RCAC-Deep Learning Series-20250305_123943-Meeting Recording

March 5, 2025, 5:39PM

1h 13m 1s

Elham jebalbarezi sarbijan 2:02

OK.

Mm hmm.

OK.

Yeah. So this is first session.

That we want to talk about deep learning, but we maybe a new approach of checking deep learning and what has been done from 1950 when AI started 1st until now we go sometimes to the technical side, but sometimes to the philosophy of deep learning and all the.

Challenges that has been solved during all these several decades.

I will cover in this session I I think.

Yeah, I was here.

This session is more about history of deep learning and if we have time I'll cover something about traditional deep neural networks.

I can call them traditional because nowadays everyone is talking about Transformers.

I'll talk about CNNRNN auto encoders.

Attention, when everyone was excited about deep learning just for future engineering and representation learning, and then later we moved to generative AI.

Maybe next session we talk about discriminative versus generative AI and how we thought of moving from discriminative to generative AI and generative deep learning.

Then we go to details of Transformers and what happened over there and then we expanded to different domains, text, image, video and speech.

Then later we can talk about LLM in practice.

All the tricks that we have to be careful about when we are using LLM for different applications. We can have a session we plan to have a session if audience like it to talk about AI safety and governance and all issues that we have with AI with L.

Mainly these days and then an idea is just to cover AI and deep learning for different domains and different departments. We can have seminars on how different departments are using deep learning and what are the challenges they have over there.

For today I'll go mainly to history and basics of deep learning.

We had two AI winters already.

In previous decades, we want to know that if we are close to another aiv and actually what is an AI Winther we go start a story of AI and deep learning from 1950 to now how we moved from single neurons to deep neural networks all the remain.

Challenges and then all the challenges that have been solved from 1950 to now because it's a good question that why we had neural networks and even deep neural networks from 1950.

But it was not famous until after 2010, mainly.

And then if we have time in this session, I'll talk about CNNRNN and attention models in traditional form before Transformers and also about deep learning for feature extraction, feature engineering and representation learning. OK.

So first, with some definitions. Everyone maybe is like a bit confused about AI machine learning and deep learning.

AI is a general term for whenever we try to solve our problem by somehow mimicking human intelligence to solve some problem.

But AI always doesn't need training.

A model doesn't need to extract patterns from data, doesn't need supervised, unsupervised semi supervised learning.

So whenever we are just have mothers rule based models.

Expert system, evolutionary computation and others. They are kind of AI but not machine learning.

Machine learning is a subgroup of AI which cares a lot about analysis of data.

We try to extract patterns of data and then try to map our input to output input feature to output feature.

And we heard a lot about learning supervised, unsupervised, reinforcement learning, and all of them, which are kind of machine learning and then moving to deep learning, which is a subgroup of machine learning.

We work with neural networks and we make them deep to solve.

Many problems in this session. We go through deep neural networks from 1950 today so.

This picture is very interesting to me.

It shows that how AI has been changing from 1950 until now.

I'll explain later that 1954 first time what AI was introduced in a conference and then it was in height until around 1970, which we had first AI winter and then again in hives until around 1980s at the end, which we had second AI met.

And now AI in the best version of itself during all the history from 1950 until now.

And everyone is curious to know what happens next if it will stay in this huge investment and funding and everything cares about AI or we will have another AI winter in next decades or even next years.

So now I want to go talk about this AI, Winther and what happened then? So 1956 for first time in a workshop.

Minsky, which was a professor in MIT, use word, artificial intelligence, and everyone was so optimistic about AI they thought that in less than one generation we have artificial intelligence in level of human intelligence.

So lots of fundings arrived there, mainly from DARPA for many, many different applications, but later at end of 1960s, they found that.

They are not getting as much as they hoped for from mainly from neural networks then.

So DARPA just reduced funding for a while, but at the end 1974 first AI winter happened.

Then they found that AI research is unlikely to produce anything truly useful in near future.

Maybe later, but not now.

So the excitement disappeared then.

So if I want to go more to the details of this first AI winter.

Some researchers said that if we want to have artificial intelligence, we have to mimic human brain.

So they try to understand how a neurons in brain work, and everyone AI researchers, they were trying to provide a simple mathematical equation to simulate human brain neurons.

So the 1st.

Very famous version of human neuron mathematical version is this one which they said that if we get lots of input signals and just check their rates based on the importance and sum them up and pass them through an activation function which first version was just a step fun.

The output will be active or inactive and this is exactly what a neuron does in our brain.

So based on some input it decide if it can be active output or inactive so.

And then first perceptron model was proposed by ROSENBAUGH in 1958. This model was so simple, this model is not differentiable.
We cannot calculate gradient.
No one cared about optimization or gradient.
The only thing was a neuron which in output it has a step function and by tweaking the weights in the first layer you decide if you want to achieve your output or not.
This is perceptron learning rule.
As I said, it doesn't have guardian it.
It's not differentiable and it's a very simple rule.
Which manually you try to update your neurons to to predict your output and then 1969 the first AI winter was closed because Minsky that MIT professor reached first time introduced artificial intelligence for said that despite claim by Rosenbaugh.
This perceptron layers cannot solve any function.
That claim was so huge, he said, that if you get just X or function, you will see that X or needs at least two neurons or two layers.
To be classified because X or is a non linear function, so you cannot claim that with one layer of neurons you can solve any problem.
And then this is first AI winter.
So if I want to put it in one page, rosenbaugh introduced perceptron and he said this person can then recognize patterns and lead to AI and AI close to human intelligence.
Minsky, 1969 said that no, it cannot even do like XOR layers.
Actually, what they forgot over there was like, how about several layers of neurons instead of just talking about one layer? But technology was not ready because the model was not differentiable. And even if we.
Stack two layers of neural networks to each other for X or function.
We cannot train it. So then the first AI happened and until 1986, when people invented back propagation so we didn't have back propagation until 1968, and then about second AI winter when people were so disappointed about neurons and neural networks, they moved to non.
Machine learning AI. They use expert systems to just make something like computers like human brain and what happened several years later after many investment. They found that these systems can make mistakes.
And they are very difficult to update.
They are very expensive and they are unable to learn very well. So then everyone was

disappointed about expert system also and 2nd aiv Inter happened and what is interesting here to me is again Minsky said that in his book three years before 2nd AI Winter he predicted the.

Second AI Winter and he said that.

These systems, because of difficulty in a scaling and achieving true intelligence.

This is an overhyped expectation from AI and it doesn't work.

So the second AI winter happened in 1990s, and there was no funding and interest available for AI.

OK.

So if I want to make a conclusion for AI winters, we can see that AI winters happen when there is a huge hike.

And there is huge investment when everyone excited with advertisement expectations are so high and meeting the expectations is hard.

So that's one of the reasons that.

AI winters happen.

The second reason is about economy situation. If there is not funding available, they will cut funding for research mainly and of course AI and the and also if projects fail to deliver what they promised on time. This is also another reason that funding is not available for rese.

Research, including AI and of course, if there is not funding, no one invest on hard AI problem nowadays working on AI is almost easy, but all the past years all the researchers who work on fixing issues for deep learning, they made it to today. So if there is.

No investment for core research and there is no payment for PhD students.

So we are more close to the winters for science.

Now I want to go through this table which is again very interesting for me and I hope that is interesting for you as well about, OK.

When first time in history we started neural networks.

So we know that the first person who tried to understand human to try to understand human brain was Aristotle.

And then later, other researchers they tried to mimic human brain and make mathematical equations.

And it was ongoing and nothing highlighted happens by then.

Until you 1956, which I talked before the first time they introduced word AI in a conference by Minsky and then.

1958 first perceptron, as I mentioned before was introduced by Rosenbaugh and it's very similar to the neurons that we are using today, though it was not differentiable and it was very limited. 1969 the first AI Vinter happened because of the problem of deep NE.

Neural networks for XOR function and nonlinear function 1974. A first version of back propagation was introduced, but everyone was busy with non machine learning.

AI all the funding was on the other side, so no one cared that much about this back propagation and it was not very famous 1984 first time someone introduced new cognition, which is further off today's convolutional neural networks.

So we had CN, NS from 1980.

But researchers were busy on the other side of AI, actually.

And then 1982, Minsky predicted the second winter in his book, and he said that the current AI models and expert systems are not powerful or not general, and are very expensive.

And then 1986 was time to get back to neural networks after they fail in Rule based system and non machine learning AI everyone was moved back again to neural networks 1986.

The prosperous time for neural network started.

At first they introduced recurrent neural networks, then later the backpropagation version we know today is introduced 1986 by and itself.

The all the claims against.

Xor and non linear function which was made for neural network Yan Li Con which I think everyone knows today and.

Yang likan.

Hinton and Benji all together got that touring award for deep learning he introduced at 1989, CN NS-400 and digital recognition, which actually is the time that CNN got so famous.

So CNN that we know today was introduced at 19891997.

Everyone was working to fix issues by RNN, which I'll explain later.

What are the issues and how they solve it?

So they introduced LSM.

To solve issues with recurrent neural network, but still we didn't have deep neural network really deep see, it was shallow neural network because of issues that I mentioned later.

But everything was much more in influence and hype. When NVIDIA first time in

1999 introduced their first GPU.
By then GPU were used only for gaming and just predicting pixels in like.
Graphic computer graphics and 2006 the real deep neural networks were introduced by Hinton.
He said that if there are many problems and we cannot train and back propagate for many reason in deep neural networks he he introduced layer wise pre training. So he introduced concept of pre training which today everyone is using for first time so.
What they did actually, I'll go in details of this one because it was a very important breakthrough for feature engineering and representation learning. And then 2006.
First time they use GP US for deep learning and that was the maybe beginning of the story that today we.
Have with supercomputing and AI.
2012 Geoffrey Hinton worked a lot on drop out for overfitting and generalization issue for deep neural networks.
Because even if we had deep neural network except this solution by Hinton, we didn't have any other solution to train deep neural networks and it was introduced at 2012 and then end of the story that I want to cover in here 2017 Transformers first introduced for.
To replace CNN and RNN, but mainly for text and the later 2020.
We had vision transformer for first time to replace convolutional neural networks for vision task and image processing.
OK.
Now if I want to conclude this story with from another aspect, we heard a lot about big data. I think 2015, if you didn't have the buzzword of big data in your proposal, you couldn't get any funding.
We have big data and then we have big machines, but let's see how they affect this AI and machine learning and deep neural network learning path.
Before 2000, we didn't have big data.
We didn't have big machine and he didn't have training methods for deep neural networks, so nothing was ready.
We had neural networks, we knew what is deep neural network, but we couldn't do anything with it.
The technology was not ready for it.
So then everyone if if someone worked then in machine learning we remember that everyone was using SVM, linear regression, logistic regression and decision trees.

Because if you don't have big machines and big data, then you can't get much better performance with these simple models than complicated nonlinear models.

Later in 2000, we had big machines and GP US and we had some training methods, but still training methods were not good enough.

I'll explain later why and then also we didn't have big data sets collected and saved and saved and processed very well, so.

People were working on RNN, CNN and auto encoder but not deep learning really.

And then 2010 and later.

All the learning methods were in a good shape.

Big data for extracted from all the social networks and everywhere was there big machine was there.

Now it's time for everyone to work on deep neural networks in a non expensive way and affordable for most of people.

And what they claim new works demonstrate that it's sufficient data and computational resource deep models could achieve a state-of-the-art performance in many complex tasks which lead to modern deep learning revolution.

And this picture also is interesting to me because it shows that if you don't have enough data, the performance is likely. And as you can see, performance for traditional machine learning is much better when we don't have enough data.

But if you have more data, now it's time to move to deep neural networks.

So that's why if we don't have good data and good machine, everyone just can use a simple SVM. Then moving to a deep neural networks.

OK.

Now I move a bit to definitions of deep neural networks and details of deep neural networks. As I explained, starting from 1958, they provided this perceptual model of human brain neurons.

We can see that all the input signals after applying some weight for checking the importance they get to that summation or the call it transfer function and then it goes to an activation function to decide if finally this neuron is an active neuron or is not an active.

Neuron.

And if you put lots of these neurons next to each other, and also if you put these neurons in layers, stacking to each other, you have deep neural networks.

So the number of neurons that they are next to each other in the same layer is big.

After deep neural network, a number of layers stacking to each other is depth of the neural network.

So these two together define complexity of a deep neural network.

OK.

So now we have deep neural networks, but let's see how is it different from the previous models. A deep neural network has a huge space because we know that all these connections between neurons are parameters that we have to train during our training using our data using our.

Big machine. So this is space Toby in previously we had linear regression and it was a linear surface there to train with a few parameters.

Now you have this surface to train. You can see that this surface which you have to train waves to minimize your loss function. It's very nonlinear, so it has many optimum points.

Many, many like minimum and maximum points and you don't know that which one is the best for you.

So here your optimization method is very important. When you start. It's very important if you initialize your model like here you get to this optimum point. If you start from here you get to this one.

So it's very important.

Where do you start your optimization?

You cannot naively use a random initialization like what we did previously with scvm and other linear regression models.

Now we can talk about deep neural networks with another language.

Previously we had linear regression, which is a simple linear model.

You learn West and B to separate class.

A and Class B and then later they tried to make it a bit non linear to have less error in their training.

So they made logistic regression, which is just an activation function over the linear function for separation.

Then we move to shallow neural networks, which you just a bit nonlinear and then later we move to deep neural networks.

So what we can say from another aspect, if we don't start the story from human brain, we can say that deep neural network is exactly.

Moving from machine learning linear models to machine learning, very complex models, you can see that here each of these Sigma is a non linear activation function.

So you are stacking lots of these non linear models together.

So it will result in a very nonlinear model, which can lead almost any complex function.
OK.
Now something about deep neural networks or deep neural networks. The heven of machine learning.
They don't have any problem. Any issues?
The answer is no.
There are lots of problem.
We know that deep neural networks are working well.
Performance is quite good.
Accuracy is quite good, but there are lots of problems we don't care about because solving them is not easy and actually.
It needs the like theory of AI and machine learning to solve them.
The first problem is about their complexity and nonlinearity. When a model is very complex and very non linear, then theoretical analysis of this model is not easy.
Previously like 10 years ago, 20 years ago, we just checked if a function is convex, then OK, I get to the opt.
Answer If it's non convex then I had to apply some other condition, but for deep neural networks function is so complex that we even cannot write it on the paper, let alone to find.
It's like to worry and it's convergent.
All the other problem, the other problem we have with deep neural networks. We know that neural networks can learn almost any complex function if it has enough read and enough depth. But we don't know theory of it.
We don't have any formulation of how this depth and how this bit is working for our model, how changing each of them change the final result.
So this is an ongoing study behind the scenes. The other problem we have it's like, as I said, dysfunction is very complex.
This this function is non converse non convex.
So we know already that a stochastic gradient descent you just set a function in Python And you get good result.
But we don't know why.
A stochastic gradient descent without any nice theories behind it is working so well.
And we don't know that how among all those options for minimum it gets us to one of them, which is good and acceptable.

But the other problem we have with deep neural networks is, as we know a lot about like large language model issues today is privacy and security. When your model is so big and you feed almost any data you see to it, you are not sure if later there.
Is a data leakage over there or not?
So if you use of data with different in different ways, it's a concern which I provide some example in next page and the other problem which is I think everyone heard about interpretability of deep neural networks, deep neural networks are called black box.
No one knows that we know output is code, but we don't know how this output is generated.
So if there is any bias inside them, if there is any problem in the output troubleshooting this problem, finding out where is the problem of model in terms of like finding some errors and biases is not easy like the previous models.
Just jumping for two page, two large language models as I mentioned about privacy in deep neural networks, we know that one of the huge problems today we have large language model is the risk in misinformation and disinformation, which you may hear in different paper.
Misinformation is then your large language model.
Generate wrong information unintentionally.
It makes mistakes.
But it's not abusing your data or anything.
Which one?
Very important subgroup of new information, this hallucination. There are lots of papers about hallucination.
Hallucination is when your neural networks uses its language fluency to make stories that are fake.
They are not real.
And it happened before that. It makes the stories about some famous people, which is really kind of, yeah, it's a very big issue with it.
And the other problem as I mentioned is this information, which is actually large language model abuse. For example, fake news generation and all the Internet boats and other things which some people are using.
So these are two huge problem with privacy of large language models at least.
And then two examples of safety, ethics and governance, which we need for AI, is like one problem that still it happens with chat gpta lot.

It's very big to find resources for the data it provides for you. Most of the time when you ask it to give you references for, it gives you references that they all run, and the other problem like when Google tried their their first Google AI system, if you.
Ask it about depression.
So it searched there for you and one suggestion was the best way to do suicide.
So OK.
Just one option is just jumping from Golden Gate, so they try to avoid all this problem before releasing all these Google AI systems after that.
OK, now I want to move a bit to the technical side of deep neural networks.
And to explain why a story of deep neural networks takes several decades, from 1950 to 2010, which deep learning was very famous, and everyone could use it.
So one very famous problem of deep neural networks is when they get deep, they cannot update gradient fail, they cannot train well. We call it vanishing and exploding gradient, which I'll explain the problem and solutions. And then we had problem with activation functions. The function I said it.
At the end, if Neuron is active or not active, so how?
What was the problem?
How it was solved and the other problem is model overfitting for deep neural networks.
We know that all complex models are prone to overfitting in machine learning.
So if you make a model very complex, you have to be careful about overfitting.
So how this problem happened and how they solved it is also another thing we want to discuss about.
OK, at first I move to gradient descent.
We had gradient descent for many years.
Gradient descent is when you calculate gradient of your model over the entire data set and then you update your parameters using this gradient.
It is very slow and it needs a huge memory. If you have many parameters and if you have a big data set later I tell the year in next page they invented a stochastic gradient descent.
It's very simple, but it made deep neural network possible. Instead of calculating gradient for all your data.
We calculate gradient on data batch by batch. So random partitions of our data and we do this update little by little instead of updating your model on all of your data set at once.

So this page has lots of text, but it's actually provides a very big picture of what happened in 1951, was first time that people used a stochastic gradient descent, but. And research on a stochastic gradient descent was ongoing until 2018 and still is ongoing.

So what did they try to fix during all these years?

The first benefit of a stochastic gradient descent is that stochastic gradient descent is random based on order of the batches and everything, so it avoids local minima as I showed in that picture with lots of non linearities.

We don't know how, but a stochastic gradient gradient descent helps to get rid of all these local minima and improve generalization.

It jumps over bad local minimas and gets you to a good point.

So your model will be more generalizable. The second benefit of a stochastic gradient descent is scale and speed.

As I said, if you want to calculate gradient over all the data, it will be very slow.

It needs a huge memory, but this one needs it's very fast and doesn't fill up all of your memory.

The third benefit is like today we are so happy with deep learning.

Why? Because I train it on one task and then fine tuning on another task.

We call it like continual learning or online learning.

It doesn't need you to get all the data from scratch and update your gradient.

Get the new batch of your data which you care about in your new task and update the gradient so you just update your model on a portion of data.

This is something that is the one of the main important points about deep neural networks, which everyone is enjoying it today and the last benefit which actually the the last part of research which still is ongoing, this stochastic gradient descent lets us to play a lot with Lear.

Rate and the speed and how we want to check all parameters and help us to do lots of research during past year to find different versions of a stochastic gradient descent which converts to better Optima in this problem.

And you can see here that the learning rate is very important.

All these CL people have been doing research on how to do that and how to like an adopted way change this learning rate. If the learning rate is very small, your model is very slow. If you are learning that is very big, your model jumps over optimum it.

Cannot get to the point in a good time, so you need to be very.

Careful about how do you change this learning rate? And then I mentioned about a huge problem of deep models when models get deep.

Very important problems happen.

Then you want to predict your output.

You pass your input.

It calculates layer by layer, moves forward and provide output because it feed forward process and then then you want to update your model based on the error in the last layer based on the output you try to make gradient first. Regarding the this layer layer three and then.

Regarding layer two, regarding layer one and then get to the input so you back propagate error to the first layers and what happens. Sometimes this gradient gets very big or very small, very move backward.

So this gradient doesn't have any information anymore.

You can train the last layers very well, but you cannot learn anything using gradient descent for the last layer. You can see that this happens.

Your gradient gets very small, moving backward or gradient gets very big moving backward, which it doesn't have any information anymore.

So people work on these from like 191958 until we had the real deep deep learning and how they solve the problem.

Is OK what are the sources of this problem?

One problem is saturation of activation function.

So you have to be careful about activation function.

I'll explain it later.

The other problem is very initialization. As I showed on the non linear surface.

When you start, it's very important.

So there are lots of nice theoretical research on how to initialize it, which are out of this scope.

I don't talk about it.

Just be careful about how do you initialize your models and then normalization is very important. If output of some of your layers is getting bigger and bigger and bigger later your gradient also gets bigger and bigger and bigger.

So be careful about batch normalization and different version of normalization after each layer and also as I mentioned, learning rate is important if learning rate is so huge then gradient back propagation is huge.

So your model doesn't learn anything moving to the next page.

I'll talk about how people in different years propose ideas to solve this problem.

So the first real deep learning, as I mentioned before was proposed by Hinton.
How they solve this problem?
By training each layer separately and then stacking them to each other, they said that if we first initialize each layer and then stack them to each other, at least we start from a good point and we don't get into vanishing or exploding gradient.
The second solution, which was ongoing for many years, different people proposed different ways for initializing debates to start from. A good point on our nonlinear surface, the third solution is about activation function under saturation, which I explained in detail in next page.
The third solution is about batch normalization, which we try to keep a good estimate for our output each layer.
Don't let them the numbers to get very big or very small and.
The last solution is the visual connection which it was proposed in Resnet.
Paper everyone. I think working in image processing knows a lot about Red net and residual networks, so they added some skip connection to have a model which is deep and at the same time it's not deep which you can read more about it.
OK, about activation function and saturation.
What is saturation and how it leads to vanishing and exploding gradient?
Saturation is when your activation function at the end of your neural output has an upper and lower limit like you see for sigmoid, it starts to 0 for the very negative numbers and it starts to one for very positive number and also for tangent hyperbolic for many years.
We use these functions for activation function, but they found that if.
Your function has this shape. When you calculate the gradient for very big numbers, very small number to gradient in this surface is 0.
So your model cannot learn anything because your gradient is very small.
It's 0.
So many research was done just to discover this simple function.
It looks very simple, but it doesn't let saturation happen at the end of your activation function, so your model always has something to back propagate instead of just staying 0.
For very big and very small numbers.
And the research on this was is maybe still ongoing, but the main findings happened by like 2015 on different functions.
So we need an activation function which is not very expensive, but it's working well.

That's why some other functions like ILU has been proposed later.
But they are not that famous as relo, because that is less expensive.
We don't want to pay a lot of cost for all neurons.
We have many neurons in our deep neural network.
And then 2014, Hinton proposed dropout method, which everyone now sets it by 1 number in their Python code.
But what is dropout?
We said that the neural networks have many rate and many connections between so.
They are prone to overfitting.
What is overfitting, which happens?
Which is very, very important concept for machine learning.
Overfitting is when your deep neural network relies too heavily on some specific neurons and gets very sensitive to their features. When your model is sensitive to something, it shows huge reaction for any change in that neuron, even if you input noise to that neuron, then your mother shows.
Lots of reactions.
We don't want that.
We don't want our model to be sensitive to a little part of the model.
So what they do, they propose drop out.
Dropout is a technique that, during training, randomly disabled some of these connections and drops out some of the neurons so it doesn't let you to get stuck to some of the rules.
It helped the model to have a good prediction, even if some of neurons are not active over there.
So yeah, it makes your model.
Highly reliable, more generalized and less sensitive to some specific neurons. And then OK.
Now I want to wrap up this part of the presentation at the end.
We know that deep neural networks, what we learned until now, they have parameters and hyperparameters.
Parameters are whatever that you have to train using your training data by minimizing the last function, and you learn it during training and this actually define your model. When you say the model.
All the parameters are saved in your system.
But what are hyperparameters, whatever that is setting that you have to do before

starting your training just to control the learning process is hyperparameter.
So hyperparameters are not trained during training using running descent and the values are not part of the model.
They are not saved with the model.
If you want them, you have to add them extra some extra information and then yeah, they all not learn from the data set and you don't optimize your model over them.
Most of the time they are not even differentiable and OK what are parameters and hyper parameters for the neural networks?
All right.
And all the connection in your model are parameters and all the hyper parameters which actually are the pain of deep learning is huge number of hyper parameters.
You have to be careful about many many different settings before starting your training. Otherwise your model doesn't converge to a good point in that linear non linear surface.
You have to be careful about learning rate because it affects your gradient descent.
You have to be careful about batch size again for your gradient descent you have to be careful about number of epochs because of your convergence.
You have to be careful about your optimizer. Again, for your gradient descent number of layer which is depth of your model, number of neurons in each layer which is bit of your model.
Activation functions, which is very important.
It's you don't want something, but you want something that doesn't get saturated.
Dropout rate, which makes your model generalized and simple, but don't make it too simple.
A high dropout making model.
Very simple.
And then rate initialization, which I said for deep neural networks start point is super super important because function is quite non convex.
You have to be careful about that and their regularization which how you again help your model overfitting and make it more simple.
Yeah, so so I can move to CNN RNN attention on auto encoders, but I am not sure about time get started, I think, Carol.
Oh yeah.
So we can have some questions, yeah.
So yeah.

So do you want Michael's?

I can shop OK.

I'm OK. OK.

So you showed us how the the idea starts with the function of a neuron.

Yeah, yeah, yeah.

In the brain or in our bodies, in in the nervous system. Mm hmm and.

That knowledge was already available in the 50s. Yeah, at some level.

So it gave birth to to this approach to so. So we numerical problem.

I'm wondering if the pure understanding of the neurons function improved in the last 70 years and if this helps the current neural networks designs and functions and.

OK, so I answer as much as I what I know is like there is.

I remember I want in a talk 2015 and the presenter was so famous and excited then because they said they claimed that they found a better representation of brain dead.

But you know, in research where funding goes and what is already working well is very important.

So I think there has been lots of advancement there, lots of changes over there, but this simple neural model is working well and also.

2015 I was implementing my first deep neural network.

It was a huge pain because I had to calculate back propagation by myself.

I calculated all the gradients 1 by 1 and whenever my model was not working well, I calculated in a bad way.

But now you don't calculate anything.

You just call a function, so I think when you have something easy available and it's working fine.

You use it, but of course there is lots of ongoing research on getting closer to brain.

Yeah. Still is ongoing, yeah.

Also, I think you should add one more reason for the AI winners.

And if this guy winski. Yeah, it looks like every time he publishes something and he closes. AI winther.

Oh no, he makes.

Yeah, yeah, he made.

He has started AI.

He made the first AI winter.

He predicted the second AI winter.

Yes. Yeah, he was so important. He passed away. 2016, I think if he still was alive, he was one of the touring winners. Yeah.

Unfortunately, he passed away right before Llms Llms were offered 2020. He passed away 2016, but otherwise I I was pretty.

Oh, let's see.

I'm not sure if you are close to another AI venture or not. At least with this new $5 billion, we are not still close. Yeah.

OK.

Can you read?

I don't want to change this screen, the question is how does people optimize hyperparameters like grid search?

Oh yeah, so hyperparameters are not differentiable, so there is not any nice method based on gradient to do that.

So yeah, the solution is something like grid search or some heuristic versions of research. What they do like every time you just get a mix of them, you first fix them, put them side, go to the next few hyperparameters.

So it's like it's a bunch of heuristics I think.

Yeah.

The last few years everything has been getting bigger and bigger and bigger and bigger, and there's only so big models can get.

Well, it's been exciting to watch, I think.

I.

I don't know if we'll have another AI winter just because either a there's not enough GP US in the world or fast enough networks, but there's no more data left to train on.

Yeah, exactly.

So I think the the next sort of epoch or phase is going to be, I mean there's a lot of attention now going towards.

For lack of a better term, smarter AI models, which are smaller, sparse, more exotic kind of structures. And I don't know if the tools yet.

That's kind of what one of the topics that I'm interested in is.

What are the tools for?

You know we have dropout, but I would.

I would.

I would hate to throw dozens of H100.

Is that something that's mostly zeros?

Yeah. Oh, exactly.

There's gotta be something a bit more sophisticated coming down the Pike.

Yeah, so, OK, OK, let me give you 2 answer. The the first answer is yes. They are working on small deep learning model or small language models.

Now you can see that even I think I clear, DCL has a workshop or tutorial on small language models.

What they do?

They get actually what open AI.

Also accused dip sick for that.

It's like they what they are going to do is like getting the big models and compress them using sparsification using factorization using quantization and we see that today NVIDIA and all the others are proposing new machines which can handle quantized data. Or we just call a function they.

Quantize your model and they make it very small.

Still, accuracy is good, but model is not paying anymore for the zeros.

So yeah, now we I think they already moved towards small language models.

They are using all the good investment on that part to say that we are not still at the end of the story, there is a still a lot to go.

But.

The other answer which is not that relevant maybe to the question is like is just 2015.

I asked my supervisor to let me to work on deep neural networks and what I heard was no, just no because he hated them.

He said that deep neural networks doesn't have any novelty.

They don't have any theory behind them. If you want to really do a PhD in machine learning, go work with numbers.

Go work with optimization.

Go work with matrices directly instead of just playing with a model to get a good accuracy.

And I didn't work on it until four years after four years. My supervisor didn't have any funding, so they moved to deep neural networks because otherwise they had to change their job.

So still lots of people think that the neural networks, they don't have any theory, so, but as long as it's paying off, we work on it and we invest on it so.

Yeah, we are. We are.

We are working.

I mean now I think everyone is working on small models. The smaller models and also multimodal models to just expand it to different domains. Different kind of data. But as you said there is no more data and there is no bigger machines.
We have to do something with what already we have.
We cannot make anything bigger anymore, yeah.
OK, so I think maybe we can wrap up and I don't move to.
The CNN RNN attention.
Just what I want to mention. This is interesting to me because attention has not been invented at 2017 with self attention and transformer. What has been invented in 2017 has been self attention and self supervised learning, not attention.
Attention has been invented at 2014 and 2015 jointly with and it's been used a lot with RNN and CNN and also.
2010 No one was working deep neural networks for.
Like doing any specific task.
But they were using deep neural networks just for representation learning.
Then I remember all the workshop had buzzword of representation, learning and feature engineering in them. No one cared about end task, but everyone was curious on how to get trade off extracting sift features, cfidf and all of them and just give my raw image or document directly to.
Deep neural network and it extract all good features for me, so we can talk about it in next session.
But yeah, my team will decide about that.
And also if you have any feedback or if you want to learn anything or you have any suggestion about timing, you can contact us.
OK, OK.
Thank you.

◉ **Primus Chimdia Kabuo** stopped transcription