Demystifying GPU architecture for AI processing

Blog 2 –Understanding of basic building block for processing AI algorithms.

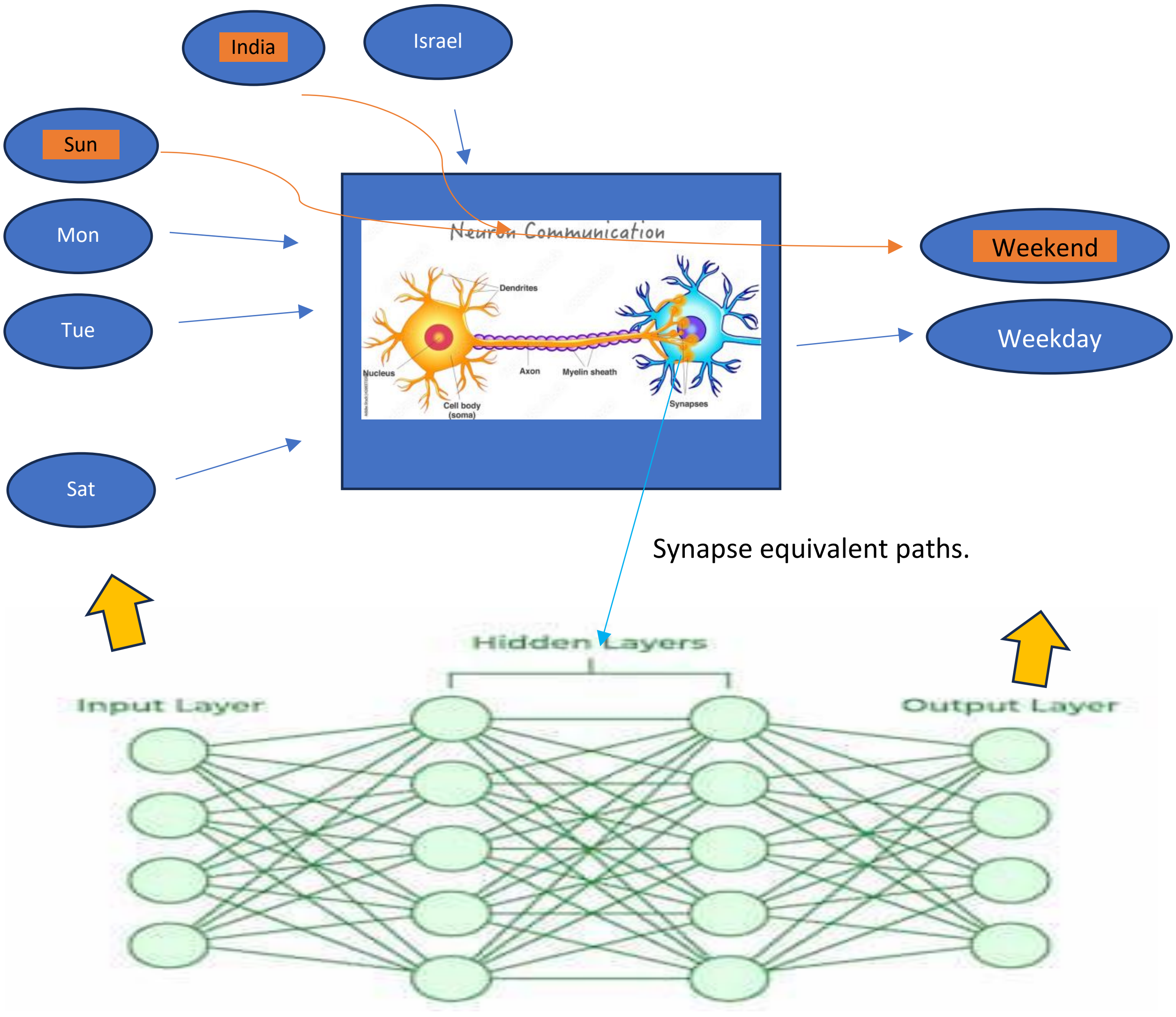
In Blog 1, we covered the need for an optimal computing machine for AI processing and explored basic architectures such as CISC and RISC machines. Now, let's delve into the fundamental processing building blocks required for AI processing.

Studying cognitive processing and associated brain's decision-making neural networks, has paved the way for the creation of structured mathematical algorithms, tailored for building AI models. Neural networks in the brain have billions of interconnected neuron structures, scaling up to create formidable processing capabilities. This network of neurons forms a wonderful decision-making tree. The same concept is used in digital structures for processing information for decision-making.

Let us see this decision-making process with a simple example in cognitive world Vs Digital equivalent implementation. Let us say there are 7 inputs indicating days of the week and from our model we are looking for answer if any specific selected day is weekday or weekend. In the brain's neural network, there are neuron representing the trigger points for specific input conditions. When input neurons get triggered, it triggers corresponding next level of neuron through intermediate pathways. In our example, if day is Sunday, it will activate neuron belongs to Sunday, which will have established path to weekend and hence Sunday neuron will activate to weekend neuron. These paths are called synaptic pathways. These are strengthened through repeated passing of electrochemical signal transmission through it. There will not be any pathway from Neuron belong to Monday to Weekend. This pathway strengthening takes place by repeated learning process.

Similarly, in the digital realm, neurons are modelled by data structures and pathways between neuron data structures are controlled through weights as multiplying factors. These weights range from 0 (indicating no connection) to 1 (representing a very strong connection), with intermediate values indicating different levels of connection strength. Like brain's learning, we iterate through the network multiple times, comparing output values to desired outcomes. Adjusting weights eventually achieves the desired outcome, finalizing the data model for the specific problem at hand. If we add one more variable of country in our problem. It will add one additional layer of neurons to get answer. *(Here, the goal is to understand a simple concept and observe how complexity begins to build upon it.)*

Brain neurons establish connections by strengthening synapses through repeated learning, incorporating information such as distinguishing between weekdays and weekends. Activation of a neuron corresponding to a specific day triggers the illumination of connected output neurons that represent weekdays or weekends in the network.



Analogy between Brain's neural network learning process Vs Digital data structure and algorithms

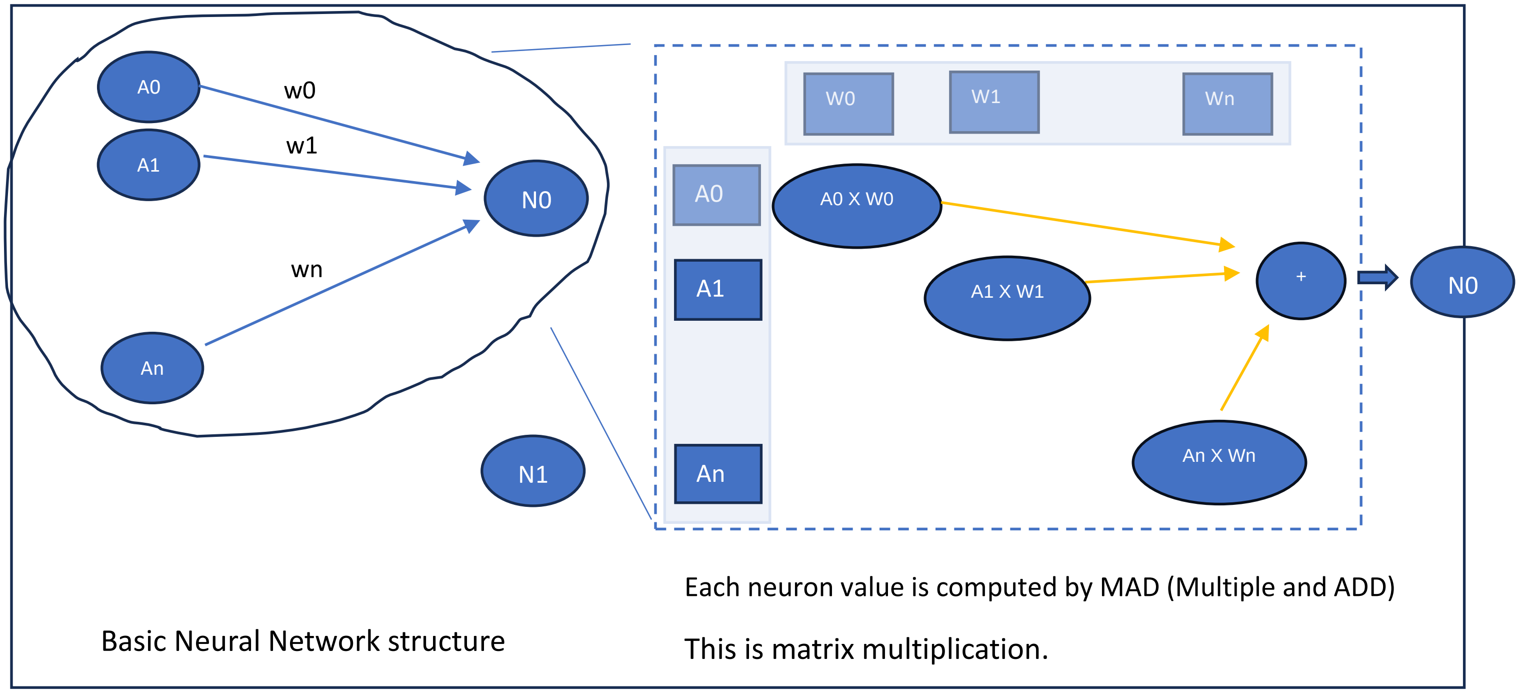
Each oval block is representing a neuron.

Many neural structures and algorithms are evolving:

Various neural network implementation structures vary based on factors like layer count, weight calculation method (backward or forward propagation), hierarchical learning techniques, and more. Examples include Feedforward Networks, Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) etc. These networks evolve according to specific applications; for instance, some specialize in image recognition while others excel in natural language processing. However, fundamentally, all neural network models rely on iterative computation to determine neuron values. **And hence it is our fundamental building block in designing efficient processing architectures**.

Basic building block of processing value for Neuron involves matrix multiplication.

Each neuron's value is systematically computed based on the values from the preceding neuron layer multiplied by their associated weights (which represent the probability of influencing the current neuron's value). Essentially, this process across all neurons in the input layer boils down to matrix multiplication and addition operations i.e. Sum of (input Value[m] * Weight[n])



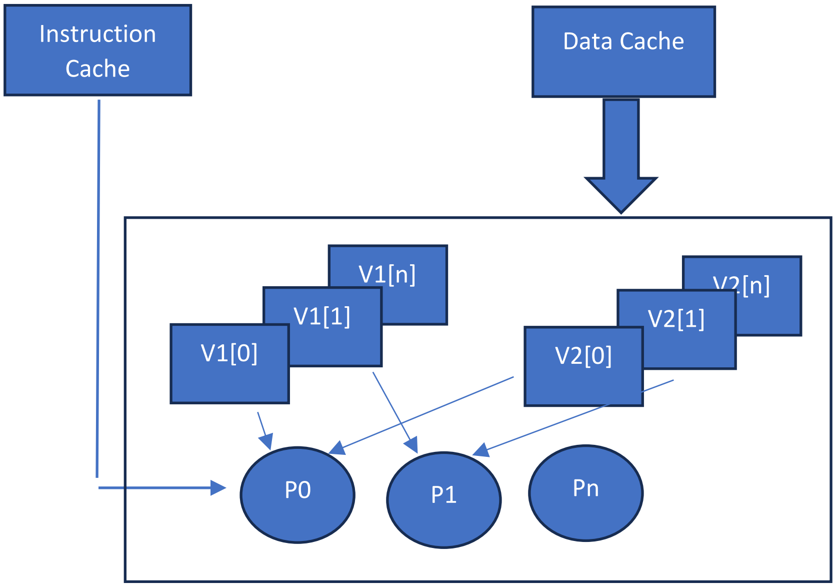
Basic Neural Network structure

Each neuron value is computed by MAD (Multiple and ADD)

This is matrix multiplication.

Single neuron value computation

If we execute each operation for 2 data elements sequentially using a scalar (sequential) processor then it will take large number of cycles. But if we execute them in parallel for multiple data elements (such as Value[m] * Weight[n]) with a single instruction then it will save significant amount of time and power. Hence vector (multiple data elements in parallel) processing becomes more efficient.



We need such large vector processing blocks in parallel to compute values of multiple neurons simultaneously. This scalability demand requires processing capabilities in the order of TeraFLOPS per second. Typically, this results in thousands of processing engines running in parallel.

We know from previous blog that CISC/RISC machines architecture do not offer vector (multiple data elements in parallel) processing. Whereas GPU is vector processor and does scale very well for this requirement. Hence it brings efficient solution for AI processing.