Pattern Recognition Letters 132 (2020) 115–122

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Multi-task learning using variational auto-encoder for sentiment classification

Guangquan Lu[a], Xishun Zhao[a,*], Jian Yin[b], Weiwei Yang[b], Bo Li[b]

[a] Institute of Logic and Cognition, Department of Philosophy, Sun Yat-sen University, Guangzhou 510275, China
[b] Guangdong Key Laboratory of Big Data Analysis and Processing, Sun Yat-sen University, Guangzhou 510006, P.R. China

ARTICLE INFO

Article history:
Available online 28 June 2018

Keywords:
Sentiment classification
Opinion mining
Deep learning
Multi-task learning
Variational auto-encoder
LSTM
Big data

ABSTRACT

With the rapid growth of the big data, many approaches in the representation of text for sentiment classification have been successfully proposed in natural language processing. However, these approaches remedy this problem based on single-task supervised objectives learning and do not consider their relative of multiple tasks. Based on these defects, in this work, we consider these tasks are relative, and use weight-shared parameters for learning the representation of text in neural network model, we introduce and study a multi-task approach with variational auto-encoder generative model (**MTVAE**) by jointly learning them. Experimental results on six subsets of Amazon review data show that the proposed approach can effectively improve the sentiment classification accuracy by other relative tasks.

1. Introduction

Sentiment classification [37,47] is an important task in natural language processing (NLP), and its aim is to classify the given documents or sentences as expressing a positive or negative option. There is a large majority of literature [8,23,35,36,50] on this task. Be different from other natural language processing task (e.g. POS tagging, parsing, summarization extraction), in this paper, we focus on the text representation learning in neural network for sentiment classification.

Recently, due to the successes of deep neural network (DNN) techniques in computer vision [24,25], speech recognition [2,14,56], natural language processing absorbs these technology advantages, many distributed representations approaches [33,44,45] (including the word level, sentence level and document level) have spring up. Feature engineering is labor intensive, and that word distributed representation (or word embedding) [3], which is a dense, low-dimensional and real-valued vector for a word, can be learned by the unsupervised big data directly by using deep neural network model. According to the advantages of word distributed representation, in academia and industry circles, many researchers turned instead to deep neural network technology.

With the development of NLP in deep neural network, a deep learning model in the form of the recurrent neural networks (RNNs) is very popular for capturing time dependencies in temporal data (e.g. handwriting data [19], music data [5] or text data [6]). We use recurrent neural networks for capturing the representation of text sequences. In practical sentiment classification applications, training data is extremely expensive and is obtained by taking out lots of manual labor. Moreover, training data is obtained by manually often have some mistakes (e.g. noise, incompletes), all of these bring about other difficult situation about the objective task.

It was exhilarated that some generative models have their ability to learn from unlabeled data. Specially, a deep generative model called variational auto-encoder (VAE) [31] has been proposed. Since it is appropriate for unlabeled data on the learning tasks, then it is used in unsupervised and semi-supervised learning [30,41] scene for improving the performance. Sequential variational auto-encoder with recurrent neural network has been successed in different fields, for example, image generation [21], handwriting recognition and natural speech [11], and so on.

RNN-based variational auto-encoder generative model that incorporates distributed latent representation of text, and employing LSTM for their decoding architecture has been proposed by Bowman et al. [6]. They demonstrate their method have effectiveness in imputing missing words, but the latent space of text have some negative influence during the process of training the latent representation. Miao et al. [43] have used Multi-Layer Perceptron (MLP) for encoder (inference network) to construct their neural network, and used a softmax decoder (generative model) reconstruct their text data. Patidar et al. [42] adopted an LSTM-based variational auto-encoder to build a FAQ-bot, and their experiments show that

* Corresponding author.
E-mail addresses: guangquanl85@gmail.com (G. Lu), hsszxs@mail.sysu.edu.cn (X. Zhao), issjyin@mail.sysu.edu.cn (J. Yin), yangww8@mail2.sysu.edu.cn (W. Yang), libo68@mail.sysu.edu.cn (B. Li).

https://doi.org/10.1016/j.patrec.2018.06.027
0167-8655/

generative model can be fixed some linguistic training bias by generating novel sentences. Yang et al. [57] have proposed a method that using the dilated convolutional neural networks (CNN) as a decoder in VAE for language modeling and semi-supervised classification tasks. Xu et al. [54] have appended the label as a condition to the decoder structure for text classification tasks. Their results indicated they are able to obtain the latent representation well, but they ignored the relationship among several tasks by learning these tasks separately.

People often learn the skills together. For example, physical exercises (running and jumping), learning languages (English, Chinese and French and so on). They are all doing the relative tasks together unconsciously. Moreover, some skills obtained can help to get other skills, which also illustrated the knowledge can be inductive transfer. Borrowing from this idea, multi-task learning have spring up in artificial intelligence and big data analysis. Caruana [7] has studied neural network approaches for inductive transfer, they can improve learning capabilities in a task by using the information contained of other relative tasks in his thesis. They can learn the relative tasks in parallel using a shared representation simultaneously. Besides, the shared representation is learned which is learned in each task can help other tasks improving their generalization performance. Related work [1,12,38,39] has used for NLP models by jointly learning correlated tasks. Multi-task learning allows to the tasks keep relation to learn a representation, and these methods also shared the parameters to transfer knowledge during the training process.

In addition, motivated by the aforementioned interpretation, we hypothesize our sentiment classification tasks are relative. We propose deep network architecture of sharing parameters information between two sentiment classification tasks, and illustrate the benefits by jointly learning them. More specifically, variational auto-encoder has been proved an extremely effective generation model to learn the feature latent representation from sequence [21,6,11], and it also promised in sentiment classification tasks. And the recurrent neural networks is extremely suited for learning time dependencies in temporal data, furthermore, long short-term memory (LSTM) [27] is a form of RNNs, and it can avoid the gradient explosion in training process. Combining the advantages of both variational auto-encoder and LSTM, we model the weight-shared parameters neural network in multi-tasks learning for sentiment classification. To remedy the binary classification task and five-point classification task at the same time, we learn the text latent representation with variational auto-encoder simultaneously.

We demonstrate that each task is benefit for other tasks learning, and it can be learning the text latent representation well by the VAE. Moreover, it performs substantially better than the popular LSTM model. We selected the Multi-Layer Perceptron for decoders which it can be effective for our learning tasks. Experimental results show that jointly learning of multiple text tasks can improve the text representation capability, in other word, it can learn useful feature latent representation (e.g. semantic information).

In summary our main contributions are as follows:

(1) We used the LSTM for encoder (inference network) with the VAE for the text latent representations, and employed Multi-Layer Perceptron for decoder (generative model) reconstruct the text data, which our model is designed as a hybrid structure neural network (**MTVAE**) for sentiment classification.

(2) Different with supervised learning methods for sentiment classification, we combine with the generative model, five-point classification task and binary classification for training together simultaneously. Therefore, the important information representations, especially, the weight-shared parameters of related tasks are learned robust.

(3) We explore the use of Multi-Layer Perceptron for decoder with sentiment classification and find it perform better than using LSTM for decoder. Experimental results show that our multi-task learning model outperforms most of state-of-the-art approaches.

The article is organized as follows. In the next section, we introduce the preliminaries for sentiment classification. And then our multi-task learning architecture presented in Section 3. Section 4 presents some experiments comparing analysis in different models. Section 5 discusses the previous studies related to multi-task learning for sentiment classification, and Section 6 concludes and further work.

2. Preliminaries

The variable-length text is represented as a fixed-length vector in neural network, in addition, the recurrent neural network is deep learning structure, which can capture the contextual of text effectively. Specially, it is used in NLP for language model popularly. However, RNN structure has some defects on handling long term dependencies sequential input data because the vanishing gradient problem [4] was occurring. Fortunately, LSTM [27] has proposed for overcoming the gradient explosion in training process. In this section, we describe the basic knowledge of our approach, comprising of the variational auto-encoder, LSTM, Multi-Layer Perceptron and Multi-task learning.

2.1. Variational auto-encoder

Let us consider the samples of some continuous or discrete variable $\mathbf{x}$ from the dataset, and we hypothesize the variable $\mathbf{x}$ are generated by an unobserved continuous random variable $\mathbf{z}$. An auto-encoder [17,42] is usually viewed as consisting of two parts, an encoder representation function (latent representation) $\mathbf{z} = f(\mathbf{x})$ and a decoder that produces a reconstruction function $\hat{\mathbf{x}} = g(\mathbf{z})$. The objective of auto-encoder is for minimizing the reconstruction error $\hat{\mathbf{x}}$ with respect to input $\mathbf{x}$. Previously, the auto-encoder is used for feature learning or dimensionality reduction, but recently, the relative between latent variable models and auto-encoder are used for generative model, and it can be trained by back-propagation method.

Variational auto-encoder is known as an unsupervised learning approach for modeling complicated big data distributions. It is also comprised of an encoder and a decoder, but it does not have the tuning parameters to the sparsely penalty. So it is different from the sparse auto-encoder. Similar to the [15,31], we denote $\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})$ as the encoder (recognition model) and $\mathbf{P}_\theta(\mathbf{x}|\mathbf{z})$ as the decoder (generative model), where the ϕ and θ are denoted as the parameters to encoder and decoder, respectively. Due to the posterior density $\mathbf{P}_\theta(\mathbf{z}|\mathbf{x}) = \mathbf{P}_\theta(\mathbf{x}|\mathbf{z})\mathbf{P}_\theta(\mathbf{z})/\mathbf{P}_\theta(\mathbf{x})$ is intractable, so VAE uses the encoder $\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})$ to approximate to the true posterior $P_\theta(\mathbf{z}|\mathbf{x})$. In order to solve this problem, we employ the variational Bayesian methods, and assume $\mathbf{E}_{\mathbf{Q}_\phi(\mathbf{z})}\mathbf{P}(\mathbf{x}|\mathbf{z})$ and $\mathbf{P}_\theta(\mathbf{x})$ are relative.

In order to capture latent information of latent variables $\mathbf{z}$, we assume $\mathbf{z}$ is sampled from any arbitrary distribution with probability density function $\mathbf{Q}_\phi(\mathbf{z})$. Generally, we can measure the different between $\mathbf{Q}_\phi(\mathbf{z})$ and the $\mathbf{P}_\theta(\mathbf{z}|\mathbf{x})$ by using the Kullback–Leibler (**KL**) divergence:

$$\mathbf{D}_{KL}[\mathbf{Q}_\phi(\mathbf{z})||\mathbf{P}_\theta(\mathbf{z}|\mathbf{x})] = \mathbf{E}_{\mathbf{Q}_\phi(\mathbf{z})}[\log \mathbf{Q}_\phi(\mathbf{z}) - \log \mathbf{P}_\theta(\mathbf{z}|\mathbf{x})]. \tag{1}$$

Since the **KL**-divergence is non-negative, $\log \mathbf{P}_\theta(\mathbf{x})$ does not depend on $\mathbf{Q}_\phi$, we apply the Bayes rules to the $\mathbf{P}_\theta(\mathbf{z}|\mathbf{x})$, the equation can be written as:

$$\begin{aligned}\mathbf{D}_{KL}[\mathbf{Q}_\phi(\mathbf{z})||\mathbf{P}_\theta(\mathbf{z}|\mathbf{x})] = \mathbf{E}_{\mathbf{Q}_\phi(\mathbf{z})}[&\log \mathbf{Q}_\phi(\mathbf{z}) - \log \mathbf{P}_\theta(\mathbf{x}|\mathbf{z}) \\ &- \log \mathbf{P}_\theta(\mathbf{z})] + \log \mathbf{P}_\theta(\mathbf{x}).\end{aligned} \tag{2}$$

Note that rearranging the expectation of the right hand side, the relative of **KL**-divergence, and tuning the Eq. (2), we can get:

$$\log \mathbf{P}_\theta(\mathbf{x}) - \mathbf{D}_{KL}[\mathbf{Q}_\phi(\mathbf{z})||\mathbf{P}_\theta(\mathbf{z}|\mathbf{x})] = \mathbf{E}_{\mathbf{Q}_\phi(\mathbf{z})}[\log \mathbf{P}_\theta(\mathbf{x}|\mathbf{z})] - \mathbf{D}_{KL}[\mathbf{Q}_\phi(\mathbf{z})||\mathbf{P}_\theta(\mathbf{z})]. \quad (3)$$

Due to the $\mathbf{Q}_\phi$ can be any arbitrary distribution and $\mathbf{x}$ is fixed, so VAE does a good conjunction mapping $\mathbf{x}$ to the $\mathbf{z}$ space that can produce $\mathbf{x}$. The core of VAE is that it construct a $\mathbf{Q}_\phi$ which depend on $\mathbf{x}$. Moreover, we can get the following equation:

$$\log \mathbf{P}_\theta(\mathbf{x}) - \mathbf{D}_{KL}[\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})||\mathbf{P}_\theta(\mathbf{z}|\mathbf{x})] = \mathbf{E}_{\mathbf{Q}_\phi(\mathbf{z})}[\log \mathbf{P}_\theta(\mathbf{x}|\mathbf{z})] - \mathbf{D}_{KL}[\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})||\mathbf{P}_\theta(\mathbf{z})] = \mathcal{L}(\theta, \phi; \mathbf{x}). \quad (4)$$

In order to maximize the $\log \mathbf{P}_\theta(\mathbf{x})$, we should make the $\mathbf{D}_{KL}[\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})||\mathbf{P}_\theta(\mathbf{z}|\mathbf{x})]$ minimization. Specially, if $\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})$ match $\mathbf{P}_\theta(\mathbf{z}|\mathbf{x})$ perfectly, in other word, the **KL**-divergence $\mathbf{D}_{KL}[\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})||\mathbf{P}_\theta(\mathbf{z}|\mathbf{x})]$ will equal zero, and the problem equivalent to optimize the lower bound $\mathcal{L}(\theta, \phi; \mathbf{x})$.

As common as in machine learning, in our experiment, the posterior $\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})$ and prior $\mathbf{P}_\theta(\mathbf{z})$ distributions are drawn from a diagonal Gaussian distribution, and these parameters are used "reparameterization trick" in Kingma et al. [31] to solve. Therefore, to optimize the lower bound $\mathcal{L}(\theta, \phi; \mathbf{x})$, training process need to solve the model parameters θ and approximation parameters ϕ jointly used stochastic gradient ascent by back-propagation. Note that, in order to use back-propagation, the posterior $\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})$ must be continuous.

2.2. Multi-Layer perceptron and LSTM network

Multi-Layer perceptron is also named feed forward neural network or deep feed forward network, and it is widely used for nonlinear modeling. Generally, MLP is an approximation algorithm, which it is used for learning function approximation. Nielsen et al. [46] have proved that MLP with one hidden layer can perform universal approximation function. However, in order to learn much more representation, many researchers usually considered more than one hidden layer for training the weight parameters.

To complete the MLP neural network training, we should choose the activation functions to compute the hidden layer values, and also we need to use the back-propagation algorithms for computing the gradients functions. Because of the MLP can learn nonlinear function approximation well, motivated by these observations, in this paper, we choose MLP for decoder, and then obtaining the latent representation of text sequences to achieve sentiment classification.

Recurrent neural networks [51] can be defined as special kind of deep neural networks for processing sequential data. These structures of networks have the ability in persisting the previous information on account of that has a combination of networks in loop. In order to solve the sentiment classification problem, the variable-length texts are taken as an input sequence which is similar with [9], and they are also mapped as a fixed-length vector for encoder the text sequence.

However, the vanilla recurrent neural network does not preserve the long term dependencies in input sequences because these network model training exist the exploding or vanishing gradients problem [4,53]. Fortunately, the long term dependencies problem has been overcome by the long short-term memory network (LSTM) [27]. LSTM network can remember previous information for long periods of time. And the more previous information is preserved, the more accuracy of model will obtain. So LSTM is a good choice to train text sequences for sentiment classification. Generally, LSTM networks have the form of a chain of repeating modules that has four neural network layers interacting in a very special way. And the four layers composed of the vector of memory cells $\mathbf{c}_t \in \mathbf{R}^n$. They control the stored, updated, forgotten and exposed of text sequence information inside the network together.

There are many different versions on LSTM [16,32,58]. Following [18,55], the input gate, forget gate, output gate, memory cell are defined as $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ and $\mathbf{c}_t$ respectively, and $\hat{\mathbf{c}}_t$ is a vector of generate candidate values. We use this architecture for encoder, and in these literatures, the computation at each step equations is defined as:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{wi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1}) \quad (5)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{wf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1}) \quad (6)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{wo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1}) \quad (7)$$

$$\hat{\mathbf{c}}_t = \tanh(\mathbf{W}_{wc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1}) \quad (8)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \hat{\mathbf{c}}_t \quad (9)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (10)$$

where σ is the logistic sigmoid function, $\mathbf{x}_t$ is the input at the current time t, and the vector $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t \in [0, 1]^n$, $\odot$ denotes element wise multiplication, in addition to that each of these vectors equals to the dimension of hidden layer $\mathbf{h}$.

2.3. Multi-tasks learning

Multi-task learning [7] has been proven to obtain more efficient model than many tasks learning one by one, and it can capture intrinsic relatedness of tasks simultaneously. Multiple relatedness tasks shared deep layers parameters with multiple loss function can improve features representation produced on neural network. During to the tasks are correlated and the parameters sharing can improve its generalization performance, several multi-task learning methods have been proposed, e.g. [1,12,38,39,59,62].

Though the excellent performance of multi-task learning, to best of our knowledge, there is no attempt using VAE involved in the literature of sentiment classification. Collobert et al. [12] proposed a general deep neural network architecture for natural language processing, which shows that jointly with other tasks learning simultaneously can improve modeling performance. Liu et al. [38] and Liu et al. [39] used different deep architectures neural network for multi-task learning, and settle multiple text sequences tasks simultaneously, which demonstrates the accuracy promotion effect of classification. Balikas and Moura [1] applied multi-task learning for fine-grained twitter sentiment analysis, which demonstrates learning correlated tasks at the same time can improve the fine-grained classification performance.

However, as the multi-task learning has been applied for text sequence, these network are not encoder-decoder structure neural network, so they do not learn the global representation of text sequence shared parameters sufficiently. Inspired by the variational auto-encoder can extract global features from sequences [6] effectively, in this paper, we adopt a multi-task learning framework [7], used LSTM-VAE as encoder and Multi-Layer Perceptron network as decoder structurally, and applied hybrid structure network for learning shared parameters to improve the accuracy of sentiment classification simultaneously.

3. Multi-task learning architecture for sentiment classification

In this section, we describe our approach in detail. Many existing neural network methods using VAE are based on a single task [42,43,54], and these models lacked the capability for learning the relative among different tasks. To deal with this problem and based on the VAE network structure, we proposed a novel

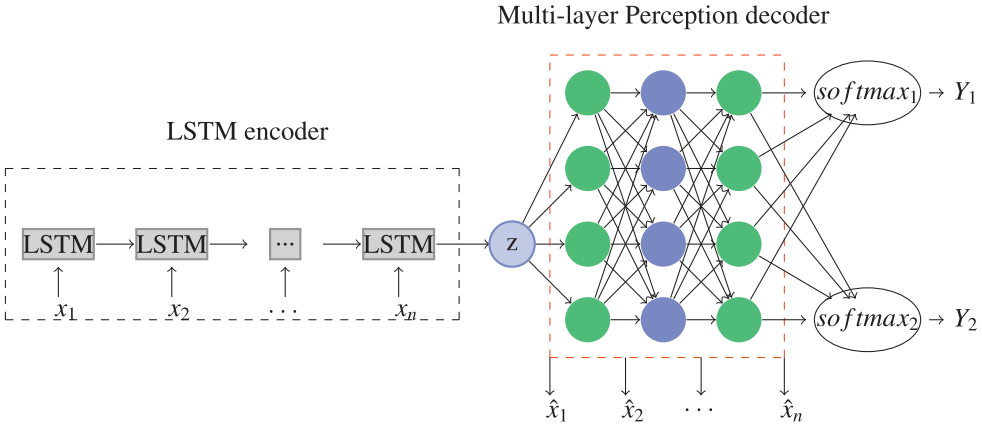

Fig. 1. The proposed sketch for modeling text sequence with multi-task learning.

Table 1
Configuration of the share parameters model.

#Structure	#Layer	#Parameters
Encoder	Embedding	5,894,700
	LSTM	17,024
	Perception	3,300
	Perception	3,300
Decoder	Lambda	0
	Perception	3,232
	Perception	33,000
Binary/five-point	softmax	2,002/5,005

multi-task learning approach to learn the text sequence representation, binary classification and five point classification simultaneously. For improving the final performance, the unsupervised pre-training is adopted for achieving these tasks.

3.1. The proposed sketch for modeling text sequence

Our objective is to construct a hybrid network with VAE, LSTM and MLP for binary classification and five-point classification simultaneously. The function of our network is to learn a mapping $\mathbf{F}: \mathbf{X} \rightarrow (\hat{\mathbf{X}}, \mathbf{Y}_1, \mathbf{Y}_2)$, where $\mathbf{X}$ is the text sequence input, $\hat{\mathbf{X}}$ is the prediction of text sequence reconstruct, $\mathbf{Y}_1$ is the value denoted as sentiment polarity output, e.g. positive and negative, and $\mathbf{Y}_2$ is the value denoted as sentiment output, e.g. very negative, negative, neutral, positive, very positive. To this end, we construct a sketch for modeling text sequence with multi-task learning approach, which is using shared parameters layers and multi-loss function together. The structure of the proposed network is illustrated in Fig. 1 and the detail configuration is in Table 1. Specially, the number parameters in the Lambda layer is zero, because we used "reparameterization trick", and these parameters are generated by a diagonal Gaussian distribution. In the encoder parts, the perception layers are for computing the parameters of Gaussian distribution, and it is not presented in Fig. 1, because these components are served for $\mathbf{P}_\theta(\mathbf{z})$.

The inspiration of proposed network structure is intuitive. A key problems of learning method is that LSTM is used for encoder text sequence to get the posterior probability $\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})$, which we hypothesize it is diagonal Gaussian. And the LSTM output can be parameterized as the mean μ and variance σ. The sample z is come from $\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})$, and the decoder is on the sample by connecting z with every word embedding of previous LSTM output. To remedy the sentiment classification problem and apply multi-task learning for text sequences, we combine the decoder network, binary classification and five-point classification together, and share their model parameters in different tasks, the main reason that we consider these tasks are relative, which can improve the performance by other tasks learning. Finally, the decoder network is used by multi-layer perception on account of multi-layer perception has the strong reconstructing ability.

3.2. Variant auto-encoder structure and sentiment classification

Based on our proposed sketch network, we have used LSTM units as encoder and multi-layer perception as decoder of VAE. The variable length input text sequence in the encoder is as shown in Fig. 1. And the input text sequence is converted into a vector representation by using word embedding layer, then it is fed to the LSTM layer. The last hidden layer of LSTM, which is applied to predict μ and σ of the posterior distribution $\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})$. By sampling and using the re-parameterization trick [31], the sampled encoding z passes through the start layer of Multi-layer perception for decoder, then the highest probability of word representation will feed into the next layer for prediction.

In Eq. (4), we take the first term of variational bound $\mathbf{E}_{\mathbf{Q}_\phi(\mathbf{z})}[\log \mathbf{P}_\theta(\mathbf{x}|\mathbf{z})]$ as an objective function, and the posterior probability $\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})$ will become small. The second term of $\mathbf{D}_{KL}[\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})||\mathbf{P}_\theta(\mathbf{z})]$ is Kullback–Leibler divergence between the latent posterior $\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})$ and the prior distribution $\mathbf{P}_\theta(\mathbf{z})$, it will discourage the training process rather than memorizing each $\mathbf{x}$ as a isolated latent value.

Crucially, training the VAE using the loss function given in Eq. (4) is difficult. In order to learn global latent representation of text sequence, VAE encodes the important information in the latent vectors $\mathbf{z}$, it will have a very small cross entropy term and have a non-zero **KL** divergence, if that the part of network does not generate new text sequences. Contrarily, if the **KL** loss is large, it learns the insignificant representation of text sequences, furthermore, it will affect the performance of sentiment classification tasks. So we should train these network via the loss function gradually decrease in the direction of convergence. In order to achieve the VAE training, Bowman et al. [6] used cost annealing and word-dropout at the decoder to avoid extreme circumstances. In our multi-task learning, we also use the word-dropout for the decoder for training.

For utilizing the multi-task learning in sentiment classification simultaneously, we let the model shared parameters in the part of encoder structure and decoder structure. To do this, we rewrite the variational bound of Eq. (4):

$$\mathbf{E}_{\mathbf{Q}_\phi(\mathbf{z})}[\log \mathbf{P}_\theta(\mathbf{x}|\mathbf{z})] - \mathbf{D}_{KL}[\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x})||\mathbf{P}_\theta(\mathbf{z})] = \mathcal{L}(\theta, \phi; \mathbf{x}). \tag{11}$$

Based on Kingma et al. [30] proposed the semi-supervised learning method with variational auto-encoder and Xu et al.

[54] using semi-supervised learning with variational auto-encoder for text classification, we extend this variational auto-encoder deep neural generated model, and conduct the multi-task learning for sentiment classification. Specifically, our method has the following components: an LSTM encoder network, a multi-layer perception decoder network, a binary classifier and five-point classifier. These components are denoted by $\mathbf{Q}_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$, $\mathbf{P}_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z})$, $\mathbf{Q}_\phi(\mathbf{y}^{(binary)}|\mathbf{x})$ and $\mathbf{Q}_\phi(\mathbf{y}^{(five)}|\mathbf{x})$, respectively. For performing the representation learning of our sentiment classification tasks, which do not consider the structure of encoder and decoder, we have the follow Eqs. (12) and (13).

$$\hat{\mathbf{y}}^{(binary)} = softmax(\mathbf{W}^{(binary)}\mathbf{h}^{(binary)} + \mathbf{b}^{(binary)}), \tag{12}$$

$$\hat{\mathbf{y}}^{(five)} = softmax(\mathbf{W}^{(five)}\mathbf{h}^{(five)} + \mathbf{b}^{(five)}), \tag{13}$$

where $\hat{\mathbf{y}}^{(binary)}$, $\hat{\mathbf{y}}^{(five)}$ are prediction probabilities for binary classification and five-point classification, and $\mathbf{W}$, $\mathbf{b}$ are the weight and bias term which need to be learned.

The binary classification and five-point classification tasks are trained to minimize the cross-entropy of the predicted and true distributions, respectively. So we get

$$L(\hat{\mathbf{y}}^{(binary)}, \mathbf{y}^{(binary)}) = -\sum_{i=1}^{N_1}\sum_{j=1}^{2}\mathbf{y}_i^j \log(\hat{\mathbf{y}}_i^j), \tag{14}$$

$$L(\hat{\mathbf{y}}^{(five)}, \mathbf{y}^{(five)}) = -\sum_{i=1}^{N_2}\sum_{j=1}^{5}\mathbf{y}_i^j \log(\hat{\mathbf{y}}_i^j), \tag{15}$$

where $\hat{\mathbf{y}}_i^j$ is the predicted probabilities, $\mathbf{y}_i^j$ is the ground-true label, N_1 and N_2 are the number of training samples in binary classification task and five-point classification task, respectively.

In order to adopt the joint loss of binary classification and five-point classification task to train the multi-task learning for sentiment analysis, we can obtain the following classification cost function:

$$L_c = \lambda_1 L(\hat{\mathbf{y}}^{(binary)}, \mathbf{y}^{(binary)}) + \lambda_2 L(\hat{\mathbf{y}}^{(five)}, \mathbf{y}^{(five)}), \tag{16}$$

where λ_1 and λ_2 are the weights of binary classification and five-point classification, respectively.

Combine with the variational bound of Eq. (11) and the Eq. (16), we can get the global joint loss function:

$$J = \lambda_1 L(\hat{\mathbf{y}}^{(binary)}, \mathbf{y}^{(binary)}) + \lambda_2 L(\hat{\mathbf{y}}^{(five)}, \mathbf{y}^{(five)}) + \lambda_3 \sum_{\mathbf{x}\in\mathbf{X}} \mathcal{L}(\theta, \phi; \mathbf{x}), \tag{17}$$

where $\mathbf{X}$ is denoted the space of training set, λ_3 is a hyper-parameter weight of additional variational bound. Clearly, the proposed structure of multi-task learning in sentiment analysis can be optimized by the Adam [29]. The parameters λ_1, λ_2 and λ_3, are used for balancing the three loss functions, specially, the joint loss function will be considered as a variational auto-encoder, if λ_1 and λ_2 are set to 0.

3.3. Training

In order to learn the parameters of our multi-task learning model, following [12,39,40], the training process can be summarized as Algorithm 1. The task t is a binary classification, five-point classification or variable auto-encoder. In each iteration, the task t is selected randomly, and this training process calculates the sum of all tasks loss function. More specifically, training the VAE requires end-to-end with reparameterization trick [31], the pre-training will be used to initialized, based on its parameters, other tasks will use these parameters to initialize their network simultaneously, this will make our models converge more stably and swiftly.

Algorithm 1 Training multi-task learning with variational auto-encoder.

Require: training dataset $\mathbf{X}(\mathbf{X}^{(binary)} + \mathbf{X}^{(five)})$, $\mathbf{Y}^{(binary)}$, $\mathbf{Y}^{(five)}$, λ_1, λ_2, λ_3, learn rate ℓ;
Ensure: model Ω: { $\mathbf{W}^{(binary)}$, $\mathbf{W}^{(five)}$, b };
Initialize model Ω: { $\mathbf{W}^{(binary)}$, $\mathbf{W}^{(five)}$, $\mathbf{b}$ } randomly;
repeat
  Select a task t randomly;
  Select mini-batch samples from task t;
    ($\mathbf{X}^{(binary)}$, $\mathbf{Y}^{(binary)}$) for binary classification ;
    ($\mathbf{X}^{(five)}$, $\mathbf{Y}^{(five)}$) for binary classification ;
    ($\mathbf{X}$) for variational auto-encoder;
  Calculate the Loss: $J(\Omega)$ by Eq. (17);
  Calculate gradient: $\nabla(\Omega)$;
  Update model: $\Omega = \Omega - \ell\nabla(\Omega)$;
until maximum iteration

The computational complexity of learning MTVAE models per weight and time step with the Adma [29] optimization technique is $O(1)$. So following the LSTM [27], we are easy to obtain the update complexity per time step of MTVAE algorithm is $O(W)$, where W is the number of neurons weights. Thanks to GPU technology, our method is very efficient. And MTVAE is local in space and time, there is no need to store additional observed values during training process with unlimited size. More detailed discussion can be found in [52].

4. Results and discussion

In this section, we discuss the experimental results of our proposed multi-task learning model on two related sentiment classification tasks. All the experiments were performed using Keras [10] and scikit-learn [48] on Intel Core i5-7600K CPU based Personal Computer running at 3.8Ghz*4, 24GBytes of memory, on NVidia GTX 1080 8GBytes of GPU *1, and Ubuntu 16.04-64bit based operating system. In order to evaluate the performance of our sketch network model, we introduce our model parameters settings and experimental comparison.

4.1. Data sets and metrics

In this paper, we adopt the commonly classification accuracy (ACC) as evaluation metric to measure our proposed model performance. And the classification accuracy [22] can be defined as Eq. (18)

$$ACC = \frac{N_{correct}}{N}, \tag{18}$$

where N is the number of test data set and $N_{correct}$ is classified correctly.

To verify our proposed multi-task learning model, we make the experiments on six sentiment classification datasets[1] from the Stanford Network Analysis Project (SNAP) [26], Amazon Instant Video Full, Digital Music Full, Automotive Full, Amazon Instant Video Polarity, Digital Music Polarity, Automotive Polarity. Similar to the [13,61] construction of the dataset, we construct our binary classification tasks of datasets, the classification of full number of stars the user has given, and the other polarity label is considered stars 1 and 2 as negative, and starts 5 as positive. Table 2 is a summary about multi-task learning datasets.

[1] Available at http://jmcauley.ucsd.edu/data/amazon.

Table 2
the Statistics of six data sets used in this paper.

Data set	#Train samples	#Dev. samples	# Test samples	#Classes	#Types
Amazon Instant Video Full	22,276	7425	7425	5	review text
Amazon instant video polarity	14,696	4898	4898	2	review text
Digital music full	38,824	12,941	12,941	5	review text
Digital music polarity	24,829	8276	8276	2	review text
Automotive full	12,285	4094	4094	5	review text
Automotive polarity	9046	3015	3015	2	review text

All these datasets were divided in three parts (Train, Validation and Test) in 60:20:20 ratio. Train Samples and Validation Samples are used for training Multi-task learning model, here we select the best model based on the loss on validation data, i.e., J, see Eq. (17). In the selected data set, the number of train samples varies from 9046 to 38,824, the number of test samples varies from 3015 to 12,941, and the number of classes varies from 2 to 5. Specially, the number of train samples is very large, that also illustrates they are well fit to sentiment analysis tasks.

Table 3
Performance of Single task and Multi-task learning.

Data set\Model	Single task	MTVAE
Amazon instant video full	62.70%	**64.73%**
Amazon instant video polarity	94.26%	**94.75%**
Digital music full	62.87%	**65.42%**
Digital music polarity	**95.14%**	93.87%
Automotive full	67.72%	**68.95%**
Automotive polarity	**93.33%**	92.74%

4.2. Experiment setting

We train the network for sentiment analysis using the gradient-based optimization Adma [29] and backpropagation. In our experiments, because the review texts are short, and are similar to the tweets, so we use the word embeddings which are trained on tweets Glove embeddings[2] of Pennington et al. [49]. The vocabulary size is about 1.2M. We use a mini-batch size of 50 and the review texts of similar length are organized to be a batch. We use the Adam to train all the model with the learning rate is selected from range [0.001, 0.01]. The dimension of embedding for each word is 100, the dimension of z is 100; the hidden layer size of LSTM is 32. Bowman et al. [6] proposed the cost annealing trick was used to smooth training by tuning the weight of KL cost from 0 to 1. The word dropout rate is also setting from 0.1 to 0.4 in our experiments. The weights of the layers were initialized from a uniform distribution. In particular, we used pre-training to assure VAE convergence. Generally, we train VAE first, and the maximize epoch is 1000, then train the binary classification and five-point classification simultaneously. Specially, after 1000 epoch in VAE, we used the probability 0.5 for choosing a batch in five-point classification, 0.4 probabilities for choosing a batch in binary classification and 0.1 for VAE. Other hyper-parameters that obtain the best performance on the development set will be chosen for the final test sets evaluation. As shown in Table 1, the number of parameters which need to learn is very large, for example, VAE is 5,954,556, binary classification is 5,956,558 and five-point classification is 5,959,561 parameters, which should be learned on training process simultaneously.

4.3. Experimental results and comparison analysis

To verify the proposed method (**MTVAE**), we first compare our method shared with a pair of tasks with itself for single task sentiment classification. In training single task sentiment classification, we only consider the loss of binary classification or five-point classification. Table 3 shows the performance of single task and multi-task learning by the test set accuracies. The second column ('Single task') of the table shows the result of our method for each single task. And the column ('Joint Learning task') of the table shows the result of three combinations for the six datasets. In particular, joint learning has more than 3.24%, 4.06%, 1.82% and 0.52% improvement for datasets in Amazon Instant Video Full, Digital Music Full, Automotive Full and Amazon Instant Video Polarity data sets, respectively. It directly explains that in five-point classification task, joint learning can learn more abundant text feature representation than single task learning. Furthermore, although the Single task outperform our model on Digital Music Polarity and Automotive Polarity data sets, we have much fewer parameters for the tasks, comparing the parameters which are trained alone in totals. The performance of five-point classification task with joint learning increasing is larger than binary classification task, this also illustrates joint learning is more suitable for solving complex classification tasks.

Secondly, we compare our joint learning method with the state-of-the-art model using word-level embedding: **CNN** [28], **LSTM** [27], **BiLSTM** [20], **PV** [33]. We also use the **SVM** classification accuracy as Baseline. All the methods of maximize epoch is 50. And we list these comparison methods as follow:

1. [28] (**CNN**) is based on convolutional neural networks which consider the property of text sequences space using word2vec.
2. **LSTM** [27] is a form of RNNs, which has been applied to language model and speech recognition. Its network structure is built sequence history information of long distance feature representation for predicting the current output contents.
3. **BiLSTM** [20] is a vary form LSTM, which can use the history information and future information to predict current output contents.
4. **PV** [33] is a method using logistic regression on the paragraph vectors and its objective is for training distributed representation of documents for NLP tasks.

In all comparison method, we used the word embedding for features input except the **PV** method, and then conducted comparing these different models on the performance of sentiment classification tasks. Table 4 shows the performance of sentiment classification tasks which are evaluated by classification accuracy on six datasets among different methods. From this Table 4, we have following observations:

(1) Our proposed **MTVAE** using variational auto-encoder (**MTVAE**) outperforms each of the competing methods on the five-point sentiment classification tasks. In particular, our proposed approach has more than 1.53% improvement for Digital Music Full data set, compared to the second best method **BiLSTM**. And it has more than 14.83% improvement for Amazon Instant Video Full data set, compared to the worst method SVM. Moreover, this gives a richer feature representation learning for relative difficult tasks, which can learn a more robust representation in five point sentiment

[2] Available at http://nlp.stanford.edu/data/glove.twitter.27B.zip.

Table 4
Performance of shared-layer multi-task learning and other state-of-the-art models.

Data set\Model	SVM	CNN	LSTM	BiLSTM	PV	MTVAE
Amazon instant video full	56.37%	61.24%	63.95%	64.46%	61.34%	**64.73%**
Amazon instant video polarity	85.34%	92.96%	94.41%	94.61%	91.75%	**94.75%**
Digital music full	55.02%	62.18%	63.40%	64.43%	59.75%	**65.42%**
Digital music polarity	85.98%	92.92%	93.14%	93.75%	91.89%	**93.87%**
Automotive full	67.94%	68.31%	68.45%	68.34%	65.53%	**68.95%**
Automotive polarity	92.61%	**93.23%**	91.81%	93.06%	92.21%	92.74%

classification tasks, and it also shows the advantage of our model in Big Data analysis.

(2) In sentiment classification, the experiment results show that **MTVAE** has better performance than SVM and PV methods. Mainly, SVM and PV are not the deep neural network model, they do not construct a richer representation for the review texts. **CNN** and **BiLSTM** outperform our model in Automotive polarity data set while do not better than our model in other data sets. This illustrates that **CNN** can capture the space structure of review texts and **BiLSTM** can learn the bidirectional dependences structure of text sequences on small data set, because the samples in Automotive polarity are smaller than other data sets.

(3) At last, the performance of task learning can improve by other relative tasks learning. Particularly, it is a very effective method on training neural network for reducing the computing resources and it can capture much more non-linear relationships between the text feature representation learning.

5. Related work

Variational inference have been proposed to tackle the problem of natural language processing [6,42,57]. Our work is focused on variational inferences with multi-task learning for sentiment classification. Bowman et al. [6] used VAE and language model to find some interesting negative results. Patidar et al. [42] and Yang et al. [57] also proposed their neural network structure modeling text process based on VAE, however, these model can not be used to multi-task learning. Our work fills the gaps and successed in applying these technologies with multi-task learning for sentiment classification tasks. Yang et al. [57] used variational inference with language modeling text, which is different from our model in using multi-layer perception for decoder to learn text feature representation.

We use the LSTM as encoder and the MLP as decoder is inspired by Nielsen and Rumelhart [46,51], LSTM can effectively process the sequential data and MLP can perform universal approximation function well. Combining with LSTM and MLP, our model can learn a intricacy network architecture for strong text representation capacity and get much better performance.

Balikas and Moura [1] propose a multi-task learning approach based on recurrent neural network, in their architecture model, they need introduce additional features information to the neural network architecture. They discovered their method on twitter sentiment analysis leads to better results. Different from this, we absorb the VAE well-known architecture and apply multi-task learning for sentiment analysis, which can learn better feature representation for our tasks. It is worth mentioning that we will consider additional features information as future work. It will be potentially choice for obtaining additional gains.

Though many traditional approaches performing feature selection [34,60,63–66] and then achieving big data analysis (also sometime processing simultaneous) obtain good performance, it is labor intensive than the deep learning methods. Different from these traditional methods, our model employed the novel deep learning technologies, which can learn valid text representations for several relative tasks.

Liu et al. [40] used a multi-task deep neural networks for learning text representation with semantic classification and information retrieval tasks. However, they use bag-of-word representation, which make it missing words sequence information. The literature [38,39] designed a kind of shared memory neural network for multi-task learning, which can control shared layer of information flow. Unlike these works, in our paper, we adopt the encoder-decoder networks, which can effectively illustration the process of text representation learning, and it has a comparatively reasonable interpretation on feature representation learning.

To the best of our knowledge, this work is the first using VAE with multi-task learning for sentiment analysis, and gives a novel neutral network approach for capturing the feature representation of text sequences. The experiments prove that approach is effective for binary classification and five-point classification tasks.

6. Conclusions and future work

In this paper, we proposed a novel multi-task learning architecture to model text sequence via VAE. In sentiment analysis tasks, comparing other single task learning method, our proposed method obtains the better performance. Based on encoder-decoder neural network, we can capture the text feature representation well. Besides, multi-task learning architecture designed shows it employs shared parameters which can avoid much more parameters learning than single tasks respectively. Furthermore, employing shared parameters neural network can learn relative of tasks, and it can reduce over fitting risks on training. Experiments results show that our proposed method outperforms the state-of-the-art single task learning method on sentiment classification tasks.

In further work, we will design multi-task learning model with other additional features [1], develop the semi-supervised learning methods with much more unlabeled data resources. we also pay attention to the word-level attention mechanism to solve other text sequence problem, such question answering system, simultaneously.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (61472453, U1401256, U1501252, U1611264, U1711262, U1711261). And it is partially supported by NSSFC grant 13&ZD186, the China Key Research Program (grant no. 2016YFB1000905),the Guangxi Science Research and Technology Development Program (grants no. 15248003-8).

References

[1] G. Balikas, S. Moura, Multitask learning for fine-grained twitter sentiment analysis, in: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2017, pp. 1005–1008.
[2] Y. Bengio, N. Boulangerlewandowski, R. Pascanu, Advances in optimizing recurrent networks, in: International Conference on Acoustics, Speech, and Signal Processing, 2013, pp. 8624–8628.
[3] Y. Bengio, A.C. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828.

[4] Y. Bengio, P.Y. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Trans. Neural Netw. 5 (2) (1994) 157–166.
[5] N. Boulangerlewandowski, Y. Bengio, P. Vincent, Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription, in: International Conference on Machine Learning, 2012, pp. 1159–1166.
[6] S.R. Bowman, L. Vilnis, O. Vinyals, A.M. Dai, R. Jozefowicz, S. Bengio, Generating sentences from a continuous space, in: Conference on Computational Natural Language Learning, 2016, pp. 10–21.
[7] R. Caruana, Multitask learning, Mach. Learn. 28 (1) (1997) 41–75.
[8] T. Chen, R. Xu, Y. He, X. Wang, Improving sentiment analysis via sentence type classification using biLSTM-CRF and CNN, Expert Syst. Appl. 72 (2017) 221–230, doi:10.1016/j.eswa.2016.10.065.
[9] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, empirical methods in natural language processing (2014) 1724–1734.
[10] F. Chollet, et al., Keras, 2015, (https://github.com/fchollet/keras).
[11] J. Chung, K. Kastner, L. Dinh, K. Goel, A.C. Courville, Y. Bengio, A recurrent latent variable model for sequential data, in: Neural Information Processing Systems, 2015, pp. 2980–2988.
[12] R. Collobert, J. Weston, A unified architecture for natural language processing: deep neural networks with multitask learning, in: International Conference on Machine Learning, 2008, pp. 160–167.
[13] A. Conneau, H. Schwenk, L. Barrault, Y. Lecun, Very deep convolutional networks for natural language processing, arXiv: Computation and Language (2016).
[14] L. Deng, M.L. Seltzer, D. Yu, A. Acero, A. Mohamed, G.E. Hinton, Binary coding of speech spectrograms using a deep auto-encoder, in: Conference of the International Speech Communication Association, 2010, pp. 1692–1695.
[15] C. Doersch, Tutorial on variational autoencoders, arXiv: Machine Learning (2016).
[16] F.A. Gers, J. Schmidhuber, Recurrent nets that time and count, in: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, 3, 2000, pp. 189–194.
[17] I. Goodfellow, Y. Bengio, A. Courville, Deep learning, MIT Press, 2016. http://www.deeplearningbook.org.
[18] A. Graves, Generating sequences with recurrent neural networks, Neural Evol. Comput. (2013).
[19] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, J. Schmidhuber, A novel connectionist system for unconstrained handwriting recognition, IEEE Trans. Pattern Anal. Mach. Intell. 31 (5) (2009) 855–868.
[20] A. Graves, A. Mohamed, G.E. Hinton, Speech recognition with deep recurrent neural networks, in: International Conference on Acoustics, Speech, and Signal Processing, 2013, pp. 6645–6649.
[21] K. Gregor, I. Danihelka, A. Graves, D.J. Rezende, D. Wierstra, Draw: a recurrent neural network for image generation, in: International Conference on Machine Learning, 2015, pp. 1462–1471.
[22] L. Guangquan, L. Bo, Y. Weiwei, Y. Jian, Unsupervised feature selection with graph learning via low-rank constraint, Multimed. Tools Appl. (2017). https://doi.org/10.1007/s11042-017-5207-7.
[23] L. Gui, Y. Zhou, R. Xu, Y. He, Q. Lu, Learning representations from heterogeneous network for sentiment classification of product reviews, Knowl. Based Syst. 124 (2017) 34–45. https://doi.org/10.1016/j.knosys.2017.02.030.
[24] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Computer Vision and Pattern Recognition, 2015, pp. 770–778.
[25] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: European Conference on Computer Vision, 2016, pp. 630–645.
[26] R. He, J. Mcauley, Ups and downs: modeling the visual evolution of fashion trends with one-class collaborative filtering, in: International World Wide Web Conferences, 2016.
[27] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[28] Y. Kim, Convolutional neural networks for sentence classification, in: Empirical Methods in Natural Language Processing, 2014, pp. 1746–1751.
[29] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, in: International Conference on Learning Representations, 2015.
[30] D.P. Kingma, S. Mohamed, D.J. Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Neural Information Processing systems, 27, 2014, pp. 3581–3589.
[31] D.P. Kingma, M. Welling, Auto-encoding variational Bayes, in: International Conference on Learning Representations, 2014.
[32] J. Koutnik, K. Greff, F.J. Gomez, J. Schmidhuber, A clockwork RNN, in: International Conference on Machine Learning, 2014, pp. 1863–1871.
[33] Q.V. Le, T. Mikolov, Distributed representations of sentences and documents, in: International Conference on Machine Learning, 2014, pp. 1188–1196.
[34] C. Lei, X. Zhu, Unsupervised feature selection via local structure learning and sparse learning, Multimed. Tools Appl. (2017) https://doi.org/10.1007/s11042-017-5381-7.
[35] J. Liang, P. Liu, J. Tan, S. Bai, Sentiment classification based on AS-LDA model, Procedia - Procedia Comput. Sci. 31 (2014) 511–516, doi:10.1016/j.procs.2014.05.296.
[36] C. Liao, C. Feng, S. Yang, H. Huang, Neurocomputing topic-related chinese message sentiment analysis, Neurocomputing 210 (2016) 237–246, doi:10.1016/j.neucom.2016.01.110.
[37] B. Liu, Sentiment analysis and opinion mining, Synth. Lect. Human Lang. Technol. 5 (1) (2012) 1–10.
[38] P. Liu, X. Qiu, X. Huang, Deep multi-task learning with shared memory, in: Conference on Empirical Methods in Natural Language Processing, 2016.
[39] P. Liu, X. Qiu, X. Huang, Recurrent neural network for text classification with multi-task learning, in: International Joint Conference on Artificial Intelligence, 2016, pp. 2873–2879.
[40] X. Liu, J. Gao, X. He, L. Deng, K. Duh, Y. Wang, Representation learning using multi-task deep neural networks for semantic classification and information retrieval(2015) 912–921.
[41] L. Maaloe, C.K. Sonderby, S.K. Sonderby, O. Winther, Auxiliary deep generative models, in: International Conference on Machine Learning, 2016, pp. 1445–1453.
[42] M. Patidar, P. Agarwal, L. Vig, G. Shroff, Correcting linguistic training bias in an faqbot using lstm-vae, Interactions between Data Mining and Natural Language Processing, 2017.
[43] Y. Miao, L. Yu, P. Blunsom, Neural variational inference for text processing, in: International Conference on Machine Learning, 2016, pp. 1727–1736.
[44] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv: Computation and Language (2013a).
[45] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, neural information processing systems (2013) 3111–3119.
[46] R.H. Nielsen, Kolmogorov's mapping neural network existence theorem, in: Proceedings of the IEEE First International Conference on Neural Networks (San Diego, CA), III, Piscataway, NJ: IEEE, 1987, pp. 11–13.
[47] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up? Sentiment classification using machine learning techniques, in: Empirical Methods in Natural Language Processing, 2002, pp. 79–86.
[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[49] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[50] S. Poria, E. Cambria, A. Gelbukh, Aspect extraction for opinion mining with a deep convolutional neural network, Knowl. Based Syst. 108 (2016) 42–49.
[51] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533–536.
[52] J. Schmidhuber, A local learning algorithm for dynamic feed forward and recurrent networks, Conn. Sci. 1 (4) (1989) 403–412.
[53] Y.B. Sepp Hochreiter, Gradient flow in recurrent nets: the difficulty of learning longterm dependencies, Wiley-IEEE Press, 2001.
[54] W. Xu, H. Sun, C. Deng, Y. Tan, Variational autoencoder for semi-supervised text classification, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017.
[55] T. Wen, M. Gasic, N. Mrksic, P. Su, D. Vandyke, S.J. Young, Semantically conditioned LSTM-based natural language generation for spoken dialogue systems, in: Empirical Methods in Natural Language Processing, 2015, pp. 1711–1721.
[56] M. Wohlmayr, M. Stark, F. Pernkopf, A probabilistic interaction model for multipitch tracking with factorial hidden Markov models, IEEE Trans. Audio Speech Lang. Process. 19 (4) (2011) 799–810.
[57] Z. Yang, Z. Hu, R. Salakhutdinov, T. Berg-Kirkpatrick, Improved variational autoencoders for text modeling using dilated convolutions, in: International Conference on Machine Learning, 2017.
[58] K. Yao, T. Cohn, K. Vylomova, K. Duh, C. Dyer, Depth-gated lstm, Neural Evol. Comput. (2015).
[59] C. Zhang, Z. Zhang, Improving multiview face detection with multi-task deep convolutional neural networks, in: Workshop on Applications of Computer Vision, 2014, pp. 1036–1041.
[60] S. Zhang, X. Li, M. Zong, X. Zhu, R. Wang, Efficient KNN classification with different numbers of nearest neighbors, IEEE Trans. Neural Netw. Learn. Syst. 29 (5) (2018) 1774–1785.
[61] X. Zhang, J.J. Zhao, Y. Lecun, Character-level convolutional networks for text classification, Neural Inf. Process. Syst. (2015) 649–657.
[62] Z. Zhang, P. Luo, C.C. Loy, X. Tang, Facial landmark detection by deep multi-task learning, in: European Conference on Computer Vision, 2014, pp. 94–108.
[63] W. Zheng, X. Zhu, Y. Zhu, R. Hu, C. Lei, Dynamic graph learning for spectral feature selection, Multimed. Tools Appl. (2017) https://doi.org/10.1007/s11042-017-5272-y.
[64] X. Zhu, X. Li, S. Zhang, Z. Xu, L. Yu, C. Wang, Graph pca hashing for similarity search, IEEE Trans. Multimedia 19 (9) (2017) 2033–2044.
[65] X. Zhu, S. Zhang, R. Hu, Y. Zhu, et al., Local and global structure preservation for robust unsupervised spectral feature selection, IEEE Trans. Knowl. Data Eng. 30 (3) (2018) 517–529.
[66] Y. Zhu, X. Zhu, M. Kim, D. Shen, G. Wu, Early diagnosis of alzheimer's disease by joint feature selection and classification on temporally structured support vector machine, in: IPMI, 2016, pp. 264–272.