#h1 23 - Introduction to Artificial Neural Networks

#h2 Inspiration from Biology

#h3 23.1 - Introduction

#pg Artificial Neural Networks (ANNs) represent the latest and most promising way to achieve artificial intelligence through machine learning. Although they have been around since the 60s, there was limited interest at first, partly because of the success of more conventional machine learning algorithms of the time and the fact that they were hard and, in some cases, impossible to train on certain types of problems. However, through a series of theoretical advances, faster and specialized hardware, and the availability of enormous datasets through the internet, ANNs are now considered almost analogous to machine learning and AI and considered the most promising way forward. Specifically, deep belief neural networks have taken on exceptionally complex problems and succeeded in solving them beyond expectation. The phrase 'Team A developed a neural net that can predict X with remarkable precision' seems to be almost every headline on any tech news-related media outlet. Inspired by their biological counterparts, neural networks offer incredible advantages over traditional machine learning algorithms. There are potential pitfalls that come with the ANNs. It is safe to say that every strength ANNs have over conventional algorithms has an associated weakness that comes with it. Unless we are aware of those pitfalls, results can be devasting, especially in a world where they are already allowed to make many important decisions on our behalf.

#h3 23.2 - Biological Neural Networks

#pg Before discussing the strengths and weaknesses of artificial neural networks, it is perhaps helpful to briefly discuss their biological counterparts: the neural networks you find in nature. The functional unit of a biological neural network (that is: our brains, spinal cord, and all nerves throughout the body) is the neuron. The neuron is a cell with a specific function: to receive and transmit distal information. All cells in our body communicate with each other through chemical processes, but these are highly local. It is the neurons that bring the information from your little toe all the way up to your brain. Roughly speaking, we divide all neurons into four classes. The specialized neurons in our sense organs receive information from the physical outside world and turn them into the electrical signals used throughout the brain, a process called transduction. These include photoreceptors, the hairs in our inner ear, and touch receptors in our skin. On the other side, we have the neurons that signal our muscles to make us move. Between those two classes, we have the most common type of neuron: the cortical interneuron. Although they do come in different flavors, most of our cortex is remarkably similar in structure and the type of neurons it contains. They communicate through chemical and electrical processes, taking inputs from other neurons and sending outputs to a third set of neurons if specific conditions are met. The fourth class is a group of other highly specialized cells in our brain that are evolutionarily older and have particular tasks. Still, we can ignore these in our current discussion. People love to throw a statistic around regarding the number of neurons and the number of connections between them. Although humans have millions of neurons, we also know that the brain is far from fully connected, meaning that most neurons only connect to a moderate number of other neurons in terms of inputs and outputs.

#h3 23.3 - The Neuron

#pg The neuron consists of three parts: dendrites, cell body, and axon. The dendrites are branch-like structures that receive input from other cells. The cell body integrates the information received from these dendrites and generates an action potential when these inputs are sufficiently strong. An action potential is a brief pulse of current that travels outward through the third structure called the axon. Doing so activates the dendrites of other neurons connected to this axon's terminal. The dendrites and axons do not physically touch. Instead, the signaling between neurons is done using neurotransmitters. The action potential causes the axon to release neurotransmitters into the gap between the axon and dendrites, and the dendrites can detect this.

#im ../assets/figures/023/023-01.png 50 512 Figure 23.1 - Schematic of the interneuron, the most common neuron found in the mammalian brain

#h3 23.4 - Pathways and centers

#pg Another principle of biological neural is that neurons aren't randomly connected. Throughout the brain, we find clusters of neurons that all seem to respond to some input or action. The location of these areas is quite similar between individuals. The first proper understanding of localization of function came through the work of Broca, who noticed that soldiers shot in one particular area of the brain suffered from a specific aphasia: an inability to understand human speech. However, it was clear they could still hear another person speaking. They were not rendered deaf by their injury. Broca's area is one of many uniquely specialized areas found in almost all individuals and almost always in the exact same location relative to other functional and anatomical landmarks.

#im ../assets/figures/023/023-02.png 50 256 Figure 23.2 - At the mesoscopic level, the human brain divides into distinct anatomical and functional areas, as demonstrated by Brodmann

#br

#pg However, this functional specialization is not the whole story of how the brain works. Although specialization, without a doubt, is a core principle of the brain, research has shown that more global states and state changes are also related to the function of the brain. It is, therefore, not wholly justified to see a biological neural network (such as our brains) as simply a large set of interconnected and interdependent modules with a fixed flow of information. Nevertheless, much can be accomplished by mimicking how the brain processes information through an extensive collection of neurons, working in parallel and selectively connected to each other to transform and pass on information from our senses. As far as I know, we build ANNs based on this principle, and for the time being, this principle alone.

#h3 23.5 - ANNs as Abstractions

#pg One of the most fascinating mechanisms in biological neural networks is generating an action potential. Recall that an actional potential is a brief pulse of current that a neuron transmits to other neurons it connects to. A mechanism behind it depends on the neuron maintaining a voltage difference between in and outside its cell body. To do so, it uses specialized proteins embedded in its outer membrane that function as pumps regulating the influx and outflux of ions: electrically charged sodium (Na+), calcium (Cl-), and potassium (K+). In receiving inputs from other cells, the cell starts to shift away from its resting voltage difference. When the inputs reach a certain threshold (i.e., receiving sufficient input), a cascade of events very quickly and briefly changes the voltage of the cell (action potential or spike), after which the cell returns to its baseline voltage difference. The spike itself travels down the axon, where it releases neurotransmitters that serve as the inputs to other neurons. This summary hardly does justice to an incredible and incredibly complex mechanism. But here is the thing. For an artificial neural network, this does not matter. As we discussed before, ANNs can abstract these processes away, presenting the entire process of generating an action potential as a single output value, as a mathematical function of the input values. ANNs capture just enough detail to mimic what makes neural networks so powerful: a large number of nodes that operate in parallel by sending and receiving information. Note that there might be a point where neuroscience does confirm additional biological processes currently abstracted away by ANNs do matter. In which case, we could build even more powerful ANNs. But the current paradigm has already proven to be extremely powerful. So powerful that ANNs seem to be replacing the conventional algorithms of machine learning. On the one hand, this could be good news. If all machine learning consisted of neural networks, we would have a unified framework rather than the thousands of specialized algorithms of conventional machine learning. However, with each benefit of using ANNs, there are potential pitfalls, as we will discuss next.

#h3 23.6 - Versatility vs. the kitchen sink

#pg Conventional algorithms we usually considered to be optimal for a particular problem. Before the rise of neural networks, machine learning engineers developed an understanding or intuition of which algorithm to use for what type of problem and data type. Some algorithms were specific to regression, others to classification problems. And some algorithms were specifically designed to work on a particular data type, like categorical variables, while others required numerical data. In contrast, neural networks over a more flexible machine learning paradigm. By tweaking some internal parameters and architectures, neural nets can take on any machine learning problem. Their inputs can be images, words, or DNA sequences, and their outputs can be classifications, reconstructions, forecasts, etc. Simply by tweaking a few internal characteristics. You can achieve much more without learning an extensive collection of separate algorithms. However, with that comes the potential pitfall. By being versatile, they can hide substantial input/output data issues that any other algorithm would readily reveal. And it makes our combined efforts in working with and understanding these algorithms much harder.

#im ../assets/figures/023/023-03.png 50 384 Figure 23.3 - The Kitchen Sink

#h3 23.7 - Scalability versus a loss of insight

#pg ANNs are highly scalable, But with better hardware, and so are other machine learning algorithms. Faster GPUs, more memory, and other optimizations will allow most algorithms to work with data of ever-increasing size. However, ANNs have another way to scale in depth. A deep belief neural network is a neural network with additional layers between input and output. One extra layer is good. Two is probably better. Three? Ultimately, the layers add a level of complexity that the algorithm can use to solve the problem of mapping inputs onto their expected or desired outputs. Imagine that you see an image of an animal coming towards you at high speed. Your decision on whether to be happy (it's a puppy) or run like hell (it's a predator) depends on your categorizing that animal. But if the pixels in that image are all wired up directly to that happy/run decision unit, it becomes a very complex problem with probably no stable solution. Each pixel's RGB value now directly impacts the decision, even though many of those pixels are irrelevant or completely uninformative to predict whether they are part of a puppy or a tiger coat. Therefore, we introduce layers between input and output where the information is abstracted away from the pixels. First, perhaps pixels combine into blobs. Blobs combine into lines or surfaces, surfaces combine into a representation of different body parts, and finally, these various body parts combine into an animal category. This scalability in complexity is a potent tool. And for that reason, it is also a dangerous tool. Because higher levels of complexity make it less and less clear what the algorithm is truly encoding, it becomes hard and usually impossible to relate any set of inputs to the activity of these hidden deep and complex layers. It is only when they reach the end that a prediction made by the algorithm seems to be aligned with what we expected and wanted the algorithm to do.

#h3 23.8 - Flexibility versus bias

#pg Very much related to the previous strength (and weakness) of ANNs is their ability to use highly variable data. Not only do the inputs vary significantly between classes (should we be interested in classification as our algorithmic objective), but they also vary within classes. Think of all the possible photographs of cats on the internet, all differing in shape, size, color, camera angles, etc. This variability was a significant challenge not overcome by the traditional machine learning approach. As a result, a lot of effort when into feature engineering – researchers designed specific transforms of the inputs into more stable, functional, and meaningful features. A simple example would be to subtract the mean intensity value from each input into a machine vision classification task so that the algorithm doesn't have to account for the fact that an object can be under very different forms of illumination. But human-guided feature engineering costs time and effort, and features designed to work one problem fail in another. Enter ANNs. Because of the scalability, researchers realized they could be far looser with what was offered to the algorithm as inputs. The deeper layers would figure that out. Feature engineering was left to the algorithm, and humans could focus on designing and improving architectures instead. But at least feature engineering forced people on the problem to recognize and understand the very nature of the input data. A growing concern is that powerful flexibility makes it easier to introduce skewed or biased data and more challenging for us to detect. It is a loss of due diligence, and the consequences can be terrible. We might not realize that our data varied along dimensions we did not intend to vary. The ANNs are agnostic on what those dimensions are. Again, it simply finds the best mapping between input and output. But the 'best' mapping might be unexpected, unforeseen, unknown, and most importantly, unwanted. Biased algorithms are a huge problem in our society, probably beyond our imagination. As a result, systemic inequalities in our societies are perpetuated or even exacerbated as we find ourselves more and more in a self-fulfilling world. An algorithm has learned that black people default more on loans than their white counterparts. When we think of it as a correlation, black people defaulting more frequently is a true statement. The numbers add up, and the statistics are sound. But correlation does not imply causation. Defaulting and being black are both caused by long-standing racial inequality. However, the algorithm saw the pattern and acted on it: it started denying loans more frequently, making it harder for them to get out of debt than white people. The problem continues, and the problem gets worse.

#im ../assets/figures/023/023-04.png 50 384 Figure 23.4 - At some point, the AI we create might have a few questions regarding ethics, bias, and morality to ask us as well.

#h3 23.9 - The Quick History of ANNs

#pg Alexander Bain (1873) and William James (1890) independently proposed the theoretical foundation for contemporary neural networks. In their work, our cognitive processes resulted from interactions among neurons within the brain. For Bain, every combination of inputs led to a specific set of neurons firing. When certain combinations of inputs occurred repeatedly, the connections between the neurons such inputs activated strengthened. According to his theory, this repetition led to the formation of memory. The general scientific community at the time was skeptical of Bain's theory because it required what appeared to be an inordinate number of neural connections within the brain. It is now apparent that the brain is exceedingly complex and that the same brain "wiring" can handle multiple problems and inputs. James's theory was like Bain's. However, he suggested that memories and actions result from electrical currents flowing among the neurons in the brain. His model, by focusing on the flow of electrical currents, did not require individual neural connections for each memory or action. C. S. Sherrington conducted experiments to test James's theory. He ran electrical currents down the spinal cords of rats. However, instead of demonstrating an increase in electrical current as projected by James, Sherrington found that the electrical current strength decreased as the testing continued over time. Importantly, this work led to the discovery of the concept of habituation. The first step towards artificial neural networks took place in 1943, when Warren McCulloch, a neurophysiologist, and a young mathematician, Walter Pitts, wrote a paper on how neurons might work. They modeled a simple neural network with electrical circuits, which laid the foundation for computational models for neural networks based on mathematics and algorithms. They called this model threshold logic. In the late 1940s, psychologist Donald Hebb created a hypothesis of learning based on the mechanism of neural plasticity, now known as Hebbian learning. Hebbian learning is a 'typical' unsupervised learning. These ideas started being applied to computational models in 1948 with Turing's B-type machines. Farley and Clark, in 1954, first used these computational devices, then called calculators, to simulate a Hebbian network at MIT.

#im ../assets/figures/023/023-05.png 50 384 Figure 23.5 - McCullough and Pitts. Together they laid the foundations for artificial neural networks and defined its various key concepts.

#br

#pg In 1958, Rosenblatt created the perceptron, an algorithm for pattern recognition based on a two-layer learning computer network using simple addition and subtraction. The most significant contribution to the field by Rosenblatt is perhaps not the network but the formulation of a learning rule to automatically train the network on a set of input-output relations. This learning rule would eventually evolve into gradient descent, still in use in today's neural networks with minor variations.

#im ../assets/figures/023/023-06.png 50 384 Figure 23.6 - Frank Rosenblatt, pioneer of artificial neural networks.

#h3 23.10 - Conclusion

#pg We will pause here when it comes to telling the story of artificial neural networks. Instead, in the next chapter, we will build our first neural networks. These straightforward neural networks, consisting of one or a handful of nodes, should help you prepare for the slightly more complex material still ahead.

#h3 23.11 – Demo Notebooks

#pg We will start building actual neural networks in the next chapter, so no notebooks accompany the current chapter. However, this website has a fun graphical interface to explore building logic gates, the first type of neural network we will start building in Python ourselves in the next chapter.

#h3 23.12 - References

#bs

#be

#be

#bp Gurney, K. (2018). An Introduction to Neural Networks. United States: CRC Press.

#bp Hebb, D. (2005). The Organization of Behavior: A Neuropsychological Theory. United Kingdom: Taylor & Francis.

#bp Mack, S. H., Kandel, E. R., Koester, J. D., Siegelbaum, S. A. (2021). Principles of Neural Science, Sixth Edition. United States: McGraw-Hill Education.

#bp Minsky, M. (1988). The society of mind. United Kingdom: Simon & Schuster.

#bp Rosenblatt, F. (1962). Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. United States: Spartan Books.

#bp Russell, S., Norvig, P. (2016). Artificial Intelligence: A Modern Approach. CreateSpace Independent Publishing Platform.

#be