A Performance Comparison of Artificial Neural Networks and Spiking Neural Networks

Jerry Bard
Suncoast Community High School

The continual development of artificial neural networks (ANNs) on von-Neuman based hardware poses a threat to potential advances in artificial intelligence (AI). At current rates, power usage of such models will rival that of large cities or even supersede the amount of power produced globally. Therefore, it is imperative to make a transition to a more power efficient computing architecture to make way for future advances in AI. The development of neuromorphic hardware has paved a path for a new set of neural networks called spiking neural networks (SNNs) to act as a viable alternative for machine learning tasks on such brain-like hardware. The primary motive of the research is to contribute towards a search for viable applications of SNNs in their current infancy. The project wishes to address whether SNNs can outperform ANNs image classification of handwritten digits on the MNIST dataset. Despite training on non-optimal von Neumann hardware, statistically significant evidence suggests that the SNN outperformed the ANN in terms of the mean accuracy of the task. As a result of the findings, the paper suggests that further research should focus on translating older ANN models to SNN counterparts that outperform on their tasks and finding applications of SNNs on von Neumann hardware that outperform ANNs. Both prior focuses will narrow the gap between ANN and SNN performance, and in doing so makes a transition to more power efficient and uninhibited AI advancement a reality.

counterparts: equivalente

Keywords: Spiking Neural Network, Neuromorphic Computing

Introduction

Artificial Neural Networks

From self-driving cars to advances in healthcare, the use of machine learning algorithms has improved and will continue to improve nearly all aspects of one's day-to-day life. The ability to solve complex cognitive tasks is a fundamental problem that artificial intelligence addresses with machine learning. The most recent advances in machine learning have leveraged the design of artificial neural networks (ANNs). ANNs are computing systems that relate complex relationships between sets of inputs to sets of outputs by implicitly encoding patterns into neuronal structures that vaguely mimic the neurons of the human brain.

Mathematically speaking, an ANN is a function that maps an input vector space $\mathbb{R}^n$ to an output vector space $\mathbb{R}^m$ (where n is the number of inputs and m is the number of outputs) via artificial neuronal structures. Although other tools like differential equations exist to perform such tasks, most require prior knowledge between parameters to go about forming a relationship. Therefore, ANNs excel at tasks where relationships between parameters are too complicated to express in analytically (e.g. image classification, recommendation algorithms, spam detection, etc.).

An example of an ANN would be a multilayered perceptron (MLP), which maps an input vector to an output vector through a neuronal structure dependent upon synaptic weights and biases. The weights and biases are randomly initialized, and then trained on a training data set of corresponding inputs and outputs to mitigate the error of the ANN for the task at hand. The activation a^l_j of j^{th} in the l^{th} is given by Equation 1 below

$$a^l_j = \sigma\left(\sum_k w^l_{jk} a^{l-1}_k + b_j\right) \tag{1}$$

where the sum is over all k neurons in the $(l-1)^{th}$ layer, a^{l-1} is the activation of the previous layer, w^l_{jk} is the weight of the k^{th} neuron of the previous layer on the j^{th} neuron of the current layer, and σ is the activation function of the model (Michael, 2019).

Although ANNs have provided a basis for unprecedented advancements in research and industry applications of machine learning, ANN deployment is computationally demanding due to its reliance on the standard von Neumann computation architecture and its limitations. The hardware limitation of the architecture, known as the von Neumann bottleneck, stems from its separation of the memory and processor that forces the processor to spend time waiting for data to be accessed from memory, and wastes energy in doing so. As of recent years, processor speeds have continued to increase as memory management speeds have failed to compensate, which has made the bottleneck ever more noticeable. The increasing trend of power consumption poses a threat to future research in artificial intelligence and machine learning. More specifically, implementation of software ANNs on the von-Neuman architecture with complementary metal-oxide-semiconductor (CMOS) hardware consumes a substantial amount of power. If data storage and communication increase at a similar rate, energy consumed by such models will surpass 10^{27} Joules, thus exceeding the amount of energy being produced globally (Sangwan & Hersam, 2020). Unless a non-von-Neuman architecture is adopted, breakthroughs in machine learning will cease.

The Solution: Spiking Neural Networks

As was done with the first development of ANNs, inspiration for a solution to the bottleneck may be derived from the human brain. The main advantage that the human computation architecture has over the von-Neumann architecture is that the processor and memory exist in a common location: the brain. The shared area between the memory and processor eliminates the time that it takes for the processor to fetch memory, and in doing so eliminates the bottleneck (Computerphile, 2020). The field that develops hardware which exploits the computational processes of the brain is called neuromorphic engineering. The analogue of the ANN for the von Neumann architecture to neuromorphic hardware is called the spiking neural network (SNN). SNNs are a type of neural network that take yet another more biologically accurate to the neural network by transmitting information via discrete spikes rather than the continuous ones that ANNs rely on. By sending the data discreetly, the SNNs packages data into dense clusters, which requires less communication overall. The basic computing unit of the SNN would be the spiking neuron, which acts as the activation function for a given neuron. The most common model that SNNs have borrowed from neuroscience would be the leaky fire integrate (LIF) model that is governed by Equation 2 from Fang et al. (2020) below

$$\tau \frac{dV(t)}{dt} = -(V(t) - V_{reset}) + X(t) \tag{2}$$

Although more sophisticated spiking neuron models exist, such as those seen in Izhikevich (2003) and Hodgkin and Huxley (1952), the LIF model described in Wulfram et al. (2014).

Project Purpose

Since the development of SNNs is still an area of machine learning research that is in its infancy, the ability contribute research regarding potential applications of SNNs is what prompted the paper to go about a performance comparison. The problem the project wishes to address is whether the use of SNNs can be preferable to ANNs for image classification tasks. More specifically, the project statistically compares accuracy of ANNs and SNNs on the classification of handwritten digits in the MNIST dataset. The hypothesis of the project is if spiking neural networks are used for machine learning tasks, then they can outperform artificial neural networks.

Method

The project compares the LeNet5 convolutional neural network (CNN) to that of its spiking variant on the task of classifying handwritten digits. The project tests accuracy of 50 trained models on randomized initial weights and biases over 5 epochs for both models, and then reach conclusions based on whether there is a statistically significant difference between the mean accuracy of the models on a 0.05 confidence level.

Task: MNIST Image Classification

The MNIST dataset is a set of 60,000 training and 10,000 testing samples of small 28x28 grayscale images composed of the digits 0 through 9. The MNIST classification task acts as a sufficient benchmark for comparison given the large size of the data set.

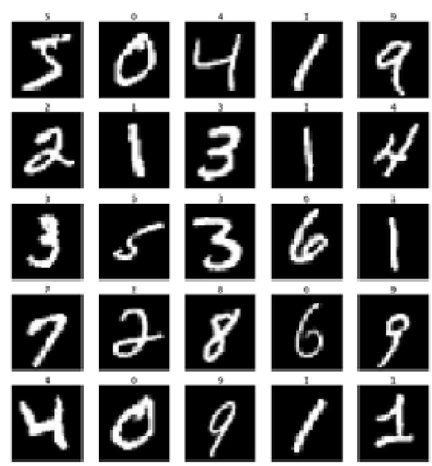

Figure 1
First 25 images of the MNIST dataset

As seen in Figure 1 above, samples in the dataset are organized by sets of 784 input pixels with values between 0 and 255 (where 0 represents the black values and 255 represents white values) and correspond output labels between 0 and 9

Model 1: LeNet-5

The first model is an implementation of the LeNet-5 convolutional neural network (CNN) architecture in TensorFlow, Google's end-to-end machine learning API. LeNet-5 was first proposed by Lecun et al. (1998) for the purposes of handwritten digit recognition in the MNIST data set, which makes it an appropriate model for the task at hand. The CNN architecture is described in more detail in Table 1 below

Layer	Feature	Size	Kernel Size	Stride	Activation
Image	1	32x32	-	-	-
Conv.	6	28x28	5x5	1	tanh
Avg. Pooling	6	14x14	2x2	2	tanh
Conv.	16	10x10	5x5	1	tanh
Avg. Pooling	16	5x5	2x2	2	tanh
Conv.	120	1x1	5x5	1	tanh
FC	-	84	-	-	tanh
FC	-	10	-	-	softmax

Table 1
Summary of parameters for the LeNet-5 CNN Architecture

The information in Table 1 may also be displayed in Figure 2 below

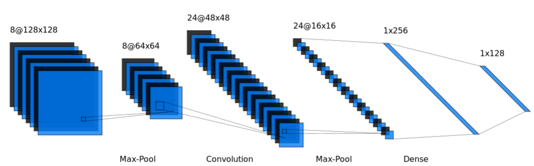

Figure 2
LeNet-5 CNN Architecture Diagram made in NN-SVG

Model 2: Spiking LeNet-5

The spiking variant of the LeNet-5 CNN architecture replaces the five non-linear layers with their spiking neuron counterparts whilst maintaining the convolutional and max-pooling layers. The membrane potential v_i of the spiking neurons are described by the leaky integrate and fire (LIF) model that can be adapted for the purposes of machine learning as described in Equation 2 below

$$\tau \frac{dv_i}{dt} = -v_i + \sum_j w_{ij}\delta(t - t_{ij}) + \tau u_{ext}^{(i)} \tag{3}$$

Where the spikes are treated as Dirac's delta impulses, w_{ij} are the given weights for a layer, τ is the leakage time, and $u_{ext}^{(i)}$ is the external current of the prior non-spiking layer (Yanguas-Gil, 2020). The actual model is implemented in python with the Pytorch framework and methods further discussed by Yanguas-Gil (2020).

Procedure

The models were initialized with random weights 50 times and trained over 5 epochs. After training over 5 epochs, the accuracy was then tested over a testing set of 10,000 images. The accuracy of model on the testing set was then recorded for each trial.

Results

The models were initialized with random weights 50 times and trained over 5 epochs. After training over 5 epochs, the accuracy was then tested over a testing set of 10,000 images. The accuracy of model on the testing set was then recorded for each trial.

Sample Mean	Sample SD	Standard Error	Confidence. Interval (95%)
97.69	0.2070	0.0293	[97.63, 97.75]

Table 2
Model 1 Testing Accuracy

Sample Mean	Sample SD	Standard Error	Confidence. Interval (95%)
98.09	0.2055	0.0291	[98.03, 98.15]

Table 3
Model 2 Testing Accuracy

Although the sample mean accuracy was higher for Model 2 than Model 1, a two-sample hypothesis test must be used to infer information about the population overall. Before doing so, one must verify that the assumptions of randomness, independence, and normality are appropriate to ensure that the conclusions made are valid. The assumption of randomness is valid given that the initial weights that the model optimizes during training are random, the training set is shuffled during each epoch, and the testing set is shuffled every trial via TensorFlow and PyTorch for model 1 and 2 respectively. Each trial is independent of the other given that the model reinitializes weights as it goes from one trial to the next (i.e. the previous trial has no effect on the current trial given that the initial parameters are randomized) so the assumption of independence is also valid. Finally, the assumption of normality may be validated by the large sample size, and the shape of the histograms. The sample size for each model is 50, which is sufficiently large (i.e. greater than 30) to suggest normality thanks to the central limit theorem.

Usa backpropagation aplicada a spikes utilizando distribuciones de probabilidad(?)
(Coarse scale representation of spiking neural networks: backpropagation through spikes and application to neuromorphic hardware)

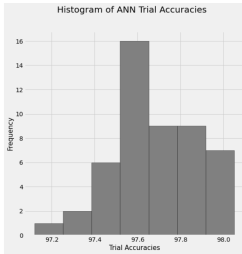

Figure 3
Model 1 Histogram (Grey)

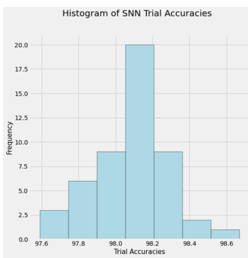

Figure 4
Model 2 Histogram (Blue)

Upon analysis of Figure 3, one would find that the distribution is unimodal with a slightly negative skew without any extreme outliers. Although the slight negative skew would act as evidence against normality, the fact the distribution is unimodal, has a sufficiently large sample size by the central limit theorem to assume normality, and has no extreme outliers outweighs the very slight negative skew and ultimately validates the assumption of normality for Model 1. Analysis of Figure 4 suggests that the frequency distribution of accuracies for Model 1 is unimodal, symmetrical, and has no extreme outliers which all act as reasons to validate the assumption of normality. Given that all of the assumptions were validated, the two-sample t-test seems to be the appropriate statistical test for comparison.

A two-sample t-test for the difference of population mean accuracies of Model 2 and Model 1 may be constructed by letting the null hypothesis state the mean accuracies are equal and the alternative hypothesis state that the mean accuracy is greater for model 2 than model 1. Next, a t-test statistic is found to be 9.697 with the information in Table 2, Table 3, and the sample sizes of 50. A corresponding p-value for this right-tailed hypothesis test may be found by evaluating the t cumulative distribution for 49 degrees of freedom from the prior t-test statistic to infinity, which gives a p-value of 2.77 E-13. A p-value of this size would provide extremely strong evidence to reject the null hypothesis which suggests that the population mean frequency of model 2 is larger than model 1. Furthermore, the actual difference between the population mean accuracies of model 1 and model 2 can be quantified with a confidence interval. A 95% confidence interval calculation with similar prior information gives an interval of (0.32, 0.48). The proper interpretation of the confidence interval would suggest that one can be 95% confident that the difference between the population mean accuracies of model 1 and model 2 is between 0.32% and 0.48%,

Discussion

The essentially negligible right-tailed p-value of 2.77 E-13 suggests that there is very strong statistically significance evidence to support that the population mean accuracy for model 2 is greater than model 1. Consequently, the results support the initial hypothesis since the evidence suggests the SNN was able to outperform the ANN on the image classification task. In addition to supporting the hypothesis, the results also provide a quantitative difference via the 95% confidence interval of (0.32, 0.48).

One improvement of the project would be to use neuromorphic hardware to train the SNN model since SNNs are best suited for such hardware. Because neuromorphic hardware is only available for industry and academic research it was impossible to do so, thus the project defaulted to training both models on a von-Neuman based cpu and could not adequately measure power usage or optimally train the SNN. Although the inability to train SNNs on neuromorphic hardware could be considered a negative, a more positive way to view the results of the project would be to state that it sheds light on a potentially beneficial area of future research. Even though the SNN was trained in a non-optimal setting, the model was still able to outperform the ANN thus suggesting there are other applications of SNNs outperforming ANNs in non-optimal environments. Therefore, further research should consist of finding situations where SNNs outperform ANNs von-Neuman based hardware. Since neuromorphic hardware is available for a very small group of researchers, the applications to which SNNs run optimally are only studied by few. If more applications of SNNs on von-Neumann based hardware are discovered, then more mainstream attention will inspire further research, which will help close the gap between SNN and ANN performance. If the gap between SNNs and ANNs is narrowed on von-Neuman based hardware, then it will allow for a smoother transition when neuromorphic hardware becomes mainstream.

Another shortcoming of the project would be that the LeNet-5 model is an outdated model for image classification that was first created in the 90s. When compared with newer ANN models, SNN equivalents tend to do worse for most metrics on the same task due to limited optimization methods. Therefore, the consensus of the field tends to state that SNNs only outperform ANNs in terms of power usage, but not other metrics like accuracy. While this might be the case when comparing newer ANNs to SNNs equivalents, the results of this paper suggest otherwise for older models. Hence, future research regarding the translation of older ANN models into SNN models for machine learning tasks is an area that more researchers should focus on. By developing SNNs that outperform older ANN counterparts, the limitations that hold SNNs back will become more apparent, which will ultimately help develop SNNs that compare to modern day SNNs, which further closes the gap between SNNs and ANNs.

In closing the gap between ANNs and SNNs by emphasizing research in SNN applications on von-Neumann hardware, and the translation of older ANN models to SNN models, a future where SNNs replace ANNs on neuromorphic hardware to increase power efficiency becomes more likely. Increasing power efficiency will better the field of AI overall since the limitations of the von-Neumann architecture will not be there to hold potential advances back. By spurring more innovation, AI will be allowed to positively impact people society on a even greater level

References

Computerphile. (2020). https://www.youtube.com/watch?v=Qow8pIvExH4

Fang, W., Yu, Z., Chen, Y., Masquelier, T., Huang, T., & Tian, Y. (2020). Incorporating learnable membrane time constant to enhance learning of spiking neural networks. https://arxiv.org/abs/2007.05785

Hodgkin, A. L., & Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, *117*(4), 500–544. https://doi.org/10.1113/jphysiol.1952.sp004764

Izhikevich, E. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, *14*(6), 1569–1572. https://doi.org/10.1109/tnn.2003.820440

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Michael, A. N. (2019). http://neuralnetworksanddeeplearning.com/chap2.html

of Secondary Curriculum, D. (2017). Palm beach regional science engineering fair 2017-2018 handbook.

Sangwan, V. K., & Hersam, M. C. (2020). Neuromorphic nanoelectronic materials. *Nature Nanotechnology*, *15*(7), 517–528. https://doi.org/10.1038/s41565-020-0647-z

Science, S., & of Florida, E. F. (2017). Ssef of florida rules supplement to the international science and engineering fair rules. https://ssefflorida.com/

Wulfram, W. M., Kistler, W. M., Naud, R. M., & Paninski, L. M. (2014). *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press. https://scholar.google.com/scholar_lookup?title=Neuronal%20dynamics:%20From%20single%20neurons%20to%20networks%20and%20models%20of%20cognition&author=W.%20Gerstner&publication_year=2014

Yanguas-Gil, A. (2020). Coarse scale representation of spiking neural networks: Backpropagation through spikes and application to neuromorphic hardware. *International Conference on Neuromorphic Systems 2020*. https://doi.org/10.1145/3407197.3407221