Learning Deep and Wide: A Spectral Method for Learning Deep Networks

Ling Shao, *Senior Member, IEEE*, Di Wu, and Xuelong Li, *Fellow, IEEE*

***Abstract*— Building intelligent systems that are capable of extracting high-level representations from high-dimensional sensory data lies at the core of solving many computer vision-related tasks. We propose the multispectral neural networks (MSNN) to learn features from multicolumn deep neural networks and embed the penultimate hierarchical discriminative manifolds into a compact representation. The low-dimensional embedding explores the complementary property of different views wherein the distribution of each view is sufficiently smooth and hence achieves robustness, given few labeled training data. Our experiments show that spectrally embedding several deep neural networks can explore the optimum output from the multicolumn networks and consistently decrease the error rate compared with a single deep network.**

***Index Terms*— Deep networks, multispectral embedding, representation learning.**

I. Introduction

Recent publications suggest that unsupervised pretraining of deep, hierarchical neural networks improves supervised pattern classification [1]–[4]. Learning machines that are able to automatically build feature extractors instead of hand-crafting them is a wide research area in pattern recognition. The main benefit of these models is their high generation since they can automatically learn to extract salient patterns directly from the raw input, without any use of prior knowledge. Recent advancement and applications using learned features have yielded excellent results in several tasks, e.g., object recognition and video sequence classification. Krizhevsky *et al.* [5] train a large, deep convolutional neural network (CNN) to classify 1000 different classes; Baccouche *et al.* [6] learn a sparse shift-invariant representation of the local salient information using a spatio-temporal convolutional sparse autoencoder, without any use of prior knowledge, and classify each sequence by a long short-term memory recurrent neural network [7]. Meanwhile, various architectures and techniques have been proposed to enhance the learning capacity: a multiresolution deep belief network (DBN) [8] combines a Laplacian pyramid with deep learning to learn coarse structures from low-resolution images, leading to a better generative model; multicolumn deep neural networks proposed by Cireşan *et al.* [9]–[11] use GPUs to train several deep neural columns and average the output of each individual network under the condition that given enough labeled data, their networks do not need additional heuristics, such as unsupervised pretraining or carefully prewired synapses.

Inspired by microcolumns of neurons in the cerebral cortex, several deep neural columns are trained and become experts to unfold their potential when they are wide. Conventional multicolumn deep neural networks average the output of the prediction under the condition that there are enough labeled training data and an individual neural network is close to the global optimum. However, simple output averaging may not achieve the model's optimum if only few labeled data are provided. As indicated in [9], several deep neural columns are trained to become experts to unfold their potential when they are wide. However, if the labeled training instances are few, i.e., fine-tuning information is scarce, the deep networks can suffer from overfitting. Such a setting is pervasive in real-world applications, such as the gender prediction (Section IV-C), where the randomized, controlled experiments may be costly, unethical, and intrusive.

In this brief, we show how combining several deep network columns as a basic building block into the multicolumn deep nets and embedding the spectral relationships can further enhance robustness and hence decrease the error rate. We define the wide deep net as the juxtaposition of multiple randomly initialized nonconvex deep nets, and refer to our proposed architecture as multispectral neural networks (MSNN). The multicolumn procedure can be easily implemented in a parallelized, multithreaded fashion that requires no significant extra training time for MSNN. Our architecture does this by combining several techniques in a novel way.

1) Through encouraging the neural networks to learn deep models reusing intermediate features to extract more abstract representations that are more correlated with the underlying causes generating the data, we utilize the penultimate layer of the hierarchy as our intermediate feature space in contrast to the paradigm that outputs the top predictor layer (also known as softmax output layer). Such nets can be DBNs or CNNs with fully connected penultimate layers.
2) Our architecture renders the networks to learn wide horizontally to explore the feature space admitting stochasticity of the deep nets, rendering a mixture-of-experts style field. Unlike the conventional multicommittee systems that extract the trivial 1-D winner-take-all regions, that is, the top part of the hierarchy

Manuscript received July 21, 2013; accepted February 22, 2014. Date of publication March 11, 2014; date of current version November 17, 2014. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2012CB316400, in part by the University of Sheffield, in part by the China Scholarship Council, in part by the National Natural Science Foundation of China under Grant 61125106, and in part by the Shaanxi Key Innovation Team of Science and Technology under Grant 2012KCT-04.

L. Shao is with the College of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China, and also with the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield S1 3JD, U.K. (e-mail: ling.shao@sheffield.ac.uk).

D. Wu is with the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield S1 3JD, U.K. (e-mail: stevenwudi@gmail.com).

X. Li is with the Center for Optical Imagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: xuelong_li@opt.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2014.2308519

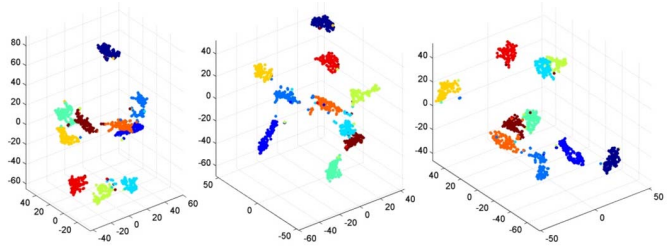

Fig. 1. Visualization of three individual nets penultimate layers using t-SNE on the MNIST data set. Different classes are color coded. All three nets are of the same architecture but with different initializations, and it can be seen that the resulting embeddings lie in quite distinct subspaces, albeit in the identical network architecture.

becomes a standard multilayer perceptron [9], we feed the intermediate representation into a multispectral graph Laplacian to explore the complementary property of intermediate representations wherein the distribution of each view is sufficiently smooth.

3) We resolve the open issue in [12] pertaining to the selection of the optimal dimension of the low-dimensional embedding from a graph cut point of view. In particular, when applying the deep architecture to representation learning, our approach successfully utilizes the smoothness of the intermediate feature, albeit the possible noise and degeneracies in the raw feature space cause the malfunction of the spectral clustering.

II. Architecture and Motivations

A. Problem Formulation: Randomness and Local Optimum

We first give an intuitive view of greedy learning for DBN and then generalize to other deep learning paradigms. The learning rule for a restricted Boltzmann machine is much more closely approximating the gradient of another objective function that is the difference between two Kullback–Leibler divergences [13]. The learning works well, even though it is only crudely approximating the gradient of the log probability of the training data [14]. However, it ignores one tricky term in this objective function, so it is not even following that gradient. Indeed, Sutskever and Tieleman [15] have shown that it is not following the gradient of any function. Nevertheless, constrative divergence learning guarantees to find a local optimum, and is good enough to achieve success in many significant applications. When collecting the pairwise statistics for learning weights or the individual statistics for learning biases, the weights have random initial values to break symmetry. The learning raises the effective mixing rate. As pointed out in [16], if the learning rate is sufficiently small compared with the mixing rate of the Markov chain, the persistent chain will stay very close to the stationary distribution, and the chain will mix before the weights have changed enough to significantly alter the unconditional expectation with a small learning rate. Therefore, the fantasy particle [16] of these chains cannot fully explore the feature space. Fig. 2(a) illustrates that when using the persistent Markov chain to estimate the model's expectations, particles will be trapped in a local optimum given high-dimensionality of the feature space, which is inevitable for computer vision-related tasks.

To overcome this explaining away effect, Cireşan *et al.* [9] pointed out that several deep neural columns that are randomly initialized can be trained, admitting the local optimum resulting from stochasticity of the network, to become experts to unfold their potential when they are wide. Fig. 1 shows the t-SNE feature [17] of the penultimate layers from three convolutional neural nets with the same architecture, but different random initializations can result in vastly different embeddings. A democratically output-averaged model, also known as

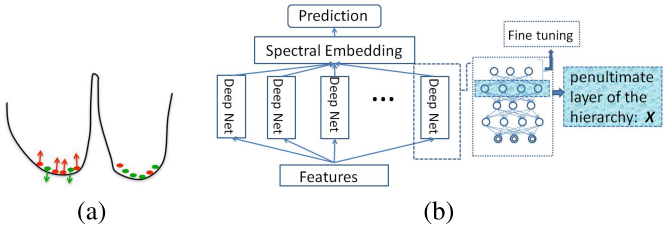

Fig. 2. (a) Wherever the fantasy particles (red dots) [16] outnumber the positive data (green dots), the energy surface is raised. This makes the fantasies rush around hyperactively, and they move around much faster than the mixing rate of the Markov chain defined by the static current weights. (b) Architecture: features are fed into multicolumn deep nets, penultimate layers of the hierarchies are fused into a multispectral space, and the embedded features are used for prediction.

late fusion, is adopted in [9], and it serves well when enough labeled training data are given. However, if the labeled training instances are limited, i.e., fine-tuning information is scarce, we argue that the deep network can suffer from overfitting.

To encode several different features in a unified embedding, multiview spectral embedding [12] can find such a smooth embedding. However, the input feature space should be physically meaningful and discriminative enough so that the zero-eigenvalues of the graph Laplacian will be correspondent to the connected components [18]. Previous works rely on the so-called handcrafted features that are manually designed to be optimal for a specific task, e.g., SIFT and its variants [19]–[21], and MHI and its variant [22]. Note that the trained deep networks can project the raw feature space into a discriminative feature space that is highly correlated with the class label. Especially, the deep architecture improves the variational lower bound of log $P(v;\theta)$ [23]. For constructing the graph Laplacian, we can utilize deep generative models' ability to extract semantically meaningful features directly, allowing the intrinsic stochastic properties of the deep generative models to fully explore the feature space. Meanwhile, since the discriminative features have already been projected into a meaningful manifold by multispectral embedding, we utilize the second top layer of the hierarchy [as shown in Fig. 2(b) top right], combining several columns of deep nets as a basic building block into the multicolumn deep nets and embedding the spectral relationships. This novel architecture is termed as multispectral neural networks (MSNN).

III. Learning Deep and Wide: Multicolumn Spectral Embedding

In this section, we first describe the proposed approach with the single-column feature using spectral embedding, and then resolve the open problem left in [12] pertaining to the selection of the optimal dimension of the low-dimensional embedding via multiple spectra. To better present the technique, we provide key notations used in the rest of this brief. Capital letters, e.g., X, represent matrices or sets—in our architecture of MSNN, X denotes the discriminative feature sets from the second top layer of the hierarchy of a deep net as in Fig. 2(b) top right; lower case letters, e.g., x, represent vectors, and x_i is the ith element of x. Superscript (i), e.g., $X^{(i)}$, represents data from the ith column deep net.

To effectively and efficiently learn the complementary nature of different views, we adopt the spectral methods to search for a low-dimensional representation and sufficiently smooth embedding over all views simultaneously. Von Luxburg [18] elegantly presents the intuition behind spectral clustering. The fully connected graph is defined as the graph where all nodes within are connected with positive similarity with each other, and all edges are weighted by

the adjacency matrix W, and the similarity function we adopt is the heat kernel: $w_{i,j} = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$, where σ controls the width of the neighborhoods (dispersity). The symmetric normalized graph Laplacian is defined as

$$L_{\text{sys}} := D^{-1/2}LD^{-1/2} = I - D^{-1/2}WD^{-1/2} \tag{1}$$

where D is a degree matrix defined as the diagonal matrix with the degrees $d_i = \sum_{j=1}^{n} w_{i,j}$ on the diagonal and unnormalized graph Laplacian matrix, which is defined as $L = D - W$.

A. Multispectral Embedding

We briefly introduce the core algorithm for multispectral embedding and further cast light on the unaddressed problem pertaining to the selection of the optimal dimension of the low-dimensional embedding from a graph cut point of view.

Given the multiple-view data with n objects having m views, i.e., a set of matrices $X = \{X^{(i)} \in \Re^{m_i \times n}\}_{i=1}^{m}$, where each representation $X^{(i)}$ is a feature matrix from view i, the objective is to find a low-dimensional and sufficiently smooth embedding of X, i.e., $Y \in \Re^{d\times n}$, wherein $d < m_i (1 \le i \le m)$ and d is a predefined number that will be discussed in Section III-B. Fig. 3 shows the working principle. We first build a patch from the penultimate layer of the deep net for a sample on a view. Based on the patches from different views, the part optimization can be performed to get the optimal low-dimensional embedding for each view. Afterward, all low-dimensional embeddings from different patches are unified as a whole by the global coordinate alignment. Finally, the solution is derived by using the expectation maximization (EM) like alternating optimization.

1) Part Optimization: Given the ith view $X^{(i)} = [x_1^{(i)}, \ldots, x_n^{(i)}] \in \Re^{m_i \times n}$, consider an arbitrary point $x_j^{(i)}$ and its k related ones in the same view (e.g., nearest neighbors), $x_{j1}^{(i)}, \ldots, x_{jk}^{(i)}$; the patch of $x_j^{(i)}$ is defined as $X_j^{(i)} = [x_j^{(i)}, x_{j1}^{(i)}, \ldots, x_{jk}^{(i)}] \in \Re^{m_i \times (k+1)}$. For $X_i^{(i)}$, there is a part mapping $f_j^{(i)} : X_j^{(i)} \to Y_j^{(i)}$, wherein $Y_j^{(i)} = [y_j^{(i)}, y_{j1}^{(i)}, \ldots, y_{jk}^{(i)}] \in \Re^{d\times(k+1)}$. To preserve the locality in the projected low-dimensional space, the part optimization is

$$\underbrace{\arg\min}_{Y_j^{(i)}} \sum_{l=1}^{k} \|y_j^{(i)} - y_{jl}^{(i)}\|^2 \left(w_j^{(i)}\right)_l \tag{2}$$

where $w_j^{(i)}$ is a k-dimensional column vector weighted by $(w_j^{(i)})_l = \exp(-\| x_j^{(i)} - x_{jl}^{(i)} \|^2/\sigma^2)$. Throughout our experiments, the heat kernel σ is set to 20% of the total range of the feature distance function, as in [24]. Therefore, (2), which is the part optimization for $X_j^{(i)}$, can be reformulated to

$$\underbrace{\arg\min}_{Y_j^{(i)}} \operatorname{tr}\left(Y_j^{(i)} L_j^i \left(Y_j^{(i)}\right)^T\right) \tag{3}$$

where L_j^i is the graph Laplacian for the view i.

2) Global Coordinate Alignment: Based on the locality information encoded in graph Laplacians L_j^i, (3) finds a sufficiently smooth low-dimensional embedding $Y_i^{(i)}$ by preserving the intrinsic structure of the jth patch on the ith view. Because of the complementary property of multiple views (in our architecture, the penultimate layer output) to each other, a set of nonnegative weights $\alpha = [\alpha_1, \ldots, \alpha_m]$ are imposed on part optimizations of different views independently. The larger the α_i, the more important role the view $X_j^{(i)}$ plays in learning to obtain the low-dimensional embedding $Y_j^{(i)}$. By summing over all views, the multiview part optimization is

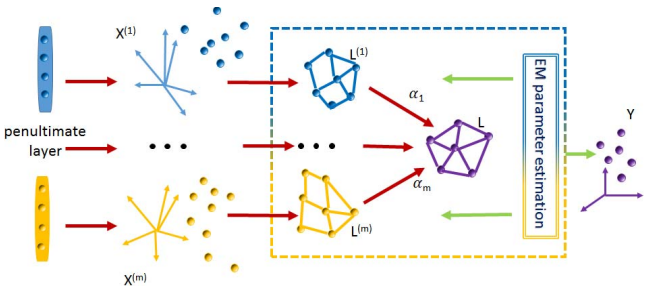

Fig. 3. Working principle for incorporating multispectral embedding in our architecture: the penultimate layer is encoded as a patch for a sample view; multiview embedding finds a low-dimensional space wherein the distribution of each view is sufficiently smooth to explore the complementary property of different views.

$$\underbrace{\arg\min}_{Y=\{Y_j^{(i)}\}_{i=1}^{m},\alpha} \sum_{i=1}^{m} \alpha_i^r \operatorname{tr}\left(Y_j^{(i)} L_j^i \left(Y_j^{(i)}\right)^T\right) \tag{4}$$

$$\text{s.t. } YY^t = I; \quad \sum_{i}^{m} \alpha_i = 1, \quad \alpha_i \ge 0. \tag{5}$$

The constraint $YY^T = 1$ is imposed on (5) to uniquely determine the low-dimensional embedding Y. The normalized graph Laplacian we choose for the system is L_{sys}, as it is a symmetric and semidefinite matrix. The low embedding is solved by Rayleigh quotient formulation. Exponent r is the coefficient for controlling the interdependency between different modalities/views and should satisfy $r \ge 1$. Pronounced independence between different modalities prefers a smaller r, while rich complementary prefers a larger r. In our system, the value r has little influence over the low-dimensional embedding and is set to be fixed as 1.5. The penultimate-layer intermediate representation from an individual deep net is treated as a sample view, hence the number of views m is equal to the number of individual nets.

Equation (5) is a nonlinearly constrained nonconvex optimization problem and an EM like iterative algorithm can be used to obtain a local optimal solution. The alternating optimization iteratively updates Y and α in an alternating fashion. By introducing the Lagrange multiplier λ to take the constraint $\sum_i^m \alpha_i = 1$ into consideration, we get the Lagrange function

$$L(\alpha, \lambda) = \sum_{i=1}^{m} \alpha_i^r \operatorname{tr}(YL^iY^T) - \lambda\left(\sum_{i}^{m} \alpha_i - 1\right). \tag{6}$$

By setting the derivative of $L(\alpha, \lambda)$ with respect to α_i and λ to zero, we have

$$\begin{cases} \dfrac{\partial L(\alpha,\lambda)}{\partial \alpha_i} = r\alpha_i^{r-1}\operatorname{tr}\left(YL_n^{(i)}Y^T\right) - \lambda = 0 \quad i = 1, \ldots, m \\ \dfrac{\partial L(\alpha,\lambda)}{\partial \lambda} = \sum\limits_{i=1}^{M} \alpha_i - 1 = 0. \end{cases} \tag{7}$$

Therefore, α_i can be obtained by

$$\alpha_i = \frac{(1/\operatorname{tr}(YL^iY^T))^{1/(r-1)}}{\left(\sum\limits_{i=1}^{m} \alpha_i \operatorname{tr}(YL^iY^T)\right)^{1/(r-1)}}. \tag{8}$$

B. No More Pesky Parameter Tuning

We cast light on the choice of the lower embedding dimension d and the interpretation of weights α_i dispatched to different views, which was left over in [12] as an open problem. Beforehand, in practice, due to noise and possible degeneracies in the feature spaces, parameter tuning (e.g., cross validation) was adopted by most authors. However, we argue that the low dimension d should be fixed to be the number class-1. This fixed dimension works particularly well in our deep representation learning architecture due to the fact that the deep network can extract intrinsic properties from the raw feature space and entangle the different explanatory factors of variation behind the data. Our reasoning is as follows: according to the graph cut theorem, the multiplicity k^1 of the eigenvalue $\mathbf{0}$ of the graph Laplacian L equals the number of connected components in the graph. Similarly, spectral embedding finds d smallest eigenvalues in the spectrum of L that correspond to the smallest variation of the cluster. The smallest eigenvalue of L is always 0 [18] and the corresponding eigenvector is the constant one vector $\mathbf{1}$. Therefore, the veritable number of d should be class-1. In addition, the cross-validation experiments in [12] are in agreement with our reasoning. Second, we explicitly express the physical meaning of the weights α_i as a measurement of the closeness of intraclass distance from each individual view. From (8), we can see that α_i is proportional to the inverse trace of YL^iY^T, and

$$\mathrm{tr}(YL^iY^T) = \sum \lambda_i \tag{9}$$

where λ_i are the eigenvalues of the graph Laplacian L^i. Hence, $\alpha_i \propto 1/(\sum \lambda_i)$. In spectral clustering [18], a small eigenvalue (closer to 0) represents the closeness of intraclass distance from each individual view. A well-clustered view (in our architecture, a view is an individual deep net), i.e., a view that is easier to be classified, is more significant than other views. Hence, a larger α_i assigns larger significance to the corresponding deep net.

The overall architecture of our system is shown in Fig. 2(b). The merit of combining two techniques is that a good similarity graph is nontrivial, and spectral embedding can be quite unstable under different choices of the parameters for the neighborhood graphs. Hence, spectral embedding cannot serve as a black box algorithm that automatically detects the correct clusters in any given data set. In the meanwhile, the deep neural network architecture can serve as an ideal candidate for this black box algorithm to pretrain for the initial feature space in spite of the fact that an individual deep architecture may suffer from the problem of being stuck in a local optimum.

IV. Experiments

We perform the sanity check for our algorithm as an effective way of combining multiple deep nets against two baselines: the output average model and the weighted majority voting scheme. We assess the performance on three data sets, i.e., MNIST [25] handwriting, Yale face [26] and gender prediction [27].

A. MNIST

The MNIST data set is a widely used benchmark for machine learning algorithms. To verify the algorithm's robustness when there are only very few labeled training instances, we use a subset (1K) of the original data set for training and test on the full 10 000 instances as in [28]. Note that we compare our architecture horizontally versus the CNNs and do not tweak or preprocess the original data set as in [10] to enhance the final classification rate. To have a fair comparison, we set the parameters of our CNN exactly the same as in [28], which has six hidden layers: 1) convolutional layer with 100 5×5 filters per input channel; 2) max-pooling layer of size 2×2; 3) convolutional layer with 150 5×5 filters per map; 4) max-pooling layer of size 2×2; 5) convolutional layer of 200 maps of size 3×3; and 6) a fully connected layer of 300 hidden neurons. The output layer has a softmax activation function with one neuron per class. The learning rate starts with 0.01 and is annealed during training, with 0.99 decaying every epoch with a total of 100 epochs. No deformations are applied to MNIST to increase the virtual number of training samples. The penultimate layer, in this case, the last fully connected layer, is used for spectral embedding.

[1]Multiplicity: the number of eigenvectors belonging to λ_i.

TABLE I

Test Error Rates of the Five Individual Deep Nets Trained on 1K MNIST, Yale Face, With 40% Pixels Corrupted by Random Noise With Standard Deviation of 0.5 and Cross Entropy Loss of Gender Recognition

Methods	Col 1	Col 2	Col 3	Col 4	Col 5
CNN (MNIST 1K) [%]	7.30	7.24	7.08	7.32	7.39
GRBM (Yale Face) [%]	8.89	8.89	11.11	8.89	8.89
DBN (Gender Recog)	0.560	0.501	0.708	0.516	0.476

TABLE II

Test Error Rates With Baseline Comparison: It Can Be Seen That Error Rates of MSNN Are Lower Than Those of Any Individual Deep Net and Are Better Than the Results of Model Averaging, With a Relative Improvement of 4.55% and With 4.95% Improvement Against Weighted Majority Voting on MNIST 1K, 25% on Yale Face, With 40% Pixels Corrupted By Random Noise With Standard Deviation of 0.5 and Significantly Decreased Cross Entropy Loss for Gender Recognition. LR: Logistic Regression, RFLM: Robust Fitting of Linear Models, RF: Random Forest

Methods	MNIST(1K)	Yale Face	Gender Recog
MSNN	**6.36**	**6.67**	**0.456**
Output Average	*7.04*	*8.89*	*0.533*
Weighted Majority Voting	*7.07*	*8.89*	*0.535*
CAE [29]	7.23	-	-
CNN [29]	7.63	-	-
SVM on raw	9.16	13.3	0.614
LR	-	-	0.646
RFLM	-	-	0.646
RF	-	-	0.650

The results of all individual nets are shown in Table I. Note that our individual CNNs achieve slightly better results than those reported in [10]—we conjecture that the discrepancy may result from our evenly chosen 100 instances for each training class. In addition, Table II demonstrates that the error rate of MSNN is lower than that of any individual CNN and the proposed model improves the relative error rate by 5.08%–9.07%.

B. Yale Face Recognition

The Yale face recognition dataset [26] has real-valued data, and we demonstrate that our model can be easily extended to other deep net paradigms, e.g., Gaussian restricted Boltzmann machines (GRBMs) for modeling a real-valued visible layer. The GRBM has been successfully applied to tasks including image classification, video action recognition, and speech recognition [3], [29]–[31]. The GRBM can be viewed as a mixture of diagonal Gaussians with shared parameters, where the number of mixture components is exponential in the number of hidden nodes and the mixing proportions of the

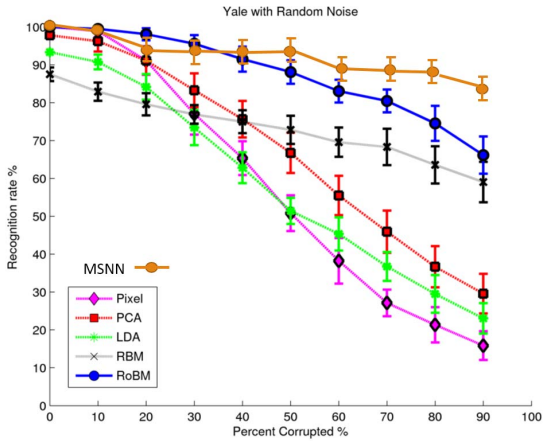

Fig. 4. Recognition rates on the Yale face database as a function of the percentage of pixels corrupted by noise. Random noise with a standard deviation of 0.5 is added to the corrupted pixels.

component are defined by marginalizing out the visible nodes from the joint distribution. The energy term is

$$E(v,h;\theta)=-\sum_{i=1}^{D}\frac{(v_i-b_i)^2}{2\sigma_i^2}-\sum_{i=1}^{D}\sum_{j=1}^{F}W_{ij}h_j\frac{v_i}{\sigma_i}-\sum_{j=1}^{F}a_jh_j. \tag{10}$$

The conditional distributions required for inference and generation are given by

$$P(h_{j=1}|\mathbf{v}) = g\left(\sum_i W_{ij}v_i + a_j\right) \tag{11}$$

$$P(v_{i=1}|\mathbf{h}) = \mathcal{N}\left(v_i|\mu_i,\sigma_i^2\right) \tag{12}$$

where $\mu_i = b_i + \sigma_i^2\sum_j W_{ij}h_j$ and $\mathcal{N}$ is the normal distribution.

The Yale face database contains 15 subjects with 11 images per subject. The face images are frontal but vary in illumination and expression. Following the standard protocol as in [32], we randomly select eight images per subject for training and three for testing and crop images to the resolution of 32×32. A GRBM is trained as the first layer between the visible nodes and the first hidden nodes. Deep belief net is constructed as a basic column with an architecture of $1024 - 1000 - 500 - 2000 - 15$ and 50% dropout [2]. Table I shows five nets on the Yale face database, with 40% pixels corrupted by random noise with a standard deviation of 0.5, and Table II demonstrates that MSNN embedding has achieved 25% error rate improvement against two other baseline mixture-of-experts schemes. We also demonstrate that our model performs on par with the robust Boltzmann machine [32] given heavily corrupted input via Fig. 4 and, because of the dropout technique combining with multiple columns, our model is particularly stable for this salt-and-pepper noise data.

C. Gender Recognition

The prediction of gender from handwriting is a very interesting research area. It has many applications, including the forensic field, where it can help investigators focus more on a certain category of suspects. The data set [27] includes a total of 475 writers producing four handwritten documents: the first page contains an Arabic handwritten text that varies from one writer to another, the second page contains an Arabic handwritten text that is the same for all the writers, the third page contains an English handwritten text that varies from one writer to another, and the fourth page contains an English handwritten text that is the same for all the writers.

The training set consists of the first 282 writers for which the genders are provided, and the remaining 193 writers are used for testing. We further divide the 282 known gender writers into the first 200 writers as the training set and the rest, 82 writers, as the validation set. The results are evaluated using the Logloss evaluation metric define by

$$\text{Logloss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)$$

where N is the number of samples, log is the natural logarithm, $\hat{y}_i$ is the posterior probability that the ith sample elicits a response, and y_i is the ground truth: $y_i = 1$ means that the handwriting elicits a male writer, and $y_i = 0$, a female writer.

We use features provided in [27] for general evaluation purposes: characterizing features are extracted from the handwriting and to have a pen-independent system, images are first binarized using the Otsu thresholding algorithm, and the set of geometrical features include directions, curvatures, and tortuosities. Note that in writer identification, features do not correspond to a single value, but a probability distribution extracted from the handwriting images to characterize writer individuality. Hence, the features are in the range of 0–1 and we can use the processed features as the raw input for our DBN. We preprocess the data by standardizing variables using $(x - \text{mean})/\text{std}$ and remove zero variables from the total number of 7066 to 4564, which is about 35% reduction. A $4564-2000-500-2000-15$ deep belief net is constructed as a basic column. As shown in Table II, the decreased cross-entropy proves the consistent efficacy of MSNN.

V. Conclusion

In this brief, we have introduced a novel architecture for learning a low-dimensional and sufficiently smooth embedding over the popular deep net paradigms. By exploring the complementary property of different columns to obtain an effective low-dimensional embedding, MSNN extracts the most discriminative features from the multicolumn framework. The penultimate layer renders more generic representation learning and opens up more ideas for using intermediate features.

References

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[2] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[3] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 14–22, Jan. 2012.

[4] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep Boltzmann machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2231–2239.

[5] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1. 2012, p. 4.

[6] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Spatio-temporal convolutional sparse auto-encoder for sequence classification," in *Proc. Brit. Mach. Vision Conf.*, 2012, pp. 1–12.

[7] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *IEEE Trans. Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jun./Jul. 2005.

[8] Y. Tang and A.-R. Mohamed, "Multiresolution deep belief networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2012, pp. 1203–1211.

[9] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1237–1242.

[10] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Feb. 2012, pp. 3642–3649.

[11] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *IEEE Trans. Neural Netw.*, vol. 32, no. 2, pp. 333–338, Aug. 2012.

[12] T. Xia, D. Tao, T. Mei, and Y. Zhang, "Multiview spectral embedding," *IEEE Trans. Syst., Man, Cybern.*, vol. 40, no. 6, pp. 1438–1446, Dec. 2010.

[13] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.

[14] G. Hinton, "A practical guide to training restricted Boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[15] I. Sutskever and T. Tieleman, "On the convergence properties of contrastive divergence," in *Proc. Conf. Artif. Intell. Statist.*, pp. 789–795, 2010.

[16] T. Tieleman and G. Hinton, "Using fast weights to improve persistent contrastive divergence," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 1033–1040.

[17] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[18] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 7, no. 4, pp. 395–416, 2007.

[19] R. Arandjelović and A. Zisserman, "Multiple queries for large scale specific object retrieval," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–11.

[20] Y. Luo, D. Tao, C. Xu, H. Liu, and Y. Wen, "Multiview vector-valued manifold regularization for multilabel image classification," *IEEE Trans. Neural Netw. Learn.*, vol. 24, no. 5, pp. 709–722, May 2013.

[21] B. Xie, Y. Mu, D. Tao, and K. Huang, "m-SNE: Multiview stochastic neighbor embedding," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 41, no. 4, pp. 1088–1096, Aug. 2011.

[22] D. Wu and L. Shao, "Silhouette analysis-based action recognition via exploiting human poses," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 236–243, Feb. 2013.

[23] R. Salakhutdinov, "Learning deep generative models," Ph.D. dissertation, Dept. Comput. Sci., Univ. Toronto, ON, Canada, 2009.

[24] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[25] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[26] (2006). *The Yale Face Database* [Online]. Available: http://cvc.yale.edu/projects/yalefaces/

[27] A. Hassaïne, S. Al-Maadeed, and A. Bouridane, "A set of geometrical features for writer identification," in *Proc. Adv. Neural Inf. Process.*, 2012, pp. 584–591.

[28] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. Int. Conf. Artif. Neural Netw. Mach. Learn.*, 2011, pp. 52–59.

[29] A. Krizhevsky, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, ON, Canada, 2009.

[30] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 873–880.

[31] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 140–153.

[32] Y. Tang, R. Salakhutdinov, and G. Hinton, "Robust Boltzmann machines for recognition and denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2264–2271.