MATH 400: AI & Neural Network

Lizi Shao

May 15, 2024

Abstract

This paper aims to discuss the relationships between AI and neural networks and mathematical foundations of an elementary neural network. I first explore different concepts in artificial intelligence (AI) and briefly discussion the applications of AI. Then, I explain the mathematical modeling of neural networks, a specific type of AI system. Subsequently, I discuss how learning occurs in neural networks mathematically. Finally, I reflect on per feedback and provide an analysis on the future potential of AI research.

1 Introduction: Different AI Systems

1.1 Imaginary AI and AI after ChatGPT

In movies like *2001: A Space Odyssey* and *The Matrix* and video games like *Halo*, artificial intelligence (AI) has been overwhelmingly depicted as the result of machines with far superior computational power than humans becoming self-aware. Moreover, fictional works involving AI often incorporate an element of impending doom caused by these intelligent human creations. Such works portray a "man vs. technology" type of modern conflicts where human characters face challenges posed by technological innovations.

In market and business investment, generative AI like OpenAI's ChatGPT has recently become a high-interest sector attracting attention from investors. From a business viewpoint, generative AIs are a tool with the potential to promote worker productivity and reduce operation costs when specifically tailored to meet the demand of particular industries. Similarly, commentators are also wary of the costs behind utilizing large language models (LLM) like ChatGPT, such as inference cost

(the computational cost that burdens GPUs to call a LLM) and prompt engineering cost (cost involved in engineering and structuring textual inputs to a LLM that would elicit the optimal or most effective response). [4]

1.2 AI in Scientific Reality

In disciplines where actual research on AI occurs, however, the definition of AI is much more pragmatic and, arguably, scientific. According to IBM, AI is a technology that "enables computers and machines to simulate human intelligence and problem-solving capabilities." This means that an AI system can generally discover information and infer from its discoveries. Specifically, discovering involves finding information related to a subject or a pair of subjects (like finding the function form of the relationship a list of predictor variables and a particular response variable in statistical analysis); inferring, on the other hand, involves extrapolating new information from the information you already discovered (like making a prediction about the predictor variable y using a list of x, given a model has already been fitted). [2]

The primary distinction between an AI and any other advanced human technology is that AI can complete human tasks with limited human assistance. We can anticipate from this distinction that many technologies that we were already very familiar with before the advent of ChatGPT are already examples of AI, such as GPS guidance, autopilots, and Siri, as they exemplify machines that, to an extent, replace human intelligence in activities like navigation and directioning. In fact, technologies like GPS guidance and autopilots are a specific form of AI that we call artificial narrow intelligence (ANI). ANIs are defined to be AI systems that have the ability to perform specific tasks with limited human input. Although ANIs excel in their area of expertise, their inability to apply their computational power and intelligence to other areas, much like an autopilot on an aircraft can not drive a ground vehicle, confines scientists and researchers from giving them fancier and more powerful names. Using this definition, even powerful LLM/generative AI, such as ChatGPT and Claude, that are so popular today are considered ANI in strict scientific terms. [2] [8]

In contrast to ANI, artificial general intelligence (AGI) is a theoretical AI system with the ability to perform human tasks in all domains without prior prompting and engineering. In other words, scientists expect AGI to be capable of learning and adapting to new situations like a human being. This cognitive autonomy is the primary distinction between ANI and AGI. Notably, LLMs like ChatGPT exhibit AGI-like behavior as they are able to answer domain-specific questions from any and all disciplines, but such ability requires prior human-assisted data training. [2] [9]

Despite achieving progress in AI development with products like ChatGPT, AGI still remains a distant target. However, scientists have already been theorizing an even stronger form of AI: artificial super intelligence (ASI). Similar to AGI, ASI is classified as a type of strong AI, meaning it has the cognitive ability to adapt and learn like a human. A theoretical ASI surpasses AGI as it is expected to not only adapt and learn but also vastly exceed human cognitive skills in any measurable metric. [10]

1.3 Current AI Research

While concepts like AGI and AGI and how (or whether) humans should achieve them are under contentious debate in the scientific community, researchers study and build AI systems using more pragmatic paradigms. For example, machine learning is an important subfield of AI research that studies how to make computers learn without having them explicitly programmed beforehand. Within machine learning, there are two basic types of learning approaches: supervised learning and unsupervised learning. The main difference between supervised and unsupervised learning lies in the particular concepts that the AI system is trying to learn: supervised learning focuses on comprehending and answering questions with a straightforward, yes-or-no answer, whereas unsupervised learning deals with problems that might not have a clear answer. [2]

A hypothetical example of an application of supervised learning is object recognition. We can have the object recognition system tell us whether the input object is a fruit, a living organism, or neither: in the overwhelming majority of cases, there is a clear and correct answer (an orange is clearly distinguishable from, say, an orange cat). In contrast, an unsupervised learning application could, for example, segment and classify customers based on certain characteristics. In a business setting, such an application allows companies to group customers and optimize sales strategies without necessarily attaching a label to a specific type of customer. [2] [7]

2 The Mathematics behind Neural Networks

2.1 Neural Networks: Mathematical Modeling a Biological Entity

Within the landscape of AI research, deep learning (DL) is an important subset of machine learning. Deep learning, like machine learning, involves basic approaches like

supervised and unsupervised learning. The primary difference between deep learning and machine learning is that deep learning is "deep": deep learning typically takes in unstructured data and eliminates essential pre-processing procedures in machine learning like feature extraction (extracting the variables important for the machine learning system from the data to make a prediction or response). Arguably, like not restricting yourself to specific types of questions in a textbook, you learn "deeper" when you intake information from a broader range and process them without making constraints. [8]

Central to deep learning is the idea of neural networks. We will explore the mathematics behind neural networks in the rest of the paper. First modeled by neurophysiologist Warren McCulloch and mathematician Walter Pitts in the 1940s, Neural networks mimic the biological processes of how neurons intake and output information in the brain. Neurons, each made up of dendrites, a cell body, and an axon, are the building blocks of the brain. Within a neuron, dendrites are slim branches protruding the cell body; they receive the electrochemical stimulation from other neurons and send it to the cell body for processing. The cell body intakes the electrochemical stimulation from the dendrites and decides when to "fire" a "response" (action potential) through complex mechanisms involving the cell nucleus and the axon hillock. Finally, the axon is a long projection transmitting the fired response to other neurons or neural cells. [1] [5]

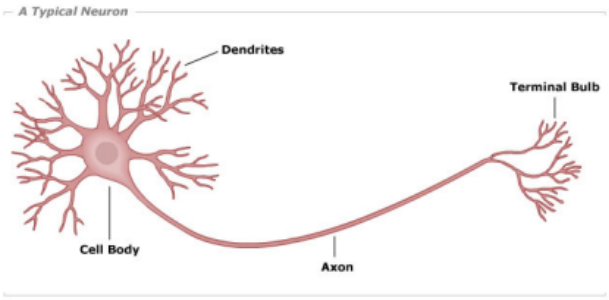

Figure 1: Illustration of a neuron [1]

Mathematically, a single neuron is modeled as a series of functions from R^n to R. The mathematical model, which we will call a neural network, takes in a vector x in R^n as input. Together, different entries of x form what we call the input layer: $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$. These entries represent the pieces of information received by the dendrites in a neuron. After an input layer is formed, $\mathbf{x}$ is multiplied by a weight vector $\mathbf{w} = (w_1, w_2, \ldots, w_n)^T$, where $\mathbf{w}$ represents the dendrites that pass along the

information to the cell body [1]:

$$(w_1, w_2, \ldots, w_n) \cdot (x_1, x_2, \ldots, x_n)^T = \mathbf{w}^T\mathbf{x}$$

Then, the summation of the weighted inputs is added by a bias term b, which we can think of the "resting state" of the neuron (the state that the neuron is in when it's not processing outside signal). The result is $z(\mathbf{x})$, which we will call the new input function [1]:

$$z(\mathbf{x}) = (w_1, w_2, \ldots, w_n) \cdot (x_1, x_2, \ldots, x_n)^T + b = \mathbf{w}^T\mathbf{x} + b$$

Subsequently, we apply an activation function σ to $z(\mathbf{x})$. As the name suggests, the activation function represents the cell body that processes and fires the action potential. There are various options for the activation function, and Fig 2 depicts one of them [1]:

$$\hat{y} = \sigma(z_3) = \frac{1}{1+e^{-z_3}}$$

Figure 2: The Sigmoid activation function [1]

Finally, if the neuron has only one middle hidden layer, i.e., the mathematical model only involves one layer of the activation function (as shown in Fig 3), the output of the activation function is simply $\hat{y}$, the predicted output of the neural network, representing the information the neuron sends to other neurons via the axon. [1]

$$\hat{y} = \sigma(z)$$

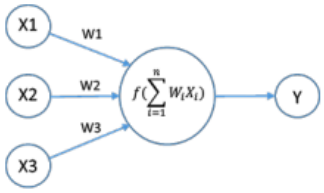

Figure 3: A neural network with one middle hidden layer [1]

In real life, a neural network comprises a series of middle hidden layers, and each middle layer is made up of multiple nodes, each of which contains the output of the activation function given the nodes from the previous layer as the input to the activation function. A heuristic behind this is that to an extent, the more hidden layers a neural network has, the more effective and accurate the learning is. In addition to multiple hidden layers, each with multiple nodes, a neural network could also have multiple nodes in the output layer. For example, one can train a neural network to predict a number of properties a given car has, making the output a column vector. [1]

2.2 Learning in a Neural Network: Updating the Weights and Biases

If the architecture of the neural network resembles a biological entity and can be expressed mathematically concisely with linear algebra, learning in a neural network occurs in a process we call backpropagation, also known as "backward propagation of errors." Backpropagation is a process used to adjust the weights and the biases (bias terms) of a network to minimize the error in the predicted output. Practically, the process involves two preliminary steps. [3]

Forward Pass: this is where the network makes a prediction by sending some input through the middle hidden layers. The input the network receives is a part of the training data, meaning it is a sample from the dataset we use to build the neural network and contains both the input features and the corresponding true output. In this initial phase of training, the weights and biases in each middle hidden layer of the network are assigned randomly. [3]

Loss Calculation: after the network makes a prediction, a loss function is calculated. The loss function measures the difference between the prediction and the true output corresponding to the input data. Like the activation function, there are various options for the loss function, and choosing which one to use is a context-determined decision. Using the previous example where the neural network only has one middle hidden layer, the loss function has the following form [3]:

$$L(\hat{y}) = L(\sigma)$$

Backpropagation: After loss is calculated through the loss function, backpropagation occurs. At the heart of backpropagation are the concept of gradient in multivariable

$$L = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

Figure 4: Binary cross-entropy: a loss function for binary classifier [3]

calculus and an algorithm called gradient descent. Recall that the gradient is a vector that consists of the derivatives of a function with several input variables:

$$\nabla f(\mathbf{x}) = \frac{df(\mathbf{x})}{d\mathbf{x}} = \left[\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right]$$

From multivariable calculus, we know that for a given x, the gradient $\nabla f(\mathbf{x})$ points in the direction of fastest increase. In the context of a neural network, we are interested in finding the gradient of the loss function L with respect to the weights $\mathbf{w}$ and the biases b (for simplicity, I will revise the concept of gradient and explain gradient descent using weights only, but the process works similarly for biases as well). Because the loss function is a composite function with respect to $\mathbf{w}$, we aim to calculate the following [3]:

$$\nabla L(\mathbf{w}) = \frac{dL(\mathbf{w})}{d\mathbf{w}} = \frac{dL}{d\hat{y}} \cdot \frac{d\hat{y}}{dz} \cdot \frac{dz}{d\mathbf{w}} = \left[\frac{\partial L}{\partial w_1}, \ldots, \frac{\partial L}{\partial w_n}\right]$$

The gradient ∇L is a vector that represents the direction of fastest increase of the loss function L for a set of weights w. Following this logic, we minimize L by moving $\mathbf{w}$ in the opposite direction of ∇L by a factor η, which we call the learning rate, a hyperparameter that we manually determine [3]:

$$\mathbf{w_{new}} = \mathbf{w_{old}} - \eta \cdot \nabla L(\mathbf{w})$$

Through backpropagation, the weights are adjusted in the direction that minimizes the loss function L. The neural network continually learns to make better predictions by iterative updates of the weights with different batches of the training dataset. [3]

2.3 Different Types of Gradient Descent

In practice, there are different types of gradient descent. We will discuss three of the popular ones.

(1) Batch Gradient Descent (BGD): the entire training dataset is used to compute the loss and the gradients in one go. Specifically, in BGD, weights and biases are updated once using the average of the gradients of all the training samples in the dataset. BGD allows us to directly achieve an optimal neural network but is fairly slow. [12]

(2) Stochastic Gradient Descent (SGD): weights and biases of the neural network are updated for individual training samples. SGD is preferable over BGD if we intend to train the neural network with very large datasets, as BGD involves calculating the gradient of each training sample and would become increasingly inefficient with larger datasets. [12]

(3) Mini Batch Gradient Descent (MBGD): this is a compromise between the two extremes. The training data is divided into small batches, e.g., 64, 100, 128 samples, and weights and biases of the network are updated per each individual batch. [12]

After the entire training dataset passes through the neural network exactly once, whether in one whole batch like in BGD or in multiple batches like in MBGD, an epoch is completed. Typically, building a good neural network requires multiple epochs. A neural network is complete after the model runs through the predetermined number of epochs and is validated using validation datasets.

3 How Far Away Are We from "Intelligent" AI?

It is important to note that although generative AI models like ChatGPT and deep neural networks are built differently, they are all constructed using complex mathematical structures with the intention of completing specific human tasks. Therefore, one can gain insight into the gap between the most advanced AI models, like ChatGPT, and strong AI, such as AGI, by discussing the mathematics behind other similar AI models like neural networks. Much like neural networks, which use backpropagation and gradient descent to learn from data, generative models employ advanced and sophisticated algorithms and vast amounts of data to become fluent in human conversation. However, just like neural networks are only a model of real neurons in the brain, ChatGPTs produce texts in a way that only approximates how humans speak. In fact, generative AI approximates a human text by determining the best word to use based on the previous words in a sentence through a probabilistic model. Looking through the lens of neural learning, we get a sense that despite their seemingly advanced capabilities to converse in human languages, generative AI is still far away from truly able to understand and reason like a human. [6]

4 Peer Feedback

One thing I noticed from the peer feedback was that many people enjoyed the structure of the presentation and the fact that I went over different confusing concepts involved in AI research, explained how they should be used, and are not actually interchangeable terms. This prompted me to go over the concepts in greater detail in the paper. Some also suggested that the math was a little over their head, so this paper also explained the mathematics presented in class with a clearer and simpler version. In addition, many of my fellow students expressed interest in the final bit of the presentation where I talked about consciousness and showed a video on the possible origin of consciousness. I also noticed Alex's comment where he mentioned seeing great potential for computers to become conscious like humans. However, looking at the mathematics behind neural networks really got me thinking: despite being a highly complex mathematical model, can a human-made model truly simulate the inner processes of the brain that result in consciousness? The last section of the paper also included a brief discussion on this topic. Overall, I believe I learned a lot from my peer's feedback, and my paper was, in many ways, constructed around some of the feedback provided by my peers.

5 References

[1] Adam Oken, An Introduction To and Applications of Neural Networks, available at https://www.whitman.edu/documents/Academics/Mathematics/2017/Oken.pdf, accessed May 14, 2024

[2] "AI versus machine learning versus deep learning versus neural networks: What's the difference," available at https://www.ibm.com/think/topics/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks, accessed May 15, 2024

[3] Cristian Leo, The Math Behind Neural Networks, available at https://towardsdatascience.com/the-math-behind-neural-networks-a34a51b93873, accessed May 14, 2024

[4] Hugo Huang, What CEOs Need to Know About the Costs of Adopting GenAI, available at https://hbr.org/2023/11/what-ceos-need-to-know-about-the-costs-of-adopting-genai, accessed May 14, 2024

[5] Jaspreet, A Concise History of Neural Networks, available at https://towardsdatascience.com/a-concise-history-of-neural-networks-2070655d3fec, accessed May 14, 2024

[6] Reece Rogers, What's AGI, and Why Are AI Experts Skeptical, available at

https://www.wired.com/story/what-is-artificial-general-intelligence-agi-explained/, accessed May 14, 2024

[7] Supervised versus Unsupervised learning, available at https://www.ibm.com/think/topics/supervise vs-unsupervised-learning, accessed May 14, 2024

[8] What is AI, available at https://www.ibm.com/topics/artificial-intelligence, accessed May 14, 2024

[9] What is AGI, available at https://aws.amazon.com/what-is/artificial-general-intelligence/, accessed May 14, 2024

[10] "What are the different types of artificial intelligence," available at https://online.wlv.ac.uk/what-are-the-different-types-of-artificial-intelligence/, accessed May 15, 2024.

[11] What is a neural network, available at https://aws.amazon.com/what-is/neural-network/, accessed May 14, 2024

[12] Sushant Patrikar, Batch, Mini Batch & Stochastic Gradient Descent, available at https://towardsdatascience.com/batch-mini-batch-stochastic-gradient-descent-7a62ecba642a, accessed May 14, 2024