A case weighted similarity deep measurement method based on a self-attention Siamese neural network

Zijun Cheng
Faculty of Information Technology, Beijing University of Technology
Aijun Yan (yanaijun@bjut.edu.cn)
Faculty of Information Technology, Beijing University of Technology

Research Article

Keywords: case retrieval, similarity measure, self-attention mechanism, Siamese neural network, feature weights

Posted Date: September 14th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2039632/v1

A case weighted similarity deep measurement method based on a self-attention Siamese neural network

Zijun Cheng [1, 2], Aijun Yan [1, 2, 3,†]

[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China;

[2] Engineering Research Center of Digital Community, Ministry of Education, Beijing 100124, China

[3] Beijing Laboratory for Urban Mass Transit, Beijing 100124, China

Abstract: To improve the accuracy of similarity measures in case-based reasoning, in this paper, we propose a deep metric learning method based on a self-attention mechanism and a Siamese neural network to realize the weighted similarity measure between cases. The method maps the original case features to the new feature space through the Siamese neural network and then assigns the feature weights through the scoring function in the self-attention mechanism. Finally, a metric function is added to the contrastive loss to measure the case similarity. Experiments show that the accuracy of this method is better than other algorithms in the similarity measure and can improve the accuracy of case retrieval.

Key words: case retrieval, similarity measure, self-attention mechanism, Siamese neural network, feature weights

1. Introduction

Case-based reasoning (CBR) is a method of reasoning and solving new problems (target cases) by using empirical cases (source cases) to solve similar problems. It includes four activities: *retrieve*, *reuse*, *revise* and *retain*. Referred to as the 4R cycle [1], it is widely used in industrial control [2], medical applications [3], image processing [4] and many other fields. In CBR, case retrieval retrieves the most similar cases from the case base according to the similarity between new problems and empirical cases and provides alternative solutions to new problems for case reuse [5]. The similarity measure plays an important role in case retrieval and affects the quality of the CBR problem solution.

Similarity measure methods usually use the measure function (or improved measure function) [6] or a hybrid similarity measure algorithm [7] to calculate the similarity between cases. Since the contribution of different features to the solution of the problem varies, directly using the above metric function to calculate the similarity will lead to a decrease in accuracy. Therefore, the weighted similarity measure method can improve the accuracy of the measure to varying degrees. There are three types of distribution methods of feature weights: subjective methods, objective methods and subjective and objective combination methods. Subjective weight methods include the analytic hierarchy process [8], eigenvector method [9], and direct scoring [10]. Objective weight methods include the entropy weight method [11], genetic algorithms [12], and neural networks [13]. For example, in Ref. [13], deep belief networks were combined to preserve the features of sample attributes, to achieve the distribution of feature weights and to improve the similarity measure function through Bayesian network parameter learning. In Ref. [14], information entropy was used to calculate feature weight in the k-NN retrieval method and then used a distance metric function to calculate similarity. In Ref. [15], combining the subjective weight with the objective weight was proposed to form the comprehensive weight and then the mixed similarity measure was combined with the comprehensive weight to establish the calculation formula of global similarity.

† Corresponding author at: Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China.

Email address: chengzijun@emails.bjut.edu.cn (Z. Cheng), yanaijun@bjut.edu.cn (A. Yan).

The subjective method and the subjective and objective combination methods mentioned above depend on the professional knowledge and experience of experts and are subjective. Although objective methods do not interfere with human factors, different methods still have some disadvantages. For example, the entropy weight method does not consider the interaction between features, and the genetic algorithm easily falls into a local minimum. Therefore, further research is required to improve the accuracy of the similarity calculation between cases in the weighted similarity measurement method based on the feature weight distribution.

In the field of machine learning, the core concept of metric learning is to calculate the similarity between samples to reduce the distance between samples of the same class and increase the distance between samples of different classes [16]. For example, Refs. [17, 18] mentioned measuring the distance (or similarity) between data by calculating the corresponding pairwise matrix function. Learning nonlinear features and the similarity between data through deep structure is called deep metric learning [19]. One effective method is to use the SNN to complete the deep metric learning process, but the SNN ignores the distribution of feature weights in the calculation process, which still impacts the accuracy of the similarity measure. Therefore, in this paper, we use the self-attention (SA) mechanism [21] to obtain the characteristics of global and key information and combines it with the SNN to propose a weighted similarity depth measurement method based on the SASNN. In this method, a self-attention mechanism is added between the hidden layer and the output layer of the SNN to measure the correlation among features and between features and cases to realize the weight distribution of case features. On this basis, a metric function is added to the output layer to calculate the similarity between cases. Experiments are conducted to test the effectiveness of the SASNN method.

The rest of this paper is organized as follows. Section 2 introduces SNN metric learning and its problems, Section 3 describes the SASNN weighted similarity depth measurement method, Section 4 includes the experiment and analysis results, and Section 5 concludes the paper.

2. SNN Metric Learning and Problem Analysis

SNN is a deep metric learning method that can handle nonlinear data structures. It was originally proposed by Bromley et al. for signature verification [20]. It has a Siamese structure. The two subnetworks have the same architecture, parameters and weights, and the weights are shared. Its output is the similarity between the two feature vectors by the contrast loss function. The learning mechanism for using the SNN for similarity measurement between cases is as follows.

In general, the basic case consists of two parts, describing the characteristics of the problem and the solution to the problem. In case library C, the total number of cases is m, and the case can be expressed as:

$$c_i = (x_i, y_i), i = 1, 2, \mathrm{L}\ m \tag{1}$$

where $x_i \in ¡^{m}$ is the problem description of the i-th source case and y_i is the solution. The number of features is n, and the representation of x_i can be expressed as:

$$x_i = (x_{i1}, x_{i2}, \mathrm{L}\ , x_{in}) \tag{2}$$

Before using the SNN to calculate the similarity, the different cases are combined in pairs, and the original case base C forms a new case pair dataset $S = \{(x_i, x_i, y_{ij}), i, j = 1, 2, \mathrm{L}\ , m\}$. A binary label $y_{ij} = \{0, 1\}$ is assigned to every pair. If two data vectors x_i and x_j are similar, then y_{ij} takes the value 0. If the vectors correspond to different vectors, then y_{ij} takes the value 1.

We use f to represent the mapping function of the SNN. The input vectors x_i and x_j share weights through the network and are then transformed into new vectors $h_i = f(x_i)$ and $h_j = f(x_j)$. In the new

feature space, to obtain the similarity between the vectors h_i and h_j, a metric function $D_W = (h_i, h_j)$ must be defined in the feature space, which is commonly implemented by Euclidean distance [22].

We choose the contrast loss function to guide the training of the network to learn the similarity of data pairs. For brief representation, $D_W = (h_i, h_j)$ is rewritten as D_W, and the contrast loss function is expressed as follows:

$$l(x_i, x_j, y_{ij}) = (1 - y_{ij}) D_W^2 + y_{ij} \max(\tau - D_W, 0)^2 \tag{3}$$

Where τ is the threshold, such that a pair contributes to the loss only if their distance belongs to (0, τ), which means that only the Euclidean distance of dissimilar features is considered between (0, τ). When the distance exceeds τ, the loss value is set to 0. Finally, the judgment results of whether the case pair is similar are output as the basis for case retrieval.

In the process of using the SNN to learn the similarity measure between cases, the original features are mapped to a new feature space through the network, and the new feature vectors h_i and h_j are directly used to calculate the similarity using the metric function. This method has the following drawback. First, the contribution of the new feature vector in the case is different; the SNN regards it as equivalent to being directly used in the calculation, which leads to error in the results. Second, features interact with each other and are related to each other. Ignoring this correlation also leads to inaccurate determination of similarity. Therefore, the use of the SNN for the similarity measurement process requires further study.

3. SASNN weighted similarity depth measurement method

Considering the above problem, to make full use of the contribution of each feature to the case solution, we allow the SA mechanism [21] to redistribute the resources which are originally evenly distributed according to the importance of the object and capture the internal correlation of the data or features. It can both represent the contribution of the features in the case and represent the correlation between the features [23]. Therefore, the SA mechanism is introduced into the SNN metric learning process, and a SASNN weighted similarity depth measurement method is proposed. The structure principle and algorithm implementation process of the method are introduced in the following subsections.

3.1 The structure of SASNN

The structure of the combination of SA and SNN is shown in Figure 1, which is divided into three parts: the SNN layer, SA layer and similarity measurement layer. They are explained in detail as follows:

(1) SNN layer: Both branch networks consist of four parts; the first two are two convolutional layers and a pooling layer, the third is three convolutional layers and a pooling layer, and the fourth is three fully connected layers alternating with three dropout layers. The number of channel convolutional layers of the network is 32, 64, and 128 in the first three parts, respectively, and the ReLU activation function is used in each layer of the network. The dropout layer is used to prevent overfitting in the network training process and improve the generalization ability of the network. The main function of the SNN layer is to realize the feature extraction and processing of the original data after the above multilayer network processing and convert the two inputs x_i and x_j into feature vectors h_i and h_j.

(2) SA layer: The SA layer projects the output feature vectors h_i and h_j received from the SNN layer into three subspaces: the query vector quantum space Q, the key vector quantum space K, and the value vector quantum space V. Then, the weight of each feature is calculated using the scoring function, and the weighted feature vectors a_i and a_j are output. The MatMul module performs the dot product operation,

and the Scsle module performs the scaling operation to smooth the final attention distribution, which can alleviate the problem of the small gradient of the Softmax function when the dimension of the input vector is high.

(3) Similarity measure layer: The similarity measure is performed on the weighted feature vectors a_i and a_j. The Euclidean distance metric function is embedded into the lambda layer to construct the similarity measure layer, and the similarity calculation is performed. The sigmoid function compresses the numerical range of the calculation result to (0,1) and finally outputs the similarity y_{ij} between x_i and x_j.

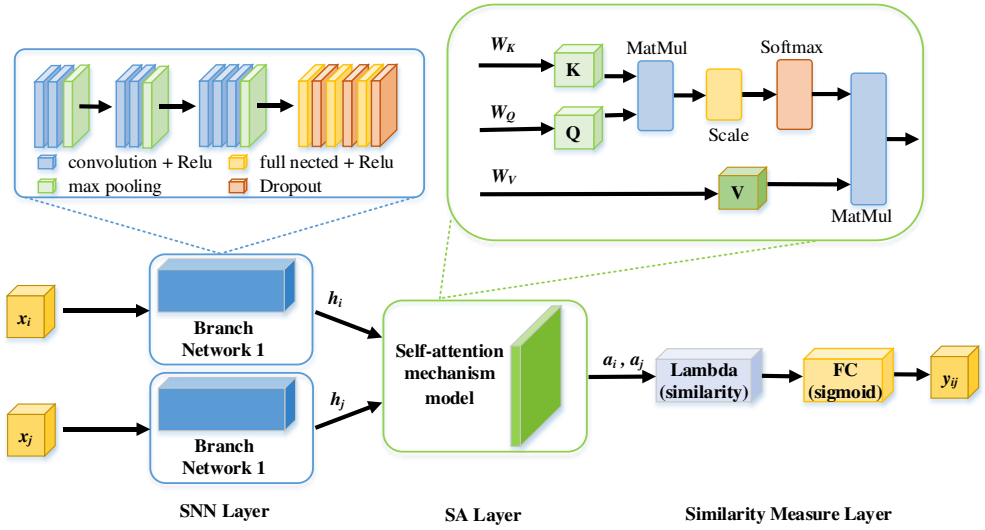

Figure 1 SASNN Model Structure

3.2 Algorithm implementation

Suppose there is a pair of cases $(x_i, x_j), x_i, x_j \in ¡^m$; when the pair is input into the network, the original data are mapped to a new feature space via two branch networks. For simplicity, each branch network is considered a hidden layer, and the corresponding input and output are represented by x and h, respectively. Suppose that $f: ¡^m \to ¡^d$ is a map of x to h; then, the following holds:

$$h = f(x, W_s, b_s) \tag{4}$$

Where W_s and b_s are the network weight and bias, respectively, and new feature vectors $h_i \in ¡^d$ and $h_j \in ¡^d$ are obtained.

Different weights are assigned to the feature vector (h_i, h_j) by the self-attention layer. Taking h_i as an example, the feature vector h_i is projected into Q, K, and V spaces. Ignoring the subscript i for simplicity, we obtain:

$$\begin{cases} q = W_Q h_i = \{q_1, q_2, \mathrm{L}\ , q_n\} \\ k = W_K h_i = \{k_1, k_2, \mathrm{L}\ , k_n\} \\ v = W_V h_i = \{v_1, v_2, \mathrm{L}\ , v_n\} \end{cases} \tag{5}$$

The attention score is calculated by the attention score function and normalized by the softmax function, where the scaled dot product function is used:

$$s_k(k, q_k) = \frac{k^T q_k}{\sqrt{d_k}} \tag{6}$$

$$\alpha_k = softmax(s_k) \tag{7}$$

where d_k is the input dimension, $1 \le k \le n$, and α_k is the weight coefficient of v; thus, the attention

value is obtained:

$$A(h) = \sum_{k=1}^{n} \alpha_k \cdot v \tag{8}$$

Setting $a_i = A(h_i)$, $a_j = A(h_j)$, the weighted feature vectors of the case are obtained.

Therefore, the distance of the feature vector, $D_W(a_i, a_j)$, is calculated by the Euclidean distance metric function, and the sigmoid function is used to compress the output at (0,1) to obtain the similarity sim, $sim(a_i, a_j)$, that is:

$$D_W(a_i, a_j) = \sqrt{\sum_{k=1}^{n} (a_{ik} - a_{jk})^2} \tag{9}$$

$$sim(a_i, a_j) = sigmoid(D_W(a_i, a_j)) \tag{10}$$

The contrast loss function, $l(x_i, x_j, y_{ij})$, measures the prediction result according to the value of $sim(a_i, a_j)$, that is, the error. In the learning process, the optimizer minimizes the error of each training sample, updates the network parameters, and identifies the optimal model.

$$l(x_i, x_j, y_{ij}) = \begin{cases} sim^2(a_i, a_j), y_{ij} = 0 \\ \max(\tau - sim(a_i, a_j), 0)^2, y_{ij} = 1 \end{cases} \tag{11}$$

To obtain the optimal model, the main task of SASNN is to train and learn the weight parameters in the network. Supposing that $\boldsymbol{W} = (W_s, W_Q, W_K, W_V)$ are the weight parameters of SASNN, the objective function of SASNN can be expressed as:

$$J(\boldsymbol{W}) = \sum_{ij} l(x_i, x_j, y_{ij}, \boldsymbol{W}) + \mu R(\boldsymbol{W}) \tag{12}$$

where $R(\boldsymbol{W})$ is the regularization term, which is used to prevent the overfitting of the neural network and improve the generalization ability of the network. μ>0 is the regularization parameter to control the regularization strength. The contrast loss function $l(x_i, x_j, y_{ij}, \boldsymbol{W})$ is used to meet the similar or dissimilar conditions of the case pairs to enhance the case characteristics, which makes the distance between similar cases as small as possible while ensuring that the distance between dissimilar cases is as large as possible, realizing the similarity measurement of case pairs.

Based on the above analysis, the pseudocode of the SASNN algorithm is as follows:

Algorithm 1: Siamese neural networks based on self-attention mechanism

Input: case pairs (x_i, x_j) label y_{ij}
Output: predict labels y'_{ij},similarity sim

Initialization SASNN parameters: network weights W, bias b;
while $epoch \leq epoch_{max}$ **do**
  $h = f(x, W_s, b_s)$;
  Get the feature vector h_i, h_j by SNN layer;
  $k = W_k h, q = W_q h, v = W_v h$;
  $s_k(k, q_k) = \frac{k^T q_k}{\sqrt{d_k}}$;
  $\alpha_k = softmax(s_k)$;
  $A(h) = \sum_{i=1}^{n} \alpha_k v$;
  $a = A(h)$;
  Get the feature weights $a_i, a_j$by SA layer;
  $D_w = \sqrt{\sum_{i=1}^{n} (a_{ik} - a_{jk})_2}$;
  $sim(a_i, a_j) = sigmoid(D_w(a_i, a_j))$;
  Updata SASNN parameters W, b;
end
Return SASNN model

4. Experiment

To investigate the effectiveness of the SASNN algorithm and the effect of practical application, we selected three representative algorithms to compare with the SASNN algorithm: the k-NN (k-nearest neighbor) algorithm based on Euclidean distance, the deep forest measurement algorithm SDF (Siamese Deep Forest) based on tree structure, and the deep neural network measurement algorithm SNN. The fault diagnosis experiment was implemented to demonstrate the practicability of the algorithm. In the following subsections, the data processing, comparative experiments and fault diagnosis applications of the experiments are introduced.

4.1 Data processing

The experimental data were selected from 10 classification data sets in the UCI database. Table 1 lists the names, abbreviations, feature number *F*, case number *N*, and category number *C* of these data sets. Before the experiment, these data were standardized and matched as follows.

Table 1 Data set information

Data set	Abbreviation	*F*	*N*	*C*
Connectionist Bench	CB	60	208	2
Ionosphere	Iono	33	351	2
Mammographic Mass	MM	5	961	2
Breast Cancer Wisconsin	BCW	30	569	2
Cardiotocography	Cradi	22	2126	3
Statlog (Vehicle Silhouettes)	Sta(vs.)	18	846	4
Glass Identification	GI	9	214	6
Yeast	Yea	8	1484	10
Seeds	Sed	7	210	3
Parkinsons	Parki	22	195	2

(1) Standardization: Due to the same feature value of the original data, the range and unit of different feature values were different, which posed a challenge to the training of the network model. Therefore, the following Min-max standardization was used to normalize the data:

$$y_i = \frac{x_i - \min\limits_{1\le j\le n}\{x_j\}}{\max\limits_{1\le j\le n}\{x_j\} - \min\limits_{1\le j\le n}\{x_j\}} \tag{13}$$

Here, $y_i \in [0,1]$.

(2) Case matching: After preliminary processing of the data, a combination of case pairs was performed, and a similar case and a dissimilar case are randomly selected for each case to form a case set $S = \{(x_i, x_i, y_{ij}), i, j = 1,2,\mathrm{L}\ ,m\}$. If the case pairs ($x_i$, x_j) were similar, then $y_{ij} = 0$; conversely, $y_{ij} = 1$. For the 2-category problem, a case was randomly selected for each case in the same class and different classes. For multiclass data sets, a dissimilar case was randomly selected in each different class, and the same number of similar cases was selected in the same class. They were paired with the same case to expand the number of case pairs to improve the performance of similarity measure learning.

4.2 Contrast experiment

The ratio of the training set and test set was 3:1, the learning rate was 0.001, and the batch size was set to 128. The initial number of iterations of the experiment was 1000, and we set it to stop training early to prevent network overfitting. The RMSProp algorithm was selected as the optimization algorithm,

accuracy was selected as the evaluation function, and the ReLU function was selected as the activation function (excluding the output layer).

In the training process of the SASNN model, the criterion for similar results was set to less than 0.5 and that for dissimilar results was set to greater than 0.5. Figure 2 shows the comparison diagram of the loss function and accuracy of the model training. With the increase in the number of iterations, the loss decreased continuously, and the accuracy increased continuously and tended to be stable quickly, indicating that the convergence process of the model was faster. The function curves of the training set and the test set were similar, indicating that the stability was better.

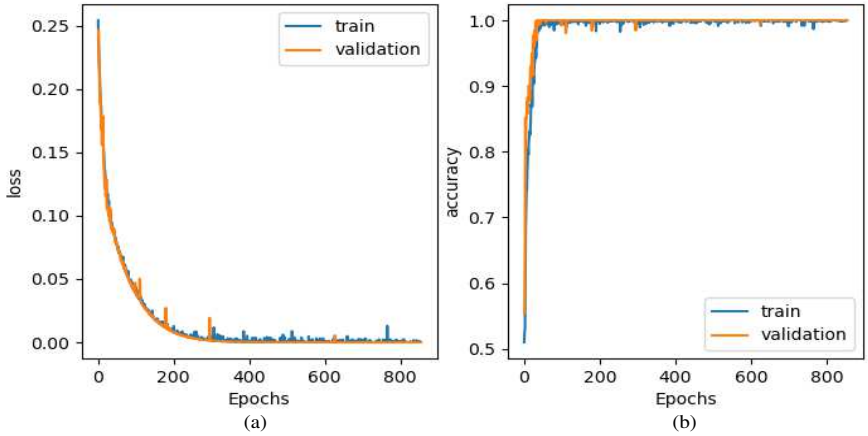

Figure 2 Experimental results. (a) loss, (b) accuracy

Figure 3 shows the data for some cases; case 0 is the target case and cases 1-7 are the source cases for the similarity measurement. Cases 1, 2 and 0 have similar data with $y_{ij}=0$; the others are not similar to case 0 with $y_{ij}=1$. Table 3 shows the results of using the Euclidean metric function and SASNN method to calculate similarity. Since the distance metric function was used to calculate the similarity, the smaller the corresponding value is, the smaller the distance, and the greater the similarity.

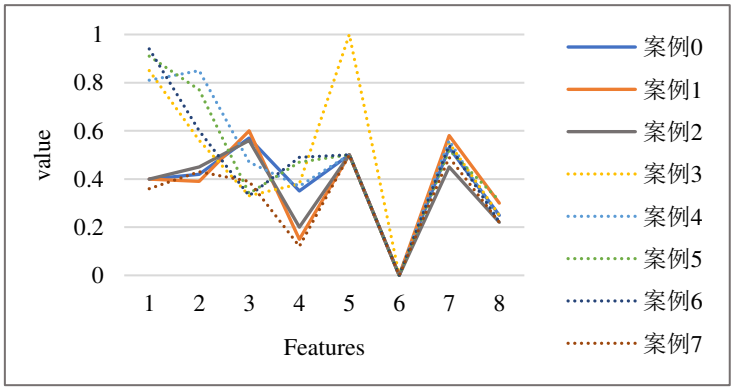

Figure 3 Data for cases 0 to 7

Table 2 Similarity of case pairs

(x_i, x_j)	(x_0, x_1)	(x_0, x_2)	(x_0, x_3)	(x_0, x_4)	(x_0, x_5)	(x_0, x_6)	(x_0, x_7)
y_{ij}	0	0	1	1	1	1	1
Euclidean	0.257	0.232	0.848	0.435	0.864	0.393	0.227
SASNN	0.336	0.208	0.899	0.867	0.963	0.861	0.919

Combining Figure 3 with Table 2, we found that the feature values of case 0 were very close to those of cases 1 and 2 and far from those of the other cases, which was consistent with the calculation results of SASNN similarity in the table. However, the results calculated by the Euclidean distance formula were

different. Taking the last group of cases as an example, case 7 was not similar to case 0, and the result calculated by the Euclidean distance metric function was smaller than the distance between similar cases. This means that in the process of case retrieval, dissimilar cases were misjudged as similar cases so that case retrieval obtained incorrect results, which seriously affects the follow-up process of case reasoning. At the same time, it demonstrated that SASNN can obtain better data representation through the extraction, transformation and dimension reduction of case features, improve the accuracy of similarity measurement, and reduce the error rate of case retrieval.

Table 3 Model prediction results

DATA SET	SNN		SASNN	
	Training	Testing	Training	Testing
CB	0.996	0.995	1.00	0.998
Iono	0.964	0.946	0.981	0.986
MM	0.760	0.740	0.792	0.735
BCW	0.996	0.994	1.00	0.997
Cardi	0.973	0.969	0.994	0.980
Sta(vs.)	0.993	0.831	0.998	0.869
GI	0.978	0.974	0.972	0.984
Yea	0.991	0.881	0.998	0.903
Sed	1.00	1.00	1.00	1.00
parki	1.00	1.00	1.00	1.00

Table 3 compares the prediction results of the SNN and SASNN on the training set and testing set in different data sets. It shows that the accuracy of both models was relatively high. Comparing the two models, we found that the SASNN model with an attention mechanism was better than the SNN model. This demonstrated that SASNN can reasonably allocate feature weights through a self-attention mechanism to improve the accuracy of the similarity measure.

The accuracy of the two methods was not high on the MM data set. Combining Table 1 with Table 3, we found that the number of features of the MM data set was very small. This means that less initial information was available, which is not conducive to the calculation of the similarity measure. In addition, the same feature value of similar cases was quite different. For any two similar cases, the values of multiple features may be quite different. This affects the accuracy of the measurement calculation to some extent.

Table 4 Experimental comparison results

DATA SET	k-NN(k=1)	k-NN(k=2)	k-NN(k=3)	SDF	SNN	SASNN
Iono	0.886	0.898	0.864	0.920	0.946	**0.983**
MM	0.738	0.725	0.725	0.783	0.759	**0.755**
Yea	0.534	0.520	0.545	0.739	0.890	**0.922**
Sed	0.925	0.868	0.868	0.968	1.00	**1.00**
Parki	0.939	0.898	0.918	0.893	1.00	**1.00**

To further study the performance of the model, some data sets were selected for comparison with k-NN, SDF and SNN. The experimental results are shown in Table 4. Note that the accuracy values of the SNN and SASNN in Table 4 are not the same as those in Table 3. The reason is that in Table 4, rerandomization was used to compose case pairs from all cases as a validation set and the trained model was input to obtain its accuracy. This makes the SASNN results more general and convincing. The accuracy of the SDF model comes from the best data results in Ref. [24]. As seen from Table 4, for

different data sets, the SASNN model had basically the highest accuracy and was more suitable for case similarity measures.

However, observing the results of the MM data set, we found that the accuracy of different methods was relatively low. This showed that the existing methods were limited by the initial information and case base data and still need to be further improved for cases with fewer features and larger feature value differences.

4.3 Fault diagnosis application

To illustrate the practical application of the SASNN method, we selected the fault diagnosis problem in the industrial process for the experiment. Gearbox fault diagnosis [25] was used to illustrate the experiment.

Table 5 Gearbox and bearing fault description

Location	Type	Description
Gear	Root	Crack occurs in the root of gear feet
	Surface	Wear occurs in the surface of gear
	Chipped	Crack occurs in the gear feet
	Miss	Missing one of in the gear
Bearing	Ball	Crack occurs in the ball
	Inner	Crack occurs in the inner ball
	Outer	Crack occurs in the outer ball
	Combination	Crack occurs in both the inner and outer ring

The faults were mainly classified into two categories. One was gear faults, including root, surface, chipped, and miss faults. The other was bearing faults, including ball, inner, outer, and combination faults. The specific characteristics and descriptions of the fault are shown in Table 5.

The gearbox data set contained bearing data and gear data sets, for a total of 8 characteristic parameters. Each subset contained four failure types and one health state. Note that the normal states of the two subsets were consistent, so there were nine states in total. Figure 4 illustrates the bearing fault data to show the bearing in five different states. From the diagram, it is clear that the data values of the inner, outer and ball were very close, so it is difficult to diagnose the gearbox fault.

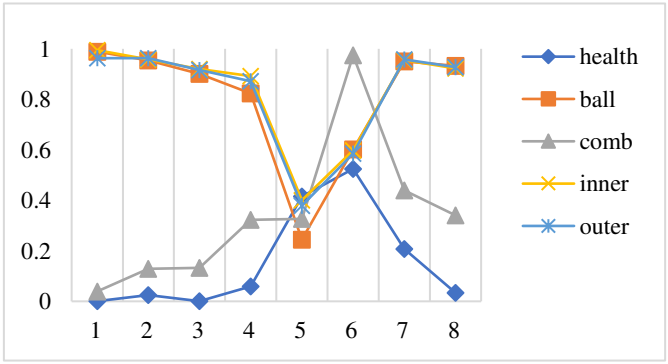

Figure 4 Bearing fault data

In the experiment, 1000 bearing data sets and 1000 gear data sets were selected and combined into a gearbox data set. Half of each data set were set aside as a training set and a testing set in the ratio of 3:1. The remainder used as the validation set. The gear fault, bearing fault and gearbox fault were tested, and the results are shown in Table 6.

Table 6 Fault diagnosis results

Methods	SNN			SASNN		
	Training	Testing	Validation	Training	Testing	Validation
Gear	0.994	0.964	0.973	0.998	0.970	0.967
Bearing	0.936	0.933	0.932	0.940	0.939	0.939
Gearbox	0.901	0.908	0.902	1.000	0.953	0.965

Table 6 shows that the accuracy of SASNN was relatively high, which was not affected by the small difference in data values and can accurately judge the gearbox fault. For gearbox faults and bearing faults, the results of the two methods were not much different, but in gearset faults, the accuracy of SASNN was higher than that of SNN. The reason is that gearset faults included gearbox faults and bearing faults, and the complexity of the mixed data set was much higher than that of the first two data sets, indicating that SASNN can better handle such complex data. It has high application value in practical data settings.

5. Conclusion

To improve the accuracy of similarity measurement, in this paper, we proposed a weighted similarity measurement method combining SA and SNN and conducted comparative experiments on multiple different data sets. The results showed that SASNN improved the accuracy of similarity measurement, thereby reducing case retrieval error. The main advantages of SASNN are as follows:

(1) Through the nonlinear transformation of the network, a better feature representation can be obtained. While reducing the distance between similar cases, the distance between dissimilar cases can be expanded, which effectively avoids the distance trap problem in the measurement calculation.

(2) Since the SNN is a one-shot model, it is applicable to two cases where the number of categories is small and the amount of data in each category is large or the number of categories is large (or the specific number cannot be confirmed), and the amount of data in each category is small. It is not limited by the size of the data and has higher robustness to imbalanced data. Therefore, SASNN can reduce the limitation of case base size or case imbalance and is suitable for similarity calculation of multiple data sets.

(3) The SA mechanism can fully measure the importance of features. It avoids the interference of human factors and can be used to assess the relationship between features and features, features and cases objectively and comprehensively to realize the reasonable distribution of feature weights.

The SASNN method also has certain limitations. In the experiment, we found that the accuracy of the calculation results was low for cases with less initial information and large differences in feature values. Therefore, future studies should be conducted to determine how to improve the method to increase the accuracy of the above similarity calculation.

Declarations

Funding

This work was supported by the National Natural Science Foundation of China (61873009, 62073006) and the Beijing Natural Science Foundation of China (4212032).

Author Contribution

Zijun Cheng: Data curation, Investigation, Software, Validation, Visualization, Writing-original draft. **Aijun Yan**: Conceptualization, Formal analysis, Funding acquisition, Project administration, Resources, Supervision, Writing-review & editing. All authors reviewed the manuscript.

Competing Interest

The authors have no relevant financial or non-financial interests to disclose.

Data Availability

References

[1] Aamodt A, Plaza E. Case-based reasoning: foundational issues, methodological variations, and system approaches. *Ai Communications*. 1994, 7(1): 39-59.

[2] Xiong Y, Liu Y, Chen Y. Research on the decision-making of processing method of scrap iron and steel on case-based reasoning. *Journal of Physics: Conference Series*. 2021, 1820(1): 012176.

[3] Jinli D, Zhibin L, Feng J. A dynamic case-based reasoning system for responding to infectious disease outbreaks. *Expert Systems with Applications*. 2022, 204: 117628.

[4] Nasiri S, Helsper J, Jung M. DePicT Melanoma Deep-CLASS: a deep convolutional neural networks approach to classify skin lesion images. *BMC Bioinformatics*. 2020, 21(2): 84-97.

[5] Homem T, Santos P, Costa A. Qualitative case-based reasoning and learning. *Artificial Intelligence*. 2020, 283: 89-111.

[6] Lin K. A case-based reasoning system for interior design using a new cosine similarity retrieval algorithm. Journal of Information and Telecommunication. 2019, 4(1):91-104.

[7] Zheng J, Wang Y M, Lin Y. Hybrid multi-attribute case retrieval method based on intuitionistic fuzzy and evidence reasoning. Journal of Intelligent and Fuzzy Systems. 2019, 36(12):1-12

[8] An S H, Kim G H, Kang K I. A case-based reasoning cost estimating model using experience by analytic hierarchy process. *Building and Environment*. 2007, 42(7): 2573-2579.

[9] Takeda E, Cogger K O, Yu P L. Estimating criterion weights using eigenvectors: A comparative study. *European Journal of Operational Research*. 1987, 29(3): 360-369.

[10] Bottomley P A, Doyle J R. Comparing the validity of numerical judgements elicited by direct rating and point allocation: Insights from objectively verifiable perceptual tasks. *European Journal of Operational Research*. 2013, 228(1): 148-157.

[11] Xiao F. EFMCDM: Evidential fuzzy multicriteria decision making based on belief entropy. *IEEE Transactions on Fuzzy Systems*. 2019, 28(7): 1477-1491.

[12] Kwon N, Song K, Ahn Y, et al. Maintenance cost prediction for aging residential buildings

based on case-based reasoning and genetic algorithm. *Journal of Building Engineering*. 2020, 28: 101006.

[13] Zhang Z, Wang J, Wei X. A model for case retrieval based on ANN and nearest neighbor algorithm. 7th International Conference on Machine Learning and Cybernetics. 2008, 1(1): 142-147.

[14] Dong X, Lu M. Optimal road accident case retrieval algorithm based on k-nearest neighbor. Advances in Mechanical Engineering. 2019, 11(2): 1-7.

[15] Wang H, Sun B, Shen X. Hybrid similarity measure for retrieval in case-based reasoning systems and its applications for computer numerical control turret design. *Journal of engineering manufacture*. 2018, 232(5): 918-927.

[16] Duan Y, Lu J, Feng J, et al. Deep localized metric learning. *IEEE Transactions on Circuits & Systems for Video Technology*. 2018, 28(10):2644-2656.

[17] Bellet A, Habrard A, Sebban M. A survey on metric learning for feature vectors and structured data. *Computer Science*, 2013, arXiv:1306.6709.

[18] Kulis B. Metric learning: a survey. *Foundations and Trends in Machine Learning*, 2012, 5(4): 287-364.

[19] Kaya M, Bilge H S. Deep metric learning: a survey. *Symmetry*. 2019, 11(9): 1066.

[20] Bromley J, Bentz J W, Bottou L, et al. Signature verification using a siamese time delay neural network. *Pattern Recognit Artif Intell*. 1993, 7(4):737–744.

[21] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. *Advances in Neural Information Processing Systems, 31st Annual Conference on Neural Information Processing Systems, (NIPS 2017)*. Dec. 4-9, 2017, p 5999-6009, Long Beach, CA, United states.

[22] Rossi A, Hosseinzadeh M, Bianchini M, et al. Multi-modal siamese network for diagnostically similar lesion retrieval in prostate MRI. *IEEE Transactions on Medical Imaging*. 2021, 40(3): 986-995.

[23] Li Z, Lin H, Zheng W. Interactive self-attentive siamese network for biomedical sentence similarity. *IEEE Access*. 2020, vol. 8, pp. 84093-84104.

[24] Utkin L V, Ryabinin M A. A siamese deep forest. *Knowledge-Based Systems*. 2018, 139(1):13-22.

[25] Shao S, Mcaleer S, Yan R. Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics*, 2019, 15(4): 2446-2455.