Comparison of Spiking and Artificial Neural Networks (November 21, 2019)

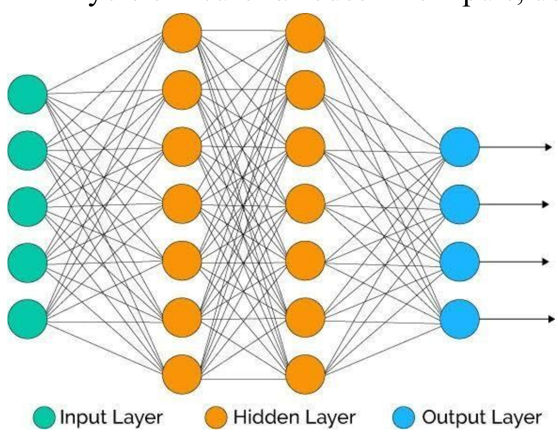
Batu Helvacıoğlu *Department of Physics, Department of Computer Engineering Koç University*
bhelvacioglu18@ku.edu.tr

INTRODUCTION

Artificial intelligence experts use many different kinds of algorithms such as neural networks to create intelligent computer programs. Neural networks are algorithms that model the human brain to classify, predict, recognize data and they are already used in many applications in 2019 such as language processing, predictions, image recognition and self-driving cars. However these neural networks still have very limited capabilities compared to humans. Computing power, time, energy and amount of data limit these algorithms. Computer scientists continue to work on neural networks to make them more accurate and efficient. Focusing efforts on developing a certain type of neural network will be helpful to remove these limits. There are many types of artificial neural networks, but they can be simplified into two categories: spiking neural networks (SNN's) and the rest. Compared to SNN's all the other types of artificial neural networks are relatively similar to each other. Therefore in this paper rest of the artificial neural network types will be simply referred to as artificial neural networks (ANN's). This paper supports that focusing development on SNN's instead of regular ANN's will be a better fit to expand the capabilities of neural networks because of SNN's higher computational efficiency and better handling of time dependent data.

I. NEURAL NETWORKS

Neural networks are the biological method to process information. Brains of every animal consist of complicated webs of interconnected neurons which are called neural networks. These web of neurons are able to learn from data and process data. Although scientists have not been able to discover how neural networks work, they took inspiration from them to create artificial intelligence. Computer scientists created artificial neural networks that are similar to biological neural networks in some respects. Artificial neural networks (ANN's) are algorithms consisting of layers of nodes that do mathematical operations and train themselves by adjusting weights of each node. ANN's are modeled after the human brain in the sense that layers of neurons/nodes take inputs, do



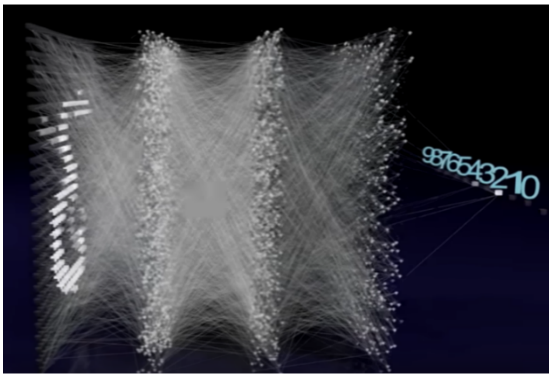
a calculation and pass on a value. However ANN's have continuous inputs whereas biological neural networks deal with discrete inputs. ANN's have an input layer. This input layer is passed on to every node in the next layer. If the previous layer has 10 nodes, a node in the next layer will receive 10 inputs, 1 from each node in the previous layer. Following figure is a model of an artificial neural network.

Fig. 1 A model of an ANN
Source: Adapted from [1]

In biological neural networks this is not the case. Biological neurons use a certain threshold called activation potential to sort through inputs and accept only the inputs that pass this threshold. On the other hand an ANN node uses weights associated with each input it receives to sort these inputs. A mathematical operation is done to these weighted inputs to create an output that is passed on the next layer as an input. Nodes at the last layer which is the output layer will represent a result. Without the correct weights for all the nodes, this algorithm will produce nothing useful. To do the intended task the ANN needs to be trained by changing the weights of inputs. One of the most common and basic methods to train is backpropagation. This method uses calculus to find the optimal weights that minimizes the error of the algorithm. Training is done repeatedly with vast amounts of training data until the algorithm achieves the desired accuracy. Training an ANN is not analogous with how the human brain learns. ANN's manipulate weights by using calculus methods on a continuous input. ANN's and variations of ANN's are the most widely used method to create artificial intelligence algorithms. SNN's are also a variance of ANN's but they differ from all the others because they take discrete inputs instead of continuous inputs. SNN's are less common than ANN's because traditional methods like backpropagation do not work on discrete inputs. However having discrete inputs like biological neural networks have advantages in other areas such as efficiency and time dependent data.

II. COMPUTATIONAL EFFICIENCY

SNN's are more accurate models of the human brain than ANN's because inputs in SNN's are spatially localized like biological neural networks. Biological neurons activate only when the input they receive is stronger than their activation potential. This activation threshold ensure that an arbitrary input does not activate every single neuron in the brain. The brain saves energy by activating only the necessary neurons. Same savings apply to SNN's in the sense that an input will not be passed on to every single node regardless of their weights. On the other hand in an ANN an input will be passed to every node even with extremely small weights. An input with a small weight will not affect



the outcome in a significant way while costing the same computing power as any other node. Fig. 2 and Fig. 3 shows the active nodes and connections between nodes with white.

Fig.2 Simulation of ANN that recognizes handwritten numbers
Source: Adapted from [2]

It can be seen from Fig. 2 that a lot of nodes are active meaning they do calculations and spend computation power. This algorithm is supposed to identify a handwritten digit. However the input of the algorithm does consist of empty space alongside the digit and the algorithm processes the empty state as well. SNN's eliminate this waste by cutting out

insignificant inputs. Fig. 3 is a model of a SNN doing the same task.

Fig.3 Simulation of SNN that recognizes handwritten numbers
Source: Adapted from [2]

In Fig. 3 there are no lines coming out from the empty space around the handwritten digit. This means that the SNN is not processing the unnecessary data unlike the ANN. The SNN has significantly less active nodes than the ANN meaning it requires less computational power to do the same task.

III. HANDLING TIME DEPENDENT DATA

A large portion of artificial intelligence algorithms are used in predicting the future with the data from the past. Natural language processing, stock market predictions, weather forecasts, self-driving cars and many more applications require time dependent data to function. However using sequential data with ANN's is a challenging task. Input of ANN's are continuous and do not depend on time. Incorporating time to ANN's requires previous data of the sequence to be inputted again with the present data. Looping previous data forward increases the complexity and computation cost of the neural network. On the other hand SNN's encode information differently and do have time dependent inputs. Inputs in SNN's are discrete pulses unlike ANN's. A spike moves forward through the nodes as time passes activating some of the nodes as it passes through. Rate of spikes and time between spikes can also be used to encode information alongside the spike itself. [3] SNN's do not get unnecessarily complex and computationally intensive like recurrent neural networks since they have time as an essential part of the input. [4]

IV. TRAINING SNN'S

Inputs of SNN's are localized in terms of space and time. While this discrete nature gives SNN's computational efficiency and simplicity advantages, it also causes problems with training. Since SNN's have discrete spikes as inputs and not continuous signals, backpropagation is not viable as calculus needs continuity. This phenomenon causes some of the existing training methods for ANN's to be incompatible with SNN's. In cases like processing time-dependent data, efficiency benefits of having an SNN outweighs the cost of developing new training methods. For example a SNN training method developed at the University of Zurich required 5 times fewer computations while matching the accuracy of an ANN. [5] However ANN's are already extremely accurate in a lot of simpler tasks and developing a SNN that has equivalent accuracy is not worth the effort.

V. ENERGY EFFICIENCY

Years of existing research enables ANN's to be more practical in many cases. First ANN's were invented in 1958 while SNN's were invented in 2002. This gives ANN's a big head start. Specialized architectures that are already developed can result in ANN's being as energy efficient or even more energy efficient than SNN's.[6] Even though SNN's are computationally more efficient, hardware specialization allows ANN's to be more energy efficient. Many companies such as Intel, Nvidia, Google, Tesla, Apple are manufacturing ANN accelerators already. SNN accelerators are still far behind ANN accelerators, but research is being done to decrease the gap. Although not commercially available, manufacturers such as IBM have been developing high-profile SNN accelerators.

CONCLUSION

SNN's are promising candidates to overcome the limits that neural network algorithms face. They have the potential to offer better accuracy and efficiency. This potential should be harvested by focusing on developing SNN's instead of ANN's. ANN's are more practical in 2019 because they are the result of years of research. On the other hand, the potential of spiking neural networks are just being realized.

Even in its early stages, SNN's can still match or exceed ANN's in most cases. Further efforts can unlock the true potential of SNN's by developing specialized hardware and easier training methods.

REFERENCES

[1] H. Strand, "How do I build a neural network predictive model with five outputs in Python? I have over 7000 datasets.", *Quora*, 2019. [Online]. Available: https://www.quora.com/How-do-I-build-a-neural-network-predictive-model-with-five-outputs-in-Python-I-have-over-7000-datasets. [Accessed: 04- Nov- 2019].

[2] Denis Dmitriev, *Neural Network 3D Simulation*, Nov. 15, 2016. Accessed on: Nov. 4, 2019. [Video file]. Available: https://www.youtube.com/watch?v=3JQ3hYko51Y

[3] A. Tavanaei, M. Ghodrati, S. Kheradpisheh, T. Masquelier and A. Maida, "Deep learning in spiking neural networks", *Neural Networks*, vol. 111, pp. 47-63, Mar. 2019. Available: https://www.sciencedirect.com/science/article/abs/pii/S0893608018303332. [Accessed 4 November 2019].

[4] D. Soni, "Spiking Neural Networks, the Next Generation of Machine Learning", *Medium*, Jan. 11, 2019. [Online]. Available: https://towardsdatascience.com/spiking-neural-networks-the-next-generation-of-machine-learning-84e167f4eb2b. [Accessed: 04- Nov- 2019].

[5] J. Lee, T. Delbruck and M. Pfeiffer, "Training Deep Spiking Neural Networks Using Backpropagation", *Frontiers in Neuroscience*, vol. 10, p. 1,Nov. 8, 2016. Available: https://www.frontiersin.org/articles/10.3389/fnins.2016.00508/full#h1. [Accessed 4 November 2019].

[6] S. Narayanan, A. Shafiee, and R. Balasubramonian, "INXS: Bridging the throughput and energy gap for spiking neural networks," *2017 International Joint Conference on Neural Networks (IJCNN)*, p. 1, 2017.