Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2023) 251-264

ECCOMAS
Proceedia

UNCECOMP 2023
5th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Science and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Athens, Greece, 12-14 June 2023

PREDICTION UNCERTAINTY FLATTENING FOR PERFORMANCE IMPROVEMENT OF DEEP LEARNING

Chao Liu, Xinlei Zhou, Xizhao Wang*

College of Computer Science and Software Engineering
Shenzhen University
3688 Nanhai Avenue, Nanshan District, Shenzhen, Guangdong Province, China
(*) Corresponding Author. E-mail: xizhaowang@ieee.org

Abstract. *The representation, measure, and handling of uncertainty in an application of deep learning have a significant impact on the performance of learning systems. Uncertainty is embedded in the entire process of learning from data, which can roughly be categorized as three types, i.e., data uncertainty, model uncertainty, and prediction uncertainty. Appropriately modeling and processing uncertainties in different phases of learning can significantly improve accuracy and robustness. Focusing on the 3rd type of uncertainty, i.e., the prediction uncertainty which lies on the phase from the model output (i.e., a probability distribution on label space) to the final class(s) determination, we present in this paper a new viewpoint regarding uncertainty processing. Theoretically and experimentally, it establishes a statistical relationship between the prediction uncertainty and the learning performance, which implicitly points out that some results about uncertainty processing in traditional textbooks may be incorrect. Based on this viewpoint, we propose an approach to optimize the output layer of neural network models. The method improves the prediction accuracy by replacing the last layer of softmax with a random weight model which is a simple and fast non-iterative training algorithm. Extensive experiments and simulations on the synthetic and real datasets demonstrate the effectiveness and good generalization of the proposed scheme.*

Keywords: Deep learning, prediction uncertainty, random weight network, generalization

ISSN:2623-3339
Peer-review under responsibility of the organizing committee of UNCECOMP 2023.
doi: 10.7712/120223.10336.19750

1 INTRODUCTION

The concept of uncertainty has been widely applied in various branches and disciplines of the natural sciences. For a long time, uncertainty has played a significant role in fields such as economics, psychology, and the social sciences. Similarly, computer scientists recognized the importance of uncertainty for the study of artificial intelligence systems at an early stage, especially with the appearance of expert systems. Inconsistency, consistency, incompleteness, imprecision, and fuzziness in knowledge representation are manifestations of uncertainty [1]. Decision theorist Douglas W. Hubbard claims that "uncertainty" is used to express the fact that we do not have enough knowledge to describe the current situation or estimate future results [2]. In the learning paradigm, current mainstream machine intelligence methods are based on partial sampling that cannot accurately represent the entire dataset. In addition, unseen data is predicted based on models with cognitive biases. According to Douglas, there is uncertainty in the predicted results no matter which algorithms or models are used to fit the distribution of the whole dataset with partial sampling. Except that, complex data can be represented in various forms, and the dimensions of features and categories increase dramatically. These phenomena present many challenges, such as data noise, data missing, the long-tailed distribution, and the number of hyperparameters or the solution space of models are huge. All these problems further increase the uncertainty in the machine learning modeling process and seriously affect the effectiveness and reliability of traditional learning algorithms [3].

Currently, there is no general mathematical definition of uncertainty, and there is no universal formula applicable to all situations. Uncertainty is typically considered in a specific context when modeling [3]. The current mainstream view holds that there are two types of uncertainty in intelligent systems under the learning paradigm, i,e., aleatoric uncertainty and epistemic uncertainty [4]. Among them, epistemic uncertainty (also known as model uncertainty) describes the cognitive state of the system for the current existing information, reflecting the ignorance of the correct prediction value due to the unknown of the correct model [5]. Data uncertainty [6] (also known as accidental uncertainty and stochastic uncertainty) describes the randomness originating from data collection and generation, which is an inherent property of data rather than a model property, so it is inevitable. In classification tasks, the category label of a sample is normally output as a probability vector due to the discontinuity of the target value. In the output vector, each element indicates the possibility of the sample belonging to each class. Therefore, another type of uncertainty should be categorized at the stage from model output to final category determination, namely prediction uncertainty. Therefore, on the basis of mainstream views, we expand the uncertainty involved in the machine learning modeling process into three categories, including data uncertainty, model uncertainty, and prediction uncertainty. Prediction uncertainty can be quantified by entropy[7], ambiguity [8], unspecified[9] and other methods. In addition, Li Pingke et al. [10] proposed an uncertainty measurement model for fuzzy variables under the framework of credibility theory. Bronevich et al. [11] proposed an axiomatic method for measuring imprecise probability. Couso et al. [12] proposed an upper and lower probability model that can handle random variables with uncertainty information. In addition, to deal with inconsistency in decision-making, many scholars[13, 14] have proposed a series of rough set models and algorithms to measure, represent and model the uncertainty caused by data inconsistency. Wang et al. [15] researched the uncertainty of rough sets under different knowledge granularity. Zeng et al.[13] proposed a dynamic update fuzzy rough approximation method for mixed data in the case of attribute value changes. Chen et al. [14] discussed coverage rough set metrics based on granularity and evidence theory. However, there

are few studies relevant to the mechanism of the relationship between prediction uncertainty and learning performance.

Therefore, this article focuses on classification problems, discussing the ability of predictive uncertainty in optimizing the performance of deep learning models. Specifically, this article establishes a statistical relationship between predictive uncertainty and learning performance, indicating the potential of maximizing uncertainty in optimizing model performance. It is verified experimentally that models with greater uncertainty have stronger generalization performance within a certain range. This implicitly points out that some of the results in traditional textbooks on the treatment of uncertainty may be incorrect. Inspired by this, we propose a method to optimize the output layer of the neural network. Our method introduces a simple and easy-to-use model, the extreme learning machine, into the deep neural network. In this way, the output vector of the network is calculated using a generalized inverse matrix. In other words, the predicted probability distribution of the softmax activation function in the last layer is replaced by the analytical solution of the extreme learning machine. The method achieves a significant improvement in testing accuracy. The main contributions of this article can be summarized as follows:

- Summarizes the three types of uncertainty involved in the machine learning modeling process, namely data uncertainty, model uncertainty, and prediction uncertainty from the model output (probability distribution of the label set) to the final class determination.
- This paper presents a novel perspective on uncertainty processing. Within a certain range, the larger model with prediction uncertainty has a stronger generalization performance.
- The extreme learning machine is introduced to replace the output layer with the softmax activation function. The effectiveness of this method is confirmed experimentally.

2 THREE TYPES OF UNCERTAINTY IN A LEARNING PROCESS

As shown in Fig.1, uncertainty is embedded in the entire process of learning from data. Modeling and processing uncertainties appropriately during different phases of learning can significantly improve accuracy and robustness. According to the characteristics of machine learning, this article categorizes the uncertainties involved in the modeling process into data uncertainty, model uncertainty, and prediction uncertainty. Data uncertainty describes the randomness inherent in the data generation process. Model uncertainty reflects the cognitive state of the system and is not a potential stochastic phenomenon. In addition, this article will focus on the third type of uncertainty, i.e., the prediction uncertainty which lies in the phase from the model output to the final class(s) determination. The following content will provide a detailed introduction to these three types of uncertainty and the corresponding quantification methods.

Data uncertainty describes the randomness inherent in the data generation process. This kind of randomness is caused by unavoidable and unpredictable changes during data collection, and it has an impact on the data. In this case, data uncertainty cannot be eliminated by collecting more data [16]. As an example, air turbulence and the accuracy of an instrument can unavoidably bias results when measuring gravitational acceleration. Data uncertainty is a kind of typical uncertainty in machine learning because almost all input-output pairs collected from the real world contain noise [17]. Taking observation noise that disrupts the target value $f^*(x)$ as an example, data uncertainty can be expressed as:

$$d(x) := f(x) - f^*(x) \tag{1}$$

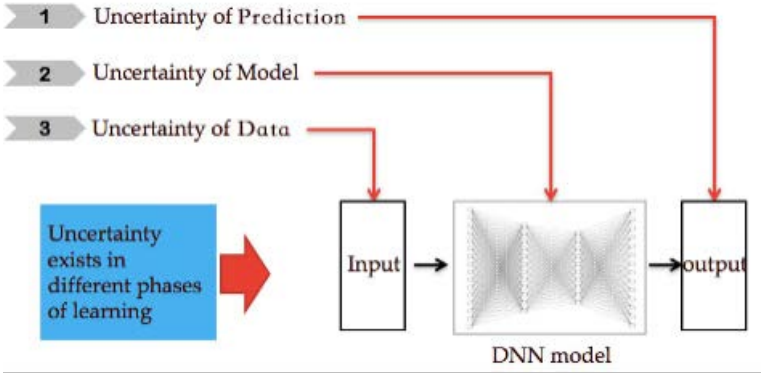

Figure 1: Three types of uncertainty in a process of learning from data.

where, $f(x)$ is the observed target value. Assuming the observed noise $d(x)$ follows a normal distribution $N(0, \sigma^2)$, data uncertainty can be quantified by capturing σ^2. Nix et al. [17] proposed the classical method for quantifying target noise based on maximum likelihood estimation. In this method, data uncertainty is captured via a neural network structure. The noise in the input data is obtained by adding a computation unit $\sigma^2(\boldsymbol{x})$ to the output layer on the basis of the target value prediction node $\hat{y}$. Specifically, assuming that there is non-negligible noise in the observed data, the loss function is approximated using maximum likelihood estimation for the target value $f^*(x)$ as follows:

$$L \approx \sum_i \frac{1}{2}\left(\frac{[d_i - \hat{y}(x_i)]^2}{\sigma^2(x_i)} + ln[(\sigma^2(x_i))]\right) \tag{2}$$

Due to the data noise estimation unit $\sigma^2(x)$ being non-negligible, the second term in the final loss function C cannot be ignored. We can see that if we neglect the noise contained in the target value, i.e., $\sigma^2(x)$ is constant, the above loss function degenerates into mean squared error loss.

Model uncertainty describes the cognitive state of the system, which is influenced by factors such as the scale of data sampling, model selection, optimization strategy, and hyperparameter settings. Compared with data uncertainty, it is not a potential stochastic phenomenon [16]. This uncertainty reflects the confidence level of the model in making correct predictions and can be reduced by measures such as collecting missing sample information. For regression problems, model uncertainty can be represented by the variance of predicted target values [6]:

$$m(x) := \sigma^2 + \frac{1}{T}\sum_{t=1}^{T}(f^{\widehat{\omega_t}}(x) - E(y))^2 \tag{3}$$

where $E(y) = \frac{1}{T}\sum_{t=1}^{T} f^{\widehat{\omega_t}}(x)$ is the expected value of the predicted results repeated T times, and $f^{\widehat{\omega_t}}(x)$ represents the predicted value when the model parameters are $\widehat{\omega_t}$. In this case, $m(x)$ quantifies the model uncertainty caused by different parameters of models with the same structure. During testing, the variation of the model parameter $\widehat{\omega_t}$ produces a set of models. We use the variance of the outputs of this set of models to quantify this type of uncertainty.

In the process of modeling a classification model, the predicted values of categories are usually output in the form of probabilities. These probabilities indicate the possibility of a sample belonging to each category rather than accurately indicating the membership category.

We define this kind of prediction uncertainty as the unclear state of the sample category division, which is caused by the non-continuity of the classification target. There are lots of methods to quantify such uncertainty, and this paper will detail two methods: classification entropy and fuzziness. Given a classification dataset $D_{train} = (X, Y) \subset R^n \times 0, 1^C$ containing C categories, the predicted vector $\mu_i = (\mu_{i1}, \mu_{i2}, ..., \mu_{ij}, ..., \mu_{iC})$ output by the classifier represents the probability that the i-th sample belongs to each category, where $\mu_{ij}(1 \leq i \leq N, 1 \leq j \leq C) \in [0, 1]$ and satisfies $\sum_{j=1}^{C} \mu_{ij} = 1$. The classification entropy discusses the purity of the distribution of each subset S_j in the set S [3], which is defined as:

$$E_{ntro}(S) = -\sum(\frac{|S_j|}{|S|} \log \frac{|S_j|}{|S|}) \tag{4}$$

where $|\cdot|$ represents the number of samples in the set. Let $\mu_{ij} = \frac{|S_j|}{|S|}$, then the output uncertainty can be quantified using classification entropy as follows:

$$E_{ntro}(\mu_i) = -\sum_{j=1}^{C}(\mu_{ij} \log \mu_{ij}) \tag{5}$$

The above formula shows that when the probability values of each category are equal, the classification entropy of the predicted result reaches the maximum, which means the uncertainty of the output is maximum. When the predicted probability $\mu_{ij^*} = 1$ in a single category j^*, the purity of the sample category distribution reaches the maximum, the classification entropy of the predicted result is minimized, and the uncertainty of the output is also minimized.

On the other hand, Zadeh established fuzzy set theory [18] based on classical set theory to provide mathematical methods for describing fuzzy objects. De Luca and Termini [8] proposed that fuzziness is an average intrinsic information of fuzzy sets, which can characterize the uncertainty related to the situation described by the fuzzy set, and defined a quantitative measurement method for measuring the degree of fuzziness of fuzzy sets through non-probabilistic entropy. According to De Luca and Termini's viewpoint [8], the model output vector can be viewed as a fuzzy set, and the fitted result $f(x_i)$ of the model is the membership function. The probability values represented by each element in the fuzzy set are the membership degrees of the sample x_i belonging to each category. The degree of fuzziness can be calculated according to the following formula:

$$F_{uzzy}(\mu_i) = -\frac{1}{C}\sum_{j=1}^{C}(\mu_{ij} \log \mu_{ij} + (1 - \mu_{ij}) \log(1 - \mu_{ij})) \tag{6}$$

When all elements in the fuzzy set μ_i are equal, the fuzziness of the predicted result reaches the highest level. When an element $\mu_{ij^*} = 1$ in the fuzzy set, the fuzziness is minimized, which means the uncertainty of the output is minimized.

3 METHODOLOGY

In this section, we present a new perspective regarding uncertainty processing and analyze the relationship between prediction uncertainty and learning performance.

3.1 A new viewpoint regarding uncertainty processing

Classification is one of the most fundamental tasks of deep learning, which refers to the modeling problem of predicting the class label for a given case of the input data. The last layer

of the deep learning model outputs a vector of probability distributions for the classification task, whose component values represent the probability of belonging to each class. Usually, we consider the class corresponding to the largest component value to be the predicted label for the case. However, for a given training case, the probability distributions of the output of different models may differ, but their predicted class labels may be the same. As shown in Fig.2, models A and B are two different deep learning models. For Case 1, Case 2, Case 3 in the training set, the class labels predicted by the two models are the same (i.e., they have the same training accuracy). In this case, which model should we choose from better generalization performance on the testing set?

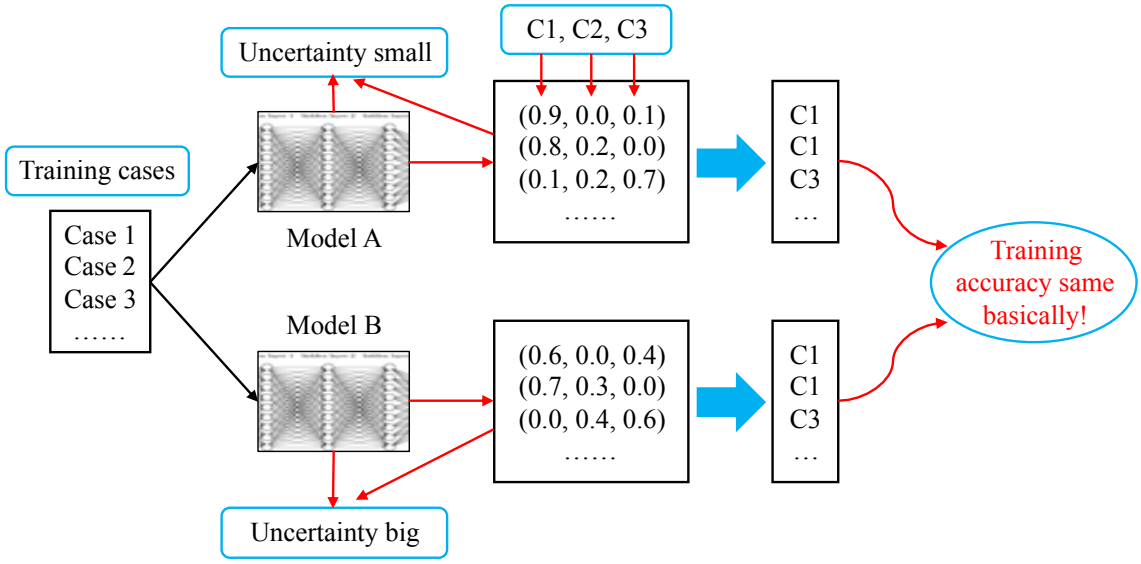

Figure 2: A new viewpoint regarding prediction uncertainty processing.

To address the above issues, this paper presents a novel viewpoint regarding uncertainty processing with the angle of prediction uncertainty of deep learning models. When the training accuracy of different models is basically the same, the model with large prediction uncertainty has better generalization performance.

To verify the viewpoint proposed in this paper, we use softmax with temperature parameters as the last layer of the neural network to analyze the relationship between the prediction uncertainty of the model and the generalization. Then, we present a method to flatten the prediction uncertainty. The schematic diagram of the network structure used in this paper is shown in Fig.3.

3.2 Neural network structure with softmax end

A multilayer perceptron (MLP) is the most basic neural network architecture, which consists of input layer, hidden layer, output layer, and activation function. The layers of the network are fully connected to each other, and each connection has a weight w. The overall composition is a linear mapping from input to output. The output of the hidden layer is as follows.

$$\begin{cases} h_1 = f(W_0 x + b_0) \\ h_i = f(W_{i-1} h_{i-1} + b_{i-1}) \end{cases} \tag{7}$$

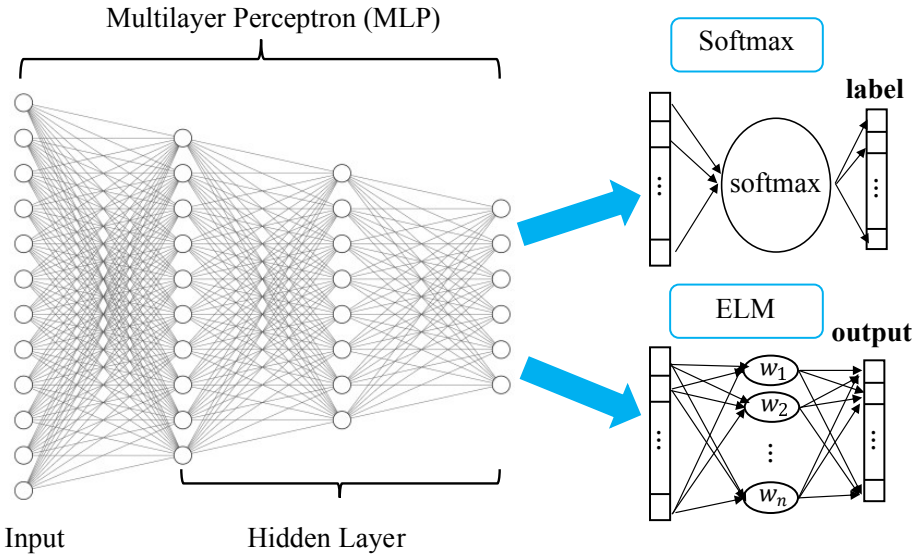

Figure 3: Schematic diagram of the network structure.

where $f(\cdot)$ is the activation function; h_i denotes the output vector of the i_th hidden layer; x is the input data (number of cases: m, number of features: n); W_i and b_i indicate the weights and biases between the network at layer i and the network at layer $(i-1)$, respectively.

The output vector h of the MLP network is used as the input of the softmax classifier to obtain the probability distribution of the predicted class labels of the input cases. The specific form of this probability distribution depends on the temperature parameter of softmax, as shown in Eq.8.

$$\hat{y}_i = \frac{\exp(z_{ij}/T)}{\sum_{k=1}^{K}\exp(z_{ik}/T)} \tag{8}$$

where z_{ij} is the output vector of the last layer of the MLP, which represents the j_{th} element of the feature vector of the i_th sample; K is the number of class labels; and T is the temperature parameter. When calculating the probability distribution of softmax, T can adjust the "weight" of different values. When $T <= 1$, the weight of the maximum value in the label probability distribution will be larger, which means that the label will be more "sharp", i.e., the model will have less prediction uncertainty. Conversely, when $T > 1$, the weight of the maximum value in the label becomes smaller, and the weights of the values other than the maximum value become larger, making the label more "smooths", i.e., the prediction uncertainty of the model is larger.

In this paper, we use to cross entropy as the loss function of the model, as shown in Eq. 9.

$$\mathcal{L} = -\sum_{i=1}^{K} y_i \log(\hat{y}_i) \tag{9}$$

where y_i is the real label of the case.

Remark: The output of softmax is a probability distribution on label space and the entropy of this distribution, i.e., the prediction uncertainty, can be adjusted by changing the temperature parameter T. It provides a simple approach to modifying the prediction uncertainty without changing the order of distribution components, i.e., without changing the final result of classification. However, this type of prediction uncertainty adjustment is subject to the formular (8) and its flexibility is a little weak. More effective approaches to prediction uncertainty adjustment are still to be studied.

3.3 Neural network structure with ELM end

The extreme learning machine (ELM) is a machine learning algorithm proposed by huang et al. [19] based on single hidden layer feedforward neural networks (SLFNs). Compared with the traditional feedforward neural network, the innovation points of ELM are as follows: (1) the connection weights W of the input and the hidden layer, and the bias b of the hidden layer can be randomly set without adjustment after setting. (2) The connection weights between the hidden layer and the output layer does not need to be adjusted iteratively, but are determined once by solving the generalized inverse matrix.

The output of ELM model can be expressed as:

$$f(x) = \sum_{i=1}^{L} \beta_i H_i(x) \tag{10}$$

where L is the number of hidden layer nodes; β_i is the weight of the i_{th} hidden node to the output node; $H_i(x)$ denotes the output function of the i_{th} hidden node, then

$$H_i(x) = g(W_i * x + b_i) \tag{11}$$

Where W_i is the input weight connecting the i_{th} hidden node; b_i is the bias of the i_{th} hidden node; $g(\cdot)$ is the activation function.

Given N cases $\{(x, y)\}_{l=1}^{N}$, where $x_l \in R^n$ is the input vector and $y_l \in R^m$ is the corresponding expected output, i.e

$$y_l = \sum_{i=1}^{L} \beta_i H_i(x), l = 1, 2, \ldots, N \tag{12}$$

Since the weight matrix between the hidden layer and the output layer is a pseudo-inverse matrix, we can obtain

$$Y = H\beta \tag{13}$$

where H is the output matrix of the hidden layer and Y is the target matrix. The process of training ELM then translates into finding the least-squares solution to Eq. 13, i.e

$$\min_{\beta} \|H\beta - Y\| \tag{14}$$

$$\beta = H^{+}Y \tag{15}$$

where H_+ is the M-P (Moore-Penrose) generalized inverse matrix of the hidden layer output matrix H. In the classification task, the decision equation of ELM is formulated as

$$label(x) = argmax f(x) \tag{16}$$

In this section, we use the ELM model to replace the softmax layer in Section 3.2. Specifically, (h, y) is used as the input of the ELM model for model training. where h is the output vector of the last layer of the MLP, and y is the real label of the example.

Remark: The output of ELM is not a probability distribution on label space and the prediction uncertainty is evaluated by normalization of outputted vector. A comparison of prediction uncertainty between outputs of softmax and ELM is listed in Tab. 3. There are not obvious and straightforward rules discovered from the comparison, but some trends can be speculated.

4 EXPERIMENTS

More experiments have been conducted for our approach verification. In this section, we present some selected datasets, evaluation metrics, implementation details and experimental results, and provide a comprehensive analysis of the experimental results.

4.1 Datasets and evaluation metrics

Nine datasets are selected from UCI machine learning database (https://archive.ics.uci.edu) for our approach verification. The details of each dataset are shown in Tab.1, and the training and testing sets are divided according to the ratio of 7:3 (except for Avila and Hand-written-Digits, both of which have already well-divided the training and test sets).

Table 1: Distribution of the datasets.

Datasets	Features	Class	Train set	Test set
Avila	10	12	10430	10437
MAGIC	10	2	13314	5706
Musk-2	166	2	4618	1980
HandwrittenDigits	16	10	7494	3498
SDD	46	11	40956	17553
Landsat	36	6	4504	1931
Letter	16	26	14000	6000
Segmentation	19	7	1589	682
Waveform	21	3	3500	1500

This study uses classification accuracy as a performance indicator of the model and calculates the information entropy of the model output using Eq. 5, which in turn quantifies the prediction uncertainty of the model.

4.2 Implementation details

In this study, the MLP network includes an input layer and three hidden layers, where the number of neurons in the input layer is the number of batch input instances, and the number of neurons in the three hidden layers is set to be twice, 5 times, and twice the number of neurons in the input layer, respectively. The temperature parameter of softmax is set to T=[0.1, 1.0, 10, 100]. The search space of the learning rate is [0.0001, 0.0002, 0.0003, 0.0005, 0.001, 0.002, 0.003, 0.005, 0.01, 0.02, 0.03, 0.05]. The selection of these parameters is empirically, without a processing of parameter optimization.

4.3 Experimental results

Experimental setups are given in section 4.2. The relationship between prediction uncertainty and training (test) accuracy on the 9 datasets is shown in Fig. 4. The horizontal axis is the information entropy of the prediction label distribution, which indicates the prediction uncertainty; the vertical axis indicates the training (test) accuracy. Analyzing Fig. 4, we found that the accuracy of the training set showed a trend of ”decreasing and then increasing” as the training entropy increased, while the accuracy of the test set showed a trend of ”increasing and then decreasing” as the test entropy increased (excluding Landsat datasets). As the entropy of the predicted label probability distribution (prediction uncertainty) increases, the accuracy

of the training and test sets shows an opposite trend. Specifically, models with larger prediction uncertainty within a certain range have stronger generalization performance. The range of prediction uncertainty is related to the specific dataset features.

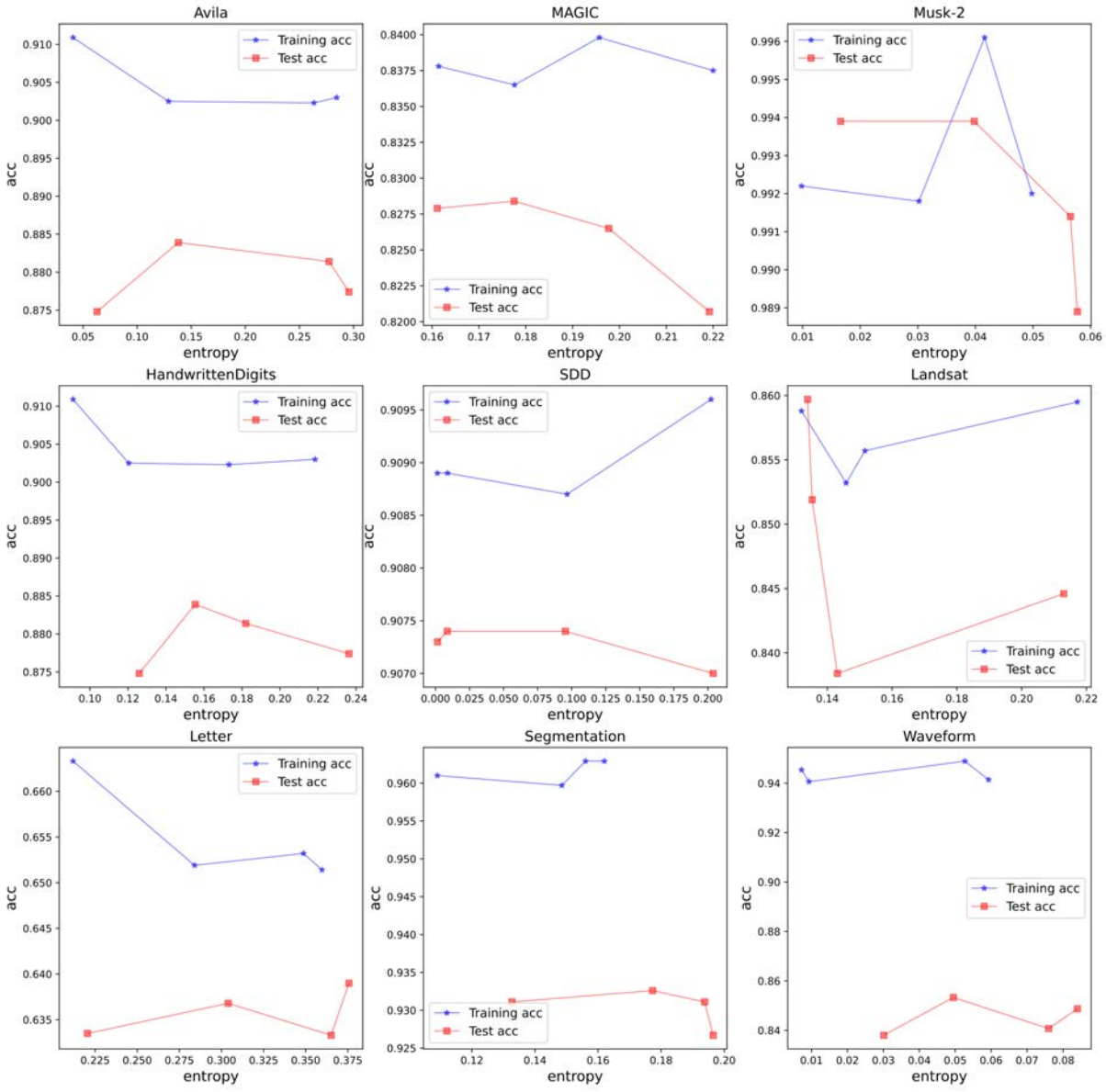

Figure 4: Relationship between training(test) accuracy and prediction uncertainty.

Therefore, we extracted some meta-features about the complexity measure of the dataset, which are described in detail in Tab. 2. Where, F1 is the Fisher discriminant ratio, which represents the separability of the class by calculating the maximum discriminant power of each feature; N1 is the proportion of class boundary sample, which represents the complexity of the class boundaries by counting the number of boundary samples; C1 is the mean of the variance of each feature in the training set; C2 is the mean of the standard deviation of each feature in the test set.

Comparing Fig. 4 and Tab. 2, it is found that the generalization ability of the model decreases significantly with larger values of the proportion of class boundary samples (N1) on the

Table 2: Meta-features of the datasets.

Datasets	Features	Class	F1	N1	C1	C2
Avila	10	12	0.8058	0.2209	0.2625	0.1968
MAGIC	10	2	0.7875	0.2899	0.0645	0.1619
Musk-2	166	2	0.9266	0.0770	0.3799	0.6473
HandwrittenDigits	16	10	0.2780	0.0157	0.0003	0.0002
SDD	46	11	0.0363	0.0716	0.0	0.0
Landsat	36	6	0.2173	0.1507	0.0393	0.0323
Letter	16	26	0.3915	0.1016	0.0001	0.0
Segmentation	19	7	0.0334	0.0647	0.0003	0.0002
Waveform	21	3	0.6197	0.3284	0.0001	0.0001

Waveform, Avila and MAGIC datasets.

In addition, we also analyzed the distribution of samples misclassified by different models with essentially the same training accuracy. As shown in Fig. 5, the prediction uncertainty of the model on the HandwrittenDigits and Letter datasets is Model A <Model B <Model C <Model D. The analysis of the results found that the misclassified samples of Model B and Model C are basically the same, indicating that the prediction uncertainty was within a certain range and the models have similar generalization to the test samples.

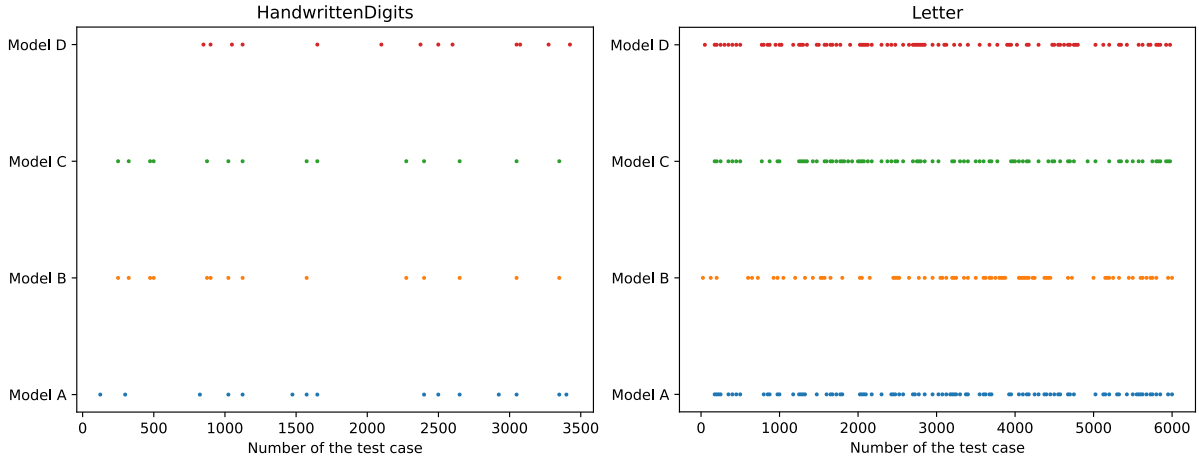

Figure 5: Number of test cases misclassified by different models.

According to the method in Section 3.3, we use the ELM model to replace the softmax layer, and the comparison of its experimental results is shown in Tab. 3. The results on most of the datasets show that the MLP with ELM end model predicts a probability distribution with increased information entropy compared to MLP with softmax end, and its accuracy on the test set is significantly improved. That is, the MLP with ELM end model proposed in this paper can improve the generalization of the model by increasing the prediction uncertainty.

Fig. 6 shows the probability prediction distribution of a case. The test sample is from the Hand-written-Digits dataset, and its ground-truth class is ”9”. In contrast to the MLP with softmax End model, the MLP with ELM End model to flatten the residual probability of all incorrect classes while guaranteeing the probabilistic predicted value of the correct class. This enables the model to make correct prediction even when the prediction uncertainty is large.

Table 3: Comparison of the results of MLP ending with softmax and ELM respectively.

Datasets	MLP with softmax End		MLP with ELM End	
	Acc	Entropy	Acc	Entropy
Avila	**0.7512**	0.1381	0.6716	3.1925
MAGIC	0.8207	0.2192	**0.8298**	0.6553
Musk-2	**0.9939**	0.0398	0.9227	0.4644
HandwrittenDigits	0.8839	0.1554	**0.9957**	2.7775
SDD	0.9073	0.0015	**0.9990**	2.6356
Landsat	0.8446	0.2130	**0.8566**	1.9804
Letter	0.6368	0.3040	**0.9500**	4.4435
Segmentation	0.9326	0.1773	**0.9897**	2.1754
Waveform	0.8533	0.0495	**0.8827**	0.8682

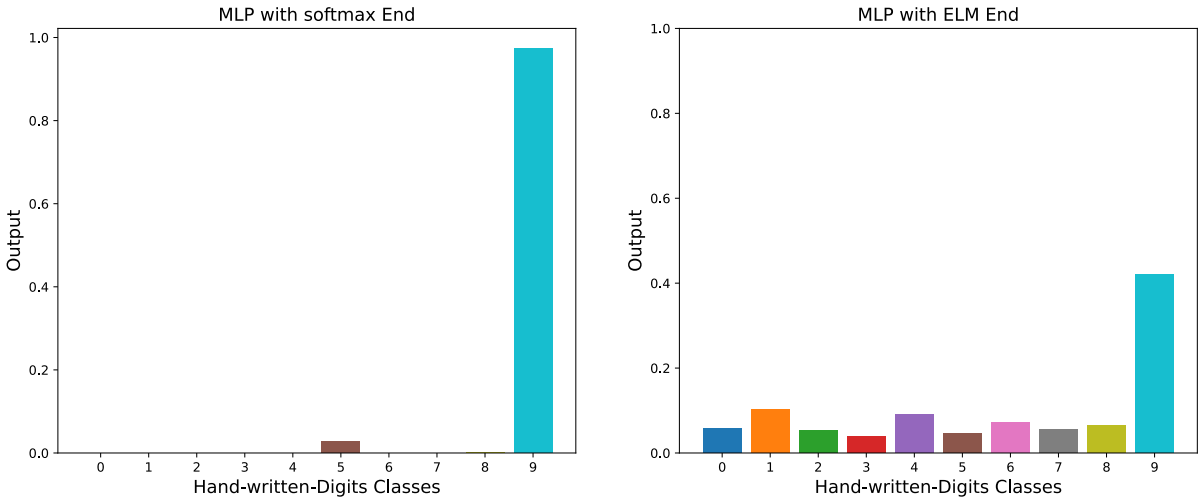

Figure 6: Prediction uncertainty for softmax and ELM output.

5 CONCLUSION

In this study, we summarized three types of uncertainty involved in the process of machine learning, namely data uncertainty, model uncertainty, and prediction uncertainty from the model output (a probability distribution on the label space) to the final class determination. A novel viewpoint on uncertainty processing is presented, which indicates that, to some extent, the model with larger prediction uncertainty has a stronger generalization performance. Furthermore, we find that the accuracies of the training and testing sets show inconsistent trends as the prediction uncertainty increases. Finally, we developed an easy-to-use approach to improving the generalization ability of model by replacing the final softmax layer with a random weight layer.

REFERENCES

[1] Rudolf Kruse, Erhard Schwecke, and Jochen Heinsohn. *Uncertainty and vagueness in knowledge based systems: numerical methods*. Springer Science & Business Media, 2012.

[2] Frank Hyneman Knight. *Risk, uncertainty and profit*, volume 31. Houghton Mifflin, 1921.

[3] Xizhao Wang. Uncertainty modeling in learning from big data, 2018.

[4] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[5] Eric Jang. Uncertainty: a tutorial. [EB/OL]. `https://blog.evjang.com/2018/12/uncertainty.html` Accessed Sep 27, 2022.

[6] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

[7] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[8] Aldo De Luca and Settimo Termini. A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory. *Information and control*, 20(4):301–312, 1972.

[9] Yufei Yuan and Michael J Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and systems*, 69(2):125–139, 1995.

[10] Pingke Li and Baoding Liu. Entropy of credibility distributions for fuzzy variables. *IEEE Transactions on Fuzzy Systems*, 16(1):123–129, 2008.

[11] Andrey Bronevich and George J Klir. Measures of uncertainty for imprecise probabilities: an axiomatic approach. *International journal of approximate reasoning*, 51(4):365–390, 2010.

[12] Inés Couso and Luciano Sánchez. Upper and lower probabilities induced by a fuzzy random variable. *Fuzzy Sets and Systems*, 165(1):1–23, 2011.

[13] Anping Zeng, Tianrui Li, Jie Hu, Hongmei Chen, and Chuan Luo. Dynamical updating fuzzy rough approximations for hybrid data under the variation of attribute values. *Information Sciences*, 378:363–388, 2017.

[14] Degang Chen, Xiaoxia Zhang, and Wanlu Li. On measurements of covering rough sets based on granules and evidence theory. *Information Sciences*, 317:329–348, 2015.

[15] Yiguo Wang and Qinghua Zhang. Uncertainty of rough sets in different knowledge granularities. *Chinese Journal of Computers*, 31(9):1588–1598, 2009.

[16] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural safety*, 31(2):105–112, 2009.

[17] David A Nix and Andreas S Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pages 55–60. IEEE, 1994.

[18] Lotfi A Zadeh. Information and control. *Fuzzy sets*, 8(3):338–353, 1965.

[19] Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE International Joint Conference on Neural Networks*, volume 2, pages 985–990, 2004.