Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 31 (2015) 29 – 34

www.elsevier.com/locate/procedia

15th CIRP Conference on Modelling of Machining Operations

Analysis of Feature Extracting Ability for Cutting State Monitoring Using Deep Belief Networks

Yang Fu [a], Yun Zhang*[a], Haiyu Qiao [a], Dequn Li [a], Huamin Zhou [a], Jürgen Leopold [b]

[a] State Key Laboratory of Material Processing and Die & Mold Technology, Huazhong University of Science and Technology, Wuhan, 430074, P. R. China.
[b] Formerly Fraunhofer Institute for Machine Tools and Forming Technology, Chemnitz, 09661, Germany

* Corresponding author. Tel.: +86-027-87543492; fax: +86-027-87554405. *E-mail address:* marblezy@163.com

Abstract

Information extracting method from numerous measured signals is a critical technique for intelligent manufacturing application to further reduce the manpower cost and improve the productivity and workpiece quality. Manually defining signal features, as the common way, unfortunately will lose most of the information and the performance can't be guaranteed. In the past few years, machine learning method with deep structure has been the most promising automatic feature extracting method which has made great breakthrough in computer vision and automatic speech recognition. In this paper, deep belief networks are employed using vibration signal obtained from end milling to build feature space for cutting states monitoring. Greedy layer-wise strategy is adopted to pre-train the network and standard samples are used for fine-tuning by applying back-propagation method. Comparisons are made with several manually defined features both in time and frequency domain, like MFCC and wavelet method. Different modeling methods are also employed in the research for comparisons. Results show that the deep learning method has similar ability to characterize the signal for cutting states monitoring compared to those manually defined features. And the modeling accuracy is much better than other traditional modeling methods. Furthermore, benefitting from the potential capability in information fusion, deep learning method would be a promising solution for more complex applications, like tool wear monitoring, machining surface prediction et al.

Peer-review under responsibility of the International Scientific Committee of the "15th Conference on Modelling of Machining Operations

Keywords: Intelligent manufacturing; Cutting states monitoring; Deep belief networks; Feature extraction.

1. Introduction

Intellectualization is a development trend of machining industry [1-4]. Owing to rapid changing in market and growing of manpower cost, it is in urgent need of more intelligent, optimized and self-adaptive manufacturing solutions [3, 5], especially for complicated applications, in which the theoretical analysis method could not provide practical suggestions. More and more signals, which can currently be obtained from an actual machining process, with the advancement of sensor technology, provide an opportunity to make breakthrough in machining Intellectualization [6, 7].

A critical technique for intelligent manufacturing is extracting features from numerous measured signals. A common way to do that is manually construction. Samanta et al. employed statistic parameters of the vibration signal, root mean square, variance, skewness, kurtosis, etc., in the artificial neural network for fault diagnostics of rolling bearings [8]. Yao et al. extracted standard deviation and energy ratio of the decomposed sub-signals applying wavelet decomposition to indicate the chatter phenomenon [9]. Teti et al. summarized the features used in time and frequency/time-frequency domain for various machining signals [7]. However, manual construction of features is usually a way solving a certain problem, and the problem-unconcerned information might be lost. For different applications, different features must be built. This task is quite time-consuming, and becomes even difficult when the subject contains too many coupled components [10]. In addition, the performance cannot be guaranteed.

Deep learning or called deep encoder [11], which simulates the hierarchical way that the brain processes the information [12], is a newly developed machine learning method which employs deep structure to model the data distribution and inner structure [13, 14]. Different from the traditional learning

Peer-review under responsibility of the International Scientific Committee of the "15th Conference on Modelling of Machining Operations
doi:10.1016/j.procir.2015.03.016

methods, deep learning method introduces hierarchical structure to firstly extract features from low level to high level and after proper fine-tuning, the highest level features will be input to a classifier or a regression machine to construct the relationship. In the past decade, deep learning method has made great success in computer vision [15] and automatic speech recognition [16], and has been regarded as be the most promising automatic feature detecting method [13].Recently deep learning has been introduced to some engineering application. Van et al. employed deep belief networks to classify the faults of compressor valves [5]. Tamilselvan used deep belief learning method in failure diagnosis to detect the health state of the power transformer [17]. They still regard deep learning as a traditional machine learning method and only focus on the accuracy with several low dimensional manually defined features.

This study is inspired by the success of the deep learning method applied in automatic speech recognition. The vibration signal is similar to the voice, which just comes from different speakers and differs in frequency and amplitude distribution, which makes us to believe that the deep learning can also achieve wonderful performance as it does in the voice. Deep belief network (DBN), which is a major kind of deep learning method, is introduced to construct a cutting states classifier using vibration signals. The capability of deep belief networks for automatic feature extracting is discussed.

2. Methodology

2.1. A brief introduction to deep belief networks

Deep belief networks are a generative model constructed by stacking a number of restricted Boltzmann machines (RBMs), as illustrated in Fig. 1. A common recognition model is usually composed of three components: collecting observing signals, extracting features and building relationships. The three components all need a lot of manual efforts. Deep belief networks provide a framework to build model directly from what we observe to what we want to know. The layer by layer structure is a kind of hierarchical feature representation. The network training process is self-adaptive and can replace the brainwork-consuming feature extracting component. The training procedure is conducted layer by layer using massive unlabeled samples and after the preparation, a much smaller size of labelled sample set is used to fine-tune the whole network using back-propagation (BP) algorithm [11].

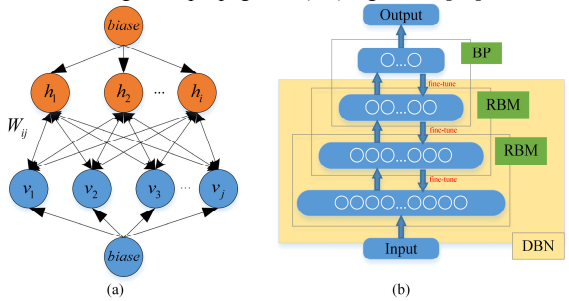

Fig. 1 Architecture of RBM (a) and DBN (b).

RBM is a generative stochastic artificial neural network based on statistical mechanics，which can learn a probability distribution over the training set [18]. It is composed of two layers of binary stochastic units, a visible layer and a hidden layer. Regarding it as an undirected graphical model, all visible units are connected to all hidden units, and there are no connections within each layer, as illustrated in Fig. 1(a). The model parameters are visible units biases $\mathbf{b}$, hidden units biases $\mathbf{c}$ and connection weights $\mathbf{W}$.

The theoretical derivation of RBM starts from the definition of network energy for a certain network state, which defines a probability distribution over the joint state of the visible units and the hidden units, expressed as

$$E(\mathbf{v},\mathbf{h}) = -\sum_{j=1}^{nV} b_j v_j - \sum_{i=1}^{nH}\sum_{j=1}^{nV} h_i W_{ij} v_j - \sum_{i=1}^{nH} c_i h_i \tag{1}$$

where v_i and h_i are the binary states of the visible unit i and the hidden unit j. nV and nH respectively represent the number of visible units and hidden units. The joint distribution over the visible and hidden units can be defined as:

$$P(\mathbf{v},\mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v},\mathbf{h})} \tag{2}$$

where $Z = \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}$, is called partition function. And the associated two margin distributions are:

$$P(\mathbf{v}) = \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}}{Z} \quad P(\mathbf{h}) = \frac{\sum_{\mathbf{v}} e^{-E(\mathbf{v},\mathbf{h})}}{Z} \tag{3}$$

The conditional probability of the hidden units $\mathbf{h}$ over the visible units $\mathbf{v}$ can be given as:

$$\begin{aligned} P(\mathbf{h}\mid\mathbf{v}) &= \frac{P(\mathbf{v},\mathbf{h})}{P(\mathbf{h})} = \frac{e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\tilde{\mathbf{h}}} e^{-E(\mathbf{v},\tilde{\mathbf{h}})}} \\ &= \frac{\exp\left(\sum_{j=1}^{nV} b_j v_j + \sum_{i=1}^{nH}\sum_{j=1}^{nV} h_i W_{ij} v_j + \sum_{i=1}^{nH} c_i h_i\right)}{\sum_{\tilde{\mathbf{h}}} \exp\left(\sum_{j=1}^{nV} b_j v_j + \sum_{i=1}^{nH}\sum_{j=1}^{nV} \tilde{h}_i W_{ij} v_j + \sum_{i=1}^{nH} c_i \tilde{h}_i\right)} \\ &= \frac{\exp\sum_{i=1}^{nH}\left(\sum_{j=1}^{nV} h_i W_{ij} v_j + c_i h_i\right)}{\sum_{\tilde{\mathbf{h}}} \exp\sum_{i=1}^{nH}\left(\sum_{j=1}^{nV} \tilde{h}_i W_{ij} v_j + c_i \tilde{h}_i\right)} \\ &= \prod_{i=1}^{nH} \frac{\exp\left(\sum_{j=1}^{nV} h_i W_{ij} v_j + c_i h_i\right)}{\sum_{\tilde{\mathbf{h}}} \exp\left(\sum_{j=1}^{nV} \tilde{h}_i W_{ij} v_j + c_i \tilde{h}_i\right)} \end{aligned} \tag{4}$$

Similar derivation can be done to $P(\mathbf{v}\mid\mathbf{h})$ as

$$P(\mathbf{v}\mid\mathbf{h}) = \prod_{j=1}^{nV} \frac{\exp\left(b_j v_j + \sum_{i=1}^{nH} h_i W_{ij} v_j\right)}{\sum_{\tilde{\mathbf{v}}} \exp\left(b_j \tilde{v}_j + \sum_{i=1}^{nH} h_i W_{ij} \tilde{v}_j\right)} \tag{5}$$

As there are no hidden-hidden or visible-visible connections, the units in one layer are conditionally independent when the other layer is given, so we can obtain

$$P(h_i = 1\mid\mathbf{v}) = \frac{1}{1+\exp\left(-\sum_{j=1}^{nV} W_{ij} v_j - c_i\right)} \tag{6}$$

$$P(v_j = 1\mid\mathbf{h}) = \frac{1}{1+\exp\left(-b_j - \sum_{i=1}^{nH} h_i W_{ij}\right)} \tag{7}$$

In order to train the RBM, maximum likelihood estimation is a good way. Given a training set, the log likelihood of the model for a single training sample is

$$L(\boldsymbol{\theta}) = \log P(\mathbf{v} \mid \boldsymbol{\theta}) = \log \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} - \log \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \tag{8}$$

where $\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ is the parameters to be estimated. The gradient can be given as:

$$\begin{aligned} \frac{\partial L}{\partial \theta} &= \frac{\partial}{\partial \theta}\left(\log \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} - \log \sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})} \right) \\ &= \sum_{\mathbf{h}} \frac{e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}} \left(-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right) - \sum_{\mathbf{v},\mathbf{h}} \frac{e^{-E(\mathbf{v},\mathbf{h})}}{\sum_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}} \left(-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right) \\ &= \sum_{\mathbf{h}} P(\mathbf{h} \mid \mathbf{v}) \left(-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right) - \sum_{\mathbf{v},\mathbf{h}} P(\mathbf{v},\mathbf{h}) \left(-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right) \end{aligned} \tag{9}$$

In order to simplify the equation, two symbols are introduced as follow:

$$\begin{aligned} \langle \theta \rangle_{\text{data}} &= \sum_{\mathbf{h}} P(\mathbf{h} \mid \mathbf{v}) \left(-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right) \\ \langle \theta \rangle_{\text{model}} &= \sum_{\mathbf{v},\mathbf{h}} P(\mathbf{v},\mathbf{h}) \left(-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right) \end{aligned} \tag{10}$$

The partial derivative of energy function to model parameters is summarized as follow:

$$-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial W_{ij}} = h_i v_j \quad -\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial b_j} = v_j \quad -\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial c_i} = h_i \tag{11}$$

The first item $\langle \theta \rangle_{\text{data}}$ can be calculated easily while the second one $\langle \theta \rangle_{\text{model}}$ needs to traverse all the possible value combinations of the visible units and hidden units which is a NP-hard problem. In order to solve this problem, Prof. Hinton proposed contrastive divergence (CD) algorithm [19] in which the $\langle \theta \rangle_{\text{model}}$ is obtained using Gibbs sampling method, which is based on the Markov Chain Monte Carlo (MCMC) strategy. The Gibbs sampling starts with a train sample, and alternately samples the hidden units and visible units using equation (6) and (7) by k steps, as illustrated below:

$$\begin{aligned} &\mathbf{v}^{(0)} = t, && \mathbf{h}^{(0)} \sim P\left(\mathbf{h} \mid \mathbf{v}^{(0)}\right) \\ &\mathbf{v}^{(1)} \sim P\left(\mathbf{v} \mid \mathbf{h}^{(0)}\right), && \mathbf{h}^{(1)} \sim P\left(\mathbf{h} \mid \mathbf{v}^{(1)}\right) \\ &\cdots\cdots \\ &\mathbf{v}^{(k)} \sim P\left(\mathbf{v} \mid \mathbf{h}^{(k-1)}\right), && \mathbf{h}^{(k)} \sim P\left(\mathbf{h} \mid \mathbf{v}^{(k)}\right) \end{aligned} \tag{12}$$

When $k \to \infty$, the accurate model distribution can be obtained and $\langle \theta \rangle_{\text{model}}$ can be calculated. In practice, Pro. Hinton pointed out that the CD learning with $k = 1$ can provide adequate results to properly estimate the model gradient. Therefor the second term $\langle \theta \rangle_{\text{model}}$ can be estimated using Gibbs sampling as

$$\begin{aligned} \langle \theta \rangle_{\text{model}} &= \sum_{\mathbf{v},\mathbf{h}} P(\mathbf{v},\mathbf{h}) \left(-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right) \\ &= \sum_{\mathbf{v}} P(\mathbf{v}) \sum_{\mathbf{h}} P(\mathbf{h} \mid \mathbf{v}) \left(-\frac{\partial E(\mathbf{v},\mathbf{h})}{\partial \theta} \right) \\ &= \frac{1}{l} \sum_{l} \sum_{\mathbf{h}} P(\mathbf{h} \mid \mathbf{v}) \left(-\frac{\partial E\left(\mathbf{v}^{(l)},\mathbf{h}^{(l)}\right)}{\partial \theta} \right) \end{aligned} \tag{13}$$

where l is the number of samples used to estimate the model represented distribution in the Gibbs sampling.

In practical application, the training data is divided into mini-batches to enhance the computing efficiency. And a common strategy is set l equal to the size of the mini-batch. The total gradient is also divided into "mini-batch" by the size of the data mini-batch to avoid changing the learning rate when the size of a mini-batch changes. Therefore the updating rules of the parameters using stochastic gradient descent algorithm can be given as:

$$\theta := \theta + \varepsilon \Delta\theta = \theta + \varepsilon \left(\langle \theta \rangle_{\text{data}} - \langle \theta \rangle_{\text{model}} \right) \tag{14}$$

where ε is the learning rate. The gradient for a mini-batch with size l can be expanded as:

$$\begin{aligned} \Delta W_{ij} &= \frac{\sum_{s=1}^{l} \left(h_{(s),i}^{(0)} v_{(s),j}^{(0)} - h_{(s),i}^{(k)} v_{(s),j}^{(k)} \right)}{l} \\ \Delta b_j &= \frac{\sum_{s=1}^{l} \left(v_{(s),j}^{(0)} - v_{(s),j}^{(k)} \right)}{l} \\ \Delta c_i &= \frac{\sum_{s=1}^{l} \left(h_{(s),i}^{(0)} - h_{(s),i}^{(k)} \right)}{l} \end{aligned} \tag{15}$$

where the notation $(\bullet)_{(s),i}^{(k)}$ represents the parameter $(\bullet)$ of the s-th training sample's i-th element, and k implies the sample obtained after k-step Gibbs sampling.

The whole structure is trained greedily layer by layer using unlabelled training data on a series RBM units, and after all RBMs are well trained, their parameters are then unfolded to the DBN network, and back-propagation algorithm is performed to fine-tune the whole network using a much small set of labelled data.

2.2. Experimental setup and test configurations

In order to obtain vibration signals in different cutting states, end milling experiments were conducted. Accelerometer (PCB 356A15 3D 2-5k HZ ±5%) was mounted on the spindle housing to measure the real-time vibration signals. LMS SCADAS Lab was employed to sample signals in 20480Hz and transmitted them to a laptop. Cutting experiments were conducted by straightly milling an aluminum brick with a three teeth end mill cutter. Spindle speed and depth of cut both varied from low level to high level to activate the chatter phenomenon. 3324Hz, which is the main chatter frequency, was identified to be a nature frequency by impact test.

2.3. Data preparation

The measured vibration signals will be divided into a series of frames, using a fixed sampling window (256 points) and a small frame shift. The frame shift is set relatively small to 1/8 the length of the sampling window to create more samples. In order to ensure the quality of the train samples, the segments in transition states are abandoned. Hamming window is used to eliminate energy loss problem. Finally a set of 47700 samples is extracted from three cutting experiments for the DBM training. And the whole sample set is divided into two parts, a subset with 45000 samples as the training data, and a small subset with 2700 samples as the testing data. Three

different cutting states are included, namely idling moving, stable cutting and chatter.

3. Results and discussions

3.1. Comparisons among modeling methods with manually defined features

In order to demonstrate the influence of manual feature extraction, different modeling methods with different features have been performed. The modeling performance of DBN with a BP output layer is compared with neuron network (NN) with only one hidden layer and support vector machines (SVM). Three different kinds of features are included in the comparisons. The first is the raw data just with traditional normalization. The second is Mel-frequency cepstrum coefficient (MFCC), which is a paramount feature in automatic speech recognition [16]. It is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear Mel-scale of frequency. The last is a wavelet feature pair proposed by Yao [9], which includes the energy ratio (T1) and standard variation (T2) of the sub-signal where the chatter frequency located.

Table 1 Errors of different modelling methods with different features.

Features	Method	Dimension	Training Error	Testing Error
None		256	6.22%	0.00%
MFCC	NN	12	4.11%	13.41%
Wavelet		2	0.027%	0.037%
None		256	20.23%	19.41%
MFCC	SVM	12	0.30%	0.36%
Wavelet		2	0.051%	0.037%
None		256	0.01%	0.00%
MFCC	DBN	12	0.13%	0.27%
Wavelet		2	0.02%	0.00%

Table 1 shows the training error and testing error of NN, SVM and DBN with different features. It is obviously that the performance of NN and SVM depends on the selection of features. In general, model performance would be improved with decrease of the feature dimension. For example, the SVM method is troubled with the hyper-parameters selection when trained on the raw signals because the dimension and number of the training samples are both too large. However, without any alteration of network structures, the DBN method consistently presents wonderful performances in all features. The classification accuracy is stable and high for both training samples and testing samples, which indicates that the method does not encounter with any overfitting problem. In contrast, severe overfitting, which is represented by a large testing error and a relatively much smaller training error, occurs when NN is trained on MFCC feature.

Table 2 Errors of clustering using k-means and DBN with different features.

Features	Dimension	Classes	K-means	DBN
None	256	3	54.83%	0.01%
MFCC	12	3	17.00%	0.13%
Wavelet	2	3	6.52%	0.02%
	2	2	0.05%	0.00%

Errors of clustering using k-means and DBN with different features are listed in Table 2. K-means method, one of the clustering methods, extracts the distribution structure of the training set to build clusters, and is different from the former supervised learning methods like NN and SVM. It can be seen as a simple feature extraction method. When the feature varies from 256-dimensional raw signal to 2-dimensional wavelet pair, the performance of k-means obviously improves. It should be noticed that the performance of k-means becomes different in distinguishing 2 and 3 classes with wavelet feature. Fig. 2 can help us to explain the phenomenon. The wavelet feature pair is designed to distinguish whether the machine tool is in chatter state or not, whereas the idling moving is not taken into consideration. Consequently, the idling moving state and stable state are not well separated in the feature space, resulting in a higher error rate of the k-means method in 3-class classification. This implies that proper feature would significantly improve the performance of a modeling method, and different features should be constructed for different problems. However, the manual definition of features turns to be a fussy procedure. On the contrary, DBN performs well no matter what feature set is chosen.

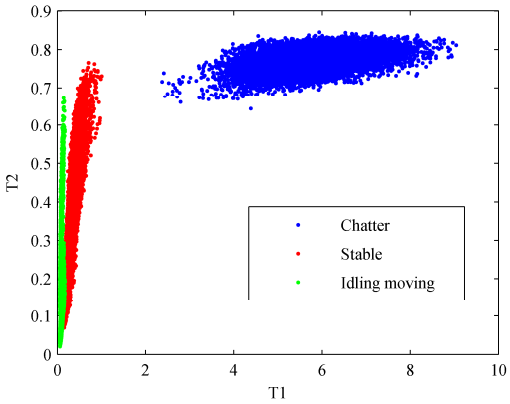

Fig. 2 The feature space of the wavelet feature pair.

DBN provides a uniform framework to model the data relationship and exhibits amazing power in modeling complicated data structure without much manual interference. It has been proved in theory that with enough hidden units, DBN can model any kind of data distribution and the extension of the training data set will always improve the model performance until the data distribution has been perfectly captured [14].

3.2. Feature extracting ability analysis of DBN

The modeling accuracy of DBN with BP layer totally depends on the output feature of the last DBN layer. A DBN with 256-256-256-100-1 structure is trained using the above training set to exhibit the feature extracting ability. Fig. 3(a) illustrates the feature space extracted from the training data using the DBN method. In comparison, Fig. 3(b) shows the space of top two principal components from principal component analysis (PCA). Obviously, the output feature space of DBN clearly separates the three states with a

relatively large margin, even if stable cutting state tangles with idling moving state in the feature space from PCA. DBN is obviously more capable than PCA in extracting critical structure from the given data. Based on the well prepared features of DBN, any kind of classification method would achieve an outstanding performance. In addition, the DBN is quite flexible in any dimension of constructed feature space as long as the training data is enough.

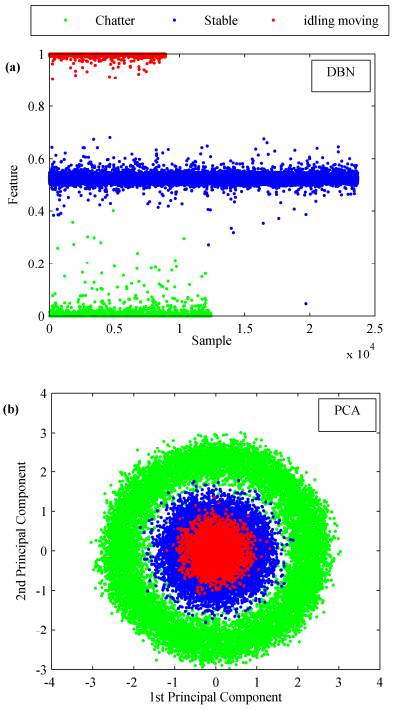

Fig. 3 Feature space obtained using DBN and PCA: (a) the feature space of DBN output and (b) the first two principal components of PCA.

The DBN structure is visualized for a better understanding of what is done during the training. By sampling [20] the first 256-256 RBM layer, the visualized results on the raw signals are shown in Fig. 4. The results are obtained by firstly sampling the RBM from the beginning hidden state where only the current hidden unit is 1 and the others are 0. And the bricks, which represent the feature signal associated with a hidden unit, are taken from the associated visible unit sequence in row-wise. The bricks can be seen as signals in different frequencies and amplitudes and some of them are translated to signal sequences, as shown in Fig. 5.

Fig. 5 shows that each brick represents a distinct signal sequence with particular frequency and amplitude. David Hubel and Torsten Wiesel proposed a theory about "Orientation Selective Cell" to explain what happens in human brain [12]. The cell will active when similar image with its saved memory is captured and tell the brain what is in sight. We can infer that the hidden units in the DBN behave just like the selective cell which turns prominent when similar vibration signal segment is input into the network, otherwise very small. In the hierarchical network structure of DBN, the high layer nonlinearly combines the low level features to form more complicated features to acquire a more elaborate description of the input data distribution. In the former comparison with PCA, we intentionally set the unit number of last layer to 1, and the network automatically and successfully separates the feature space in the specified dimension.

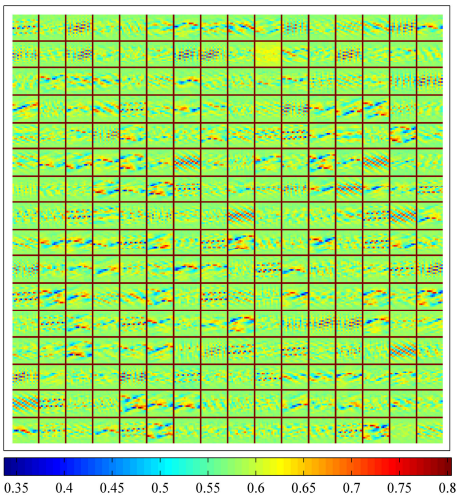

Fig. 4 Visualized result of the first layer of DBN.

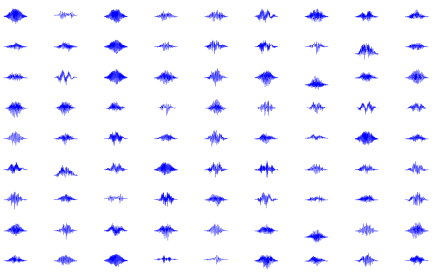

Fig. 5 Translated signal sequences of some pick-out bricks in Fig. 4.

4. Conclusions

Deep belief networks (DBNs) are introduced in this study to directly model the cutting states based on the raw measured vibration signal. Comparisons among different modelling methods with different features have been given. DBN shows amazing performances in both raw signals and training data with feature extracting preparations, and no obvious overfitting is observed.

Several conclusions have been found:

(a) Deep belief networks can steadily achieve high performance on the raw vibration signal without too much data preparation;

(b) Automatic feature extracting is conducted when the network is trained, and the elaborate feature representation is critical to the high performance of DBN;

(c) The DBN can be seen as a more powerful tool than PCA in separating the data, when used to reduce data dimensionality.

Because of the wonderful performance and simple preprocessing, DBN is a very promising tool to be used in more complicated applications where numerous different kinds of signals need to be taken into consideration to construct a more detailed data description.

Acknowledgements

The authors would like to acknowledge financial support from the National Program on Key Basic Research Project (Grant No. 2013CB035805, 2012CB025903), National Natural Science Foundation Council of China (Grant No. 51105152, 51125021), National Key Technology Support Program (Grant No. 2013BAF03B01).

References

[1] Dimopoulos, C. and A.M. Zalzala, Recent developments in evolutionary computation for manufacturing optimization: problems, solutions, and comparisons. Evolutionary Computation, IEEE Transactions on, 2000. 4(2): p. 93-113.
[2] Jardim-Goncalves, R., et al., Knowledge framework for intelligent manufacturing systems. Journal of Intelligent Manufacturing, 2011. 22(5): p. 725-735.
[3] Thomas, A. and D. Trentesaux, Are Intelligent Manufacturing Systems Sustainable?, in Service Orientation in Holonic and Multi-Agent Manufacturing and Robotics. 2014, Springer. p. 3-14.
[4] Chandrasekaran, M., et al., Application of soft computing techniques in machining performance prediction and optimization: a literature review. The International Journal of Advanced Manufacturing Technology, 2010. 46(5-8): p. 445-464.
[5] Tran, V.T., F. AlThobiani, and A. Ball, An approach to fault diagnosis of reciprocating compressor valves using Teager–Kaiser energy operator and deep belief networks. Expert Systems with Applications, 2014. 41(9): p. 4113-4122.
[6] Abellan-Nebot, J.V. and F.R. Subirón, A review of machining monitoring systems based on artificial intelligence process models. The International Journal of Advanced Manufacturing Technology, 2010. 47(1-4): p. 237-257.
[7] Teti, R., et al., Advanced monitoring of machining operations. CIRP Annals-Manufacturing Technology, 2010. 59(2): p. 717-739.
[8] Samanta, B. and K.R. Al-Balushi, Artificial neural network based fault diagnostics of rolling element bearings using time-domain features. Mechanical Systems and Signal Processing, 2003. 17(2): p. 317-328.
[9] Yao, Z., D. Mei, and Z. Chen, On-line chatter detection and identification based on wavelet and support vector machine. Journal of Materials Processing Technology, 2010. 210(5): p. 713-719.
[10] Mierswa, I. and K. Morik, Automatic feature extraction for classifying audio data. Machine learning, 2005. 58(2-3): p. 127-149.
[11] Hinton, G.E. and R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks. Science, 2006. 313(5786): p. 504-507.
[12] Hubel, D.H. and T.N. Wiesel, Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. The Journal of physiology, 1962. 160(1): p. 106.
[13] Jones, N., The learning machines. Nature, 2014. 505: p. 146-148.
[14] Bengio, Y., Learning deep architectures for AI. Foundations and trends® in Machine Learning, 2009. 2(1): p. 1-127.
[15] Bengio, Y., Deep learning of representations: Looking forward, in Statistical Language and Speech Processing. 2013, Springer. p. 1-37.
[16] Hinton, G., et al., Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. Signal Processing Magazine, IEEE, 2012. 29(6): p. 82-97.
[17] Tamilselvan, P. and P. Wang, Failure diagnosis using deep belief learning based health state classification. Reliability Engineering & System Safety, 2013. 115: p. 124-135.
[18] Hinton, G., A practical guide to training restricted Boltzmann machines. Momentum, 2010. 9(1): p. 926.
[19] Carreira-Perpinan, M.A. and G.E. Hinton. On contrastive divergence learning. in Proceedings of the tenth international workshop on artificial intelligence and statistics. 2005. Citeseer.
[20] Erhan, D., A. Courville, and Y. Bengio, Understanding representations learned in deep architectures, 2010, Technical Report 1355, Université de Montréal/DIRO.