Feature Extraction and Representation Learning via Deep Neural Network

T. Anuradha, Arun Tigadi, M. Ravikumar, Paparao Nalajala, S. Hemavathi, and Manoranjan Dash

Abstract Selection of a text characteristic is a prerequisite for text mining and information retrieval. Traditional techniques of feature extraction demand the use of custom features that must be made by hand. For new applications, deep learning allows the acquisition of new effective feature representations from training data rather than having to spend a lengthy time developing an effective feature by hand. Deep learning has made tremendous strides in text mining as a new feature extraction technique. This means that instead of using handmade features that strongly rely on designers' prior knowledge and cannot be fully utilised with big data, deep learning employs features from massive datasets. Massive datasets, containing millions of parameters, can be used to train deep learning models automatically to represent those datasets' features. This work initially describes the most prevalent text feature extraction approaches and then goes into greater depth on how deep learning is regularly utilised in text feature extraction, as well as how it can be used in future.

Keywords Deep learning · Text feature · Massive datasets · Feature representation

T. Anuradha (✉)
Department of Electrical and Electronics Engineering, KCG College of Technology, Chennai, India
e-mail: tanura1872@gmail.com

A. Tigadi
K.L. E Dr M.S.Sheshgiri College of Engineering and Technology, Constituent College of KLE Technological University, Hubballi, Belagavi Campus, Hubballi, India

M. Ravikumar
Department of Mechanical Engineering, New Horizon College of Engineering, Bangalore, India

P. Nalajala
Department of ECE, Institute of Aeronautical Engineering, Hyderabad, India

S. Hemavathi
Battery Division, Central Electrochemical Research Institute (CECRI), Chennai, India

M. Dash
Faculty of Management Sciences, Siksha O Anusandhan (Deemed to be University), Bhubaneswar, India

A. P. Pandian et al. (eds.), *Computer Networks, Big Data and IoT*, Lecture Notes on Data Engineering and Communications Technologies 117,
https://doi.org/10.1007/978-981-19-0898-9_44

1 Introduction

Depending on how the data is represented and interpreted, some information processing activities might be easy or difficult. Although 210 by 6 may be easily divided using the long division method, the situation becomes more problematic if the digits 210 and 6 are converted to roman numerals as CCX divide by VI. Most people will convert CCX to Arabic numeral form first and then utilise that to begin the division step. This broad approach can be applied to a wide range of fields, including daily life, computer science, and deep learning in particular [1]. In this sense, supervised learning-trained feedforward networks do representation learning. A linear classifier, such as SoftMax regression, is employed as the network's last layer [2]. The rest of the network teaches the classifier how to represent itself. Consequently, supervised training improves the classification process by giving the representation at each hidden layer new attributes. Classes that were not previously linearly separable, for example, could become so in the last hidden layer. Finally, a model like a nearest neighbour classifier might theoretically be used as the final layer. In the penultimate layer, depending on the type of final layer, different qualities should be learned by the features there. No explicit conditions are imposed on learned intermediate features during supervised training of feedforward networks. Algorithms for other types of representation learning are generally expressly created to mould the representation in a specific way. Take for instance learning a representation that simplifies density estimation. To make modelling easier, we can devise an objective function that promotes the representation vector's elements to be more self-sufficient. Unsupervised deep learning algorithms, like supervised networks, develop a representation as a by-product of their primary training objective [3]. It makes no difference how anything is represented when it comes to communication. You can learn more than one task simultaneously with a shared internal representation (some supervised, some unsupervised). Unsupervised and semi-supervised learning can both be accomplished via representation learning. We frequently have a lot of unlabelled data and a little amount of labelled data for training. Experiments on the labelled subset using supervised learning approaches frequently lead to overfitting. Unlabelled data can be used in semi-supervised learning to help alleviate the overfitting issue. If the unlabelled data is properly represented first, we can use it to solve the supervised learning problem [4].

2 Learning Representation

A new conference, ICLR1, has been created to focus on representation learning, which has now become a separate subject within the machine learning community. As we will see in the next section, learning a representation is an important part of the storey, so framing the problem as one of learning a representation makes sense, while representation learning research has exploded, an impressive run of empirical

achievements has occurred both in academia and in industry, fuelling the recent growth in the field [5]. Some of the high highlights will be highlighted shortly here.

- **Speech recognition and signal processing**: Speech recognition was one of the first applications of deep learning networks. There has been a recent reversal in neural networks, deep and representation learning, and this has had a significant impact on speech recognition, with many companies releasing new versions of their MAVIS speech systems based on deep learning. Word mistake rates were cut in half compared to a previous model using Gaussian mixtures and training on the same amount of data as today's state-of-the-art model. The state of the art in polyphonic transcription has been significantly surpassed by representation learning algorithms in music, with relative error improvements ranging from 5 to 30% on a common benchmark of four datasets.
- **Object recognition:** This dataset had a 1.4% error rate while using support vector machines (SVMs). This dataset was used to kick-start deep learning in 2006. There is now a 0.81% inaccuracy in MNIST's knowledge-free version, which is state of the art in terms of unconstrained challenges (e.g. employing a convolutional architecture). Deep networks continue to hold the most recent records in this field. Recently, deep learning has progressed beyond just recognising digits to recognising objects in photographs, with the most recent accomplishment occurring on the ImageNet dataset, where the error rate was decreased from 26.1 to 15.3%.
- **NLP:** In addition to speech recognition, representation learning has numerous other uses in natural language processing (NLP). Word embeddings are distributed representations that are learned for each word. It was only by including a convolutional architecture and semantic role labelling that the SENNA system was established, which shares representations across a variety of tasks like as language modelling and part-of-speech tagging. SENNA is as good as or better than current state-of-the-art predictors, yet it is simpler and faster. When learning word embeddings and visual representations, it is possible to link the two together. As a result of using enormous volumes of data to map photographs and searches in the same area, in a short period of time, Google's picture search has grown substantially. This new neural net language model could outperform current state of the art in terms of both perplexity and speech recognition word error rate, decreasing i. Perplexity and BLEU scores have been boosted using statistical machine translation models similar to these (SMT). It was found that researchers could beat the present state of the art in whole sentence paraphrase identification by nearly doubling the F1 score using recursive autoencoders. On improve word sense disambiguation accuracy, representation learning can be applied to a subset of Senseval-3 and see an increase in accuracy from 67 to 70%.
- **Multi-task and transfer learning, domain adaptation:** When a learning algorithm can use commonalities across distinct learning tasks to share statistical strength and transfer knowledge between them, this is referred to as "transfer learning" (Fig. 1).

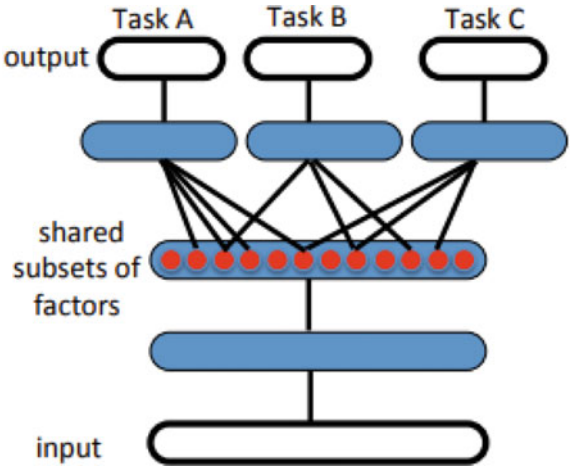

Fig. 1 Explaining representation learning

2.1 Advantages of Representations

- ***Distributed representations***

As long as a learnt representation is large enough, it can capture many different possible input combinations. To represent huge input regions with a limited number of parameters, there are a number of strategies including RBMs, sparing coding, and neural networks with several layers (where k denotes the number of nonzero components in a sparse representation and N denotes the number of zero components). All of these representations are dispersed or sparse. As a result of multi-clustering, the generalisation of clustering to distributed representations can be thought of as a form of object recognition that uses a histogram of cluster categories to find similar objects across different patches of an image [6]. This is an extremely popular method for extracting hierarchical features for object recognition. For example in sparse code or with a restricted Boltzmann machine, each parameter can be reused in many other cases that are not merely next to each other. However, with local generalisation, separate parts of input space are basically associated with their own unique set of parameters. The number of hidden units or features in a distributed representation can be activated by a single input, and this number grows exponentially over time. In a single-layer model, an input hyperplane corresponds to a code or representation for each feature, and the pattern of activation for that input corresponds precisely to the code or representation for that input [7]. The most common clustering algorithm does not employ a non-distributed representation, such as k-means, which uses a one-hot code to decide which of several cluster centroids best reflects an input vector.

- ***Depth and abstraction***

In this study, we discuss representation learning strategies in depth, which is an important consideration. As we will see, deep architectures are notoriously difficult

to train properly, despite recent advances in the field. Despite these difficulties, deep architectures provide two key advantages that keep us interested in finding new training procedures.

- ***Disentangling factors of variation***

We want our representations to be distributed and invariant, but we also want them to separate the variables that cause variation. The input distribution tends to change independently of other explanatory components when studying a sequence of real-world inputs.

Many sources interact richly to provide complex data. Object classification, for example, might be made more difficult by the interactions of these variables. Object shapes and material properties interact to create a picture, for instance. All of these elements come together to form an image. Complex patterns can be produced when shadows from items in a scene fall on top of one another, giving the impression of object boundaries when there are not any it is our belief that to overcome these problems the method we use will have to rely on a large number of unlabelled samples to create models that distinguish among the various explanatory sources [8]. As a result, a representation for AI-related tasks should be significantly more resilient to the complex and richly structured variations that can be found in natural data sources.

- ***Good criteria for learning representations***

The production of a clear training objective or target is one of the challenges of representation learning. Other machine learning activities such as categorisation cannot do this, thus, it is distinct from them. In dealing with classification, it is obvious (at least conceptually) that we want to limit the amount of wrong classifications. Representation learning has no connection to the final goal of learning a classifier or other predictor, which is frequently the case. This is similar to the credit assignment problem that can be seen in reinforcement learning programmes [9]. A good representation, according to our theory, separates out the fundamental causes of variation, but how can we put it into practise? We can incorporate priors like those described above (potentially data-dependent ones) that assist the representation better do this disentangling, even if we do not optimise the likelihood under a decent model.

Deep representations

Feature learning and deep learning saw a breakthrough in 2006, thanks to Geoff Hinton, Lee, and a slew of other researchers. Gluttonous layer-by-layer unsupervised pre-training was a key concept. In order to train a feature hierarchy, it was necessary to combine previously learned transformations with unsupervised feature learning iterations, each of which contributed weight to the deep neural network one step below. Lastly, a neural network classifier or a deep Boltzmann machine might be created by combining the layers to create a deep supervised predictor.

There was still a significant difference between the unsupervised pre-training results and the results obtained with no pre-training. You could use a previous layer's

results as new inputs for a subsequent layer (on top of the raw input). Iterative pre-training is another option, which involves pre-training all previously added layers in a supervised manner at each stage of the iteration.

Using an unsupervised model, integrating pre-trained layers from unsupervised learning is less evident than combining single layers to produce an improved model [10]. First, pre-trained RBMs in DBNs were presented as the top layer of a DBN, with the lower layers being read as a directed sigmoid belief network and the lower layers as an RBM This generative model could be improved further, but it is unclear how.

3 Text Feature Extraction Methods

Extracting text features is critical since it has a direct impact on text classification accuracy. A sentence is seen as a dot in N-dimensional space in vector space model (VSM). Each dot's datum dimension indicates a different (digitised) text characteristic. In addition, keyword sets are frequently used in text features. Meaning that a set of predetermined keywords is used to compute the weights of the textual terms, and a digital vector is then formed, which is the text's feature vector. Methods for extracting text features that are already available include those described below, such as text filtration, fusion, mapping, and clustering.

A. ***Filtering method***

Fast and efficient filtering is the best method for extracting text features on a big scale. Word frequency, information gain, and a mutual information strategy are among the text feature extraction filtering strategies used.

- ***Word frequency***: To measure a word's frequency, you count how many times it appears in a passage of text. Using word frequency to pick features reduces the dimensionality of feature space by excluding words with frequencies below a predetermined threshold. Words with low frequency have little effect on filtration, which is why this strategy is based on it. Information retrieval researchers, on the other hand, believe that words with a lower frequency of occurrences can occasionally hold more information. As a result, in the feature selection process, deleting large numbers of terms based only on their frequency is improper.
- ***Mutual information***: Computational linguistics models often use the mutual information (MI) method for measuring the mutuality of two objects. It is used in filtration to check for feature distinction across different themes. Mutual information and cross-entropy have similar definitions. It is usual practise to count the number of mutual terms shared by a feature word and a class in order to estimate the amount of mutual information the two have. Nothing is presupposed about the link between feature words and classes in this strategy. Hence, it is ideal for registering text classification features and class descriptions in data bases.

- ***Information gain***: There are many machine learning techniques that use information gain (IG) [11]. To determine how much of the topic's projected material is actually included in the text, look for a well-known feature inside a text on the subject. Computing information gain allows researchers to identify qualities that are more common in positive samples than negative ones. There are numerous mathematical ideas and sophisticated theories and formulas involving entropy involved in the evaluation approach known as information gain. The quantity of information a feature item can supply without considering the entropy of any other features, but the difference in entropy values between features is described as the item's ability to provide overall categorisation information. Items with little information gain are deleted, and the rest are sorted in descending order using the information received from each feature item. This is done using training data.

B. ***Fusion method***

Fusion necessitates the use of specialised classifiers, and the search must be undertaken with a time interval that increases exponentially [12]. There is a lot of variability in terms of timing. As a result, it should not be used to extract features from big texts. Fusion techniques that use the weighting method fall under a distinct category. It assigns a weight (0, 1) to each feature so users can practise using it while making tweaks. The linear classifiers' weighting mechanism is quite efficient. Example-based learning methods like the K-closest neighbours (KNN) algorithm.

- ***Weighted K-nearest neighbours (KNN)***: As part of the KNN classifier weighted feature extraction challenge, Han applied several of his earlier ideas. For each categorisation of continuous cumulative data, the approach has a strong classification influence. KNN's absence of parameters and statistical pattern recognition-based text categorisation capacity may lead to higher accuracy and recall rates in classification.
- ***The centre vector weighted method***: It is suggested by Shankar that a weighted centre vector classification approach be used, which first establishes a method of characterising abilities to distinguish between right and wrong and then generates a new centre vector. Algorithm requires numerous weighted techniques

C. ***Mapping method***

Text classification has used mapping frequently and successfully. In latent semantic index (LSI) and PCA, it is commonly employed.

- ***Latent semantic analysis:*** It is a theory or method of computation used to acquire and demonstrate knowledge [13]. There is no link between words and texts because of the statistical computation approach used to evaluate a large number of text sets. This statistical computation method then utilises the latent semantic structure extracted from the text sets to represent the words and texts. By mapping texts from high-dimensional VSM to lower-level latent semantic space, the basic premise of latent semantic analysis is established. ***Least squares mapping method:*** Jeno studied high-dimensional data reduction from the centre vector and least squares perspectives.

D. ***Clustering method***

Text feature comparison is taken into account in the clustering process because it is critical to text feature comparison. As a result, the core of each class is used to replace the class's individual features. As a result of its low compression ratio and steady categorisation accuracy, this approach has several advantages. It has the drawback of being exceedingly time-consuming.

- ***CHI (chi-square) clustering method***: Instead of the usual algorithm's pattern of each word having a corresponding one-dimension, CHI clustering computes the contribution of each feature word to each class and groups those words together. This approach has the advantage of being quite simple to implement.
- ***Concept indexing***: Text categorisation uses a basic but effective dimensionality reduction technique called concept indexing (CI). A basic vector structure subspace (CI subspace) is utilised for each class's centre, and each text vector is mapped to that subspace to represent it. When there is more classification contained in training sets than in the text vector space, it reduces vector space dimensionality because the CI subspace is smaller in dimensionality [14]. Text vector mapping can be viewed as an indexing procedure in this concept space, with each class centre serving as a generalisation of text contexts within that classification.

4 Deep Learning Approach

Hinton et al. introduced a new category of unsupervised learning in 2006 called "deep learning". Studies on artificial neural networks inspired the idea for this project. A deep learning structure is a multi-layer perceptron with many implicit layers. Dispersed feature representation can be found via deep learning, which uses lower-level attributes to create higher-level property categories or features.

In contrast to surface-based algorithms, which have a number of advantages, deep learning algorithms have numerous disadvantages, such as a limited capacity to generalise for challenging classification tasks when using few samples of complex function. If the input data is characterised according to their distribution, then the implementation of complex function approximation is called deep learning. However, while dealing with samples, the essence of each dataset's feature is rarely studied. Instead of using handmade features, deep learning automatically learns new features from huge data, which makes it superior to standard pattern recognition approaches. Only one well-known good feature has developed in the history of computer vision progress in the last five to ten years. Semantic parsing, retrieval, semantic role labelling, sentimental analysis, question answering, machine translation (including named entity recognition), text classification (including summarisation), and text generation are all common natural language processing (NLP) tasks in which deep learning technology is used. Two prominent models used in this study are the convolution neural network and the recurrent neural network.

Following that, a number of text feature extraction methods, enhancement methods, and stages are discussed.

A. ***Autoencoder***

Using Rumelhart et al. autoencoder's as a feedforward network, researchers were able to learn a compressed and distributed representation of data for the first time. The input and output of an autoencoder are often separated by a secret layer. The hidden layer's representation is smaller than either the input or output layers' because it contains fewer units. It is possible to train an autoencoder without supervision by feeding it the same input data over and over again until you get the desired results. The training procedure is identical to that of a typical neural network with backpropagation, with the exception of the error, which is calculated by comparing the output to the input data. The deep counterpart of an autoencoder is a stacked autoencoder, which is constructed by stacking layers on top of each other. There are several layers in a neural network, and each one takes in the previously learnt representation as input and output. Gravelines et al. discussed a stacked sparse autoencoder, which is an autoencoder with regularisations for sparsity to learn a sparse representation.

B. ***Restricted Boltzmann machine***

Restricted Boltzmann machine (RBM) is a Boltzmann machine invented by Smolensky that has no connections between any of its visible or hidden units. It was originally known as harmonium. Visible units (i.e. data samples) are part of this network, as are some concealed ones (correspondingly visible vectors) (correspondingly hidden vectors). Binary velocities, such as the visible vector and the hidden vector, have states of 0 or 1. A bipartite graph represents the entire system. When comparing visible and hidden units, edges are only found between them; otherwise, no edge connections exist (Fig. 2).

C. ***Deep belief network***

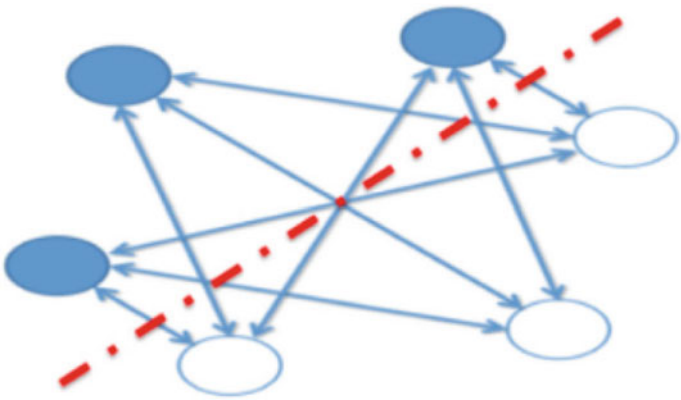

Fig. 2 RBM structure

Since greedy RBMs may be taught, Hinton et al. proposed DBNs. In DBN's network topology, there is a layer on top of the layers that contains one of the constrained Boltzmann machines.

DBN's training paradigm is broken down into two stages: implementation and evaluation.

- RBM networks should be trained independently and without supervision for each layer to ensure that feature information is preserved as vectors are mapped to various feature spaces.
- Input feature vectors from RBM are used as input feature vectors in the BP network, which then trains an entity relationship classifier under supervision using the output feature vectors from RBM. Layer-specific RBM networks may only optimise the weights of the feature vectors in their own layer, not the feature vectors of the entire DBN. The entire DBN network is fine-tuned by an RBM backpropagation network, which sends error information to each tier of RBM.

In deep learning terminology, step one is referred to as pre-training, while step two is referred to as fine-tuning. In the supervised learning layer, any classifiers based on a given application domain can be employed. BP networks are required to be used (Fig. 3).

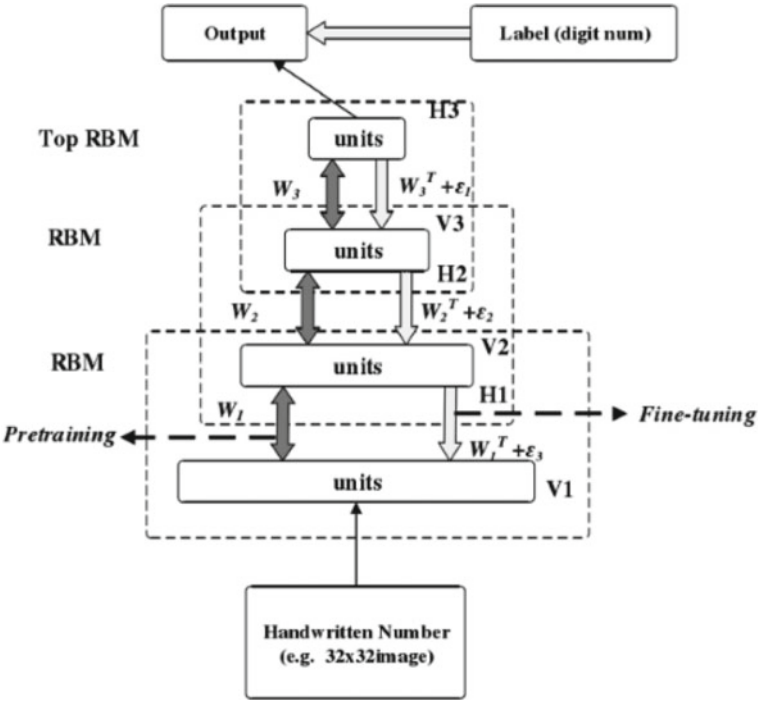

Fig. 3 Structure of DBN

D. *Convolutional neural network*

As a new and extremely effective identification approach emerges, convolution neural network (CNN) has attracted considerable attention from researchers. Hubel and Wiesel proposed the idea of a receptive field in the 1960s based on their study of the visual cortex cells of cats. Fukushima was moved to provide neuropsychological ideas in the first deployment of the CNN network, and he also believed that a wild notion was first implemented in the artificial neural network sector. When it comes to pattern recognition, LeCun et al. found that the error gradient algorithm training in the convolutional neural network produced the best results when compared to other methods.

CNN is a type of artificial neural network because of its versatility and ability to extract local features from large amounts of input. Using shared network structure weights makes it more like biological neural networks, reducing network complexity by reducing weight numbers and allowing CNN to be employed in a variety of pattern recognition applications with excellent results. Shared network structure weights. The results were excellent. Combining local perception area with CNN ensures displacement invariability by sharing the weight and dropping the sample in space or time. This allows the data to be used to its greatest potential. There are other more applications for CNN that have been discovered over the course of many years of research, including as the identification of faces, documents, speech, and licence plates. By using permutation encoding technique, Kussul could identify faces, recognise handwriting digitally, and recognise objects with a certain level of performance in 2006.

E. *Recurrent neural network*

Sequential data is processed using RNNs. There are three layers in a typical neural network model: input layers, hidden layers, and output layers. The nodes in these layers are all disconnected from one another. For occupations requiring sequential inputs, such as voice and language, RNNs (Fig. 4) are usually preferred. They kept a "state vector" in their hidden units when working with input sequences, containing information about the prior history of each preceding component. Hidden unit outputs

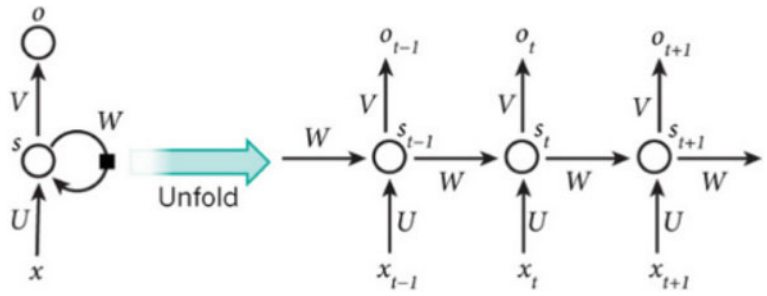

Fig. 4 RNN circuit

must be trained with backpropagation because they are analogous to the outputs of individual neurons in a deep multi-layer network.

Because the backpropagated gradients either rise or decrease at each step, it is been difficult to train RNNs because their dynamic capabilities often explode or vanish.

Hidden units with values st in prior time steps are fed into the artificial neurons (shown on the left by the black square, which represents a one-step delay in time). An input sequence of elements (xt) can be converted into an output sequence of elements, where the components in each ot rely on the input sequence (for t′t), using this technique. The number of time steps is split by three, and the same set of parameters are used in each time step. A number of new RNNs have been developed, including the simple RNN (SRN), the bidirectional RNN, the deep (bidirectional) RNN, and the echo state network.

5 Conclusion

For text mining and information retrieval, the selection of a text feature item is a prerequisite step. If an extract metric, such as a reduction in the dimension of feature vector spaces, is met, it is applicable to initial feature subsets from test sets. Uncorrelated or unnecessary features will be removed during feature extraction. Feature extraction, as a preprocessing strategy for the learning algorithm, can enhance the learning algorithm's accuracy while decreasing training time. If you compare deep learning to other machine learning methods, you will find that the former can detect more complex interactions between features, learn lower-level features from nearly unprocessed original data, and mine characteristics that are difficult to detect. Although the recurrent neural network (RNN) has been widely employed in natural language processing (NLP), it is rarely used in text feature extraction for the simple reason that RNN focuses on time-sequenced input. The generative adversarial network model, first introduced in 2014 by Ian J. Goodfellow, has also achieved noteworthy accomplishments within the deep learning generative model field in under two years. It presents a novel frame that may be utilised to estimate and build an opponent process model compared to previous algorithms. This is a significant advancement in the field of unsupervised representation learning. It is now primarily used to create natural-looking photographs. However, in terms of text feature extraction, it has made little progress. Deep learning has some problems. In order to support supervised perception as well as reinforcement learning, significant volumes of data are required. Our dataset on diabetes now has data from 302 hospitals, and this data will let us employ deep learning in text feature extraction to better deal with medical issues. And, they are terrible at advanced plans, able to do nothing but very basic pattern discrimination tasks. Having unreliable, inaccurate, and unjust data necessitates further development in future. As a result of text feature

extraction's inherent properties, each approach has both advantages and insurmountable limitations. If at all possible, use a variety of extraction methods to get at the same piece of data.

References

1. Peng, S., Sun, S., Yao, Y.-D.: A survey of modulation classification using deep learning: signal representation and data preprocessing. IEEE Trans. Neural Netw. Learn. Syst. https://doi.org/10.1109/TNNLS.2021.3085433
2. Jia, X., et al.: Semi-supervised multi-view deep discriminant representation learning. IEEE Trans. Pattern Anal. Mach. Intell. **43**(7), 2496–2509 (2021). https://doi.org/10.1109/TPAMI.2020.2973634
3. Jiang, W., et al.: Statistical feature extraction and hybrid feature selection for material removal rate prediction in chemical mechanical planarization process. In: 2021 5th IEEE Electron Devices Technology & Manufacturing Conference (EDTM), 2021, pp. 1–3. https://doi.org/10.1109/EDTM50988.2021.9421002
4. Hao, Q., Li, S., Fang, L., Kang, X.: Multiscale feature extraction with Gaussian curvature filter for hyperspectral ımage classification. In: IGARSS 2020—2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 80–83. https://doi.org/10.1109/IGARSS39084.2020.9323640
5. Guo, H., Liu, Y., Zhao, J., Yang, D.: Research on feature extraction of Tai Le recognition. In: 2020 IEEE 3rd International Conference on Computer and Communication Engineering Technology (CCET), 2020, pp. 95–98. https://doi.org/10.1109/CCET50901.2020.9213168
6. Zhao, G., Li, T., Yang, Z.: An extended knowledge representation learning approach for context-based traceability link recovery: extended abstract. In: 2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE), 2020, p. 22. https://doi.org/10.1109/AIRE51212.2020.00010
7. Wu, C., Qi, G., Zhao, H., Chen, Z.: Feature extraction of cultural gene ımage based on PCA method. In: 2020 International Conference on Computer Engineering and Application (ICCEA), 2020, pp. 860–863. https://doi.org/10.1109/ICCEA50009.2020.00189
8. Joy, A.A., Hasan, M.A.M.: A hybrid approach of feature selection and feature extraction for hyperspectral ımage classification. In: 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), 2019, pp. 1–4. https://doi.org/10.1109/IC4ME247184.2019.9036617
9. Yang, L., Zhang, J., Yang, Y.: A feature extraction technique in stereo matching network. In: 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2019, pp. 393–396. https://doi.org/10.1109/IAEAC47372.2019.8998024
10. Mestri, R., Limaye, P., Khuteta, S., Bansode, M.: Analysis of feature extraction and classification models for lip-reading. In: 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), 2019, pp. 911–915. https://doi.org/10.1109/ICOEI.2019.8862649
11. Zhang, S., Zhai, J., Xie, B., Zhan, Y., Wang, X.: Multimodal representation learning: advances, trends and challenges. In: 2019 International Conference on Machine Learning and Cybernetics (ICMLC), 2019, pp. 1–6. https://doi.org/10.1109/ICMLC48188.2019.8949228
12. Lu, M., Li, F.: Survey on lie group machine learning. Big Data Min. Anal. **3**(4), 235–258 (2020). https://doi.org/10.26599/BDMA.2020.9020011

13. Chen, J., Gong, Z., Wang, W., Liu, W., Dong, X.: CRL: collaborative representation learning by coordinating topic modeling and network embeddings. IEEE Trans. Neural Netw. Learn. Syst. https://doi.org/10.1109/TNNLS.2021.3054422
14. Sharma, V., Tapaswi, M., Sarfraz, M.S., Stiefelhagen, R.: Clustering based contrastive learning for ımproving face representations. In: 2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020), 2020, pp. 109–116. https://doi.org/10.1109/FG47880.2020.00011