

DAISY THE GREAT	00:00	Built my house on hollow ground
CRAIG:	00:07	Hi, this is Craig Smith with a new podcast about artificial intelligence. This week I talked to Yann Lecun, one of the brightest minds in machine learning today. His work lies behind some of the most critical AI applications, most notably computer vision systems that power everything from face recognition software to self-driving cars. Yann recently won the Turing Award, the highest prize in computer science and artificial intelligence research, together with his long-time collaborators, Yoshua Bengio and Geoff Hinton. We talked about Yann's first computer, about how the oboe and other wind instruments led him into computer science, and about his work on self-supervised learning, which he believes will take us to human-level intelligence in machines. I hope you find the conversation as fascinating as I did.
CRAIG:	01:09	You're French.
YANN:	01:11	I'm French, French and American now. But yes, I was born near Paris. I grew up near Paris. On the, on the outskirts.
CRAIG:	01:18	in the banlieues or?
YANN:	01:19	Yeah. Yeah. Uh, not that close. North-northwest kind of Soisy-sous-Montmorency , Enghien-les-Bains , Eaubonne , this, this kind of this area. I was always interested in engineering, science and things like this and I was interested in, uh, you know, at the abstract level, the question of intelligence, how did human intelligence appear. Um, what is intelligence really? So that's a question I was fascinated by since I was a kid essentially. So that goes back a long time. You know, I saw 2001: A Space Odyssey when I was nine years old and you know, I thought the concept of an intelligent machine was, uh, it was amazing. And that movie also put together, uh, not, not just intelligent machines but how, you know, some hypothesis about how human intelligence evolved. Right?
CRAIG:	02:04	Yeah.
YANN:	02:05	So, so that was always a topic I was fascinated by. And what you see here in this room is that, all this decoration, these are pictures from ...
CRAIG:	02:11	Oh, sure,
YANN:	02:12	... the 2001 movie.
CRAIG:	02:12	Yeah.

YANN: [02:13](#) Um, so, um, so that's how I got interested in this question. And you know, I always thought I could be a scientist or an engineer. My Dad was an engineer, mechanical engineer. Um, I studied engineering, electrical engineering and uh, I took a lot of, uh, courses in applied math and physics and things that were kind of uh fairly fundamental, and I started doing projects in, I guess what you could call AI, I guess. I got interested in neuroscience.

CRAIG: [02:43](#) This was all in high school or this is now in college?

YANN: [02:44](#) This is in college.

CRAIG: [02:46](#) And that was where?

YANN: [02:48](#) That was at a school called [ESIEE](#) which is a, not a huge, top engineering school. It was a decent engineering school for electrical engineering in Paris.

YANN: [02:58](#) Uh, that school has since moved to, east of Paris, like near Disney, [EuroDisney](#). But back then it was, uh, in Paris in the [15th arrondissement](#). And, um, and I did a bunch of research projects with various professors on sort of, you know, kind of computer models of neurons and stuff like that. And I, I discovered the existence of learning machines by reading a philosophy book, actually. It was a debate between [Noam Chomsky](#), the linguist and [Jean Piaget](#), the developmental psychologist, and they were arguing about the, you know, nature-nurture debate. Right. And, uh, you know, is, is language acquired? Is it innate?

CRAIG: [03:38](#) Yeah. I just had a conversation with [Ken Church](#), who was a student about all of this.

YANN: [03:43](#) Right. Yeah. And he is a former colleague from [Bell Labs](#) actually.

CRAIG: [03:46](#) Oh, that's right.

YANN: [03:47](#) So, uh, you know, I, I read this thing and there was, uh, one of the talks was, that was trans, transcribed in that book was by [Seymour Papert](#) from MIT where he was singing the praise of a model called the [perceptron](#), which I'd never heard of before. Um, which you know, is one of the early, kind of simple learning models, right, from the fifties and sixties. Um, so I, I read about this and I said, a machine that's capable of learning, I find that absolutely fascinating. I always thought learning was indistinguishable or, or inseparable from, from intelligence. That you could not, you know, I thought the task of building an intelligent machine was, was impossible. But building a machine that could learn, maybe it was a possibility. Right. Um, so I started reading the entire literature. This must've been in maybe third year of college or something.

CRAIG: [04:35](#) What year would that be?

YANN: [04:36](#) That would be 1980, 1981.

CRAIG: [04:39](#) Wow. Yeah.

YANN: [04:40](#) Um, and then discovered that nobody was actually working on this anymore. The entire field had been abandoned in the late sixties because of a book coauthored by the same guy, right, Seymour Papert, who was, you know

CRAIG: [04:54](#) Oh, was that the [Minsky](#) book? I didn't realize he was the co-author.

YANN: [04:54](#) Minsky and Papert book. [Perceptrons](#).

CRAIG: [04:57](#) Yeah, Perceptrons, right.

YANN: [04:58](#) So that pretty much killed the field, or at least had a big impact on the, on the field. Um, and there he was, you know, he was 10 years later actually kind of praising the perceptron, you know, it's kind of interesting, um, to argue against Chomsky's argument that language is innate.

CRAIG: [05:17](#) Yeah.

YANN: [05:18](#) Uh, so I, I, I got interested in this and I thought, why, why is it that people abandoned this idea? You know, that sounds like a really good, really cool idea. And then realized very quickly that what people were kind of, you know, the wall that people were hitting against at that time was that they knew that the perceptron in itself was very limited and you, you, you, you know, there was a need for being able to kind of build multilayer neural nets.

YANN: [05:46](#) And they knew that. They just didn't figure out how to do it. And it's probably mostly because they had the wrong neurons. The neurons people were using in neural nets at the time were binary neurons and that's incompatible with things like [backprop](#).

CRAIG: [05:57](#) Right.

YANN: [05:57](#) Um, and so the idea didn't just, just didn't come up, even though the basic idea of doing backprop actually existed in the context of [optimal control](#) since the 60s. So yeah. Um, so, you know, the whole field kind of died or rather changed name because instead of working on intelligent machines, the people working on the perceptron at the time and, and relative models started with renaming what they were doing. It was called

[adaptive filters](#) and things like that, but it was the same thing really, that we're doing now. Um, and so I started thinking about how can we train multilayer networks and kind of stumbled on an idea that, um, which is very close to backprop.

YANN: [06:40](#)

So this must have been 1983 or so.

CRAIG: [06:42](#)

Yeah.

YANN: [06:43](#)

Uh, which was the idea of using the weights that are used the neural net forwards and use them backwards. I wasn't using them to backpropagate gradient. I was using them to backpropagate targets. So basically, to compute [inaudible] targets for every neuron. My neurons were still binary. And the reason was the computers we had access to at the time were very slow at computing multiplications. Right. And so, if you have binary neurons, you don't need to do multiplications. Um, and then, uh, and then I talked to a friend of mine who was doing a PhD in, in, in, in control in robotics, who told me about those methods that, you know, in robotics that people had or in, in optimal control that people had come up, come up with in the 60s. And I said, that's, that was very much like the stuff I'm working on.

YANN: [07:24](#)

Um, and so I, I. So, that's when I kind of came up with backprop. But then, um, but then a month later or so, a couple months later, I met [Geoff Hinton](#) who came to a meeting in France and he, we, we, I really wanted to meet him because he had written a paper on [Boltzmann machines](#), which I think was, you know, it was, it was the first paper I saw that basically allowed to train neural nets that had hidden units. Right. So, I thought that's, you know, I want to talk to him, either him or [Terry Sejnowski](#). I had actually met Terry a couple of months earlier. Um, and so, I meet, I meet, um, Geoff at this meeting in France and, uh, and, and we start talking and I tell him what I'm working on. I had a paper in the proceedings of the conference he came to that, talked about this target prop idea and he, he read it and he said, that's really close to backprop.

YANN: [08:20](#)

Uh, and so we talked together and, and I told him what I was working on, which was backprop and he told me what he was working on, which was also backprop. And so, you know, uh, and he said, oh, I'm writing a paper and you know, I'm going to cite your paper in mine. Uh, I was absolutely delighted by it, um, and uh, and that's, you know, that's how I met him.

CRAIG: [08:41](#)

Was he as prominent then in the, in the field.

YANN: [08:45](#)

He was somewhat because of the Boltzmann machine paper and it was just the, the beginning of the wave or the second

wave of neural nets. This is mid 1985 so nobody knows about backprop yet, but, um, um, but Boltzmann machines had been around for a couple of years and, um, it was clear that there was, you know, there's going to be a lot of people trying to kind of restart, you know, working on neural nets. Um, I mean he wasn't nearly as famous as he is now of course, but, uh, but, but, but he was kind of, you know, fairly well known. So, um, so then he told me, uh, I'm organizing a summer school next summer, um, one year later, and said, you know, um, um, I'm going to invite you to that summer school and we're going to bring together all the students who are working on neural nets and all that stuff. And that was sort of, really sort of the, the founding event, if you want, of the, sort of, neural net community really.

CRAIG: [09:41](#)

That was in California.

YANN: [09:43](#)

That was a [Carnegie Mellon](#).

CRAIG: [09:44](#)

Oh, Carnegie. He was at Carnegie Mellon at that time.

YANN: [09:46](#)

That was in 1986, the one I'm talking about.

CRAIG: [09:48](#)

Okay.

YANN: [09:49](#)

So I was at [Université Pierre and Marie Curie](#), which now has different name. Now it's called [Sorbonne University](#). But, uh, but I actually didn't spend much time at that school because, uh, the, the person who was officially my advisor was a professor at that university. So that's where it was officially registered. But I was actually spending most of my time in two labs. One was at the engineering school where I did my undergrad, because that school happened to have fairly powerful computers for the time, uh, which I could use. And, uh, also in an independent lab where, uh, my advisor spent some time and, and a few other people who had kind of a common interest in what they called [automata networks](#), which was sort of connected with neural nets. And you know, this people kind of got involved in neural nets, uh, pretty, uh, pretty, pretty quickly. So, um, in 1987, I, uh, I graduated, this is just when, uh, you know, the backprop paper was, you know, had been published a year before the, uh, [NETtalk](#), uh, thing that Terry Sejnowski builds, you know, he'd kind of run around and give talks about this.

YANN: [10:54](#)

All of a sudden, you know, the people in France who been basically ignoring me for years, uh, started talking to me because, you know, I was, I was the local expert, that was involved in neural nets, right.

CRAIG: [11:04](#)

Right.

YANN: [11:05](#) Um, so I finished my PhD. Uh, Geoff was actually on my thesis committee and then I did a postdoc, I started a postdoc with him in Toronto. So, he moved to Toronto from CMU in the summer of 1987 and I arrived in Toronto two weeks after him.

CRAIG: [11:20](#) I see, I see.

YANN: [11:26](#) Yeah. So, I, uh, I was invited in a winter in 1987 to, um, Montreal. I gave a talk there and in the room was this, um, this kid who asked really smart questions about, about neural nets and there were very, very few people working on neural nets at the time. Uh, that was Yoshua. I kept an eye on him.

CRAIG: [11:39](#) Was [Sammy](#) there by chance?

YANN: [11:40](#) I didn't meet Sammy until three years later, I think.

CRAIG: [11:45](#) I just think it's fascinating that there are two brothers in this field and then it turns out your brother works for Google.

YANN: [11:51](#) Google in Paris.

CRAIG: [11:52](#) Yeah. It's amazing.

YANN: [11:54](#) He's not working on machine learning, optimization research

CRAIG: [11:55](#) Yeah, but still it's remarkable. What did your parents do that they have these two brilliant kids doing stuff?

YANN: [12:04](#) Well, my mom was a homemaker. My, uh, my dad was a mechanical engineer working in the aerospace industry and he was kind of a, a bit of a mechanical genius. I mean, my brother and I learned everything from him.

CRAIG: [12:17](#) Wonderful.

YANN: [12:17](#) You know, when we were kids we would like, you know, build model airplanes and you know, electronics and stuff like that.

CRAIG: [12:24](#) Yeah. What was your first computer?

YANN: [12:27](#) Oh, my first computer I bought when I was still in high school. This was in 1977. I'm sorry, so this was before you could buy a computer with a screen and a keyboard. Right. This was a single board computer which had, um, it was just a printed circuit board with a [6502 microprocessor](#), 1K of RAM, uh, 4K of ROM or something, , you know, a [hexadecimal keyboard](#) and LED display with six digits, seven segments, six digits. And you would program it directly in, in machine language, you know, in hexadecimal.

CRAIG: [12:57](#) Who was the maker?

YANN: [12:59](#) Um, it was company call MOS Technologies. [SynerTek](#), actually. The company that made the board was called SynerTek, SynerTek, I don't know how you pronounce it. Um, uh, it didn't last very long, but um, um, but you know, you, you basically had to know like how a microprocessor worked inside to be able to program those things. You know, there was no, you know, high-level language or [BASIC](#) or whatever, right? It was directly, um, uh, machine language to program into, and my main motivation for getting into this was two things, you know, long term, AI, whatever, but short term, music. I was into electronic music, uh, not just electronic, music in general. And I wanted to use computers for music essentially.

CRAIG: [13:42](#) And, and the music thing faded or did you do it on the side?

YANN: [13:47](#) [It didn't fade](#). I, um, I mean I have a bunch of analog synthesizers and various other things, and I build wind controllers. I'm, uh, I'm a wind, wind, uh, uh, instrument player.

CRAIG: [13:59](#) Oh, which instruments.

YANN: [14:01](#) You know, like oboe, recorder, various exotic renaissance instruments. So, but I, I build [wind controllers](#), you know, um, so you can blow into them, you know, it sort of figures out your fingering and these sort of various controls and, and so there's, there's a bunch of that I just bought and there's a bunch that I built.

CRAIG: [14:19](#) Wow. Fascinating.

MUSIC INTERLUDE

CRAIG: [14:27](#) Uh, okay. So, so fast forward then, uh, there was a moment when neural nets died again, largely because of insufficient hardware, I think. And then, and that was the point at which you, Geoffrey, that's why everyone talks about the three of you together, right? Didn't you come together and sort of revitalize the field?

YANN: [14:52](#) It's slightly more complicated than that. So, uh, there was like a, you know, a very favorable period, the late eighties, early nineties. So, I joined Bell Labs in the late eighties. Uh, I hired Yoshua actually at Bell Labs. He worked there for a few years with us. The early, the first half of the nineties. Uh, and other people actually, [Léon Bottou](#), [Patrice Simard](#). I mean, a bunch of, you know, [Vladimir Vapnik](#), [inaudible]. I mean, those are kind of the big names in the community. Um, and, um, and we became really successful with the techniques we developed.

[Convolutional nets](#), you know, we kind of were able to build systems that could read checks and zip codes and various other things and the engineering divisions of the company actually commercialized that.

CRAIG: [15:39](#)

Yeah. I remember

YANN: [15:39](#)

It was quite successful. But just at the time they started being successful commercially, in 1995 or so, two things happened. First, the machine learning community lost interest in neural nets. And the second thing is that AT&T, which, and Bell Labs broke itself up. Uh, for the second time.

CRAIG: [15:59](#)

Yeah.

YANN: [16:00](#)

Uh, and, uh, uh, and that was a bit difficult for me because, um, when the company broke up, the research group who I was in stayed with AT&T, so that took the name AT&T Labs. The reason being that the guy in charge of, who became in charge of Bell Labs, uh, didn't like machine learning at all. So, all the machine learning people went to AT&T. Uh, the engineering group was in [Lucent Technologies](#), which was one of the spin offs. And the product group was [NCR](#), which was yet another spin off. And so, the whole project was basically disbanded. So, you know, this was like my biggest technical success, uh, as well as technology transfer success. And all of a sudden it was taken away, taken away from me almost on the same day that we were celebrating its deployment essentially.

YANN: [16:53](#)

At the same time I was promoted to department head, so now I had to also run a lab. And this was the early days of the Internet. So, basically, I kind, kinda tried to figure out what, what I should do next and for six years essentially didn't work on machine learning at all. I wrote papers on stuff we did before, but, uh, I started a new project called [DjVu](#), DjVu. That was sort of a image compression technology that was somewhat successful at one point. Um, and then ran the department until early 2002 when I left AT&T and that's, that's when I restarted working on machine learning, deep learning, et cetera. And that's just about when Yoshua, Goeff and I kind of got together and sort of started the, the deep learning conspiracy if you want.

CRAIG: [17:37](#)

Yeah. And where were you then when, when, when you left?

YANN: [17:40](#)

So I left AT&T and went to the [NEC Research Institute](#) in Princeton. Right. Uh, I stayed there for about 18 months. Uh, I brought a lot of people from my lab there, about four, four people or five people actually from my lab, from AT&T, Léon Bottou,, Vladimir Vapnik, [Eric Cosatto](#), [Hans Peter Graf](#) and couple other people. And, um, but then I only stayed 18 months

because NEC basically was kinda, you know, very sort of complicated transition period and they were not interested in the - like other people I wanted to work with at the NEC Research Institute were leaving. They were, you know, physicists, uh, neuroscientists, a quantum physicist and so, you know, all the interesting people kind of started leaving. So, I, you know, I started looking for another gig, and that's when I, I joined NYU, that was in 2003.

CRAIG: [18:32](#) Oh, is that right? I didn't realize you'd been here that long. Yeah. Yeah. So, so a lot of the really important work has been done at NYU?

YANN: [18:42](#) Uh, yeah. So, I mean, a lot of the foundational work was done, right, when I was in Toronto and Bell Labs, uh, in the late eighties, early nineties. Uh, and then I kind of, we started working on this at NEC, but really sort of the more recent work yeah, was here at NYU.

MUSIC INTERLUDE

CRAIG: [18:59](#) So can you talk about the importance of learning representations in neural nets? I mean, neural nets depend on representations. So that's the whole point.

YANN: [19:13](#) All of AI relies on representations. The question is where do those representations come from? So, uh, the classical way to build a pattern recognition system was, was to build what's called a [feature extractor](#), which turns the input signal, whether it's an image or audio or whatever into representation, generally a vector or a list of numbers that represent the salient features of the input that are useful for the task that you're trying to do, right? So, if you had tried to do speech recognition, you want some representation that takes into account the nature of the sound that is being pronounced but doesn't care about the identity of the speaker, for example, or the pitch of the voice, right? Unless this is Chinese.

YANN: [19:53](#) Um, if you want to do speak recognition, like figuring out who is speaking, but you don't care about what, what is being spoken, then it's other features that you need to extract from the input. Um, same for images, right? If you want to, uh, recognize the object in the image, there are certain features that are probably useful to extract. But people had for things like, um, like image recognition, people, had been working on the problem of what are the right features for recognizing objects. And there was no real good way of extracting features that would be general for any recognition problems that you had. So, people had this, a whole lot of papers on what features you should extract if you want to recognize, uh, written digits and other features you

should extract if you want to recognize like a chair from the table or something or detect.

CRAIG: [20:40](#)

And that was just based on trial and error or intuition?

YANN: [20:43](#)

Intuition, engineering, you know, a little bit of theory but, but mostly, you know, [signal processing](#) methods and stuff like that. Um, you know, in supervision that means, it means edge detection, things like that, you know, and uh, audio processing for representing speech for example, that means doing, for your transform to extract the [spectral content](#) and then doing some operations on it. Um, so, you know, every specialty has its own way of extracting features to represent the input signal. Right? And most people spend their entire career trying to figure out what are good ways to extract features. Um, so the idea of a multilayer neural net is that you can, you can think of the first layers are as the first two layers as extracting features for the following layers to use, um, and classify if it's a classifier.

YANN: [21:36](#)

Um, and if you can train the entire thing end to end, that means the system learns its own features. You don't have to engineer the features anymore, you know, they just emerge from the learning process. So that, that, that's what was really appealing to me and, and the idea that, um, is necessarily some sort of hierarchical structure in those features. Uh, and the reason why you need some sort of hierarchy is because the perceptual world, like the natural data is compositional in the sense that, uh, you know, pixels kind of assemble to form edges for example, then edges assemble to form motifs like corners and crosses and things like this, uh, [inaudible]. And then those motifs assemble to form kind of more complex shapes like circles and squares and those assemble to form parts of objects and those assemble to form objects, et cetera.

YANN: [22:42](#)

So we have this sort of natural compositional hierarchy and it's the same in speech. Uh, you have, you know, raw signal and then phones, phonemes, words, sentences, etc. You have the same in text. Just about any natural language. We have this sort of composition, compositional hierarchy because the world is compositional.

CRAIG: [22:48](#)

Um, and the visual cortex also has these layers. Is that right?

YANN: [22:55](#)

That's right. Yeah. So, I mean there is, anatomical layers and there's functional layers. So, we're here, we're talking about the functional layers, right? So, the, the, the visual signal goes from your retina to your piece of the brain at the bottom called LGN and then it goes to the back of the brain, V1, V2, V4, IT, and IT in the temporal cortex is where objects are encoded object categories are encoded. And you know, some neurons will fire

when you look at a chair, regardless of what chair it is, if it's occluded or not, if, you know, the type or orientation, what color, what - it doesn't matter. Right? So that's called [invariant representation](#).

CRAIG: [23:26](#)

But there is a hierarchy of components, right? Yeah. It starts there, there are neurons that fire when it, when it sees a cross or when it sees an edge, and then that gradually is built up.

YANN: [23:38](#)

So this is, it's called the [ventral pathway hierarchy](#), right? V1, V2, V4, IT. These are the four big, uh, you know, visual cortex areas that are used for, uh, kind of recognition of objects in the, in the visual field. That's only five layers. I mean, there's more kind of internal layers. Um, and so, the next question I asked myself very early on, uh, before even I got to Toronto when I was finishing my PhD is can we build a network whose architecture would be somewhat inspired by the, what we know of the visual Cortex? There was very classical [work from the 60s](#) by [Hubel](#) and [Wiesel](#) on the architecture of the visual cortex, right? Simple cells, complex cells, et cetera. Um, and, uh, it was a very natural idea and which people already had in the 60s of, uh, connecting neurons to a small area, individual fields so the technical features and things like this, right?

YANN: [24:34](#)

Um, and I built neural nets like this before even I came up with backprop, you know, tried to kind of reproduce this kind of architecture with the crude software tools that were available. Um, so what I set out to do, when I got in Toronto was, I started a project of an ambitious project of writing a neural net simulator with Leon Bottou, whom I met just before I left France. And we, we started, uh, writing a neural net simulator together called [SN](#), which turned out to be very instrumental in allowing us to do the experiments with early convolutional nets and things like this. Um, you need, you know, at the time you need a, you needed a lot of investment in software to be able to do those things. There was no [MatLab](#), there was no [Python](#), there was no, you know, you basically had to write everything yourself.

YANN: [25:19](#)

Right. We even wrote our own, we even had to write our own [lisp interpreter](#). So, um, so anyway, uh, I, I got to Toronto, I worked on the software. Leon was writing, you know, keeps working on the software on his, on his side as well. Um, and then, you know, finally I can try, um, convolutional nets. Um, and, and it's based on the idea that I was inspired by, you know, papers by [Kunihiko Fukushima](#) on the [neocognitron](#) where they tried to kind of build also a sort of model of the cortex using those simple cell-complex cell hierarchy idea. Um, and this model was a little overly complicated. It was, it was trying to stick to, to kind of the

neuroscience and the biology. Uh, and uh, um, it's a lot of, you know, things to adjust in this model to make it work.

- YANN: [26:12](#) It was a little Byzantine in many ways. It didn't have backprop so we couldn't train it end to end. Um, we had to come up with some sort of unsupervised learning algorithm for it. Um, so what I set out to do was basically, you know, built, um, one of those kind of visual hierarchical model inspired by Hubel and Wiesel, that turned into backprop. That was convolutional nets. So, this started working in the spring of 1988 when I was still in Toronto, did some early experiments there. And then I moved to Bell Labs. And at Bell Labs they had a big dataset of - a big dataset of 9,000 training samples of, you know, zip code digits - and I try to, you know, my code was ready. I just tried it on the dataset and within two months I had, you know, better results than anybody else.
- CRAIG: [26:58](#) Wow. So, so, so, uh, neural nets learn the representation so you don't have to prepare the representations and then put them in as inputs.
- YANN: [27:09](#) Right. So, because of this multilayer structure and in the case of convolutional nets because of the, the local nature of those, uh, neurons, they only look at a small thing. Then it exploits this, uh, compositional nature of, uh, natural signals if you want. Um, and you know, a lot of it was intuition. Some of it was a little bit of a biological, uh, inspiration. Uh, uh, of course the whole idea of convolution and putting in things like this are very classical things in signal processing, you know, [inaudible] learning. Right. Um, and since then there's been theoretical work that kind of show that this kind of architecture is a good idea for some types of, uh, of, of signals you can, you can prove it. Okay. But, but back then, you know, it was more kind of [inaudible].
- YANN: [27:58](#) There is a limit to what you can apply deep learning to today due to the fact that you need a lot of labeled data to train them. And so, it's only economically feasible when you can collect that data and you can actually label it properly. Uh, and that's only true for a relatively small number of applications.
- CRAIG: [28:21](#) Mmm.
- YANN: [28:21](#) So that's one mode of training, right, supervised learning. It works great for, you know, categorizing objects and images for translating from one language to another if you have lots of parallel text. Uh, you know, it works great for speech recognition if you have collected enough data. Um, but it doesn't work for all kinds of stuff. Like it doesn't work for translating every language into every other language because we don't have parallel text for every language, right, every pair of language. It's

very important for Facebook. People use, you know, thousands of languages on Facebook and we don't have parallel data for every pair of language.

CRAIG: [28:55](#)

Mmm.

YANN: [28:57](#)

It's very important also for a lot of areas where it's very expensive to collect data, like medical images for example. Um, you will never have enough data for, and then there is a lot of situations where collecting data is just not the right thing. So, or, or is not sufficient. For example, if you want to train a system to hold a dialogue with someone, you cannot just collect the training set and train the system to hold the data. You actually have to train it to with people like talking with people, right? Um, if you want to train a system to interact with an environment, you have to have an environment in which it can train itself to interact. So that's one problem. The second problem is there's a second type of learning called reinforcement learning, which has gathered a lot of press, you know,

CRAIG: [29:42](#)

I met [Richard Sutton](#) at, uh, in Montreal

YANN: [29:45](#)

Richard is one of the founders of this area. And, uh, it's sort of a weaker form of learning in the sense that, uh, you can rely on the, instead of telling the system here is the correct answer, you only tell it, tell the system you are right or you're wrong or you give it to a number that corresponds to how right or wrong you think it is. Um, and that number, you know, it can be generated automatically by the environment. So, for example, you know, you want to learn to ride a bike if you, if you fall, that's the negative reinforcement. If you keep riding the bike for another second, that's a small positive, uh, uh, reward, if you want, right? So, by trying to figure out the sequence of action that maximizes the reward, then you know, maybe you'll, you'll learn to do, to ride a bike. Um, here's the problem, though: any human is, almost any human, is capable of learning to drive a car in about 30 hours of training with hardly any supervision.

YANN: [30:40](#)

If you were to use reinforcement learning, at least in its current form, to get a car to drive itself, it would have to crash thousands of times. It would have to drive hundreds of thousands of hours if not millions. Yeah. Crash dozens of times, kill many pedestrians, destroy itself, multiple times run off cliffs multiple times, before it figures out how not to do it. Yeah. Um, and so what that tells you is that, um, we're missing something really essential in human and animal learning. Uh, that is not reflected in the type of reinforcement learning or supervised learning that our machines can do. Right. Right. A kid can learn, uh, you know, the, the, the meaning of ten new words per day,

can figure out what an elephant is, with just two pictures. Right. We can do this to some extent with learning today using transfer learning, pretraining the machine to with lots of images and then you can retrain it to recognize objects with very few samples.

MUSIC INTERLUDE

YANN:

[31:43](#)

But um, but there is something we've been missing and one hypothesis that I have and you know, Yoshua agrees and Geoff has been saying this for 30 years or more, is that that thing should be unsupervised learning, um, which means just learn how the world works. Uh, just learn the dependencies, the structure, the regularity of the world by observing it. Um, so I have a form of it called [self-supervised learning](#), which is a very natural idea. Um, imagine that you, you give the machine a piece of input. Let's imagine it's a video clip, for example. You mask a piece of the video clip and you ask the machine, 'pretend you don't know this and you know, try to predict what is masked from what you're seeing. So, predict the future of this video clip, what's going to happen in that video from what you can see from the past' or uh, or 'here's an image I'm going to block piece of it, now can you reconstruct that piece?' Um, in the context of text, you give it a, a window of, I dunno, a dozen words on a, on a text and you take out 20% of the words and you ask the system, 'can you predict what words are missing?'

And so when the machine trains itself to do this kind of filling in the blanks, it has to develop some representation of the data so it can do this job, you know, so to be able to predict what's going to happen in the video, you kind of have to understand, you know, that there are objects that move independently of backgrounds and the objects that are animate rather than the inanimate. The inanimate objects have predictable trajectories. The other ones don't. Right, things like that. Right. Um, and so presumably by training a system to predict or filling in the blanks, it's, it's going to have to understand a lot about the structure of the world.

YANN:

[33:28](#)

And so the idea is that you would train a system in a self-supervised manner with tons and tons of data. Uh, there's no limit to how many YouTube videos you can make the machine watch. Uh, it will distill some representation of the world out of this. And then what you would do is when, whenever a particular task comes in, like learning to drive a car or recognizing particular objects, you use that representation as input to a classifier and you train that classifier, supervised. So that's the, that's the whole idea, and in fact, this is an idea that, uh, Geoff, Yoshua and I actually started with, uh, when we, when we got together to kind of restart the, to start the deep

learning conspiracy, you know, around 2003, 2004. The idea was to use unsupervised learning to pre-train a network and then fine tune it using supervised learning because we had this idea that it was very difficult and perhaps hopeless to train a very, very large, very deep network using backprop. It wouldn't work.

YANN: [34:33](#)

So the idea was we would pre-train it using those kinds of unsupervised methods. And so, Geoff worked on Boltzmann machines, Yoshua and I worked on Boltzmann machines and using autoencoders and various other things and I worked on [sparse autoencoders](#), these sort of various methods that we proposed to do this. Until we realized with all the, the, the tweaks that we developed in the process, uh, like, you know, rectify nonlinearities and drop out and things like that, that in fact you could train very deep, very large neural nets, um, um, with backprop from scratch. If you had [GPUs](#). So, so the hope is that by, by training a system to, you know, in this kind of, uh, in this kind of way, the kind of representation that would be extracted will be sort of more complete, if you want, less degenerate, than the kind of representations that are learned when you just train a machine to, you know, for a particular task, right?

YANN: [35:27](#)

When you train a machine for a particular task, it just learns the features that are useful for that particular task. In fact, uh, one thing that that became clear, um, um, pretty, pretty quickly was that the best way to train a convolutional net is not to train it to distinguish one class from another, like with two classes, like for, to train a neural net to do, I don't know, pedestrian detection, right? So, you have images with a pedestrian and images without. That's just a two-class problem. Um, it doesn't work that well. It works okay. You can beat records actually, uh, as students did this back in the, um, like around 2010. But, uh, but it's not ideal. It's much better if you train the machine to uh, categorize lots and lots of categories. The more categories you ask it to classify, the better the representations, the more robust they are,

CRAIG: [36:17](#)

The more general.

YANN: [36:18](#)

The more general they are.

YANN: [36:19](#)

Yeah. And so ultimately, you want it to just encode the image, right? Don't try to classify, just, just tell me like what are the useful, relevant pieces of information in the image that will allow you to reconstruct that image. Maybe with a little loss of details, but you know, most of the information will be encoded. So that's the idea of an auto encoder, right? You have a neural net that takes an image, but you see some sort of

representation of that image and then tries to reconstruct the image from the representation, that's an auto encoder.

Um, so let, let, let's take the example of video production, right? So, you, you give the machine a video clip and you ask it, 'what's going to happen next?' Uh, and it cannot possibly predict exactly what's going to happen next. Because you know, if it's a picture of someone talking, that person can say a word or another, can move, uh, the head in one direction or the other and the system has no way to predict what's gonna, what's gonna happen. The example I use very often is, uh, let's say you have a video clip where you know, someone takes a, takes a pen and puts it on the table and you let the pen go, you know the pen is going to fall, but you can't really predict in which direction. So, if you use, if you train a neural net to minimize the distance between the predicted image and the image that it's observed or the frame, you know, in the video, it cannot do a good job. It will have to predict the average of all the possible futures. And that ends up being a blurry image. Um, it's a, you know, a super position of, of me moving my head to the left and to the right because it doesn't know if I'm going to move to the left or the right, right?

YANN:

[37:49](#)

Uh, so you get blurry predictions. So, one way to, uh, uh, get around this problem is you, you, you, you have an extra variable that you draw randomly. It's called a latent variable. You draw it randomly and depending on the value that you draw, it's not a single variable, it's going to be a vector, right? So, depending on which values you draw, the prediction is going to change. And now the, the game, the name of the game now is to, is to train that machine to make predictions. Uh, so that, you know, as you draw different values of this latent variable, the predictions basically go through all the possible futures in the video. Right? Um, and the problem with this is that how can you tell the machine whether its prediction is good or not. To do that you have to train a second neural net and that neural net is trained to tell the difference between a good prediction and a bad prediction, that's called the discriminator or quick, uh, and [adversarial generative neural network](#) is the idea of training those two networks together. Essentially against each other, basically.

CRAIG:

[38:55](#)

And uh, when you're, when you're talking about learning representations, for example, from watching, uh, millions or hundreds of millions of videos, I asked [Pieter Abbeel](#) this, where is that, um, knowledge that learning stored? Is it simply in the weights of the network?

YANN:

[39:17](#)

Yeah. Yeah.

CRAIG: [39:18](#) So you're talking about very big networks.

YANN: [39:22](#) It could be very, very big networks.

CRAIG: [39:23](#) What's the biggest network that you've worked on in this representation learning?

YANN: [39:29](#) You know, there is, there's two numbers, right. In a, in a neural net or there is a number of different things you can, you know, and how you can describe.

CRAIG: [39:35](#) The layers and the numbers of ...

YANN: [39:38](#) One is how many layers. One is, you know, how many neurons per layer and what's the pattern of connection. And the other one is how many [free parameters](#) that are, like how many tunable parameters, right? Cause in convolutional nets you have one parameter controls, multiple connections. So, um, and you know, people in natural language, for example, you know, train routinely model, train models that have a billion parameters.

CRAIG: [39:53](#) Wow.

YANN: [40:00](#) Um, so those are pretty big networks. Yeah. Uh, conv nets, many of them have a relatively small number of parameters, like it's in the tens of millions. You know, it's, it's amazing to say now that it's actually a small number of parameters because, um, if I project myself back 30 years ago, you know, a big network had, you know, 60,000 parameters. Um, but, uh, but it could be extremely large because, uh, sometimes you want to apply, you want to apply the convolutional net on a large image at high resolution so it can detect small objects anywhere in the image. And so, the overall size of the network is gigantic. In fact, it's, you know, can be tens, tens of billions of operations or even hundreds of billions.

CRAIG: [40:42](#) Yeah. Um, talking again, the Pieter Abbeel, um, he was saying, yes, the, the learning is stored in the weights, but there are also systems to store, uh, experience in databases and, so that the system can refer - in in effect it's like a memory.

YANN: [41:07](#) Yeah. Well. So that's actually a very interesting topic of, uh, research now for a lot of people, which is, uh, basically, uh, uh, sort of augmenting neural nets with some sort of working memory. Um, so if you want a neural net to do just perception, right? Perception is sort of very sort of reactive thing. You know, you give an image, you go through a bunch of layers and you get the answer. Um, but a lot of tasks that we like machines to do involve reasoning or, or, or even

CRAIG: [41:37](#) decision making, right?

YANN: [41:38](#) Right. Um, well, everything is decision making. Yeah. But, you know, several steps of reasoning, uh, referring to past events, um, um, you know, things like that, like having a working memory, right, um, where you can hold facts and things like that.

YANN: [41:54](#) So, you know, if I tell you a story, uh, and in fact this is actually kind of a scenario that people here at Facebook, have built several were years ago - the story is, uh, uh, John goes to the kitchen, John picks up the milk, uh, John goes to the den, uh, John drops the milk. Uh, now John goes to the kitchen, uh, Jane goes to the den and she picks up the milk. Uh, then she goes to the backyard and drops the milk. Where is the milk? Okay. So, you know, you have to, or you know how many people in the kitchen, right? So, you know, you can listen to a story, you kind of maintain a state of the world in your, your memory, which you have to keep somewhere. Right? Uh, and then someone asks you a question and you have to kind of answer that question. So, you have to kind of figure out what's the state of the world. Uh, you know, where's the milk is easy to remember because I just told you that a Jane, you know, dropped it. But, like, how many people are in the kitchen - you have to remember that John went back to the kitchen. Right, right.

Um, and so, um, so there's a dataset of this type that, uh, [Justin Weston](#), [Antoine Bordes](#) and a couple others here built a few years ago called the [bAbI tasks](#), which is exactly this kind of scenario. And they invented a particular type of neural net to solve this problem called a [memory network](#). So, it's basically a neural net, recurrent neural net. Yeah. And next to it is, is a memory. Um, but that memory is itself a neural net. It's a, it's a special kind of neural net which is designed to kind of store data and, and, and retrieve it basically very in this, you can think of it as particular architecture of a neural net. Uh, and every time step, when, um, you know, the, the neural net, can ask a question to the memory, sends a query to the memory, gets an answer back and then as a function the answer asks something else of the memory and gets the answer back, et Cetera. And so now you can have a network, that can do a chain of reasoning.

Um, and then people have built on this idea a lot over the last few years. The latest models that work best in a natural language understanding are transformer networks and the transformer network is sort of a network in which groups of neurons inside the network are basically those memory modules. And they're very similar to those memory modules. So I think there's a, there's a lot of hope that, uh, we're going to make progress in AI because of ideas of this type.

MUSIC INTERLUDE

- CRAIG: [44:32](#) Uh, you, you use this, this analogy, uh, several times. I don't know when you first used it - I saw it in Long Beach - [gateau genoise](#), the unsupervised learning is the cake and the, the supervised is the icing and the reinforcement is the cherry. Uh, so, uh, are there systems that you've worked on that have, have had this chain, uh, of, or is the research still very discreet? People are working on unsupervised or people that are working on supervised and some people on the reinforcement.
- YANN: [45:20](#) So the thing that has come the closest, at least in my work, I mean there's a lot of people, uh, uh, at, uh, Berkeley for example, at Stanford, like [Sergey Levine](#), [Chelsea Finn](#), Pieter Abbeel to some extent, right? And a few other people, at DeepMind and Facebook who, who've worked on, uh, what's called [model-based reinforcement learning](#) systems. Right. And it's not a new idea. People have, you know, have had this idea for a long time where the system has kind of a predictive model of the world, um, which allows it to predict, for example, you know, if I'm driving a car and next to a cliff, I turned the wheel to the right, I'm going to run off a cliff and nothing good's going to happen, right? I don't need to actually try. I know. I have a model of the physics in my, in my head that tells me this is bad.
- YANN: [46:05](#) Um, and so there's quite a lot of activity now on model-based reinforcement learning. And it, it, it didn't happen too much in the past for the same reason that people resisted using deep learning for a long time, which is that the theory doesn't work. So, the theory tells you, you know, there's a proof that model-free reinforcement learning, will converge in certain conditions. Right? Um, there's no such proof for model-based and experimentally if don't do it right, model-based reinforcement learning learns faster, but it doesn't work as well as model-free. Uh, and so that caused people in the mid-nineties, at the same time that people abandoned neural nets for simpler models, they also abandoned model-based reinforcement learning for model-free. Um, uh, and this, there's this joke about like, it's, um, it's a popular joke in, in France, you know, the, the French like mathematics, right?
- YANN: [46:59](#) And the joke is um, yeah, yeah, yeah, it works in practice, but does it actually work in theory? And you know, the whole community essentially had this attitude that, uh, yeah, it works in practice, but we don't understand why. We don't think we have good theoretical ways to understand why, so we're just not going to work on it anymore. In my opinion, this is, you know, looking for your lost car keys under the street light, even though you lost it someplace else. Um, so you know the joke?

CRAIG: [47:25](#) Yeah.

YANN: [47:27](#) So, uh, so this model basically reinforcement learning I think is becoming really interesting, particularly for people who work on robotics because it's very hard to train a robotic system in simulation for things like grasping because simulators are not very good. Um, um, so there's a lot of interest. My own work. Uh, I try to stay away from reinforcement learning. I like, uh, because I like the efficiency of gradient-based learning and reinforcement learning, you know, basically you can't use gradients cause you can't estimate the gradient of the objective function. You can estimate it, but you can't compute it directly. So, I've done things like our [latest paper](#) at ICLR, um, just a month ago was, was about the idea of uh, training, uh, a predictive model of what cars around you are going to do. Um, so it can run this model, uh, for several steps, maybe several seconds. And it will predict what the cars around you are going to do. And of course, you know, you can't exactly predict. So, there is some, you know, some variable you can draw and it will predict multiple futures. There's a cost function you can compute, which is how far you are from the other cars, whether they're going to bump into you, whether you are in your lane or not, you know, whether you are going at the speed you want, you know, various costs like this.

YANN: [48:46](#) And um, and what you can do is train a neural net to, uh, produce the correct, the best, uh, steering policy and braking-accelerating policy so as to minimize the likelihood of, of collision, uh, by just minimizing these costs. And so, you're using this model to predict what the future is gonna, is gonna do. You have a cost function that's differentiable and you just train a neural net to, you know, optimize this objective, there's no reinforcement learning. It does the same thing that a lot of people try to do with reinforcement learning, but because the cost function is differentiable, there's no need for reinforcement training.

CRAIG: [49:27](#) There's certainly generalization going on, but uh, it's, it's generalization that's fairly narrow. I mean, because it's generalization within one data set.

YANN: [49:40](#) That's right.

CRAIG: [49:41](#) You have the training data and then you pull out a test data and it generalizes from the training to the test data. But if you use a different data set, it usually doesn't work. So there, there, that's one question is, is how do you get to that generalization? And then the other question is, uh, is there real transfer learning going on anywhere where the learning that's stored in the, in the weights can actually be applied to a new problem?

YANN: [50:15](#) Well, so, um, I mean transfer learning works, right? So, if you have a big dataset, regardless of what the task is. For image recognition, you pre-train on this dataset, and then you fine tune on whatever data you have, which may be smaller, right? So, this kind of stuff works. You still need data for the, the second problem. One thing that people are trying to do is um, uh, is sort of [multitask learning](#). So, you train a neural net on multiple datasets and you hope to get some sort of more generic, um, image recognizer or whatever it is. And then it's easier to specialize it for a particular task because it's learned already a lot of different, uh, tasks. So that's one thing that Facebook has been doing, for example, where you take, um, I don't know, 4 billion images from Instagram, um, where people, you know, when they post a picture on Instagram, they, they put Hashtags.

CRAIG: [51:04](#) Right.

YANN: [51:05](#) And so what the Facebook people did was, um, select 17,000 most frequent hashtags that correspond to actual objects, physical concepts, uh, and then train a neural net on those 4 billion images to predict which of the 70,000 hashtags are present. And you know, it does a pretty terrible job at it, but, but it learns to represent images in such a way that it can do that prediction as well as you can. Then you chop off the last layer and you fine tune the, the network on [ImageNet](#), [COCO](#), or whatever tasks that you're interested in. And you can beat records this way. Um, uh, this was a paper published last year.

And so that, that's a sort of edging towards kind of almost unsupervised learning in a sense that the data is not carefully curated, which is whatever people type for hashtags. Um, but ultimately what you want to do is self-supervised learning. So, you know, um, I'm not giving you hashtags. I'm just, you know, here are pictures. You can have as many billions as you want and encode pictures, um, in such a way that the features that would be elaborated by the system then will be useful for any tasks, any vision tasks, that you can imagine. That's the, that's the challenge for the, of self-supervised learning.

CRAIG: [52:27](#) Ultimately the goal is to knit together all these different techniques and strategies into general artificial intelligence. Is that something that you stay away from or do you have an opinion about?

YANN: [52:45](#) Okay, well, I have a lot of opinions on this. Yes. Uh, okay. First of all, I don't like the term AGI, artificial general intelligence.

CRAIG: [52:53](#) I've been corrected on that point before.

YANN: [52:55](#) Uh, because human intelligence is actually very specialized. We like to think of ourselves as being generally intelligent. We're not, we're very specialized machines. Um, so AGI is a misnomer. Um, human level intelligence, that's a, that's a better question to ask. So, can, you know, can we build with machines at some point that will be as intelligent as humans in all the tasks that humans are intelligent. Um, and the answer is, of course, there's no question. It's a question, it's a matter of time. Um, and, and it's very important to make progress in that direction because we'd like to have machines that have some level of common sense because we'd like to be able to build, you know, virtual assistants that help people in their daily lives. Um, can answer any questions you have, you know, can kind of manage your interaction with the digital world and with each other. Um, uh, so that, you know, that would be kind of transformative in terms of, uh, the {inaudible} that's available. We'd like, uh, image recognition systems that don't get easily fooled.

YANN: [54:05](#) We'd like self-driving cars that are very robust and, you know, that understand how the world works and that, you know, they make the right decisions when they see unusual situations. Um, so that's a really important question for practical reasons. Um, and then the question is, you know, how is it that, um, you know, the best of our AI systems have less common sense than a house cat or actually a rat, you know, [Washington Square Park rat](#). There is something that, you know, animals, some learning process that animals have, uh, access to, to acquire all the knowledge they have about the world that we don't have in our machines. So, one hypothesis, or my money is on things like self-supervised learning, but you know, there might be other, uh, other favorite approaches from other people.

YANN: [54:55](#) Uh, so, so it's a very important problem to solve, um, for machines to learn by observation, uh, run without requiring too many labeled samples, uh, perhaps accumulate enough background knowledge by observation that some sort of common sense will emerge. Um, and we'll have, you know, not just intelligent virtual assistants, we will have dexterous robots, you know, you know, the household robots that everybody has been dreaming of, right. We don't have the technology today. So yeah, that's a, you know, very, very intriguing question. There's a lot of people at Facebook working on this.

CRAIG: [55:31](#) A question, but one that you're optimistic can be answered.

YANN: [55:34](#) Well, yeah, I mean the, there's no question that it can be answered. It's a matter of how much, you know, how long is it gonna take and how is it going to be done.

CRAIG:

[55:45](#)

That's it for this week's podcast. I want to thank Yann for his generosity. For those of you who want to see a video of this interview, visit eye-on.ai. You can also go into greater depth about the things we talked about today by downloading a transcript of this show from the site. I've inserted links and the transcript to make it easier to follow some of things Yann talked about. We've been getting good feedback from listeners and I hope to hear from more of you. You could help us a lot by rating and reviewing the podcast on whatever platform you use to find us. Let us know whether you find the podcast interesting or useful and whether you have any suggestions about how we can improve.

The singularity may not be near, but AI is about to change your world. So, pay attention.