*American Journal of Artificial Intelligence*
2025, Vol. 9, No. 2, pp. 129-132
https://doi.org/10.11648/j.ajai.20250902.14

SciencePG
Science Publishing Group

Research/Technical Note

# An Alternative Way of Determining Biases and Weights for the Training of Neural Networks

**Huseyin Murat Cekirge***

Mechanical Engineering, the City College of New York, the City University New York, New York, USA

## Abstract

The determination of biases and weights in neural networks is a fundamental aspect of their performance, traditionally employing methods like steepest descent and stochastic gradient descent. While these supervised training approaches have proven effective, this technical note confidently presents a groundbreaking alternative that eliminates randomness in calculations altogether. This innovative method calculates biases and weights as precise solutions to a system of equations. This approach not only reduces computational demands but also enhances energy efficiency significantly. By strategically incorporating target values during training, we can expand the number of target values within acceptable variance limits, enabling the formation of square matrices. This ensures that input and output nodes remain perfectly balanced through the generation of fictive data, particularly for output nodes. These fuzzy sets guarantee that the variance of neuron target values stays within permissible limits, effectively minimizing error. The generated data is intentionally minimal and can also be produced using random processes, facilitating effective learning for the neural networks. Unlike conventional techniques, the values of biases and weights are determined directly, leading to a process that is both faster and less energy-intensive. Our primary objective is to establish an efficient foundation for the training data of the neural network. Moreover, these calculated values serve as robust initial parameters for integration with other determination methods, including stochastic gradient descent and steepest descent. This presentation showcases a powerful new algorithm, poised to significantly enhance the efficiency and effectiveness of neural network training.

## Keywords

Neural Networks, Stochastic Random Steepest Descent, Steepest Descent, Data Training in Neural Networks, Initialization of Data Training

## 1. Introduction

An artificial neural network (ANN) considers biological principles with mathematics to solve problems of artificial intelligence. Zell [1], Dawson [2], Yang and Yang [3], and Shao and Shen [4]. The calculation of biases and weights in neural networks is being calculated or trained by using stochastic random descent method or similar optimization methods at the layers of the neural networks.

To ensure effective training, it is crucial that the network is exposed to both accurate and inaccurate values upfront, a process known as supervised learning. This approach empowers the network to learn and produce precise outputs based on the correct answers it is trained with. Initially, ran-

*Corresponding author: hmcekirge@usa.net (Huseyin Murat Cekirge)

**Received:** 22 July 2025; **Accepted:** 4 August 2025; **Published:** 18 August 2025

dom values are assigned to the network's parameters—specifically the biases and weights—that influence how input signals are processed and interpreted. The methodology focuses on gathering data to derive insightful conclusions using an ANN. The architecture of an ANN includes an input layer, one or more hidden layers, and an output layer, with nodes within these layers interconnected by thresholds and weights. Each node functions with its own unique linear regression model, contributing to the overall effectiveness of the network in solving problems efficiently,

$$y_i = b_i + \sum_j w_{ij}\, x_j \ , \tag{1}$$

where

$y_i, x_j$, $b_i$ and $w_{ij}$ are outputs, inputs, biases and weights, respectively.

i and j are the indices of outputs and inputs.

i = 1 to $n_1$, j=1 to $n_2$; $n_1$= number of outputs and $n_2$= number of inputs.

The determination of biases and weights is the essential problem in the training of the ANN. It is best to determine these values. However, there are many unknowns, such as biases and fewer equations. For that reason, several optimization techniques are being used, Levitan and Kaczmare [5], Hestenes and Stiefel [6], Straeter [7], Speiser and Ambros [8], Rosenblatt [9], Bishop [10], Needell, Srebro and Ward [11], and Bottou [12]. To ensure effective training, it is crucial that the network is exposed to both accurate and inaccurate values upfront, a process known as supervised learning. This approach empowers the network to learn and produce precise outputs based on the correct answers it is trained with. Initially, random values are assigned to the network's parameters—specifically the biases and weights—that influence how input signals are processed and interpreted. The methodology focuses on gathering data to derive insightful conclusions using an ANN. The architecture of an ANN includes an input layer, one or more hidden layers, and an output layer, with nodes within these layers interconnected by thresholds and weights. Each node functions with its own unique linear regression model, contributing to the overall effectiveness of the network in solving problems efficiently.

## 2. Analysis

Equation (1) can be written as,

$$\sum_j x_j \ w_{ij} = y_i \ , \tag{2}$$

where the bias is considered as $w_{i1} = b_i$ and $x_1 = 1$, it is treated as a weight. For i+1 inputs, including the bias, there are j+1 weights which are unknowns. For the target output $y_i$, there are a single equation, however there exist $i + 1$ unknowns, denoted as $w_{ij}$. For a resolution for this situation, another i number of equations are required. These auxiliary equations $y_i$ could be created by considering behavior of neural networks. The another n auxiliary $y_j^k$ outputs can be created, by considering a variance of σ which is minimal percentages target value and a training parameter,

Where the bias is considered as $w_{i1} = b_i$ and $x_1 = 1$, it is treated as a weight. For (i + 1) inputs, including the bias, there are (j + 1) weights, which are unknowns. For the target output $y_i$, there is a single equation; however, there are multiple $(i + 1)$, $w_{ij}$ unknowns. To resolve this situation, an additional (i) number of equations is required. These auxiliary equations can be created by considering the behavior of neural networks. Another (n) auxiliary $y_j^k$ outputs can be generated by taking into account a variance (σ), which represents a minimal percentage of the target value, along with a training parameter.

$$y_i^k = y_i^k - (k-1)\ \delta\sigma \ \ or \tag{3}$$

$$y_i^k = y_i^k + (k-1)\ \delta\sigma \ \ and \tag{4}$$

$$\delta\sigma = \frac{\sigma}{k-1} \ \ and \ \ k = i + 1. \tag{5}$$

There are k equations, where

$$k = 1,2, \ldots\ldots, i + 1 \tag{6}$$

The values of $y_i^k, s$ can be effectively determined through a random assignment process. It is crucial to understand that the coefficients $x_j^k$ must not be identical across all equations; such uniformity would lead to a singular matrix, which is unacceptable. Therefore, except for the case k=1, one or more coefficients per row in each equation may be altered. Such modifications must preserve the distinctness of the coefficients in each row relative to all other rows and may lead to deviations in δσ or its fractional components. These coefficients may also be switched between the values 0 and 1. The suggested equations should not deviate significantly from the first or base equation. The process involves trial and error and relies on a heuristic approach. The error function serves as a measure for evaluating the quality and appropriateness of the selections. It is still less time-consuming compared to optimization methods, whether or not they involve randomization The computations can be terminated once the cost function yields satisfactory results. It is imperative that the matrix $x_j^k = X_j^k$ is not singular, as this is essential for maintaining its integrity of the computations. If $Y_j^k$ is the inverse of matrix $X_j^k$, then;

$$w_{ij} = Z_j^k \ \ y_i^k. \tag{7}$$

This methodology functions as a hybrid numerical method that enhances the speed and robustness of compu-

tations across various AI challenges. It is essential to highlight that the bias is represented as $w_{i1} = b_i$ for all k values. These parameters should be carefully evaluated to determine their appropriateness for training the Artificial Neural Network (ANN), taking into account a range of different values. Moreover, the selected $\sigma$ value acts as an essential training parameter, chosen as a small fraction of the target values. By identifying an appropriate value, an effective process will be in place, this approach can evolve into a well-structured training method for the artificial neural network (ANN), ultimately producing acceptable and cost-effective results.

## 3. Conclusion

This methodology is designed to seamlessly integrate with the innovative techniques introduced by Li et al. [13], Wang et al. [14], and Apedo and Tao [15] for training data in artificial intelligence applications. By leveraging this approach, Xavier, He, and other initialization methods can accelerate training processes, as highlighted in the research by Xavier and Bengio [16] and He et al. [17]. The adoption of this method will substantially enhance both the efficiency and effectiveness of AI training initiatives.

The methodology introduced involves the creation of a fictitious matrix for accurately determining biases and weights. This technique can work in conjunction with other established methods, providing a significant advantage. It not only reduces computational time compared to traditional iterative methods but also minimizes energy consumption. Additionally, this approach can serve as a reliable initial guess for these values during the training of artificial neural networks (ANNs). The objective is to emulate a system of equations, similar to those in Hardesty [18], Bishop [19], Vapnik [20], and Goodfellow *et al.* [21]. To obtain a unique solution, an additional i independent equations are needed. This approach ensures practical flexibility, supporting accurate and iterative refinement of AI training models. This is accomplished by maintaining a minimal error in the neural input nodes, effectively simulating the realistic behavior of ANNs. As a result, this methodology proves to be both practical and valuable in enhancing neural network performance. In other words, this process definitively mimics the realistic behavior of an artificial neural network (ANN).

The methodology focuses on squaring a matrix while ensuring it remains nonsingular, taking into account the target output values of the training data. By eliminating the need for optimization, randomization, and extensive iterations, computational efficiency is significantly improved. Randomly selected values simplify the squaring process without distorting the target output values, ensuring that the matrix continues to be nonsingular. Ultimately, this approach addresses the solution of a system of linear equations within the fundamental structure of the computations. To achieve maximum flexibility and efficiency, it is imperative to implement a robust set of effective procedures in the squaring process. This strategic approach will effectively eliminate the risks of nonconvergence and noncompliance with training data by implementing a strong deterministic framework. The strategies for squaring the matrix require commitment to maintaining the integrity of the training process, ensuring there is no deviation from our target input values and possibly close initial values. Of course, the error function must be used synchronously for preventing the off target deviations. The approach is essential for ensuring consistency and reliability in outcomes, and it may be conceptualized as a form of 'dictated learning. Furthermore, this methodology could potentially be applied within the framework of backpropagation algorithms and integrated into other artificial intelligence procedures, including perturbation techniques, Cekirge [22]. Perturbation methods typically begin with a zeroth-order solution, which can be determined by this method and serves as the baseline for subsequent refinements. Finally, this approach is anticipated to reduce computational complexity, processing time, and energy consumption across all training procedures for artificial intelligence problems.

In artificial intelligence computations, streamlining processes is crucial. Removing lengthy iterations, randomizations, and optimizations enhances efficiency. This leads to improved performance and significantly faster results, making artificial intelligence solutions more impactful and effective.

## Abbreviations

| | |
|---|---|
| ANN | Artificial Neural Network |
| AI | Artificial Intelligence |

## Author Contributions

Huseyin Murat Cekirge is the sole author. The author read and approved the final manuscript.

## Conflicts of Interest

The author declares no conflicts of interest.

## References

[1] Zell, Andreas (2003). Simulation neuronaler Netze [Simulation of Neural Networks] (in German) (1st ed.), Addison-Wesley. https://www.degruyter.com/document/isbn/9783486243505/html

[2] Dawson, Christian W. (1998). "An artificial neural network approach to rainfall-runoff modelling". Hydrological Sciences Journal. 43 (1): 47-66. Bibcode: 1998HydSJ. 43. 47D. https://doi.org/10.1080/02626669809492102

[3] Yang Z. and Yang Z. (2014). Comprehensive Biomedical Physics. Karolinska Institute, Stockholm, Sweden: Elsevier. p. 1. Archived from the original on July 28, 2022, retrieved on July 28, 2022. https://shop.elsevier.com/books/comprehensive-biomedical-physics/brahme/978-0-444-53632-7

[4] Shao, Feng and Shen, Zheng (January 9, 2022)."How can artificial neural networks approximate the brain?". Front. Psychol. 13: 970214. https://doi.org/10.3389/fpsyg.2022.970214

[5] Levitan, Irwin and Kaczmarek, Leonard (August 19, 2015). "Intercellular communication". The Neuron: Cell and Molecular Biology (4th ed.). New York, NY: Oxford University Press. pp. 153-328. https://global.oup.com/academic/product/the-neuron-9780199773893

[6] Hestenes, Magnus R. and Stiefel, Eduard (December, 1952). "Methods of Conjugate Gradients for Solving Linear Systems" (PDF). Journal of Research of the National Bureau of Standards. 49 (6): 409. https://doi.org/10.6028/jres.049.044

[7] Straeter, T. A. (1971). On the Extension of the Davidon-Broyden Class of Rank One, Quasi-Newton Minimization Methods to an Infinite Dimensional Hilbert Space with Applications to Optimal Control Problems (PhD thesis). North Carolina State University. hdl: 2060/19710026200 - via NASA Technical Reports Server.

[8] Speiser, Ambros (2004). "Konrad Zuse und die ERMETH: Ein weltweiter Architektur-Vergleich" [Konrad Zuse and the ERMETH: A worldwide comparison of architectures]. In Hellige, Hans Dieter (ed.). Geschichten der Informatik. Visionen, Paradigmen, Leitmotive (in German), Berlin: Springer. p. 185. https://doi.org/10.1007/978-3-642-18631-8

[9] Rosenblatt, F. (1958). "The Perceptron: A Probabilistic Model For Information Storage and Organization In The Brain". Psychological Review. 65 (6): 386-408. Cite Seer X 10.1.1.588.3775. https://doi.org/10.1037/h0042519

[10] Bishop, C. M. (2006). Pattern Recognition and Machine Learning (Information Science and Statistics), Springer. https://www.springer.com/gp/book/9780387310732

[11] Needell, D., Srebro, N. and Ward, R. (January, 2015). Stochastic gradient descent weighted sampling, and the randomized Kaczmarz algorithm. https://arxiv.org/pdf/1310.5715.pdf

[12] Bottou, L. (1991) Stochastic gradient learning in neural networks. Proceedings of Neuro-Nimes, 91. https://leon.bottou.org/publications/pdf/nimes-1991

[13] Li, P., Tao, H., and Zhou, H. et al. (2025). Enhanced Multiview attention network with random interpolation resize for few-shot surface defect detection. Multimedia Systems 31, 36. https://doi.org/10.1007/s00530-024-01643-y

[14] Wang, Z., Tao, H. and Zhou, H. et al. (2025). A content-style control network with style contrastive learning for underwater image enhancement. Multimedia Systems 31, 60. https://doi.org/10.1007/s00530-024-01642-z

[15] Apedo, Y., Tao, H. (2025) A weakly supervised pavement crack segmentation based on adversarial learning and transformers. Multimedia Systems 31, 266. https://doi.org/10.1007/s00530-025-01850-1

[16] Glorot, Xavier and Bengio, Y. (January, 2010) Understanding the difficulty of training deep feedforward neural networks, Journal of Machine Learning Research 9: 249-256. https://doi.org/10.5555/1756006.1953039

[17] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing and Sun, Jian (2015). "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification". arXiv: 1502.01852.

[18] Hardesty L. (April 14, 2017). "Explained: Neural networks". MIT News Office. Archived from the original on March 18, 2024, retrieved on June 2, 2022. https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414

[19] Bishop C. M. (August 17, 2006). Pattern Recognition and Machine Learning. New York: Springer. https://www.springer.com/gp/book/9780387310732

[20] Vapnik V. N. (1998). The nature of statistical learning theory (Corrected 2nd print. ed.). New York Berlin Heidelberg: Springer. https://www.springer.com/gp/book/9780387987804

[21] Goodfellow, Ian; Bengio, Yoshua and Courville, Aaron (2016). Deep Learning. MIT Press. Archived from the original on 16 April 2016, retrieved on June 1, 2016. https://mitpress.mit.edu/9780262035613/deep-learning/

[22] Cekirge, H. M. (2025), Tuning the Training of Neural Networks by Using the Perturbation Technique, American Journal of Artificial Intelligence, Vol. 9, No. 2, pp. 107-109. https://doi.org/10.11648/j.ajai.20250902.11