**Craig:** Hi, I'm Craig Smith, and this is Eye on AI. This week, I talk to Oriol Vinalys, who leads DeepMind’s deep learning team, about AlphaCode, his group’s code-writing language model, and DeepMind’s winding road toward artificial general intelligence.

Before we begin, I want to mention our sponsor, ClearML, a collaborative open source MLOps solution. They're sponsoring the podcast. They offer an end-to-end system for building and deploying machine-learning models. Check them out at clear.ml. Tell them Eye on AI sent you.

**Craig:** Meanwhile, I hope you find the conversation with Oriol as interesting as I did.

**Craig:** Okay. So, Oriol I have to ask you a couple of things: one, what's the significance of the big black bear behind you, and two, I'm curious about the meaning of your first name.

**Oriol:** Right? The bear behind me actually traveled all the way back from California, where I did my PhD at UC Berkeley, which has as a mascot a bear.

**Oriol:** And then it's a good question about the Oriol name. It's a Catalan name. I'm from Barcelona originally, and I believe it doesn't have any translation to Spanish. It's a compound name, which is Joseph-Oriol.

**Oriol:** But again, it's a very local name from Barcelona.

**Craig:** Okay. And I know you've given your background many times before, but that's generally how I like to start. So, you are a Spanish by birth.

**Oriol:** Yes.

**Craig:** Can you talk a little bit about your education and then how you got to DeepMind?

**Oriol:** Yeah, certainly. So, I did my undergraduate degree in Barcelona where I'm from, then moved to the US where I lived for probably 10 years. I did my masters at UC San Diego, so, in California, and then moved to do my PhD in a bit more northern California in Berkeley, where I did the PhD starting in 2009.

**Oriol:** It was a very interesting time because machine learning was definitely not as popular, deep learning certainly wasn't very popular. So, it was a unique experience to be in a university environment but discovering a lot of what was going on in the field. And there were only a very few people that I was able to hang out with at the university.

**Oriol:** But luckily, I did a few internships also doing my PhD, many of which were at Microsoft Research in Redmond in Washington state. And that is where I met also a lot of students from Geoff Hinton's lab, where it sparked my interest in deep learning, which I then exported to Berkeley.

**Oriol:** And then in 2013, I finished my PhD. I did my very last internship at Google in Google Brain. It was a fun time because Geoff Hinton was an intern himself as well, actually, at the time. We had both internships at the same time. And then after graduating, I joined Google Brain, which was very small compared to what it is now. I worked there for a few years and then moved to DeepMind in 2016, back in London, back in Europe where I'm from.

**Craig:** Yeah. Yeah. You were an intern at Microsoft the same time as Ilya Sutskever, is that right?

**Oriol:** Yes. Yes, many of my friends I met through Microsoft Research internships, including Ilya Sutskever, Alex Krzyzewski, George Dahl, and many of these folks and then you would obviously hang out in conferences in the very small workshops that we held in deep learning at the time.

**Oriol:** And yeah, many friendships started through internships and of course at school as well.

**Craig:** Yeah, it's remarkable. I just had Andrew Ng on the podcast and he talked about those days when Geoff Hinton was his intern. Yeah. My conversation is a little less technical than many of those that you've had.

**Craig:** I'd like to start with AlphaCode and I'm particularly interested in AlphaCode because I'm not a coder. And I've talked to a lot of people about no code and low code platforms that are emerging.

**Craig:** I'm wondering where that's going. And I hear a lot of skepticism about, applications of things like AlphaCode because of the limitations and the actual complexity of useful code. And then I'd also like to hear a little bit about DeepMind’s work on large language models, which are the rage right now.

**Craig:** And whether there's a relationship between RETRO, I believe is DeepMind's large language model, and AlphaCode. And then where does the research go from there? So, if you can just talk, I won't interrupt very much, about how AlphaCode started. What is the purpose of that research?

**Craig:** What are the likely applications, if any of the AlphaCode model and how it relates to large language models and then where does it go from there?

**Oriol:** Great. Yeah, I'm happy to talk you through a bit of a few of these projects that you might be learning from our blog and the news that comes up from the lab.

**Oriol:** So AlphaCode in particular, If I have to trace its inception, it starts actually doing another project that we ran a couple of years ago called AlphaStar, where we were building an agent to play a video game, StarCraft, that I actually am very passionate about. And the fact of the matter is we've been using a lot of these sort of benchmarks that humans found interesting through many years, as a sort of benchmarks themselves as well for our agents and the technology we developed.

**Oriol:** So, one of the sorts of trajectories you can trace back if you look at DeepMind's history and certainly many of the successes in the field is to see which benchmarks exist and what's the performance improvements over those. Obviously, DeepMind started perhaps most notoriously with Atari playing Atari games.

**Oriol:** And then StarCraft was perhaps the last game, which was a step up in complexity. And Go was maybe an intermediate between those two. So AlphaGo being another very famous of course project that the world learned about a few years ago. So AlphaCode, in a way from a techniques perspective actually borrowed a lot of past research and that's actually one, maybe one meta point I would like to make very early that I'm leading the deep learning team.

**Oriol:** So deep learning, I find it quite useful to see it as a toolbox, right? So, the toolbox is ever expanding. There's new research, new papers come out, people share code, share insights, and you're adding this tool now to your toolbox. And then anytime there's a hard challenge out there, what's happened now, the revolution, so to speak, in deep learning is that you can go and seek in the toolbox and find patterns, find past problems you've solved, things that were useful.

**Oriol:** Take them out of this toolbox and then reapply them to a new domain. So AlphaCode, in fact, borrows a lot of the things we learned doing AlphaStar because it uses, in fact, it connects also with language models because it first learns what code looks like. Coding is no less than a sequence of words that happen to conform a program.

**Oriol:** So that sort of same principle applies to coding, and it applies to the game of StarCraft as well because playing a game is no less than issuing a sequence of instructions that look a bit like code, actually, move this piece up on the board here, et cetera, et cetera. At the time we were doing AlphaStar, coding seemed like a very interesting challenge. And when you look at a domain like coding or programming in general, one of the things you want to do, is there any existing benchmark in that domain that will serve the purpose of, can we do this with the current tools we have in the toolbox?

**Oriol:** If yes. Then it's useful to know. If not, maybe we'll invent new tools to be added for further projects that go beyond what we do towards AGI. So, with that in mind, looking at coding in particular, there are a few things you could imagine doing with an AI to enhance software engineering, but we went perhaps the more traditional route we had been doing with games, like Atari, Go and StarCraft, which is look at the very serious competition that has not been designed for AIs to compete in, take that platform, hopefully a popular one where humans really have expanded and try to exhaust all the skills that humanity that programmers have developed and then participate in it to see where we are in this benchmark.

**Oriol:** And that's where we thought about a coding contest or coding competition as the platform to do such a challenge, which is fairly unique and fairly different than perhaps other work that regards more about completion of code or trying to assist coding in some sort of, oh, I don't know about this language or this API, and then you have a smart way to access that.

**Oriol:** Instead of doing that, what we did is code competitions, which in a nutshell can be described as, there is a problem statement. It's almost like an exam that you would do at school. So, there's some sort of story that tells you about, explains what an algorithm should be doing in words, in plain words.

**Oriol:** So, it's natural language at the input. And then humans usually take this description. This description contains a few examples, right? Oh, like for example, if the input to this program you're about to write is this, the output should be that. And then from this description humans basically put a few hours trial and error, right?

**Oriol:** They code, they code some Python, C++, or whatever popular language they choose to. They code the solution. Then they submit it to a web server that then has some secret tests that would evaluate, hey, is this program fast enough, importantly, and also, correct. So, we thought that's great.

**Oriol:** There's actually tens of thousands of participants in weekly competitions. People that take these competitions very seriously, by extremely clever coders. So, we took these as the sort of benchmark and the result was that we achieved median performance, meaning average human that competes in this. So that's a bit what happened in AlphaCode in terms of the result and a bit of background.

**Craig:** Yeah, AlphaCode uses the transformer algorithm as the core. Does it draw at all on RETRO, on a large language model?

**Oriol:** Yeah. So, this is a good question.

**Oriol:** So first of all, actually AlphaStar, AlphaFold, AlphaCode and all the large language model work you're seeing from DeepMind and obviously from many other labs, use transformers, which is one of these tools that we added in the toolbox in 2017. It's a very powerful neural network that allows you to model sequences.

**Oriol:** Basically, you input the sequence and the transformer does a transformation - it's actually a good name - on top of it to then predict either, whether the sequence is, I don't know, a positive or a negative sentence in natural language processing, or what's the 3D coordinate of a protein or an atom in the protein or, in the coding case or language case in general, it predicts what's the next level that proceeds the code. So, it indeed uses transformers. Retro is actually a particular language model that uses an extra, an additional idea. So, it expands a transformer with a large memory bank. Basically, you're trying to predict the next word. So, you're seeing a few letters, maybe of a program you're trying to predict.

**Oriol:** What's the next letter that you should write to make this program, correct? But then critically it indexes a large dataset of lines, like trillions of documents to retrieve relevant information. And this is another tool in the toolbox, in fact, the retrieval process that enables models to become as good as bigger models, but at much less parameter count.

**Oriol:** In the coding example, we actually tried some of these ideas because it's very natural to think, hey, when you're coding something, you actually retrieve, you search for similar algorithms or try to inform yourself about what the code might look like. And in fact, in these competitions, that's very commonly done.

**Oriol:** But, in actuality, we found this to not be so useful for AlphaCode in particular. But then the Retro paper showed its capabilities on just regular language models. And in our large language models research, this is an active area to expand this toolbox, to make it more useful, to essentially supercharge transformers to access this memory bank a bit like us.

**Oriol:** We are supercharged by Google search. When we're looking for information, we don't remember everything, right? So, this idea behind RETRO is actually more a new tool that's still a work in progress. We published, of course, a paper. There's some activity in research. But in AlphaCode, in particular, it didn't help.

**Oriol:** So, we just use the plain transformer.

**Craig:** I see. So AlphaCode itself is a large language model.

**Oriol:** So how did we create AlphaCode? The step and maybe that's where most of the inspiration came from, was AlphaStar. We thought about the same process as we created for these agents that played the game.

**Oriol:** The first step is imitating humans, right? So large language models, take a large data set of language sequences, and try to imitate the statistics of what will come next after each letter or word, train a big model, and then that's what you're seeing these days in large language models. The way that we did AlphaStar was actually the same.

**Oriol:** We had sequences of actions, these actions that I mentioned, move this unit here, move these unit there. We trained a large transformer, an LSTM model - it’s a different recurrent neural network that is also very popular – to imitate human actions, but human actions, human words that are in some data set. They're essentially the same or mapped to one another.

**Oriol:** So, in AlphaCode, the first step, like in AlphaStar, was to pre-train an agent that was able to imitate human moves. In this case, of course, this is a very language centric task. We took GitHub, which is a massive repository of human code, and then we said, hey, let's start from pre-training on this large amount of data, so that we have a first transformer model that is able to produce code that looks reasonable.

**Oriol:** And this is actually a key ingredient of maybe all of these tools that could potentially help software engineers to auto-complete but be much more than auto-complete. So that's the first step of AlphaCode. But now the crucial step, there's two extra sort of algorithmic things you do. Once you have this pre-trained amount of knowledge, this is mostly knowing how to continue programs, but it does not know about this particular format, which is, I'll give you a long language description of the algorithm you should be doing.

**Oriol:** And then please just generate code that is correct. Period. There's no ambiguity in the metrics, which is very important. And probably when you talked to Pushmeet actually about AlphaFold and many other examples, having a good metric and a good benchmark is critical to assess advancements. So, similar to there, we collected all the data that exists in these platforms, in these websites that people have been competing in for many years.

**Oriol:** So, we'd actually talked to the creator of one of these platforms. And this person was very excited actually to see what an AI could do. They really thought it would be impossible to do anything reasonable, which is always a good sign when you work on these projects. And we collected, let's say 15,000 examples. It's not a large amount of data in machine learning, right?

**Oriol:** 15,000 is actually smaller than MNIST, right? This very popular data set that sparked some of the early work on convolutional neural networks. So, we created this specialized data set of input-output examples, 15,000 of them, which contained the language description. And then the code that could be tens or hundreds of lines of code that implement a solution to that particular problem from past competitions, right? Then we took these massive language model that we pre-train, we fine tune it, which is actually a very common tool again, in the toolbox. If you will, this is a common practice, you will do with neural nets. We fine tune it, to be good at this particular setting, so to speak.

**Oriol:** Although honestly, I was skeptical and surprised to see that with such small amount of data, we were able to indeed steer this model, this transformer model to now become not very good at, but it started to be reasonable at solving some of the easiest problems. And once you see something promising, that is when you start iterating, you start developing new sort of components, new tools, new algorithmic insights. And basically, we started from solving maybe 2% of the problem, the test set we had collected from this website through many innovations, to the 34% or so, a solve rate that made us believe this might be at human level. At which point we took that neural network and actually placed it in the web server competition.

**Oriol:** After a competition was held, we entered, and we then measured ourselves. So, the last, very step though, that was needed besides some innovation, some techniques that are perhaps very deep in the toolbox, not so generic yet, but we might be using them in further projects, of course, was the idea that came from actually AlphaGo, which is the idea of search. You have this natural description, natural language description. You can generate one program, but you can generate 10 programs or 50 or a hundred or a thousand. And then the question is to find a program that looks promising enough that you will bother to go to the web server and submit it. We don't, we didn't want to submit millions of programs to the web server.

**Oriol:** We actually thought 10 attempts is maybe the max that we will attempt to at most, since that felt reasonable and talking to the creator of the website also thought that seems reasonable as well. So, we created many different program possibilities by essentially sampling the language model almost let's say a million times.

**Oriol:** And then we developed something very similar to what the value functions do in, for instance, AlphaGo, which is what is the programs we should be picking out of these millions of programs that we should submit. So, then we developed a few techniques to filter down all these programs, that some of them didn't compile, but some of them did compile and some of them pass the test we knew about, but we didn't know if they were correct.

**Oriol:** So, we filtered them down to 10 at most. And then we submitted to the server and the results we got, I think impressed really like many people that probably have quite a lot of knowledge about machine learning and then this programming contest. But at the same time, I think what's cool about this is, it's human level, but it's not at the top level. And there is still a long gap and a long way to go. In a way, this is almost the beginning where we have a very good benchmark. We think this benchmark tests reasoning, problem-solving abilities, et cetera. And the transformer is a great tool, but maybe we need better tools or enhancing tools to then climb this ladder or leaderboard.

**Oriol:** That is one of the many that exist of course, in the machine learning community.

**Craig:** Yeah. Yeah. I have a couple of questions. The pre-training on GitHub that's an unsupervised task, right?

**Oriol:** Yes. There is a potential debate what unsupervised means. So, I want to maybe just maybe explain. So, typically it is unsupervised in the sense that this is a dataset that exists. And you just take, let's say a large collection of programs that were written for whatever purposes. And you just learn a system that will predict what is the next word or token or character. And in that sense, it is unsupervised because we didn't label this data set, we didn't do any extra work on it. But in a way it might be supervised because this is, someone wrote that code and it's not random programs, right? These programs were not randomly generated. But I would say, yeah, this is an unsupervised pre-training step that most language models undertake these days.

**Craig:** But the second step of then training on the competition set that's in effect a labeled data set because you have scores for how well the different programs performed and that sort of thing.

**Oriol:** Yeah. So, the second step, maybe the best sort of parallel to trace here is with that of machine translation.

**Oriol:** So, imagine we know we can get a lot of English text on the internet. We can get a lot of Spanish texts as well. So that would be what we did in pre-training. We got a lot of Python and C++ texts, and we just trained the model on that. But then we needed to know how to translate, and it is a very good way to put it, actually. How to translate this natural language description high-level description, translate that to an algorithmic solution in Python or C++, the two languages we focused on for no reason then other than that, they are popular.

**Oriol:** So, the second data set needs to be paired, right? This needs to have on the one hand, the description, and the other hand, you have the solution for that description that someone brought with other intents. So maybe that of course constitutes what we would call supervised sequence-to-sequence learning, which is actually a project that I worked on with folks at Google Brain a long time ago for machine translation initially, but which has been applied to many other sorts of domains, right?

**Oriol:** You can translate English to Spanish. You can translate an image to text. You can translate the text to an image. You can translate a specification of a problem to the code that solves it, right. And that's maybe the easiest way to see it parallel to that.

**Craig:** Yeah. So, the purpose of AlphaStar isn't to create an agent that can beat everybody at StarCraft. It's a research effort to advance the science. And is that the same case with AlphaCode or in AlphaCode's case is the end goal more specific that you really want to create an agent that can code from natural language?

**Oriol:** So that's a great question. I think there's always the tension. When I think, this is actually a very nice portrayal deep learning. We care about the toolbox, right? So, if you think of, oh, I'm a deep learning researcher, I'm leading the deep learning team. What is our goal? Our goal is to expand with more powerful tools so that any new problem, any new domain, any new fields, right? Like biology, like we went pretty far from machine learning and there is more to come that we can have the powerful tools that will be needed for very specific domains that have very hard problems to be solved. So, from a deep learning and perhaps zooming out from a solving AGI standpoint, which is ultimately what we're after at DeepMind, absolutely the view of, these are benchmarks.

**Oriol:** We get a lot of good information from them. We can generate new tools to advance, to go to the next steps. That view is perfectly fine. And in fact, many of these projects, I would say, have these characteristics as benchmarks that some of them might be solved and we might revisit or not. And in the coding case, now going to your second point, since there seems to be something fundamentally difficult about this, it's not over, we now are at the beginning in a way of discovering how this wonderful sort of benchmark that humans created that challenges them intellectually, et cetera.

**Oriol:** How will these push the boundaries of the techniques, but always the eye in deep learning is these techniques will be applied to other fields and domains. And so, this connection with all the, you can almost connect, if you look at all the many projects and work that goes at DeepMind, of course from the inside it is perhaps easier, but it is quite natural to start seeing how these projects connect to one another.

**Oriol:** They connect with the techniques. They also connect wonderfully with the teams that are behind these projects. There's a lot to be said about, of course, the researchers behind it. You're seeing maybe a few of them, you get to talk to a few of them, but it is also interesting to understand that these teams that go behind one of these hard goals, then will proceed to either keep iterating over in this case, the coding one. Maybe some people will move on to other projects and so on.

**Oriol:** And this is actually how I think the field has advanced in a way that perhaps it's a bit invisible to those who might not be as in as we are.

**Craig:** Yeah. Yeah, of course. Every time you work on a project, you carry the knowledge to your next project. A lot of that knowledge is not written down in papers.

**Oriol:** Absolutely, yeah.

**Craig:** But again, with AlphaCode, because I'm not a coder and I'm fascinated by no code solutions and low code solutions, do you think there's a possibility that AlphaCode could progress to the point where it could write more complex programs?

**Oriol:** Yeah, absolutely. So now going a bit deeper into coding as a very interesting, very meta domain, right?

**Oriol:** You have to, we're coding to generate these tools. These tools in the box are actually pieces of code and so on. So, it's actually a very dear domain to us researchers in machine learning. So, there is quite a lot of excitement of course, about beyond the benchmark for towards AGI.

**Oriol:** Could this be just something that people find useful? Absolutely. But we are talking to, there's obviously a lot, also a lot of teams across Google that are working on similar projects. And you might have seen also other research. That's the beautiful part of the community, right? So, I think we are exploring right the power, how these techniques, maybe not the actual agent, not AlphaCode, if we call AlphaCode the set of weights in the neural network, but how can we now take some of the techniques and apply them in a setting that maybe it's more about making software engineers more productive, et cetera.

**Oriol:** And I think from what I can see, there's a very real possibility of a transformation. It's probably going to be slower than some people think, but that's fine, on how we do and produce code. Will the limit of as maybe writing down the problem like these competitions have that you want to generate some sort of program that you just write it in natural language, no coding required?

**Oriol:** And then the solution comes out at the other end. Perhaps, AlphaCode clearly does that with something that felt quite difficult, or it actually was impossible just a couple of years ago. So, could these go beyond the setting, which is code competitions? This is a narrow domain in a way. It's just specific kinds of algorithms.

**Oriol:** They're very hard, but they're just short programs. Can we go from here to a more end to end approach? I believe so. The question is, are we going to focus on it? And I think for us, the main mission is always AGI. But of course we are lucky to have many partners and many, that's how the field works.

**Oriol:** Really, actually, maybe, you could see this advance clearly will unlock this. And the publication process is actually enabling others to also explore possibilities of this kind of domain. So as a researcher and me personally, you always are balancing how deep are you going to go into these domain vs well, I am a deep learning researcher. The tools are there, go ahead. Use them. And that's also wonderful when that happens, which obviously has happened as well, many times in our field.

**Craig:** Yup. And on this toolbox, you've been out talking about quite a bit. Does DeepMind have a platform that researchers or practitioners or developers can use?

**Oriol:** Yeah. So, the toolbox, as I've been referring to is clearly a sort of very virtual toolbox of ideas, right? Like the transformer model, knowledge distillation, LSTM, convolutional networks. So, it's a toolbox of ideas. And then from the idea to the actual implementation there is indeed a lot of tools, actual tools that you would call them in software engineering to then develop and almost like it's like a puzzle to work on a new application, to crack a new benchmark, to go to like I know physics and try to make a change there. There's a lot of very cool things happening where people in different fields because the tools are not only the toolbox of ideas, but the tools actually to run them are fairly developed.

**Oriol:** They can go ahead and try them. And it's amazing to see how other fields are embracing this approach and basically getting it. So academic labs like ours in industry and in universities, they're part of some ecosystem of like software, which actually, if I think about why.

**Oriol:** What has happened? Why is this so popular? Why in 2009, deep learning wasn't as popular as now? Software, and the actual open sourcing of platforms. That's been a key in the community. So indeed, there's obviously a few choices depending on what you want to do as a practitioner and as a researcher, but you can certainly map instantiations or, realizations of the virtual tools to then, any of the platforms that are very popular.

**Oriol:** Now, there is obviously PyTorch, TensorFlow, JAX and the ones to come. As we say in the field, if in five years we are using the same tools as in software tools, something is wrong because the field is advancing so fast that when you create these open-source libraries that are used, the research takes over and there's something unexpected that you did not account for. And sometimes you need to start from scratch and have a new framework and a new platform.

**Craig:** Yeah. So, you're not personally, you're not drilling down into the possibilities of AlphaCode as a particular domain. Where is your research going next or are you still working on AlphaCode.

**Oriol:** Yeah, that's a good question. Obviously, we are still working on AlphaCode. We have in the deep learning team and obviously DeepMind in general, we have quite a few projects. So AlphaCode, usually when you see a publication, sometimes it's a milestone that we hit, or a checkpoint.

**Oriol:** But, we keep working on it. As part of the many projects that I oversee personally as the leader of the deep learning team. I've been, from a personal standpoint, I've been very actually if I look at, when I joined DeepMind in 2016, one of the things that I immediately catch upon was this idea, at the time maybe the popular name was meta-learning, which is the idea that you can learn to learn so to speak, right? Traditionally, in supervised learning, you take some data set, you train a neural network, and that neural network does something useful like translating, transcribing speech, or writing code. But this idea of meta learning that you can teach a model, something new was happening at the time through a very simplistic view of this approach, which, at the time we obviously imagined that this is still a very valid benchmark, right?

**Oriol:** That we were working on still to advance the toolbox. But ImageNet was, oh, there's a closed data set, thousands of categories of objects. And there you go you need to be more accurate in the next generation of neural networks. But what happened at the time is we took ImageNet, and we invented a new task, which was, when I test your model, I'm not going to ask you from an image, can you classify it onto one of the thousand objects. Instead, I'm going to give you a few input-output examples that you've never seen, right? New categories of objects. And then you need to create a system that is able to observe this information very quickly with very little data, right?

**Oriol:** One example per category, perhaps, and do well at that. And that was happening at the time through image classification. We call this few shot learning and then feed forward, it turns out that these large language models are excellent at future learning and at learning sort of things that you can induce at test time, so to speak, right?

**Oriol:** So, these models are trained to be good at predicting these unsupervised data of all the texts on the internet or all the code on GitHub. But then there is a lot of research to be done in my opinion, but a lot of exciting opportunities for teaching this model something new, but not learning, fine tuning is one approach that is not the most elegant perhaps.

**Oriol:** So, I think what I'm now focusing more on is these capabilities of meta-learning as we called them back in the day. But now you could call this few-shot learning or this ability that you can literally have a model that you can instruct, so to speak, just say, hey, here is an image of these objects.

**Oriol:** Here is another image of these other objects. Can you now play this game with me? And it's a transformation of how we are testing models in machine learning that has been ongoing for a long time. Of course, like this the field always has newer benchmarks and words for describing things.

**Oriol:** But right now, AlphaCode is perhaps an instantiation of that in very particular ways, but I'm very excited about some of the recent work also that the team has been doing on this particular aspect where it's not only about language, but about vision, which links very nicely back to this idea of, hey, can we classify these two objects from one another?

**Oriol:** We could at the time, but it was a very cumbersome way to do so. Now, we can literally not only classify but teach almost any computer vision tasks to models by basically showing, hey, these emails, tell me something about it. You can ask it more and so on. And in part of our large language models research, the latest actually release which was just released a few days ago was this model Flamingo, which is a visual language model, that knows a lot about language, but actually is grounded in images.

**Oriol:** And I think that sort of link that comes from when I joined DeepMind to say, wow, that's amazing. I wanted to learn about reinforcement learning. So naturally I went into projects like AlphaStar and AlphaCode to now just say wait a second this is something super exciting.

**Oriol:** So that's one of the kinds of focuses that I'm switching my sort of research attention towards for sure. Yeah.

**Craig:** Can you talk a little bit; you mentioned a few times already AGI and DeepMind is famous for having that as an ultimate goal.

**Craig:** Is there a roadmap? And you mentioned guideposts or milestones. Do you guys sit down and have an overall roadmap that may get dimmer the further you get in the future of how all of these things are contributing and how you expect them to converge some day?

**Oriol:** Yeah, absolutely.

**Oriol:** The roadmap toward a sort of AI that would be general that would be able to learn with you the way I actually just described it. That's very much in discussions, of course, at DeepMind and in the community, that is, how to get there. What are the right benchmarks? And all that entails.

**Oriol:** I think this is basically probably the main topic of discussion in the main machine learning conferences, to some extent, and certainly at DeepMind since that's our mission. We were doing these projects with a lot of, with the mindset that's where we're going. So that's why perhaps it was useful to depict this level of difficulty of perhaps single domains, right?

**Oriol:** Like from Atari, to chess to Go, to StarCraft, right? That's a thread that doesn't happen randomly. This is something that you can see where we're going. In fact, I did work on StarCraft back in my Berkeley days which is a fun fact. And at the time I remember the mindset we were not ready to do StarCraft.

**Oriol:** We were working on Atari in this end-to-end fashion, like machine learning fashion. So clearly you can see, especially externally, which is obviously the things that you see more visibly, you can certainly depict these little roads that are increasing in difficulty and impressiveness towards AGI.

**Oriol:** And then of course, where to go next is the question. I remember someone, when I joined my PhD, they said 90% of your PhD is the question you're asking. So, it is a very hard question to find what is the next thing on the road that will provide progress and signal towards that.

**Oriol:** So indeed, we talk a lot about it, but what you're seeing is hopefully, it's something that will make sense when we look back. And it's actually quite a useful exercise to actually, for me to be talking to you because sometimes you on the fly, you might not realize it as easily until you connect the dots and say, of course we were seeking this, for instance, learning ability of the models.

**Oriol:** And now maybe the timing wasn't quite ready there. Now it is. We're focusing on that as a key capability of intelligence. So, I would say, yes, we definitely roadmap what it might be like to reach the goal of a truly intelligent system like AGI would be.

**Craig:** Yeah. And that kind of intelligence system would be multimodal. It would have vision, language, agency - you'd be able to take actions. So, the game trajectory and that's the agency trajectory, and AlphaCode is the language trajectory. Is that the way to think of it? And then eventually these threads will intertwine.

**Oriol:** Yeah. I'm quite impressed with the way you put it.

**Oriol:** I think that's actually a very nice description of what an AGI to me personally, would look like. There, there would be arguments to be said that perhaps language is all you need, but there could be arguments to say, we don't need language, right? But from looking back at my own research kind of passions, these clearly is tracing by nicely what it might be like you said, like multimodality vision and language, and then capacity to take actions that feels like baking ingredients. And indeed, you can see how we advance the toolbox on these ingredients when we work on these.

**Oriol:** So that's why, by the way, why do we work on ImagNet? Because it enhances the visual capabilities of whatever ultimate models may come up in the future. It's still a very valid benchmark to test those abilities in a very nicely controlled environment. They are very well thought out benchmarks and linking back to AlphaCode, probably the reasoning abilities required. They're still primitive, but the way we advance them in part will be, hey, how are we doing on these benchmarks? And that may stick for years, depending of course, on the rate of progress.

**Craig:** Yeah. You're focused largely throughout your career on sequence-to-sequence learning. And there's been a lot of talk at OpenAI at Qinghua University and some other institutes about developing these systems for video. Being able to describe something and have an AI system produce video in the way that Dall-E produces an image. Are you working on video at all in that way?

**Oriol:** Yeah, so personally I haven't worked on video other than the new model that is Flamingo, which you could think of as a language model. It has a particular way to understand special words that are either images or videos actually. So, we added, basically we enhance the capacity of the amazing language models with some borrowings from ImageNet and vision to have them be doing both.

**Oriol:** Like you can input a video or an image as part of what you're talking about with the model. So that's maybe very recent and it included video as inputs. And as outputs, there is definitely lots of work to be done. It's one of the, I remember talking to one of the professors at Berkeley who is a very well-known computer vision researcher about the idea of the generating.

**Oriol:** You only truly understand the world. If you can take a still image and maybe generate a video from it. So, I've been always fascinated by this. I very lightly touched upon this topic back when we were in the very early days of sequence-to-sequence learning. So, we had one, one of the benchmarks we were doing what's called moving MNIST, which is boringly MNIST digits that moves around - it's a benchmark for video prediction or generation. That was a few years ago.

**Oriol:** But from the recent past at DeepMind, actually not me personally, but there's been work that had very nice follow-up work, in fact, connecting nicely to the science part of our mission, which is to advance science with AGI. So, we had a model that was called, it's called video GANs.

**Oriol:** So GANs from the generative adversarial networks and boringly, like the name is just applied to videos and it was a fairly good model. There's a paper out there with samples. It's, I don't know if it's three years old or something the paper now, and I'm sure there's many more advances beyond. But again, connecting these dots, this model that generated videos, like I don’t know, short videos, clips, were the technique behind the weather prediction efforts. Because the recent paper that came from, many people, including some folks in my team thought predicting how the clouds move over time. That's a video prediction problem.

**Oriol:** So, they use that technique and some enhancements for that problem. Indeed, videos is definitely part of actually in this case, a solution to predicting weather. Very exciting things to do in that domain. I actually, in my opinion. But yeah, personally I think there's a lot to do and it seems very exciting.

**Oriol:** The idea of taking a picture and animating it. So, we'll see maybe in the future, we'll see some of that research happening as well in the group.

**Craig:** This has really been fascinating. I hope I have the opportunity to talk again. I'd love to meet you. I'll be going to conferences again soon. So, thanks Oriol. I appreciate it.

**Oriol:** Yeah. It was my pleasure. As I said it's quite useful, even for oneself to just be looking back a bit more like casually, and as you said, not from the most technical perspective, but also like on why do we do the projects we do.

**Oriol:** Thanks. Thanks, likewise, for your excellent questions and your time.

**Craig:** That's it for this week's episode. I want to thank Oriol for his time. I also want to thank ClearML for their support. Take a look at what they have to offer at clear.ml.

**Craig:** And remember, the singularity may not be near, but AI is about to change your world. So, pay attention.