How Close are ANNs to the Brain?

Shashi Kant Gupta
160645

Syed Naveed
160736

1 Introduction

Artificial Neural Networks were modelled after deriving analogies from the structure of brain in hope of getting close to do things in the way how our brain does and hoping it to give a good set of results on tasks which brain performs well. Several anologies has been developed and are performing incredibly well in their domain. But neuroscientists have mostly criticised about ANN and Deep Learning models as to be just a bunch of computation with nothing as close to how real brain works. So, here we tried to answer about this critisim by analysing the already existing models and relating their closeness to the brain.

We started with the study of Conventional NN, CNN and RNN model and evaluated them on MNIST digit dataset to draw some conclusion on their functioning. We performed rigurous training of several CNN and RNN model to achieve some remarkable test-set performances and drawn some important conclusions based on their implementation. We then proceeded to study about neural information processing in brain i.e how information are carried in human brain from one neuron to another and learning is carried in the brain. How analogies were made on this which led to the develpment of ANN. Next we will discuss about the controversies and shortcomings of present models. Then we performed a short survey of present research which were going on to overcome the shortcomings and develope a more biologically plausible Deep Learning models. Finally, at the end we will propose an model which is developed on the basis of biological principles with a hope to gear toward a more scalable Deep Learning algorithms and to bridge the gap between Deep Learning and Neuroscience.

2 Conventional NN, CNN and RNN

We started with gaining theoritical knowledge of Neural Networks specifically in CNN and LSTM using the sources [1, 2]. We trained our models on the MNIST digit dataset of 60000 training images and 10000 test images of size (28, 28). We used validation split of 2% i.e. trained on 58800 training data and used the rest 1200 for validation. All of these were performed on kaggle kernel which provide NVIDIA Tesla K80 GPU, RAM of 14GB, and disk space of 5.2GB. Initially, we trained several models using some 10000 training datas to arrive at some good model which was then trained on complete dataset and dropouts were controlled so as to get less deviation in between train and test datas. All these were performed using keras library with tensorflow as backend.

2.1 CNN Architecture and Parameters

We trained our architecture for a batch-size of 32 and for 12 numbers of epochs. Categorical Crossentropy loss was used for optimisation purpose with Adadelta as optimiser. We achieved an best accuracy of 99.49% - 99.50% on our best model. Which we further improved by using ensembling. We tried averaging and stocking method stocking gave us better result in which we created 5 different CNN architecture one of which was our best model and others where modelled with some slight changes, after ensembling we achieved an accuracy of 99.60%.

CNN Architecture		
S. No.	CNN Structre	Test Accuracy
1.	Conv[3x3](32) - Conv[5x5](128) - MaxPool[2x2] - Conv[3x3](128) - MaxPool[2x2] - Droput(0.2) - Dense(128) - Droput(0.3) - Dense(10)	99.50%
2.	Conv[3x3](32) - Conv[3x3](64) - MaxPool[2x2] - Conv[5x5](128) - MaxPool[2x2] - Conv[3x3](64) - MaxPool[2x2] - Droput(0.2) - Dense(128) - Droput(0.5) - Dense(10)	99.31%
3.	Conv[3x3](32) - Conv[3x3](64) - MaxPool[2x2] - Conv[5x5](128) - MaxPool[2x2] - Droput(0.2) - Dense(128) - Droput(0.5) - Dense(10)	99.38%
4.	Conv[3x3](32) - Conv[3x3](64) - MaxPool[2x2] - Conv[5x5](128) - MaxPool[2x2] - Conv[3x3](64) - MaxPool[2x2] - Droput(0.2) - Dense(128) - Dense(10)	99.45%
5.	Conv[3x3](32) - Conv[5x5](128) - MaxPool[2x2] - Conv[3x3](128) - MaxPool[2x2] - Droput(0.2) - Dense(128) - Droput(0.3) - Dense(10)	99.50%
6.	Ensembling (Stocking) of 1 to 5.	99.60%

2.2 RNN Architecture and Parameters

We transformed the image data to a time distributed data as considering 28 identities in columns as informations at some time 't' for total duration of 28 time units. This transformation was done for in two differen ways once we considered the rows as time axis for the next one we considered columns as the time axis.
We trained our architecture for a batch-size of 32 and for 15 numbers of epochs. Categorical Crossentropy loss was used for optimisation purpose with Adadelta as optimiser. We achieved a best accuracy of 98.99% . Their ensemling gave us a dramatic improvement upto 99.38%

RNN Architecture			
S. No.	Time Axis	CNN Structre	Test Accuracy
1.	Col	GRU(128) - Dense(64) - Dense(10)	98.88%
2.	Col	LSTM(128) - Dense(64) - Dense(10)	98.77%
3.	Row	GRU(128) - Dense(64) - Dense(10)	98.99%
4.	Row	LSTM(128) - Dense(64) - Dense(10)	98.97%
6.	—	Ensembling (Stocking) of 1 to 4.	99.38%

2.3 Conclusion

Following conclusions can be drawn keeping in mind their usefulness in our new proposed model:

- It's possible to convert a non-sequential data as time series data by some methods like it was used above. Can be related to processing in brain to be like we percieved the image left-right or top-bottom
- The dramatic increase in accuracy after ensembling of the two tranformed image data can be related as chances are huge that we don't always percevive images in any one definite direction.
- CNN layers can be thought as receptor field [1] for sensing data before actually arriving at the main classification (decision making) stage.

3 Present Brain View of Neural Networks

This is the study carried by us to understand about how information processing is done in real neurons so that we can relate it to the artificial neurons. Also what analogies were drawn initially which inspired neural network models in machine learning. We will try to explain this further in

section 3.6 that the present model is indeed doing it which was initially just assumed to be in that way.

3.1 Communication in Biological Neurons

Sensors/ Receptors receives inputs as current and produces voltages in the connecting neurons. If the produced voltage is above a certain threshold it produces spikes (see Figure 1, spikes are generated at a threshold of 1.0V) which travel along the axon. These axons branches out and connects to dendrites of other neurons via synapses which produce voltages in them based on the spiking activity of initial neuron. Whenever a spikes occurs at pre-synapse neuron it increase the voltage of post-synapse neuron by a magnitude equivalent to synapse 'weight' (See the figure 1 which illustrate this). Information encoded as the firing rate of neuron i.e. number spikes per unit time and also the precise timing of this spikes also carries a lot information. Well firing rates contributes most the encoded information but in many situation precise timing of spikes where necessory. [3]

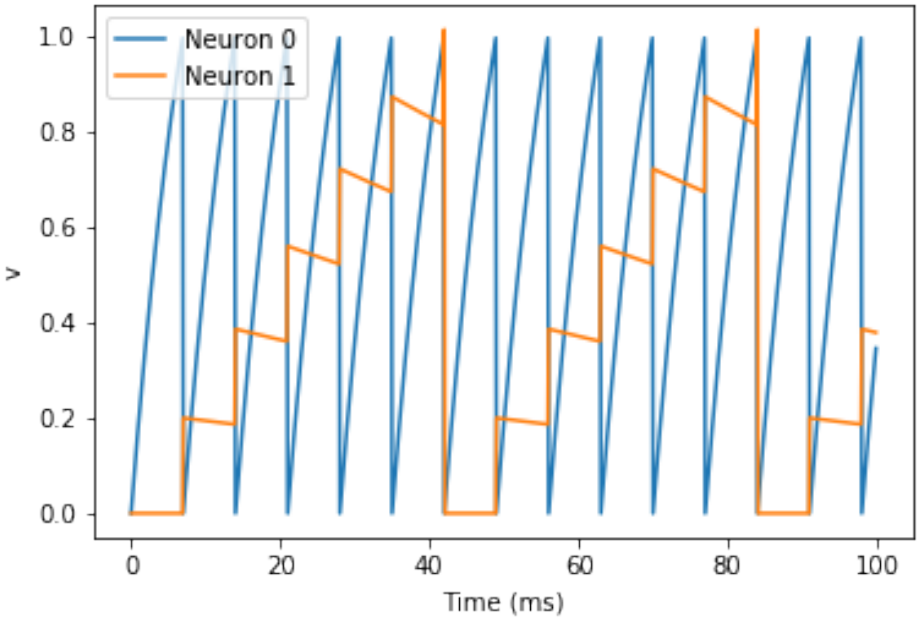

Figure 1: An example showing how spikes of one neuron produces spikes in another. Plot generated using Brian2 Simulator

3.2 STDP: Learning in Biological Neurons

Spike Time Dependent Plasticity (STDP) is most popularly explained model of learning in biological neurons which says that synaptic weights changes if there are pre-synaptic spikes around a post-synaptic spike. This change is positive if the post-synaptic spike is after the pre-synaptic spike and vice-versa. Also this change is inversly proportional to the change in time between a pre-synaptic spike and a post-synaptic spike. (Look for Figure 2 for an Illustration of this rule)

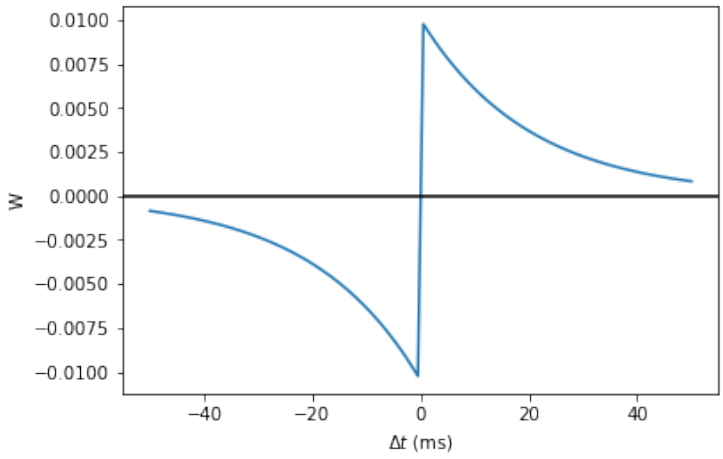

Figure 2: Visualisation of STDP

3.3 Analogy

Assuming that precise timing of spikes doesn't matter and only the firing rate contributes most to the information stored in spiking activites the artificial neuron model is drawn as shown in the figure 3. Also it also assumes that **f(Wx)** gives the firing rate of neurons, which is criticized by many but it indeed indicates the firing rate which had been explained by us in section 3.6. Learning is through backpropagation of errors. [1]

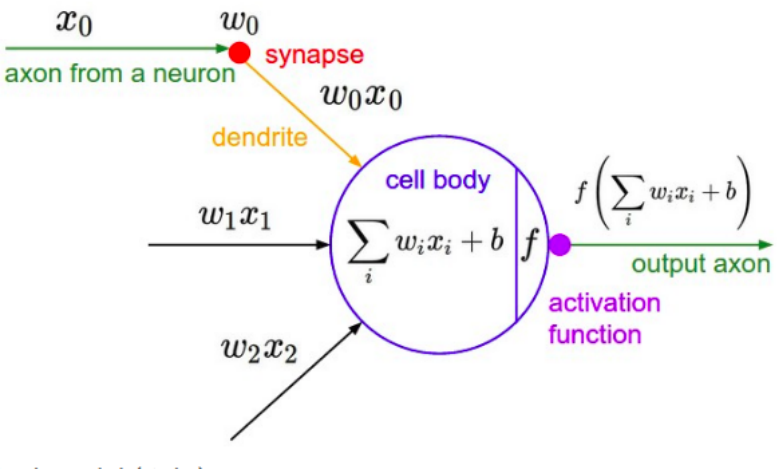

Figure 3: Artificial Neuron. Source: http://cs231n.github.io

3.4 Issues

Backprop is not biologically plausible as there will be a need of feedback path to communicate the errors, if the feedback path exists then also it need to be of exact weight of feed-forward path to perform backprop. Biological neurons communicate via Spikes whereas these communicates via a real value numbers. Loss of information of precise timing of Spikes.

3.5 Explaining How Conventional NN Carries Info about Firing Rates

Here we tried explaining how the present model can be realized as spikes and they really do carry the information of firing rate forward in the network! Taking the feature x_i of observed variable **x** as the produced current at time t_i. The produced voltage at time $t = t_i$ can be assumed to be:

$$V_i = V_{i-1} + W_i x_i \tag{1}$$

Taking threshold Voltage at which spikes occurs, to be V_{th}. So, no of spikes $\approx \sum W_i x_i / V_{th} \propto$ firing rate for some Δt over the full observations. It can also be shown that the generated spikes carry the same info as that of a RELU activation to next neuron. Since the increase in the voltage of next neuron is only when spikes occur in the pre-synapse neuron, which would occur only when it is non-negative. Increase in Voltage of post-synapse neuron = (no of pre-spikes)*W2 $\approx W2_{ij} * (\sum W1_{jk} x_k / V_{th})$, this is what is normally communicated in a neural network!

4 Present Research

Recently, a lot of research is going in the domain of making Deep Learning to be more biologically plausible we will briefly discuss about two of the important contributions which can be counted to be as important remark in this domain.

4.1 Peter U. Diehl's Work

Peter U. Diehl [4], Implemented an SNN architecture in Brian2 SNN Simulator based on STDP learning rule to get 95% accuracy on the MNIST digit dataset. This can be one of important landmark for the proof of working of STDP learning rule for object classification and learning. But lot of work still needed to be done because he worked out this on an SNN simulator which can't be used as a Deep Learning tool for classification and other machine learning task. We need to work out on it to prepare something which can be helpful for the deep learning community. Still, this plays an important landmark for the working STDP in learning.

4.2 Yoshua Bengio's Work

Yoshua Bengio [5], Shown a Machine Learning Interpretation of STDP rule and a learning rule governed by it using a latent variable setup for post-synaptic voltage. He proposed a STDP update rule which makes more sense from a machine learning point of view. Assumed STDP rule (S - pre-spike, V - post-synapse voltage):

$$\Delta W_{ij} \propto S_i \Delta V_{ij} \tag{2}$$

Can be related to an **SGD** update on Objective **J** if: $\Delta V_{ij} \approx \partial J / V_{ij}$. Taking post-synapse volatage as a latent variable h, he proposed an EM learning algorithm as (hypothesis: **J** comes out of a variational bound on the likelihood of the data):

$$log(p(x;\theta)) \geq E_{q(H|x)}[log(p(x,H;\theta))] \tag{3}$$

$$J = log(p(x,h;\theta)) + regularizer \tag{4}$$

5 Spike Inspired Deep Learning Model

This part has been removed because of ongoing work and privacy reasons.

6 Conclusion and Future Work

6.1 Conclusion

- Although the ANN has really progressed a lot but its still very far from Neuroscience perspective.

- Work has been going to bridge the gap between the two and to achieve a scalable learning algorithm.
- We tried to add a little contribution to it, by propsing a new idea to bridge the gap between Deep Learning and Neuroscience.

6.2 Future Work

This part has been removed because of ongoing work and privacy reasons.

Description of tools/softwares

Keras Deep Learning Library, Tensorflow, Python, Brian2 SNN Simulator, Kaggle Kernel

Acknowledgments

We would like to express our gratitude to **Prof Piyush Rai** for giving us this opportunity to work on this project

What we learnt

This project have been a great learning experience for us. We explored out vast amount literature and present work going on Deep Learning and its biological feasibility. This also introduced us to Brian2 SNN Simulator to simulate the functioning brain neurons to see how they behave. And lots of the Deep Learning libraries like Keras and Tensorflow. This also gave us a good experience in how to train NN model to get desired accuracy i.e. how to balance different parameters of the model.

Disclaimer

All of the presented the work carried out in the project has not been re-used from any another course project at IITK or elsewhere, or any other project you might have done elsewhere (e.g., an internship).

References

[1] Fei Fei Li. Convolutional neural networks for visual recognition.

[2] Colah Blog. Understanding lstm networks.

[3] Filip Ponulak1 and Andrzej Kasiński. Introduction to spiking neural networks: Information processing, learning and applications. *Acta Neurobiol Exp*, 2011.

[4] Peter U. Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Front. Comput. Neurosci.*, 03 August 2015.

[5] Yoshua Bengio, Dong-Hyun Lee, Jörg Bornschein, and Zhouhan Lin. Towards biologically plausible deep learning. *CoRR*, abs/1502.04156, 2015.

[6] Adam H. Marblestone, Greg Wayne and Konrad P. Kording. Toward an Integration of Deep Learning and Neuroscience. *Front. Comput. Neurosci., 14 September 2016*

[7] Thomas Mesnard, Wulfram Gerstner, Johanni Brea. Towards deep learning with spiking neurons in energy based models with contrastive Hebbian plasticity. *CoRR, 2016*