Neural Process Lett (2017) 45:173–182
DOI 10.1007/s11063-016-9519-9

Learning Invariant Features Using Subspace Restricted Boltzmann Machine

Jakub M. Tomczak[1] · **Adam Gonczarek**[1]

Published online: 7 April 2016

Abstract The subspace restricted Boltzmann machine (subspaceRBM) is a third-order Boltzmann machine where multiplicative interactions are between one visible and two hidden units. There are two kinds of hidden units, namely, *gate units* and *subspace units*. The subspace units reflect variations of a pattern in data and the gate unit is responsible for activating the subspace units. Additionally, the gate unit can be seen as a pooling feature. We evaluate the behavior of subspaceRBM through experiments with MNIST digit recognition task and Caltech 101 Silhouettes image corpora, measuring cross-entropy reconstruction error and classification error.

Keywords Feature learning · Unsupervised learning · Invariant features · Subspace features · Deep model

1 Introduction

The success of machine learning methods stems from appropriate data representation. Clearly this requires applying feature engineering, i.e., handcrafted proposition of a set of features potentially useful in the considered problem. However, it would be beneficial to propose an automatic features extraction to avoid any awkward preprocessing pipelines for hand-tuning of the data representation [2]. Deep learning turns out to be a suitable fashion of automatic representation learning in many domains such as object recognition [23], speech recognition [20], natural language processing [7], neuroimaging [13] multimodal learning from images and text annotations [27], pose recovery [29], or domain adaptation [9].

Fairly simple but still one of the most popular models for unsupervised feature learning is the restricted Boltzmann machine (RBM). Except automatic feature learning, RBMs can be stacked in a hierarchy to form a deep network [1]. The bipartie structure of the RBM

✉ Jakub M. Tomczak
jakub.tomczak@pwr.edu.pl

[1] Department of Computer Science, Faculty of Computer Science and Management, Wrocław University of Science and Technology, wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

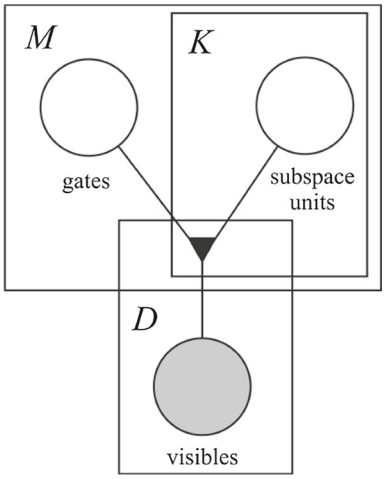

Fig. 1 A graphical representation of the subspaceRBM. The *triangular symbol* represents a third-order multiplicative interaction

enables block Gibbs sampling which allows formulating efficient learning algorithms such as contrastive divergence [10]. However, lately it has been argued that the RBM fails to properly reflect statistical dependencies [22]. One possible solution is to apply higher-order Boltzmann machine [17,24] to model sophisticated patterns in data.

In this work we follow this line of thinking and develop a more refined model than the RBM to learn features from data. Our model introduces two kinds of hidden units, i.e., *subspace units* and *gate units* (see Fig. 1). The subspace units are hidden variables which reflect variations of a feature and thus they are more robust to invariances. The gate units are responsible for activating the subspace units and they can be seen as pooling features composed of the subspace features. The proposed model is based on an energy function with third-order interactions and maintains the conditional independence structure that can be readily used in simple and efficient learning.

The paper is organized as follows. In Sect. 2 the proposed new model is presented. In Sect. 3, the learning procedure of subspaceRBM is outlined. In Sect. 4, we relate our approach to other deep models. Next, in Sect. 5, the proposed model is evaluated empirically on two image corpora and the results are discussed. Finally, in Sect. 6, the conclusions are drawn and future research are indicated.

2 The Model

The RBM is a second-order Boltzmann machine with restriction on within-layer connections. This model can be extended in a straightforward way to third-order multiplicative interactions of one visible x_i and two types of hidden binary units, a gate unit h_j and a subspace unit s_{jk}. Each gate unit is associated with a group of subspace hidden units. The energy function of a joint configuration is defined as follows:

$$E(\mathbf{x},\mathbf{h},\mathbf{S}|\boldsymbol{\theta}) = -\sum_{i=1}^{D}\sum_{j=1}^{M}\sum_{k=1}^{K} W_{ijk}x_i h_j S_{jk} - \sum_{i=1}^{D} b_i x_i - \sum_{j=1}^{M} c_j h_j - \sum_{j=1}^{M} h_j \sum_{k=1}^{K} D_{jk} S_{jk}. \tag{1}$$

where $\mathbf{x} \in \{0,1\}^D$ denotes a vector of visible variables, $\mathbf{h} \in \{0,1\}^M$ is a vector of gate units, $\mathbf{S} \in \{0,1\}^{M \times K}$ is a matrix of subspace units, the parameters are $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{b}, \mathbf{c}, \mathbf{D}\}$, $\mathbf{W} \in \mathbb{R}^{D \times M \times K}$ is a weight tensor, $\mathbf{b} \in \mathbb{R}^D$ is a vector of visible biases, $\mathbf{c} \in \mathbb{R}^M$ is a vector of gate biases, and $\mathbf{D} \in \mathbb{R}^{M \times K}$ is a matrix of subspace biases.

The model defined by the Gibbs distribution with the energy function as in Eq. 1, that is:

$$p(\mathbf{x}, \mathbf{h}, \mathbf{S}|\boldsymbol{\theta}) = \frac{1}{Z(\boldsymbol{\theta})} \exp\{-E(\mathbf{x}, \mathbf{h}, \mathbf{S}|\boldsymbol{\theta})\}, \tag{2}$$

where

$$Z(\boldsymbol{\theta}) = \sum_{\mathbf{x},\mathbf{h},\mathbf{S}} \exp\{-E(\mathbf{x}, \mathbf{h}, \mathbf{S}|\boldsymbol{\theta})\} \tag{3}$$

is a partition function, is further called *subspace restricted Boltzmann machine* (subspaceRBM).

For the subspaceRBM the following conditional dependencies hold true:[1,2]

$$p(x_i = 1|\mathbf{h}, \mathbf{S}) = \text{sigm}\left(\sum_j \sum_k W_{ijk} h_j S_{jk} + b_i\right), \tag{4}$$

$$p(s_{jk} = 1|\mathbf{x}, h_j) = \text{sigm}\left(\sum_i W_{ijk} x_i h_j + h_j D_{jk}\right), \tag{5}$$

$$p(h_j = 1|\mathbf{x}) = \text{sigm}\left(-K\log 2 + c_j + \sum_{k=1}^{K} \text{softplus}\left(\sum_i W_{ijk} x_i + D_{jk}\right)\right), \tag{6}$$

which can be straightforwardly used in formulating a contrastive divergence learning algorithm. Notice that in Eq. 6 a term $-K\log 2$ influences the hidden unit activation which is linear to the number of subspace hidden variables. Moreover, the probability of an example $\mathbf{x}$ is as follows:

$$p(\mathbf{x}) \propto \exp\left(\sum_i b_i x_i + \sum_j \log\left[2^K + \exp(c_j) \prod_k \left(1 + \exp\left(\sum_i W_{ijk} x_i + D_{jk}\right)\right)\right]\right). \tag{7}$$

3 Learning

In training, we take advantage of the Eqs. 4, 5, and 6 to formulate an efficient three-phase block-Gibbs sampling from the subspaceRBM (see Algorithm 1). First, for given data, we sample gate units from $p(\mathbf{h}|\mathbf{x})$ with $\mathbf{S}$ marginalized out. Then, given both $\mathbf{x}$ and $\mathbf{h}$, we can sample subspace variables from $p(\mathbf{S}|\mathbf{x}, \mathbf{h})$. Eventually, the data can be sampled from $p(\mathbf{x}|\mathbf{h}, \mathbf{S})$.

We update the parameters of the subspaceRBM using contrastive divergence learning procedure [8,10]. For this purpose, we need to calculate the gradient of the log-likelihood function. The log-likelihood gradient takes the form of a difference between two expectations, namely, over the probability distribution with clamped data, and over the joint probability

[1] $\text{sigm}(a) = \frac{1}{1+\exp(-a)}$.

[2] $\text{softplus}(a) = \log\left(1 + \exp(a)\right)$.

distribution of visible and hidden variables. Analogously to the standard RBM, in the subspaceRBM these two expectations are approximated by samples drawn from the three-phase block-Gibbs sampling procedure.

Algorithm 1: One iteration (epoch) of Stochastic Gradient Algorithm with contrastive divergence gradient approximation for subspaceRBM.

Input : Training data consisting of N examples $\{\mathbf{x}_n\}_{n=1}^N$, model parameters $\{\mathbf{W}, \mathbf{b}, \mathbf{c}, \mathbf{D}\}$, learning rate α.

for $n = 1 \rightarrow N$ **do**
  % Positive phase
  For given $\mathbf{x}_n$ generate $\hat{\mathbf{h}}$ using (6).
  For given $\mathbf{x}_n$ and $\hat{\mathbf{h}}$ generate $\hat{\mathbf{S}}$ using (5).
  % Negative phase
  For given $\hat{\mathbf{h}}$ and $\hat{\mathbf{S}}$ generate $\tilde{\mathbf{x}}$ using (4).
  For given $\tilde{\mathbf{x}}$ generate $\tilde{\mathbf{h}}$ using (6).
  For given $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{h}}$ generate $\tilde{\mathbf{S}}$ using (5).
  % Update
  for $j = 1 \rightarrow M$ **do**
    for $k = 1 \rightarrow K$ **do**
      for $i = 1 \rightarrow D$ **do**
        $W_{ijk} \longleftarrow W_{ijk} + \alpha \left(x_{i,n}\hat{h}_j\hat{S}_{jk} - \tilde{x}_i\tilde{h}_j\tilde{S}_{jk}\right)$
      end
      $D_{jk} \longleftarrow D_{jk} + \alpha \left(\hat{h}_j\hat{S}_{jk} - \tilde{h}_j\tilde{S}_{jk}\right)$
    end
  end
  $\mathbf{b} \longleftarrow \mathbf{b} + \alpha \left(\mathbf{x}_n - \tilde{\mathbf{x}}\right)$
  $\mathbf{c} \longleftarrow \mathbf{c} + \alpha \left(\hat{\mathbf{h}} - \tilde{\mathbf{h}}\right)$
end

4 Related Works

The standard RBM can reflect only the second-order multiplicative interactions. However, in many real-life situations, higher-order interactions must be included if we want our model to be effective enough. Moreover, often the second-order interactions themselves might represent little or no useful information. In the literature there were several propositions of how to extend the RBM to the higher-order Boltzmann machines. One such proposal is a third-order multiplicative interaction of two visible binary units x_i, $x_{i'}$ and one hidden binary unit h_j [11,22], which can be used to learn a representation robust to spatial transformations [19]. Along this line of thinking, our model is the third-order Boltzmann machine but with different multiplicative interactions of one visible unit and two kinds of hidden units.

The proposed model is closely related to the special kind of *spike-and-slab restricted Boltzmann machine* [6] called the *subspace spike-and-slab RBM* (subspace-ssRBM) [5] where there are two kinds of hidden variables, namely, *spike* is a binary variable and *slab* is a real-valued variable. However, in our approach both the spike and slab variables are discrete. Additionally, in the subspaceRBM the hidden units $\mathbf{h}$ behave as gates to subspace variables rather than spikes as in ssRBM.

Similarly to our approach, gating units were proposed in the *Point-wise Gated Boltzmann Machine* (PGBM) [25] where chosen units were responsible for switching on subsets of

hidden units. The subspaceRBM is based on an analogous idea but it uses sigmoid units only whereas PGBM utilizes both sigmoid and softmax units.

Our model can be also related to RBM forests [15]. The RBM forests assume each hidden unit to be encoded by a complete binary tree. In our approach each gate unit is encoded by subspace units. Therefore, the subspaceRBM can be seen as a RBM forest but with flatter hierarchy of hidden units and hence easier learning and inference.

Lastly, the subspaceRBM but with the softmax hidden units $\mathbf{h}$ turns to be the implicit mixture of RBMs (imRBM) [21]. However, in our model the gate units can be seen as pooling features while in the imRBM they determine only one subset of subspace features to be activated. The subspaceRBM brings an important benefit over the imRBM because it allows the subspaceRBM to reflect multiple factors in data.

5 Experiment

Goal In this paper, we present a new model for capturing binary inputs that can be further used as a building block in a deep network. Typical building block of a deeper architecture is the RBM. Therefore, in the experiment we aim at answering the following question:

– Is the subspaceRBM preferable to the RBM in terms of reconstruction error and as a better feature extractor?

We want to point out that we verify whether the subspaceRBM can be treated as a better alternative to the RBM. We believe that the positive answer to the stated question will give us a good starting point for further experiments with deep models using the subspaceRBM.

Data We performed the experiment using CalTech 101 28×28 Silhouettes[3] (CalTech, for the sake of brevity), and MNIST.[4] CalTech dataset consists of 4100 training images, 2264 validation images, and 2307 test images. In the dataset the objects are centered and scaled on a 28×28 image plane and rendered as filled black regions on a white background [18]. MNIST consists of 28×28 images representing hand-written digits from 0 through 9 [16]. The data is divided into 50,000 training examples, 10,000 validation images, and the test set contains 10,000 examples. In the experiments, we performed learning with different number of training images (10,100, and 1000 per digit) and the full training set.

Training protocol In the experiment, we compared the subspaceRBM with the RBM for the number of gate units equal $M = 500$ and different number of subspace units $K \in \{3, 5, 7\}$. The subspaceRBM was trained using the presented contrastive divergence (see Algorithm 1) and a minibatch of size 10 was used. In order to choose the value of the learning rate we performed the model selection using the validation set and the learning rate was $\{0.001, 0.01, 0.1\}$. The number of iterations (epochs) over the training set was determined using early stopping according to the validation set cross-entropy reconstruction error, with a look ahead of 5 iterations.

The RBM was used with 500, 1500, 2500 and 3500 hidden units, which corresponds to the same number of gates units multiplied by the number of subspace units in the subspaceRBM. The RBM was trained using the contrastive divergence with 1-step Gibbs sampling. The learning rate was determined using the model selection using the validation set and its possible

[3] http://www.cs.ubc.ca/~bmarlin/data.shtml.

[4] http://yann.lecun.com/exdb/mnist/.

Table 1 Average test classification error with one standard deviation for the RBM and different settings of the subspaceRBM evaluated on subsets of MNIST

Model	Classification error (%)			
	$N = 100$	$N = 1000$	$N = 10{,}000$	$N = 50{,}000$
RBM $M = 500$	24.20 ± 0.53	8.31 ± 0.31	3.78 ± 0.11	3.23 ± 0.14
RBM $M = 1500$	25.83 ± 0.75	8.17 ± 0.17	3.06 ± 0.02	2.12 ± 0.02
RBM $M = 2500$	27.40 ± 0.86	***7.77*** ± 0.11	3.01 ± 0.05	1.94 ± 0.12
RBM $M = 3500$	30.65 ± 0.44	8.28 ± 0.13	***2.93*** ± 0.04	***1.8*** ± 0.01
subspaceRBM $M = 500$, $K = 3$	***23.78*** ± 0.53	8.95 ± 0.50	4.27 ± 0.62	3.81 ± 0.19
subspaceRBM $M = 500$, $K = 5$	24.40 ± 0.37	**8.33** ± 0.13	3.69 ± 0.03	3.18 ± 0.14
subspaceRBM $M = 500$, $K = 7$	24.41 ± 1.23	8.75 ± 0.04	**3.62** ± 0.02	**2.63** ± 0.04

The best results of subspaceRBM are in bold and the best results among all models are in italic and bold

values were the same as in the case of the subspaceRBM. Similarly to the subspaceRBM, the early stopping procedure was used with looking ahead of 5 epochs.

We evaluated the subspaceRBM as a feature-extraction scheme by plugging it into the classification pipeline developed by [4]. For classification the logistic regression[5] used the probabilities of gate units, $p(h_j = 1|\mathbf{x})$, as inputs. Analogously was done for the RBM.

We did 3 full runs for each dataset and averaged the results.

Evaluation methodology The performance of the subspaceRBM and the RBM was measured using cross-entropy reconstruction error (reconstruction error, for the sake of brevity), classification error, and mean number of active gate units. The cross-entropy reconstruction error for original object $\mathbf{x}$ and its reconstruction $\tilde{\mathbf{x}}$ is defined as follows:

$$L(\mathbf{x}, \tilde{\mathbf{x}}) = -\big(\mathbf{x} \log \tilde{\mathbf{x}} + (1 - \mathbf{x}) \log(1 - \tilde{\mathbf{x}})\big), \tag{8}$$

where $\tilde{\mathbf{x}}$ is a reconstruction calculated by first sampling hidden units for given data using equations (5) and (6), and further sampling visible variables for sampled hidden units using (4). Similarly, in the case of RBM, the reconstruction is calculated in an analogical manner.

It has been advocated that the cross-entropy reconstruction error is a good proxy of the log-likelihood while using contrastive divergence learning [1].

We would like to highlight that in the experiment we aim at evaluating capabilities of the proposed model and comparing it with the RBM. Therefore, we resigned from applying sophisticated learning techniques, e.g., weight decay, momentum term, sparsity regularization [12]. We believe that application of more advanced training protocol could disrupt this comparison. As a consequence, we have obtained results that were worst than current state-of-the-art but these allow to evaluate mainly models instead of learning algorithms.

5.1 Results

MNIST The averaged results with one standard deviation of the subspaceRBM and the RBM are presented in Table 1 (for test classification error), in Table 2 (for test reconstruction error), and the average number of active units calculated on test data is outlined in Table 3. A random subset of subspace features for the subspaceRBM ($M = 500$, $K = 7$) trained on 50,000 images is shown in Fig. 2.

[5] The ℓ_2 regularization was applied with the regularization coefficient equal $\lambda \in \{0, 0.01, 0.1\}$.

Table 2 Average test reconstruction error with one standard deviation for different settings of the RBM and the subspaceRBM evaluated on subsets of MNIST

Model	Reconstruction error			
	$N = 100$	$N = 1000$	$N = 10{,}000$	$N = 50{,}000$
RBM $M = 500$	140.35 ± 9.31	87.15 ± 3.29	75.03 ± 2.57	73.41 ± 0.59
RBM $M = 1500$	144.47 ± 16.76	89.88 ± 0.59	75.13 ± 0.057	72.37 ± 0.22
RBM $M = 2500$	161.75 ± 11.32	88.00 ± 0.51	75.98 ± 0.11	71.98 ± 0.39
RBM $M = 3500$	230.72 ± 8.48	91.71 ± 0.08	75.76 ± 0.27	71.17 ± 0.01
subspaceRBM $M = 500$, $K = 3$	123.28 ± 2.35	82.26 ± 1.43	71.76 ± 0.89	71.57 ± 0.54
subspaceRBM $M = 500$, $K = 5$	$\mathbf{121.27} \pm 1.26$	$\mathbf{81.66} \pm 0.75$	71.86 ± 0.43	69.35 ± 1.54
subspaceRBM $M = 500$, $K = 7$	123.70 ± 0.77	82.67 ± 0.76	$\mathbf{71.23} \pm 1.06$	$\mathbf{68.64} \pm 1.56$

The best results are in bold

Table 3 Number of active units for the RBM and different settings of the subspaceRBM evaluated on subsets of MNIST

Model	Number of active units			
	$N = 100$	$N = 1000$	$N = 10{,}000$	$N = 50{,}000$
RBM $M = 500$	70	65	50	37
RBM $M = 1500$	60	60	43	36
RBM $M = 2500$	45	60	44	36
RBM $M = 3500$	32	63	44	34
subspaceRBM $M = 500$, $K = 3$	62	93	85	120
subspaceRBM $M = 500$, $K = 5$	58	74	78	107
subspaceRBM $M = 500$, $K = 7$	41	66	66	79

CalTech The summary results of the performance of the subspaceRBM and the RBM are presented in Table 4. A random subset of subspace features for the subspaceRBM ($M = 500$, $K = 3$) is shown in Fig. 3.

5.2 Discussion

We notice that application of subspace units is beneficial for better reconstruction capabilities (see Tables 2 and 4). For classification it is advantageous to use subspaceRBM in the case of small sample size regime (for MNIST dataset with N equal 100 and 1000, and for CalTech data, see Tables 1 and 4) with smaller number of subspace units. However, this result is rather not surprising because for over-complete representations simpler classifiers work better. On the other hand, for the small sample size there is a big threat of overfitting. Introducing subspace units to the hidden layer restricts the variability of the representation and thus preventing from learning noise in data. In the case of classification for larger number of observations (for MNIST data with N equal 10,000 and 50,000, see Table 1), best results were obtained for K equal 5 and 7. This result suggests that indeed the subspace units lead to features that are more robust to small perturbations.

Comparing the subspaceRBM to the RBM with comparable size, i.e., $M \in \{1500, 2500, 3500\}$, it turns out that in terms of the classification error RBMs with larger number of

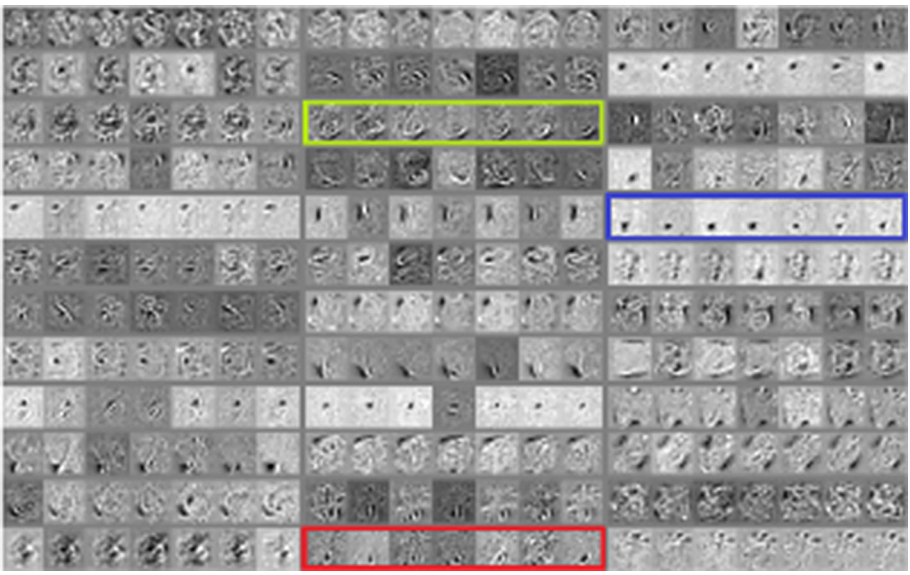

Fig. 2 Random subset of subspace features for MNIST ($N = 50{,}000$) and the subspaceRBM with $M = 500$ and $K = 7$. Relevant three groups of filters are outlined in *red*, *blue* and *green* which evidently tend to learn similar pattern with offsets in position, curvature or rotation. (Color figure online)

Table 4 Average test results with one standard deviation for different settings of the RBM and the subspaceRBM evaluated on CalTech

Model	Classification error (%)	Reconstruction error	Number of active units
RBM $M = 500$	35.03 ± 0.38	103.05 ± 2.16	85
RBM $M = 1500$	37.34 ± 1.06	115.68 ± 6.28	85
RBM $M = 2500$	40.29 ± 1.71	105.26 ± 7.03	82
RBM $M = 3500$	46.97 ± 7.82	110.58 ± 8.34	75
subspaceRBM $M = 500$, $K = 3$	$\mathbf{34.51} \pm 0.16$	69.09 ± 10.93	202
subspaceRBM $M = 500$, $K = 5$	35.97 ± 0.81	$\mathbf{66.76} \pm 8.42$	175
subspaceRBM $M = 500$, $K = 7$	37.37 ± 1.44	67.99 ± 5.53	112

The best results are in bold

hidden units obtained much better results. However, this result follows from the fact that it is easier to discriminate if there are more available features. Of course, this statement is true only if the features represent reasonable patterns (i.e., different than noise), and the sample size is appropriate (see Table 1 for $N = 100$ and Table 4 where larger RBMs tend to be heavily overfitted). Nonetheless, the reconstruction error for any dataset is in favor of the subspaceRBM. This effect can be explained as follows. During reconstructing data lots of features are useless but they still contribute to the reconstruction but rather as a source of noise. Therefore, the more features are in a model, the more noise is incorporated to the reconstruction. However, it seems that the larger number of subspace units results in better reconstruction. This means that the subspaceRBM indeed captures different forms of a feature and incorporating more subspace units is beneficial.

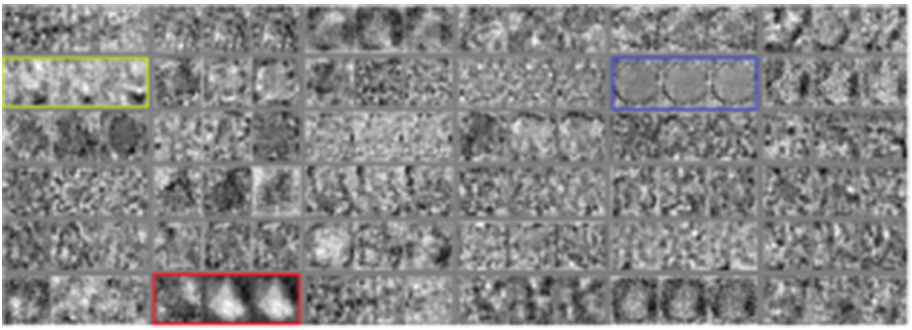

Fig. 3 Random subset of subspace features for CalTech and the subspaceRBM with $M = 500$ and $K = 3$. Relevant three groups of filters are outlined in *red*, *blue* and *green* which evidently tend to learn similar pattern with offsets in position, curvature or rotation. (Color figure online)

Eventually, it is worth noticing that on average the number of active hidden units is higher for the subspaceRBM in comparison to the RBM. This result may be explained by the sum of softplus terms used in calculating the conditional probability (see Eq. 6). The effect of increased activity of hidden units is especially apparent in the case of CalTech where on average about half of gate units are active (see Table 2).

6 Conclusion

In this paper, we have proposed an extension of the RBM by introducing subspace hidden units. The formulated model can be seen as the third-order Boltzmann machine with third-order multiplicative interactions. We have showed that the subspaceRBM does not reduce to a vanilla version of the RBM (see Eq. 7). The carried-out experiments have revealed that the proposed model is advantageous over the RBM in terms of reconstruction and classification error.

We see several possible extensions of the outlined approach. In our opinion, the examination of the effect of high activity of gate units is very appealing. It has been advocated [9] that sparse activity of hidden units provides more robust representation, therefore, we plan to apply some kind regularization enforcing sparsity [14] or features robustness [26]. Moreover, it would be beneficial to utilize other learning algorithms instead of the contrastitve divergence, such as, sampling methods [3], score matching [28] and other inductive principles, e.g., Maximum Pseudo-Likelihood [18]. Last but not least, subspaceRBM can be used as a building block in a deep model. However, we leave investigation of stated issues as future research.

Acknowledgements The research conducted by the authors has been partially co-financed by the Ministry of Science and Higher Education, Republic of Poland, namely, Jakub M. Tomczak: Grant No. B50106W8/K3, Adam Gonczarek: Grant No. B50137W8/K3.

References

1. Bengio Y (2009) Learning deep architectures for AI. Found Trends Mach Learn 2(1):1–127
2. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828
3. Brügge K, Fischer A, Igel C (2013) The flip-the-state transition operator for restricted Boltzmann machines. Mach Learn 93(1):53–69
4. Coates A, Ng AY, Lee H (2011) An analysis of single-layer networks in unsupervised feature learning. In: International conference on artificial intelligence and statistics, pp 215–223
5. Courville A, Desjardins G, Bergstra J, Bengio Y (2014) The spike-and-slab RBM and extensions to discrete and sparse data distributions. IEEE Trans Pattern Anal Mach Intell 36(9):1874–1887
6. Courville AC, Bergstra J, Bengio Y (2011) A spike and slab restricted Boltzmann machine. In: International conference on artificial intelligence and statistics, pp 233–241
7. Dahl GE, Adams RP, Larochelle H (2012) Training restricted Boltzmann machines on word observations. In: Proceedings of the 29th international conference on machine learning
8. Fischer A, Igel C (2014) Training restricted Boltzmann machines: an introduction. Pattern Recognit 47(1):25–39
9. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier networks. In: Proceedings of the 14th international conference on artificial intelligence and statistics, vol 15, pp 315–323
10. Hinton GE (2002) Training products of experts by minimizing contrastive divergence. Neural Comput 14(8):1771–1800
11. Hinton GE (2010) Learning to represent visual input. Philos Trans R Soc B 365(1537):177–184
12. Hinton GE (2012) A practical guide to training restricted Boltzmann machines. In: Montavon G, Orr GB, Müller K-R (eds) Neural networks: tricks of the trade. Springer, Berlin, pp 599–619
13. Hjelm RD, Calhoun VD, Salakhutdinov R, Allen EA, Adali T, Plis SM (2014) Restricted Boltzmann machines for neuroimaging: an application in identifying intrinsic networks. NeuroImage 96:245–260
14. Ji N, Zhang J, Zhang C, Yin Q (2014) Enhancing performance of restricted Boltzmann machines via log-sum regularization. Knowl Based Syst 63:82–96
15. Larochelle H, Bengio Y, Turian J (2010) Tractable multivariate binary density estimation and the restricted Boltzmann forest. Neural Comput 22(9):2285–2307
16. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
17. Leisink MA, Kappen HJ (2000) Learning in higher order Boltzmann machines using linear response. Neural Netw 13(3):329–335
18. Marlin BM, Swersky K, Chen B, Freitas ND (2010) Inductive principles for restricted Boltzmann machine learning. In: International conference on artificial intelligence and statistics, pp 509–516
19. Memisevic R, Hinton GE (2010) Learning to represent spatial transformations with factored higher-order Boltzmann machines. Neural Comput 22(6):1473–1492
20. Mohamed AR, Dahl GE, Hinton G (2012) Acoustic modeling using deep belief networks. IEEE Trans Audio Speech Lang Process 20(1):14–22
21. Nair V, Hinton GE (2008) Implicit mixtures of restricted Boltzmann machines. NIPS 21:1145–1152
22. Ranzato M, Krizhevsky A, Hinton GE, et al (2010) Factored 3-way restricted Boltzmann machines for modeling natural images. In: International conference on artificial intelligence and statistics, pp 621–628
23. Salakhutdinov R, Tenenbaum JB, Torralba A (2013) Learning with hierarchical-deep models. IEEE Trans Pattern Anal Mach Intell 35(8):1958–1971
24. Sejnowski TJ (1986) Higher-order Boltzmann machines. AIP Conf Proc 151:398–403
25. Sohn K, Zhou G, Lee C, Lee H (2013) Learning and selecting features jointly with point-wise gated Boltzmann machines. In: Proceedings of The 30th international conference on machine learning, pp 217–225
26. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
27. Srivastava N, Salakhutdinov R (2014) Multimodal learning with deep Boltzmann machines. J Mach Learn Res 15:2949–2980
28. Swersky K, Ranzato M, Buchman D, Marlin BM, Freitas ND (2011) On autoencoders and score matching for energy based models. In: Proceedings of the 28th international conference on machine learning (ICML-11), pp 1201–1208
29. Tompson J, Stein M, LeCun Y, Perlin K (2014) Real-time continuous pose recovery of human hands using convolutional networks. ACM Trans Gr 33(5):169:1–169:10