

Estimating Parameters of Gamma Distribution Using Neural Network

RASHA MOHAMED MANDOUH, MAHMOUD RIAD MAHMOUD, EHAB RASHEEDY ALI

Department of Mathematical Statistics
Faculty of Graduated Studies For Statistical Research
Cairo University, Cairo
EGYPT

Abstract: Widely used in scientific and engineering domains, the Gamma distribution provides an essential framework for describing waiting-time mechanisms and positively skewed patterns. Achieving accurate parameter estimates is therefore critical for maintaining the reliability of statistical inference and improving predictive performance. In this paper, the classical methods for estimating the gamma distribution are assessed in comparison to a new method of estimation using neural networks. The integration of Artificial Neural Networks (ANN) into the estimation process enabled the development of a streamlined yet robust neural architecture capable of simultaneously inferring the key parameters of the Gamma distribution. This design leverages the ANN's ability to learn complex nonlinear relationships within the data, allowing the model to capture the underlying distributional structure and perform joint parameter estimation more efficiently than traditional analytical methods. The ANN neural network uses the concept of matching theoretical and sample quantities to estimate gamma parameters using mean, standard deviation, median, skewness and kurtosis. An extensive simulation study was conducted to demonstrate the power of the proposed ANN-based method. Then, the results of the proposed method were compared to MLE and moment-based methods.

Key-Words: Gamma distribution parameters, Neural network, Maximum likelihood estimation, Method of moment, parameter estimation, Backpropagation.

Received: April 9, 2026. Revised: June 13, 2026. Accepted: July 16, 2026. Published: October 14, 2026.

1 Introduction

Laplace in 1836 worked on the distribution of a “precision constant” that resulted in the gamma distribution, which was used to model waiting times, e.g. the waiting time until death [5]. In applications, the gamma distribution is usually used due to the variety of shapes the distribution assumes for different values of parameters [4]. Moreover, it is used in Bayesian statistics; It is often used as a conjugate prior distribution of scale parameters. For example, the scale parameter in an exponential distribution or a normal distribution with a known mean. Furthermore, it is applicable in many fields like the size of insurance claims [2]. A random variable X is said to have a gamma distribution if its probability density function is of the form

$$f(x) = \frac{(x - \mu)^{\alpha-1} e^{-\frac{(x-\mu)}{\beta}}}{\beta^{\alpha} \Gamma(\alpha)}, \quad x > \mu, \alpha > 0, \beta > 0 \quad (1)$$

The distribution in (1) — denoted $\text{gamma}(\alpha, \beta, \mu)$ — is type III of Pearson's system of distribution [5]. It depends on three parameters (α, β, μ) . If $\mu = 0$, the distribution is termed a two-parameter gamma distribution, denoted $\Gamma(\alpha, \beta)$ with pdf.

$$f(x, \alpha, \beta) = \frac{1}{\alpha \Gamma(\beta)} \left(\frac{x}{\alpha}\right)^{\beta-1} e^{-\frac{x}{\alpha}}, \quad x > 0, \beta, \alpha > 0 \quad (2)$$

The gamma distribution appears naturally as the distribution of the sum of squares of independent unit normal variables in the theory associated with

normally distributed random variables [5]. Figure 1 displays gamma pdf for different parameter values.

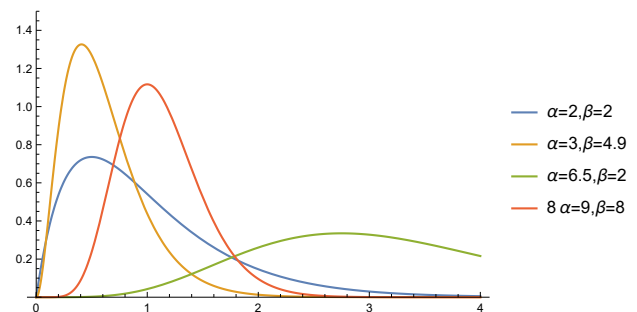


Fig. 1. PDF for Gamma Distribution at different parameter values

2 Estimating Gamma Parameter Using Method of Moments

The method of moments consists of equating sample moments with unobservable population quantities for the quantities to be estimated. Suppose $X_1, X_2, \dots, X_n$ are independent identically distributed random variables with a gamma PDF. Considering the moment generating function for the gamma distribution we get the following.

$$Mx(t) = \int_0^{\infty} e^{tx} \frac{1}{\alpha^2 \Gamma(\beta)} x^{\beta-1} \exp\left(-\frac{x}{\alpha}\right) dx \quad (3)$$

$$= \alpha^{-\beta} \left(\frac{1}{\alpha} - t\right)^{-\beta} \quad (4)$$

Computing the population moments for the gamma we have the first population moment is

$$U_1(\alpha, \beta) = E(X) = \mu = \alpha\beta. \quad (5)$$

And the second population moment is

$$U_2(\alpha, \beta) = E(X^2) = \alpha^2\beta(\beta + 1). \quad (6)$$

Let the first and second sample moments be m_1 and m_2 respectively. Equating the population moments with the sample moments we get

$$m_1 = \alpha\beta \quad (7)$$

$$m_2 = \alpha^2\beta(\beta + 1) \quad (8)$$

where

$$m_1 = \frac{1}{n} \sum_{i=1}^n x_i \quad (9)$$

$$m_2 = \frac{1}{n} \sum_{i=1}^n x_i^2 \quad (10)$$

Solving (5) and (6) for α and β we get the following as the method of moment estimators of α and β

$$\tilde{\beta} = \frac{m_1^2}{m_2 - m_1^2} \quad (11)$$

and

$$\tilde{\alpha} = \frac{m_2 - m_1^2}{m_1}. \quad (12)$$

For more details see[6].

3 Maximum Likelihood Estimation of Gamma Distribution

The two parameter gamma distribution has a density function given by equation (4.1). Let $x_1, x_2, \dots, x_n$ ($n > 1$) represent a random sample of values of X . The log likelihood function is given by

$$l = n - \beta \log \alpha - \log \Gamma(\beta) + (\lambda - 1) \sum_{i=1}^n \log x_i - \frac{1}{\alpha} \sum_{i=1}^n x_i. \quad (13)$$

Taking partial derivatives with respect to α and β we get

$$\frac{\partial l}{\partial \beta} = n(-\log \alpha - \psi(\beta)) + \sum_{i=1}^n \log x_i \quad (14)$$

$$\frac{\partial l}{\partial \alpha} = n \frac{-\beta}{\alpha} + \frac{1}{\alpha^2} \sum_{i=1}^n x_i, \quad (15)$$

where $\psi(\beta) = \frac{\partial}{\partial \beta} [\log \Gamma(\beta)]$ is the digamma function.

The maximum likelihood estimates (MLEs) of the parameters of the gamma distribution are the values

of the parameter that produce the maximum of the log likelihood function.

Solving equations will give the MLE as

$$\hat{\alpha} = \frac{\bar{x}}{\hat{\beta}} \quad (16)$$

$$\overline{\log x} - \log \bar{x} = \log \hat{\beta} - \psi(\hat{\beta}) \quad (17)$$

Analytically the explicit expression for the MLEs cannot be obtained and their solutions require numerical techniques, like Newton-Raphson Method

4 Neural networks

Artificial intelligence (AI) is used to develop algorithms that enable machines to perform cognitive tasks. Any AI system should have 3 abilities: (1) store knowledge, (2) apply the stored knowledge to solve problems, and (3) acquire new knowledge through experience. Any AI system is comprised of representation, reasoning, and learning [3].

Artificial Neural Networks (ANN) are among the biggest advancements in artificial intelligence (AI) computation method. ANN has more advantages than traditional AI methods due to its ability to solve non-linear problems for predicting data involving multiple inputs and outputs.

Recent advances in statistical modeling have increasingly leveraged artificial intelligence techniques, especially neural networks, to transcend the limitations of classical estimation methods. For instance, transformer-based frameworks now allow distributional parameter estimation without requiring explicit likelihood functions, making them applicable to varied distribution families.[9]

Furthermore, learning-based approaches in domains such as microstructural MRI and dynamical systems demonstrate that neural networks not only accelerate parameter recovery but also enhance robustness in scenarios where traditional solvers struggle with ill-posed or high-dimensional inputs.[10]

Together, these studies underscore a global trend: neural-network-based estimation methods are expanding the scope of classical statistical inference, enabling researchers to estimate complex distributions and model parameters in a data-driven, computation-efficient manner.

ANN consists of many interconnected processing units (neurons) that are organized into weightily linked layers. Each ANN is associated with an activation function which decide whether a neuron should be activated or not. The neural network used in this paper is the MLP (multilayer perceptron) neural network, where the back-propagation training rule is used. [8] developed the back-propagation training rule, See [3] for further details about the fundamentals of artificial neural networks. Any neural network may be single layer Perceptron

or Multi Layer Perceptron We will only discuss Multi Layer Perceptrons below since they are more useful than Single Layer Perceptrons for practical applications today.

4.1 MLP neural networks

Multilayer perceptrons have been applied successfully to solve different and difficult problems through supervised training with a highly popular algorithm developed by Rumelhart et al 1986 known as the error back-propagation algorithm [8]. This algorithm is based on the error correction learning rule [3]. Figure 2 illustrates an MLP topology with a single hidden layer. In an MLP network, neurons are arranged in a layered frontal topology, each neuron has its output as a weighted sum of two sets of inputs: input signal passing through a transfer function before being pulled together and a bias.

In MLP, an input is connected to every neuron in the first hidden layer, and the output of every neuron in the first layer is connected to every neuron in the second layer and so on. Hence, the determination of the number of hidden layers and the neurons within these layers is of great importance [3]. The number of input and output neurons is determined by the defining problem. However, there might be some uncertainty regarding which specific input to use. If the needed number of hidden units is uncertain, start with the use of a single hidden layer and estimate the number of neurons in it [1].

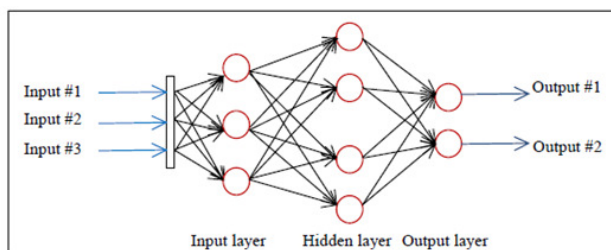


Fig. 2. Topology of an MLP with a single hidden layer. Source: [1].

4.2 Training the MLP

After selecting the number of layers and number of units within each layer, the training algorithms are responsible for setting the network's weights and thresholds (biases) to minimize the prediction error. Back propagation algorithm is one of the most efficient algorithms used to train MLP; it minimizes the error and adjust the weights and thresholds automatically by using historical cases gathered by the user [3]. It is equivalent to fitting the model represented by the network to the training data available.

Errors in the network configuration can be detected by running all the training cases through the network, comparing the actual output to the target / desired

outputs. The error function can combine differences together to get the network error. The Sum of Squared Error (SSE) is the most common error function used in the literature, where we square and sum together the individual output unit errors in each case.

4.3 Back-Propagation algorithm

It is an algorithm used to monitor learning. It uses the methods of mean square error and gradient descent to realize a modification to the connection weight of network. This modification to the connection weight is required to achieve the minimum error sum of squares. In this algorithm, small value is given to the connection value of network first, and then, a training sample is selected to calculate gradient of error relative to this sample.

The BP learning process is composed of:

(1) Forward propagation of operating signal: the input signal is propagated from the input layer, via the hidden layer, to the output layer. During the forward propagation of the operating signal, the weight value and offset value of the network are maintained constant and the status of each layer the of neuron will only exert an effect on that of next layer of neuron, If the expected output is not be achieved in the output layer, it can be switched into the back propagation of error signal.

(2) Back propagation of error signal: the difference between the real output and expected output of the network is defined as the error signal; in the back propagation of error signal, the error signal is propagated from the output end to the input layer in a layer-by-layer manner. During the back propagation of error signal, the weight value of network is regulated by the error feedback. Ongoing modification of weight value and offset value make the real output of network closer to the expected one [7].

4.4 Using perceptron neural network to estimate Gamma parameters

Let $X_1, X_2, \dots, X_n$ be a random sample drawn from a Gamma distribution with parameters α and β . The objective of this study is to estimate the unknown parameters α and β . Instead of relying solely on classical estimators such as the method of moments or maximum likelihood estimation, we propose a data-driven approach based on artificial neural networks. In this framework, a set of descriptive statistics computed from the sample, namely the mean, standard deviation, median, skewness, and kurtosis, are used as input variables to the neural network. The trained ANN learns a nonlinear mapping between these statistics and the corresponding parameters of the Gamma distribution. In order to use of neural network to estimate Gamma parameters, given the values of mean, standard

deviation, median, skewness and kurtosis which are functions of the two parameters of a Gamma distribution, we use them as the inputs to the network [1].

Each input is a subset of the measures. We use a full set of inputs to increase the accuracy of the proposed neural network, however; increasing the number of inputs can cause major computational problems. The Gamma parameters are considered the outputs of the designed network.

The application of MLP networks in Figure 3 is consist of the following steps:

- Step 1 selecting the network structure.
- Step 2 preparing the training data.
- Step 3 choosing the activation function.
- Step 4 choosing the training algorithm.

A multilayer perceptron (MLP) architecture with a single hidden layer was adopted to estimate the parameters of the Gamma distribution. The network utilized five descriptive statistics—mean, standard deviation, median, skewness, and kurtosis—as input features, while the two Gamma parameters were produced at the output layer. Consequently, the network configuration consisted of five neurons in the input layer and two neurons in the output layer. During the training stage, several experimental trials were conducted to determine the appropriate size of the hidden layer, and 25 neurons were ultimately selected. The complete architecture of the proposed neural network is illustrated in Figure 3.

For the activation mechanism, the hyperbolic tangent sigmoid transfer function

$$\text{Tansig}(n) = \frac{2}{1 + e^{-2n}} - 1$$

was employed in both the input and hidden layers due to its suitability for nonlinear prediction and mapping tasks. In contrast, the output layer used a linear activation function,

$$\text{Purelin}(n) = n,$$

allowing the estimated parameters (α, β) to take values greater than one.

To prepare the training data, 80,000 random observations were generated from the Gamma distribution using different combinations of parameters. Specifically, the values of α and β were varied within the interval from 1 to 10. For each generated sample, the statistical measures of mean, standard deviation, median, skewness, and kurtosis were computed and used as the input variables for the neural network. The corresponding true values of the Gamma parameters served as target outputs during the training process. After organizing the dataset, a total of 1,000 input-output pairs were obtained and used in the training phase. The trained neural network demonstrated satisfactory performance in

estimating the parameters when their values were less than 10. Nevertheless, this may be viewed as a limitation, since the effectiveness of the proposed estimation approach is mainly restricted to this particular parameter range. The selected neural network was trained using the backpropagation algorithm. After 2870 training epochs, the learning process converged to a sufficiently small Mean Square Error (MSE) value of 0.0001214. This low error level indicates that the trained neural network effectively captures the underlying relationship between the input features and the output parameters, demonstrating its suitability as a reliable mapping approach for estimating the desired parameters.

The choice of the input statistics is motivated by their ability to capture key characteristics of the underlying distribution. The mean and standard deviation describe the location and scale of the sample, while skewness and kurtosis capture the asymmetry and tail behavior of the distribution. The median provides an additional robust measure of central tendency. Since the Gamma distribution is fully characterized by two parameters, α and β these descriptive statistics collectively provide sufficient information about the shape and scale of the distribution. The neural network is therefore able to learn the nonlinear mapping between these sample statistics and the corresponding Gamma parameters through the training process.

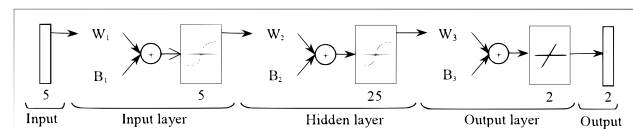


Fig. 3. Structure of the proposed neural network.

4.5 simulation study

A simulation study was conducted to examine the performance of the developed artificial neural network (ANN) in comparison with the Method of Moments and the Maximum Likelihood estimation approaches for estimating the parameters of the Gamma distribution. For each estimation method and across different sample sizes, 10,000 samples were generated in order to obtain the required input variables. The two classical methods together with the proposed ANN-based approach were then applied to estimate the parameters of the Gamma distribution. The simulation results for the applied methods are summarized in terms of the following measures:

- the mean and standard deviation of each estimated parameter based on 10,000 replications.
- the mean square error (MSE) and the corresponding 95% confidence interval for each estimated parameter.

5 Main results

We used ANN method trying different number of inputs in the input layer, multiple combinations of primary inputs were performed. Our chosen neural network is trained using back propagation algorithm. We used input 5, input 3, and input2 in a total of 15 combination as shown in the tables to assess the impact of changing the inputs in input layer on model quality. The mean, relative bias, and square error were calculated for different gamma distribution parameter values were as follows, $\alpha = (2, 3, 6.5 \text{ and } 9)$ and $\beta = (2, 4.9 \text{ and } 8)$. for sample size(30, 50,100,200).

Results showed that all the methods used yield good results in the form of small MSE and relative bias on other hand the sample size doesn't substantially affect the quality of results which will promote the use of small sample size in the future.

To evaluate the robustness of the proposed ANN model, we investigated the stability of the predictions under small perturbations of the input statistics. In addition, we examined the generalization ability of the model for parameter values outside the training range.

Table 1, Table 2, Table 3, Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, Table 10, Table 11, Table 12, Table 13, Table 14 and Table 15 present mean estimates ($\hat{\alpha}, \hat{\beta}$), bias, relative bias and confidence interval for three methods of estimation ANN, MLE and moments methods for $n = 30, 50, 100$ and 200 and $\alpha = 2, 3, 6.5$ and 9 and $\beta = 2, 4.9$ and 8 .

A comparison between the proposed ANN-based estimator, the maximum likelihood method, and the method of moments was conducted through an extensive simulation study. The comparison was performed using several quantitative performance measures including bias, standard deviation, mean squared error (MSE), and confidence intervals across different parameter settings and sample sizes. The results presented in Table 1, Table 2, Table 3, Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, Table 10, Table 11, Table 12, Table 13, Table 14, Table 15 and Table 16 indicate that the ANN-based estimator generally achieves lower bias and smaller mean squared error compared with the classical estimators in many of the considered scenarios. In addition, the computational time required by the ANN approach was significantly smaller than that of the other estimation methods. These findings demonstrate that the proposed neural network method provides an efficient and practical alternative for estimating the parameters of the Gamma distribution.

Table 17 and Table 18 report the ANN relative bias and MSE by sample size.

Table 1. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 2, \beta = 2$, and $n = 30$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	2.076	0.076	0.038	1.975	2.011
	β	2.001	0.001	0.001	2.002	2.041
MLE	α	2.197	0.197	0.099	1.921	2.01
	β	1.935	0.065	0.032	1.899	2.1
Moments	α	2.303	0.303	0.151	1.912	2.213
	β	2.292	0.292	0.146	1.845	2.152

Table 2. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 2, \beta = 2$, and $n = 50$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	2.006	0.006	0.003	1.97	2.002
	β	2.016	0.016	0.008	1.991	2
MLE	α	2.111	0.111	0.056	1.925	2
	β	1.962	0.038	0.019	1.903	2.09
Moments	α	2.171	0.171	0.086	1.916	2.202
	β	2.176	0.176	0.088	1.849	2.141

Table 3. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 2, \beta = 2$, and $n = 100$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	2.003	0.003	0.002	1.945	1.966
	β	2	0	0	1.975	1.981
MLE	α	2.056	0.057	0.028	2.002	2.041
	β	1.984	0.016	0.008	1.921	2.01
Moments	α	2.092	0.092	0.046	1.899	2.1
	β	2.098	0.098	0.049	1.912	2.213

Table 4. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 2, \beta = 2$, and $n = 200$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	2.006	0.006	0.003	1.919	2.012
	β	2.004	0.004	0.002	1.897	2.083
MLE	α	2.026	0.026	0.013	1.91	2.195
	β	1.991	0.009	0.005	1.793	2.135
Moments	α	2.043	0.043	0.021	1.789	2.254
	β	2.046	0.046	0.023	1.879	2.336

Table 5. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 3$, $\beta = 4.9$, and $n = 30$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	3.212	0.212	0.071	2.955	3.457
	β	5.214	0.314	0.064	4.498	4.926
MLE	α	3.296	0.296	0.099	2.874	3.652
	β	4.744	-0.156	-0.032	4.453	4.999
Moments	α	3.234	0.234	0.078	2.784	3.426
	β	5.725	0.825	0.168	4.765	5.026

Table 6. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 3$, $\beta = 4.9$, and $n = 50$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	3.064	0.064	0.021	2.939	3.161
	β	5.011	0.111	0.023	4.755	4.941
MLE	α	3.18	0.18	0.06	2.897	3.343
	β	4.795	0.105	0.022	4.789	5.024
Moments	α	3.142	0.142	0.047	2.954	3.456
	β	5.368	0.468	0.095	4.777	5.12

Table 7. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 3$, $\beta = 4.9$, and $n = 100$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	3.017	0.017	0.006	2.951	3.009
	β	4.917	0.017	0.003	4.852	4.881
MLE	α	3.089	0.09	0.03	2.853	3.123
	β	4.846	-0.053	-0.011	4.785	4.985
Moments	α	3.068	0.068	0.023	2.942	3.213
	β	5.121	0.221	0.045	4.866	5.023

Table 8. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 3$, $\beta = 4.9$, and $n = 200$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	3	0	0	2.964	3.005
	β	4.906	0.006	0.001	4.894	4.912
MLE	α	3.038	0.038	0.013	2.996	3.124
	β	4.884	0.016	0.003	4.875	5.124
Moments	α	3.034	0.034	0.011	2.958	3.025
	β	5.009	0.109	0.022	4.786	5.195

Table 9. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 6.5$, $\beta = 2$, and $n = 30$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	6.694	0.194	0.03	5.96	6.413
	β	2.225	0.225	0.113	2.023	2.407
MLE	α	7.175	0.675	0.104	5.789	6.784
	β	1.938	0.062	0.031	1.896	2.245
Moments	α	6.908	0.408	0.063	6.025	6.895
	β	2.63	0.63	0.315	1.972	2.312

Table 10. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 6.5$, $\beta = 2$, and $n = 50$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	6.504	0.004	0.001	6.267	6.491
	β	2.023	0.023	0.012	2	2.183
MLE	α	6.89	0.39	0.06	6.324	6.789
	β	1.967	0.033	0.017	1.942	2.125
Moments	α	6.75	0.25	0.038	6.421	6.875
	β	2.405	0.405	0.202	1.967	2.326

Table 11. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 6.5$, $\beta = 2$, and $n = 100$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	6.513	0.013	0.002	6.429	6.491
	β	2.002	0.002	0.001	1.97	2.02
MLE	α	6.684	0.184	0.028	6.173	6.75
	β	1.981	0.019	0.009	1.97	2.234
Moments	α	6.623	0.123	0.019	6.229	6.861
	β	2.205	0.205	0.102	1.913	2.3

Table 12. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 6.5$, $\beta = 2$, and $n = 200$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	6.502	0.002	0	6.486	6.591
	β	2.005	0.005	0.002	1.985	2.016
MLE	α	6.598	0.098	0.015	6.416	6.478
	β	1.988	0.012	0.006	1.966	2.016
Moments	α	6.568	0.068	0.01	6.161	6.737
	β	2.117	0.117	0.058	1.966	2.229

Table 13. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 9$, $\beta = 8$, and $n = 30$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	9.011	0.011	0.001	8.983	9.006
	β	8.007	0.007	0.001	8.011	8.034
MLE	α	9.908	0.908	0.101	8.789	9.215
	β	7.786	0.214	0.027	7.748	8.236
Moments	α	9.037	1.037	0.13	8.657	9.245
	β	9.855	0.855	0.095	7.854	8.125

Table 14. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 9$, $\beta = 8$, and $n = 50$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	9.03	0.03	0.003	8.985	8.993
	β	8.02	0.02	0.002	8.055	8.064
MLE	α	9.596	0.596	0.066	8.713	9.007
	β	7.817	0.183	0.023	7.77	8.035
Moments	α	8.615	0.615	0.077	8.525	9.216
	β	9.526	0.526	0.058	7.516	8.237

Table 15. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 9$, $\beta = 8$, and $n = 100$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	9.011	0.011	0.001	8.981	8.987
	β	8.001	0.001	0	8.022	8.027
MLE	α	8.263	0.263	0.029	8.788	9.123
	β	7.925	0.075	0.009	7.895	8.426
Moments	α	8.285	0.285	0.036	8.895	9.252
	β	9.258	0.258	0.029	7.956	8.542

Table 16. Parameter estimates, bias, relative bias, and confidence limits for $\alpha = 9$, $\beta = 8$, and $n = 200$.

Method	Param.	Est.	Bias	Rel. bias	CI lower	CI upper
ANN(5)	α	9.014	0.014	0.002	8.955	8.807
	β	8.015	0.015	0.002	8.183	7.867
MLE	α	9.129	0.129	0.014	8.964	8.941
	β	7.964	0.036	0.004	7.994	8.257
Moments	α	8.156	0.156	0.02	8.963	9.067
	β	9.143	0.143	0.016	7.952	8.371

Table 17. ANN relative bias and MSE for $\alpha = 2$ and $\beta = 2$ across sample sizes.

n	Parameter	Relative bias	MSE
30	α	0.0382	0.0024
30	β	0.0007	0.0001
50	α	0.0028	0.000
50	β	0.0081	0.000
100	α	0.0017	0.0001
100	β	0.0002	0.000
200	α	0.0028	0.000
200	β	0.0018	0.000

Table 18. ANN relative bias and MSE for $\alpha = 9$ and $\beta = 8$ across sample sizes.

n	Parameter	Relative bias	MSE
30	α	0.0007	0.000
30	β	0.0013	0.0001
50	α	0.0022	0.000
50	β	0.0037	0.000
100	α	0.0001	0.000
100	β	0.0014	0.000
200	α	0.0016	0.000
200	β	0.0018	0.000

6 Discussion

This study establishes artificial neural networks (ANNs) as a viable and effective alternative for estimating the parameters of the Gamma distribution. In comprehensive simulations, the proposed ANN-based estimator demonstrated performance that was competitive with, and often superior to, classical methods—including Maximum Likelihood Estimation (MLE), the Method of Moments (MM), and Least Squares Estimation (LSE) as measured by relative bias and mean square error (MSE).

Across a range of sample sizes ($n = 30, 50, 100, 200$), a consistent pattern emerged: the ANN approach yielded more stable and precise estimates. This advantage was particularly pronounced for small to moderate samples ($n = 30$ and 50), where traditional estimators are known to be less reliable. While all methods improved with larger samples, the ANN maintained a notable performance edge in small-sample scenarios. This is attributed to the network's capacity to model complex, nonlinear relationships between a set of input sample statistics (e.g., mean, standard deviation, skewness, kurtosis) and the target parameters (α and β). By leveraging multiple descriptive statistics as inputs, the ANN learned intricate dependencies that are not explicitly captured by conventional estimators, resulting in reduced bias and MSE across most experimental conditions.

Furthermore, the predictive power of the

ANN was enhanced by the breadth of its input features. Utilizing a richer set of inputs—particularly combinations incorporating the mean, median, skewness, kurtosis, and standard deviation—consistently led to better performance. Simpler input sets (e.g., mean and standard deviation alone) resulted in marginally higher bias and MSE, underscoring that the network's efficacy is bolstered by more comprehensive information about the sample distribution. As the sample size increased to $n=200$, the ANN estimates converged rapidly to the true parameter values, confirming the method's statistical consistency and efficiency.

These findings resonate with a growing body of literature that integrates machine learning into statistical inference. The flexibility and universal approximation properties of ANNs allow them to navigate complex parameter mappings that are analytically intractable, offering a significant advantage where likelihood functions are complex or computationally expensive. Thus, ANNs can be positioned not merely as a supplementary tool, but as a robust alternative to traditional parametric estimation.

It is important to note that the ANN's performance is contingent on several factors, including the selection of input variables, network architecture, and training regimen. Meticulous design and tuning are therefore critical to prevent overfitting and ensure generalizability. Notwithstanding these requirements, the simulation evidence strongly suggests that neural network-based estimation provides a powerful and adaptable framework for parameter estimation in continuous probability distributions such as the Gamma model.

The numerical results presented in Table 9, Table 10, Table 11, Table 12, Table 13, Table 14, Table 15 and Table 16 provide clear evidence of the advantage of the ANN-based estimator over classical estimation methods. For example, when the sample size is $n=30$, the bias of the ANN estimator for α is 0.194, whereas the corresponding bias values for the maximum likelihood and method of moments estimators are 0.675 and 0.408, respectively. Similar improvements are observed for $n=50$, where the ANN estimator achieves significantly smaller bias compared with the classical estimators. As the sample size increases, the performance of all estimators improves; however, the ANN method maintains competitive performance with consistently low bias and narrow confidence intervals. These results indicate that the neural network estimator is particularly advantageous in small-sample scenarios where traditional estimators tend to be less reliable.

Furthermore, the observed decrease in mean square error (MSE) with the inclusion of a larger number of input statistics indicates that the complexity of the neural network plays a crucial role in improving

estimation accuracy. As more descriptive features such as skewness, kurtosis, and median were incorporated, the model was able to capture more subtle characteristics of the underlying Gamma distribution. This enhanced feature representation allowed the network to learn more efficient mappings between sample statistics and distribution parameters, thereby reducing both bias and estimation error. However, this improvement also highlights the importance of selecting an appropriate network complexity to avoid overfitting while maintaining generalization capability.

7 Conclusion

The parameters of any statistical distribution could be estimated using common traditional methods as method of moment and maximum likelihood estimation. In the current article we compared those methods with the use of artificial neural network to estimate the parameters of gamma distribution.

Using Python, the artificial neural network estimator with 5 nodes was more accurate compared to the 2 nodes model when assessing. Thus we used the 5 nodes model in the comparison with method of moment and maximum likelihood estimation (using python) in obtaining the bias, relative bias and confidence interval.

The findings suggested that the artificial neural network estimator with 5 nodes in the input layer was more accurate and was associated with lower bias and relative bias when compared to the moment method. Moreover, it was observed that the bias value obtained by the artificial neural network was not affected by the sample size, thus even in small sample size, good estimation was obtained. This was in the contrary to the application of moment method, where the sample size affected the bias value.

the artificial neural network estimator was faster, and more accurate to estimate the parameters of gamma distribution compared to method of moment and maximum likelihood estimation.

The robustness and adaptability of the ANN framework make it a promising tool for complex estimation problems where analytical methods may fall short. Nevertheless, further research is warranted to explore several promising directions. Future work could investigate the impact of different network architectures, activation functions, and optimization algorithms on the estimation accuracy. In addition, extending the proposed approach to other continuous distributions, or applying it to real-world datasets, could further validate its practical applicability. Integrating hybrid models that combine neural networks with Bayesian or resampling techniques might also yield enhanced performance and interpretability.

Overall, the results provide compelling evidence

that data-driven neural estimation methods can significantly advance classical statistical modeling, paving the way for new methodologies in modern computational statistics.

To evaluate the computational efficiency of the estimation procedures, the three methods discussed in this article were implemented and comparatively analyzed. All algorithms were developed using Python 3 and executed within the same computational environment to ensure consistency and fairness in the comparison. In addition, all experiments were performed on the same machine so that any observed differences in performance could be attributed solely to the methodological differences rather than variations in hardware or software conditions. The execution time required by each method was measured and recorded. The obtained results indicate a substantial difference in computational time, with the proposed neural network-based approach demonstrating significantly faster performance compared with the other two estimation methods. This finding highlights the computational advantage of the ANN-based approach in practical applications.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

During the preparation of this work, the author(s) utilized artificial intelligence (AI) technologies solely for the purpose of linguistic refinement and grammatical correction. Following the application of these tools, the author(s) comprehensively reviewed, verified, and edited the resulting text. The author(s) maintain full accountability and responsibility for the accuracy, integrity, and final content of the publication.

References:

- [1] Abbasi, B., Rabelo, L., & Hosseinkouchack, M. (2008). Estimating parameters of the three-parameter Weibull distribution using a neural network. *European Journal of Industrial Engineering*, 2(4), 428–445
- [2] Alzaatreh, A., Lee, C., & Famoye, F. (2015). Family of generalized gamma distributions: Properties and applications. *Hacettepe Journal of Mathematics and Statistics*, 45(3), 869–886.
- [3] Haykin, S. (1994). *Neural networks: A comprehensive foundation*. Prentice-Hall.
- [4] Hoog, R. V., McKean, J. W., & Craig. (2019). *Introduction to mathematical statistics*. Pearson Publishing.
- [5] Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous univariate distributions*. John Wiley & Sons.
- [6] Kirimi, E., Ouko, A., & Kipkoech, C. W. (2014). Modified moment estimation for a two-parameter gamma distribution [Conference paper or unpublished manuscript].
- [7] Li, J., Cheng, J.-h., Shi, J.-y., & Huang, F. (2012). Advances in computer science and information engineering. In *Proceedings of the International Conference on Computer Science and Information Engineering* (pp. 553–558). Springer.
- [8] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. <https://doi.org/10.1038/323533a0>
- [9] de Almeida Martins, J. P., Nilsson, M., Lampinen, B., Palombo, M., While, P. T., Westin, C. F., & Szczepankiewicz, F. (2021). Neural networks for parameter estimation in microstructural MRI: Application to a diffusion-relaxation model of white matter. *NeuroImage*, 244, 118601. <https://doi.org/10.1016/j.neuroimage.2021.118601>
- [10] Yin, X., & Yin, D. S. (2024). Transformer-based parameter estimation in statistics. *Mathematics*, 12(7), 1040. <https://doi.org/10.3390/math12071040>

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflicts of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0. https://creativecommons.org/licenses/by/4.0/deed.en_US