MIT 6.S191: Language Models and New Frontiers

56 MIN · YOUTUBE · HTTPS://WWW.YOUTUBE.COM/WATCH?V=EV7CLSD-YSE
https://www.youtube.com/watch?v=ev7cLSd-ySE

SUMMARY

The final lecture of the course focuses on the advancements and ongoing challenges in deep learning, particularly in the context of neural networks and their applications. The discussion highlights the evolution of deep learning technologies, the limitations of current models, and the exciting frontiers in generative modeling and large language models (LLMs).

- *Deep learning has made significant strides, but challenges such as generalization, overfitting, and data quality remain.*
- *The universal approximation theorem suggests that even a single-layer neural network can approximate any continuous function, but practical limitations exist regarding model size and generalization capacity.*
- *Neural networks can fit training data well but struggle to generalize to unseen data, leading to issues like overfitting and algorithmic bias.*
- *Recent advancements in generative modeling, particularly diffusion models, offer new capabilities in generating high-quality samples by iteratively refining data.*
- *Large language models (LLMs) like GPT leverage vast datasets and next-token prediction tasks to understand and generate human-like text, but they still face challenges such as hallucination and uncertainty in their outputs.*
- *The scaling of models and data has been shown to unlock emergent properties, enhancing model capabilities significantly.*
- *The lecture emphasizes the importance of understanding the relationship between AI technologies and human creativity, encouraging students to actively engage with these models in practical applications.*

All right, let's get started again. So, this will be the last lecture given by myself and Alexander. And personally, it's perhaps my favorite one where we're going to talk about deep learning from the lens of not only how far we've come, but also what is still left unanswered and still opportunities for overcoming some of the limitations of this technology and highlight a couple of the main directions today in terms of how, as we've been discussing throughout the course, how powerful these deep learning based AI systems have gotten.

So, as Alexander mentioned, tomorrow we will have the distribution of the t-shirts. So, please come in person after the guest lectures, we will uh actually yeah to to receive your t-shirts. Uh it's always very exciting to have the momento as part of taking this course and to kind of orient you all with where we are logistically in terms of the schedule. So, we're on day three now. This is the last lecture given by Alexander and myself. And tomorrow and Friday we will have awesome guest lectures which I'll speak a little bit more about. And then Friday the course will conclude with the project proposal competition.

So on that note, we have a tremendous amount of prizes lined up for both the software labs and the project proposal competition. We to remind you right we have these three labs which hopefully you've been working towards and each is associated with a top prize in terms of the winner judged according to the submission criteria and grading rubric that is at the end of each of these labs. The detailed instructions for each of these is on the course syllabus including the link to a Dropbox file request where you will submit your labs. So please do to be eligible for the competitions.

They're currently wide open. Okay. So for those who take this course for credit as MIT students, you have two options to fulfill your credit requirement. And the first is our famous project proposal competition which is kind of like a Shark Tank style pitch where you will present a novel deep learning uh research idea or application. And depending on the number of groups that we have presenting, the presentation time is going to be between three to five minutes.

Importantly, a couple of deadlines. By tomorrow night at midnight strict, you have to submit your group if you would like to participate in the competition. Um, we're going to do a basically a freeze of the Google Sheet sign up at that time. and we need that to logistically plan, you know, the actual proposals. On Friday, as well, there's a shared slide deck where you will drop your slides and we need those slides in by 100 p.m. basically the start of class since we will be presenting off our laptop.

So, importantly, the prizes here are very, very exciting. And as John mentioned on day one, there's an opportunity if you're a winner of the project proposal competition to kind of expand your idea and contribute it as a TED TED X talk as part of TEDex MIT. And we have some great prizes. So again, please please participate. If you're taking the course for credit and you don't want to do the project proposal competition, you can submit a one-page written review of a deep learning AI paper and we'll effectively take a look at you know do you do the assignment? Is it clear? Is it um you know capturing the technical contribution of the work and that is also due Friday.

really really excitingly we have an amazing lineup of guest lectures this this year and in particular tomorrow we're going to kick things off by a guest lecture from Chris Bishop who is a technical fellow at Microsoft he has done very pioneering work at the intersection of AI and the natural sciences particularly physics and chemistry and beyond that he's made seminal contributions in terms of uh books and technical papers ers in the foundations of machine learning. So it should be a fantastic talk and that will be followed by a talk from Matias, co-founder and CTO of liquid AI. So if you've been playing with the liquid language models in software lab 3 and as Alexander demonstrated in lecture one, these are really really efficient and high quality um language models. and Matias will be doing kind of a behind thescenes look at what it takes to actually train language models in today's world and the secrets behind that. And then on Friday we will follow on with a lecture from Doug Blank who is head of research at Comet ML uh talking about deployment considerations of AI in the wild. And finally, we have one of the uh co-creators of Jax, which is a third po very popular machine learning framework developed out of Google. And he's going to provide a detailed rundown of all the secrets behind Jax and what it's all about.

>> And Jax was is the language that Gemini, Google's premier frontier AI model, is developed in. So please come. It's going to be amazing. I personally am really excited to learn from all these amazing innovators. I also want to give a very very special shout out to some of the program staff who has been instrumental in helping with this course. In particular, John Wernern who hosted the social kickoff on Monday which many of you attended. He's been instrumental in helping us, you know, bring this together as a community and namely as well Anisha and Shria who are our lead TAs this year and perhaps you've interfaced with them during office hours and they've been amazing in help as well as the rest of the program TAs.

Okay, so that ends kind of the logistics orientation portion of this last lecture. So now let's dive into really the technical content for this particular talk. So so far we've really talked about deep learning in many many different forms right from what deep learning even is all the way up to the frontiers of reinforcement learning. And we've been thinking about how deep learning gives us a toolkit, a technological foundation to answer some really outstanding challenges across different research areas and areas of society. From autonomous vehicles and robotics to biology, medicine, healthcare, reinforcement learning and unlocking new advances in strategy and language modeling, generative models, applications beyond in humanoids, and a whole host of other areas from finance to security and beyond. And hopefully through this class you've gotten a sense of both the technical foundations but also an understanding of how these methods can be applied to particular scenarios and research areas that may be relevant to you.

And we've talked about kind of two different paradigms for thinking about how deep learning gives us a toolkit to map data to outputs and learn these functions in this way where we've seen how we can go from data to decision in the traditional supervised setting but also in the context of RL but also to go back from desired outcomes and dec decisions back to the data itself in the context of these generative models that help us solve what we call these inverse problems, right? In actually being able to model data distributions at themselves.

And in both these cases, you know, we've talked about kind of implicitly how neural networks learn these very very complex functions that go from data to output or over a distribution of data itself. And to understand this idea in more detail, I think it's very helpful to go back to some of the origins of AI and the origins of the theory of neural networks in the first place. And there was in particular this theorem that was posited back in 1989 that said the following. And this theorem is known as the universal approximation theorem.

And it says that a neural network, a feed forward neural network with a single hidden layer would be sufficient to approximate to some precision any continuous function. And so we've talked about deep neural networks, deep learning, right? Stacking multiple layers together. But this theorem is not even saying that we need multiple layers stacked together. It's saying you just need one fully connected feed forward layer.

And if you believe that a problem can be reduced to a mapping between inputs and output, a neural network with that one layer should be sufficient to approximate that function to some degree of precision. Now this seems very very powerful, right? But there are a few important caveats to consider here, which is there's no guarantee on the size of that layer that this theorem proposes, right? The number of hidden units or individual neurons within within that layer could be very very very high, right? There's no guarantee or limit on the size.

And the second one is perhaps even more important, right? which says basically that this theorem is placing no guarantees on the generalization capacity of such a model. Thinking back to lecture one, we talked about the problems of overfitting and how when we overfit to our data, we very closely reach low error on training, but maybe this means poorer performance on the test set or the generalization scenario.

And it gets really at this, you know, kind of two considerations that are very dominant in deep learning today, which is how large do we have to make these models so that they have these generalpurpose capabilities and can they actually possess general purpose capabilities that enable generalization to the new scenarios that we really care about. And so I really really like this theorem and this historical perspective because it sets into context that like any technology AI deep learning is just that it's a technology it's going to go through waves of you know growth goes through waves of criticism and limitations and we have to take these all into context when we think of this technology and you know straightforwardly right now we're in this phase of exponential growth with the emergence of these very large and very generalpurpose AI systems. But it wasn't always the case.

Both due to practical hardware constraints, but also from the community, people there were times when these deep learning models were under very very heavy scrutiny from the scientific community in terms of whether they would actually lead to real breakthroughs. So I hope that this sets into perspective from a theoretical point of view and a thought experiment point of view, but also the historical point of view just how much this moment that we are in today means in terms of the history of AI.

And so in the next part of this lecture, we're going to talk about some of the concrete limitations that neural networks still face today. And the first I think is very interesting to show and connects back to this idea proposed by the universal approximation theorem is what we actually mean by generalization and fitting too closely to our training data. So there was this paper several years back now that really I think contextualizes this better than anything else which is understanding deep neural networks requires rethinking generalization.

So in this paper they took example images from this very large-scale data set called imageet and each of these images is associated with a particular label right describing what's in the image as you see here and they did a very simple experiment where for every image in this data not class but image so not across labels but across individual images they flipped a ksided die where K was the total number of classes in the data set and reassigned the label of that image according to what the random choice right the the DY's outcome was. And as you can see here, this yields to arbitrary mappings across these examples where now you can even have two instances that belong to the same class but are now mapping to completely different labels neither of which you know are the actual semantic label of the image.

Now they took this data and they trained a deep neural network model on this you know perturbed data ra ranging from the original data untouched with the original labels to if they did progressively increasing amounts of randomization to you know flip and mess up the cl the labels in this data. And what you what was really startling to see was as you may expect right if they held out a test set again with these randomized labels as you increase the degree of randomization the performance on the test set gets increasingly worse.

But what was startling is if you look now on the performance on the training set, it didn't matter how much you randomized the data, if you trained your your model long enough on those training set examples, you could still fit a very very high classification model. And so really this this showcases that this idea of function approximation with respect to the training set and what is the capacity of these models to generalize versus you know fit training data to some degree of precision. And so this shows this principle of the universal approximation theorem kind of in practice. Granted there's a big difference because this is a deep model and you know that theorem talks about a single layer but still it's this principle of mapping functions to some degree of precision and therefore limiting generalization on test sets. So this in total right gets at this idea again to drive this point home of null networks being very powerful function approximators. And when we think about what it takes to approximate a function, let's say we have the these examples here, we what the universal approximation theorem states is that we can, you know, train and build a very good function approximator to such data.

And in the theory of deep learning, we can think about these neural networks as learning some maximum likelihood most likely estimate of data within this distribution such that if we gave the model a new data point here in purple, we can produce a reasonable prediction. But if we think about the extensions beyond this regime, we don't yet have solid theoretical guarantees on the behavior of these models in these outofdistribution scenarios. And so this raises this question of how do we know when these models do or do not know? How can we understand the limits on their capacities and their cap capability to generalize out of distribution to these harder and harder scenarios?

And so I think that in concert all of these points come together to you know ground us in what the capabilities and limitations of these models are generally speaking and often times I think there's a sort of myth and a conception in the field especially when we come to a technology like deep learning or AI and say okay can it be this magic be all endall solution ution to, you know, the world's problems. And there's this comparison that we like to draw to the historical alchemist trying to turn, you know, heavy metals into gold. Maybe there's some magic solution where you turn the crank and this is the result that you get out. And often times deep learning has been cast in this lens. But there's an equal maxim which says garbage in garbage out. Right? If you train your data on your model on a very limited set of data on a problem that may not be appropriate or you know you mismatch the size of the model to the complexity of the task all these sorts of issues can result in real problems in terms of you're not actually getting a capable model right you may be able to learn your training data but will it actually perform well on test scenarios?

The answer often is no. So as you go out and think about your final projects and maybe work in your real lives and and jobs or studies, I really really encourage you to think about, you know, task at hand, inputs and outputs. What is the quality of data that you can curate or generate for the task and does deep learning even make sense in the first place as the approach to take for such a problem. And so we can see this issue of you know garbage in garbage out manifested in a couple of different ways not explicitly necessarily garbage based right but in terms of the limitations of training data distributions and how that manifests in the behavior of these models. So let's say we have this example, right? We're training a vision model on images of natural images. And if we train this network to let's say colorize black and white images, perhaps what we could get out as a result if we did this was a output like this. Now let's take a close look at this example with this image of a dog and let's say the the picture

sorry the model is trained to colorize these images that is convert from black and white to color.

Anyone notice anything interesting about >> green on the ear. Yeahong >> the tongue is also kind of reddish right. Yeah. So why could this be the case? Well, in the tongue example, probably there are a lot of images of dogs in the data set where the dog is sticking their tongue out, right? That's the over represented example in the data. And often this behavior can be replicated as a result.

This can also have much more real and serious consequences when it comes to real world scenarios and thinking about data diversity and abundance in training neural network models. So, infamously and very very tragically uh a few years ago, an autonomous vehicle that was operating in autonomous mode crashed and ended up the driver ended up not surviving the accident. And it turned out actually that the driver that was killed was reporting multiple instances over several weeks where the car in autonomous mode would swivel towards a construction barrier that ended up being the site of the accident. And it turned out that when when uh investigations were conducted into this that in the past images from street view of that location, the construction barrier was not present.

But more recently when the car was actually, you know, uh driving in the real world, that construction barrier had been erected. And so it was this out of distribution outlier that had not been in the training set and in instance resulted in this instance very tragically um resulting in this accident. And so this gets at this notion of again data diversity, data quality, abundance and how this also closely relates to this notion of uncertainty in AI models. How do we know effectively when the models are confident in their predictions or when they're lacking information and need human intervention? And this is very particularly critical in safety critical applications, right? Things like autonomous vehicles, biology, medicine, healthcare, and also in security and public surveillance and facial detection systems. And as we saw right, this can be attributed sometimes in relating to how imbalances or noise in our data can yield these uh regions of more uncertainty in terms of the network's outputs and decisions.

So as you perhaps have experienced as well in working through lab 2, right? These are very very real issues that we have strategies for both algorithmically and from a data perspective to try to counteract. Another failure mode that I'd like to consider and and highlight because it's very um famous and well known in this field is this idea of adversarial attacks or jailbreaks, right? And the idea here is to take some example or create some synthetic example that basically fools the neural network model. And this started out from these adversarial attacks on vision models, CNN's in particular, where we realized that we could take an natural image, apply a sort of perturbation, a noise-based perturbation to that image.

And when this perturbed image which has these perturbations that are imp imperceptible to the human eye is now input to the model as a test example, it completely screws up and fools the model in terms of producing now an incorrect classification. Right? Going from a correct classification of temple to incorrect classification of ostrich with very very high probability. And really the core aspect to this is what is happening in this perturbation, this application of noise that is somehow tripping up the network so severely.

And to get at and understand how this actually works, recall again back to lecture one, the way we train neural networks is with this algorithm of gradient descent where we have some objective a loss function J. And what we are trying to optimize is our set of weights W to minimize the error on that loss. How can we change our weights in some way to minimize the loss? When we train our networks with a fixed image and a true label, this is the uh update that is occurring right over the weights itself.

Conversely, when we think about generation of an adversarial image or an adversarial attack, the question is the the inverse, right? How can we modify the input such that it yields a very very large change uh an increase in the loss perturbing the input X to now given a fixed set of weights and a true label increase the loss function to then yield this adversarial perturbation.

And so it goes even further to not only apply to 2D images, but an extension was proposed by a group out of MIT Seesale right here where they were actually able to take this concept and synthesize 3D objects in the real world that were constructed adversarially such that images of these real physical objects could fool a CNN based classification. system. And so this goes to show kind of the power of these these types of approaches. Very simple approach where instead of updating weights, were perturbing the input that yield measurable differences in the outputs of these models.

Finally, we as you've been exploring, right, a very real and severe um limitation of neural networks in terms of implications for safety and fairness and ethics is the now very well appreciated notion of algorithmic bias, right? And so again emphasizing the integration between the lectures and the software labs. Hopefully you've had the chance to explore some of these ideas hands-on. And so this is not comprehensive at all.

Right? These are just some of the limitations of neural networks that exist. But I like to highlight these because you know we can look at limitations from the lens of opportunity. Where is there still opportunity to develop new technologies and solutions to counteract some of these limitations whether it's algorithmic bias, interpretability, uncertainty or other uh limitations that are highlighted here. Okay. So transitioning a little bit right for the remainder of this talk I want to take a couple of these limitations and focus on the new advances in AI that are happening today that directly overcome some of these in particular how we can go beyond this really achieve strong capabilities in the lens of generalization going beyond um just the training distribution to build these models that have very general purpose capabilities.

And as we've seen, right, neural networks depend heavily on the data. But we've also unlocked advances where we've found that scaling the data and scaling the models themselves lead to emerging capabilities that improve generalization very very significantly as we've seen with language models today for example. Okay. So yesterday I kind of teased the first of these two new frontiers that I'd like to talk about which is the new frontiers in generative modeling and in particular a class of models called diffusion models that have unlocked really really strong capabilities in generative modeling. And as we saw in lecture four, right, we talked about in depth the foundations of generative models going back to the ideas of autoenccoding, how we can learn these compressions captured in these latent variables and the beginnings of sample generation that we saw with GANs. Yet, as we discussed in a lot of the Q&A and in the lecture uh four, these approaches have a few very

notable limitations.

Namely, that they can collapse to basically generate average instances, what we call modal collapse. It's very difficult as a result to generate instances that stretch the distributions of the data. And in particular with approaches like GANs, they're very tricky to train from the perspective of having these two neural networks that are, you know, posed with this adversarial objective. And so those limitations lead into challenges in terms of training stability, training efficiency as well as the quality and novelty of the instances that are produced by these generative models.

And so recently the latest work in generative modeling has explored and expanded an approach called diffusion modeling which has become one of the dominant paradigms in generative models today. And we're going to talk in depth about how these diffusion models work and really why they are so powerful. And as as you may have experienced, right, with common textto-image uh systems where you can input a textbased prompt and have the image generated conditioned on that prompt, the backbone to these approaches is very often a diffusionbased model.

So what is a diffusion model and how does it relate to some of the principles that we introduced in lecture 4? We saw with VA and GANs, right, grounding ourselves in the foundations that the concept was to be able to generate a sample in sort of one shot directly from some distribution of noise or low uh low set of latent variables back to the data space X.

In diffusion, there's a fundamental difference to this approach which says that rather than doing generation in a single shot. What if we can decompose the problem into iterative generations that make the task easier? And this is the principle behind how diffusion models operate. They generate new samples iteratively by learning how to remove noise and doing this consecutively step by step to refine the generation and refine the generative process.

And as you can perhaps intu it from this kind of idea of refining and removing noise. What that means is when we build a diffusion model there are two processes. One is the deterministic forward noising process that transforms data to noise. And to break that down, what we do is we start with training data, let's say images. And in the forward noising process, noise is iteratively added to the data instance step by step such that we slowly wipe out the details in the data instance until the instance is pure random noise.

And these basically now constitute iterative samples that we can use to train the model. We have taken one image and gone step by step and noised it. And now the neural network actually learns the dnoising process, the reverse process to go from noise to data. So again if we break this down in terms of steps, right? Given an image, we sample some random noise pattern shown on the right. And in forward process, this noise is progressively and increment progressively added step by step such that we get these iterative time steps of noising, right? Where now at t= 0 to t= 4 in this instance, we have these images which differ by a stepwise addition of random noise.

Now the question really becomes how do we define this reverse dnoising process and the task very fundamentally is given an image at time step t at input can we learn to estimate the image at t minus one right the prior time step looking at one instance that's less noise the task right when we're actually training the neural network is to learn the neural network to do this dinoising estimation. Given an image at t, find a way to arrive at the image at the preceding step t minus one.

How can we actually do this? Right? What would be the formulation for a loss function that we could define to train this network? Any ideas? Yeah, >> one color remove one shade. >> So the idea was to remove one color. Remember that the noise is random noise, right? This is gausian noise. There's no right the this is defined from a sampling from a gausian distribution and just applying that right but we have these two instances side by side. one differs by that addition of that gausian noise relative to the other.

>> Right? So the proposal was look at the difference in the pixel values between uh t2 and t3 in this instance and compute the difference how far they are. That's exactly right. Right. We can train this model using a like a mean squared error style loss comparing the step the image at t= 3 t= 2 where the difference is going to be the pixel values and so if these images differ by additions of random noise then the model can learn how to you know remove this noise from these step-wise pairs that we have generated and that's exactly right and so It turns out that predicting the difference which is the noise is a very very effective way to train these diffusion models. And so that gives us our recipe right we have this forward noising process. We've defined a loss to learn uh to train the neural network in the reverse dnoising process. Now once we've trained our model, how do we actually produce a new data instance at inference time using the trained model, we can start with a pattern of of noise here. And to sample a brand new generation, we take this and we take our train neural network to predict that residual difference and then use this to define a new instance at that next time step incrementally. having less noise and we do this process repeatedly time step by time step as this den noising occurs we can see our image kind of come to life stepwise such that at our final time step we're back to the image space and we have the newly generated sample.

So putting this all together right in training we learn the neural network model to predict these additions of noise and to define this reverse dnoising process and at inference to sample a brand new generation. We take that train model start with noise and perform this iterative dnoising process to construct new data samples out. And so in practice, right, what this means is that in starting with random noise but still decomposing the actual objective into these iterative steps, that means that the task is simplified, right, for the model to learn. But we still have a very very high capacity of basically information capacity at the beginning to actually produce very diverse generations. And so there's maximum variability encapsulated such that diffusion models can produce very diverse high quality samples through through this way. And so different instances starting with a completely different instance of noise can yield a strikingly different sampled image as the end result.

So I think there were a couple of questions. Yeah, go ahead. Wouldn't that suggest that what is being called noise is not noise at all? In fact, there's some sort of order to the noise that that is turning one image into that other noise. >> Yeah, that's a that's a great point. So, the question is does this suggest that the noise itself is actually not noise? Well, the difference is subtle here because where the model actually gets his signal is in that comparison

of step t versus step t minus one. That gives enough supervisory signal to actually basically bring semantic structure to the problem even if the application of noise is still a noise function. So the noise itself is not you know structured meaningfully in any way. It's the difference between those two time steps that have been created in this forward process that actually lends signal and semantic structure to the problem.

>> Yeah. >> So we started with a text image problem, right? Yeah. >> So are we also capturing some kind of description of the image that gets represented in lat space through the noise? >> A great question. So the question is about now how does text to image work? So the examples I've shown are what we call about like unconditional generation. So here in this example, it's just showing image generation without injection of text. So that allows for generation of images basically from the data space. The way now we can think about conditioning on a textbased prompt and doing examples like this is leveraging the signal from the text to basically guide the the generation process. And in practice, one way that this works is by taking an embedding of the text, right? So you you know you do your tokenization you have a text uh numerical representation and there's a language-based encoder model or embedding model that then produces a vector representing that text and then in then signal from that vector can be then basically mixed with signal uh from the corresponding image such that you can basically um learn patterns between that text representation and the image representation. And in practice with these uh diffusion models that can do like text condition generation um you can basically in training train some percent of the time using the text prompts some percent of the time on just the images and use this as signal to basically push push the model based on the text prompts to produce high quality generations based on the semantic content there. So it's really about like aligning the text with the image using this some percent of the time in training but still having some percent of training where you're just going over images so you still get high quality generations out. Yeah.

>> And is the embedding trained separately or is >> Yeah. So in this case often in practice the embedding model would have been trained separately and maybe you use that prior sometimes also you could refine the embedding model based on the um image training as well so you get signal back there. Yeah. Yeah. So on that note, right, text to image generation is perhaps the the medium that you may be most familiar with in interacting with these diffusion models. But beyond that, right, it goes um beyond images, it goes beyond text to other modalities as well. So in in recent years we have seen quite strong capabilities in molecular design in particular taking these concepts of diffusion models to the biology and chemistry space right where now you can think about as input you perhaps have um 3D coordinates of atoms in a molecule or atoms in a protein that you can then uh do diffusion over in continuous space Because like pixels in an image, right?

These are continuous values. And so tremendous efforts have been uh achieved here and and really strong capabilities now being translated into many real world applications. Okay. So the final piece of this this lecture is going to talk about really a foundation to large language models. As we all know, right, this is a very prominent frontier of AI and deep learning today. And what we want to do is really convey again core principles of what language models are, how they operate, and how that has yielded some of the very strong capabilities that we've seen in practice.

So at this point right it's no secret GPT Gemini all the you know co-pilot all these tools that are out there these

powerful language models have revolutionized our world but fundamentally what are large language models or LLMs we started this class off with this definition of what is AI what is deep learning lms are simply a special class of deep learning models that for simplicity, let's think about them as very large neural networks that are trained on very large sets of text.

And we have different types of tasks or objectives that we can use to train these models in general purpose ways. And to break that down, the core sort of flow or pipeline behind how LLMs like GPT work is that we have some general purpose, you know, unanotated really data set. For example, something that's scraped from the web like the common crawl, Wikipedia, web text code, you know, all these different sources of textbased data. And the first step in training these models as we discussed in the sequence modeling lecture is to basically split this text up into sequences or chunks sequences of chunks rather and these chunks are called tokens. So if you hear of tokenization or tokens in the context of LLMs that is referred to this chunking procedure. We then write these chunks have ids numerically and those numerical ids can be used as input to the LLM training process.

And to give you a sense of scale right of these models something like GPT3 was reported to have 175 billion individual parameters and we know that today's models GPT4 5 etc have often scaled beyond that but we have also seen these small language models such as those from uh liquid AI that are scaled back in terms of the parameters by orders orders of magnitude but still demonstrate strong capabilities.

And really the dominant task in modern language modeling is a task that we refer to as next token prediction. And that is formulated according to this objective where very similar to what we saw in sequence modeling. given a sequence of tokens predict the next token and we update the model's parameters its weights based on how good that next token prediction is but how does this work really this is the concept what is actually the loss that we formulate to the model to train it in this way so let's break that down right looking at the next token prediction task so say we have this raw text MIT deep learning is so awesome and we break that into tokens these chunks which I'm just showing as words for simplicity and there's going to be the process of converting those into IDs and embedding them right to get this numerical representation that's going to now be input into the large language model in next token prediction the task right is given a series of tokens predict the next token and when we look at this and break this down. What we do is we can compute the model is outputting actually probabilities over what the most likely next token is in this sequence over the whole set of possible tokens. Meaning if the model is predicting the token at this position in the sequence, it's actually computing the likely probabilities for all the possible tokens in the vocabulary.

And just as we saw in lecture one with losses that define classification, right? This is a classification problem. Because we have a discrete set of tokens in our vocabulary, we can look at the true next token and the predicted probabilities over the possible next tokens and compute a cross entropy loss using this. And why this is so powerful is because of the way we have set this up using the data itself as its signal. We don't require any external labels other than what signal is in the structure of the text and the structure of the data itself. And so this is is a very very very powerful idea and so beautiful in its simplicity because you can just use that classification cross entropy loss to compare the prob predicted probabilities over your token vocabulary to the

true identity of the true next token and use this to basically learn structure in these very complex textbased data.

So with that in hand right of this task of you know given a sequence of tokens predict the next token when we actually take trained language models and deploy them we can see this come to life in the fact that now you know a base language model does this next token prediction task but they've actually been tuned and refined following that to be able to follow structured prompts and act as these chat bots and interactive agents where now there's a clear definition of what you as the user provide as a prompt to the model and then what the LLM generates out as a response. And so what makes this so powerful is we combine this very simple objective function of next token prediction using a cross entropy loss with the scale of data that's available and the scale of the models and compute that we are able to achieve and together this lends powerful capabilities because now we can learn patterns in these really largecale data sets and so as you probably have experienced right today's language models are extremely capable in assisting with writing, assisting with planning, doing knowledge retrieval and they have shown this mastery over natural language. But still they face challenges in terms of again some of these same principles that I introduced at the beginning of this lecture. How do we know if the generations are confident? Right? Maybe they're very likely. They're very probable. But that's not the same as being high confidence. And so how can we quantify the uncertainty of these LLMs? It's still a very open research direction.

And probably if you've, you know, worked with these models long enough, you've seen instances where the generated text seems feasible and consistent and logical. But if you know about maybe it's it's a particular field, you know, it's not really like all the way there. And the this is the notion of hallucination. And actually, there's been very recent work that has shown that language models that are well calibrated will necessarily produce hallucinations as a result.

And similar to the principles of adversarial attacks, there are still ways to jailbreak these models. and basically probe them to expose these types of uh you know adversarial failure modes. And there are still limitations in long-term planning logic which recent advances in reasoning have uh enabled tremendous new capabilities but there's still a long way to go. And so this highlights still that even with this powerful technology there are key challenges remaining.

Finally, and and I think perhaps you know what has really struck a chord with many of us over the past couple of years is this observation that these modern language models can generalize. They can, you know, perform tasks in domains that stretch their capabilities. And a lot of this is due to this notion of emergent properties or emergent abilities and work that actually systematically evaluated the performance of these models as a function of the amount of training data that they've seen and as a function of model capacity. And in particular, seminal work showed that past a particular threshold of model scale, once you cross that threshold, you get new inflection points and sta step changes in model capabilities that seem to only come with scaling the data, scaling the compute and scaling the model parameters. And so as this concept has highlighted the the idea is that there is a basically a hierarchy of abilities that can be unlocked as the models scale towards larger and larger sizes. Uh they are able to achieve more complicated emergent abilities. And I think that this notion of emergence, thinking about how these language models are effectively modeling our world through the lens of natural

language spawns and inspires a powerful idea that I would like to leave you all with as a close to this class of how do we actually think about reasoning? How do we think about augmenting and accelerating our own capabilities with AI? Because I think that especially for someone who works on these models and closely alongside these models, you know, the goal is ultimately that these stay as technology, right?

That we as humans can use to accelerate our own creative process and our own capabilities in ways that the models uh can't. And similarly that our own limitations can be augmented by the capabilities the models possess that are still very far out of reach for us as humans. And so I think and and inspire you and encourage you to really think closely about these relationships and connections between these two worlds and how you can utilize and build these models in your own life to augment and accelerate your own work and ideas.

So with that I'll close the the lecture and you know kind of close out by thinking about the opportunity that you have at hand to actually experience and work with these models handson right you're here in this class not everyone has this opportunity or comes to to sit here in front of us and so I highly highly encourage you to dive in get your hands dirty play with these models and and actually see what it takes because it will give you new understanding that the lectures um complement but don't directly give. And so please have your hand at lab 3 on fine-tuning language models. And thank you so much for that your attention. And please stay tuned for tomorrow's guest lectures, t-shirt delivery, and much more still to come.

Thank you so much.