Transferability of Learned Knowledge in Neural Networks: Impact of Trained Weights on Untrained Networks

[1]Adwaita Tulsyan

[2]Udit Chaturvedi

[3]Neha V Sharma

Abstract: - This research explores the intricate dynamics of neural networks (NNs) with a focus on understanding the profound implications of the training process on their performance. In a departure from conventional transfer learning, this study delves into the manual initialization of untrained neural networks with weights and biases extracted from trained networks of identical architecture. Despite their plain congruence, we meticulously investigate the nuanced distinctions between trained and untrained networks and analyze the disparities in their performance. This study introduces mathematical foundations, empirical findings, and implications that underscore the significance of the training process in the realm of neural networks. Applying the ideas of weight extraction, this research is inspired by the unique process of visual learning and "mirroring" seen in humans; the ability to mimic something simply by seeing the methods involved.

Keywords: Neural Networks, Transfer Learning, Untrained Networks, Manual Initialization, Performance Comparison, Weight Extraction, Weight Updating

I. Introduction

It is well known that humans are excellent and quick learners. Their ability to understand, comprehend and adapt to their environment is exceptional to humans alone. Often, they can simply observe an activity and hold the ability to perfectly imitate it. Simply by seeing the movements and the procedure involved, they can produce similar results without prior training. This ability to produce results without training, simply for manual understanding of each component is a very interesting phenomenon and area of research. Since neural networks are based on the human brain, it is exciting to find out whether the same phenomenon can be observed in neural networks, whether they can produce results without any prior training.

Neural networks have appeared as a fundamental component of contemporary artificial intelligence, demonstrating remarkable prowess across a spectrum of applications. The essence of training a neural network revolves around the refinement of its internal parameters, notably the weights and biases, to optimize its performance on specific tasks. This paper endeavors to delve into the intricacies of trained and untrained neural networks, with the aim of understanding the transformative influence of the training process.

Neural networks are a powerful tool in the field of machine learning, inspired by the structure and function of the human brain. These networks consist of interconnected nodes, called artificial neurons, that loosely mimic the biological neurons and their connections. Each neuron processes information and transmits it to others, allowing the network to learn and improve its performance over time through exposure to data. By adjusting the connections and weights between these artificial neurons, neural networks can perform complex tasks like image recognition, speech translation, and even generate creative text formats. Their ability to learn and adapt makes them increasingly valuable for various applications across diverse fields.

Neural networks rely on a series of mathematical operations to process information and learn. During forward propagation, the network calculates a weighted sum **(z)** of its inputs **(x)** and a bias term **(b)** using the equation: **(1)** $\boldsymbol{z = sum_i(w) + b}$, where **i** iterates over all input features, $\boldsymbol{w_i}$ represents the weight of the $\boldsymbol{i'th}$ connection, and $\boldsymbol{x_i}$ represents the value of the $\boldsymbol{i'th}$ input. This weighted sum then undergoes a non-linear transformation **(f)** using an activation function to introduce non-linearity into the network, resulting in the neuron's activation **(a)** expressed as: $\boldsymbol{(2)\ a = f(z)}$

[1] Student, Manipal University Jaipur, Jaipur (Rajasthan), India

[2] Student, Manipal University Jaipur, Jaipur (Rajasthan), India

[3] Professor, Manipal University Jaipur, Jaipur (Rajasthan), India

The cost function denoted as $(\boldsymbol{J}(\boldsymbol{W},\boldsymbol{b}))$, encapsulates the network's performance evaluation. For example, in a binary classification problem with logistic regression, the cost function takes the form:

$$(\mathbf{3})\ \boldsymbol{J}(\boldsymbol{W},\boldsymbol{b}) = -\frac{\mathbf{1}}{\boldsymbol{m}}\sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{m}}\left(\boldsymbol{Y}_{\boldsymbol{i}}\mathbf{lo\,g}\left((\widehat{\boldsymbol{Y}_{\boldsymbol{\iota}}})\right) + (\mathbf{1}-\boldsymbol{Y}_{\boldsymbol{i}})\mathbf{lo\,g}\left(\mathbf{1}-(\widehat{\boldsymbol{Y}_{\boldsymbol{\iota}}})\right)\right)$$

Where $(\boldsymbol{J}(\boldsymbol{W},\boldsymbol{b}))$ stands for the cost to be minimized, $(\ \boldsymbol{m}\ )$ signifies the number of training examples, $(\ \boldsymbol{Yi}\ )$ is the actual output, and $(\widehat{(\boldsymbol{Y}_{\boldsymbol{\iota}})})$ denotes the predicted output. This equation encapsulates the core of the training process, where the neural network aims to minimize $\left(\boldsymbol{J}(\boldsymbol{W},\boldsymbol{b})\right)$ by iteratively adjusting the weights and biases.

II. Literature Review

The significance of training neural networks cannot be overstated, as it forms the bedrock of their functionality and performance. The training process entails iteratively adjusting the parameters of the network, such as weights and biases, to minimize a predefined cost function. Through this iterative optimization, neural networks acquire the ability to extract meaningful patterns from input data, enabling them to make accurate predictions or classifications.

One prominent approach in neural network training is transfer learning, where pre-trained models are adapted to new tasks. This method leverages knowledge gained from prior training on similar tasks, facilitating faster convergence and often better performance on the target task. However, this study takes a distinct focus by delving into the manual initialization of untrained neural networks, sidestepping pre-trained models altogether. Instead of inheriting knowledge from pre-existing models, the networks are initialized with random weights and biases. Subsequently, they learn solely from the provided training data without any prior information or guidance from pre-trained models, a methodology commonly referred to as "training from scratch."

The research methodology employed in this study centers on the challenges associated with weight initialization in neural networks. Weight initialization plays a crucial role in determining the initial state of the network and can significantly impact its learning dynamics and final performance. By meticulously examining various strategies for initializing network weights, the study aims to elucidate the importance of this initial step and its implications for subsequent training processes.

By focusing on training neural networks from scratch, the research sheds light on the intricate dynamics involved in learning from raw data without external guidance. This approach offers insights into the fundamental mechanisms underlying neural network training and allows for a deeper understanding of how network architectures interact with training data to achieve desired outcomes.

Moreover, the absence of pre-existing knowledge or reliance on pre-trained models underscores the self-sufficiency and adaptability of neural networks initialized from scratch. By embracing the inherent randomness in weight initialization and embracing the challenge of learning without prior guidance, the study aims to explore the full potential of neural networks in capturing and leveraging complex patterns inherent in diverse datasets.

In summary, this literature review provides a comprehensive overview of the research methodology adopted in the study, emphasizing the focus on training neural networks from scratch and the importance of weight initialization in this process. It sets the stage for further exploration into the intricacies of neural network training and the potential benefits of bypassing pre-trained models in favor of a more independent learning approach.

III. Methodology

A. Intermodal Comparision

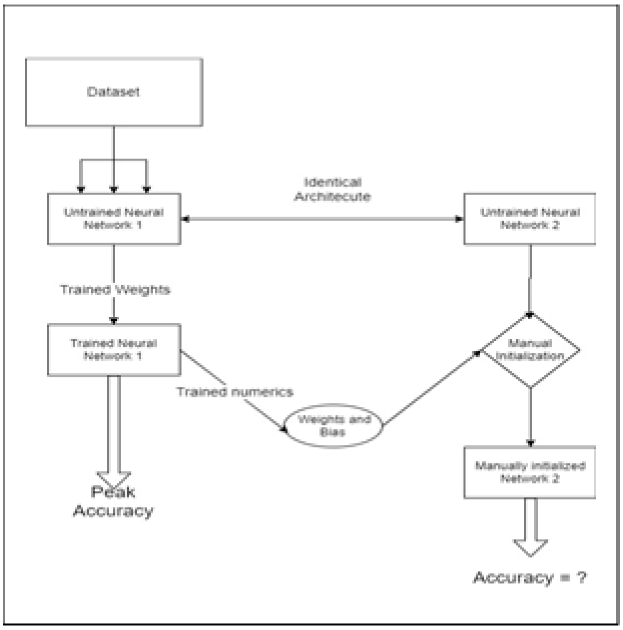

Figure 1. Architecture of intermodal execution

This research encompasses an in-depth exploration of two prominent types of neural networks: Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs). Each type possesses distinct strengths and limitations that significantly impact their accuracy and effectiveness in various tasks. The process of extracting weights and biases from ANNs is relatively straightforward, facilitated by their matrix-based representations. However, the complexity inherent in the convolutional architecture of CNNs presents significant challenges in weight extraction and initialization. Moreover, CNNs often require a substantial corpus of training data, making direct initialization a daunting task.

B. Considering CNNs

Convolutional Neural Networks (CNNs) represent a powerful subfield within deep learning, specifically designed to excel at tasks involving grid-like data, primarily images. Drawing inspiration from the hierarchical structure of the visual cortex in the mammalian brain, CNNs leverage a unique architecture to achieve exceptional performance in computer vision applications. Unlike traditional neural networks, which treat each data point independently, CNNs exploit the inherent spatial relationships present in grid-like data.

The core building block of a CNN is the convolutional layer. This layer employs a set of learnable filters, also known as kernels, that slide across the input data, identifying and extracting specific patterns or features. Imagine scanning an image with a small window that focuses on specific details like edges or corners. As the filter moves across the image, it calculates the dot product between its elements and the corresponding elements in the input data, producing a feature map.

Following the convolutional layer, a pooling layer often comes into play. This layer aims to reduce the dimensionality of the data while preserving the most critical features extracted by the convolution. Pooling commonly involves operations like averaging or taking the maximum value within a specific region of the feature map. By progressively applying convolutional and pooling layers, CNNs are able to build a hierarchical representation of the input data, starting with low-level features like edges and lines, and gradually progressing to more complex features that combine to form the overall structure of the data.

This hierarchical feature extraction capability empowers CNNs to excel in various computer vision tasks. They are widely employed in image classification, where the goal is to categorize an image into a predefined set of classes (e.g., identifying cats, dogs, or airplanes in an image). Additionally, CNNs play a crucial role in object detection, where the network not only identifies the presence of objects but also pinpoints their location within the image. Furthermore, CNNs demonstrate remarkable capabilities in tasks like image segmentation, where each pixel in the image is assigned, a label corresponding to its specific content (e.g., segmenting an image to distinguish between the sky, foreground objects, and background).

In conclusion, CNNs represent a transformative technology in the realm of computer vision, offering exceptional performance due to their ability to leverage the inherent spatial relationships within grid-like data. Their success stems from the unique architecture that leverages convolutional and pooling layers, enabling them to extract increasingly complex features and achieve groundbreaking results in various image-related tasks.

CNNs offer a straightforward path to achieving high accuracy, owing to their capability to efficiently filter and convolute large volumes of data. However, the intricate architecture of CNNs poses challenges in extracting and initializing their weights. Furthermore, the demanding nature of CNNs in terms of training data exacerbates the difficulty of initializing them without prior training.

C. Embracing ANNs

While Convolutional Neural Networks (CNNs) boast impressive capabilities in image recognition and computer vision, their inherent complexity can pose challenges for our chosen methodology. The intricate architecture of CNNs, characterized by multiple convolutional and pooling layers, often necessitates vast amounts of training data to achieve optimal performance. This data requirement, coupled with the complex computations involved during training, translates to significant computational resources and extended training times. In our project, where resource limitations and time constraints might be factors, opting for a simpler Artificial Neural Network (ANN) architecture could prove more practical. ANNs typically require less data and computational power for training compared to CNNs.

This can be particularly advantageous when dealing with smaller datasets or limited computational resources. While ANNs might not achieve the same level of accuracy as CNNs in image recognition tasks, their simpler structure allows for faster training and potentially sufficient performance for our specific goals. By carefully considering the trade-off between accuracy and resource requirements, we can select the most appropriate neural network architecture for our methodology.

By concentrating on ANNs, the research aims to delve into the nuances of weight initialization without the added complexities introduced by CNNs. This focused approach enables a more detailed examination of the interplay between initialization strategies and network performance, ultimately contributing to a deeper understanding of neural network training dynamics. Additionally, by acknowledging the challenges inherent in initializing CNNs, the research sets the stage for future investigations into overcoming these obstacles and harnessing the full potential of convolutional architectures.

IV. Method II

A. *Intramodal Comparision*

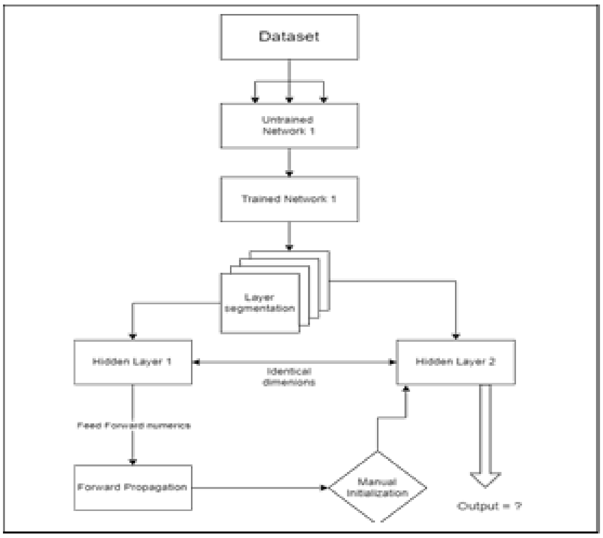

Figure 2. Intramodal execution

Transfer learning, at its core, is a method of transferring trained model prevalence into another model for similar purposes. By design, it should be possible to scale the level of transferring, whether it be up or down.

It should be possible to scale the level of transferring down to just the layers within the same network; it follows identical procedure the intermodal experiment, i.e. same architecture, structure, dimensions, etc.

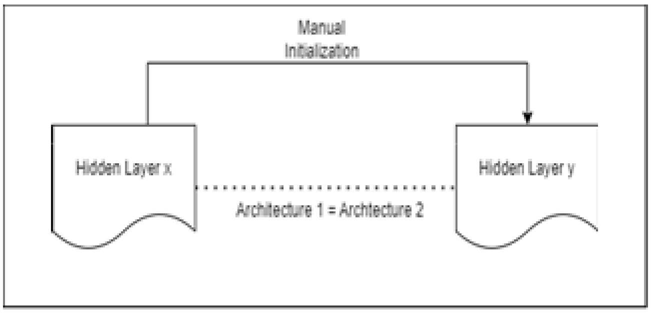

Figure 3. Showing the basis of the intramodal method

It may be noted that it is not entirely necessary for the layers taken into account to be adjacent or neighboring, either can be considered as long as the architectural congruence is accounted for.

B. *Considering general backpropagation with marked historical logs for comparison*

Normal backpropagation algorithm is also considered as a case study, as it can be considered a rudimentary form of "transferring" weights to another layer. In theory it should also be possible to scale this up to perform within different networks. Although backpropagation does not satisfy the base for manual initialization, the automatic recursion involved can be looked at in the form of repeated "updates", performed at every adjacent layer while disregarding the need for an identical architecture.

V. Results and discussions

Programming logic is not directly supported to test a network directly while skipping the training process. Libraries used for the research presented challenges in their work. It is not feasible to test a network that has no history of propagating values through itself.

Surface Similarity: At first glance, trained and untrained neural networks with identical initial weights and biases may seem identical. They share structural elements like layers, neurons, activation functions, and numeric values.

Theoretical Expectation: In theory, they should perform equally well, with identical accuracy. However, real-world testing reveals a significant difference in precision and accuracy.

Trained Network Superiority: Trained models consistently outperform untrained ones. The key to this difference lies in their adaptability. Trained networks can adjust their internal parameters based on past data, learning from previous experiences.

A major limitation is observed; the weights are an immutable attribute and hence cannot physically be altered as the research requires unless represented with another method replicating the same. Layer segmentation is not recognized as all layers are atomic to the network they belong to.

Lack of possible implications in the methods used provide for only theoretical results-

•Untrained Network Limitation: In contrast, an untrained network starts as an underfit model with no historical data, which hampers its predictive capabilities.

•Challenge of Backpropagation: Untrained networks face challenges like the inability to use methods like backpropagation due to the absence of a history of stored weights. This limits their ability to update weights effectively and leads to problems such as vanishing gradient.

It is open to discussion and discovery if the same result is achieved on different platforms and using different methodologies differently from the ones used in this research.

ACKNOWLEDGMENT

We would like to thank our co-author, Mr. Udit Chaturvedi, for their valuable contributions to the analysis and model preparation. We are also grateful to our advisor, Professor Neha Sharma, for their guidance and support throughout the research process. We would like to thank Dr Lokesh Sharma, who helped provide us with invaluable feedback and resources. Finally, we thank the reviewers for their insightful comments and suggestions that helped improve the quality of this manuscript.

REFERENCES

[1] Transfer Learning for Natural Language Processing (2018 book by Pang and Lee)

[2] Farfan, G., & Jimenez, L. M. (2020). Fossil Brachiopod identification using a new deep convolutional neural network. ("[PDF] Fossil brachiopod identification using a new deep convolutional ...") Research Gate

[3] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[4] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.

[5] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep Learning (Vol. 1). MIT press Cambridge.

[6] Zhang, S., & Zhang, C. (2020). A survey of deep learning for big data. Information Fusion, 57, 47-70.

[7] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25. ("ImageNet Classification with Deep Convolutional Neural Networks - NIPS")

[8] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition.

[9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition.

[10] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the start architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition.

[11] Artificial Neural Network - an overview | ScienceDirect Topics.

[12] https://arxiv.org/abs/2201.09679

[13] https://proceedings.mlr.press/v139/radford21a.html

[14] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. &amp; Sutskever, I.. (2021). Learning Transferable Visual Models From Natural Language Supervision

[15] Yuefeng Hao, Jongjin Baik, Hien Tran, Minha Choi, Quantification of the effect of hydrological drivers on actual evapotranspiration using the Bayesian model averaging approach for various landscapes over Northeast Asia, Journal of Hydrology, Volume 607, 2022, 127543, ISSN 0022-1694

[16] Initializing Deep Nets for Supervised Learning: How to Bake Your Knowledge into the Init (ICLR 2015 workshop)

[17] A Theoretical Framework for Transfer Learning with Deep Autoencoders (JMLR, 2015)