EST. 1597
GRESHAM
COLLEGE

AI in Business
Raghavendra Rau
22nd May 2023

Introduction

In this lecture, I would like to talk about artificial intelligence (AI) and how it is being increasingly used in business. Like my previous two lectures on blockchain and DeFi, this is somewhat of a three-parter. Last time, I spoke on how big data needs to be organized in order to be useful to process. In this part, I will try to explain how AI systems work. In the next and final lecture, I will explain how a blind reliance on these systems may lead to major issues for society.

The field of AI has had a long history with papers on artificial intelligence being written as far back as the 1950s and earlier. However, none of these models offered anything even close to something any human could do, leading to the loss of interest in AI and the rise of AI winters. Since 1950, there have been two winters. The first AI winter occurred in the late 1970s and early 1980s, when the initial hype around artificial intelligence failed to live up to expectations. Funding for AI research was cut, and many researchers left the field. The second AI winter occurred in the late 1980s and early 1990s, when a similar pattern of hype and disappointment led to another period of reduced funding and interest in AI research.

The field of AI started rebounding towards the end of the last century partly due to advances in deep learning, which have enabled AI systems to achieve breakthroughs in areas such as image and speech recognition, natural language processing, and game playing. One of the most exciting breakthroughs at that time occurred when Garry Kasparov, the reigning world chess champion at the time, was defeated by an IBM supercomputer named Deep Blue in 1997. Deep Blue's victory was largely achieved through its ability to evaluate millions of positions using brute computational force.

However, brute computational force is not exciting. A car will always be able to drive faster than a human being, and we are not astonished by this.

For humans, a truly historic moment occurred in 2016, when DeepMind, a London-based AI research company, released its AlphaGo program which made history by defeating the world champion in the ancient Chinese board game of Go. AlphaGo's victory over Lee Sedol, a top-ranked professional Go player, was seen as a major milestone in AI research because Go is considered one of the most complex board games in existence. Unlike chess, which has a finite number of possible moves, Go has an astronomical number of possible positions and requires a high degree of intuition and strategic thinking to play at a high level. To achieve this feat, AlphaGo used a combination of advanced deep learning algorithms, neural networks, and Monte Carlo tree search. The system was trained on millions of Go games and learned to recognize and evaluate different patterns and strategies. During the match against Sedol, AlphaGo made several surprising moves that even the world champion had not considered.

In 2017, the DeepMind team released AlphaZero, which within 24 hours of training achieved a superhuman level of play in three different games, chess, shogi, and go, by defeating world-champion programs Stockfish, Elmo, and the three-day version of AlphaGo Zero. How did it do it? Basically, it learnt to play the games solely through self-play and reinforcement learning. It was given no access to opening books or endgame tables.

This is the essence of AI – it uses reinforcement learning to obtain feedback and play something millions of times to achieve mastery in less time than we would learn the opening moves in chess.

Since my last lecture, the release of large-language models such as those used by ChatGPT, GPT-4, Bing, and Google Bard, have suddenly taken the business world by storm. Large language models are designed to understand natural language and generate human-like responses to a wide range of inputs. These models are trained on massive datasets of human language, allowing them to learn patterns and relationships between words and phrases.

Large language models can generate coherent and relevant text in response to prompts or questions. For example, ChatGPT can generate responses to a wide range of topics, including answering questions, providing advice, and engaging in casual conversation. They can translate text from one language to another. For example, models like Google Translate use deep learning techniques to accurately translate text between dozens of different languages. They can summarize long pieces of text into shorter, more concise versions. This is particularly useful in fields such as news and journalism, where it is important to quickly and accurately summarize important information. (I must confess I have used this functionality myself.) They can answer questions by extracting relevant information from large amounts of text. This is useful in fields such as customer support and search engines, where it is important to provide answers quickly and accurately to user queries. They can analyze the sentiment of text, determining whether it is positive, negative, or neutral. This is useful in fields such as social media monitoring and customer feedback analysis. These models have passed exams such as the SAT, bar exams, and even the sommelier exam (the written part, not the tasting part).

But what is AI? How did these systems evolve? What do they actually do? Do they understand what they are doing? Let’s start with the first question. What is AI?

What is AI?

Artificial Intelligence (AI) refers to the field of computer science that focuses on creating machines or systems that can perform tasks that typically require human intelligence. AI aims to develop algorithms, models, and systems that can exhibit capabilities such as understanding natural language, recognizing patterns, learning from experience, making decisions, and solving complex problems.

AI can be categorized into two main types: Narrow AI (also known as Weak AI) and General AI (also known as Strong AI). Narrow AI refers to AI systems that are designed for specific tasks or domains, such as speech recognition, image recognition, or recommendation systems. These AI systems excel at performing a specific task but lack the broader cognitive abilities of humans. In contrast, General AI refers to AI systems that can exhibit human-like intelligence across a wide range of tasks and possess the ability to understand, learn, and reason about diverse domains, akin to human intelligence. General AI is still a theoretical concept and has not yet been fully realized.

AI has a rich history dating back to the 1950s and has progressed significantly in recent years due to advancements in computing power, data availability, and machine learning algorithms. Machine learning, a subfield of AI, involves developing algorithms that enable systems to learn from data without being explicitly programmed. There are various machine learning approaches, such as supervised learning, unsupervised learning, reinforcement learning, and deep learning, that enable machines to improve their performance over time by processing and analyzing large amounts of data.

AI has been applied in various domains, including healthcare, finance, manufacturing, transportation, marketing, and more. For example, in healthcare, AI has been used for image analysis, drug discovery, personalized treatment plans, and disease prediction. In finance, AI is used for fraud detection, stock market prediction, and risk assessment. In manufacturing, AI is utilized for process optimization, predictive maintenance, and quality control. AI also plays a significant role in emerging technologies such as autonomous vehicles, natural language processing, and robotics.

The Historical Development of AI: Image and Text Prediction Programs

So how did AI develop? Let’s start with image recognition programs.

Image recognition, also known as computer vision, is a subset of AI that focuses on enabling machines to interpret visual information from the world, such as images and videos. The history of image recognition programs can be traced back to the 1960s and 1970s, when researchers began developing early image processing techniques to perform basic tasks such as edge detection, pattern recognition, and image segmentation.

In the 1980s and 1990s, with the advent of more powerful computing technologies and the availability of large datasets, researchers started developing more advanced image recognition algorithms, such as template matching, feature-based methods, and statistical pattern recognition techniques. These methods allowed for improved image recognition capabilities in specific domains, such as face recognition, fingerprint recognition, and optical character recognition (OCR).

With the rapid advancement of deep learning in recent years, convolutional neural networks (CNNs) have emerged as a dominant approach in image recognition. CNNs are designed to mimic the visual processing of the human brain, with multiple layers of interconnected neurons that can automatically learn to extract relevant features from images. This has led to significant breakthroughs in image recognition tasks, such as object detection, image classification, and image generation.

The main idea behind CNNs is to apply convolutional filters to an input image, which allows the network to detect different features in the image, such as edges, textures, and shapes. These convolutional filters are essentially small matrices of weights that are trained during the training process. By applying these filters to different parts of the input image, the network can identify different features and patterns. For example, how would CNN process an image of a dog?

1. First, the input image is fed into the first layer of the network. This layer applies a set of convolutional filters to the image, which detect different features such as edges, textures, and shapes.
2. The output of the first layer is then passed to the next layer of the network, which applies a second set of convolutional filters. These filters detect more complex patterns and features, such as shapes of ears and noses.
3. This process of applying convolutional filters and passing the output to the next layer continues until the output reaches the final layer of the network, which produces the classification result. The final layer might produce a probability score indicating the likelihood that the image contains a dog.

One key advantage of CNNs is that they can handle variations in scale, orientation, and position of the features in the input image. For example, if a CNN is trained to recognize dogs, it can still recognize a dog even if the dog is in a different position or orientation in the image. This is achieved through a process called pooling, which reduces the dimensionality of the output and makes the network more robust to variations in the input.

The initial dataset on which CNNs are trained are often supervised. Supervised learning is a type of machine learning where the algorithm is trained on a *labeled* dataset, meaning that each data point is associated with a label or category. During the training process, the algorithm learns to associate the input data with the correct label, so that it can accurately predict the label of new, unseen data. In addition to image classification, supervised learning examples also include speech recognition, and natural language processing. Some popular supervised learning algorithms include decision trees, random forests, and neural networks.

Text prediction programs, the other major use of AI in business are different. Text prediction, also known as natural language processing (NLP), enables machines to understand, interpret, and generate human language. The history of text prediction programs can be traced back to the 1950s, when researchers began developing early rule-based systems to analyze and process human language. In the 1960s and 1970s, with the development of computational linguistics and the availability of more computing resources, researchers started developing more sophisticated techniques, such as rule-based parsers, context-free grammars, and statistical language models. These methods allowed for basic text prediction capabilities, such as text parsing, keyword extraction, and sentence generation. Unfortunately, these did not work very well because it is very difficult to come up with grammatical rules that encompass even one language such as the English language. I gave some examples in my last lecture on big data.

In the 1980s and 1990s, with the emergence of machine learning techniques, researchers began developing statistical approaches for text prediction, such as n-gram models, hidden Markov models (HMMs), and maximum entropy models. These methods allowed for more advanced text prediction capabilities, such as language modeling, part-of-speech tagging, and sentiment analysis.

In recent years, with the advent of deep learning, recurrent neural networks (RNNs) and transformer models, such as the widely used BERT (Bidirectional Encoder Representations from Transformers), have become dominant approaches in text prediction. RNNs are designed to process sequential data, such as text, and can capture long-range dependencies, making them well-suited for tasks such as language modeling, machine translation, and text generation. Let’s discuss these two models next.

Recurrent Neural Networks and Transformer Models

Recurrent Neural Networks (RNNs) are a type of artificial neural network that are specifically designed for processing sequential data. At a high level, RNNs work by processing input data sequentially, one element at a time, while maintaining a hidden state that captures the context of previously processed data. This hidden state serves as a memory that allows RNNs to capture information from earlier elements in the sequence and carry it forward to influence the processing of subsequent elements. This ability to capture temporal dependencies makes RNNs particularly effective for sequential data analysis.

To illustrate how RNNs work, let's consider how to predict the probability of the next word in a sequence given the previous words and their context. At each time step in the sequence, the RNN takes an input word, processes it using a set of weights, and combines it with the hidden state from the previous time step. This combined input and hidden state are then passed through an activation function (such as the sigmoid or tanh function), to generate a new hidden state for the current time step. This hidden state is then used as the context for predicting the probability distribution over the vocabulary for the next word.

One of the key features of RNNs is their ability to capture long-term dependencies in sequential data. The hidden state acts as a memory that can store information from earlier time steps and carry it forward to influence the predictions at later time steps. This allows RNNs to capture context and dependencies that span across multiple elements in the sequence, making them suitable for tasks that require understanding of temporal relationships.

However, RNNs also have some limitations. One major challenge is the "vanishing gradient" problem, where the gradients used for updating the weights during training can become very small, leading to slow convergence and difficulty in capturing long-term dependencies. To address this, variants of RNNs, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), have been developed with additional gating mechanisms that help mitigate the vanishing gradient problem.

BERT models are architecturally very different. RNNs are based on a recurrent layer that allows the network to maintain a memory of previous inputs, making them well-suited for tasks that require processing of sequential data. BERT models, on the other hand, are based on a transformer architecture, which allows for parallel processing of input data and the ability to model dependencies between all input tokens in both directions.

Another key difference between BERT models and RNNs is the way they handle contextual information. RNNs process input data sequentially, with each token's meaning influenced by the tokens that come before it. BERT models, on the other hand, use a technique called masked language modeling to train the model to understand the meaning of each word in the context of the entire sentence or passage, rather than just its immediate neighbors.

Additionally, BERT models are pre-trained on massive amounts of unlabeled text data, allowing them to learn general language patterns and structures. This is a big advantage. Recall that supervised learning requires labeled data, while unsupervised learning can work with unlabeled data. Supervised learning algorithms are often used when the goal is to predict a specific outcome or label, while unsupervised learning algorithms are used for tasks where the structure and patterns of the data are of interest, even if there is no specific outcome or label to predict. In unsupervised learning, the algorithm is trained on an unlabeled dataset, meaning that there are no predetermined labels or categories for each data point. Instead, the algorithm learns to identify patterns and structure in the data on its own, without any external guidance. This pre-training makes them highly effective for tasks such as natural language understanding and sentiment analysis.

To see how these models work, consider the simple sentence: "The cat sat on the ____." How would the RNN predict the next word in the sentence?

1. Step 1: We first need to convert the words into a numerical format that the RNN can process. This is typically done using a technique called word embedding, where each word is represented as a dense vector of numbers that captures its semantic meaning.
2. Step 2: Once we have converted the words into embeddings, we feed them into the RNN one at a time. The RNN processes each word and maintains a hidden state, which captures the context and meaning of the preceding words in the sentence.
3. Step 3: To predict the next word in the sentence, we pass the current hidden state through a softmax layer, which produces a probability distribution over all possible words in the vocabulary. The word

with the highest probability is then selected as the predicted next word.

4. Step 4: In the case of our example sentence, the RNN might predict the next word to be "mat", "chair", or "window", depending on the context and the words that have come before. If the previous sentence was "I need to clean the litter box", the RNN might predict "mat" as the next word, as it is more likely to follow the context of a cat-related sentence.

This process of predicting the next word based on the context and meaning of the preceding words is repeated for each word in the sentence, generating a sequence of predicted words that form a complete sentence.

BERT would use a different approach to predict the next word. Rather than processing the sentence one word at a time, to predict the next word in the same sentence "The cat sat on the ____", BERT would take the entire sentence as input and generate contextualized embeddings for each word. These embeddings take into account not only the surrounding words, but also the global context of the entire sentence.

Once the embeddings have been generated, BERT uses a technique called masked language modeling to predict the next word. In this approach, BERT randomly masks out a certain percentage of the words in the sentence and asks the model to predict the masked word based on the context of the surrounding words.

So, in the case of our sentence, BERT might mask out the last word "___" and ask the model to predict it based on the context of the rest of the sentence. The model then generates a probability distribution over all possible words in the vocabulary, and the word with the highest probability is selected as the predicted next word.

Because BERT takes into account the entire sentence when generating embeddings and predicting the next word, it can capture more complex and subtle patterns in language compared to RNNs. However, this approach is computationally expensive and requires large amounts of pre-training data, which is why BERT models are typically trained on massive amounts of text data before they can be fine-tuned for specific tasks.

Generative Models

Generative models are a type of artificial intelligence algorithm that can generate new data that is similar to data it has been trained on. In other words, they can create new examples of data that look like they came from the same distribution as the training data. One popular type of generative model is the Generative Adversarial Network (GAN), which was first introduced in 2014 by Ian Goodfellow and his colleagues. GANs are composed of two parts: a generator and a discriminator.

The generator is responsible for creating new examples of data, while the discriminator's job is to distinguish between real and fake examples of data. During training, the generator creates fake examples of data, and the discriminator tries to correctly identify which examples are fake.

Over time, the generator learns to create more and more realistic examples of data, as it tries to fool the discriminator into thinking that its generated data is real. Meanwhile, the discriminator becomes better at identifying fake examples of data, as it is exposed to more and more examples of both real and fake data.

One example of how GANs can be used is in generating realistic images. For instance, a GAN could be trained on a dataset of celebrity faces and then used to generate new images of faces that look like they could be real celebrities. The generator would take random noise as input and output an image that looks like a celebrity face, while the discriminator would try to distinguish between the real celebrity faces and the fake ones generated by the generator.

Another type of generative model is the autoencoder, which consists of an encoder and a decoder. The encoder takes an input and compresses it into a lower-dimensional representation, while the decoder takes this compressed representation and tries to reconstruct the original input.

Autoencoders can be trained on a variety of data types, such as images, text, or audio. For example, an autoencoder could be trained on a dataset of images of faces, and then used to generate new images of faces by sampling from the compressed representation learned by the encoder.

Large-Language Models

Large language models, such as GPT (Generative Pre-trained Transformer), work by using a deep learning technique called a neural network. These models are pre-trained on massive amounts of text data, often including entire internet archives or vast collections of books, to learn how to predict the next word in a sequence of text.

The neural network consists of layers of artificial neurons that are connected in a specific way. Each neuron receives inputs from other neurons, processes them using a mathematical function, and produces an output that is fed into other neurons in the next layer. The network is trained using a process called backpropagation, which adjusts the strength of the connections between neurons to minimize the error in the model's predictions.

Once trained, the large language model can generate text by predicting the most likely next word given a previous sequence of words. The model can also be fine-tuned for specific tasks, such as language translation, summarization, or question answering.

The power of these models comes from their ability to capture complex patterns and relationships in language data, allowing them to generate highly realistic and coherent text. However, their performance also depends on the quality and diversity of the data used for training. Additionally, the large size and computational requirements of these models make them challenging to develop and deploy.

One problem is that large language models are prone to hallucinations because they are trained on vast amounts of text data and try to predict the most probable next word or sequence of words based on the context. In doing so, they may generate text that is syntactically and semantically correct but not necessarily coherent or relevant to the context. For example, a language model may generate a sentence like "The elephant flew over the moon" which is syntactically correct but semantically nonsensical. This happens because the language model has learned statistical patterns in the data and may generate text that is statistically likely but not logically or factually correct.

To address this issue, researchers are working on developing better training methods and architectures that can improve the coherence and relevance of generated text. Additionally, techniques such as fine-tuning and prompt engineering are being used to guide the language model's output towards specific tasks and domains, which can reduce the likelihood of hallucinations.

As with any technology, AI also raises important ethical, societal, and regulatory considerations. These include concerns around bias and fairness in AI decision-making, privacy and security of data, transparency and explainability of AI systems, and potential impacts on jobs and the workforce.

In the final lecture, I will expand more on these issues.

References and Further Reading

Bhatia, Aatish, 2023, Watch an A.I. Learn to Write by Reading Nothing but, New York Times, April 27, 2023 (available at https://www.nytimes.com/interactive/2023/04/26/upshot/gpt-from-scratch.html)

Chandra, Akshay L., 2018, McCulloch-Pitts Neuron — Mankind’s First Mathematical Model Of A Biological Neuron (available at https://towardsdatascience.com/mcculloch-pitts-model-5fdf65ac5dd1)

Chandra, Akshay L., 2018, Perceptron Learning Algorithm: A Graphical Explanation Of Why It Works (available at https://medium.com/towards-data-science/perceptron-learning-algorithm-d5db0deab975)

Economist, Large, creative AI models will transform lives and labour markets, April 22, 2023 (available at https://www.economist.com/interactive/science-and-technology/2023/04/22/large-creative-ai-models-will-transform-how-we-live-and-work)

Economist, Large language models’ ability to generate text also lets them plan and reason, April 19, 2023 (available at https://www.economist.com/science-and-technology/2023/04/19/large-language-models-ability-to-generate-text-also-lets-them-plan-and-reason)

Loiseau, Jean-Christophe, 2019, Rosenblatt’s perceptron, the first modern neural network (available at https://towardsdatascience.com/rosenblatts-perceptron-the-very-first-neural-network-37a3ec09038a)

Mims, Christopher, 2023, The Secret History of AI, and a Hint at What’s Next, Wall Street Journal, April 22, 2023 (available at https://www.wsj.com/articles/the-secret-history-of-ai-and-a-hint-at-whats-next-428905de)

Newport, Col, 2023, What kind of mind does ChatGPT have?, New Yorker magazine, April 13, 2023 (available at https://www.newyorker.com/science/annals-of-artificial-intelligence/what-kind-of-mind-does-chatgpt-have)

Roose, Kevin, 2023, Bing’s A.I. Chat: ‘I Want to Be Alive’, *New York Times*, Feb 16, 2023 (available at https://www.nytimes.com/2023/02/16/technology/bing-chatbot-transcript.html).

Wei, Jason, 137 emergent abilities of large language models, Nov 14, 2022 (available at https://www.jasonwei.net/blog/emergence).