Design of Neural Networks

Pam Mantri¹, John Thomas¹
¹Cognitive Tools Ltd. LLC, P.O. Box 695; 255 North Ave; New Rochelle, NY 10801, USA
E-mail: pammantri@cogtools.com

Abstract. Artificial Neural Networks (ANN) and Deep Learning (DL) are used to solve complex problems including image recognition, speech recognition and have applications in new technologies for autonomous driving, facial recognition, detecting cancers from imaging samples among others. Various design considerations are involved in the design, training, and testing of Artificial Neural Networks (ANN). These include the design of the input/output layer, the structure and number of hidden layers, the data/data-structures of variables, the transformative functions embedded in the network, the optimizers being considered, the learning rate and its systematic adjustment, the prudent usage of dropout, the parallelism-related batch-size, the number of epochs, the adaptive logic for systematically changing the network for better fit, etc. While all these methods and techniques are sensible and relevant, there lacks an overarching framework for the needed design. This paper considers the design of an ANN from an Axiomatic Design (AD) perspective that parallels the biological inspiration for ANN’s in the first place, i.e., the brain. The axiomatic design approach is used for explicating and extricating the form, function, and adaptive evolution of the underlying network.

Keywords: Artificial Neural Networks, Biological Neural Networks, Axiomatic Design, Deep Learning, Perceptron

1. Introduction
In less than a hundred years, work on Artificial Neural Networks (ANN) has progressed to the extent where ANNs are used in a multitude of disciplines. ANNs can recognize objects, translate speech to multiple languages, convert voice to text and vice-versa, identify cancer lesions from imaging studies, predict stock prices, beat humans at games like chess and GO, assist in autonomous driving to name just a few applications.

In essence, a Neural Net consists of input variables that through a series of intermediate hidden layers, each containing hidden nodes, can output values for the output variables. Research over the years has focused on techniques to make the ANN model accurate, generalized, efficient, robust in the problems it can solve and the solutions it predicts. The design of the Artificial Neural Net plays an important role in addressing this. Design architecture includes considerations of the input/output layer, the structure and number of hidden layers, the data/data-structures being submitted and its appropriate scaling, the transformative functions embedded in the network, the optimizers being considered, the learning rate and its systematic adjustment, the prudent usage of dropout, the parallelism-related batch-size, the number of epochs, the adaptive logic for systematically changing the network for better fit, use of convolutions and a multitude of other architectural considerations. The design of ANNs happens more at the abstract, architectural level instead of at the fine-grained detailed level. This is because of the
self-organizing property of the ANN’s at the fine-grained levels. The system is supposed to fill in the fine-grained details in a coordinated orchestration of self-organization.

Whether artificial or biological in origin, neural networks exhibit a hierarchical structuring which agrees well with Axiomatic Design (AD). This paper catalogs some key historic milestones in ANNs and traces the hierarchical “form-follows-function” (i.e., Design Parameter follows Functional Requirement, DP-FR) mapping of the ANN logic.

2. History of Artificial Neural Networks
2.1. Early foundational work
The history of ANNs is less than a hundred years. It can be traced to the work of McCulloch and Pitts [1], in 1943, on threshold logic-based computational model for basic OR/AND/NOT functions. Using this model, binary inputs result in an output of 1, when the input sum exceeds a threshold, otherwise, the output is zero. Another key insight was the Hebbian learning model, which in 1949 postulated synaptic firing to be the mechanism by which the brain learns (i.e., “neurons that fire together wire together”). Adaptively changing firing patterns impacted the weight of the neuronal connection, thus providing synaptic plasticity [2]. In 1957, Rosenblatt conceived the Perceptron model to explain how the neuron in the brain might operate [3]. He then prototyped the Perceptron idea in hardware and applied it for the classification of 20 X 20-pixel data. The perceptron creates a linear-weighted aggregate of all input signals (similar to linear regression) and modulates it with a non-linear activation function. A single perceptron classifies all the inputs it receives into the binary state of zero or one. For classification into multiple categories, a single layer of perceptrons may be used such that each perceptron receives all the inputs but was responsible only for one part of the classification. This is similar to ANN. A few years later in 1960, an early example of adaptive “ADALINE” neuron [4] using chemical “memistors” was proposed by Widrow. It was an ANN without a thresholding function which made it differentiable. Calculus could then be used to minimize the error. In 1962, Hubel and Wiesel published their theory of the architecture of hierarchical convergence of simple and complex cells in the cortical cells of the cat brain, during visual perception [5]. The work on neural basis of visual perception has inspired many neural networks (NN) and deep learning (DL) architectures.

2.2. AI Winter
Early breakthroughs of perceptrons led to hyped expectations in the media about outlandish futuristic possibilities from Artificial Intelligence (AI). But in 1969, Minksy and Papert [6] published an influential, critical, and skeptical view about perceptrons. Minsky/Papert showed that a single perceptron layer of the Rosenblatt algorithm could not represent the XOR computation. Single layer perceptrons were computing the weights by adjusting them based on the expected correct output. For XOR to work with perceptrons, multiple layers would be required. Many (including Minksy/Papert) knew that multiple layers could solve the XOR problem. But there lacked a self-organizing algorithm to help train the ANN. The backpropagation algorithm that could train the ANN was invented 17 years later. Over the next few years, a period of AI winter ensued.

For ANNs, since the supervised method only specifies the correct final output, the correct weights expected for the layers prior to the output layer needed to be determined. The method for that computation had not yet been identified. The presence of multiple layers, some of which are hidden, is of importance for finding features in the data. Prior to the use of Neural Nets, feature engineering was used to extract relevant features for the model. The great thing about hidden layers is that features can be extracted from the data by the algorithm. So, a leap of computational efficiency was achievable, if the key to determining the weights of the hidden layers could be addressed.
Backpropagation was identified in the early 1970s by Linnainmaa [7] and postulated to be useful in neural nets by Werbos [8] in 1974. It uses the chain rule from calculus to compute the gradients for the prior layers, as well as weights for the intermediate layer under computation. Not many researchers embraced this finding right away due to the stigma for Neural Nets during the first AI winter. However, a decade after the discovery of backpropagation, in 1986, Rumelhart, Hinton and Williams, used backpropagation [9] and addressed the problems raised by Minsky to decisively give NNs a boost of life.

2.3. AI Spring

From the 1980s to the 2000s research on Neural Nets takes off again. Inspired by the work of Hubel and Wiesel’s hierarchy model of the visual cortex, Fukushima, in 1980, proposed a neural network called Neurocognitron [10]. The goal of the network was to use self-organization to auto-detect patterns, and to recognize patterns that were not impacted by a change in position or small distortions. Backpropagation which was already discovered in the 70s was now being applied to neural nets. Once backpropagation was used to train NNs successfully, ambitious targets were set, including the use of Multilayer feedforward networks as universal approximators in 1989.

Fukushima (in 1980) and Mozer (in 1987), found that since features in an image could be at any location, it was important to preserve the approximate position to ensure that higher-order complex features could be detected [10]. This moved the design from backpropagation to deep learning. The first hidden layer of the NN was convolutional. Convolutional Neural Networks
(CNNs) were inspired by the neuronal architecture of the animal visual cortex where cortical neurons respond to their corresponding receptive field which is a part of the visual field. There are many such receptive fields, and they cover the entire visual field by partially overlapping with each other. Layers past the first hidden layer also work in a similar way but they take local hidden features from the previous layer to construct larger subsets of the image to arrive at patterns. The final layers use normal neural nets to discern higher-order larger features generated by convolutional layers to discharge the task of classification. This methodology was used by LeCun et. al. from AT&T Bell Labs, to achieve a major milestone for ANN/CNN, demonstrating that neural nets and backpropagation could be used by a computer to detect and classify handwritten digits [11].

New research ideas were explored in the domain of Energy-based models. Inspired by statistical physics and thermodynamics, Hinton and his team, introduced Boltzmann Machines, wherein a simple algorithm is used to discover features in data that contained binary vectors [12]. These are neuron-like and make probabilistic decisions using symmetrical connections about being on/off. A network of Boltzmann machines forms graphical models and resembles NN with hidden units and hidden variables. However, a network of Boltzmann machines did not have anything corresponding to layers. One problem with these was the speed of training. In 1992 Neal introduced belief nets which had directed forward connections and could be trained faster than Boltzmann’s machines [13]. Despite being faster than Boltzmann machines, the computation was still slow and newer approaches were investigated. In 1995, Dayan, Hinton, Neal and Zemel came up with the Helmholtz Machine, wherein separate weights were used for inferring hidden variables from visible variables, as well as the reverse - inferring visible variables from the hidden variables [14].

Separately in the 1980s and 1990s research was also progressing on the use of NNs via reinforcement learning to perform complicated tasks and for playing games like chess and Go. At that time, even in the better cases, hardware limitations prevented NNs from scaling and providing truly forward-looking optimal results.

Speech recognition was another area where active research was ongoing. Speech is challenging because, unlike a batch of images in computer vision, language involves a temporal sequence of spoken words. Yet another problem with speech, is the innumerable pronunciations possible. Work by Waibel et. al (in 1989) looked at using a moving window to process a temporal input subset, and then using the same computation for each sequence of words processed. This work led to the development of time-delay neural networks (TDNN) [15]. Separately, Recurrent Neural Networks (RNNs), where the links form a temporal directed graph, are better suited for this task type, as they can leverage the information in past input weights by using cycles in the graph of the network. Additionally, RNNs can be adaptive to the information being processed.

TDNNs are similar to CNNs, however the window for CNN moves across the entire input image, rather than being time-based. CNNs have become the NNs of choice for images. Even though RNNs outperformed static networks, Bengio (in 1993) found that RNNs worked poorly with long-term dependencies, because they settled in suboptimal solutions with short-term preferences. This limited RNN’s use in learning applications that had long-term dependencies such as language [16].

Backpropagation worked well on NNs with few layers, but with deep NNs the problem of vanishing or exploding gradients was encountered. This results from the cumulative error being backpropagated, which unfortunately increased or decreased exponentially with the addition of layers. The exploding/vanishing gradient problem was encountered for ANNs and RNNs, with multiple time layers, that used backpropagation to compute intermediate weights. A breakthrough to solving the problem for RNNs was achieved, in 1997, by Hochreiter and Schmidhuber, with work on Long Short-Term Memory (LSTM). Using this technique, select units of the network, use an identity activation function, and a self-loop with a weight of one, to
deal with the vanishing gradient problem [17].

2.4. Second AI Winter
Despite the breakthrough with LTSM, the perception of the limitations of backpropagation and RNNs and the effectiveness of other methods such as Support Vector Machines (SVMs) and Random Forests, coincided with a second AI winter for neural networks resulting in the drying of research funding in the 90s.

2.5. Second AI Spring
In 2003, Bengio published Neural Probabilistic Language Model to model language using NNs [18]. NN research had encountered resistance during the second AI winter, and in 2006, Hinton and others rebranded NNs as DL. In the same year, Hinton and his colleagues also published an algorithm to train deep belief nets faster [19]. They used Restricted Boltzmann machines to initialize weights instead of using random weights. In 2007, Bengio and Hinton showed that deep learning methods were useful for solving difficult problems, that were difficult to solve using shallow networks or SVMs.

In 2006, a few publications by the groups of Hinton, Bengio and LeCun illustrated how DL could learn high-level representation of the data with low error rates. To do this, some methods used multiple layers of computing units using simple gradient descent applied to convolutional networks. Another method used layer-by-layer unsupervised learning, followed by gradient descent. Overfitting was addressed using dropout (i.e., randomly subsampling the data). Fueled by new research ideas, and the confluence of big data and computational power of GPUs there was a new impetus for NN, DL, and the overall field of AI to take off again in the twenty-first century.

ANNs are extremely data hungry. To make full use of their computational power, substantial amounts of data are needed. Whereas MNIST (a database of handwriting digits) was useful for training models for handwriting recognition, newer datasets such as ImageNet by Fei Fei Li’s group were published in 2009 [20]. Currently, ImageNet has over 14 million images in over 20,000 categories, which are then further divided into subcategories. But in 2009, it was one of the largest datasets available for training, and therefore a key asset that helped shape the CNN revolution.

Another piece of the puzzle was the required computing power. Since the 1990s computers were getting faster as per the Moore’s law. But the CPU parallelism required for computing millions of weights for the nodes in the multitude of layers of the CNNs exceeded what could be effectively done using CPUs. In 2009, Andrew Ng’s group reported using GPUs for obtaining multiple orders of magnitude faster computation [21].

In 2010, more insights were gained into the reasons why backpropagation produced vanishing and exploding gradients. Glorot and Bengio identified that the random weights, per se, were not an issue, but rather the logistic sigmoid or hyperbolic tangent activation functions were unsuitable for deep networks [22]. Rectified linear unit (ReLU) activation function was found to be useful for preventing vanishing/exploding gradients. ReLU produces a sparse representation, and only key neurons have a non-zero output for an input, leading to computational efficiency, and parsimonious and robust information.

By 2011, in addition to research in the universities, the power of big data and deep learning was recognized by companies such as Microsoft, Google and IBM. Andrew Ng and Jeff Dean from Google formed Google Brain to build out a huge computational resource of 16,000 CPU cores that could train 1 billion weights. This was three orders of magnitude more than the previous Deep Belief Network (DBN) breakthrough that trained 1 million weights. Separately, thus far, many applications used supervised learning, but one of the applications by Google Brain trained YouTube videos that were not labeled. Using unsupervised learning, at scale, made sense since
far more unlabeled data exists today as compared to labeled. In 2012, Andrew Ng and Jeff Dean
published research from Google brain to detect cats using high-level features and unsupervised
learning [23, 24].

Over the next few years previous ImageNet competition benchmarks were beaten, first in 2012
by AlexNet [25] using a CNN architecture by Krizhevsky, Sutskever and Hinton and having a
top-five error rate of 17% - a 9% improvement over the previous year’s benchmark. AlexNet was
larger and deeper than previous ANNs and used a stack of CNNs on top of each other. Two years
later, GoogLeNet [26], by Szegedy et al from Google, won the 2014 ImageNet competition with a
top-five error rate of less than 7%. In 2015, ResNet or Residual Network developed by Kaiming
He [27] won the ImageNet challenge having a top-five error rate of less than 3.6% (better than
human performance estimated at 5.1%). Its architecture included an extremely deep CNN with
152 layers. Some of these highlights of neural network history are captured in Figure 1.

In less than a hundred, starting at the nascent beginning’s the field of AI, including ANN and
leading to DL, has made huge strides in terms of the performance for games such as chess, GO,
identifying objects with better than human performance, RL, etc. Despite many breakthroughs,
there are limitations in terms of what exactly it is that the model learns, leading to brittleness as
demonstrated by Adversarial examples in Machine Learning. Humans observe and perceive the
world, form concepts as they understand it, and integrate their knowledge using the conceptual
building blocks and a process of integration. The concepts we form are rich and span many
dimensions. In contrast, research efforts continue to discern exactly what a machine learns.

Work is being done in the space of “common sense” learning and Artificial General Intelligence
(AGI). Despite the limitations of AI/ANN/DL, the field is embraced in almost all disciplines and
applications exist in humanities, genomics, medicine, autonomous driving, finance to name just
a few fields. Frameworks such as PyTorch (initial release in 2016), Tensorflow (first released
in 2015, with v1.0 in 2017) have been developed making these technologies easily accessible
to researchers and practitioners. Mimicking biological neural organization, Hinton, in 2017,
proposed Capsule Neural Network to better model hierarchical relationships [28]. Attention
models [29] and Transformer NN [30] proposed in 2015 and 2017 provided further breakthroughs
in the NLP and language translation. More recently transformer-based models such as XLNet,
RoBERTa, GPT-2, and ALBERT have broken previous language benchmarks [31, 32].

Research continues to advance the technology and make the intelligence discerned by the
machine comparable to intelligence present in living organisms.

3. Biological Neural Networks (BNN)

Biological Neural Networks (BNNs) started evolving around 600 million years ago. BNNs consist
of neurons connected functionally. Neurons are highly specialized cells transmitting information
unidirectionally from one part of the body to another. A neuron consists of dendrites, a cell
body, and a single axon. The axon terminates at the axon terminus (Figure 2). The end of
a neuron can communicate with the dendrites of another neuron through synapses. Neuronal
signals are excitatory or inhibitory and are transmitted along the axon using action potentials
created by a chemical gradient of sodium and potassium charged ions.

Action potential is an all-or-nothing event that can be considered as a binary signal [33]. The
strength of the signal is governed by the number of signals firing, their timing, the location and
whether the signal is inhibitory or excitatory. Once the signal reaches the end of the neuron, it
communicates with other neurons across the synapse. At the synapse, chemical neurotransmitters
are released by the presynaptic neuron, and the signal is passed on to the postsynaptic neuron.
Neurons transmit their signals in one direction and can transmit signals over large distances (as
for example, in the case of sciatic nerves). There are many types of neurotransmitters that are
used in the synapse. They may be excitatory or inhibitory such as epinephrine, norepinephrine,
acetylcholine, serotonin, γ-aminobutyric acid (GABA), dopamine to name a few. The signal is
The 14th International Conference on Axiomatic Design (ICAD 2021)
IOP Conf. Series: Materials Science and Engineering 1174 (2021) 012023
doi:10.1088/1757-899X/1174/1/012023

Establish Rapid, Fine-Grained Coordination & Control

Establish Integrative Coordination & Control

Establish Autonomic Coordination & Control

Establish Somatic Coordination & Control

Establish Centralized Integrative Functions

Establish Distributed Integrative Functions

Establish Afferent Autonomic Coordination

Establish Efferent Autonomic Coordination

Establish Afferent Somatic Coordination

Establish Efferent Somatic Coordination

FR1  X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
FR11 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
FR12 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
FR13 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
FR14 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
FR15 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
FR16 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
FR17 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
FR18 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
FR19 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
FR20 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
DP1  X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
DP11 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
DP12 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
DP13 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
DP14 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
DP15 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
DP16 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
DP17 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
DP18 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
DP19 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X
DP20 X  X  X  X  X  X  X  X  X  X  X  X  X  X  X

Figure 2. Structural components of Neuron

Figure 3. Hierarchic Structure of Biological Neural Systems & Subsystems

terminated by degradation or reuptake of the neurotransmitter.
Functionally, there are three types of neurons - Sensory neurons that carry signals from sensory organs such as eyes, tongue, skin etc to the brain or spinal cord, motor neurons that receive signals from the brains and spinal cord and effect muscle contraction or secretory organs and finally, interneurons connecting neurons to other neurons.
Research estimates there are eighty-six billion neurons in the human brain [34]. Of these,
the cerebral cortex alone contains 16 billion neurons linked by about 100 trillion synaptic connections [35].

Primate neural systems are hierarchically organized [36] as shown in (Figure 3). Neural controls are fine-grained and execute rapidly and with high specificity. As discussed earlier, electrical signals (within the neuron) as well as chemical neurotransmitters (between neurons) are used for signaling. In contrast, endocrine controls are coarse-grained (i.e., they have a widespread effect) and use hormones sent via the circulatory system (for example, pancreas that controls the uptake of sugar via insulin secretion).

There are two distinct hierarchies when considering the inherent structure of any of the design-matrices: 1) cross-level, decompositional versus 2) in-level, dominance. For example, when we consider the reverse-engineered, primate neural-system design trace (Figure 3), the decompositional hierarchy is as shown in the tree diagram at the bottom. Each level of this decompositional structure is in effect, the total system. Thus, the totality of the nervous system at level 1 \((FR_1:DP_1)\) is equally captured as the \({{FR_{11}, FR_{12}, FR_{13}}: {DP_{11}, DP_{12}, DP_{13}}}\) complex that pertains to the Central Nervous System (CNS), the peripheral Autonomic Nervous System (ANS), as well as the peripheral Somatic Nervous System (SNS). CNS decomposes at the next level into the brain & spinal cord. Automated ANS controls include heart rate modulation, blood pressure modulation, digestion, etc. It occurs below the level of consciousness. Voluntary SNS controls include posture, motion, sense organs (odor, temperature, taste, vision, pain), etc. It occurs within conscious awareness. Within each level, there also exists a dominance hierarchy. Thus, the CNS occupies the dominant (top-left) location within the decoupled green sub-region of the design matrix. This is followed by ANS, and only then by SNS. In-level dominance may be traced by considering the underlying functions that are being performed. Damage to the brain has a widespread effect on the body; in contrast, damage to the peripheral nervous system (i.e., ANS/SNS) generally has a limited, local effect. However, this is not always true; for example, heart-rate modulation failure can directly impact CNS brain function, thereby reversing the in-level dominance. Damage to the brain has a widespread effect on the body; in contrast, damage to the peripheral nervous system (i.e., ANS/SNS) generally has a limited, local effect. However, this is not always true; for example, heart-rate modulation failure can directly impact CNS brain function, thereby reversing the in-level dominance. But by and large, the dominance pattern in most cases would be CNS > ANS > SNS.

Now consider how the signaling occurs across the neural architecture. Imagine a hiker who is traversing a wooded mountain landscape as shown in Figure 4b. Suddenly she hears rustling of leaves followed by loud threatening animal noises from behind the trees. These noises trigger her DP_{131} (i.e., her Somatic-Voluntary-Sensory hearing controls) to send afferent alarm signals up the chain of neural hierarchy, and into the emotion-processing CNS component, the amygdala. The amygdala integrates the threat signal with its coded past experience (which also includes evolutionary information) and triggers the adjacent hypothalamus that commands the autonomic nervous system into the “fight-or-flight” mode. Her heart rate increases to supply the blood vessels with increased oxygen. The circulatory system that services the body musculature is dilated while that which supplies the digestive system is constricted. This action diverts the blood to the muscles where it is urgently needed. Eyes dilate to increase light capture and improve visual acuity for better threat perception. The bronchi in her lungs dilate to increase the oxygen capture. In effect, many of her DP_{122} controls (i.e., her Autonomic-Involuntary-Motor controls) are now fully geared up for the “fight-or-flight” action.

Consider now the visual threat processing by the hiker. The visual cortex processes visual input and has inspired many NN, DL and CNN architectures. The entry point for the visual signal is the eye. Here, the five cell types in the retina convert photons into electrical signals. It starts at the photoreceptors and ends in Retinal ganglion cells (RGC). RGCS are involved in edge extraction from an image. The axons of the ganglion cells in each eye use the optic nerve to transmit signals to the primary visual cortex (V1) via the lateral geniculate nuclei (LGN).

V1 is responsible for the first stage of image processing. V1 has six layers that receive inputs
Receptive field
Local context
Global context

A
B

Figure 4. A: Layers of primary visual cortex inputs B: Visual scene with Receptive field, Local and Global surround

from different parts of the brain. These layers are stacked and are hierarchically connected. Receptive fields in layer 4c of V1 are activated in response to the smallest stimuli receiving input from LGN (Figure 4A) [37]. This is depicted using the red circle in Figure 4B. To understand the visual scene being processed, the local context is created by long-range horizontal connections within V1 (green circle in Figure 4B) that add to the information of the receptive field. Higher-level cortical structures such as V2, V3 and MT that receive higher-order visual information directly or indirectly from V1, also provide spatially extensive feedback to layers 1 and 6 of V1, thus enabling it to construct the global information of the scene (colored blue in Figure 4A, B).

4. Evolution of the Multi-Layer Perceptron
As mentioned in the historical review, in 1958 Rosenblatt proposed the perceptron model to emulate the neural computing machinery. He was then the director of the Cognitive Systems Research Program at the Cornell University Ithaca campus. In 1960, the model that Rosenblatt proposed was implemented in hardware as Mark 1 Perceptron. On July 8th, 1958, the U.S Office of Naval Research demonstrated the machine. After a few trials, the machine was able to separate out punched cards marked on the left versus those on the right. Based on this demonstration, The New York Times [38], reported that “[it] learned to differentiate between right and left after fifty attempts. The Navy said the perceptron would be the first non-living mechanism ‘capable of receiving, recognizing, and identifying its surroundings without any human training or control.’”

A decade later, in 1968, Minsky and Papert [6] evaluated the perceptron model and highlighted certain edge-case deficiencies (such as the replication of the XOR gate, or the issue of recognizing connected figures). The Minsky/Papert critique was based on a simplified model (i.e., a single layer neural network) of Rosenblatt’s original perceptron which was not likewise limited. But the mathematical rigor of the critique was sufficient to dry up the funding for Rosenblatt’s approach that attempted to emulate the nervous system in a bottom-up fashion of sensation to perception to conceptions. Instead, post-Minsky/Papert critique, research funding moved into a top-down symbolic approach to AI. This later led to the AI winter as the top-down symbolic approach ran
into its own difficulties. It has taken 60 years to recognize the pioneering foundations laid by Prof. Rosenblatt. As recounted in [39] by the faculty at Cornell: “Today, many believe Rosenblatt has been vindicated. The principles underlying the perceptron helped spark the modern artificial intelligence revolution. Deep learning and neural networks - which can classify online images, for example, or enable language translation - are transforming society. ‘The perceptron was the first neural network,’ said Thorsten Joachims, Professor in CIS”.

In this section, we review the perceptron logic and link it up with the Axiomatic Design (AD) approach. Fundamental principles of neural network design may be gleaned by reviewing Prof. Rosenblatt’s perceptron design using the AD framework. The focus is on the bottom-up formation of the FR-DP hierarchy as it relates to the XOR/XNOR roadblocks that the Minsky/Papert critique highlighted. As reported in [40] by Mengistu et al: “Hierarchical organization—the recursive composition of sub-modules—is ubiquitous in biological networks, including neural, metabolic, ecological, and genetic regulatory networks, and in human-made systems, such as large organizations and the Internet.... However, an open, important question in evolutionary biology is why hierarchical organization evolves in the first place.” By including the cost of connection along with performance in solving an AND-XOR-AND via ANN, they were able to show that [40]:

(i) Networks that do not price connection costs do not evolve to be hierarchical.
(ii) But with connection-costs being priced in into the build-up of the ANN, the “networks evolve to be both modular and hierarchical.”
(iii) They exhibit higher “overall performance and evolvability (i.e., faster adaptation to new environments).”
(iv) Also, “hierarchy independently improves adaptability after controlling for modularity.”
(v) In other words, minimizing “the cost of connections-promotes the evolution of both hierarchy and modularity.”

But from an AD perspective, the cost of network connection is a proxy for system-level information content as captured in Axiom II. This seems to suggest that Axiom II can give you both modularity and hierarchy; if so, why then do we need Axiom I? In other words, is Axiom I redundant perhaps? And the answer is no. This is because it is Axiom I that helps bridge the is-ought gap between the FR and DP domains. Without the functional domain, form is meaningless. Also, with hierarchy comes the need for cascade induction. Again, without Axiom I, we cannot do the necessary abductive cascade along the design trace (See Figure 5). This agrees with the design hierarchy as well as the zig-zag rule between the FR and DP domains as outlined in AD.

Axiomatic Design highlights the hierarchical structure of design. As Prof. Suh highlights in two separate instances in [41]:

- “Everything we do in design has a hierarchical nature to it. That is, decisions must be made in order of importance by decomposing the problem into a hierarchy.... When such a hierarchical nature of decision making is not utilized, the process of decision making becomes very complex.”
- “The designer must recognize and take advantage of the existence of the functional and physical hierarchies. A good designer can identify the most important FRs at each level of the functional tree by eliminating secondary factors from consideration. Less-able designers often try to consider all the FRs of every level simultaneously, rather than making use of the hierarchical nature of FRs and DP’s.”

Hierarchies often show up in both biological [42] as well as sociotechnical systems [43]. Capturing the functional roots of these hierarchies may be of significance.
In both BNN’s and ANN’s there are concrete artifacts that correspond to the higher-level design constructs. This is because, unlike the design of a hammer or a screw, these artifacts have adaptability, intelligence and control embedded in them. In other words, the whole design trace is actively being orchestrated and often being modified on an ongoing basis. For example, the central nervous system that sits at a higher level (as compared to the subordinate peripheral nervous system) is not just an abstraction on a conceptual hierarchy; it has a real biological presence. It is in this sense that such design traces may be considered as intelligent, dynamic and alive than being static, final and “once and for all” as is the case with an artifact such as a hammer that lacks embedded intelligence.
Since nature does not do conceptual induction (and therefore, neither cascade induction), could it not be said that Axiom I is redundant as far as nature is concerned? While it is true that nature does not indulge in conceptual induction, the Darwinian evolutionary principle of “survival of the fittest” captures the essence of Axiom-I’s “form follows function” (i.e., DP follows FR). But it is true that nature does not do this in a conceptual sense; instead, it does it painstakingly across billions of years of slowly evolving hit-and-trials, that uses and discards millions of species which have now come and gone. The amount of time it takes biological systems to evolve and adapt to changing natural conditions (even taking into consideration the punctuated equilibrium-style expedited evolution [44] takes around 100,000 years) is orders of magnitude more than what the conceptual approach offers. Using tools such as ANN, this is a definite cost that humans could help short-circuit via conceptual induction (i.e., both direct and cascade) in order to aid Axiom-I. One of the intriguing design questions that the above research as reported in [40] triggers is the following: Does the optimum number of hidden layers correspond to the optimum number of hierarchic levels? For example, with respect to the XOR/XNOR gate problem (which is discussed in detail, as below), it is established that it cannot be solved with a single perceptron layer; it needs at least two. But it may also be solved with more than two layers. Thus, (similar to Occam’s Razor) is there a corollary to Axiom II that suggests that we minimize the number of hierarchic levels to the barest minimum required?

With the above context in mind, let’s now return to the bottom-up formation of the FR-DP hierarchy as it relates to the XOR/XNOR roadblocks that the Minsky/Papert critique highlighted. A perceptron is a computing element with a set threshold $\theta$ and $n$ inputs $x$: $x_1, x_2, \ldots, x_n$, each of which is associated weights $w$: $w_1, w_2, \ldots, w_n$ such that:

- If $w \cdot x \geq \theta$, then the perceptron outputs 1 (i.e., it fires)
- If $w \cdot x < \theta$, then the perceptron outputs 0 (i.e., it is dormant)

Geometrically, the perceptron separates the input space $x$ into two half-spaces; one part that triggers the perceptron firing while the other keeps it dormant. For a two-dimensional input vector, the 16 possible perceptron elements are as shown in Figure 6. To simplify the diagrams, the vectors $x_1$ and $x_2$ are limited to 0, 1 (i.e., they compose to form the four corner joints A, B, C, D). Thus, perceptron p2 is a straight line that separates out corner A (which is coloured green.
Figure 8. Coupled Single-Layer Perceptron XNOR Design

Figure 9. Two Layer Decomposed and Decoupled XOR Perceptron

to indicate that it fires) as against corners B, C & D (which are coloured red to indicate that they will not fire). For a three-dimensional input space, the separator perceptron is a dividing plane (and, so on). As indicated in the perceptron-grid in Figure 6), perceptron’s p7 & p10 are different in the sense that there is no linear element that could select the two green cornered diagonals on one side while leaving the rejected red corners to the remaining half. This is one of the problems that the Minsky/Papert critique highlighted.

When analysed using AD (as shown in Figures 7 and 8), both p7 and p10 turn out to be coupled designs with the added frustration that there is no feasible design-space.

But once the constraint of a single layer perceptron is lifted, it is now possible to design a two-layered perceptron by judiciously combining some of the 16 elements (as shown in Figure 9). In step 1, the required corners are appropriately selected. Then they are combined with a simple OR gate. As shown in Figures 9 and 10, the two designs are hierarchical and decoupled. The
The designs are hierarchical and arise out of the need to overcome the blocks that XOR/XNOR gates pose, is salient.

5. Conclusions
In their concluding remarks on Perceptrons, Minsky & Papert state the following [6]: “Theorems are theorems, and the history of science amply demonstrates how discovering limiting principles can lead to deeper understanding.”

In this paper, we wanted to show the strategic advantage of viewing the neural-network design from an AD perspective. Some of the useful insights and research directions about the design of neural networks (or for that matter, design in general) that may be gleaned from the above exercise include:

(i) Hierarchies in the design-trace are naturally triggered when existing designs exhibit inherent limits.

(ii) Tracking both form and function (i.e., DP’s & FR’s) as suggested by AD could aid the evolution of the design trace.

(iii) There may be overarching design laws & corollaries that provide advice regarding the minimum/maximum number of hierarchical levels.

(iv) The number of levels in the hidden layer of the Artificial Neural Network may be linked to the number of hierarchic layers in the design trace.

(v) How would the above correlate with the current understanding about hierarchies in Biological Neural Networks?

Items ii-v above need to be further explored. We hope to cover these in a follow-up report.

References
[1] McCulloch W S and Pitts W 1943 A logical calculus of the ideas immanent in nervous activity The bulletin of mathematical biophysics 5 115–133
[2] Hebb D O 1949 The organization of behavior: A neuropsychological theory (Science Editions New York)
[3] Rosenblatt F 1957 The perceptron, a perceiving and recognizing automaton Project Para (Cornell Aeronautical Laboratory)

Figure 10. Two Layer Decomposed and Decoupled XNOR Perceptron
[4] Widrow B 1960 Adaptive “adaline” Neuron Using Chemical “memistors”. (Stanford Electronics Laboratories Technical Report 1553-2)

[5] Hubel D H and Wiesel T N 1962 Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex The Journal of physiology 160 106–154

[6] Minsky M and Papert S A 1969 Perceptrons: An introduction to computational geometry (MIT press)

[7] Linnainmaa S 1970 The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors Master’s Thesis, Univ. Helsinki

[8] Werbos P 1974 Beyond regression: new tools for prediction and analysis in the behavioral sciences Ph. D. dissertation, Harvard University

[9] Rumelhart D E, Hinton G E and Williams R J 1986 Learning representations by back-propagating errors Nature 323 533–536

[10] Fukushima K 1980 Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position Biological Cybernetics 36 193–202

[11] LeCun Y, Boser B, Denker J S, Henderson D, Howard R E, Hubbard W and Jackel L D 1989 Backpropagation applied to handwritten zip code recognition Neural computation 1 541–551

[12] Ackley D H, Hinton G E and Sejnowski T J 1985 A learning algorithm for boltzmann machines Cognitive science 9 147–169

[13] Neal R M 1992 Connectionist learning of belief networks Artificial intelligence 56 71–113

[14] Dayan P, Hinton G E, Neal R M and Zemel R S 1995 The helmholtz machine Neural computation 7 889–904

[15] Waibel A, Hanazawa T, Hinton G, Shikano K and Lang K J 1989 Phoneme recognition using time-delay neural networks IEEE transactions on acoustics, speech, and signal processing 37 328–339

[16] Bengio Y 1993 Advances in Pattern Recognition Systems Using Neural Network Technologies (World Scientific) pp 3–23

[17] Hochreiter S and Schmidhuber J 1997 Long short-term memory Neural computation 9 1735–1780

[18] Bengio Y, Ducharme R, Vincent P and Jauvin C 2003 A neural probabilistic language model The journal of machine learning research 3 1137–1157

[19] Hinton G E, Osindero S and Teh Y W 2006 A fast learning algorithm for deep belief nets Neural computation 18 1527–1554

[20] Deng J, Dong W, Socher R, Li L J, Li K and Li F F 2009 Imagenet: A large-scale hierarchical image database 2009 IEEE conference on computer vision and pattern recognition (Ieee) pp 248–255

[21] Raina R, Madhavan A and Ng A Y 2009 Large-scale deep unsupervised learning using graphics processors Proceedings of the 26th annual international conference on machine learning pp 873–880

[22] Glorot X and Bengio Y 2010 Understanding the difficulty of training deep feedforward neural networks Proceedings of the thirteenth international conference on artificial intelligence and statistics (JMLR Workshop and Conference Proceedings) pp 249–256

[23] Dean J, Corrado G, Monga R, Chen K, Devin M, Mao M, Ranzato M a, Senior A, Tucker P, Yang K, Le Q and Ng A 2012 Large scale distributed deep networks Advances in Neural Information Processing Systems vol 25 ed Pereira F, Burges C J C, Bottou L and Weinberger K Q (Curran Associates, Inc.) https://proceedings.neurips.cc/paper/2012/file/6aca97005c68f120682381566102863-Paper.pdf

[24] Le Q V, Ranzato M, Monga R, Devin M, Chen K, Corrado G S, Dean J D and Ng A Y 2012 Building high-level features using large scale unsupervised learning Proceedings of ICML pp 81–88

[25] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks Advances in neural information processing systems 25 1097–1105

[26] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2015 Going deeper with convolutions Proceedings of the IEEE conference on computer vision and pattern recognition pp 1–9

[27] Xie S, Girshick R, Dollar P, Tu Z and He K 2017 Aggregated residual transformations for deep neural networks Proceedings of the IEEE conference on computer vision and pattern recognition pp 1492–1500

[28] Sabour S, Frosst N and Hinton G E 2017 Dynamic routing between capsules arXiv preprint arXiv:1710.09829

[29] Chorowski J, Bahdanau D, Serdyuk D, Cho K and Bengio Y 2015 Attention-based models for speech recognition arXiv preprint arXiv:1506.05503

[30] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L and Polosukhin I 2017 Attention is all you need Advances in neural information processing systems pp 5998–6008

[31] Yang Z, Dai Z, Yang Y, Carbonell J, Salakhudinov R R and Le Q V 2019 Xlnet: Generalized autoregressive pretraining for language understanding Advances in neural information processing systems 32

[32] Topal M O, Bas A and van Heerden I 2021 Exploring transformers in natural language generation: Gpt, bert, and xlnet arXiv preprint arXiv:2102.08036

[33] Luo L 2015 Principles of neurobiology (Garland Science)
[34] Herculano-Houzel S 2009 The human brain in numbers: a linearly scaled-up primate brain *Frontiers in human neuroscience* 3 31

[35] Azevedo F A, Carvalho L R, Grinberg L T, Farfel J M, Ferretti R E, Leite R E, Filho W J, Lent R and Herculano-Houzel S 2009 Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain *Journal of Comparative Neurology* 513 532–541

[36] Saladin K 2018 Anatomy and physiology: The unity of form and function

[37] Bijanzadeh M, Nurminen L, Merlin S, Clark A M and Angelucci A 2018 Distinct laminar processing of local and global context in primate primary visual cortex *Neuron* 100 259–274

[38] Times T 1958 New navy device learns by doing *Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser* https://www.nytimes.com/1958/07/08/archives/new-navy-device-learns-by-doing-psychologist-shows-embryo-of.html (Accessed 2021-04-04)

[39] Lefkowitz M 2019 Professor’s perceptron paved the way for ai - 60 years too soon *Cornell Chronicle* https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon (Accessed 2021-04-04)

[40] Mengistu H, Huizinga J, Mouret J B and Clune J 2016 The evolutionary origins of hierarchy *PLoS computational biology* 12 e1004829

[41] Suh N P 1990 *The principles of design* 6 (Oxford University Press)

[42] Mantri P and Thomas J 2019 Nature’s design’s: The biology of survival *MATEC Web of Conferences* vol 301 (EDP Sciences) p 00023 https://doi.org/10.1051/matecconf/201930100023 (Accessed 2021-04-14)

[43] Thomas J and Mantri P 2019 Axiomatic cloud computing architectural design *MATEC Web of Conferences* vol 301 (EDP Sciences) p 00024 https://doi.org/10.1051/matecconf/201930100024 (Accessed 2021-04-14)

[44] Gould S J 2009 *Punctuated equilibrium* (Harvard University Press)