RESEARCHERID
THOMSON REUTERS
[ICESTM-2018]
ISSN 2348 – 8034
Impact Factor- 5.070

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

DEEP NEURAL NETWORKS ON STATE SPACE DYNAMICAL SYSTEMS REPRESENTATION

RamGopaL MusunuRi[*1] & A.VanamaLa KumaR.[2]
[*1&2]Assistant Professor, Dept.of CSE, PVPSIT.

ABSTRACT

This paper manages neural systems as dynamical frameworks represented by dynamical systems governed by differential or difference equations. It demonstrates that the presentation of skip associations into network architectures, for example, residual networks and dense networks, turns an arrangement of static conditions into an arrangement of dynamical conditions with changing levels of smoothness on the layer-wise changes. Closed form solutions for the state space representations of general dense networks,, and also kth arrange smooth systems, are found by and large settings. Besides, it is demonstrated that forcing kth arrange smoothness on a system design with d-numerous hubs per layer expands the state space measurement by a numerous of k, thus the viable inserting measurement of the information complex is k d-numerous measurements. It takes after that system designs of these sorts diminish the quantity of parameters required to keep up the same inserting measurement by a factor of k2 when contrasted with an identical first-arrange, lingering system, altogether propelling the advancement of system models of these sorts. Numerical reproductions have been rushed to approve parts of the created hypothesis.

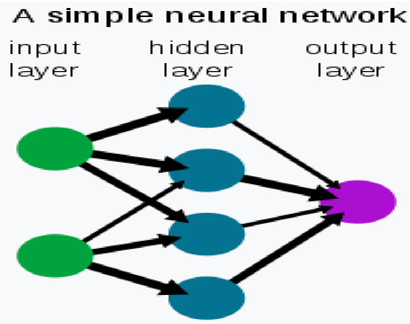
I. INTRODUCTION

The manner by which profound learning was at first used to change information portrayals was by settled Syntheses of relative changes took after by nonlinear initiations. The relative change is generally a task, for example, a completely associated weight grid, convolution or bunch standardization. Leftover systems [5] present a skip association that sidesteps these changes, in this way permitting the nonlinear initiation to go about as an annoyance term from the personality. Veit et al. [10] have appeared that these leftover systems can be comprehended as the whole gathering of all conceivable forward pass ways of sub networks. Late work reliable with the first instinct of taking in annoyances from the personality has demonstrated that lingering systems, with their first-arrange bother term, can be figured as a limited distinction estimation of a first-arrange differential condition [4]. This has the intriguing outcome that standard systems are static conditions, while lingering systems are dynamic conditions through the layers of the system.

Also, one may then characterize whole classes of Ck differentiable changes over the layers, and after that initiate arrange structures from the limited distinction approximations of these Ck differentiable changes. Numerical investigations in [4] demonstrated that remaining systems, i.e. systems with C1 differentiable changes, do in certainty easily change the portrayal of the information complex. It was too demonstrated that the C2 arrange design easily changes the facilitate portrayal of the information complex, and also permits the information manifolds to ignore each other amid the partition process. It is noticed that two manifolds lying in the plane can disregard each other if the plane is inserted in a higher dimensional space. Instinctively, this can be thought of as moving up a two dimensional bit of paper in three or higher measurements, and afterward anticipating the manifolds back down onto two measurements, and in the projection it looks as though the manifolds are disregarding each other. This impact can be comprehended as a C2 dynamical framework which has two state factors, in particular position and speed, though a C1 dynamical framework just has one state variable, specifically position.

This recommends the higher the request of smoothness forced on the system, the higher the compelling installing measurement ought to be.

ISSN 2348 – 8034
Impact Factor- 5.070



Simplified view of a feed forwardartificial neural network

The term **neural network** was traditionally used to refer to a network or circuit of neurons.[1] The modern usage of the term often refers to artificial neural networks, which are composed of artificial neurons or nodes. Thus the term may refer to either biological neural networks, made up of real biological neurons, or artificial neural networks, for solving artificial intelligence problems.The connections of the biological neuron are modeled as weights. A positive weight reflects an excitatory connection, while negative values mean inhibitory connections. All inputs are modified by a weight and summed. This activity is referred as a linear combination. Finally, an activation function controls the amplitude of the output. For example, an acceptable range of output is usually between 0 and 1, or it could be -1 and 1.

Unlike von Neumann model computations, artificial neural networks do not separate memory and processing and operate via the flow of signals through the net connections, somewhat akin to biological networks.

These artificial networks may be used for predictive modeling, adaptive control and applications where they can be trained via a dataset.

Overview

A biological neural network is composed of a group or groups of chemically connected or functionally associated neurons. A single neuron may be connected to many other neurons and the total number of neurons and connections in a network may be extensive. Connections, called synapses, are usually formed from axons to dendrites, though dendrodendritic synapses[2] and other connections are possible. Apart from the electrical signaling, there are other forms of signaling that arise from neurotransmitter diffusion.

Artificial intelligence, cognitive modelling, and neural networks are information processing paradigms inspired by the way biological neural systems process data. Artificial intelligence and cognitive modeling try to simulate some properties of biological neural networks. In the artificial intelligence field, artificial neural networks have been applied successfully to speech recognition, image analysis and adaptive control, in order to construct software agents (in computer and video games) or autonomous robots.

Historically, digital computers evolved from the von Neumann model, and operate via the execution of explicit instructions via access to memory by a number of processors. On the other hand, the origins of neural networks are based on efforts to model information processing in biological systems. Unlike the von Neumann model, neural network computing does not separate memory and processing.

Neural network theory has served both to better identify how the neurons in the brain function and to provide the basis for efforts to create artificial intelligence.

Neural networks and artificial intelligence

A *neural network* (NN), in the case of artificial neurons called *artificial neural network* (ANN) or *simulated neural network* (SNN), is an interconnected group of natural or artificial neurons that uses a mathematical or computational model for information processing based on a connectionistic approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network.

In more practical terms neural networks are non-linear statistical data modeling or decision making tools. They can be used to model complex relationships between inputs and outputs or to find patterns in data.

An artificial neural network involves a network of simple processing elements (artificial neurons) which can exhibit complex global behavior, determined by the connections between the processing elements and element parameters. Artificial neurons were first proposed in 1943 by Warren McCulloch, a neurophysiologist, and Walter Pitts, a logician, who first collaborated at the University of Chicago.[16]

One classical type of artificial neural network is the recurrent Hopfield network.
The concept of a neural network appears to have first been proposed by Alan Turing in his 1948 paper *Intelligent Machinery* in which called them "B-type unorganised machines".[17]

The utility of artificial neural network models lies in the fact that they can be used to infer a function from observations and also to use it. Unsupervised neural networks can also be used to learn representations of the input that capture the salient characteristics of the input distribution, e.g., see the Boltzmann machine (1983), and more recently, deep learning algorithms, which can implicitly learn the distribution function of the observed data. Learning in neural networks is particularly useful in applications where the complexity of the data or task makes the design of such functions by hand impractical.

The tasks to which artificial neural networks are applied tend to fall within the following broad categories:
- Function approximation, or regression analysis, including time series prediction and modeling.
- Classification, including pattern and sequence recognition, novelty detection and sequential decision making.
- Data processing, including filtering, clustering, blind signal separation and compression.

Application areas of ANNs include nonlinear system identification [18] and control (vehicle control, process control), game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems, face identification, object recognition), sequence recognition (gesture, speech, handwritten text recognition), medical diagnosis, financial applications, data mining (or knowledge discovery in databases, "KDD"), visualization and e-mail spam filtering.

Neural networks and neuroscience
Theoretical and computational neuroscience is the field concerned with the theoretical analysis and computational modeling of biological neural systems. Since neural systems are intimately related to cognitive processes and behaviour, the field is closely related to cognitive and behavioural modeling.

The aim of the field is to create models of biological neural systems in order to understand how biological systems work. To gain this understanding, neuroscientists strive to make a link between observed biological processes (data), biologically plausible mechanisms for neural processing and learning (biological neural network models) and theory (statistical learning theory and information theory).

Types of models
Many models are used; defined at different levels of abstraction, and modeling different aspects of neural systems. They range from models of the short-term behaviour of individual neurons, through models of the dynamics of neural circuitry arising from interactions between individual neurons, to models of behaviour arising from abstract neural modules that represent complete subsystems. These include models of the long-term and short-term plasticity of neural systems and its relation to learning and memory, from the individual neuron to the system level.

ISSN 2348 – 8034
Impact Factor- 5.070

Criticism

A common criticism of neural networks, particularly in robotics, is that they require a large diversity of training for real-world operation. This is not surprising, since any learning machine needs sufficient representative examples in order to capture the underlying structure that allows it to generalize to new cases. Dean Pomerleau, in his research presented in the paper "Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving," uses a neural network to train a robotic vehicle to drive on multiple types of roads (single lane, multi-lane, dirt, etc.). A large amount of his research is devoted to (1) extrapolating multiple training scenarios from a single training experience, and (2) preserving past training diversity so that the system does not become overtrained (if, for example, it is presented with a series of right turns – it should not learn to always turn right). These issues are common in neural networks that must decide from amongst a wide variety of responses, but can be dealt with in several ways, for example by randomly shuffling the training examples, by using a numerical optimization algorithm that does not take too large steps when changing the network connections following an example, or by grouping examples in so-called mini-batches.

A. K. Dewdney, a former *Scientific American* columnist, wrote in 1997, "Although neural nets do solve a few toy problems, their powers of computation are so limited that I am surprised anyone takes them seriously as a general problem-solving tool." (Dewdney, p. 82)

Arguments for Dewdney's position are that to implement large and effective software neural networks, much processing and storage resources need to be committed. While the brain has hardware tailored to the task of processing signals through a graph of neurons, simulating even a most simplified form on Von Neumann technology may compel a neural network designer to fill many millions of database rows for its connections - which can consume vast amounts of computer memory and hard disk space. Furthermore, the designer of neural network systems will often need to simulate the transmission of signals through many of these connections and their associated neurons - which must often be matched with incredible amounts of CPU processing power and time. While neural networks often yield *effective* programs, they too often do so at the cost of *efficiency* (they tend to consume considerable amounts of time and money).

Arguments against Dewdney's position are that neural nets have been successfully used to solve many complex and diverse tasks, ranging from autonomously flying aircraft [2]to detecting credit card fraud[*citation needed*].

Technology writer Roger Bridgman commented on Dewdney's statements about neural nets:
Neural networks, for instance, are in the dock not only because they have been hyped to high heaven, (what hasn't?) but also because you could create a successful net without understanding how it worked: the bunch of numbers that captures its behaviour would in all probability be "an opaque, unreadable table...valueless as a scientific resource".

In spite of his emphatic declaration that science is not technology, Dewdney seems here to pillory neural nets as bad science when most of those devising them are just trying to be good engineers. An unreadable table that a useful machine could read would still be well worth having.[19]

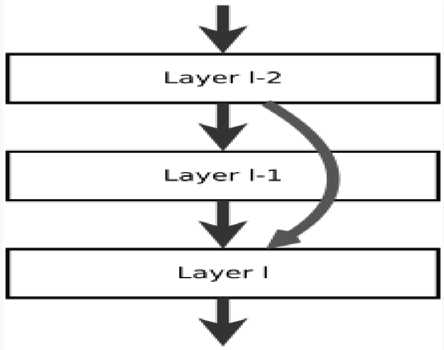
In response to this kind of criticism, one should note that although it is true that analyzing what has been learned by an artificial neural network is difficult, it is much easier to do so than to analyze what has been learned by a biological neural network. Furthermore, researchers involved in exploring learning algorithms for neural networks are gradually uncovering generic principles which allow a learning machine to be successful. For example, Bengio and LeCun (2007) wrote an article regarding local vs non-local learning, as well as shallow vs deep architecture [3].

II. SKIP CONNECTIONS INTO NETWORK ARCHITECTURES

And sometimes instead of a term short cut, you also hear the term **skip connection**, and that refers to a[l] just **skipping** over a layer or kind of **skipping** over. Almost two layers in order to process information deeper into the neural network. Residual networks and dense networks are the skip connections.

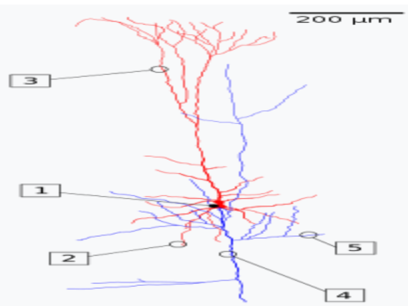
ISSN 2348 – 8034
Impact Factor- 5.070

Residual networks:



Canonical form of a residual neural network. A layer $\ell - 1$ is skipped over activation from $\ell - 2$.

A **residual neural network** is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing *skip connections* or *short-cuts* to jump over some layers. In its limit as *ResNets* it will only skip over a single layer.[1] With an additional weight matrix to learn the skip weights it is referred to as *HighwayNets*.[2]With several parallel skips it is referred to as *DenseNets*.[3] In comparison, a non-residual neural network is described as a *plain network*in the context of residual neural networks.



A reconstruction of a pyramidal cell. Soma and dendrites are labeled in red, axon arbor in blue. (1) Soma, (2) Basal dendrite, (3) Apical dendrite, (4) Axon, (5) Collateral axon.

The brain has structures similar to residual nets, as layer VI neurons gets input from layer I, skipping over all intermediary layers. In the figure this compares to signals from the (3) Apical dendrite skipping over layers while the (2) Basal dendrite collecting signals from the previous and/or same layer.[note 1][4] Similar structures exists for other layers.[5] How many layers in the cerebral cortex compare to layers in an artificial neural network is not clear, neither if every area in cerebral cortex exhibits the same structure, but over large areas they look quite similar.

The motivation for skipping over layers in ANNs is to avoid the problem of vanishing gradients, by reusing activation from a previous layer until the layer next to the current one have learned its weights. During training the weights will adapt to mute the previous layer and amplify the layer next to the current. In the simplest case only the weights for the connection to the next to the current layer is adapted, with no explicit weights for the upstream previous layer. This usually works properly when a single non-linear layer is stepped over, or in the case when the intermediate layers are all linear. If not, then an explicit weight matrix should be learned for the skipped connection.
The intuition on why this works is that the neural network collapses into fewer layers in the initial phase, which makes it easier to learn, and thus gradually expands the layers as it learns more of the feature space. During later learning, when all layers are expanded, it will stay closer to the manifold and thus learn faster. A neural network

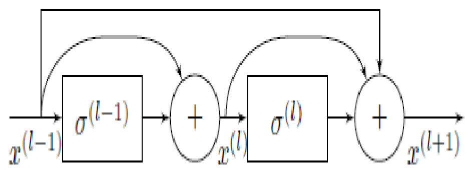
ISSN 2348 – 8034
Impact Factor- 5.070

without residual parts will explore more of the feature space, small perturbations will make it leave the manifold altogether, and thus needs it will need a lot more training data just to get it back on track.
Residual Networks as Dynamical Equations

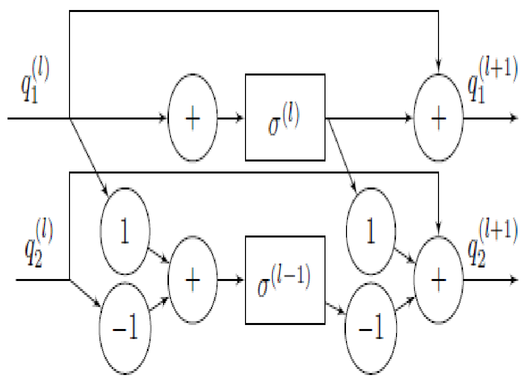
The residual network [5] has a single skip connection and therefore is simply a C1 dynamic transformation:

$$x^{(l+1)} = x^{(l)} + \sigma^{(l)}\left(x^{(l)}\right)\Delta l$$

Dense networks



(a) A dense network with $k = 2$.



(b) The equivalent state-space model of the k = 2 dense network.

A **dense network** is a **network** in which the number of links of each node is close to the maximal number of nodes. Each node is linked to almost all other nodes. The total connected case in which exactly each node is linked to each other node is called a completely connected **network**.
2.3 Dense Network for General k 2 N

The dense network [6] for general k is defined by the following system of k-many equations:

$$x^{(l+1-n)} = \sum_{l'=n}^{k-1}\left[\sigma^{(l-l')}\left(x^{(l-l')}\right)\Delta l\right] + x^{(l+1-k)} \quad \forall n = 0, 1, \cdots, k-1$$

III. CONCLUSION

This paper has developed a theory of skip connections in neural networks in the state space setting of dynamical systems with appropriate algebraic structures. We reviewed work showing that one can induce entire classes of network architectures from finite difference approximations to differential equations, and that the residual network is derived from a first order differential equation. This theory was then applied to densely connected networks, showing that dense networks to level k are in fact the interior elements of finite difference approximations to kth order differential equations. Closed form solutions for the state space representations of both Ck networks, as well as dense networks, were found. This has the immediate consequence that these kth-order network architectures are

ISSN 2348 – 8034
Impact Factor- 5.070

equivalent from a dynamical systems perspective to defining k first-order systems. This reduces the number of parameters needed to learn by a factor of k2 while retaining the same state space embedding dimension for the equivalent C0 and C1 networks.

Two carefully designed experiments were then conducted to validate and understand the proposed theory, where datasets were designed such that restricted to a certain number of nodes, the neural network is only able to properly separate the classes by using the state variables in addition to its position, such as velocity, acceleration, jerk etc. This paper explains in part why skip connections have been so successful, and further motivates the development of architectures of these types.

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, MartinWattenberg, MartinWicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Largescale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Bo Chang, Lili Meng, Eldad Haber, Frederick Tung, and David Begert. Multi-level residual networks from dynamical systems view. arXiv preprint arXiv:1710.10348, 2017.

[3] Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. Inverse Problems, 34(1):014004, 2017.

[4] Michael Hauser and Asok Ray. Principles of riemannian geometry in neural networks. In Advances in Neural Information Processing Systems, pages 2804–2813, 2017.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[6] Gao Huang, Zhuang Liu, Kilian QWeinberger, and Laurens van der Maaten. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, volume 1, page 3, 2017.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[8] Helmut Proctinger et al. Some information about the binomial transform. 1993.

[9] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[10]Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In Advances in Neural Information Processing Systems, pages 550–558, 2016