Available online at www.sciencedirect.com

ScienceDirect

The Journal of Finance and Data Science 2 (2016) 265–278

http://www.keaipublishing.com/en/journals/jfds/

An overview on data representation learning: From traditional feature learning to recent deep learning

Guoqiang Zhong*, Li-Na Wang, Xiao Ling, Junyu Dong

Department of Computer Science and Technology, Ocean University of China, 238 Songling Road, Qingdao 266100, China

Received 21 November 2016; revised 24 April 2017; accepted 2 May 2017
Available online 8 May 2017

Abstract

Since about 100 years ago, to learn the intrinsic structure of data, many representation learning approaches have been proposed, either linear or nonlinear, either supervised or unsupervised, either "shallow" or "deep". Particularly, deep architectures are widely applied for representation learning in recent years, and have delivered top results in many tasks, such as image classification, object detection and speech recognition. In this paper, we review the development of data representation learning methods. Specifically, we investigate both traditional feature learning algorithms and state-of-the-art deep learning models. The history of data representation learning is introduced, while available online resources (e.g., courses, tutorials and books) and toolboxes are provided. At the end, we give a few remarks on the development of data representation learning and suggest some interesting research directions in this area.

Keywords: Representation learning; Feature learning; Deep learning

Contents

* Corresponding author.
E-mail address: gqzhong@ouc.edu.cn (G. Zhong).
Peer review under responsibility of China Science Publishing & Media Ltd.

http://dx.doi.org/10.1016/j.jfds.2017.05.001

1. Introduction

In many domains, such as artificial intelligence, bioinformatics and finance, data representation learning is a critical step to facilitate the subsequent classification, retrieval and recommendation tasks. Typically, for large scale applications, how to learn the intrinsic structure of data and discover valuable information from data becomes more and more urgent, important and challenging.

Since about 100 years ago, many data representation learning methods have been proposed. Among others, in order to learn low dimensional representations of data with a linear projection, principal component analysis (PCA) was proposed by K. Pearson in 1901,[1] while linear discriminant analysis (LDA) was proposed by R. Fisher in 1936.[2] PCA and LDA are both earliest data representation learning algorithms. Nevertheless, PCA is an unsupervised method, whilst LDA is a supervised one. Based on PCA and LDA, variety of extensions has been proposed, such as kernel PCA[3] and generalized discriminant analysis (GDA).[4]

In 2000, the machine learning community launched the research on manifold learning, which is to discover the intrinsic structure of high dimensional data. Unlike previous global approaches, such as PCA and LDA, manifold learning methods are generally locality based, such as isometric feature mapping (Isomap)[5] and locally linear embedding (LLE).[6] In 2006, G. Hinton and his co-authors successfully applied deep neural networks to dimensionality reduction, and proposed the concept of "deep learning".[7,8] Nowadays, due to their high effectiveness, deep learning algorithms have been employed in many areas beyond artificial intelligence.

On the other hand, the research on artificial neural networks undergoes a tough process, with many successes and difficulties. In 1943, W. McCulloch and W. Pitts created the first artificial neuron, linear threshold unit, which is also called M-P model in the following research,[9] for neural networks. Later, D. Hebb proposed a hypothesis of learning based on the mechanism of neural plasticity, which is also known as Hebbian theory.[10] Essentially, M-P model and Hebbian theory paved the way for neural network research and the development of connectionism in the area of artificial intelligence. In 1958, F. Rosenblatt created the perceptron, a two-layer neural network for binary classification.[11] However, M. Minsky and S. Papert pointed out that perceptrons were even incapable of solving the exclusive-or (XOR) problem.[12]

Until 1974, P. Werbos proposed the back propagation (BP) algorithm to train multi-layer perceptrons (MLP),[13] the neural network research had stagnated. Particularly, in 1986, D. Rumelhart, G. Hinton and R. Williams showed that the back propagation algorithm can generate useful internal representations of data in hidden layers of neural networks.[14] With the back propagation algorithm, although one could train many layers of neural networks in theory, two crucial issues existed: model overfitting and gradient diffusion. In 2006, G. Hinton initiated the breakthrough on representation learning research with the idea of greedy layer-wise pre-training plus finetuning of deep neural networks.[7,8] The issues confusing the neural network community were addressed accordingly. Later on, many deep learning algorithms were proposed and successfully applied to various domains.[15,16]

In Fig. 1, we briefly show the development of data representation learning and neural networks. In general, as the time goes on, the models for representation learning become deeper and deeper, and more and more complex, while the development of neural networks is not so smooth as that of representation learning. However, in the era of deep learning, they gradually combine together for learning effective representations of data.

In this paper, we review the development of data representation learning, including both traditional feature learning and recent deep learning. The rest of this paper is organized as follows. Section 2 is devoted to traditional feature

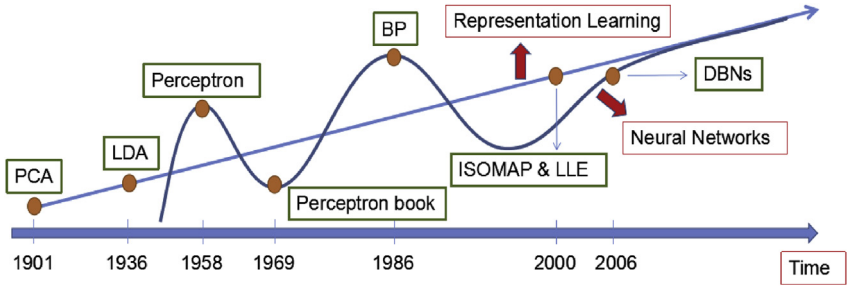

Fig. 1. The development of data representation learning and neural networks.

learning, including linear algorithms and their kernel extension, and manifold learning methods. In Section 3, we introduce the recent progress of deep learning, including important deep learning models, available online sources and public toolboxes. Section 4 concludes this paper.

2. Traditional feature learning

In this section, we focus on traditional feature learning algorithms, which belong to "shallow" models and aim to learn transformations of data that make it easier to extract useful information when building classifiers or other predictors.[17] Some manual feature engineering methods, such as image descriptors (e.g., scale-invariant feature transform or SIFT,[18] local binary patterns or LBP,[19] and histogram of oriented gradients or HOG,[20] and so on) and document statistics (e.g., term frequency-inverse document frequency or TF-IDF,[21] and so on), are not covered in this review.

From the perspective of its formulation, an algorithm is generally considered to be linear or nonlinear, supervised or unsupervised, generative or discriminative, global or local. For example, PCA is a linear, unsupervised, generative and global feature learning method, while LDA is a linear, supervised, discriminative and global method. In this section, we adopt the taxonomy to categorize the feature learning algorithms as global ones or local ones. In general, global methods try to preserve the global information of data in the learned feature space, but local ones focus on preserving local similarity between data during learning the new representations. Moreover, we usually call locality-based feature learning as manifold learning, since it is to discover the manifold structure hidden in the high dimensional data.

For instance, unlike PCA and LDA, LLE is a manifold learning algorithm. It is aimed to preserve the reconstruction property within the neighborhood of each data point. Fig. 2 illustrates an example to show how LLE learns the 2D manifold structure of data from 3D observations. LLE can discover the nonlinear relationship between low-dimensional embeddings and high dimensional data. However, it is based on the assumption that the data are densely sampled.

In the literature, van der Maaten, Postma and van den Herik provided a MATLAB toolbox for dimensionality reduction, which includes the codes of 34 feature learning algorithms.[22] Yan et al[23] introduced a general framework known as graph embedding to unify a large family of dimensionality reduction algorithms into one formulation. Zhong, Chherawala and Cheriet[24] compared three kinds of supervised dimensionality reduction methods for handwriting recognition, while Zhong and Cheriet,[25] presented a framework from the viewpoint of tensor representation learning, which considers the input data as tensors and unifies many linear, kernel and tensor dimensionality reduction methods with one learning criterion.

For concreteness, Zhong, Chherawala and Cheriet[24] compared traditional linear dimensionality reduction approach (LDA[2]), locality-based manifold learning approach (marginal Fisher analysis (MFA)[23]) and relational learning approach (probabilistic relational principal component analysis (PRPCA)[26]) on handwriting recognition applications. The comparison results and statistical tests show that locality-based manifold learning approach (MFA) generally performs well in terms of recognition accuracy, but with high computational complexity; traditional linear dimensionality reduction approach (LDA) is efficient, but not necessarily to deliver the best result; relational learning approach (PRPCA) is promising, and more efforts should be dedicated to this area.

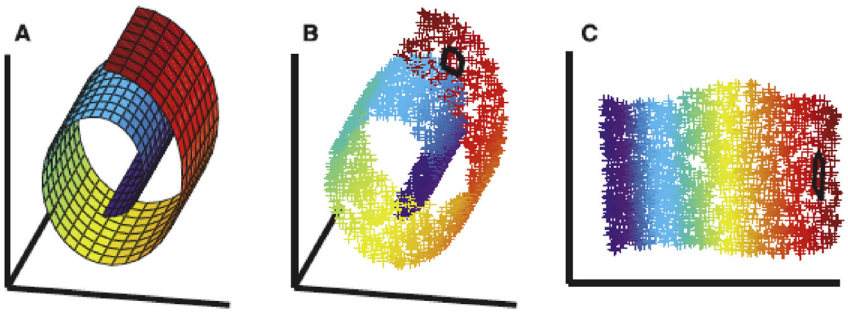

Fig. 2. Illustration of the LLE algorithm.[6] (A) A 2D manifold embedded in the 3D space (B) Samples from the 2D manifold (C) Learning results of LLE. Here, colors encode the closeness between points. The black outlines show the neighborhood of a data point.

In nature, many data can be represented in the form of tensors. For example, vectors are first-order tensors and matrices are second-order tensors. Hence, many representation learning algorithms can be unified in one formulation with respect to tensors. Zhong and Cheriet[25] proposed a convergent tensor representation learning framework. Many previous linear, kernel and tensor representation learning methods can be considered as special cases of this framework. And more, the authors proved the convergence of the learning algorithm for this general framework. Specifically, for vectorized inputs, the learning algorithm can converge to the globally optimal solution; while for high-order tensors, it can converge to a local optimum.

2.1. Global feature learning

As mentioned above, PCA is one of the earliest linear feature learning algorithm.[1,27] Due to its simplicity, PCA has been widely used for dimensionality reduction.[28] It uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables. To some extent, classical multidimensional scaling (MDS) is similar with PCA, i.e. both of them are linear method and optimized using eigenvalue decomposition.[29]

The difference between PCA and MDS is that, the input of PCA is the data matrix, while that of MDS is the distance matrix between data. If the training data are available, one can directly use PCA for feature learning. However, if only the distance (or dissimilarity) matrix between data is given, one can choose MDS to learn the low dimensional representations of data. However, if PCA is applied on the training data and MDS on the distance matrix induced from the same data set, PCA and MDS will obtain equivalent results.

Except for eigenvalue decomposition, singular value decomposition (SVD) is often used for optimization as well. Latent semantic analysis (LSA) in information retrieval is optimized using SVD, which reduces the number of rows while preserving the similarity structure among columns (rows represent words and columns represent documents).[30]

As variants of PCA, kernel PCA (KPCA) extends PCA for nonlinear dimensionality reduction using the kernel trick,[31] while probabilistic PCA is a probabilistic version of PCA.[32] Moreover, based on PPCA, Lawrence proposed the Gaussian process latent variable model (GPLVM), which is a fully probabilistic, nonlinear latent variable model and can learn a nonlinear mapping from the latent space to the observation space.[33] Here, KPCA generally maps the original data into a high dimensional feature space using a nonlinear function, and then, conducts linear dimensionality reduction in this high dimensional feature space. PPCA re-formulates PCA from the probabilistic perspective and can be used in the scenario that includes missing data. In addition, to address the problem that PCA and PPCA can only learn a linear subspace of the observed data, GPLVM learns the low dimensional embeddings in a nonlinear manner.

In order to integrate supervisory information into the GPLVM framework, Urtasun and Darrell proposed the discriminative GPLVM.[34] However, since DGPLVM is based on the learning criterion of LDA[2] or GDA,[4] the dimensionality of the learned latent space in DGPLVM is restricted to be at most $C - 1$, where C is the number of classes.

To address this problem, Zhong et al proposed the Gaussian process latent random field (GPLRF),[35] by enforcing the latent variables to be a Gaussian Markov random field (GMRF)[36] with respect to a graph constructed from the supervisory information. To the end, GPLRF is a supervised, nonlinear, Gaussian process latent variable model. Due to the constructed Gaussian Markov random field (GMRF), GPLRF encourages data belonging to the same class to be close, and meanwhile, due to the adopted Gaussian process latent variable model, GPLRF enforces data of different classes to be far apart. Hence, GPLRF can generally learn effective representations of data and result in high classification accuracy.

Among others, some more extensions of PCA include sparse PCA,[37] robust PCA[38,39] and probabilistic relational PCA.[26] The readers can refer to the corresponding references and get to know more details of them.

LDA is a supervised, linear feature learning method, which enforces data belonging to the same class to be close and that belonging to different classes to be far away in the learned low-dimensional subspace.[2] LDA has been successfully used in face recognition, and the obtained new features are called Fisherfaces.[40] Similar with Eigenfaces,[28] which is obtained via PCA, Fisherfaces is learned from the gray-level images as well and a nearest neighbor classifier can be applied for the subsequent recognition. However, compared to Eigenfaces, Fisherfaces are well separate in the low-dimensional subspace, even under severe variation in lighting and facial expressions.

GDA is the kernel version of LDA.[4] In general, LDA and GDA are learned with the generalized eigenvalue decomposition. However, Wang et al pointed out that the solution of generalized eigenvalue decomposition was only an approximation to that of the original trace ratio problem of LDA and GDA.[41] Hence, they transformed the trace ratio problem to a series of trace difference problems and used an iterative algorithm to solve it. For LDA, its approximate solution is usually obtained from the following problem:

$$\mathbf{W}^* = \underset{\mathbf{W}}{\operatorname{argmax}} \frac{|\mathbf{W}^T\mathbf{S}_b\mathbf{W}|}{|\mathbf{W}^T\mathbf{S}_w\mathbf{W}|}, \tag{1}$$

where

$$\mathbf{S}_b = \sum_{i=1}^{C} n_i(\mu_i - \mu)(\mu_i - \mu)^T, \tag{2}$$

and

$$\mathbf{S}_w = \sum_{i=1}^{C} \sum_{\mathbf{x}_j \in c_i} (\mu_i - \mathbf{x}_j)(\mu_i - \mathbf{x}_j)^T, \tag{3}$$

are the between-class scattering matrix and within-class scattering matrix, respectively, while μ_i is the sample mean of class c_i, μ is the overall sample mean, n_i is the number of samples in class c_i and C is the number of classes. To solve Problem (1), the generalized eigenvalue decomposition method is usually used:

$$\mathbf{S}_b\mathbf{w}_k = \lambda_k\mathbf{S}_w\mathbf{w}_k, \tag{4}$$

where λ_k is the k-th largest generalized eigenvalue with the corresponding eigenvector $\mathbf{w}_k$, and $\mathbf{w}_k$ constitutes the k-th column vector of the matrix W*. As the kernel version of LDA, GDA can be similarly formulated and solved using the generalized eigenvalue decomposition method. Nevertheless, the solution of Problem (4) is not necessary to be the optimal one for LDA. To address this problem, Wang et al provided a convergent algorithm for the following trace ratio problem[41]:

$$\mathbf{W} = \underset{\mathbf{W}^T\mathbf{W}=\mathbf{I}_d}{\operatorname{argmax}} \frac{Tr(\mathbf{W}^T\mathbf{S}_p\mathbf{W})}{Tr(\mathbf{W}^T\mathbf{S}_t\mathbf{W})}, \tag{5}$$

where $\mathbf{S}_p$ and $\mathbf{S}_t$ are two symmetric and positive semi-definite matrices, $Tr(\cdot)$ is the trace of a matrix, $Tr(\mathbf{X}) = \sum \mathbf{X}_{kk}$, and d is the dimensionality of the target space. As a result, they transformed Problem (5) into a sequence of trace difference problems

$$\mathbf{W}^i = \underset{\mathbf{W}^T\mathbf{W}=\mathbf{I}_d}{\operatorname{argmax}} Tr[\mathbf{W}^T(\mathbf{S}_p - \eta^i\mathbf{S}_t)\mathbf{W}], \tag{6}$$

where $\eta^i = \frac{Tr(\mathbf{W}^{i-1^T}\mathbf{S}_p\mathbf{W}^{i-1})}{Tr(\mathbf{W}^{i-1^T}\mathbf{S}_t\mathbf{W}^{i-1})}$, and used an iterative algorithm to solve it. Furthermore, Jia et al put forward a novel Newton−Raphson method for trace ratio problems, which can be proved to be convergent as well.[42]

Zhong, Shi and Cheriet[43] have presented a novel method called relational Fisher analysis (RFA), which is based on the trace ratio formulation and sufficiently exploits the relational information of data. Suppose the given relational matrix is $\mathbf{R} \in \Re^{N\times N}$. The learning problem of RFA can be formulated as

$$\mathscr{L} = \min_{\mathbf{W}} \frac{\mathrm{tr}(\mathbf{W}^T\mathbf{X}\mathbf{L}_\mathbf{I}\mathbf{X}^\mathbf{T}\mathbf{W})}{\mathrm{tr}(\mathbf{W}^\mathbf{T}\mathbf{X}\mathbf{L}_\mathbf{P}\mathbf{X}^\mathbf{T}\mathbf{W})} + \alpha\mathrm{tr}(\mathbf{W}^T\mathbf{X}\mathbf{R}\mathbf{X}^T\mathbf{W}), \tag{7}$$

where $\mathbf{L}_\mathbf{I}$ defines the intrinsic graph, $\mathbf{L}_\mathbf{P}$ defines the penalty graph, and α is the tradeoff factor.

Zhong and Ling[44] analyzed an iterative algorithm similar to that introduced in Wang et al,[41] for the trace ratio problems and proved the necessary and sufficient conditions for the existence of the optimal solution of trace ratio problems. The sufficient and necessary conditions for the existence of the optimal solution of trace ratio problems are

that there is a sequence $\{\lambda_1^*, \lambda_2^*, \ldots, \lambda_n^*\}$ which converges to λ^* as $n \to +\infty$, where λ^* is the optimal value of the trace ratio in Problem (5).

More extensions of LDA may include incremental LDA,[45] DGPLVM[34] and marginal Fisher analysis (MFA).[23]

Except feature learning algorithms mentioned above, there are many other feature learning methods, such as independent component analysis (ICA),[46] canonical-correlation analysis (CCA),[47] ensemble learning based feature extraction,[48] multi-task feature learning,[49] and so on. Moreover, to directly process tensor data, many tensor representation learning algorithms have been proposed.[23,50–58]

Firstly, Yang et al proposed the 2DPCA algorithm and shew its advantage over PCA on face recognition problems.[50] Unlike PCA, the inputs of 2DPCA are 2D images rather than 1D vectors, so that the images (e.g., faces) do not need to be transformed into a vector prior to feature extraction any more. Secondly, Ye, Janardan and Li proposed the 2DLDA algorithm, which extends LDA to applications with two-order tensors.[51] while classic LDA uses the vectorized representations of data, 2DLDA works on 2D images directly. And more, 2DLDA can deal with the singularity problem with low cost in both time and space. Recently, a large margin low rank tensor representation learning algorithm is introduced, the convergence of which can be theoretically guaranteed.[55]

2.2. Manifold learning

In this sub section, we focus on locality-based feature learning methods, and call them manifold learning methods. Although most of the manifold learning algorithms are nonlinear dimensionality reduction approaches, some are linear dimensionality reduction methods, such as locality preserving projections (LPP)[59] and MFA.[23] Meanwhile, note that some nonlinear dimensionality reduction algorithms are not manifold learning approaches, as they are not aimed to discover the intrinsic structure of high dimensional data, such as Sammon mapping[60] and KPCA.[31]

In 2000, "*Science*" published two interesting papers on manifold learning. The first paper introduces Isomap, which combines the Floyd-Warshall algorithm with classic MDS.[5] Based on local neighborhood of the samples, Isomap computes the pair-wise geodesic distance between data using the Floyd-Warshall algorithm, which is used to find the shortest distance between each pair of points[61]), and then, learn the low-dimensional embeddings of data using classic MDS on the computed pair-wise geodesic distances.

The second paper is about LLE, which encodes the locality information at each point into the reconstruction weights of its neighbors.[6] Following the idea of LLE, Belkin and Niyoji proposed the Laplacian eigenmaps (LE) algorithm based on the correspondence between the graph Laplacian and the Laplace Beltrami operator.[62] LPP can be considered as a linear version of LE,[59] which has been applied to face recognition and the low dimensional representations of faces are called Laplacianfaces.[63] Fig. 3 shows the two dimensional embeddings of Laplacianfaces.

Moreover, Zhang and Zha introduced the local tangent space alignment (LTSA) algorithm by representing the local geometry of the manifold using tangent space.[64] Later on, many manifold learning algorithms were proposed.[23,59,65–68] In particular, the work of Zhong et al[69] combines the idea of LTSA[64] and LE,[62] which computes the local similarity between data using the Euclidean distance in the local tangent space and employs LE to learn the low dimensional embeddings of data.

Cheriet et al[70] applied manifold learning approaches to shape-based recognition of historical Arabic documents and obtained noticeable improvement over previous methods.

In addition to the methods mentioned above, some related work that needs pay attention to includes the algorithms for distance metric learning,[71–74] semi-supervised learning,[75] dictionary learning,[76] and non-negative matrix factorization (NMF),[77] which to some extent take account of the underlying structure of data. Particularly, from the perspective of matrix factorization $\mathbf{X} \approx \mathbf{WH}$, PCA constraints the columns of $\mathbf{W}$ to be orthonormal and the rows of $\mathbf{H}$ to be orthogonal to each other, while NMF does not allow negative entries in the matrix factors $\mathbf{W}$ and $\mathbf{H}$. This constraint leads to a parts-based representation because they allow only additive, not subtractive, combinations.

3. Deep learning

In the literature, 4 survey papers on deep learning have been published. Bengio[78] introduced the motivations, principles and some important algorithms of deep learning, while in Bengio,[17] from the perspective of representation learning, Bengio, Courville and Vincent reviewed the progress of feature learning and deep learning. LeCun, Bengio

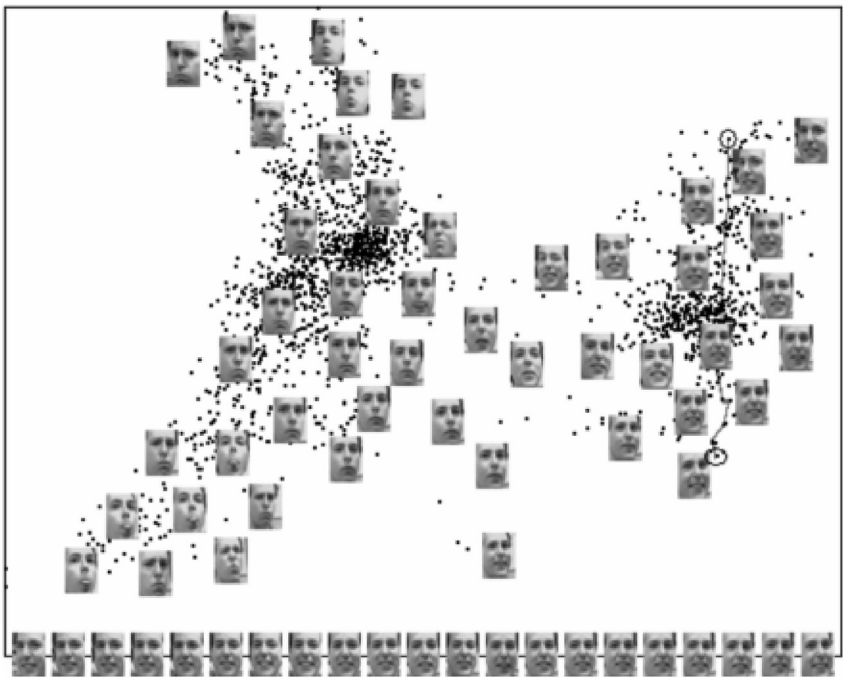

Fig. 3. Two dimensional embeddings of Laplacianfaces.[63]

and Hinton[79] introduced the development of deep learning and some important deep learning models including convolutional neural network[80] and recurrent neural network.[81] Schmidhuber[82] reviewed the development of the artificial neural networks and deep learning year by year. With these survey papers, the readers who are interested to deep learning may easily understand the research area of deep learning and its history.

To learn deep learning algorithms, some internet resources are worth being recommended.

(1) The first one is the Coursera course taught by Professor Hinton.[a] This course is about artificial neural networks and how they're being used for machine learning.
(2) The second one is the tutorial on unsupervised feature learning and deep learning, provided by some researchers at Stanford University.[b] Except basic knowledge on unsupervised feature learning and deep learning algorithms, this tutorial includes many exercises. Hence, it's quite suitable for deep learning beginners.
(3) The third one is the deep learning website.[c] This website provides not only deep learning tutorials, but also reading list, softwares, data sets and so on.
(4) The fourth one is a blog, which is written in Chinese.[d] The host of this blog records the process how she/he learned deep learning and wrote the codes model by model. Nevertheless, many other blogs and webpages are also useful and helpful, such as http://blog.csdn.net/and Wikipedia.
(5) The last but not the least is the deep learning book written by Professor Goodfellow, Bengio and Courville, which has been published by MIT Press. Its online version is free.[e]

With these courses, tutorials, blogs and books, the students and engineers who may study or work on deep learning can basically understand the theoretical details of the deep learning algorithms.

[a] https://www.coursera.org/learn/neural-networks#.
[b] http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial.
[c] http://deeplearning.net/.
[d] http://www.cnblogs.com/tornadomeet/.
[e] http://www.deeplearningbook.org/.

3.1. Deep learning models

Here, we review some deep learning models, especially that proposed after the publication of.[17]

As mentioned above, there exist two crucial issues for training multi-layer neural networks until 2006: model overfitting and gradient diffusion. Due to less available data and huge volume of parameters, the training of multi-layer neural networks is very easy to stick in a poor local optimum. Furthermore, with a deep architecture, the computed gradient close to the input layer will be generally very tiny when using the back propagation algorithm to train the deep models. To the end, this gradient diffusion phenomenon prevents the deep architecture from being trained optimally. To address these two issues, deep learning has been proposed and successfully applied in many areas. Fig. 4 shows a three-layer deep architecture.

The renewal of deep learning is mainly due to the great progress of three aspects: feature learning, availability of large scale of labeled data, and hardware, especially general purpose graphics processing units (GPGPU). In 2006, Hinton and Salakhutdinov proposed to use greedy layer-wise pre-training and finetuning for the learning of deep neural networks, which results in higher performance than state-of-the-art algorithms on MNIST handwritten digits recognition and document retrieval tasks.[8]

Since this groundbreaking work by Hinton and Salakhutdinov, many excellent ideas on deep learning have been proposed. Deng and Yu[84] defined the concept of deep learning as a class of machine learning algorithms that:

- use a cascade of many layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input. The algorithms may be supervised or unsupervised and applications include pattern analysis (unsupervised) and classification (supervised).
- are based on the (unsupervised) learning of multiple levels of features or representations of the data. Higher level features are derived from lower level features to form a hierarchical representation.
- are part of the broader machine learning field of learning representations of data.
- learn multiple levels of representations that correspond to different levels of abstraction; the levels form a hierarchy of concepts.

Bengio et al[15] introduced the stacked autoencoders and confirmed the hypothesis that the greedy layer-wise unsupervised training strategy mostly helps the optimization, by initializing weights in a region near a good local minimum, giving rise to internal distributed representations that are high-level abstractions of the inputs, and bringing better generalization.

Vincent et al[85] proposed the stacked denoising autoencoders, which are trained locally to denoise corrupted versions of the inputs. Zheng et al[86] showed the effectiveness of deep architectures that are built with stacked feature learning modules, such as PCA and stochastic neighbor embedding (SNE).[87] While PCA is a linear dimensionality reduction method, SNE is a nonlinear one. It tries to place the data in a low-dimensional space so as to optimally preserve original neighborhood identity. With the stacked feature learning modules, deep architectures can abstract effective representations of data layer by layer.

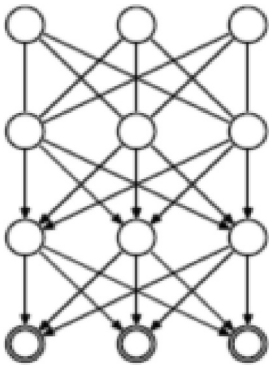

Fig. 4. A three-layer deep architecture.[83]

To improve the effectiveness of the deep architectures built with stacked feature learning models, Zheng et al applied the stretching technique[88] on the weight matrix between the top two layers except the classifier, and demonstrated the effectiveness of the proposed method on handwritten text recognition tasks.[89] Let $\mathbf{A} \in \Re^{D\times d}$ be a matrix whose columns denote the weight vectors learnt by the last feature learning model in the stacked deep architecture. Let $\mathbf{W} \in \Re^{d\times L}, L > d$ be a random matrix whose entries are sampled from the standard normal distribution $\mathcal{N}(0,1)$. The stretched matrix $\mathbf{A}_s \in \Re^{D\times L}, L > M$ is defined as

$$\mathbf{A}_s = \frac{1}{\sqrt{L}}(\mathbf{A} \times \mathbf{W}). \tag{8}$$

With $\mathbf{A}_s$, the stretching operation maps the learned features to a high dimensional space, which generally results in high recognition accuracy. Additionally, in Roy et al,[90] a tandem hidden Markov model (HMM) using deep belief networks (DBNs)[7] was proposed and applied for offline handwriting recognition. For concreteness, in this work, DBNs are adopted to learn the compact representations of sequential data, while HMM is applied for (sub-)word recognition.

In 2012, Krizhevsky, Sutskever and Hinton created the "AlexNet" and won the ImageNet Large Scale Visual Recognition Competition (ImageNet LSVRC).[91] In AlexNet, the dropout regularization[92] and the nonlinear activation function called rectified linear units (ReLUs)[93] were used. To speed up the learning on 1.2 million training images from 1000 categories, AlexNet was implemented on GPUs. Between 2013 and 2016, all the models performed well in the ImageNet LSVRC are based on deep convolutional neural networks (CNNs), such as OverFeat,[94] VGGNet,[95] GoogLeNet[96] and ResNet.[97]

In Donahue et al,[98] an interesting feature extraction method based on AlexNet was proposed. The authors showed that features extracted from the activation of a deep convolutional network (e.g., AlexNet) trained in a fully supervised fashion on a large, fixed set of object recognition tasks can be repurposed to novel generic tasks. Accordingly, this feature was called deep convolutional activation feature (DeCAF).

Zhong et al[99] introduced two challenging problems related to photographed document images and applied DeCAF to set the baseline results for the proposed problems. Here, the first problem is that, for some photographed document images, which book do they belong to? The second one is, for some photographed document images, what is the type of the book they belong to? To address these two problems, Zhong et al input the photographed document images to "AlexNet" and extracted the activation outputs of the sixth fully connected layer as new representations of the images. And then, a support vector machine (SVM) is applied for the classification tasks.

Cai et al[100] considered the problem that whether DeCAF is good enough for accurate image classification, and based on the reducing and stretching operations, the authors improved DeCAF on several image classification problems. Based on the AlexNet[91] and VGGNet,[95] Zhong et al proposed a deep hashing learning algorithm, which greatly improved previous hashing learning approaches on image retrieval.[101] Fig. 5 shows an end-to-end deep hashing learning network, which includes a hash layer before the classifier layer. More importantly, this network can learn the hashing code, hashing function and compact representations of data simultaneously.

Recently, deep learning models are gained much attention from recurrent neural networks (RNNs),[81,102] long short-term memory (LSTM),[103,104] attention based models[105,106] and generative adversarial nets.[107] The applications are generally focused on image classification, object detection, speech recognition, handwriting recognition, image caption generation and machine translation.[108–110]

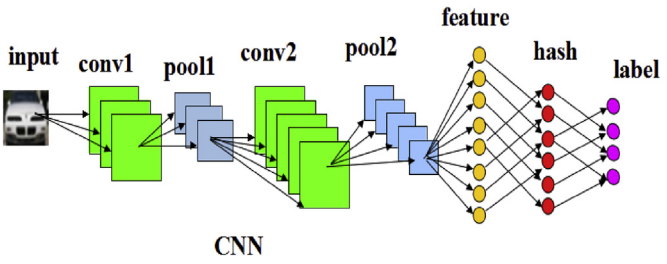

Fig. 5. The end-to-end deep hashing learning network.[101]

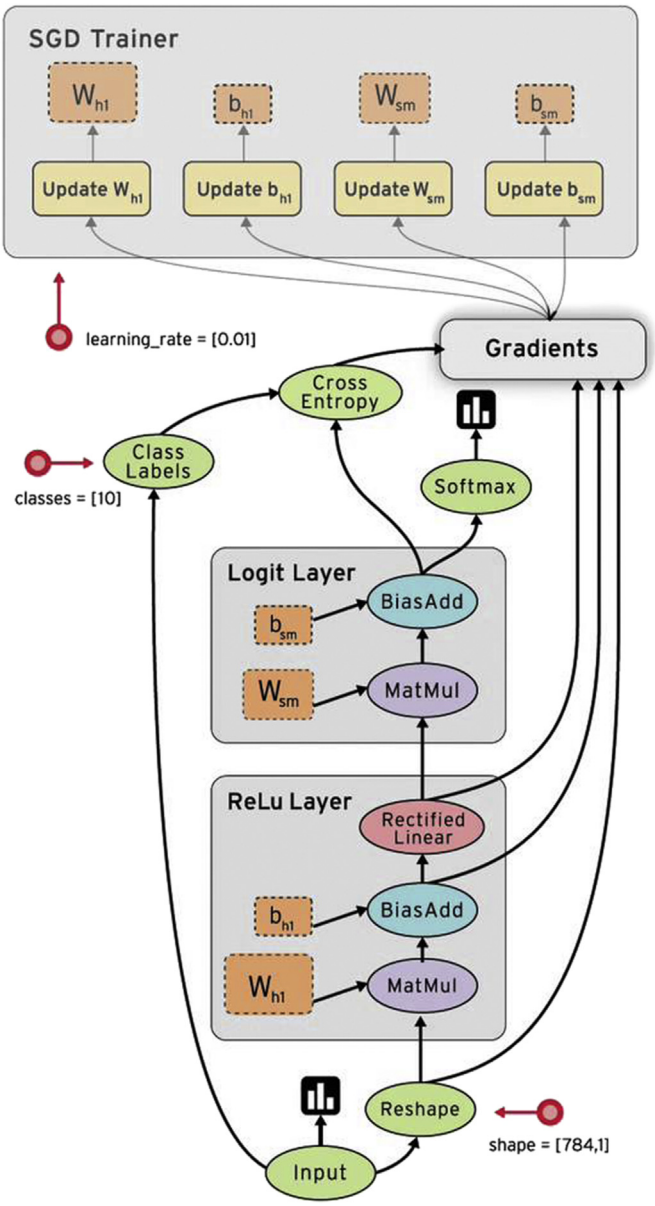

Fig. 6. An example of the data flow graph in TensorFlow.

3.2. Deep learning toolboxes

There are many deep learning toolboxes commonly shared on the internet. In each toolbox, the codes of some deep learning models, such as DBNs,[7] LeNet-5,[80] AlexNet[91] and VGGNet,[95] are often provided, respectively. The researchers may directly use the codes or develop new models based on the codes under certain licenses. In the following, we briefly introduce Theano,[f] Caffe,[g] TensorFlow[h] and MXNet.[i]

[f] http://www.deeplearning.net/software/theano/index.html.
[g] http://caffe.berkeleyvision.org/.
[h] https://www.tensorflow.org/.
[i] http://mxnet.io/index.html.

Theano is a Python library. It is tightly integrated with NumPy, and allows the users to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. Moreover, it could perform data-intensive calculations on GPUs with up to 140 times faster than with CPU. The deep learning tutorial provided at http://deeplearning.net/tutorial/ is just based on Theano.

Caffe is a pure C++/CUDA toolbox for deep learning. However, it provides command line, Python and MATLAB interfaces. The Caffe codes run fast, and can seamless switch between CPU and GPU.

TensorFlow is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. Fig. 6 shows an example of the data flow graphs[j]. TensorFlow has the automatic differentiation capability to facilitate the computation of derivatives.

MXNet is developed by many collaborators from several universities and companies. It supports both imperative and symbolic programming, and multiple programming languages, such as C++, Python, R, Scala, Julia, Matlab and Javascript. In general, the running speed of MXNet codes is comparative with that of Caffe codes, and much faster than that of Theano and TensorFlow. Currently, MXNet is supported by major Public Cloud providers including Azure and AWS.[k] Amazon has chosen MXNet as its deep learning framework of choice at AWS.[l]

4. Conclusion

In this paper, we review the research on data representation learning, including traditional feature learning and recent deep learning. From the development of feature learning methods and artificial neural networks, we can see that deep learning is not totally new. It's the consequence of the great progress of feature learning research, availability of large scale of labeled data, and hardware. However, the breakthrough of deep learning not only affects the artificial intelligence area, but also greatly improves the progress of many domains, such as finance[111] and bioinformatics.[112]

For the future research on deep learning, we suggest three directions: the fundamental theory, novel algorithms, and applications. Some researchers have tried to analyze deep neural networks.[113−115] However, the gap between the theory and application of deep learning is still quite large. Although many deep learning algorithms have been proposed, most of them are based on deep CNNs or RNNs. Therefore, creative deep learning algorithms need to be proposed, to solve real world problems, such as unsupervised models and transfer learning models. Moreover, deep learning algorithms have been preliminarily exploited in many domains. However, to solve some challenging problems in natural language processing and computer visions among others, more sophisticated models and algorithms are desired.

Finally, we emphasize that deep learning is not the everything of machine learning and the only way to realize artificial intelligence. To solve real world problems, many models and algorithms for intelligent data analytics are indispensable.

References

1. Pearson K. On lines and planes of closest fit to systems of points in space. *Philos Mag*. 1901;2:559−572.
2. Fisher R. The use of multiple measurements in taxonomic problems. *Ann Eugen*. 1936;7:179−188.
3. Schölkopf B, Smola A, Müller K-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput*. 1998;10:1299−1319.
4. Baudat G, Anouar F. Generalized discriminant analysis using a kernel approach. *Neural Comput*. 2000;12:2385−2404.
5. Tenenbaum J, Silva V, Langford J. A global geometric framework for nonlinear dimensionality reduction. *Science*. 2000;290:2319−2323.
6. Roweis S, Saul L. Nonlinear dimensionality reduction by locally linear embedding. *Science*. 2000;290:2323−2326.
7. Hinton G, Osindero S, Teh YW. A fast learning algorithm for deep belief nets. *Neural Comput*. 2006;18:1527−1554.
8. Hinton G, Salakhutdinov R. Reducing the dimensionality of data with neural networks. *Science*. 2006;313:504−507.
9. McCulloch W, Pitts W. A logical calculus of ideas immanent in nervous activity. *Bull Math Biophys*. 1943;5:115−133.
10. Hebb D. *The Organization of Behavior*. New York: Wiley; 1949.
11. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev*. 1958;65:386−408.
12. Minsky M, Papert S. *Perceptrons: An Introduction to Computational Geometry*. MIT Press; 1969.
13. Werbos P. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Ph.D. thesis. Harvard University; 1974.

[j] http://www.tensorfly.cn/.
[k] https://aws.amazon.com/mxnet/.
[l] http://fortune.com/2016/11/22/amazon-deep-learning-mxnet/.

14. Rumelhart D, Hinton G, Williams R. Learning representations by back-propagating errors. *Nature*. 1986;23:533−536.
15. Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. In: *NIPS*. 2006:153−160.
16. Ranzato M, Poultney C, Chopra S, LeCun Y. Efficient learning of sparse representations with an energy-based model. In: *NIPS*. 2006:1137−1144.
17. Bengio Y, Courville A, Vincent P. Representation learning: a review and new perspectives. *IEEE Trans Pattern Anal Mach Intell*. 2013;35:1798−1828.
18. Lowe D. Object recognition from local scale-invariant features. In: *ICCV*. 1999:1150−1157.
19. Ojala T, Pietikäinen M, Harwood D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recogn*. 1996;29:51−59.
20. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: *CVPR*. 2005:886−893.
21. Rajaraman A, Ullman J. *Mining of Massive Datasets*. Cambridge University Press; 2011.
22. van der Maaten L, Postma E, van den Herik H. *Dimensionality Reduction: A Comparative Review*. Tilburg University; 2009. Technical Report TiCC-TR 2009-005.
23. Yan S, Xu D, Zhang B, Zhang H-J, Yan Q, Lin S. Graph embedding and extensions: a general framework for dimensionality reduction. *IEEE Trans Pattern Anal Mach Intell*. 2007;29:40−51.
24. Zhong G, Chherawala Y, Cheriet M. An empirical evaluation of supervised dimensionality reduction for recognition. In: *ICDAR*. 2013:1315−1319.
25. Zhong G, Cheriet M. Tensor representation learning based image patch analysis for text identification and recognition. *Pattern Recogn*. 2015;48:1211−1224.
26. Li W-J, Yeung D-Y, Zhang Z. Probabilistic relational PCA. In: *NIPS*. 2009:1123−1131.
27. Joliffe I. *Principal Component Analysis*. Springer; 2002.
28. Sirovich L, Kirby M. Low-dimensional procedure for the characterization of human faces. *J Opt Soc Am A*. 1987;4:519−524.
29. Cox T, Cox M. *Multidimensional Scaling*. Boca Raton: Chapman and Hall; 2001.
30. Dumais S. Latent semantic analysis. *ARIST*. 2004;38:188−230.
31. Schölkopf B, Smola A, Muller K-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput*. 1998;10:1299−1319.
32. Tipping M, Bishop C. Probabilistic principal component analysis. *J R Stat Soc B*. 1999;6:611−622.
33. Lawrence N. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *J Mach Learn Res*. 2005;6:1783−1816.
34. Urtasun R, Darrell T. Discriminative Gaussian process latent variable models for classification. In: *Proceedings of the International Conference on Machine Learning*. 2007:927−934.
35. Zhong G, Li W-J, Yeung D-Y, Hou X, Liu C-L. Gaussian process latent random field. In: *AAAI*. 2010.
36. Rue H, Held L. *Gaussian Markov Random Fields: Theory and Applications, Volume 104 of Monographs on Statistics and Applied Probability*. London: Chapman & Hall; 2005.
37. Zou H, Hastie T, Tibshirani R. Sparse principal component analysis. *J Comput Graph Stat*. 2006;15:265−286.
38. Candès E, Li X, Ma Y, Wright J. Robust principal component analysis? *J ACM*. 2011;58:11.
39. Bouwmans T, Zahzah E-H. Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance. *Comput Vis Image Underst*. 2014;122:22−34.
40. Belhumeur P, Hespanha J, Kriegman D. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans Pattern Anal Mach Intell*. 1997;19:711−720.
41. Wang H, Yan S, Xu D, Tang X, Huang T. Trace ratio vs. Ratio trace for dimensionality reduction. In: *CVPR*. 2007.
42. Jia Y, Nie F, Zhang C. Trace ratio problem revisited. *IEEE Trans Neural Netw*. 2009;20:729−735.
43. Zhong G, Shi Y, Cheriet M. Relational fisher analysis: a general framework for dimensionality reduction. In: *IJCNN*. 2016:2244−2251.
44. Zhong G, Ling X. The necessary and sufficient conditions for the existence of the optimal solution of trace ratio problems. In: *CCPR*. 2016:742−751.
45. Ghassabeh Y, Rudzicz F, Moghaddam H. Fast incremental LDA feature extraction. *Pattern Recogn*. 2015;48:1999−2012.
46. Hyvarinen A, Karhunen J, Oja E. *Independent Component Analysis*. 1st ed. New York: J. Wiley; 2001.
47. Hotelling H. Relations between two sets of variates. *Biometrika*. 1936;28:321−377.
48. Zhong G, Liu C-L. Error-correcting output codes based ensemble feature extraction. *Pattern Recogn*. 2013;46:1091−1100.
49. Zhong G, Cheriet M. Adaptive error-correcting output codes. In: *IJCAI*. 2013:1932−1938.
50. Yang J, Zhang D, Frangi AF, Yang J-Y. Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE Trans Pattern Anal Mach Intell*. 2004;26:131−137.
51. Ye J, Janardan R, Li Q. Two-dimensional linear discriminant analysis. In: *NIPS*. 2004:1569−1576.
52. He X, Cai D, Niyogi P. Tensor subspace analysis. In: *NIPS*. 2005:499−506.
53. Fu Y, Huang TS. Image classification using correlation tensor analysis. *IEEE Trans Image Process*. 2008;17:226−234.
54. Zhong G, Cheriet M. Image patches analysis for text block identification. In: *ISSPA*. 2012:1241−1246.
55. Zhong G, Cheriet M. Large margin low rank tensor analysis. *Neural Comput*. 2014;26:761−780.
56. Jia C, Zhong G, Fu Y. Low-rank tensor learning with discriminant analysis for action classification and image recovery. In: *AAAI*. 2014:1228−1234.
57. Jia C, Kong Y, Ding Z, Fu Y. Latent tensor transfer learning for RGB-d action recognition. In: *ACM Multimedia (MM)*. 2014.
58. Zhong G, Cheriet M. *Low Rank Tensor Manifold Learning*. Cham: Springer International Publishing; 2014:133−150.
59. He X, Niyogi P. Locality preserving projections. In: *NIPS*. 2003.
60. Sammon J. A nonlinear mapping for data structure analysis. *IEEE Trans Comput*. 1969;C-18:401−409.

61. Floyd RW. Algorithm 97: shortest path. *Commun ACM*. 1962;5:345.
62. Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. In: *NIPS*. 2001:585−591.
63. He X, Yan S, Hu Y, Niyogi P, Zhang H. Face recognition using Laplacianfaces. *IEEE Trans Pattern Anal Mach Intell*. 2005;27:328−340.
64. Zhang Z, Zha H. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J Sci Comput*. 2004;26:313−338.
65. Donoho D, Grimes C. Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. *Proc Natl Acad Sci*. 2003;100:5591−5596.
66. Lafon S. *Diffusion Maps and Geometric Harmonics*. Phd thesis. Yale University; 2004.
67. Weinberger K, Saul L. Unsupervised learning of image manifolds by semidefinite programming. *Int J Comput Vis*. 2006;70:77−90.
68. Wang C, Mahadevan S. Manifold alignment using procrustes analysis. In: *ICML*. 2008:1120−1127.
69. Zhong G, Huang K, Hou X, Xiang S. *Local Tangent Space Laplacian Eigenmaps*. New York: SNOVA Science Publishers; 2012:17−34.
70. Cheriet M, Moghaddam R, Arabnejad E, Zhong G. *Manifold Learning for the Shape-based Recognition of Historical Arabic Documents*. Elsevier; 2013:471−491.
71. Xing EP, Ng AY, Jordan MI, Russell SJ. Distance metric learning, with application to clustering with side-information. In: *NIPS*. 2002:505−512.
72. Weinberger KQ, Blitzer J, Saul LK. Distance metric learning for large margin nearest neighbor classification. In: *NIPS*. 2005:1473−1480.
73. Zhong G, Huang K, Liu C-L. Low rank metric learning with manifold regularization. In: *ICDM*. 2011:1266−1271.
74. Zhong G, Zheng Y, Li S, Fu Y. Scalable large margin online metric learning. In: *IJCNN*. 2016:2252−2259.
75. Belkin M, Niyogi P, Sindhwani V. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *J Mach Learn Res*. 2006;7:2399−2434.
76. Lee H, Battle A, Raina R, Ng A. Efficient sparse coding algorithms. In: *NIPS*. 2006:801−808.
77. Dhillon I, Sra S. Generalized nonnegative matrix approximations with bregman divergences. In: *NIPS*. 2005:283−290.
78. Bengio Y. Learning deep architectures for AI. *Found Trends Mach Learn*. 2009;2:1−127.
79. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521:436−444.
80. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proc IEEE*. 1998;86:2278−2324.
81. Sutskever I. *Training Recurrent Neural Networks*. Phd thesis. Univ. Toronto; 2012.
82. Schmidhuber J. Deep learning in neural networks: an overview. *Neural Netw*. 2015;61:85−117.
83. Salakhutdinov R, Hinton G. Deep boltzmann machines. In: *AISTATS*. 2009:448−455.
84. Deng L, Yu D. Deep learning: Methods and applications. *Found Trends Signal Process*. 2014;7:197−387.
85. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol P-A. Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J Mach Learn Res*. 2010;11:3371−3408.
86. Zheng Y, Zhong G, Liu J, Cai X, Dong J. Visual texture perception with feature learning models and deep architectures. In: *CCPR*. 2014:401−410.
87. Hinton G, Roweis S. Stochastic neighbor embedding. In: *NIPS*. 2002:833−840.
88. Pandey G, Dukkipati A. Learning by stretching deep networks. In: *ICML*. 2014:1719−1727.
89. Zheng Y, Cai Y, Zhong G, Chherawala Y, Shi Y, Dong J. Stretching deep architectures for text recognition. In: *ICDAR*. 2015:236−240.
90. Roy P, Zhong G, Cheriet M. Tandem Hmms Using Deep Belief Networks for Offline Handwriting Recognition. *Front Inf Technol Electron Eng*. 2016. http://dx.doi.org/10.1631/FITEE.1600996.
91. Krizhevsky A, Sutskever I, Hinton G. Imagenet classification with deep convolutional neural networks. In: *NIPS*. 2012:1106−1114.
92. Hinton G, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov R. *Improving Neural Networks by Preventing Co-adaptation of Feature Detectors*. 2012. CoRR abs/1207.0580.
93. Nair V, Hinton G. Rectified linear units improve restricted boltzmann machines. In: *ICML*. 2010:807−814.
94. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. *Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks*. 2013. CoRR abs/1312.6229.
95. Simonyan K, Zisserman A. *Very Deep Convolutional Networks for Large-scale Image Recognition*. 2014. CoRR abs/1409.1556.
96. Szegedy C, Liu W, Jia Y, et al. *Going Deeper with Convolutions*. 2014. CoRR abs/1409.4842.
97. He K, Zhang X, Ren S, Sun J. *Deep Residual Learning for Image Recognition*. 2015. CoRR abs/1512.03385.
98. Donahue J, Jia Y, Vinyals O, et al. Decaf: a deep convolutional activation feature for generic visual recognition. In: *ICML*. 2014:647−655.
99. Zhong G, Yao H, Liu Y, Hong C, Pham T. Classification of photographed document images based on deep-learning features. In: *ICGIP*. 2014.
100. Cai Y, Zhong G, Zheng Y, Huang K, Dong J. Is decaf good enough for accurate image classification?. In: *ICONIP*. 2015:354−363.
101. Zhong G, Xu H, Yang P, Wang S, Dong J. Deep hashing learning networks. In: *IJCNN*. 2016:2236−2243.
102. Graves A, Liwicki M, Fernandez S, Bertolami R, Bunke H, Schmidhuber J. A novel connectionist system for unconstrained handwriting recognition. *IEEE Trans Pattern Anal Mach Intell*. 2009;31:855−868.
103. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9:1735−1780.
104. Wand M, Koutník J, Schmidhuber J. Lipreading with long short-term memory. In: *ICASSP*. 2016:6115−6119.
105. Chorowski J, Bahdanau D, Serdyuk D, Cho K, Bengio Y. Attention-based models for speech recognition. In: *NIPS*. 2015:577−585.
106. Bahdanau D, Chorowski J, Serdyuk D, Brakel P, Bengio Y. End-to-end attention-based large vocabulary speech recognition. In: *ICASSP*. 2016:4945−4949.
107. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets. In: *NIPS*. 2014:2672−2680.
108. Deng L, Li X. Machine learning paradigms for speech recognition: an overview. *IEEE Trans Audio Speech Lang Process*. 2013;21:1060−1089.

109. Xu K, Ba J, Kiros R, et al. Show, attend and tell: neural image caption generation with visual attention. In: *ICML*. 2015:2048−2057.
110. Sutskever I, Vinyals O, Le Q. Sequence to sequence learning with neural networks. In: *NIPS*. 2014:3104−3112.
111. Heaton J, Polson N, Witte J. *Deep Learning in Finance*. 2016. CoRR abs/1602.06561.
112. Min S, Lee B, Yoon S. *Deep Learning in Bioinformatics*. 2016. CoRR abs/1603.06430.
113. Erhan D, Bengio Y, Courville A, Manzagol P-A, Vincent P, Bengio S. Why does unsupervised pre-training help deep learning? *J Mach Learn Res*. 2010;11:625−660.
114. Cohen N, Sharir O, Shashua A. On the expressive power of deep learning: a tensor analysis. In: *COLT*. 2016:698−728.
115. Eldan R, Shamir O. The power of depth for feedforward neural networks. In: *COLT*. 2016:907−940.