Maximum Entropy Learning with Neural Networks

Woraphon Yamaka(✉)

Center of Excellence in Econometrics, Faculty of Economics,
Chiang Mai University, Chiang Mai 50200, Thailand

Abstract. Conventionally, the back-propagation (BP), maximum likelihood (ML) and Bayesian approaches have been applied to train Artificial Neural Networks (ANN). This study presents a Generalized Maximum Entropy (GME) learning algorithm for ANN, designed specifically to handle limited training data and unknown error distribution. Maximizing only the entropy of parameters in the ANN allows more effective generalization capability, less bias towards data distributions, and robustness to over-fitting compared to the conventional algorithm learning. In the implementations, GME is compared with the conventional algorithms in terms of their forecasting performances in both simulation and real data studies. The findings demonstrate that GME outperforms other competing estimators when training data is limited and the distribution of the error is unknown.

Keywords: Artificial neural network · Comparison of estimators · Entropy

1 Introduction

Neural networks have received considerable attention in recent years, for being a self-learning and self-adaptive model with the powerful abilities in handling the nonlinear problem and complex issue (Chen et al. 2018; Ramos et al. 2021). Recently, the technique has been utilized in many purposes like prediction and classification (Chen et al. 2018; Ramos et al. 2021; Yamaka, Phadkantha, and Maneejuk 2021). In this study, I aim at introducing an alternative algorithm, which is the generalized maximum entropy estimation (GME) (Golan, Judge, and Miller 1996), to artificial neural networks (ANN) to improve the prediction performance.

Estimation of the neural network parameters is quite challenging as it needs to adjust the weights and biases to ensure that the output is close to the desired output (Lin et al. 2016). Many estimation techniques and concepts have been proposed and developed to tune weight and bias parameters in the neural networks (Chon and Cohen 1997). It should be noted that these parameters are both learnable parameters which are used to link the input layer, the hidden layer and the output together. For example, if we have a single layer network, the input data is multiplied with the weight parameter; then a bias is added before passing the transformed input data to the next hidden layer. Next, the output layer can be obtained by multiplying the transformed input data with another

N. Ngoc Thach et al. (Eds.): *Optimal Transport Statistics for Economics and Related Topics*,
SSDC 483, pp. 150–162, 2024.
https://doi.org/10.1007/978-3-031-35763-3_8

weight parameter followed by the inclusion of an additional bias to obtain the output. Traditionally, parameters are estimated using the methods of back-propagation (BP) (White 1989), maximum likelihood (ML) (Gish 2020) and Bayesian (Müller and Insua 1998).

From the computational point of view, the BP algorithm minimizes the cost function, which is commonly assumed to be mean square error, in order to obtain the optimal parameters. Many iterative learning steps are required in the learning process to obtain a better learning performance. However, it is well known that given the cost function as mean square error, it leads to the strong assumption that all the feature components are equivalent (Wang, Du and Wang 2017). Thus, Gish (2020) proposed a probabilistic view of neural networks to derive the maximum likelihood estimation. Specifically, the cost function of the BP algorithm is replaced by the likelihood function. The basic concept of the ML method is that the optimal parameters should be chosen such that the probability of the observed sample data is maximized. This estimation has several attractive properties including: consistency, asymptotic normality, and efficiency when the sample size approaches infinity (Chen et al. 2013). Lin et al. (2016) argued that although the learning process of these estimators are generalized correctly to the new inputs after sufficient training, the learning speed is slow and is not incremental in nature (old input should still be trained with the new input) (Fu, Hsu, Principe 1996). Also, if we limit the training data to reduce the computational cost of the estimations and gain a better control over the training data, we may face the overfitting problem (Chu et al. 2021). It should be noted that overfitting occurs when the network has memorized the training input, but it has not learned to generalize to new inputs, leading to overconfident predictions. In the Bayesian approach, these issues can be handled in a natural and consistent way. The non-informative priors are used to handle the complexity of the data and network; as a result, the model is weighted by the posterior probability given the data sample. However, this estimation still suffers from some complicated problems such as the training time, the efficient parameter estimation, the random walk in the high-dimensional parameter cases (Kocadağlı 2015).

According to the above view about the estimation methods, despite these estimations generally perform well, they have inherent additional limitations (Kocadağlı and Aşıkgil 2014; Lin et al. 2016; and Yang, Baraldi, and Zio 2016). First of all, in the cases of ML and Bayesian, determining the most suitable distribution (likelihood and posterior distributions) requires an expert, otherwise it is possible to construct the incorrect functional structure. Secondly, when the neural network model is being trained using the BP and ML, a large training data is required. Thirdly, it has often been found that BP and ML are prone to overfitting (Dorling et al. 2003). Thus, we need to limit the complexity of the network making it suitable to the learning problem defined by the data (Bishop 1995).

To overcome these limitations, GME is suggested to estimate the weight and bias parameters of ANN. GME-ANN can be one of the popular neural networks models for dealing with prediction problem. This study aims at investigating the possibility of developing a ANN model based on the use of GME. Unlike ML and Bayesian, before ANNs are being trained, the prior information regarding the likelihood and posterior distributions are not required. GME allows us to produce methods that are capable of learning complex behaviors without human intervention. It also has an ability to fit the data without making specific assumptions; therefore, I hypothesized that estimation with GME (GMS-ANN) would enable the resulting parameter estimates to be more unbiased to data distributions and robust to over-fitting issues compared to those ML, and Bayesian. In addition, there are many pieces of evidence confirming the high estimation performance of GME (Alibrandi and Mosalam 2018; Maneejuk, Yamaka, and Sriboonchitta 2020), despite small sample size and limited training data. In this study, thus, the performance of each estimation approach and their relative performance with a focus on small sample sizes are investigated.

The rest of this paper is organized as follows. Section 2 describes the proposed methodology. Section 3 presents the experiment studies. The real data example is reported in Sect. 4. Finally, Sect. 5 provides the conclusion of this study.

2 Model Setup

The idea is to build an entropy function with a neural network constraint to replace the loss function or probability function discussed in the previous section. In other words, the GME is used as the estimator to adjust the weights and biases of the neural network by maximizing the Shannon entropy with the ANN equation constraint. In particular, weights and biases in ANN are reparametrized as the discrete random variables on bounded supports. The sum of entropy distributions of the weights and biases is maximized subject to model consistency constraints. The weights and biases of interest are then calculated as the expectation of random variables on the prescribed supports under the derived distributions of the entropy maximization.

2.1 ANN with Three Layers

In this section, I provide three layered ANN consisting of an input layer with I input neurons, one hidden layer with H hidden neurons, and one output layer, as the example. Mathematically, the hidden and input layers of ANN can be expressed as

$$y_i = \sum_{h=1}^{H} \left\{ \omega_h^O f^I \left(\sum_{k=1}^{K} \omega_{k,h}^I x_{i,k} + b_h^I \right) + b_h^O \right\} + \varepsilon_i, \tag{1}$$

where y_i, for $t = 1, ..., T$, and $x_{i,k}$, for $k = 1, ..., K$, are output and input variables, respectively. $\omega_{k,h}^I$ is the weight parameter of input $x_{i,k}$ that connects the input $x_{i,k}$ and the hth neuron in the hidden layer, b_h^I is the bias for hth neuron in the hidden layer. f^I is the activation function that provides the nonlinearity to the ANN structure, and scales its received inputs to its output range. In this study, the logistic function is employed as it is easy to calculate and its first derivative is simple (Kocadağlı and Aşıkgil 2014). Likewise, I use ω_h^O and b_h^O to denote weight and bias terms, respectively. ε_i is the error term.

Learning occurs through the adjustment of the path weights and node biases. Traditionally, all the weight and bias parameters are estimated by the BP method. The optimal parameters are estimated by minimizing the squared difference between observed output and estimated output. The loss function can be written as follows,

$$Loss = \frac{1}{N}\sum_{i=1}^{N}\left\{y_i - \sum_{h=1}^{H}\left\{\omega_h^O f^I\left(\sum_{k=1}^{K}\omega_{k,h}^I x_{i,k} + b_h^I\right) + b_h^O\right\}\right\}, \tag{2}$$

2.2 Maximum Entropy Learning for ANN Model

In this study, the maximum entropy (ME) of Jaynes (1982) is generalized to estimate weights and biases in the ANN equation. As I mentioned before, all parameters are calculated as the expectation of random variables on the prescribed supports under the derived distributions of the entropy. More precisely, the random variables are treated as the probabilities and the information entropy of these probabilities can be measured by Shannon's entropy (Shannon 1948)

$$H(\mathbf{p}) = -\sum_d p_d \log p_d, \tag{3}$$

where p_d is the probability of the possible outcome d, such that $\sum_d p_d = 1$. Under this maximum entropy principle, the distribution is chosen for which the information is just sufficient to determine the probability assignment. In addition, it seeks information within the data without imposing arbitrary restrictions. In this study, I follow the idea of Golan, Judge, and Miller (1996) and generalize the ME solution to the inverse problems with error, expressed in the ANN framework.

To estimate the unknown parameters in Eq. (1), say ω_h^O, $\omega_{k,h}^I$, b_h^O and b_h^I, for $h = 1, ..., H$ and $k = 1, ..., K$, we reparameterize them as the expectation of weights on the prescribed supports. The weight parameters. Each weight has a bounded support space, $\mathbf{z}_{hk} = [\underline{z}_{hk,1}, ..., \overline{z}_{hk,M}]$, associated with the hth neuron and kth variable, which is symmetrically built around zero and weighted by the vector $\mathbf{p}_{hk} = [p_{hk,1}, ..., p_{hk,m}]$. Note that $\underline{z}_{hk,1}$ and $\overline{z}_{hk,M}$ are, respectively, the lower and the upper bounds. In the ANN structure, there are input and output weights, and hence the output and input probability vectors ($\mathbf{p}_h^O = [p_{h,1}^O, ..., p_{h,M}^O]$ and $\mathbf{p}_{hk}^I = [p_{hk,1}^I, ..., p_{hk,M}^I]$) associated with output and input supports ($\mathbf{z}_h^O = [\underline{z}_{h,1}^O, ..., \underline{z}_{h,M}^O]$ and $\mathbf{z}_{hk}^I = [\underline{z}_{hk,1}^I, ..., \underline{z}_{hk,M}^I]$) are introduced in this reparameterization. Thus, I reparameterize ω_h^O and $\omega_{k,h}^I$ as

$$\begin{aligned}\omega_h^O &= \sum_{m=1}^{M} z_{h,m} p_{h,m}^O \\ \omega_{hk}^I &= \sum_{m=1}^{M} z_{hk,m} p_{hk,m}^I\end{aligned} \tag{4}$$

where $p_{h,m}^O$ and $p_{hk,m}^I$ are output and input probability estimates specified on the supports $z_{h,m}$ and $z_{hk,m}$ respectively. In terms of b_h^O and b_h^I, the reparameterization of these biases is also somehow analogous to the weight parameter representation in probability and compact supports,

$$\begin{aligned} b_h^O &= \sum_{m=1}^{M} r_{h,m} q_{h,m}^O \\ b_h^I &= \sum_{m=1}^{M} r_{h,m} q_{h,m}^I \end{aligned} \tag{5}$$

where $q_{h,m}^O$ and $q_{h,m}^I$ are, respectively, the output and input probability estimates specified on the supports $r_{h,m}$. Just like the estimated weights and biases, the error ε_i is also viewed as the expected mean value of finite support v_i. Again, we can view error as the expected values of a random variable defined on a probability distribution. Thus, ε_i has a bounded support space $\mathbf{v}_i = [\underline{v}_{i,1}, ..., \overline{v}_{i,M}]$, associated with i th observation, and weighted by the vector $\mathbf{w}_i = [w_{i,1}, ..., w_{i,M}]$.

$$\varepsilon_i = \sum_{m=1}^{M} v_i w_{im}, \tag{6}$$

Pukelsheim (1994) suggested using the three-sigma rule for setting the support space of the error, such that $\underline{v}_{i1} = -3\sigma$ and $\overline{v}_{iM} = 3\sigma$, where σ is the standard deviation of y. Now, the ANN model (Eq. 1) under the reparameterization becomes

$$y_i = \sum_{h=1}^{H} \left\{ \left(\sum_{m=1}^{M} z_{h,m} p_{h,m}^O \right) f^I \left(\sum_{k=1}^{K} \sum_{m=1}^{M} z_{hk,m} p_{hk,m}^I x_{i,k} + \sum_{m=1}^{M} r_{h,m} q_{h,m}^I \right) + \sum_{m=1}^{M} r_{h,m} q_{h,m}^O \right\} + \sum_{m=1}^{M} v_i w_{im}, \tag{7}$$

The entropy term is maximized subject to the requirements of the proper probability distributions for $p_{h,m}^O$ $p_{hk,m}^I$, $q_{h,m}^O$, $q_{h,m}^I$ and $w_{i,m}$ and the N information-moment constraints of the ANN model. These unknown probabilities are assumed to be independent and can be estimated jointly by solving the constrained optimization problem with an objective function based on Shannon's entropy and constrains.

$$\begin{aligned} \mathbf{H}(\mathbf{p}^I, \mathbf{p}^O, \mathbf{q}^I, \mathbf{q}^O, \mathbf{w}) = & \underset{\mathbf{p}^I, \mathbf{p}^O, \mathbf{q}^I, \mathbf{q}^O, \mathbf{w}}{\mathbf{argmax}} \left\{ -\mathbf{H}(\mathbf{p}^I) - \mathbf{H}(\mathbf{p}^O) - \mathbf{H}(\mathbf{q}^I) - \mathbf{H}(\mathbf{q}^O) - \mathbf{H}(\mathbf{w}) \right\} \\ & = -\sum_{h=1}^{H} \sum_{k=1}^{K} \sum_{m=1}^{M} p_{hk,m}^I \log p_{hk,m}^I - \sum_{h=1}^{H} \sum_{k=1}^{K} \sum_{m=1}^{M} p_{hk,m}^O \log p_{hk,m}^O - \sum_{h=1}^{H} \sum_{m=1}^{M} q_{h,m}^I \log q_{h,m}^I \\ & - \sum_{h=1}^{H} \sum_{m=1}^{M} q_{h,m}^O \log q_{h,m}^O - \sum_{i=1}^{N} \sum_{m=1}^{M} w_{im} \log w_{im} \end{aligned} \tag{8}$$

subject to Eq. (7) and additional contrarians

$$\sum_{m=1}^{M} z_{h,m} p_{h,m}^O = 1, \tag{9}$$

$$\sum_{m=1}^{M} z_{hk,m} p_{hk,m}^I = 1, \tag{10}$$

$$\sum_{m=1}^{M} r_{h,m} q_{h,m}^{I} = 1, \tag{11}$$

$$\sum_{m=1}^{M} r_{h,m} q_{h,m}^{O} = 1, \tag{12}$$

$$\sum_{m=1}^{M} v_i w_{im} = 1. \tag{13}$$

Then, the Largrangian function is

$$\begin{aligned}
\mathbf{L} = & -\mathbf{H}(\mathbf{p}^{I}) - \mathbf{H}(\mathbf{p}^{O}) - \mathbf{H}(\mathbf{q}^{I}) - \mathbf{H}(\mathbf{q}^{O}) - \mathbf{H}(\mathbf{w}) \\
& + \lambda' \left[y_i - \sum_{h=1}^{H} \left\{ \left(\sum_{m=1}^{M} z_{h,m} p_{h,m}^{O} \right) f^{I} \left(\sum_{k=1}^{K} \sum_{m=1}^{M} z_{hk,m} p_{hk,m}^{I} x_{i,k} + \sum_{m=1}^{M} r_{h,m} q_{h,m}^{I} \right) + \sum_{m=1}^{M} r_{h,m} q_{h,m}^{O} \right\} - \sum_{m=1}^{M} v_i w_{im} \right] \\
& + \rho \left[1 - \sum_{m=1}^{M} z_{h,m} p_{h,m}^{O} \right] + \Phi \left[1 - \sum_{m=1}^{M} z_{hk,m} p_{hk,m}^{I} \right] + \phi \left[1 - \sum_{m=1}^{M} r_{h,m} q_{h,m}^{I} \right] + \vartheta \left[1 - \sum_{m=1}^{M} r_{h,m} q_{h,m}^{O} \right] \\
& + \varphi \left[1 - \sum_{m=1}^{M} v_i w_{im} \right]
\end{aligned} \tag{14}$$

The GME estimator generates the optimal probability vectors $\overset{\frown}{\mathbf{p}}^{I}$, $\overset{\frown}{\mathbf{p}}^{O}$, $\overset{\frown}{\mathbf{q}}^{I}$, $\overset{\frown}{\mathbf{q}}^{O}$ and $\overset{\frown}{\mathbf{w}}$ that can be used to calculate point estimates of the unknown weights, biases and the unknown random errors through the reparameterizations in Eqs. (4–5), respectively. As noted by Golan et al. (1996), since the Largrangian function function (Eq. 14) is strictly concave, I can rake the gradient of **L** to derive the first-order conditions. I would like to note that the number of supports M is less controversial; and usually used in the literature is in the range between 3 and 7 points since there is likely no significant improvement in the estimation with more points in the support.

3 Experiment Study

In this section, I present the Monte Carlo simulations to illustrate the performance of ANN with GME estimation. More precisely, the suggested estimation is compared with the ML, Bayesian, and BP algorithms. In the case of GME, I set the number of support as 3 ($M = 3$), whereas $\mathbf{z}_{k}^{O} = \mathbf{z}_{hk}^{I} = \mathbf{r}_{h,m} = [-5, 0, 5]$ and $\mathbf{v}_i = [-3(sd(y)), 0, 3(sd(y))]$. For ML, the normal likelihood function is assumed, while the Gaussian approximation for the joint posterior probability distribution of the network weights and biases is assumed for Bayesian estimation. In the experiment, the output variable is simulated from

$$y_i = 1 + 0.5(\sin(0.5x_i)) + \varepsilon_i, \tag{15}$$

where $\sin(\cdot)$ is the sinusoidal function. The simulated y_i becomes nonlinear and fluctuates overtime. Also, the precision of the estimations under different sample sizes and error distributions is to be investigated. Thus, I generated the error term from the normal and non-normal distributions, consisting of $N(0,\ 1)$, $t(0,\ 1,\ 4)$, and $Unif(-1,\ 1)$. Then, I generated a new sample during each Monte Carlo iteration by using Eq. (15) with the small sample sizes of 50 and 100 observations. The data are divided into training and test sets in which 70% of the total observations is used as the training data (in-sample data), while the rest is the test data (out-of-sample).

The simulation studies are carried out on a 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz, RAM 16 GB DDR5 workstation. The root mean square error (RMSE) is employed to report computation errors in all estimations. As there are several estimations considered and compared in this study, I set the same structure of ANN for all estimations. To be more specific, I set the learning rate $\eta = 0.001$, and the maximal error threshold 0.05. In addition, the single layer with sigmoid activation function is assumed, and the number of hidden neurons for those types of ANN models is set as 5. The above simulation process is repeated 100 times in order to estimate the mean value and standard deviation of RMSE (Table 1).

Table 1. Results of RMSE ($n = 50$)

In-sample	$\varepsilon_i \sim normal$			
	GME	BP	Bayesian	ML
Mean	0.814	0.668	0.670	0.660
SD	0.688	0.071	0.071	0.071
Out-of-Sample	$\varepsilon_i \sim normal$			
	GME	BP	Bayesian	ML
Mean	1.307	0.904	0.903	0.921
SD	1.123	0.166	0.170	0.165
In-sample	$\varepsilon_i \sim student - t$			
	GME	BP	Bayesian	ML
Mean	1.221	1.067	1.069	1.069
SD	1.273	0.115	0.116	0.115
Out-of-Sample	$\varepsilon_i \sim student - t$			
	GME	BP	Bayesian	ML
Mean	2.083	1.802	1.993	1.881
SD	2.114	0.693	0.893	0.701
In-sample	$\varepsilon_i \sim unif$			
	GME	BP	Bayesian	ML
Mean	2.117	2.652	2.745	2.784
SD	0.884	1.803	1.867	1.864
Out-of-Sample	$\varepsilon_i \sim unif$			
	GME	BP	Bayesian	ML
Mean	3.124	3.983	4.093	4.394
SD	1.093	2.343	2.431	2.993

Note: (1) The mean value of RMSE across 100 replications is reported, with the standard deviation in parentheses.

Table 2. Results of RMSE ($n = 100$)

In-sample	$\varepsilon_i \sim normal$			
	GME	BP	Bayesian	ML
Mean	0.743	0.535	0.573	0.544
SD	0.480	0.056	0.066	0.055
Out-of-Sample	$\varepsilon_i \sim normal$			
	GME	BP	Bayesian	ML
Mean	1.100	0.809	0.811	0.802
SD	1.023	0.123	0.136	0.126
In-sample	$\varepsilon_i \sim student - t$			
	GME	BP	Bayesian	ML
Mean	1.132	0.952	0.980	0.943
SD	1.341	0.327	0.207	0.321
Out-of-Sample	$\varepsilon_i \sim student - t$			
	GME	BP	Bayesian	ML
Mean	1.902	1.801	1.811	1.850
SD	2.493	0.955	0.907	0.939
In-sample	$\varepsilon_i \sim unif$			
	GME	BP	Bayesian	ML
Mean	2.334	5.685	5.693	5.383
SD	1.824	3.321	5.256	3.343
Out-of-Sample	$\varepsilon_i \sim unif$			
	GME	BP	Bayesian	ML
Mean	4.930	9.224	10.039	9.383
SD	2.549	4.003	5.023	5.034

Note: (1) The mean value of RMSE across 100 replications is reported, with the standard deviation in parentheses.

Reported in Tables 2, 3 are the mean and standard deviation of RMSE for in-sample goodness-of-fit and out-of-sample predictive accuracy across two horizons with three different error distributions. From these two tables, I can draw the following conclusions. (1) RMSE from the BP estimator is lower than that of the GME, ML, and Bayesian estimators, when the error of the ANN model is generated from normal and student-t distributions. The possible reason is that the small sample sizes of 50 and 100 may lead to a problem in the ML estimator as it relies on the asymptotic theory (Yamaka and Sriboonchitta 2020). Although the Bayesian estimation does not carry the assumptions of the asymptotic theory, which means that large sample size is not necessary for drawing valid statistical inferences, the conjugate prior for the weight parameter in this study may not be well-specified and thereby leading to the higher RMSE than BP and ML. (2) When the error is assumed to be uniformly distributed, the GME estimator outperforms BP, Bayesian, and ML, because the mean of RMSE of the former is smaller than the latter. (3) With regard to the standard deviation, it is observed that the standard deviation from GME is relatively high in all error distributions, except uniform. This indicates that the variance of the GME is relatively high when the error distribution is known. However, it is also interesting to see that the proposed GME is superior to other estimations both in goodness-of-fit and predictive accuracy over all sample sizes, when the uniform error distribution is given. Therefore, in the case that the distribution of the error is unknown, the GME is considered a useful method as there is no need to assume the theoretical probability distribution for the errors to make statistical inference.

Table 3. Computational time (second) with different sample sizes

Method	Observations			
	50	100	500	1000
GME	19.139	35.993	104.335	904.024
BP	0.105	0.194	0.460	0.841
Bayesian	0.786	0.842	0.903	0.661
ML	0.203	0.225	0.509	0.798

Finally, it is interesting to assess the computational cost of each estimation for small and large sample sizes $\{n = 50, 100, 500, 1000\}$. It can be observed in Table 3 that GME spends 19.139s to 0. 904.024s CPU time along 50 to 1000 observations. When comparing the computational performance between GME and other estimations, I found that GME runs slower than the others. This indicates that GME performs very poorly in the present simulations. This is not surprising due to the more parameters in the GME estimation. In other words, as the weight and bias parameters of ANN are derived from the expectation of probabilities on the prescribed supports, there will be a larger number of unknown parameters in the GME estimation. Although the GME takes high computational cost, it can provide more accurate prediction results particularly when the data is non-normally distributed.

4 Case Study

Boston Housing is a dataset obtained from the UCI Machine Learning Repository. There are 506 observations for predicting the price of houses in Boston. The data contained 14 variables, consisting of 13 continuous variables (per capita crime rate by town, proportion of non-retail business acres per town, proportion of residential land zoned for lots over 25,000 sq.ft., nitrogen oxides pollutant concentration, average number of rooms, proportion of owner-occupied units built prior to 1940, weighted distances to five Boston employment centers, index of accessibility to radial highways, property-tax rate, pupil-teacher ratio by town, the proportion of blacks, percent lower status of the population and median house value) and one discontinuous variable (Charles river dummy variable). In this study, I consider median house value as output, while the rest are inputs. In the simulations, 354 training data and 152 testing data were randomly generated from the Boston Housing database.

Table 4. Forecast performance on the Boston housing data set

Estimation		RMSE
GME	In-sample	1.909
	Out-of-sample	2.839
BP	In-sample	2.632
	Out-of-sample	4.014
Bayesian	In-sample	4.623
	Out-of-sample	4.872
ML	In-sample	3.834
	Out-of-sample	3.993

The performance of each estimator is reported in Table 4. Note that the structure of ANN is assumed to be the same for all cases. With this study's focus on the improvement of the ANN estimation, the ANN having three layers and three hidden neurons is used. It can be seen that the GME has the lowest error out of the estimators compared in this real data study. The performance of the GME evaluated over the out-of-sample dataset is illustrated Fig. 1. It is clearly seen that the predicted values obtained from the GME estimator are close to the out-of-sample data. This indicates the high performance of the GME in estimating the ANN model.

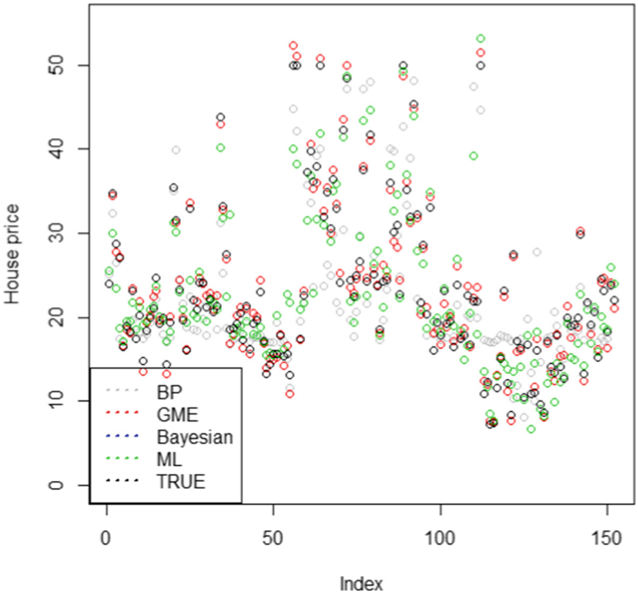

Fig. 1. Fitting to test data

5 Conclusion

In this study, the GME estimator is suggested to be applied to ANN for its having several interesting and significant features different from the traditional estimators, namely BP, Bayesian, and ML. The estimator is effective in terms of goodness-of-fit and predictive ability by reparametrizing the weight and bias parameters as the expectation of random variables on the prescribed supports under the derived distributions of the entropy maximization, which is confirmed by the Monte Carlo simulations and real data example in this study. Moreover, using this estimator enables the production of a novel method capable of learning complex behaviors without human intervention and the model can be fitted without making specific assumptions. Therefore, I hypothesized that estimation with GME (GMS-ANN) would enable the resulting parameter estimates to be more unbiased to data distributions and robust to over-fitting issues compared to those of BP, ML, and Bayesian.

In order to compare the performance of GME and other competing estimators, the ANN structures are always assigned the same number of hidden neurons for both simulation and empirical studies. The RMSE is used for performance comparison. The results show that GME estimator produces the lowest RMSE estimates compared with BP, ML, and Bayesian when the errors are generated from uniform distributions. In other words, when the error distribution is unknown, these experiment results confirm an advantage of the GME approach. However, considering the computational cost, GME performs very poorly in the present simulations for all sample sizes due to the large number of probability estimates. It should be noted that in order to obtain as good performance as possible for GME, long time effort is needed to find the appropriate probabilities for weight, bias, and error terms.

As the activation function in this study was assumed to be sigmoid, the performance of GME should be investigated considering other activation functions, such as exponential, ReLu and tanh. I leave this issue in the further study. Also, as the number of support and the value of bound can affect the estimation results, I would suggest varying the number and value of support bounds to validate the performance of GME in estimating ANN models.

References

Alibrandi, U., Mosalam, K.M.: Kernel density maximum entropy method with generalized moments for evaluating probability distributions, including tails, from a small sample of data. Int. J. Numer. Meth. Eng. **113**(13), 1904–1928 (2018)

Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)

Chon, K.H., Cohen, R.J.: Linear and nonlinear ARMA model parameter estimation using an artificial neural network. IEEE Trans. Biomed. Eng. **44**(3), 168–174 (1997)

Chen, S., Mao, J., Chen, F., Hou, P., Li, Y.: Development of ANN model for depth prediction of vertical ground heat exchanger. Int. J. Heat Mass Transf. **117**, 617–626 (2018)

Chen, B., Zhu, Y., Hu, J., Principe, J.C.: System Parameter Identification: Information Criteria and Algorithms. Newnes (2013)

Chu, J., Liu, X., Zhang, Z., Zhang, Y., He, M.: A novel method overcomeing overfitting of artificial neural network for accurate prediction: application on thermophysical property of natural gas. Case Stud. Therm. Eng. **28**, 101406 (2021)

Dorling, S.R., Foxall, R.J., Mandic, D.P., Cawley, G.C.: Maximum likelihood cost functions for neural network models of air quality data. Atmos. Environ. **37**(24), 3435–3443 (2003)

Fu, L., Hsu, H.H., Principe, J.C.: Incremental backpropagation learning networks. IEEE Trans. Neural Netw. **7**(3), 757–761 (1996)

Gish, H.: Maximum likelihood training of neural networks. In: Artificial Intelligence Frontiers in Statistics, pp. 241–255. Chapman and Hall/CRC (2020)

Golan, A., Judge, G., Miller, D.: Maximum Entropy Econometrics: Robust Estimation with Limited Data. Wiley, Chichester (1996)

Jaynes, E.T.: On the rationale of maximum-entropy methods. Proc. IEEE **70**(9), 939–952 (1982)

Kocadağlı, O., Aşıkgil, B.: Nonlinear time series forecasting with Bayesian neural networks. Expert Syst. Appl. **41**(15), 6596–6610 (2014)

Kocadağlı, O.: A novel hybrid learning algorithm for full Bayesian approach of artificial neural networks. Appl. Soft Comput. **35**, 52–65 (2015)

Lin, P., Fu, S.W., Wang, S.S., Lai, Y.H., Tsao, Y.: Maximum entropy learning with deep belief networks. Entropy **18**(7), 251 (2016)

Maneejuk, P., Yamaka, W., Sriboonchitta, S.: Entropy inference in smooth transition kink regression. Commun. Stat.-Simul. Comput. 1–24 (2020)

Müller, P., Insua, D.R.: Issues in Bayesian analysis of neural network models. Neural Comput. **10**(3), 749–770 (1998)

Pukelsheim, F.: The three sigma rule. Am. Stat. **48**(2), 88–91 (1994)

Ramos, V., Yamaka, W., Alorda, B., Sriboonchitta, S.: High-frequency forecasting from mobile devices' bigdata: an application to tourism destinations' crowdedness. Int. J. Contemp. Hosp. Manag. (2021)

Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(3), 379–423 (1948)

Wang, X., Du, J., Wang, Y.: A maximum likelihood approach to deep neural network based speech dereverberation. In: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 155–158. IEEE (2017)

White, H.: Some asymptotic results for learning in single hidden-layer feedforward network models. J. Am. Stat. Assoc. **84**(408), 1003–1013 (1989)

Yamaka, W., Phadkantha, R., Maneejuk, P.: A convex combination approach for artificial neural network of interval data. Appl. Sci. **11**(9), 3997 (2021)

Yamaka, W., Sriboonchitta, S.: Forecasting using information and entropy based on belief functions. Complexity **2020** (2020)

Yang, Z., Baraldi, P., Zio, E.: A comparison between extreme learning machine and artificial neural network for remaining useful life prediction. In: 2016 Prognostics and System Health Management Conference (PHM-Chengdu), pp. 1–7. IEEE (2016)