun from that, then you're getting reward. That's kind of like eligibility trace.

Hadelin: Exactly. That's how it's more powerful and that's how it managed to solve 'Doom.'

Kirill: Yeah, for sure. So there we go, that's a little bit of reward for you, so now you know a bit about Q-values and how they are linked to rewards. We also talked about convolutional neural networks, so now we're moving onto the next one.

Hadelin: Module three!

Kirill: The top, top, top.

Hadelin: So let's recap. In the zero module, we just had Q-learning, so no brain, we had no brain. In the first module we had a brain, but basically with only neurons. In the second module we had a brain with eyes this time, eyes with neurons. And guess what happens with module three?

Kirill: What happens in module three?

Hadelin: Well, in module three not only do we have neurons and eyes, but we add memory. So you see how we're trying to get closer and closer to a human brain. And we add the memory with what we call recurrent neural networks, more precisely LSTM – long short-term memory. It doesn't mean anything, long short-term memory, but that's just the name of the model and basically means that you can learn the temporal properties of the input images, of what's happening in the environment on long-term. It's not like an instant memory, it's a long memory. So that's very powerful.

And I was really happy to find out about this because not only we have this brain with eyes and memory, but also in the last module we implement one of the state-of-the-art models of AI, which is called A3C. It's the most powerful AI model you can buy right now and we even consider it outdates the deep convolutional Q-learning we had before. So it's super, super powerful, but still, the module three had a very challenging problem – 'Breakout,' which we didn't think of in the beginning. Remember, in the promo video we said we're going to try to solve 'Breakout' first because we thought it was the most simple one. But not at all, it was

actually the most difficult one and the reason is we have this tiny ball—

Kirill: Tell the listeners what ‘Breakout’ is. Until I saw it I didn’t remember what it was.

Hadelin: Sure. ‘Breakout’ is an Atari game and you basically have this paddle at the bottom of the screen and you have this ball that you have to catch each time, it has to bounce on the paddle. And you have those bricks at the top of the screen. Your goal is basically to break all the bricks by catching the ball each time because if the ball is not caught by the paddle, you lose a life. You have three lives in total.

Kirill: Everybody’s played this game. When you say ‘Breakout,’ it’s probably hard to picture it, but once Hadelin explained it, you remember you played this game a million times when you were a kid most likely.

Hadelin: Yeah, I’m sure of it. So in this game we have this tiny ball—

Kirill: Yeah. And that’s what makes it hard, right? You can’t see it, it’s really hard to detect.

Hadelin: It’s hard to detect. It’s especially much harder to detect than the big monsters in ‘Doom.’ So that’s why ‘Doom’ is actually more simple. And not only the ball is tiny, but also it’s hard to predict the movement of the ball. Sometimes it goes very fast, so it’s hard for the AI to predict where the ball is going to be and that’s why it was so challenging. We had to try many models to solve it. I was actually worried at some point because I was trying the simple A3C, you know, the common A3C, but that was not enough. We had to add some specific features inside the A3C and those features were actually the

memory, adding in LSTM in the neural network, and also it was to add entropy, that's another thing, and the shared model.

So if you take the course you will see we have this big shared model that will update the weight at the same time for all the agents and that's how the problem is solved. There is something very special about this model, it's the implementation that was made by some developers and one of them is the creator of PyTorch, Adam Paszke, who is in the top 10 of the most influential people in machine learning today.

Kirill: Is he French?

Hadelin: No, I think he's from Poland.

Kirill: Oh, cool.

Hadelin: But thanks for thinking the French are the best in machine learning. I'm flattered.

Kirill: (Laughs) You're just so passionate when you talk about him, so proud. I thought he has to be French.

Hadelin: No, no, I'm just honest with the guy. He's great. He made an awesome implementation of the A3C so we had to take it again; we had to code it again. It was so interesting and so technical and so powerful. That's what we implemented in the last module and basically we have a brain with neurons, eyes and memory. A lot of specifications.

Kirill: But you ran a bit ahead of yourself there when you talked about the three agents. You mentioned the memory and

that's the top of the top, but what about these agents? Why is there more than one all of a sudden?

Hadelin: Okay. So let's take the machine learning field, let's get back to machine learning. Remember, in machine learning we have two models: decision tree and Random Forest. Decision tree is like a tree making some predictions. But then a Random Forest is a team of decision trees. And by having a team of decision trees, of course predictions are more powerful.

Kirill: You average them out.

Hadelin: Yes, you average them out and basically you're getting closer to the best prediction because when you work on a team, of course you play better. And that's the same principle for the A3C. This time, instead of having one agent, instead of having one brain, we have several brains, we have a team of brains. And of course that will work better because we also have this shared function, which is the V-function, a critic that will sort of keep track of which agent is doing best and also getting some information of the environment, which state some agents are getting stuck into. So it's very useful information for the agents. For example, if one agent gets stuck in a state, this information will be contained in the critic V-function so that the other agents don't get stuck in the state.

Kirill: And then that one gets unstuck.

Hadelin: Yes.

Kirill: Thanks to the other guys.

- Hadelin: Yes. So, basically, we have a team now. We have several brains so, of course, you make better play.
- Kirill: I was really surprised that only Google just recently thought of that. How come nobody else thought of that before?
- Hadelin: Yeah, I agree. This looks natural. Especially since Random Forest existed for a long time, so why didn't we think of making some kind of Random Forest AI? I think I know the answer.
- Kirill: What's that?
- Hadelin: I think it's related to the fact that training an AI is really, really compute-intensive, it takes a lot of time, it takes a lot of time to train because you take a lot of time to compute a gradient. But then, remember, in the AI course we implement the model with a very powerful tool, which is PyTorch, because it handles what we call the dynamic graph. Basically it has a dynamic graph which contains some variables I'll attach with a gradient. Basically that allows us to compute some very compute-intensive calculus of the gradient in a very short time. So that's very powerful.
- You can backpropagate the signal – backpropagating the signal meaning that we compute a gradient to obey the weight to improve the predictions and reduce the losses – and basically those gradients are now computed superfast, so you get some superfast training, much faster than what we had before and thanks to that, we can even train a couple of agents much faster, which we couldn't do before.
- Kirill: That's cool. And you mentioned dynamic graph. Tell us a bit more, what's a dynamic graph?

- Hadelin: Okay. Dynamic graph is like a graph of some what we call torch variables, and each of these variables is like a tensor. Tensor is an array of a single type, an advanced array.
- Kirill: Like a matrix?
- Hadelin: Yes. Each of these tensors is attached to a gradient. And all these variables that contain a tensor and a gradient are in a dynamic graph. It's structured in such a way that when we have to compute the gradient of composition functions, thanks to the graph, the gradient of these composition functions will be computed very fast, very quickly. Otherwise you would have some huge computations if you have to compute a gradient of some composition functions, because you have several layers and therefore you have one layer that is a function of a previous layer, which is a function of the previous layer.
- So computing the gradient of all this, the composition function, would lead to huge calculations. But thanks to this dynamic graph and the way it is structured, the gradient is computed very quickly.
- Kirill: He's drawing a graph with his fingers. (Laughs) You can't see it, but—
- Hadelin: Yeah. I do it for myself. (Laughs)
- Kirill: Gotcha. Okay, so PyTorch has that, but TensorFlow doesn't have that?
- Hadelin: Exactly. TensorFlow doesn't have that, but I think they're working on it.



- Kirill: And that's why Yann LeCun says PyTorch rocks. (Laughs) Imagine Yann LeCun listening to this.
- Hadelin: Okay, I have to say what he said. He made a post on Facebook, I think, and he was comparing TensorFlow and PyTorch. And he said, with a very simple answer, "TensorFlow cannot handle dynamic graphs," and that said everything.
- Kirill: That's it. And Yann LeCun is one of the forefront developers or Meta developers, pioneering researchers in AI and he works in Facebook and Facebook develops PyTorch, right?
- Hadelin: That's right, with NVidia and a couple of other institutions.
- Kirill: So you've got two competing schools of thought. You've got Google, with their TensorFlow, which is very hyped and which is very popularized and they have lots of videos and lots of people have heard about them. And also, its advantage is that it has Keras, which makes it very easy to use, especially for beginners. That's why we used TensorFlow in our deep learning course.
- Hadelin: Yeah. But you could definitely not do some AI with Keras.
- Kirill: Yeah, but it's easy to start, it's easy to learn AI in Keras.
- Hadelin: Yeah.
- Kirill: And then PyTorch is just more advanced in what it can do.
- Hadelin: Yes. But I'm sure AI Google is working on a very powerful implementation of these dynamic graphs for TensorFlow. Otherwise they wouldn't be able to do such amazing things like DeepMind and Magenta. You know, Magenta is working on these projects like making some music with

[indecipherable 37:05]. They're making great music, like a couple of notes, but that's still great music with TensorFlow. They want to keep TensorFlow, they're not using PyTorch. So basically they're implementing some deep learning models, some very powerful deep learning models for that.

Kirill: Okay. Well, I think that sums up the overview of the course and what we did there. Let's talk a bit more about careers. Hopefully we'll wake some of our listeners up that dozed off 20 minutes ago. (Laughs) Tell us about AI. Why would somebody listening to this podcast, who is interested in a career in data science, or in changing to a career in data science – you know, that's all they wanted to hear about switching this episode on, about data science – why should they consider artificial intelligence as a potential future next step?

Hadelin: Simply because AI is becoming one of the major branches of data science. We already heard that, that machine learning is the brain centre of data science, but AI is taking a higher and higher place in this data science world. And now there is even an AI job – AI engineer. Before, we didn't have that. Like one or two years ago, we didn't have this. And that's because, as we said at the beginning of this podcast, AI can solve amazing real world challenges. So of course that's becoming a super important job, because you can solve these amazing problems with these amazing solutions. So, yes, I really encourage going deep into AI, because you will definitely make the world a better place with this as long as it doesn't go too far. And you will definitely solve some exciting problems and make the world a better place.

Kirill: Fantastic. So, AI is coming. And do you think it will always be this hard? No, I wouldn't say hard. You know, what we are doing in this course, the forefront of AI, it's very complex. Are listeners going to have to delve into something as complex as we discussed, or is AI going to become simpler over time for people to use?

Hadelin: I'm sure AI will become simpler over time. It's exactly like what happened for machine learning. At the beginning, we only had these from scratch implementations of machine learning. That was very overwhelming. You had to understand all the lines of code and you had to understand the heart of the models, how it works, all the mathematics behind it.

But then there were the libraries, like scikit-learn or Keras. So that today everybody can implement and use some machine learning models very quickly in a couple of lines of code. For AI, we don't have that right now, but I'm sure we'll have it very soon with some super performing libraries of AI where you only need to implement a few lines of code to just understand how you need to structure your input and what output you need to return. Because basically, when you think about this, if we don't come through the heart of the model, we only need to understand what input we need to input and what are the rewards and what are the actions. That would be the only input and then you can solve the problem. So that's what I think we'll eventually have with AI so that everybody can use it.

Kirill: That would be fantastic, if everybody could use it for good. And I got a quote here from Andrew Ng, basically he was a

lead researcher for Baidu and he used to work at Google, before that with Geoffrey Hinton, I think he was one of Geoffrey Hinton's students or something like that, and then went to Baidu and now he's doing his own thing, and he's also the founder of Coursera.

Hadelin: Yes, and he's definitely in the top 3 or top 5 most influential people in machine learning. Everybody knows his machine learning course.

Kirill: Yeah. He said recently, not so long ago, that AI is the new electricity.

Hadelin: Yes, I agree.

Kirill: You agree? What do you think of that?

Hadelin: It's the next electricity or we could even say the next industrial revolution. Because it's the new thing that makes the world a new revolution in solving some problems that were impossible to solve a couple of years ago. So, yeah, it's like the new big step of the world, I think. I totally agree with him. It's this new technology, this new industrial revolution.

Kirill: Yeah, it's going to really revamp how everything works. And it's a good analogy, right? Before electricity, everybody thought that they had figured it out, businesses were working, trade was happening, countries were flourishing, and then electricity came along and like 10-20 years later, everybody was using electricity because who wasn't using electricity was way behind.

Hadelin: And now, in 10 years everybody will be using AI, either at home or at work.

- Kirill: Yeah, for sure. And then on the flipside, you mentioned in this podcast already—
- Hadelin: It shouldn't go too far.
- Kirill: Yeah, it shouldn't go too far. Tell us more about that.
- Hadelin: When we talk about AI, there's weak AI and strong AI. That's a very important distinction to understand. Weak AI is just an AI that doesn't have a conscience, and strong AI is an AI that has a conscience. Imagine the world becomes populated with strong AIs that all have a conscience. Well, that could end up into a warzone, because imagine that AI gets conscious of itself and makes some kind of a plan to do something on the humanity. That could totally be possible. If you guys have seen the movie "Ex Machina," that's a great example of a strong AI, because it has a conscience and basically does something not good for humans at the end of the movie. I'm not going to ruin it, but—
- Kirill: Yeah, let's not ruin the movie. And if you have not seen that movie yet, you have to see it because we've already recommended it a couple of times.
- Hadelin: That's a great example of a strong AI and a great example of the threat of AI.
- Kirill: Okay. I'm just trying on my laptop to bring up Ray Kurzweil predictions. So let's have a look at that – futurism.com, I think. There we go. If you're listening to this, we're looking at futurism.com and there I just looked up "The Dawn of the Singularity," it's predictions by Ray Kurzweil. Do you know Ray Kurzweil?
- Hadelin: Yes.



- Kirill: So this is a futurist who's been making predictions since the 80s and his prediction rate, astonishingly, is 83% correct.
- Hadelin: That's amazing.
- Kirill: He predicted things like the iPad back in the 80s or whenever. So, here is a really cool infographic. What it's saying is how AI is going to come. I'm not going to look through all of them, but one of them, which is really scary, is "By 2029, artificial intelligence claims to be conscious and openly petition for recognition of this fact."
- Hadelin: There we go, we have strong AI by 2029.
- Kirill: Crazy, right? Petitioning for themselves.
- Hadelin: Yes.
- Kirill: Then, "By 2040, non-biological intelligence is now billions of times more capable than biological intelligence." What's next? "2099 – organic human beings are a small minority of the intelligent life forms on Earth." If you're listening to this, by the time your kids are grown up—
- Hadelin: Protect them.
- Kirill: Yeah. (Laughs) They should be on Mars or somewhere. What do you think of all this? Do you think this is going to happen?
- Hadelin: Well, what I'm thinking right now is that I'm really happy we didn't go up to module four, by adding a conscience to our AI, because it would've destroyed the course.
- Kirill: (Laughs) Oh, yeah, for sure.



- Hadelin: This is scary. I'm pretty scared about this, but I'm sure we'll do the necessary and it will go up to the government level or political level to set up control of the AI. Because otherwise, if his predictions are right, as his previous predictions were pretty correct, we must do something about it. Otherwise we'll have to go to Mars, as you just said.
- Kirill: Yeah, interesting. But nevertheless, this doesn't defeat the point that you should study AI. Whether you're going to study it to build it or to protect from it or to control it, to govern it, it's going to be the central question for the whole human race within 5, 10, 15 years from now. You know, get on the train early, find out for yourself how you can contribute to making the world a better place through artificial intelligence rather than letting it just cause havoc and do its own thing.
- Hadelin: Absolutely, yes.
- Kirill: Okay. So, I think we'll finish up on that note.
- Hadelin: Yes. It's actually a good note to finish. (Laughs)
- Kirill: Yeah. The world is safe and everything will be good.
- Hadelin: Yes. Thanks to you, guys.
- Kirill: Yeah. And watch "Ex Machina."
- Hadelin: Yeah, watch "Ex Machina" for sure, great movie.
- Kirill: Okay. Thanks a lot, Hadelin, for coming on.
- Hadelin: Thanks for having me. It was as always a great time.



Kirill: Yeah. This is episode number 3 and you have approximately 30,000 downloads already on your previous two total. So, yeah, I'm looking forward to number 4.

Hadelin: Yes. What's it going to be about?

Kirill: We will find out. Okay, thanks a lot, man.

Hadelin: Thanks.