**CRAIG:** Hi, I'm Craig Smith and this is Eye on AI.

**CRAIG:** This week I talked to Yann LeCun, one of the seminal figures in deep learning development and a longtime prop. Of self supervised learning. Yen spoke about what's missing in large language models and about his new joint embedding predictive architecture, which may be a step toward filling that gap. He also talked about his theory of consciousness and the potential for AI systems to someday exhibit the features of consciousness.

**CRAIG:** It's a fascinating conversation that I hope you'll enjoy.

**CRAIG:** Okay, so Jan, it's great to see you again.

**CRAIG:** Good to see you again.

**CRAIG:** I wanted to talk to you about where you've gone with self supervised learning since last we spoke. In particular, I'm interested in how it relates to large language models because the large language models really came on stream since we.

**CRAIG:** And in fact in your talk about gepa, which is

**YANN:** joint embedding, predictive. , you

**CRAIG:** go there, go. Thank you. Yeah. You mentioned that large language models lack a world model, so I wanted to talk first about where you've gone, will self supervised learning, and where this latest paper stands in your trajectory.

**CRAIG:** To start, if you could just introduce yourself and we'll go from there.

**YANN:** Okay. So my name is uh, YK or who want to do it English Way, and I'm a professor at New York University and at the Courant Institute in the Center for Data Science. And I'm also the Chief AI scientist at FAIR, which is the fundamental AI research lab - that's what FAIR stands for - at Meta, the old Facebook.

**CRAIG:** So tell me about where you've gone with self supervised learning, how the joint embedding predictive architecture fits into your research, and then if you could talk about how that relates to what's lacking in large language models.

**YANN:** Okay. Self supervised running has been, has basically brought about a revolution in natural language processing because.

**YANN:** they're used for pre-training transformer architectures, and the fact that we use transformer architectures for that is somewhat orthogonal to the fact that we use self supervised running, but the way those systems are trained is that you take a piece of text, you remove some of the words, you replace them by blank markers, and then you train the very large neural net to predict the words that are missing.

**YANN:** That's the pre-training phase. And then in the process of training itself the system learns good representations of text that you can then use as input to a subsequent downstream task, I don't know, translation or, hate speech detection or something like that. So that's been truly a revolution over the last three, four years, and including in sort of very practical applications like every sort of top performing content moderation systems on Facebook, Google, YouTube, et cetera, use this kind of technique. and there's all kinds of other applications of that too. Now, large language models are partially this, but also the idea that you can train those things to just predict the next word in a text.

**YANN:** And if you use that, you can have those system generate text continuously. So there, there's a few issues with this. First of all, those things are what's called generative models in the sense that they predict the words, the information that is missing, words in this case. And the problem with GT models is that it's very difficult to represent uncertain predictions.

**YANN:** So in the case of words, it's easy because we just have the system produce essentially what amounts to a score or probability for every word in the dictionary . And so I cannot tell you if the word missing in a sentence like the blank chases the mouse in the kitchen. It's probably a cat. Could be a dog, but it's probably a cat, right?

**YANN:** So you have some distribution of probability over all words in the dictionary. And you can handle uncertainty in the prediction this way.

**YANN:** But then what if you want to apply this to, let's say video, right? So you show a video to the system. You remove some of the frames in that video and you're train it to predict the frames that are missing.

**YANN:** For example, predict what comes next in a video. And that doesn't work. and it doesn't work because it's very difficult to train the system to predict an image, a whole image. We have techniques for that, for generating images or for actually predicting good images that could fit in the video, it doesn't work very well.

**YANN:** Or if it works, it doesn't produce internal representations that are particularly good for a downstream task, like object recognition or something of that type. So attempting to transfer those SSL method that are successful in N LP into the realm of images has not been a big success.

**YANN:** It's been somewhat of a success in audio, but really the only thing that works in the domain of images is those joint invading architectures where instead of predicting the image, you predict a representation of the image, right? So you feed, let's say, one view of the scene to, to the system. You run it through something on that, that computes a representation of it, and then you take a different view of the same scene.

**YANN:** You run it through the same network that produce another representation and you train the system in such a way that those two representations are as close to each other as possible. And the only thing the systems can agree on is the content of the image. So they end up encoding the content of the image independently of the viewpoint.

**YANN:** The difficulty of making this work is to make sure that when you show two different images, it will produce different representations. So to make surethat they are informative about the inputs and your system didn't collapse and just produce always the same representation for everything. But that's the reason why the generative architectures that have been successful in N L P aren't working so well in images is their inability to represent complicated uncertainties if you want.

**YANN:** So now that's for training a system in SSL to learn representations of data. But what I've been proposing to do in the position paper I published a few months ago is the idea that we should use SSL to get machines to learn predictive world models. So basically to predict how the world is going to evolve.

**YANN:** So predict the continuation of a video, for example. Possibly predict how it's going to evolve as a consequence of an action that an intelligent agent might take. Cuz if we have such a world model in an agent, the agent being capable of predicting what's gonna happen as a consequence of its action, will be able to plan a complex sequence of actions to arrive at a particular goal.

**YANN:** And that's what's missing from all the, pretty much all the AI systems that everybody has been working on or has been talking about loudly, except for a few people who are working on robotics where it's absolutely necessary. So some of the interesting work there comes out of the robotics community, the sort of machine learning and robotics community, because there you need to have this capability for planning

**CRAIG:** and the work that you've been doing, is it possible to build that into a large language model? Or is it incompatible with the architecture of large language models?

**YANN:** It is compatible with large language models, and in fact, it might solve some of the problems that we are observing with large language models. One problem with large language models, is that when you use them to generate text, you initialize them with a prompt, right?

**YANN:** So you type an initial segment of a text, which could be in the form of a question or something, and then you hope that it will generate a consistent answer to that text. And the problem with that is that those systems generate text that sounds fine, grammatically sematically, but sometimes they make very stupid mistakes.

**YANN:** And those mistakes are due to two things. The first thing is that to generate that text, they don't really have some sort of objective other than just satisfying the sort of statistical consistency with the prompt that was typed. So there's no way to control the type of answer they will produce, at least no direct way if you want.

**YANN:** That's the first problem. And then the second problem, which is much more acute, is the fact that those large language models have no idea of the underlying reality that language describes. And so there is a limit to how smart they can be and how accurate they can be because they have no experience of the real world, which is really the underlying reality of language.

**YANN:** So their understanding of reality is extremely superficial and only contained in whatever is contained in language that they've been trained on. And that's very shallow. most of human knowledge is completely non-linguistic. It's very difficult for us to realize that's the case. But most of what we learn has nothing to do with language.

**YANN:** Language is built on top of a massive amount of background knowledge that we all have in common, that we call common sense. And those machines don't have that. But a cat has it. A dog has it. So we're able to reproduce some of the linguistic abilities of humans without having all the basics that a cat or a dog has about how the world works.

**YANN:** And that, that's why those systems have failures, essentially. So I think what we would need is an ability for machines to learn how the world works by observation in the manner of babies and infants and young animals. Accumulate like all the background knowledge about the world that constitutes the basis of common sense, if you want .

**YANN:** and then use this world model as the tool for being able to plan sequences of actions to arrive at a goal. So setting goals is also an ability that humans and many animals have. Setting sub goals for arriving at an overall goal and then planning sequences of actions to satisfy those goals. And those language models don't have any of that.

**YANN:** They don't have an understanding of the underlying world. They don't have a capability of planning, for planning, they don't have goals. They can't set themselves goals,other than through typing a prompt, which is a very weak way of doing it.

**CRAIG:** Sure. Where are you in your experimentation with this JEPA architecture?

**YANN:** Pretty early.

**YANN:** So we have forms of it, simplified form of them that we call joint embedding architectures with other 'P,' without the predictive. And they work quite well for learning representations of images. So you take an image, you distort it a little bit, and you train a neural net to produce essentially identical representations for those two distorted versions of the same image.

**YANN:** And then you have some mechanism for making sure that it produces different representations for different images. And so that works really well. We have simple forms of JEPA, the predictive version where the representation of one image is predicted from the representation of the other one. One version of this was actually presented at NeurIPS.

**YANN:** This is called VicRegL, for local. and it works very well for training the neural net to learn representations that are good for image segmentation, for example. But we're still working on a recipe if you want, for a system that would be able to learn the properties of the world by watching videos.

**YANN:** Understanding, for example, very basic concepts like the world is three-dimensional. The system could discover that the world is three dimens. by being shown video with a moving camera. And the best way to explain how the view of the world changes as the camera moves is that every pixel has a depth, that explains parralax motion, et cetera. Once that concept is learned, then the notion of objects and occlusion, objects are in front of others, naturally emerges because objects are part of the image that move together with parralax motion. At least in animate objects. Animate objects are objects that move by themselves.

**YANN:** So there could be also a natural distinction. This ability to spontaneously form categories. Babies do this at the age of a few months. They have an idea without having the names of anything they know. They can tell the car from a bicycle, a chair, a table, a tree, et cetera.

**YANN:** And then on top of this we can build notions of intuitive physics, the fact that objects that are not supported will fall, for example. Babies learn this at the edge of nine months, roughly. It's pretty late, and inertia , things of that type. And then after you've acquired those sort of basic knowledge, background knowledge about how the world works, then you have pretty good ability to predict.

**YANN:** And you can also predict perhaps the consequence of your actions when you start acting in the world. and then that gives you the ability to plan. Perhaps it gives you some basis for common sense, so that's the progression that we need to do. We don't know how to do any of this yet, . We don't have a good recipe for training a system to predict what's gonna happen in the video, for example, with any degree of usefulness,

**CRAIG:** just for the training portion, how much data would you need?

**CRAIG:** It seems to me you would need a tremendous amount of data.

**YANN:** We need a couple hours on Instagram or YouTube. That would be.

**CRAIG:** Really!

**YANN:** The amount of data of raw video data that's available is incredibly large. If you think about, let's say a five year old child, and let's imagine that this five year old child can usually analyze visual percept maybe 10 times a second.

**YANN:** Okay? So that's ten times per second, and if you count how many seconds there are in five years? It's something like 80 millon. So the child has seen 800 million frames, right? Or something like that if you - it's an approximation. Let's say a billion. Sure. It's not that much data. We can have that tomorrow by just saving a YouTube video or something.

**YANN:** So I don't think it's an issue of, of data. I think it's more an issue of architecture, training, paradigm, mathematics. And principles on which to base this. One thing I've said is if you want to solve that problem, we abandon five major pillars of, of machine learning, one of which is those generative models, and to replace them with those joint embedding architectures.

**YANN:** A lot of people in vision are already convinced of that. Then to abandon the idea of doing probabilistic modeling, so we're not gonna be able to represent usefully the probability of the continuation of a video from condition on what we have already observed. We have to be less ambitious about our mathematical framework if you want.

**YANN:** So, I've been advocating for many years to use something called energy-based Models, which is a weaker form of modeling under uncertainty, if you want. Then there is another concept that has been popular for training John meeting architectures over the last few years, which had the first paper in the early nineties actually on something called Siamese Networks.

**YANN:** So it's called Contrastive Learning. And I'm actually advocating against that too. So I'm used to this idea that once in a while you have to come up with new ideas and, and it's gonna be very difficult to convince people who are very attached to those ideas to abandon them. But I think it's time for that to happen.

**CRAIG:** Once you've trained one of these networks and you've established a world model , how do you transfer that to the equivalent of a large language model? One of the things that's fascinating about the development of LLMs in the last couple of years is that they're now multimodal and they're not purely text and language.

**CRAIG:** So how do you combine these two ideas? Or can you, or do you need to?

**YANN:** Yeah. There's, there's two or three different questions in that one question. One of them is, can we usefully transform existing language models whose purpose is only to produce text in such a way that they can do the planning and objectives and things like that?

**YANN:** The answer is yes. That's probably fairly simple to do. Can we train Language model purely on language and expected to understand the underlying reality. And the answer is no. And in fact, I have a paper on this in of all places, a philosophy magazine called Noema, which I co-wrote with a card-carrying philosopher who is a postdoc in my lab at NYU, where we say that there is a limit to what we can do with this because most of human knowledge is non-linguistic.

**YANN:** If we only train systems on language, they will have a very superficial understanding of what they're talking about. So if we want systems that are robust and work, we need them to be grounded in reality. It's an old debate whether AI should be grounded or not. And so the approach that some people have taken at the moment is to basically turn everything, including images and audio, into text or something similar to text. So you take an image, you cut it into little squares, you turn those squares into vectors, that's called tokenization. And now an image is just a sequence of tokens. The text is a sequence of words. Right? Right. And you do this with everything, and you get those multi-model systems and they do something.

**YANN:** Okay. Not clear, that's the right approach long term, but they do something. I think the ingredients that are missing there is the fact that I. If we're dealing with sort of continuous type data or like video, we should use the joint embedding architecture, not the generative architectures that large language models currently use.

**YANN:** First of all, I don't think we should tokenize them because a lot of it get lost in translation when we tokenize images and videos. There's a problem also, which is that those systems don't scale very well with the number of tokens you feed them. So it works when you have a text and you need a context to predict the next word.

**YANN:** That is maybe the 4,000 last words. It's fine, but 4,000 tokens for an image or video is tiny. Like you need way more than that, and those systems scale horribly with the number of tokens who should down. So we're gonna need to do a lot of new innovations and architectures there. And my guess is that we can't do it with generative models.

**YANN:** We'll have to do it with joint embedding.

**YANN:** How

**CRAIG:** does a computer recognize an image without tokenization?

**YANN:** So convolutional nets, for example, don't tokenize. They take an image as pixels. They extract local features, they detect local motifs on on different windows on the image that overlap. And then those motifs get combined into other slightly less local motifs.

**YANN:** And it's this kind of hierarchy where representations of larger and larger parts of the image are, are constructed as you go up in the layers. But there's no point where you cut the image into squares and you turn them into individual vectors. It's more progressive. So there's been a bit of a back and forth competition between the transformer architectures that tend to rely on this tokenization and convolutional nets, which which don't, or in different ways. And my guess is that ultimately what would be the best solution is a combination of the two, where the first few layers are more like convolutional nets.

**YANN:** They exploit the structure of images and video certainly, and then by the time you get to up to several layers, there, the representation is more object based and there you have an advantage in using those, those transformers. But currently, basically the image transformers only have one layer of convolutions at the bottom, and I think it's a bit of a waste and it doesn't scale very well when you want to apply them to video.

**CRAIG:** On the timeline, this is all moving very fast.

**YANN:** It's moving very fast. Yeah.

**CRAIG:** Uh, how long do you think before you'll be able to scale this new architecture?

**YANN:** It's not just scale, it's uh, actually coming up with a good recipe that works that would allow us to just plug a, a large neural net or a small neural net on, on, on YouTube and then learn how the word works by watching in a video.

**YANN:** We don't have that recipe. We don't have, probably don't have the architecture other than some vague idea, which I call hierarchical JEPA. But there's a lot of details to figure out that we haven't figured out. There's probably failure mode that we haven't yet encountered that we need to find solutions for.

**YANN:** And so I can't give you a recipe and I can't tell you if we'll come up with a recipe in the next six months, year, two years, five year, 10 years. It could be quick, or it could be much more difficult than we think. But I think we're on the right path in searching for a solution in that direction. So once we come up with a good recipe, Then it will open the door to new breed of AI systems.

**YANN:** Essentially, they can, they can plan, they can reason, and will be much more capable of having some level common sense perhaps, and have forms of intelligence that are more similar to what we are observing. Animals and humans,

**CRAIG:** your work is inspired by the cognitive processes of the brain. Yeah, right. And, uh, that process of perception and then informing a world model, is that confirmed in neuroscience?

**YANN:** It's a hypothesis that is based on some evidence from both neuroscience and cognitive science. So I, what I showed is proposal for what's called a cognitive architecture, which is some sort of modular architectures that would be capable of. Things like, like planning and reasoning , capability that we observe in animals and humans, and that most current AI systems, except for a few robotic systems, don't have.

**YANN:** So I think that's important in that respect, but it's more of an inspiration really than a sort of direct copy. I'm interested in understanding the principles behind intelligence. But I would be perfectly happy to come up with some procedure that uses backprop at a level but, at a higher level kind of does something different from supervised learning or something like that, which is why I work with self-supervised learning.

**YANN:** And so I'm not necessarily convinced that the path towards satisfying the goal I was talking about of learning world models, et cetera, necessarily goes through finding biologically plausible learning procedures.

**CRAIG:** What did you think of the forward forward algorithm and were you involved in that research?

**YANN:** I was not involved, although I've thought about things that are somewhat similar for many decades, but very few of which is actually published. It's in the direct line of a series of work that Geoff has been very passionate about for 40 years of new learning procedures of different types for basically local learning rules that can train fairly complex neural nets to learn good representations and things like that.

**YANN:** So he started with the Boltzmann machine, which was a really interesting concept. That turned out to be somewhat impractical, but very interesting concept that got a lot of people started. Back prop, which of course he and I both had in, in, in developing. Something I worked on also simultaneously with back prop in the 1980s, called Target prop, where it's an attempt at making backprop more local by computing a virtual target for every neuron in a large neural net that can be locally optimized. Unfortunately, the way to compute this target is non-local, and I haven't worked on this particular type of procedure for a long time, but Yoshua Bengio has published a few papers on this over the last 10 years.

**YANN:** Yoshua, Geoff and I, when we started the deep learning conspiracy in the early 2000 to renew the interest of the community in deep learning, we focused largely on forms of kind of local self supervised learning methods. So things like in Geoff's case that was focused on restricted Boltzmann machines. Yoshua settled on something called denoising Auto Encoders, which is the basis for a lot of the large language model type training that we are using today.

**YANN:** I was focusing more on what's called sparse auto encoders. So this is different ways of doing, of training a layer if you want in a neural net to learn something useful without it being focused on any particular task. So you don't need labelled data, and a lot of that work has been put aside a little bit by the incredible success of just pure supervised learning with very deep models, we've found ways to train very large neural nets with, with very many layers, with just back prop. And so we put those technique on the side and Geoff basically is coming back to them. I'm coming back to them in a different form, a little bit with this JEPA architecture.

**YANN:** And he also had ideas in the past, something called recirculation. A lot of Infomax method, which actually the JEPA uses ideas that are similar, he's a very productive source of ideas that sometimes seems out out of left field and where the community pays attention and then doesn't quite figure it out right away.

**YANN:** And then it takes a few years for those things to disseminate and sometimes they don't.

**CRAIG:** There was a very interesting talk by David Chalmers.

**CRAIG:** At some level it was not a very serious talk because everyone knows, as you described earlier, that large language models are not reasoning, they don't have common sense.

**YANN:** He doesn't claim that they do.

**CRAIG:** No, that's right. But what you're describing with this JEPA architecture, if you could develop a large language model that is based on a world model,

**YANN:** it would not be a large language model.

**YANN:** You'll be a, oh, you'll be a world model . At first, it would not be based on language. It would be based on visual perception, maybe audio perception. If you have a machine that can do what a cat does, you don't need language. Language can be on top of this. To some extent, language is easy, which is why we have those large language models and we don't have systems learn how the world works.

**CRAIG:** Yeah. But let's say that you build this world model and you put language on top of it so that you can interrogate it, communicate with it. Does that take you a step toward what Chalmers was talking about? And I don't want to get into the theory of consciousness, but at least an AI model that would exhibit a lot of the features of consciousness.

**YANN:** David actually has two different definitions for sentience and consciousness. You can have sentience without consciousness. Simple animals are sentient in the sense that they have experience and emotions and drives and things like that, but they may not have the type of consciousness that we think we have. Okay. Or at least the illusion of consciousness we think we have. So sentience, I think, can be achieved by the type of architecture I propose. If we can make them work. Okay. Which is a big if. And the reason I think that is, is that what those systems will be able to do is have objectives that they need to satisfy.

**YANN:** Think of them as drives, and having the system compute those drives, which would be basically predictions of, of, of the outcome of a situation or a sequence of actions that the agent might take. Basically those would be indistinguishable from emotions. So if you are in a situation where you can take a sequence of actions to arrive at a result and the outcome that you're predicting is terrible, results in your destruction. Okay? That creates fear. You tried to figure out like, is there another sequence of action i can take that would not result in the same outcome. If you make those predictions, but there's a huge uncertainty in the prediction, one of which was probability half maybe, is that you get destroyed. It creates even more fear. . And then on the contrary, if the outcome is gonna be good, then it's more like elation, right? So those are long-term prediction of outcomes, which systems that use the architecture I'm proposing, I think will have.

**YANN:** So they will have some level of experience and they will have emotions that will drive their behavior because they will be able to anticipate outcomes and act on them . Now consciousness is a different story. So my full theory of consciousness, which I've talked to David about thinking he was gonna tell me I'm crazy, but he said, no, actually that overlaps with some pretty common theories of consciousness among philosophers.

**YANN:** Is, is the idea that we have essentially a single world model in our head. Somewhere in our prefrontal cortex and that world model is configurable to the situation we're facing at the moment. And so we are configuring our brain, including our world model for solving the problem that, you know, satisfying the objective that we currently set for ourselves.

**YANN:** And because we only have a single world model engine, We can only solve one such task at any one time. This is a characteristic of humans and many animals, which is that when we focus on a task, we can't do anything else. Right? We can do subconscious tasks simultaneously, but we can only do one conscious, deliberate task at at any one time, and it's because we have a single role model engine.

**YANN:** Now, why would evolution build us in a way that we have a single world model engine? There's two reasons for this. One reason is that single world model engine can be configured for the situation at hand, but only the part that changes from one situation to another. And so it can share knowledge between different situations.

**YANN:** The physics of the world doesn't change if you are building a table or trying to jump over a, a river or something. And so your sort of basic knowledge about how the world works doesn't need to be reconfigured. It's only the thing that depends on the situation at hand. So that's one reason.

**YANN:** And then the second reason is that if we had multiple models of the world, it would have to be individually less powerful because you have to all fit them within your brain. And that's a limited size. So I think that's probably the reason why we only have one. And so if you have only one world model that needs to be configured for the situation at hand, you need some sort of meta module that configures it, figures out like what situation am I in? What sub goals should I set myself and how should I configure the rest of my brain to solve that problem? And that module would have to be able to observe the state and capabilities - would have to have a model of the rest of itself of the agent, and that perhaps is something that gives us the illusion of consciousness.

**YANN:** must say this is very speculative. Okay. I'm not saying this is exactly what happens. But it fits with a few things that we know about, about consciousness.

**CRAIG:** You were saying that this architecture is inspired by cognitive science or neuroscience.

**CRAIG:** How much do you think your work, Geoff's work, other people's work at the kind of the leading edge of deep learning or machine learning research is informing neuroscience or is it more the other way around? Certainly in the beginning it was the other way around, but at this point it seems that there's a lot of information that then is reflecting back to those fields.

**YANN:** Yeah, it's always been a bit of feedback loop. So new concept in machine learning have driven people in neuroscience and cognitive science to use computational models if you want of what they are studying. And many of my colleagues, my favorite colleagues work on this. The whole field of computational neuroscience basically is around this.

**YANN:** And what we are seeing today is a big influence, or rather a wide use of deep learning models such as convolutional nets and transformers. As models, explanatory model of what goes on in the visual cortex, for example. So the people you know for a number of years now who have done FMRI experiments and then showed the same image to a subject in the FMRI machine and to a convolutional net and then try to explain the variance they observe in the activity of various areas of the brain with the activity that is observed in the corresponding neural net. And what comes out of the studies is that the notion of multilayer hierarchy that we have in convolutional nets matches the type of hierarchy that we observe in the, at least in the ventral pathway of the visual system. So V1 corresponds to the first few layers of the conventional net and in V2 to some of the following layers on V4 more.

**YANN:** And then the V4 temporal cortex to the top layers are the best explanation of each other. If you try to do the matching. One of my colleagues at Fair Paris who has a dual affiliation also with Neurospin t hat academic lab in Paris has done the same type of experiment using transformer architecture as language models essentially, and observing brain activity of people who are listening to stories and attempting to understand the story so that they can answer questions about the story or, or give a summary.

**YANN:** and there the matching is not that great in, in the sense that there is some sort of correspondence between the type of activity you observe in those large transformers and the type of activity also in the brain, but the hierarchy is not nearly as clear and what is clear is that the brain is capable of making much longer term prediction that those language models are capable of today.

**YANN:** So that begs the question of what we, what are we missing in terms of architecture? To some extent, it jives with the idea that the models that we should have should build hierarchical representations of the percept, the different levels of abstraction, so that the highest level of abstraction are able to make long-term predictions that perhaps are less accurate than the lower level, but longer term. We don't seem to have that in current models.

**CRAIG:** I had a question I wanted to ask you since our last conversation. You have a lot of things going on. You teach, you have your role at Facebook, your role, I think at C V P R, or how do you work on this? Like three days a week or two hours a day where you're just focused.

**CRAIG:** and then are you tinkering with code or with diagrams or is it in iterations with some of your graduate students? Or is this something where it's kind of always in your mind and you're in the shower and you think, yeah, that might work. I'm just curious how

**YANN:** all of the above . so first of all, One thing you have to understand is that my my position at, at Meta at FAIR is not a position of management.

**YANN:** I, I don't manage anything. I'm chief scientist, which means I try to inspire others to work on things that I think are promising, and I advise several projects that I'm not personally involved in. I work on strategy and orientations and things like this, but I don't do day-to-day management. I'm very thankful that Joel Pino is doing this for FAIR and doing very, a very good job.

**YANN:** I'm not very good at it either, so it's probably better if I don't, if I don't do it. So that allows me to spend quite a bit of time on research itself. And I don't have a group of engineers and scientists working with me. I have a group of more junior people working with me, students and postdocs, both at at FAIR and at NYU..

**YANN:** Both in New York and in Paris and, and working with students in postdoc is wonderful because they're fearless. They're very creative. Many of them have amazing talents in theoretical abilities or implementation abilities or a knack for making things work. And so what happens very often is either one of them will come up with an idea whose results surprised me and said, uh, I was thinking about this all wrong, and that's the best thing that can happen.

**YANN:** Or sometimes I come up with an idea and turns out to work, which is great. Usually not in the form that I formulated. No, normally it's, there's a lot of contributions that have to be brought to an idea for it to make it. And then what's happened also quite a bit in the last few years is I come up with an idea that I'm sure it's gonna work.

**YANN:** And a few students in postdoc tried to make it work and they come back to me and said, oh, sorry, it doesn't work. And here is a failure mode. Oh yeah, we should have thought about this. Okay, so here's a new idea to get around this problem. So for example, several years ago I was advocating for the use of generative models with latent variables to handle the uncertainty.

**YANN:** and I completely changed my mind about this. Now, I'm advocating for those joint embedding architectures that do not actually predict, and I was, I more or less invented those contrastive methods that a lot of people are talking about and using at this point, and I'm advocating against them now in favor of those methods such as VicReg or Barlow twins that basically, instead of using contrastive methods, can try to maximize the information content of representations. And that idea of information maximization , I'd known about for decades because Geoff was working on this in the 1980s when I was a postdoc with him, and he abandoned the idea pretty much. He had a couple papers with, uh, one of his students called Sue Becker in the early nineties that showed that it could work, but only in sort of small dimension, and he pretty much abandoned it.

**YANN:** And the reason he abandoned it is because of a major flaw with those methods due to the fact that we don't have any good measures of information, content, or the measures that we had are upper bound, not lower bound. So we can't try to maximize information content very well. And so I never thought that those methods could ever work because of my experience with, that..

**YANN:** And one of my postdocs Stephane Deny, actually kind revived the idea and showed me that that it worked. That was the Barlow twins paper. And so he changed my mind. And so now that we had a new tool, information maximization applied to joint embedding architectures and came up with an improvement of it called VicReg.

**YANN:** And, and now we are working on that, but there are other ideas we're working on to solve the same problem with other groups of people at the moment. Which probably will come up in the next few months. So we don't, again, we don't have a perfect recipe yet, and we're looking for one and hopefully one of the things that we are working on with Stick.

**CRAIG:** Yeah. Are you coding models? And then training them and running them, or are you conceptualizing and turning it over to someone else?

**YANN:** So it's mostly conceptualizing and mostly letting the students and postdocs doing the implementation. Although I do a little bit of coding myself, but not enough to my taste.

**YANN:** I wish I could do more. I have a lot of postdocs and students, and so I have to devote sufficient amount of my time to interact with them. And then leave them some breathing room to do the work that they do best. And so it, it's interesting question because that question was asked to, to Geoff after his talk, right? Yeah. And he said he was using MatLab and, and he said, you have to do those things yourself, because if you give a project to a student and a project come back saying it doesn't work, you don't know if it's because there is a conceptual problem with the idea or whether it's just some stupid detail that wasn't done.

**YANN:** When I'm facing this, that's when I start looking at the code and perhaps experimenting with it myself. Or I get multiple students to collaborate on that project so that if one makes a, an error, perhaps the other one will detect what it is. I love coding. I just don't do as much as I'd like to.

**YANN:** Yeah.

**CRAIG:** This JEPA or the forward forward, things have moved so quickly you think back to when the transformers were introduced, or at least the attention mechanism and that kind of shifted the field. It's difficult for an outsider to judge when I hear the JEPA talk, is this one of those moments that, wow, this idea is gonna transform the field?

**CRAIG:** Or have you been through many of these moments and they contribute to some extent, but they're not the answer to shift the paradigm.

**YANN:** It's hard to tell at first, but whenever I kinda keep pursuing an idea and promote it, it's because I have a good hunch that they're gonna have a relatively big impact.

**YANN:** And it was easy for me to do before I was as famous as I am now because I wasn't listened to that much. So I could yeah. Make some claim. And now I have to be careful what I claim because a lot of people listen to me. Yeah. And it's the same issue with Geoff. So Jeff, for example, a few years ago was promoting this idea of capsules.

**YANN:** Yeah. And everybody was thinking this is gonna be a big thing, or people started working on it. It turns out it's very hard to make it work, and it didn't have the impact that many people thought it would. Including Geoff. And it turned out to be limited by implementation issues and stuff like that.

**YANN:** The underlying idea behind it is good, but like very often the practical side of it kills it. It was the case also with Boltzmann machines, conceptually super interesting. They just don't work that well. They don't scale very well. They're very slow to to train. Conceptually. It's a very interesting idea that everybody should klnow about..

**YANN:** So there's a lot of those ideas that are conceptual that allow us, there are some mental objects that allow us to think differently about what we do, but they may not actually have that much practical impact. For forward, we don't know yet. Okay. It could be like the wake sleep algorithm that Geoff talked about 20 years ago or something, or he could be the new backprop.

**YANN:** We don't know. Or the new target prop, which is interesting, but not really mainstream. Because it, it has some advantages in some situations, but it's not, it brings you like an improved performance on some standard benchmark that people are interested in. So it doesn't have the wide appeal perhaps, so it's hard to figure out.

**YANN:** But what I can tell you is that if we figure out how to train one of those JEPA-style architecture from video, and the representations that it learns are good and the predictive model that it learns are good. This is going to open the door to a new breed of AI systems. I have no, no doubt about that.

**CRAIG:** It's exciting. The speed at which things have been moving in, particularly in the last three years

**YANN:** about, about transformers and the history of transformers. One thing I wanna say about this is, We see the most visible progress, but we don't realize like how much of a history there was behind it. And even the people who actually came up with some of those ideas don't realize that their ideas actually had roots in other things.

**YANN:** For example, back in the nineties, people were already working on things like that we now called mixture of experts and also multipl interactions, which at the time were called semi pie networks or things like that. So since the idea that instead of having two variables that you add together with. You multiply them and then you have a way or you have weights before you multiply.

**YANN:** It doesn't matter. This goes back a long time since the 1980s. Uh, and, and then you had ideas of minimally combining multiple inputs with weights that are between 0 1 1 and sum to one and are data dependent. So now we call this attention, but this is a circuit that was using in mixture of expert models.

**YANN:** Back in the early nineties also. Right. Then there were ideas of neural networks that have a separate module for computation and memory. That's two separate modules, right? So one module that is a classical neural net, and the output of that module would be an address into an associated memory.

**YANN:** That itself would be a different type of neural net. And those different types of neural net associated memories use what we now call attention. So they compute the similarity or the product between a query vector and a bunch of key vectors, and then they normalize them. So they are set to one, and then the output of the memory is the weighted sum of the value vectors.

**YANN:** There was a series of papers by my colleagues in the early days of FAIR actually in 2014 15, one called Memory Network, one called End-to-End Memory Network, one called Stack Augmented Memory Network, another one called. Key value memory, network, and then a whole bunch of things. And those use those associated memories that basically are the basic modules that are used inside the transformers.

**YANN:** And then attention mechanism like this were popularized in around 2015 by a paper from Yoshua Bengio's Group at Mila, and demonstrated that they are extremely powerful for doing things like translation, language translation in NLP. and that really started the craze on attention. And so you combine all those ideas and you get a transformer that uses something called self attention, where the input tokens are used both as queries and keys in a associative memory.

**YANN:** Very much like a memory network. And then you view this as a layer if you want. You put several of those in a layer and then you stack those layers and that's what the transformer is.. The affiliation is not obvious, but there is one. Those ideas have been around and people have been talking about it and there's similar work also, around 20 15, 16, and from Deep Mind called the neural Turing machine or Differential Neural Computer.

**YANN:** Those ideas that you have a separate module for computation and another one from memory, I think this idea is very powerful. The big advantage to transformers is that the same way computational nets are equi variant to shift.

**YANN:** If you shift the input of a al net, the output also shifts but otherwise doesn't change. In a transformer. If you permute the input tokens, the output tokens get permuted the same way, but are otherwise unchanged. So ConvNets are e equivalent to shift. Transformers are equivalent to permutation and with a combination of the two, it's great, which is why I think the combination of ConvNets at the low level and transformer at the top. I think for natural input data like image and video is a very good combination.

**CRAIG:** Is there a combinatorial effect as a field progresses? All of these ideas create a cascade of new ideas. Yeah. Is that why the field is speeding up?

**YANN:** It's not the only reason, the, there's, there's a number of reasons. One of the reasons is that you build on each other's ideas, et cetera, which of course is the hallmark of science in general. Also art. But there is a number of characteristics I think that help that to a large extent. One in particular is the fact that most research work in this area now comes with code that other people can use and build upon, right? So the habit of distributing your code in open source, I think is a, is an enormous contributor to the acceleration of of progress. The other one is the availability of the most sophisticated tools. Like PyTorch for example, or TensorFlow or JAX or things like that where researchers can build on top of each other's code base basically to come up with really complex concepts.

**YANN:** And all of this is permitted by the fact that some of the main contributors that are from industry to those ideas don't seem to be too obsessive compulsive about IP protection. Uh, so. And in particular, Facebook is very open. We may occasionally fight patents, but we are not gonna sue you for infringing them unless you sue us.

**YANN:** And Google has a similar policy. You don't see this much from companies that tend to be a little more secretive about their research, like Apple and Amazon. But

**CRAIG:** although I just talked to Sammy Bengio, he's trying to implement that openness,

**YANN:** more power to him. Good luck. It's a culture change for a company like Apple.

**YANN:** So this is not a battle I want to fight, but if he can win it, like good for him. Yeah. It's difficult. It's a very difficult battle. Also, I think another contributor is that there are real practical commercial applications of all of this. They're not just imagined, they're real. And so that creates a market and that increases the size of the community.

**YANN:** And so that creates more appeal for new ideas, right? More, more, more, uh, outlets if you want, for new ideas.

**CRAIG:** Do you think that this hockey stick curve is gonna continue for a while or do you think we'll hit a plateau ?

**YANN:** It's difficult to say. Nothing looks more like a an exponential than the beginning of a sigmoid,

**YANN:** so every natural process has to saturate at some point? Yeah. The question is when, and I don't see any obvious wall that is being hit by AI research at the moment, it's quite the opposite. Seems to be an acceleration in fact of progress. And there's no question that we need new concepts and new ideas.

**YANN:** In fact, that's the purpose of my research at the moment. Cause I think there are limitations to current approaches. So this is not to say that we just need to scale up deep learning and turn the crank and we'll get to human level intelligence. I don't believe that. I don't believe that it's just a matter of making reinforcement planning more efficient.

**YANN:** I, I don't think that's possible with the current way reinforcement planning is formulated, and we're not gonna get there with supervised learning either. I think we definitely need new and innovative concepts, but I don't see any slowdown yet. I, I don't see any people turning away from me. I think it's obviously not going to work, blah, blah, blah, despite screams of various critiques, right? Yeah, sure. About that. But, uh, but they to some extent at the moment are fighting a rear guard battle. Yeah. Because they, they plant a flag. They said, you're never gonna be able to do this, and then turns out you can't do this. So they plant a flag a little further down and haha, now you're not gonna be able to do this.

**YANN:** Right. So exciting.

**CRAIG:** Yeah. Okay. My last question. Are you still doing music?

**YANN:** I am.

**CRAIG:** And are you still building instruments?

**YANN:** I'm building electronic wind instruments. Yes. I'm in process of designing a new one.

**CRAIG:** Wow. Yeah. I think I said this last time, maybe I could get some recordings and put them into the podcast or something.

**CRAIG:** That'd be fun.

**YANN:** All right. Yeah, I probably told you I'm not such a great performer. I'm, uh, . I'm probably better at conceptualizing and building those instruments than playing them, but, But yeah, it's it's possible.

**CRAIG:** That's it for this episode. I want to thank Yann for his time. If you want to read a transcript of today's conversation, you can find one on our website www.eye-on.ai. Feel free to drop us a line with comments or suggestions at craig@eye-on.Ai.

**CRAIG:** And remember, the Singularity may not be near, but AI is about to change your world, so pay attention.