

Communication

A probability density function generator based on deep learning

Chi-Hua Chen¹, Fangying Song^{1,*}, Feng-Jang Hwang² and Ling Wu¹

¹ College of Mathematics and Computer Science, Fuzhou University, China; chihua0826@fzu.edu.tw; fangying_song@fzu.edu.cn; wuling1985@fzu.edu.cn

² School of Mathematical and Physical Sciences, University of Technology Sydney, Australia; Feng-Jang.Hwang@uts.edu.au

* Correspondence: fangying_song@fzu.edu.cn; Tel.: +86-19959346509

Abstract: To generate a probability density function (PDF) for fitting probability distributions of real data, this study proposes a deep learning method which consists of two stages: (1) a training stage for estimating the cumulative distribution function (CDF) and (2) a performing stage for predicting the corresponding PDF. The CDFs of common probability distributions can be adopted as activation functions in the hidden layers of the proposed deep learning model for learning actual cumulative probabilities, and the differential equation of trained deep learning model can be used to estimate the PDF. To evaluate the proposed method, numerical experiments with single and mixed distributions are performed. The experimental results show that the values of both CDF and PDF can be precisely estimated by the proposed method.

Keywords: probability density function; cumulative distribution function; deep learning

1. Introduction

To explain trends or phenomena, statistic tools and probability models are usually applied to analyse real data. Various common probability distributions (e.g. exponential distribution (ED), normal distribution (ND), log-normal distribution (LD), gamma distribution (GD), etc.) are used as the assumptions about the distribution of practical data [1]. The CDF and PDF of real data, however, could be complex and irregular, so serious errors would be incurred by the naive assumptions. This study proposes a deep learning method [2] that develops a neural network to learn the CDF of real data and fit the corresponding PDF.

2. Method

The proposed deep learning method is comprised of two stages: (1) a training stage for estimating the CDF and (2) a performing stage for predicting the corresponding PDF (shown in Figure 1).

In the training stage, practical data can be collected and analysed for generating actual cumulative probabilities. A deep learning model retaining multiple layers (i.e. an input layer, hidden layers and an output layer) is constructed to learn a CDF in accordance with the actual cumulative probabilities (shown in Figure 2). The input layer includes the random variable x , the actual CDF of which is denoted by F , and the output layer contains the estimated CDF of x , denoted by \tilde{F} . The CDFs of common probability distributions can be adopted as the activation functions in the hidden layers. For example, the CDFs of ED, ND, LD, and GD [1] (denoted by f_1 , f_2 , f_3 , and f_4 , respectively) can be used as the activation functions (shown respectively in Eq. (1), (2), (3), and (4), where $\tilde{\lambda}$, $\tilde{\mu}$, \tilde{s}_1 , $\tilde{\mu}_2$, \tilde{s}_2 , $\tilde{\alpha}$, and $\tilde{\beta}$ are all parameters). In Eq. (4), $\gamma(\tilde{\alpha}, \tilde{\beta}, x)$ is the lower incomplete gamma

function, and $\Gamma(\tilde{\alpha})$ is the gamma function. Furthermore, the number of neurons with different activation functions can be extended to n , and the CDF can be estimated by Eq. (5) in accordance with a weight set (i.e. $\{\tilde{w}_1, \tilde{w}_2, \dots, \tilde{w}_n\}$). The loss function of the proposed deep learning model is defined by Eq. (6) according to the estimated CDF (i.e. \tilde{F}) and the actual CDF (i.e. F) for minimising the estimation error. The gradient descent method is applied to the setting of parameters. The partial differential of E with respect to each parameter is calculated (Eq. (7)-(14)), and the parameter in the $(k+1)$ -th iteration can be updated in accordance with the aforementioned partial differential in the k -th iteration (Eq. (15)-(22), where η is the learning rate). In Eq. (13), $\psi(\cdot)$ is the digamma function. The number of hidden layers can be increased to estimate a relatively complex CDF.

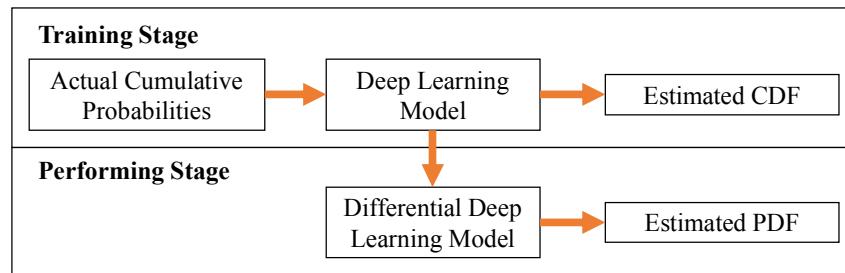


Figure 1. The proposed deep learning method for estimating practical probability distributions.

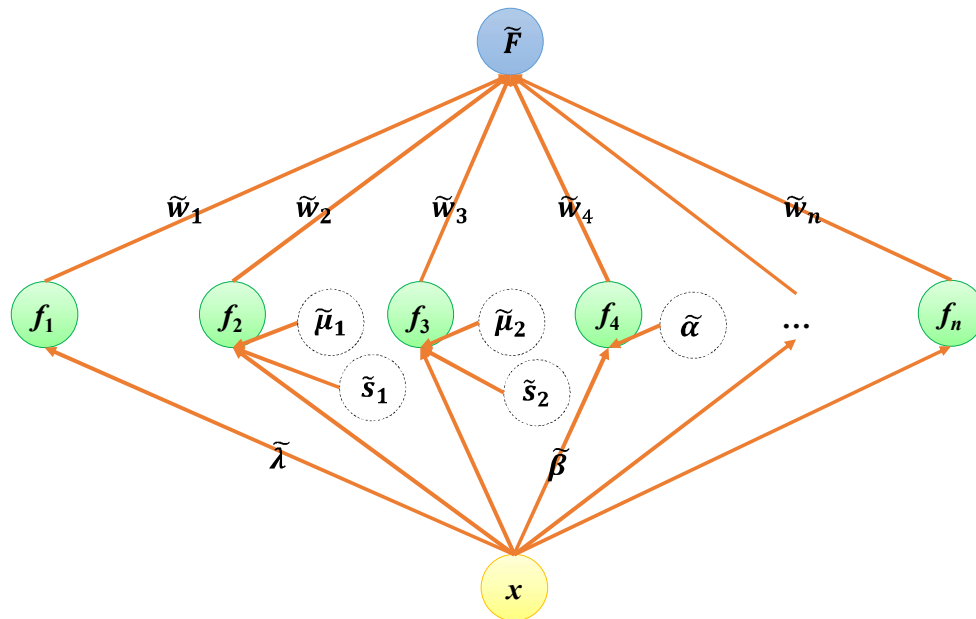


Figure 2. The structure of the proposed deep learning model.

$$f_1 = 1 - e^{-\tilde{\lambda}x} \quad (1)$$

$$f_2 = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{x - \tilde{\mu}_1}{\tilde{s}_1 \sqrt{2}} \right) \right] \quad (2)$$

$$f_3 = \frac{1}{2} \left[1 + \operatorname{erf} \left(\frac{\ln(x) - \tilde{\mu}_2}{\tilde{s}_2 \sqrt{2}} \right) \right] \quad (3)$$

$$f_4 = \frac{\gamma(\tilde{\alpha}, \tilde{\beta}, x)}{\Gamma(\tilde{\alpha})} \quad (4)$$

$$\tilde{F} = \frac{\sum_{i=1}^n \tilde{w}_i f_i}{\sum_{i=1}^n \tilde{w}_i} \quad (5)$$

$$E = \frac{1}{2}(\tilde{F} - F)^2 = \frac{1}{2}\sigma^2 \quad (6)$$

$$\frac{\partial E}{\partial \tilde{w}_j} = \frac{\partial E}{\partial \sigma} \frac{\partial \sigma}{\partial \tilde{F}} \frac{\partial \tilde{F}}{\partial \tilde{w}_j} = \sigma \times \frac{\sum_{i=1}^n [\tilde{w}_i (f_j - f_i)]}{\left(\sum_{i=1}^n \tilde{w}_i\right)^2} \quad (7)$$

$$\frac{\partial E}{\partial \tilde{\lambda}} = \frac{\partial E}{\partial \sigma} \frac{\partial \sigma}{\partial \tilde{F}} \frac{\partial \tilde{F}}{\partial f_1} \frac{\partial f_1}{\partial \tilde{\lambda}} = \sigma \times \frac{\tilde{w}_1}{\sum_{i=1}^n \tilde{w}_i} \times \left(x e^{-\tilde{\lambda} x}\right) \quad (8)$$

$$\frac{\partial E}{\partial \tilde{\mu}_1} = \frac{\partial E}{\partial \sigma} \frac{\partial \sigma}{\partial \tilde{F}} \frac{\partial \tilde{F}}{\partial f_2} \frac{\partial f_2}{\partial \tilde{\mu}_1} = \sigma \times \frac{\tilde{w}_2}{\sum_{i=1}^n \tilde{w}_i} \times \left(\frac{-1}{\sqrt{2\pi \tilde{s}_1^2}} e^{-\frac{(x - \tilde{\mu}_1)^2}{2\tilde{s}_1^2}} \right) \quad (9)$$

$$\frac{\partial E}{\partial \tilde{s}_1} = \frac{\partial E}{\partial \sigma} \frac{\partial \sigma}{\partial \tilde{F}} \frac{\partial \tilde{F}}{\partial f_2} \frac{\partial f_2}{\partial \tilde{s}_1} = \sigma \times \frac{\tilde{w}_2}{\sum_{i=1}^n \tilde{w}_i} \times \left(\frac{-(x - \tilde{\mu}_1)}{\sqrt{2\pi \tilde{s}_1^2}} e^{-\frac{(x - \tilde{\mu}_1)^2}{2\tilde{s}_1^2}} \right) \quad (10)$$

$$\frac{\partial E}{\partial \tilde{\mu}_2} = \frac{\partial E}{\partial \sigma} \frac{\partial \sigma}{\partial \tilde{F}} \frac{\partial \tilde{F}}{\partial f_3} \frac{\partial f_3}{\partial \tilde{\mu}_2} = \sigma \times \frac{\tilde{w}_3}{\sum_{i=1}^n \tilde{w}_i} \times \left(\frac{-1}{\sqrt{2\pi \tilde{s}_2^2}} e^{-\frac{(\ln(x) - \tilde{\mu}_2)^2}{2\tilde{s}_2^2}} \right) \quad (11)$$

$$\frac{\partial E}{\partial \tilde{s}_2} = \frac{\partial E}{\partial \sigma} \frac{\partial \sigma}{\partial \tilde{F}} \frac{\partial \tilde{F}}{\partial f_3} \frac{\partial f_3}{\partial \tilde{s}_2} = \sigma \times \frac{\tilde{w}_3}{\sum_{i=1}^n \tilde{w}_i} \times \left(\frac{-(\ln(x) - \tilde{\mu}_2)}{\sqrt{2\pi \tilde{s}_2^2}} e^{-\frac{(\ln(x) - \tilde{\mu}_2)^2}{2\tilde{s}_2^2}} \right) \quad (12)$$

$$\frac{\partial E}{\partial \tilde{\alpha}} = \frac{\partial E}{\partial \sigma} \frac{\partial \sigma}{\partial \tilde{F}} \frac{\partial \tilde{F}}{\partial f_4} \frac{\partial f_4}{\partial \tilde{\alpha}}$$

$$= \sigma \times \frac{\tilde{w}_4}{\sum_{i=1}^n \tilde{w}_i} \times \left[\ln \left(\frac{x}{\tilde{\beta}} \right) \left(\frac{x}{\tilde{\beta}} \right)^{\tilde{\alpha}} e^{-\frac{x}{\tilde{\beta}}} \sum_{l=0}^{\infty} \frac{\left(\frac{x}{\tilde{\beta}} \right)^l}{\Gamma(\tilde{\alpha}+l+1)} - \left(\frac{x}{\tilde{\beta}} \right)^{\tilde{\alpha}} e^{-\frac{x}{\tilde{\beta}}} \sum_{l=0}^{\infty} \frac{\left(\frac{x}{\tilde{\beta}} \right)^l \psi(\tilde{\alpha}+l+1)}{\Gamma(\tilde{\alpha}+l+1)} \right] \quad (13)$$

$$\frac{\partial E}{\partial \tilde{\beta}} = \frac{\partial E}{\partial \sigma} \frac{\partial \sigma}{\partial \tilde{F}} \frac{\partial \tilde{F}}{\partial f_4} \frac{\partial f_4}{\partial \tilde{\beta}} = \sigma \times \frac{\tilde{w}_4}{\sum_{i=1}^n \tilde{w}_i} \times \left[\frac{1}{\tilde{\beta} \Gamma(\tilde{\alpha})} \left(\frac{x}{\tilde{\beta}} \right)^{\tilde{\alpha}} e^{-\frac{x}{\tilde{\beta}}} \right] \quad (14)$$

$$\tilde{w}_j^{(k+1)} = \tilde{w}_j^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \tilde{w}_j^{(k)}} \quad (15)$$

$$\tilde{\lambda}^{(k+1)} = \tilde{\lambda}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \tilde{\lambda}^{(k)}} \quad (16)$$

$$\tilde{\mu}_1^{(k+1)} = \tilde{\mu}_1^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \tilde{\mu}_1^{(k)}} \quad (17)$$

$$\tilde{s}_1^{(k+1)} = \tilde{s}_1^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \tilde{s}_1^{(k)}} \quad (18)$$

$$\tilde{\mu}_2^{(k+1)} = \tilde{\mu}_2^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \tilde{\mu}_2^{(k)}} \quad (19)$$

$$\tilde{s}_2^{(k+1)} = \tilde{s}_2^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \tilde{s}_2^{(k)}} \quad (20)$$

$$\tilde{\alpha}^{(k+1)} = \tilde{\alpha}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \tilde{\alpha}^{(k)}} \quad (21)$$

$$\tilde{\beta}^{(k+1)} = \tilde{\beta}^{(k)} - \eta \frac{\partial E^{(k)}}{\partial \tilde{\beta}^{(k)}} \quad (22)$$

57 In the performing stage, the differential equation of trained deep learning model described by
 58 Equation (23) is derived to obtain the estimated PDF (i.e. \tilde{P}).

$$\begin{aligned}
\tilde{P} &= \frac{\partial \tilde{F}}{\partial x} = \sum_{i=1}^n \left(\tilde{w}_i \times \frac{\partial f_i}{\partial x} \right) \\
&= \tilde{w}_1 \times \tilde{\lambda} e^{-\tilde{\lambda}x} + \tilde{w}_2 \times \frac{1}{\sqrt{2\pi\tilde{s}_1^2}} e^{-\frac{(x-\tilde{\mu}_1)^2}{2\tilde{s}_1^2}} + \\
&\quad \tilde{w}_3 \times \frac{1}{x\tilde{s}_2\sqrt{2\pi}} e^{-\frac{(\ln(x)-\tilde{\mu}_2)^2}{2\tilde{s}_2^2}} + \tilde{w}_4 \times \frac{x^{\tilde{\alpha}-1} e^{-\frac{x}{\tilde{\beta}}}}{\tilde{\beta}^{\tilde{\alpha}} \Gamma(\tilde{\alpha})} + \sum_{i=5}^n \left(\tilde{w}_i \times \frac{\partial f_i}{\partial x} \right)
\end{aligned} \tag{23}$$

3. Experimental Results and Discussions

In the numerical experiments, single and mixed common probability distributions with the number of records denoted by m were designed, and the proposed method was evaluated in terms of the mean absolute percentage errors (MAPEs). The MAPE of the estimated CDF (i.e. M_F) is defined as Eq. (24) according to \tilde{F} and F , and the MAPE of the estimated PDF (i.e. M_P) is defined as Eq. (25) according to \tilde{P} and P .

$$M_F = \sum_{k=1}^m \frac{|F_k - \tilde{F}_k|}{F_k} \times \frac{100\%}{m} \tag{24}$$

$$M_P = \sum_{k=1}^m \frac{|P_k - \tilde{P}_k|}{P_k} \times \frac{100\%}{m} \tag{25}$$

Six test cases, including single ED, single ND, single LD, single GD, and two mixed distributions (MDs) of single ND and single GD with different parameters, were set in the experiments, where the proposed deep learning model for comparison was constructed with f_1, f_2, f_3 , and f_4 . The simulated experimental results are shown in Tables 1, 2, 3, and 4.

Table 1. MAPEs of the estimated CDF with single distributions.

m	Case 1: ED ($\lambda=0.5$)	Case 2: ND ($\sigma=1, \mu=0.5$)	Case 3: LD ($\sigma=1, \mu=0.5$)	Case 4: GD ($\alpha=0.1, \beta=0.5$)
2	0%	0%	0%	1.29%
5	0%	0%	0%	0.20%
40	0%	0%	0%	0%

Table 2. MAPEs of the estimated PDF with single distributions.

m	Case 1: ED ($\lambda=0.5$)	Case 2: ND ($\sigma=1, \mu=0.5$)	Case 3: LD ($\sigma=1, \mu=0.5$)	Case 4: GD ($\alpha=0.1, \beta=0.5$)
2	0%	0%	0%	9.30%
5	0%	0%	0%	1.58%
40	0%	0%	0%	0.01%

Table 3. MAPEs of the estimated CDF with mixed distributions.

m	Case 5: ND + GD ($\sigma=1, \mu=0.5, \alpha=0.1, \beta=0.5$)	Case 6: ND + GD ($\sigma=1, \mu=0.5, \alpha=2, \beta=10$)
2	5.12%	22.77%
5	2.20%	6.60%
40	0.01%	0.01%

Table 4. MAPEs of the estimated PDF with mixed distributions.

m	Case 5: ND + GD ($\sigma=1, \mu=0.5, \alpha=0.1, \beta=0.5$)	Case 6: ND + GD ($\sigma=1, \mu=0.5, \alpha=2, \beta=10$)
2	31.41%	16.70%
5	16.04%	19.10%
40	1.39%	0.09%

In Case 1, a single ED with the parameter $\lambda=0.5$ was used as the benchmark, and different numbers of records (i.e. $m=2, 5$, and 40) were considered to analyse the MAPEs of the estimated CDF and PDF. Both M_F and M_P are 0%, so the proposed method can obtain a precise estimated PDF for Case 1. In Cases 2 and 3, a single ND with $(\sigma, \mu)=(1, 0.5)$ and a single LD with the same parameters were generated, respectively. The PDF was precisely estimated again by the proposed method for both cases. In Case 4, a single GD with $(\alpha, \beta)=(1, 0.5)$ was set to train and test the proposed deep learning model. The experimental results show that the estimated CDF and PDF with MAPEs no more than 0.01% can be obtained for $m=40$ even though the GD is a relatively complex probability distribution.

As for the analyses of MDs, Cases 5 and 6 were designed to combine single ND and single GD, and the MAPEs of the corresponding estimated CDFs and PDFs were provided in Tables 3 and 4. In Case 5, the parameters $(\sigma, \mu, \alpha, \beta)=(1, 0.5, 0.1, 0.5)$ were adopted to generate an MD. The MAPE of the estimated PDF can be less than 1.4% when $m=40$. In Case 6, a relatively complex MD was generated with the parameters $(\sigma, \mu, \alpha, \beta)=(1, 0.5, 2, 10)$. The proposed method can precisely estimate the CDF and PDF with MAPEs no more than 0.01% and 0.09%, respectively, for $m=40$.

4. Conclusion and Future Work

This study proposes a deep learning method to learn the potentially complex and irregular probability distributions, and the differential equation of trained deep learning model can be used to estimate the corresponding PDF. The experimental results demonstrate that the proposed method can precisely estimate the CDFs and PDFs for single and mixed probability distributions. Future research could be conducted by assessing the proposed model with real data. Moreover, the accurate CDF and PDF of real-data probability distribution can be produced to improve further analyses with game theory or queueing theory.

Author Contributions: conceptualization, Chi-Hua Chen and Fangying Song; methodology, Chi-Hua Chen and Fangying Song, Feng-Jang Hwang; software, Chi-Hua Chen and Fangying Song; validation, Chi-Hua Chen and Fangying Song; formal analysis, Chi-Hua Chen and Fangying Song; investigation, Chi-Hua Chen and Fangying Song; resources, Chi-Hua Chen and Fangying Song; data curation, Chi-Hua Chen and Fangying Song; writing—original draft preparation, Chi-Hua Chen and Fangying Song; writing—review and editing, Fangying Song and Ling Wu.

Conflicts of Interest: The authors declare no conflict of interest.

108 **References**

- 109 1. King, M.: 'Probability density function' in King, M. (Ed.): 'Statistics for process control engineers: a practical
110 approach', John Wiley & Sons Ltd., New York, 2017.
- 111 2. LeCun, Y., Bengio, Y. Hinton, G.E.: 'Deep Learning', *Nature*, 2015, 521, pp. 436-444, doi: 10.1038/nature14539