REVIEW **Communicated by Weizhi Meng**

A Survey on Deep Learning for Multimodal Data Fusion

Jing Gao
gaojing@dlut.edu.cn
School of Software Technology, Dalian University of Technology, Dalian 116620, China, and Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116620, China

Peng Li
lipeng2015@mail.dlut.edu.cn
School of Software Technology, Dalian University of Technology, Dalian 116620, China

Zhikui Chen
zkchen@dlut.edu.cn
School of Software Technology, Dalian University of Technology, Dalian 116620, China, and Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116620, China

Jianing Zhang
zhang1234567893@mail.dlut.edu.cn
School of Software Technology, Dalian University of Technology, Dalian 116620, China

With the wide deployments of heterogeneous networks, huge amounts of data with characteristics of high volume, high variety, high velocity, and high veracity are generated. These data, referred to multimodal big data, contain abundant intermodality and cross-modality information and pose vast challenges on traditional data fusion methods. In this review, we present some pioneering deep learning models to fuse these multimodal big data. With the increasing exploration of the multimodal big data, there are still some challenges to be addressed. Thus, this review presents a survey on deep learning for multimodal data fusion to provide readers, regardless of their original community, with the fundamentals of multimodal deep learning fusion method and to motivate new multimodal data fusion techniques of deep learning. Specifically, representative architectures that are widely used are summarized as fundamental to the understanding of multimodal deep learning. Then the current pioneering multimodal data fusion deep learning models are summarized. Finally, some challenges and future topics of multimodal data fusion deep learning models are described.

Neural Computation **32, 829–864** (2020)
https://doi.org/10.1162/neco_a_01273

1 Introduction

Recently, many heterogeneous networks have been successfully deployed in both low-layer and high-layer applications, including Internet of Things, vehicular networks, and social networks (Zhang, Patras, & Haddadi, 2019; Meng, Li, Zhang, & Zhu, 2019; Qiu, Chen, Li, Atiquzzaman, & Zhao, 2018). With the wide deployment of heterogeneous networks, increasing amounts of data are being generated and collected at an unprecedented speed. These data, often referred to as big data, hold such characteristics as high volume, high variety, high velocity, and high veracity (Gao, Li, & Chen, 2019; Lv, Song, Val, Steed, & Jo, 2017). Also, these huge data that contain structured, semistructured, and unstructured data are multiple-modality/multimodal. And each modality of different source, type, and distribution contains modality-specific information (Li, Yang, & Zhang, 2019; Gao, Li, & Li, 2016). For example, a sports news web page uses images to record the scenes of the sport and texts to describe content of the sport. These images and texts are the descriptions of one event with different raw forms. The reasonable fusion of these multimodal data can help us better understand the event of interest, especially when one modality is incomplete (Khaleghi, Khamis, Karray, & Razavi, 2013; Lahat, Adali, & Jutten, 2015). Thus, with the increasing availibility and accessibility of multimodal data, the fusion of the information in multimodal data is a vital topic in big data research, which provides opportunities to better understand cross-modality and shared-modality information.

Multimodal data fusion, a fundamental method of multimodal data mining, aims to integrate the data of different distributions, sources, and types into a global space in which both intermodality and cross-modality can be represented in a uniform manner (Bramon et al., 2012; Bronstein, Bronstein, Michel, & Paragios, 2010; Poria, Cambria, Bajpai, & Hussain, 2017). It can provide richer information than a single modality by leveraging modality-specific information (Biessmann, Plis, Meinecke, Eichele, & Muller, 2011; Wagner, Andre, Lingenfelser, & Kim, 2011). In the past, some multimodal data fusion methods were presented to explore the complementary and cross-modality information between modalities (Sui, Adali, Yu, Chen, & Calhoun, 2012). For example, Kettenring (1971) proposed the multimodal canonical correlation analysis for the linear intermodality relationship as well as the cross-modality generalization information. Martinez-Montes, Valdes-Sosa, Miwakeichi, Goldman, and Cohen (2004) proposed the partial least squares model linear relationships over multiple variables, discovering the variables from the multi-source data sets. Groves, Beckmann, Smith, and Woolrich (2011) presented a multimodal independent component analysis that is a probabilistic model using the Bayesian framework to combine the independent variables of each different modality. These multimodal data fusion methods are limited to big multimodal data of high volume, high velocity, high variety, and high veracity since they are based on the shallow feature that cannot capture intrinsic internal structures and external relationships in multimodal data (Li, Chen, Yang, Zhang, & Deen, 2018; Zhang, Yang, & Chen, 2016). Thus, fully mining the patterns in the multimodal data requires new multimodal computing techniques.

Multimodal big data, similar to traditional big data, are of high volume, variety, velocity and veracity. However, the variety of the multimodal big data is more prominent than the other characteristics. In particular, multimodal big data are composed of several modalities that contain part of the description of the same things of interest with each modality-independent distribution. There are also complex correlations between modalities. The full modeling of the fusion representations hidden in the intermodality and cross-modality can further improve the performance of various multimodal applications.

Deep learning, a hierarchical computation model, learns the multilevel abstract representation of the data (LeCun, Bengio, & Hinton, 2015). It uses the the backpropagation algorithm to train its parameters, which can transfer raw inputs to effective task-specific representations. There are several well-known deep architectures: convolutional neural networks (CNN), recurrent neural networks (RNN), and generative adversarial networks (GAN) (Bengio, Courville, & Vincent, 2013; Chen & Lin, 2014). These deep learning methods have made great progress in both generative and discriminative tasks based on supervised and unsupervised training strategies (Guo et al., 2016). For example, Han, Kim, and Kim (2017) presented a deep pyramidal residual network by introducing a new residual strategy, which is a representative discriminative task. This pyramidal residual network can learn effective and robust abstract representations in which the task-specific factors are amplified and the irrelevant factors are suppressed, outperforming the state-of-the-art pattern recognition accuracy. A representative generative example is the generative adversarial network that is a game theory paradigm of deep learning (Goodfellow et al., 2014). The generative adversarial network can capture the intrinsic input structure based on the Nash equilibrium between the generator and the discriminator, reconstructing input objects. Also, there are some pioneering deep learning models in multimodal data fusion domains, such as cross-modality retrieval, image annotation, and assistant diagnosis. Although the multimodal data fusion deep learning model has made some progress, it is still in a preliminary stage. Thus, we review the representative multimodal deep learning models to motivate new paradigms of multimodal data fusion deep learning.

In the recent past, enormous amounts of multimodal big data were generated from widely deployed heterogeneous networks. Traditional multimodal data fusion methods cannot properly capture the intermodality representations and the cross-modality complementary correlations of the multimodal big data, since these are shallow models that cannot learn the intrinsic representation of data. Some pioneering work inspired by deep learning methods has proposed exploring the fusion of multimodal data. These deep learning–based multimodal methods have made some progress in various domains, including language translation, image annotation, and medical assistant diagnosis. But the research of deep learning for multimodal data fusion is still in a preliminary stage, and there is no work that reviews multimodal deep learning models. This review of deep learning for multimodal data fusion will provide readers with the fundamentals of the multimodal deep learning fusion method and motivate new multimodal deep learning fusion methods. The representative architectures—DBN, SAE, CNN, and RNN—are summarized because they are fundamental to understanding multimodal deep learning fusion models. Next, the pioneering multimodal deep learning fusion models are summarized from the task, model framework, and data set perspectives. They are grouped by the deep learning architecture used. Finally, some challenges and future topics of deep learning for multimodal data fusion are described.

2 The Representative Deep Learning Architectures

In this section, we introduce representative deep learning architectures of the multimodal data fusion deep learning models. Specifically, the definition, feedforward computing, and backpropagation computing of deep architectures, as well as the typical variants, are presented. The representative models are summarized in Table 1.

2.1 The Deep Belief Net (DBN). The restricted Boltzmann machine (RBM) is the basic block of the deep belief net (Zhang, Ding, Zhang, & Xue, 2018; Bengio, 2009). The RBM is a special variant of the Boltzmann machine (see Figure 1). It consists of a visible layer and a hidden layer; there are fully connected connections between units of the visible layer and units of the hidden layer and but no connections of units in the same layer. The RBM is also a generative graphic model that uses the energy function to capture the probability distribution between visible units and hidden units in the following form,

$$P(x, h) = \frac{e^{-E(x,h)}}{Z}, \tag{2.1}$$

with the normalizing function Z calculated as

$$Z = \sum_{x} \sum_{h} e^{-E(x,h)}, \tag{2.2}$$

Table 1: Summary of the Representative Deep Learning Models.

Architecture	Representative Models	Model Features
Deep belief net	RBM (Zhang et al., 2018)	A generative graphic model that uses the energy to capture the probability distribution between visible units and hidden units.
	SRBM (Chen et al., 2017)	A sparse variant that each hidden unit connects to part of the visible units, preventing the model overfitting based on hierarchical latent tree analysis.
	FRBM (Ning et al., 2018)	A fast variant trained by the lean CD algorithm in which the bounds-based filtering and delta product reduce the redundant dot product calculations.
	TTRBM (Ju et al., 2019)	A compact variant that the parameters between the visible layer and hidden layer are reduced by transforming into the tensor-train format.
Stacked autoencoder	AE (Michael et al., 2018)	A basic fully connected network that uses the encoder-decoder strategy in an unsupervised manner to learn intrinsic features of data.
	DAE (Vincent et al., 2008)	A denoising variant that reconstructs the clear data from the noising data.
	SAE (Makhzani & Frey, 2013)	A sparse variant that captures the sparse representations of the input by adding the constraint into the loss function.
	GAE (Hou et al., 2019)	An adversarial variant that the decoder subnetwork that is also regarded as the generator, adopting game theory to more consistent features with input data.
	FAE (Ashfahani et al., 2019)	An evolving variant that constructs an adaptive network structure in the learning of representations, based on the network significance.
	BAE (Angshul, 2019)	An evolving variant adding the path-loss term in the loss function based on dictionary learning.
Convolutional neural network	Alexnet (Krizhevsky, Sutskever, & Hinton, 2012)	The nonsaturating neurons and the dropout are adopted in the nonlinear computational layers, based on a GPU implementation, respectively.
	ResNet (He et al., 2016)	A shortcut connection is used to cross several layers to back propagate the network loss to previous layers.
	Inception (Christian et al., 2017)	A deeper and wider network is designed by using the uniform grid size for the blocks with auxiliary information.

Table 1: Continued.

Architecture	Representative Models	Model Features
	SEnet (Cao et al., 2019)	Informational embedding and adaption recalibration are regarded as self-attention operations.
	ECNN (Sandler et al., 2018)	The low-rank convolution replaces the full-rank convolution to improve the learning efficiency without much accuracy loss.
Recurrent neural network	RNN (Zhang et al., 2014)	A fully connected network where the self-connection between hidden layers is used to model the time dependency.
	BiRNN (Schuster & Paliwal, 1997)	Two independent computing processes are used to encode the forward and the backward dependency.
	LSTM (Hochreiter & Schmidhuber, 1997)	The memory block is introduced to model the long-time dependency well.
	SRNN (Lei et al., 2018)	A fast variant in which the light recurrence and highway network are proposed to improve the learning efficiency for a parallelized implementation.
	VRNN (Jang et al., 2019)	A variational variant that uses the variational encoder-decoder strategy to model the temporal intrinsic features.

Notes: RBM: restricted Boltzmann machine; SRBM: sparse restricted Boltzmann machine; FRBM: fast restricted Boltzmann machine; TTRBM: tensor-train restricted Boltzmann machine; AE: autoencoder; DAE: denoising autoencoder; SAE: K-sparse autoencoder; GAE: generative autoencoder; FAE: fast autoencoder; BAE: blind autoencoder; Alexnet: Alex convolutional net; ResNet: residual convolutional net; Inception: Inception; SEnet: squeeze excitation network; ECNN: efficient convolutional neural network; RNN: recurrent neural network; BiRNN: bidirectional recurrent neural network; LSTM: long short-term memory; SRNN: slight recurrent neural network; VRNN: variational recurrent neural network.

where x is the visible unit, h represents the hidden unit, and $E\,()$ is the energy function. The energy function is expressed in the following,

$$E\,(x,h) = -\sum_j c_j x_j - \sum_i b_i h_i - \sum_j \sum_i h_i w_{ij} x_j, \tag{2.3}$$

where w_{ij} denotes the weight and c_j and b_i denote the RBM biases.

The visible and hidden marginal distributions of the RBM can be computed as follows:

$$P_\theta\,(x) = \sum_h P_\theta\,(x,h) = \frac{1}{Z}\sum_h e^{-E(x,h)}, \tag{2.4}$$

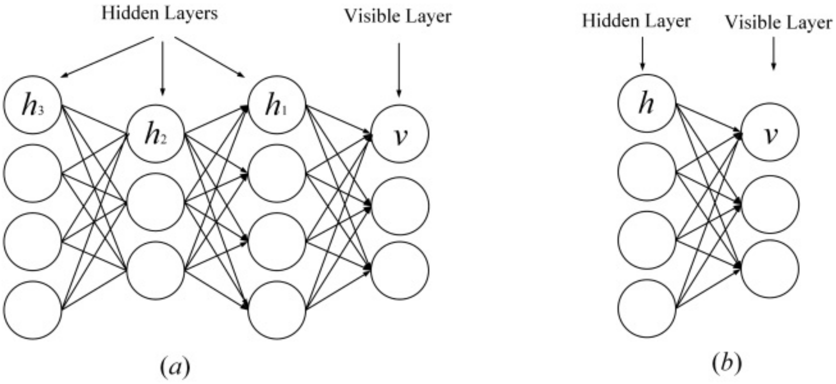


Figure 1: The paradigm of the deep belief net. (a) Deep belief network. (b) Restricted Boltzmann machine.

$$P_\theta\left(h\right)=\sum_x P_\theta\left(x,h\right)=\frac{1}{Z}\sum_x e^{-E(x,h)}. \tag{2.5}$$

Thus, according to Bayes theory, the conditional distributions of the visible and hidden units in the RBM are calculated as follows:

$$P_\theta\left(h|x\right)=\frac{P\left(x,h\right)}{P\left(x\right)}=\frac{e^{-E(x,h)}}{\sum_h e^{-E(x,h)}}=\prod_i P\left(h_i|x\right), \tag{2.6}$$

$$P_\theta\left(x|h\right)=\frac{P\left(x,h\right)}{P\left(h\right)}=\frac{e^{-E(x,h)}}{\sum_x e^{-E(x,h)}}=\prod_i P\left(x_i|h\right). \tag{2.7}$$

More specifically, in the case where the visible and hidden units are binary, the conditional distributions of the visible and hidden units in the RBM are calculated as follows,

$$P_\theta\left(h_k=1|x\right)=f\left(b_k+\sum_{i=1}^{I} w_{ik}x_i\right), \tag{2.8}$$

$$P_\theta\left(x_k=1|h\right)=f\left(c_k+\sum_{j=1}^{J} w_{kj}h_j\right), \tag{2.9}$$

where f is the sigmoid function, w is the weight matrix, b is the visible bias, c is the hidden bias, and I is the number of the visible units, while J is the number of the hidden units.

To model the probability distribution of the training data, the RBM is trained to maximize the marginal probability based on the maximum likelihood principle (Hinton, 2012), with the following loss function:

$$L_\theta = \prod_{i=1}^{n_x} P\left(x^i\right), \tag{2.10}$$

where n_x is the number of the training objects. To maximize that loss function, the gradient ascent algorithm is adopted to update the model parameters as follows:

$$\theta := \theta + \eta \frac{\partial \ln L_\theta}{\partial \theta}, \tag{2.11}$$

where η is the learning rate. And the parameter update $\partial \ln L_\theta / \partial \theta$ is calculated as follows:

$$\begin{aligned}\frac{\partial \ln L_\theta}{\partial \theta} &= \sum_{i=1}^{n_x} \frac{\partial \ln P\left(x^i\right)}{\partial \theta} \\ &= \sum_h P\left(h|x\right)\left(-\frac{\partial E\left(x,h\right)}{\partial \theta}\right) - \sum_{x,h} P\left(x,h\right)\left(-\frac{\partial E\left(x,h\right)}{\partial \theta}\right).\end{aligned} \tag{2.12}$$

To maximize the loss function, equation 2.12 is set to zero. Then the gradients of the weight, the visible bias, and the hidden bias are computed in the following forms:

$$\frac{\partial \ln L_\theta}{w_{ij}} = P\left(h_i = 1|x\right) x_j - \sum_x P\left(x\right) P\left(h_i = 1|x\right) x_j, \tag{2.13}$$

$$\frac{\partial \ln L_\theta}{b_j} = x_j - \sum_x P\left(x\right) x_j, \tag{2.14}$$

$$\frac{\partial \ln L_\theta}{c_i} = P\left(h_i = 1|x\right) - \sum_x P\left(x\right) P\left(h_i = 1|x\right). \tag{2.15}$$

Unfortunately, in those gradient-computing equations, the probability $\sum_x P\left(x\right) P\left(h_i = 1|x\right)$ is difficult to compute (Hinton, Osindero, & Teh, 2006). In fact, the Markov chain Monte Carlo (MCMC) method is used to approximate the probability, such as the contrastive divergence algorithm.

Recently, some advanced RBMs have been proposed to improve performance. For instance, to avoid network overfitting, Chen, Zhang, Yeung, and Chen (2017) designed the sparse Boltzmann machine that learns the network structure based the hierarchical latent tree. Ning, Pittman, and

Shen (2018) introduced fast contrastive-divergence algorithms to RBMs, where the bounds-based filtering and delta product are used to reduce the redundant dot product calculations in computations. To protect the internal structure of multidimensional data, Ju et al. (2019) proposed the tensor RBM, learning the high-level distribution hidden in multidimensional data, in which tensor decomposition is used to avoid the dimensional disaster.

The DBM, a typical deep architecture, is constructed by stacking several RBMs (Hinton & Salakhutdinov, 2006). It is a kind of generative model that can use the energy to capture the joint distribution between the visible objects and the corresponding labels, based on a pretraining and fine-tuning training strategy. In pretraining, each hidden layer is greedily modeled as an RBM trained in the unsupervised strategy. Afterward, each hidden layer is further trained by the discriminative information of training labels in the supervised strategy. DBN has been employed to address problems in many domains, for example, data dimension reduction, representation learning, and semantic hashing. A representative DBM is shown in Figure 1.

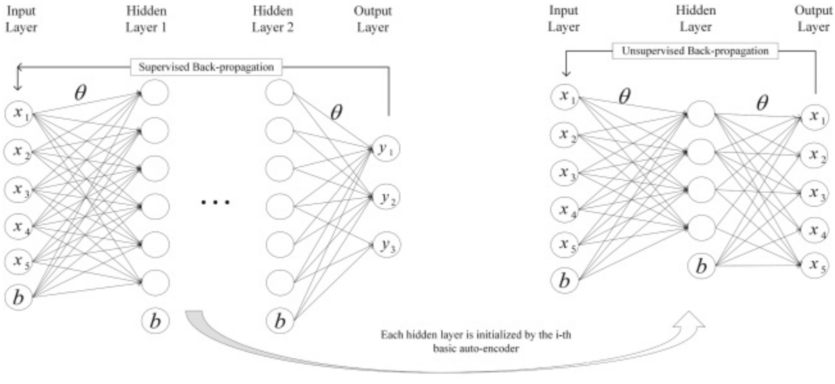
As shown in Figure 1, a DBN with l hidden layers represents the complex correlation in the following form,

$$\begin{aligned} P\left(x, h^1, h^2, \ldots, h^l\right) &= P\left(x|h^1\right) P\left(h^1|h^2\right) \\ &\quad \cdots P\left(h^{l-2}|h^{l-1}\right) P\left(h^{l-1}, h^l\right), \end{aligned} \tag{2.16}$$

where x denotes the input object, $P\left(h^{l-1}|h^l\right)$ represents the conditional distribution of the lth RBM that is composed of the $(l-1)$th and lth layers in the DBN, and $P\left(h^{l-1}, h^l\right)$ is the joint distribution of the top RBM containing the last two layers of the DBN. In equation 2.16, DBN uses the conditional distribution $P\left(h^{l-1}|h^l\right)$ to extract the directed high-level representation and the joint distribution $P\left(h^{l-1}, h^l\right)$ to learn the undirected associative memory.

To obtain conditional and joint distributions, DBN is trained by the unsupervised learning in a layer-wise manner. In other words, each hidden layer is modeled as an RBM. The output of the lower RBM is inputed to the upper one. In detail, the first hidden layer is modeled as an RBM that takes the training data as input, resulting in the empirical distribution of the first DBN hidden layer being approximated by the distribution captured by the RBM. Then the captured approximation distribution is fed to the RBM, that is, the second DBN hidden layer, to further capture the distribution in the training data in the same way. This process is repeated until the last hidden layer is trained.

After unsupervised learning, these parameters—the weights W and hidden biases b—are employed to initialize a deep discriminative neural network of the same architecture, which gives rise to the initialized weights



(a) The traditional stacked auto-encoder

(b) The basic auto-encoder

Figure 2: The paradigm of the stacked autoencoder.

near a good local minimum of the training objects. Then the deep discriminative model generally is further trained by the stochastic gradient descent algorithm to learn the discriminative knowledge in object labels (Wang, Wang, Santoso, Chiang, & Wu, 2018).

2.2 The Stacked Autoencoder (SAE). A stacked autoencoder (SAE) is a typical deep learning model of the encoder-decoder architecture (Michael, Olivier, & Mario, 2018; Weng, Lu, Tan, & Zhou, 2016). It can capture succinct features of the input by transforming the raw input into the intermediate representations in an unsupervised-supervised manner. The SAE has been widely used in many fields, including dimension reduction (Wang, Yao, & Zhao, 2016), image recognition (Jia, Shao, Li, Zhao, & Fu, 2018), and text classification (Chen & Zaki, 2017). A representative SAE is demonstrated in Figure 2.

As shown in Figure 2, SAE is stacked by several basic autoencoders. A basic autoencoder consists of an input layer, a hidden layer, and an output layer. The input layer obtains the raw signal, the intermediate hidden layer encodes the input into a compact hidden representation, and the output layer reconstructs the input signal. In particular, given the data set $\{x_1, x_2, \ldots, x_n\}$, the basic autoencoder encodes the raw signal into the compact representation at the hidden layer as follows:

$$y_h = f\left(W_e x + b\right). \tag{2.17}$$

The basic autoencoder then decodes the intermediate representation using

$$y_o = f\left(W_d x + b\right), \tag{2.18}$$

where y is the activation, W denotes the weight, and b represents the bias. f is the nonlinear function.

To make the autoencoder reconstruct the raw input, the parameters of the autoencoder are optimized by minimizing the average reconstruction error (Zhang, Yang, Chen, & Li, 2018) as follows:

$$w, b = \arg\min_{w,b} \frac{1}{n} \left\| x^i - y_o^i \right\|^2. \tag{2.19}$$

To achieve this minimizing, the stochastic gradient descent strategy is adopted to compute those parameter updates. Those weight updates are computed in the following way,

$$\frac{\partial}{\partial W_{ij}^l} J\left(W, b\right) = \frac{1}{m} \sum_{i=1}^{m} y_j^l \left(\sum_{j=1}^{s_{l+1}} W_{ji}^l \delta_j^{l+1} \right) f'\left(z_i^l\right) + \lambda W_{ij}^l, \tag{2.20}$$

where δ and λ denote the backpropagation loss and weight decay, respectively. Similarly, the bias update is obtained in the following form:

$$\frac{\partial}{\partial b_i^l} J\left(W, b\right) = \delta_i^{l+1}. \tag{2.21}$$

In the past few years, several representative variants have been proposed (Erhan et al., 2010). For example, Vincent, Larochelle, Bengio, and Manzagol (2008) proposed a denoising encoder to learn robust representations from the corrupted inputs. Specifically, each initial input is corrupted into the noising one. The autoencoder takes the corrupted input and reconstructs the clear input as follows:

$$\begin{aligned} y_h &= f\left(W x_n + b\right), \\ y_o &= f\left(W y_h + b\right), \end{aligned} \tag{2.22}$$

where x_n is the corrupted input. Furthermore, to emphasize the corrupted dimensions, a linear combination of the corrupted and uncorrupted reconstruction errors is used to train the denoising model as follows:

$$L_2\left(x, y\right) = \alpha \left(\sum_{i \in c(x)} \left(x_i - y_i\right) \right) + \beta \left(\sum_{i \notin c(x)} \left(x_i - y_i\right) \right), \tag{2.23}$$

where $c\left(x\right)$ denotes the subset of the corrupted inputs. Another representative variant is the sparse autoencoder (Makhzani & Frey, 2013), which captures the sparse representations of the input by adding the constraint into the loss function as follows:

$$L_2\left(x,y\right) = \frac{1}{m}\sum_{j=1}^{m}\left(x_j - y_j\right)^2 + \sum_{i=1}^{n} KL\left(\rho||\rho'\right), \tag{2.24}$$

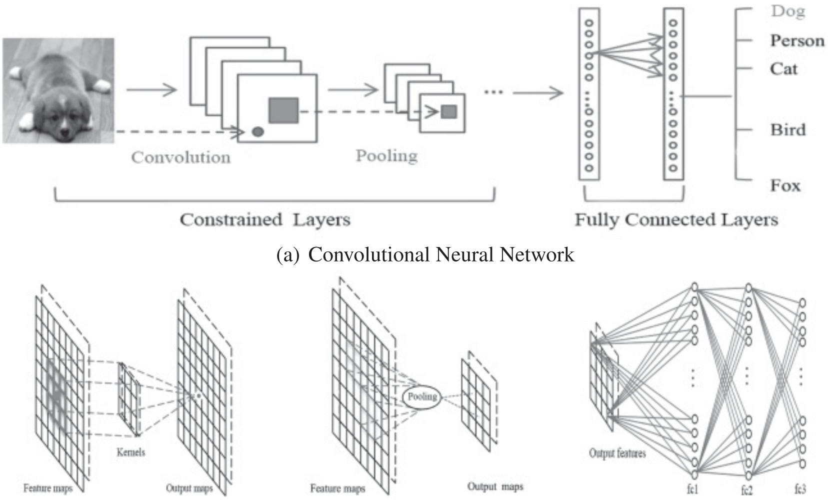
where n is the number of neurons in the hidden layer and $KL\left(\rho||\rho'\right)$ is the KL-divergence given by

$$KL\left(\rho||\rho'\right) = \rho\log\frac{\rho}{\rho'} + (1-\rho)\log\frac{1-\rho}{1-\rho'}. \tag{2.25}$$

To improve the performance of the autoencoder, some adversarial networks are proposed by adopting game theory, in which the decoder is regarded as the generator that tries to trick the discriminator. Those adversarial variants can produce more consistent features with input data (Hou, Sun, Shen, & Qiu, 2019). To analyze stream data, Ashfahani, Pratama, Lughofer, and Ong (2019) proposed a deep evolving denoising autoencoder that constructs an adaptive network structure in the learning of representations, based on the network significance. To model robust features of inputs, Angshul designed a blind denoising autoencoder by adding the path-loss term in the loss function based on dictionary learning (Angshul, 2019). More autoencoder variants can be found in Michael et al. (2018).

As shown in Figure 2, the stacked autoencoder, the most typical fully connected neural network, consists of an input layer, an output layer, and several hidden layers (Sun, Zhang, Hamme, & Zheng, 2016). To learn the compact features of the input, SAE is trained with a two-stage strategy. In the first pretraining, each hidden layer is trained as a basic autoencoder to reconstruct its inputs in the unsupervised manner. For example, the ith hidden layer is initialized as the ith autoencoder. It takes the activations of the $(i-1)$th hidden layer as input. Then it uses the backpropagation algorithm to adjust its parameters by reconstructing the activation of the $(i-1)$th hidden layer. After each of hidden layers is pretrained these above unsupervised way, the stacked autoencoder uses the discriminative knowledge contained in the data labels to fine-tune the parameters to learn task-specific representations. This two-stage training makes the stacked autoencoder avoid local optimal solutions, converging to a better performance.

2.3 The Convolutional Neural Network (CNN). DBN and SAE are fully connected neural networks. In these two networks, each neuron in the hidden layer connects to every neuron of the previous layer, a topology that produces a great number of connections. To train the weights of these connections, the fully connected neural network requires a great number



(a) Convolutional Neural Network

(b) Convolution Layer (c) Pooling Layer (d) Fully-connected Layer

Figure 3: The paradigm of the convolutional neural network.

of training objects to avoid overfitting and underfitting, which is computationally intensive. Also, the fully connected topology does not consider the location information of features contained between neurons. Thus, the fully connected deep neural network—DBN, SAE, and their variants—cannot deal with the high-dimensional data, especially large image and large audio data.

A convolutional neural network is a special kind of deep network that considers the local topology of data (Li, Xia, Du, Lin, & Samat, 2017; Sze, Chen, Yang, & Emer, 2017). A convolutional neural network includes the fully connected network and the constrained network that includes the convolutional and pooling layers. The constrained network use the convolution and pooling operations to achieve the local receptive field and parameter reduction. Like DBN and SAE, the convolutional neural network is also trained by the stochastic gradient descent algorithm. It has made much progress in medical image recognition (Maggiori, Tarabalka, Charpiat, & Alliez, 2017) and semantic analysis (Hu, Lu, Li, & Chen, 2014). A representative CNN is shown in Figure 3.

As shown in Figure 3, given the training objects $\{x_1, x_2, \ldots x_N\}$ with labels $\left\{y_1, y_2, \ldots y_N\right\}$, the CNN uses the convolutional layer to transfer input pattern maps to output feature maps as follows:

$$FM_o = f\left(FM_i * K + b\right), \tag{2.26}$$

where $*$ represents the convolution operation and FM_o and FM_i are the output and input, respectively. K denotes the convolutional kernel. The convolutional layer uses the convolutional operation to force the neuron to perceive a local receptive field of input feature maps. By using this field, the CNN can greatly decrease network parameters.

After each convolutional layer, CNN uses the pooling layer to further deal with the output feature maps. Typically, the max pooling layer is the representative layer that models input maps, as follows,

$$FM_o = f\left(\max\left(FM_I\right)\right), \tag{2.27}$$

where the max() captures the obvious pattern in its receptive field. The max operation achieves the shift invariance.

Finally, CNN uses the fully connected layers to map the hidden features to its corresponding class with the following function,

$$Y = f\left(wI + b\right), \tag{2.28}$$

where w and b are the network parameters.

Similar to the fully connected architecture, the CNN is also trained to fit the training data, using the same algorithm (LeCun et al., 1989; LeCun, Bottou, Bengio, & Haffner, 1998; Zeiler & Fergus, 2014). There are three propagation stages in the back-propagation process. At the beginning, the loss is computed in the same way as with the fully connected architecture.

The second stage is the backpropagation of the loss of the convolutional layer, where the loss is backpropagated to the previous hidden layer as follows:

$$\frac{\partial J}{\partial z^{l-1}_{i,j}} = \sum_m \sum_n w^{l-1}_{m,n} \delta^{l}_{i+m,\,j+n} f'\left(z^{l-1}_{i,j}\right), \tag{2.29}$$

where δ denotes the loss. The update of the kernel weight is computed using

$$\frac{\partial J}{\partial w^{l}_{i,j}} = \sum_m \sum_n \delta^{l}_{m,n} a^{l-1}_{i+m,\,j+n}, \tag{2.30}$$

where a is the activation.

The final stage is the backpropagation of the loss of the pooling layer. Taking the max pooling layer as an example, the loss is computed as follows:

$$\frac{\partial J}{\partial z^{l-1}_{\max}} = \delta^{l}. \tag{2.31}$$

There are no parameters that need to be trained in the pooling layer.

There are some representative CNNs. The most representative one is Alexnet (Krizhevsky, Sutskever, & Hinton, 2012). In Alexnet, the nonsaturating neurons and the dropout technique are adopted in the nonlinear computational layers to improve its performance. Furthermore, a GPU implementation is used to speed up the convolutional layer. He, Zhang, Ren, and Sun (2016) introduced ResNet to solve the accuracy degradation with the increase of depth. In ResNet, a resident block is designed by adding a shortcut connection to a network with several layers, which introduces the identity concept without extra computational cost. By using the resident module, the CNN depth is up to 1000 layers, which greatly contributes to image feature learning. Another example is the Inception-V4, in which a deeper and wider network is designed by using the uniform grid size for the blocks (Christian, Sergey, Vincent, & Alexander, 2017). To explicitly model channel interdependencies, some Squeeze-and-Excitation networks are introduced by using the global informational embedding and adaption recalibration operations, which are regarded as self-attention networks on local-and-global information (Jie, Li, & Sun, 2018; Cao, Xu, Lin, Wei, & Hu, 2019). To improve learning efficiency, some fast convolutional networks are designed by replacing the full-rank convolution with several low-rank convolutions. Those fast implementations can improve learning efficiency without much loss of accuracy (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018; Zhang, Zhou, Lin, & Sun, 2018). More convolutional variants are in Gu et al. (2018).

2.4 The Recurrent Neural Network (RNN). A recurrent neural network is a type of neural computing architecture that deals with serial data (Martens & Sutskever, 2011; Sutskever, Martens, & Hinton, 2011). Unlike deep forward architectures (i.e., DBN, SAE, and CNN), it not only maps the input patterns to the output results but also transfers the hidden states to the outputs by employing the connections between the hidden units (Graves & Schmidhuber, 2008). By using these hidden connections, the RNN models temporal dependency, which results in the sharing of parameters between objects along the time dimension. It has been applied in various domains, such as speech analysis (Mulder, Bethard, & Moens, 2015), image caption (Xu et al., 2015), and language translation (Graves & Jaitly, 2014), achieving outperforming performance. Similar to deep forward architectures, its computing also consists of the forward-pass and backpropagation stages. In forward-pass computing, RNN takes both the input and the hidden state. In backpropagation computing, it uses the backpropagation-through-time algorithm to backpropagate the loss through the time steps. Figure 4 shows a representative RNN.

In Figure 4, given a training object $\left(x^1, \ldots, x^{t-1}, x^t, x^{t+1}, \ldots, x^n\right)$ with the label $\left(y^1, \ldots, y^{t-1}, y^t, y^{t+1}, \ldots, y^n\right)$, the mapping of the RNN is expressed as follows:

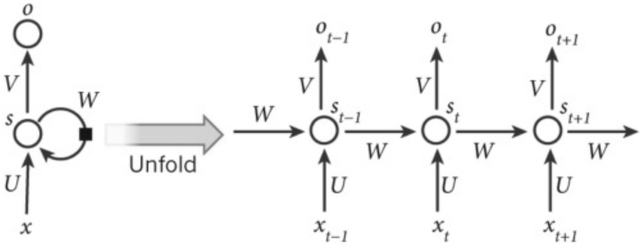


Figure 4: The paradigm of the recurrent neural network.

$$
\begin{aligned}
o^t &= f\left(Vs^t + b\right),\\
s^t &= g\left(Ux^t + Ws^{t-1} + b\right),
\end{aligned}
\tag{2.32}
$$

where f and g are nonlinear functions, b is the bias, and V, U, and W represent the weight parameters. Specifically, the matrix V weights activations of hidden nodes, the matrix U encodes the current inputs, and the matrix W transfers the previous hidden state that contains the temporal dependency.

To fit the sequential training object well, RNN uses the backpropagation-through-time algorithm to train its parameters (Hermans & Schrauwen, 2013; Zheng et al., 2015). In particular, it backpropagates the loss along the time direction via

$$\delta_k^T = \delta_t^T \prod_{i=k}^{t-1} W dig\left[f^{'}\left(z_i\right)\right]. \tag{2.33}$$

Each element of the matrix W updates as follows:

$$\frac{\partial J}{\partial w_{ji}} = \sum_{k=1}^{t} \frac{\partial J}{\partial w_{ji}^{k}} = \sum_{k=1}^{t} \delta_j^k h_i^{k-1}. \tag{2.34}$$

The RNN backpropagates the loss to the previous layer as follows,

$$\left(\delta_t^{l-1}\right)^T = \left(\delta_t^{l}\right)^T U dig\left[f^{'l-1}\left(z_t^{l-1}\right)\right], \tag{2.35}$$

with each element of the matrix U updating as follows:

$$\frac{\partial J}{\partial u_{ji}} = \sum_{k=1}^{t} \frac{\partial J}{\partial u_{ji}^{k}} = \sum_{k=1}^{t} \delta_j^k x_i^{k-1}. \tag{2.36}$$

The matrix V is updated in the same way with the weight of the fully connected layer.

Some well-known variants of the RNN have achieved impressive performance (Zhang, Wang, & Liu, 2014). For example, to model the bidirectional dependency of the sequential data, Schuster and Paliwal (1997) proposed the bidirectional RNN, where there are two independent computing processes that encode the forward dependency and the backward dependency. Another representative variant is LSTM (Hochreiter & Schmidhuber, 1997). This variant can effectively address the limitation that the standard RNN architecture cannot well model the long-time dependency by introducing the memory blocks. To speed up the training of RNNs, Lei, Zhang, Wang, Dai, and Artzi (2018) proposed a light recurrent unit, in which the light recurrence component is used to disentangle the dependency in the state computation and the highway network component is introduced to adaptively combine input and states. Jang, Seo, and Kang (2019) designed the semantic variational recurrent autoencoder to model the global text features in a sentence-to-sentence manner.

The deep RNN is stacked by several recurrent hidden layers with the cyclic connection. Thus, it can capture the deep features of the object direction, as well as the deep features along the time direction.

3 Deep Learning for Multimodal Data Fusion

In this section, we review the most representative multimodal data fusion deep learning models from the perspectives of the model task, model framework, and evaluating data set. They are grouped into four categories based on the deep learning architecture that is used. The representative multimodal deep learning models are summarized in Table 2.

3.1 The Deep Belief Net-Based Multimodal Data Fusion

3.1.1 Example 1. Srivastava and Salakhutdinov (2012) proposed a multimodal generative model based on the deep Boltzmann learning model, learning multimodal representations by fitting the joint distributions of multimodal data over the various modalities, such as image, text, and audio. In this example, the good multimodal representation is defined as follows:

- It should be similar to the raw inputs in the concept.
- It should be easy to get even with certain modalities absent and easy to fill in the lost modalities.
- It should improve classification accuracy and the retrieval tasks of both unified and multiple modalities.

To achieve a multimodal representation that satisfies these three properties, the image-text representation learning is taken as an example.

Table 2: Summary of the Representative Multimodal Deep Learning Models.

Architecture	Representative Model	Model Task	Model Features
DBN based	MDBN (Srivastava & Salakhutdinov, 2012)	Learning the joint distribution over various modalities	Uses the intermodality model to learn the modality-specific feature. Then a one-layer RBM captures the cross-modality distribution.
	DMDBN (Suk et al., 2014)	Diagnosing Alzheimer's disease	Extracts features from MRI and PET, followed a multimodal DBN Then a hierarchical classifier adaptively combines previous results.
	HPMDBN (Ouyang et al., 2014)	Estimating the human pose from multisource information	Two-layer features are extracted from three important pose views. Then an RBM models the joint distributions over multimodal.
	HMDBN (Amer et al., 2018)	Detecting sequential events with discriminative labels	The conditional restricted Boltzmann machine is adopted to extract the intercross-modality features with additional discriminative label information.
	FMDBN (Al-Waisy et al., 2018)	Recognizing faces from local and deep features	Local features of faces are modeled by the Curvelet transform. Then a DBN is built on the local features to learn deep features of faces.
SAE based	MSAE (Ngiam et al., 2011)	Exploring fusion strategies about multimodal data	The multimodality, cross-modality, and shared-modality representation learning methods are introduced based on SAE.
	GHMSAE (Hong et al., 2015)	Generating human skeletons from a series of images	The 2D image and 3D pose are transferred in the high-level skeleton space. Then the joint distributions are modeled by the MSE loss based on SAE.
	MVAE (Khattar et al., 2019)	Detection fake news	Uses the variational encoder-decoder architecture to learn the intrinsic distribution over modalities with detecting loss from the detector.
	AMSAE (Wang et al., 2018)	Learning intrinsic features of words	Uses the multimodal encoder-decoder architecture to model intrinsic features of words with the association and gating mechanisms.
CNN based	MCNN (Ma et al., 2015)	Exploring the image-sentence mapping at different levels	Uses the one-dimensional convolution to capture the image-sentence mapping at word, phase, and sentence levels, taking local topologies into consideration.

Table 2: Continued.

Architecture	Representative Model	Model Task	Model Features
	AMCNN (Frome et al., 2013)	Recognizing objects based on label and unannotated text	Improve the performance of the visual system with the help of dense features extracted from unannotated text.
	AVDCN (Hou et al., 2018)	Enhancing speech signals with auxiliary visual signals	The intermodality CNN maps audio and visual signals into shared semantic space, followed by a fully connected network that reconstructs the raw inputs.
	MFCNN (Nguyen et al., 2019)	Understanding emotion of movie clips	Uses CNN with fuzzy logic to map modality-specific signals into the shared semantic space.
RNN based	MRNN (Mao et al., 2014)	Generating novel descriptions for images	Uses the recurrent network to learn the temporal dependence between sentences and images.
	MBiRNN (Karpathy & Li, 2017)	Generating rich descriptions for images at a glance	Bridges the intermodal relationship between visual features captured by the region CNN and text features captured by BiRNN.
	MTRNN (Abdulnabi et al., 2018)	Labeling indoor scenes from RGB and depth data	Learns the multimodal joint distribution over various modalities by the RNN and transformer layers.
	MGRNN (Narayanan et al., 2019)	Predicting driver behaviors with low-quality data	Uses the gate recurrent cell with the multimodal sensor data to model driver behavior.
	ASMRNN (Sano et al., 2019)	Detecting ambulatory sleep from the wearable device data	Adopts bidirectional LSTMs to temporal features of each modality, followed by a fully connected layer that concatenates temporal features.

Notes: MDBN: multimodal deep Boltzmann machine; DMDBN: diagnosis multimodal deep Boltzmann machine; HPMDBN: human pose deep Boltzmann machine; HMDBN: hybrid multimodal deep Boltzmann machine; FMDBN: face multimodal deep Boltzmann machine; MSAE: multimodal stacked autoencoder; GHMSAE: generating human-skeleton multimodal stacked autoencoder; MVAE: multimodal variational autoencoder; AMSAE: association-gating mechanism multimodal stacked autoencoder; MCNN: multimodal convolutional neural network; AMCNN: auxiliary multimodal convolutional neural network; AVDCN: audiovisual deep convolutional network; MFCNN: multimodal fuzzy convolutional neural network; MRNN: multimodal recurrent neural network; MBiRNN: multimodal bidirectional recurrent neural network; MTRNN: multimodal transformer recurrent neural network; MGRNN: multimodal gating recurrent neural network; ASMRNN: ambulatory sleep multimodal recurrent neural network.

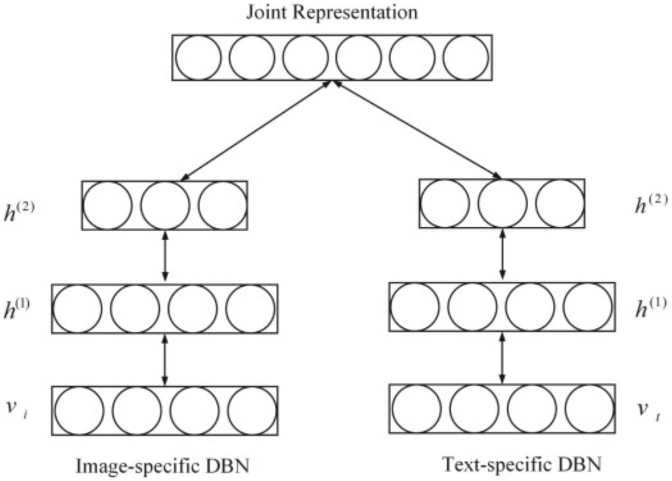


Figure 5: The multimodal deep belief net.

Two modality-specific models are used to transfer the raw-level, high-dimensional image and text into the corresponding high-abstract representations, respectively, since the joint correlations are implicit in the raw-level, high-dimensional inputs. Also, each modality is with a different statistical distribution. After obtaining each high-abstract representation of each modality, a deep Boltzmann model is used to learn the joint distribution over each modality, as shown in Figure 5. In detail, a two-layer deep belief model in which the gaussian RBM and the binary RBM are the first and second hidden layers, respectively, is used to model the image modality, as is a two-layer DBN that is a combination of a replicated softmax layer and an RBM layer to learn text features. Then the one-layer RBM is used to capture the joint representation by feeding the concatenated vector of each learned representation with the following form:

$$P\left(v_m|\theta\right) = \sum_{h^1,h^2} P\left(v_m, h^1, h^2|\theta\right). \tag{3.1}$$

Each module of the proposed multimodal DBN is initialized by the unsupervised layer-wise manner, and an MCMC-based approximate method is adopted for model training.

To evaluate the learned multimodal representation, extensive tasks are carried out, such as the generating missing modality task, the inferring joint representation task, and the discriminative task. Experiments verify that the learned multimodal representation meets the required properties.

3.1.2 Example 2. To effectively diagnose Alzheimer's disease at an early phase, Suk, Lee, Shen, and the Alzheimer's Disease Neuroimaging Initiative (2014) proposed a multimodal Boltzmann model that can fuse the complementary knowledge from the multimodal data. Specifically, to address the limitations caused by the shallow feature learning methods, a DBN is used to learn the deep representations of each modality by transferring the domain-specific representation to the hierarchical abstract representation. Then a one-layer RBM is built on the concatenated vector that is the linear combination of the hierarchical abstract representations from each modality. It is used to learn the multimodal representation by constructing the joint distribution over the different multimodal features. Finally, the proposed model is extensively assessed on the ADNI data set in terms of three typical diagnoses, achieving state-of-the-art diagnosis accuracy.

3.1.3 Example 3. To accurately estimate human poses, Ouyang, Chu, and Wang (2014) designed a multisource deep learning model that learns multimodal representation from mixture type, appearance score, and deformation modalities by extracting the joint distribution of the body pattern in high-order space. In the human-pose multisource deep model, the three widely used modalities are extracted from the pictorial structure models, which combine parts of the body based on conditional random field theory. To get the multimodal data, the pictorial structure model is trained by the linear support vector machine. After that, each of these three features is fed into a two-layer restricted Boltzmann model to capture abstract representations of the high-order pose space from the feature-specific representations. With the unsupervised initialization, each modality-specific restricted Boltzmann model captures the inherent representation of the global space. Then an RBM is used to further learn the human pose representation based on the concatenated vector of the high-level mixture type, appearance score, and deformation representations. To train the proposed multisource deep learning model, a task-specific objective function is designed that considers both body locations and human detection. The presented model is verified on LSP, PARSE and UIUC, and yields up to 8.6% improvement.

Recently some new DBN-based models for multimodal feature learning have been proposed. For instance, Amer, Shields, Siddiquie, and Tamrakar (2018) proposed a hybrid method for sequential event detection, in which the conditional RBM is adopted to extract the intermodality and cross-modality features with additional discriminative label information. Al-Waisy, Qahwaji, Ipson, and Al-Fahdawi (2018) introduced a multimodal method to recognize faces. In this method, a DBN-based model is used to model the multimodal distribution over the local handcrafted features captured by the Curvelet transform, which can merge the advantages of the local and deep features (Al-Waisy et al., 2018).

Table 3: Setting of Multimodal Learning.

	Feature Learning	Supervised Training	Testing
Classic deep learning	Audio	Audio	Audio
	Video	Video	Video
Multimodal fusion	A + V	A + V	A + V
Cross-modality learning	A + V	Video	Video
	A + V	Audio	Audio
Shared representation learning	A + V	Audio	Video
	A + V	Video	Audio

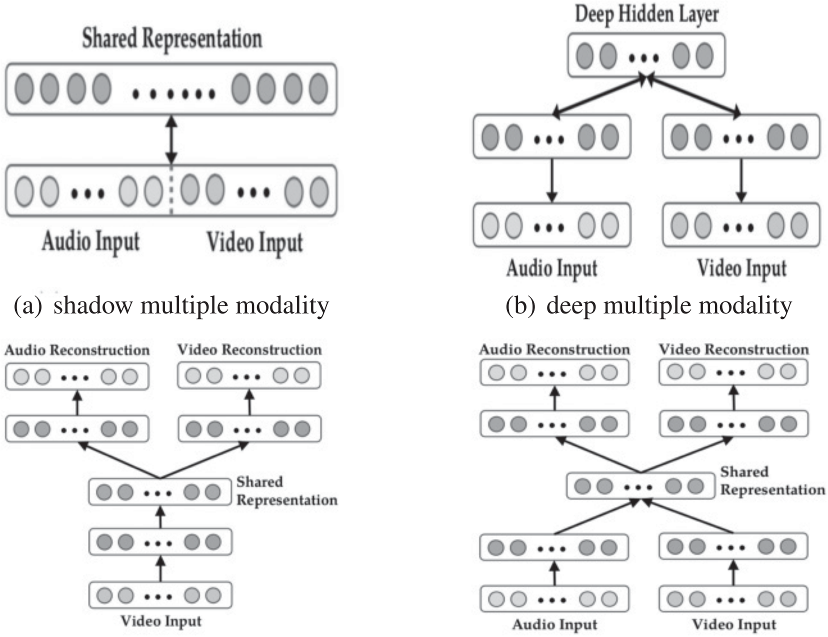
3.1.4 Summary. Those DBN-based multimodal models use the probabilistic graphical network to transfer the modality-specific representations into the semantic features in the shared space. Then the joint distribution over modalities is modeled based on the features of the shared space. Those DBN-based multimodal models are more flexible and robust in unsupervised, semisupervised, and supervised learning strategies. They are well suited to capture informative features of input data. However, they neglect the spatial and temporal topologies of the multimodal data.

3.2 The Stacked Autoencoder-Based Multimodal Data Fusion

3.2.1 Example 4. Multimodal deep learning, presented by Ngiam et al. (2011) is the most representative deep learning model based on the stacked autoencoder (SAE) for multimodal data fusion. This deep learning model aims to address two data-fusion problems: cross-modality and shared-modality representational learning. The former aims to capture better single-modality representations, leveraging knowledge from other modalities, while the latter learns the complex correlation between modalities at a midlevel. To achieve these, three learning scenarios—multiple-modality, cross-modality, and shared-modality learning—are designed, as depicted in Table 3 and Figure 6. Furthermore, in each scenario, to learn better representations, sparse coding is used by penalizing the loss function with the sparse constraints of the following form:

$$\min_{\theta} - \sum \log P\left(v, h\right) + \lambda \sum \left| p - \frac{1}{m} \sum E\left(h|v\right)^2 \right|. \tag{3.2}$$

In a multiple-modality learning scenario, the audio spectrogram and the video frame are concatenated into vectors in a linear manner. The concatenated vector is fed into a sparse restricted Boltzmann machine (SRBM), to learn the correlation between audio and video. This model can learn only the shadow joint representation of multiple modalities since the correlation is implicit in the raw-level high-dimensional representations and the



(a) shadow multiple modality

(b) deep multiple modality

(c) deep cross modality

(d) deep shared modality

Figure 6: The architectures of the multiple-modality, cross-modality, and shared-modality learning.

one-layer SRBM cannot model them. Motivated by this, the concatenated vector of the midlevel representations is fed into SRBM to model the correlation of multiple modalities, which shows better performance.

In the cross-modality learning scenario, a deep stacked multimodal autoencoder is proposed to explicitly learn the correlation between modalities. Specifically, both audio and video are presented as input in the feature learning, and only one of them is fed into the model in the supervised training and testing. This model is initialized in the way of multimodal learning and can model the cross-modality relationship well.

In the shared-modality representation, a modality-specific deep stacked multimodal autoencoder is introduced, motivated by the denoising autoencoder, to explore the joint representation between modalities, especially, when one modality is absent. The training data set that is enlarged by replacing one of modalities with zeros is fed into the model in feature learning.

Finally, detailed experiments are conducted on the CUAVE and AVLetters data sets to evaluate the performance of the multimodal deep learning for task-specific feature learning.

3.2.2 Example 5. To generate visually and semantically effective human skeletons from a series of images, especially videos, Hong, Yu, Wan, Tao, and Wang (2015) proposed a multimodal deep autoencoder to capture the fusion relationship between images and poses. In particular, the proposed multimodal deep autoencoder is trained by a three-stage strategy to construct the nonlinear mapping between two-dimensional images and three-dimensional poses. In the feature fusion stage, the multiview hypergraph low-rank representation is used to construct the inner two-dimensional representation from a series of image features, such as histograms of oriented gradients and shape context, based on manifold learning. In the second stage, a one-layer autoencoder is trained to learn the abstract representation that is used to recover the three-dimensional pose by reconstructing the two-dimensional interimage features. At the same time, a one-layer autoencoder is trained in a similar way to learn the abstract representation of three-dimensional poses. After obtaining the abstract representation of each single modality, a neural network is used to learn the multimodal correlation between the two-dimensional image and the three-dimensional pose by minimizing the squared Euclidean distance between the interrepresentation of the two modalities. The learning of the presented multimodal deep autoencoder is composed of the initialization and the fine-tuning phases. In the initialization, the parameters of each subpart of the multimodal deep autoencoder are copied from the corresponding autoencoder and the neural network. Then the parameters of the whole model are further fine-tuned by the stochastic gradient descent algorithm to construct the three-dimensional pose from the corresponding two-dimensional image.

To evaluate the proposed multimodal deep autoencoder, extensive experiments are conducted on three typical image-pose data sets—Walking, HumanEva-I, and Human 3.6M—outperforming prior models in terms of pose recovery.

Some other representative models based on SAE are proposed to learn the joint distribution over modalities. For example, Wang, Zhang, and Zong (2018) designed a multimodal stacked autoencoder for feature learning of words, which the association and gating mechanisms are adopted to improve the word features. Khattar, Goud, Gupta, and Varma (2019) designed a multimodal variational framework based on the encoder-decoder architecture. This framework is composed of an encoder that models each single-modality feature, a decoder that reconstructs each modality, and a detector for the new detection.

3.2.3 Summary. The SAE-based multimodal models use the encoder-decoder architecture to extract the intrinsic intermodality feature and cross-modality feature by the reconstruction method in an unsupervised manner. Since they are based on SAE, which is a fully connected model, a lot of parameters need to be trained. Also, they neglect the spatial and temporal topologies in the multimodal data.

3.3 The Convolutional Neural Network–Based Multimodal Data Fusion

3.3.1 Example 6. To model the semantic mapping distribution between images and sentences, Ma, Lu, Shang, and Li (2015) proposed a multimodal convolutional neural network. To fully capture the semantic correlations, a three-level fusion strategy—the word level, the phase level, and the sentence level—is designed in an end-to-end architecture. The architecture consists of the image subnetwork, the matching subnetwork, and the multimodal subnetwork. The image subnetwork is a representative deep convolutional neural network, such as Alexnet and Inception, which effectively encodes the image input into a concise representation. The matching subnetwork models the joint representation that associates the image content with the word fragments of sentences in the semantic space.

To deeply integrate the image with the sentence, the word-fragment, phrase-fragment, and sentence-fragment matching networks are devised. The word-fragment matching network is a convolutional neural network that takes the word and the concise image representation as inputs by a one-dimensional convolution and a one-dimensional max-pooling layer with a two-unit window. This word-fragment matching network can achieve the local receptive field, share parameters, and reduce the number of free parameters. The phrase-matching network first transfers the words of each sentence into the phrase fragment that contains more semantic knowledge than the word fragment. Then it models the joint multimodal distributions by using the one-dimensional convolution to combine the phase fragment with image features. Similarly, the sentence-matching network learns the semantic representation of each sentence. After that, it combines the semantic representation of sentences with the image representation at the sentence level. The last evaluating subnetwork uses a multilayer perceptron that evaluates those multimodal joint representations. Finally, an ensemble framework that combines the word, phrase, and sentence multimodal representations is proposed to mine the cross-modality correlation between images and texts.

To evaluate the learned multimodal representation, the multimodal convolutional neural networks are conducted on the Flickr8K and Flickr30K for the bidirectional image and sentence retrieval task.

3.3.2 Example 7. To scale the vision recognition system to an unlimited number of discrete categories, Frome et al. (2013) presented a multimodal convolutional neural network by leveraging the semantic information from text data. This network is composed of the language submodel and the visual submodel. The language submodel is based on the skip-gram model, which can transfer text information into a dense representation of the semantic space. The visual submodel is a representative convolutional neural network, such as Alexnet, that is pretrained on a 1000-class ImageNet data set to capture visual features. To model the semantic relationship between images and texts, the language and visual submodels are combined by a linear projection layer. Each submodel is initialized by parameters from each single modality. After that, to train this visual-semantic multimodal model, a novel loss function is proposed by combining the dot-product similarity and hinge rank loss that can give high similar scores to the correct image and label pairs. This model can yield state-of-the-art performance on the ImageNet data set, avoiding the semantically unreasonable results.

There are also some new CNN-based architectures to learn the multimodal features. For instance, Hou, Wang, Lai, Chang, and Wang (2018) proposed a multimodal speech enhancement framework. In the proposed framework, CNN is used to capture intermodality features in audio and visual signals. Then a fully connected network models the joint distribution by reconstructing the raw inputs. Nguyen, Kavuri, and Lee (2019) introduced a multimodal CNN network to classify the emotion of movie clips. In this multimodal network, the fuzzy logic combined with CNN is used to model intermodality features from audio, visual, and text modalities.

3.3.3 Summary. The CNN-based multimodal models can learn the local multimodal feature between modalities by using the local field and pooling operation. They explicitly model the spatial topologies of the multimodal data. And they are not fully connected models in which the number of parameters is greatly reduced.

3.4 The Recurrent Neural Network–Based Multimodal Data Fusion

3.4.1 Example 8. To generate captions for images, Mao et al. (2014) proposed a multimodal recurrent neural architecture. This multimodal recurrent neural network can bridge the probabilistic correlations between images and sentences. It addresses the limitation of previous work that cannot generate novel image captions, since previous work retrieves the corresponding caption in the sentence database based on the learned image-text mappings. Unlike previous work, the multimodal recurrent neural model (MRNN) learns a joint distribution over the semantic space, based on the given words and image. When an image comes, it generates the sentences word by word, based on the captured joint distribution. Specifically, the multimodal recurrent neural network consists of a language subnetwork, a vision subnetwork, and a multimodal subnetwork, as shown in Figure 7. The language subnetwork is composed of a two-layer word embedding part that captures an effective task-specific representation and a one-layer recurrent neural part that models the temporal dependency of the sentence. The vision subnetwork is essentially a deep convolutional neural network such as Alexnet, Resnet, or Inception, which encodes the high-dimensional image into a compact representation. Finally, the multimodal subnetwork is a

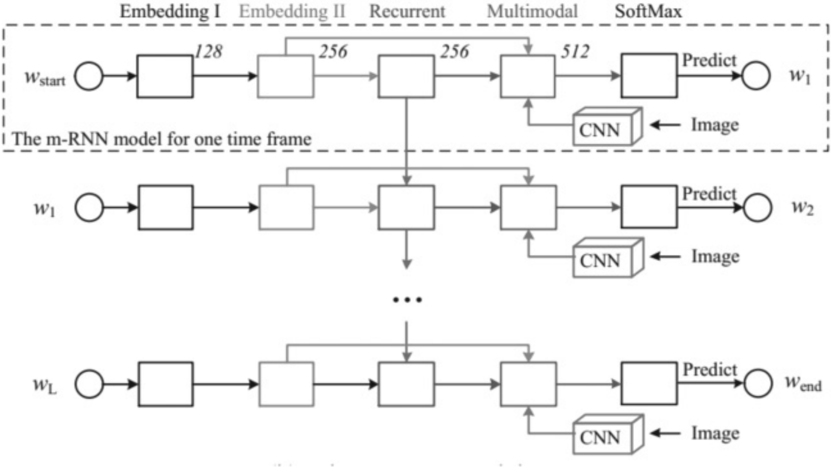


Figure 7: The paradigm of the multimodal recurrent neural network.

hidden network that models the joint semantic distribution over the learned language and vision representation, with the following form:

$$m(t) = g\left(v_w \cdot w(t) + v_r \cdot r(t) + v_I \cdot I\right). \tag{3.3}$$

Furthermore, to train the multimodal recurrent neural model, Mao et al. (2014) use an average log-likelihood loss function based on a standard language evaluation method.

After that, the backpropagation algorithm is used to update parameters of the proposed model. Finally, the image caption, image retrieval, and sentence retrieval tasks are used to evaluate the proposed models on the IAPR TC-12, Flickr 8K, Flickr 30K, and MS COCO data sets. The results show that the proposed model outperforms state-of-the-art models.

3.4.2 Example 9. Aiming to address the limitation that current visual recognition systems cannot generate rich descriptions for images at a glance, a multimodal alignment model is presented by bridging the intermodal relationship between visual and text data (Karpathy & Li, 2017). To achieve that, a twofold scheme is proposed. First, a visual-semantic embedding model is designed to generate the multimodal training data set. Then a multimodal RNN is trained on this data set to generate the rich descriptions of images.

In the visual-semantic embedding model, the region convolutional neural network is used to get the rich image representations that contain enough information on its content corresponding to the sentence. Then a bidirectional RNN is used to encode each sentence into a dense vector of the same dimension with the image representation. Moreover, a multimodal score function is given to measure the semantic similarity between images and sentences. Finally, the Markov random field method is used to generate the multimodal data set.

In the multimodal RNN, a more effective extended model is proposed, which is based on the text content and image input. This multimodal model is composed of a convolutional neural network that encodes the image input and a RNN encodes the image feature and the sentence. This model is also trained by the stochastic gradient descent algorithm. Both of the multimodal models are extensively evaluated on Flickr and Mscoco data sets and achieve state-of-the-art performance.

There are some new RNN-based multimodal deep learning methods. For example, Abdulnabi, Shuai, Zuo, Chau, and Wang (2018) designed a multimodal RNN to label indoor scenes in which the intermodality feature and cross-modality feature are learned by the RNN and transform layers. Narayanan, Siravuru, and Dariush (2019) designed the gate recurrent cell with the multimodal sensor data to model driver behaviors. Sano, Chen, Lopez-Martinez, Taylor, and Picard (2019) proposed a multimodal BiLSTM to detect ambulatory sleep in which the BiLSTM is used to extract features of data collected from wearable devices. Then each intermodality feature is concatenated by a fully connected network.

3.4.3 Summary. The RNN-based multimodal models are able to analyze the temporal dependency hidden in the multimodal data with the help of the explicit state transfer in the computation of hidden units. They use the backpropagation-through-time algorithm to train parameters. Due to the computation in the hidden state transfer, it is difficult to parallelize on the high-performance devices.

4 Summary and Perspectives

Deep learning is an active branch of data mining. Recently, many representative deep learning architectures have been proposed to deal with problems of various domains, such as feature learning, audio compression, and image generation. These representative architectures have made great progress, outperforming other methods in corresponding domains powered by the accessibility of high-volume data. Also, high-performance computing devices, such as, GPU, CPU clusters, and cloud computing platforms are used to improve training efficiency. This explosion and accessibility of multimodal data in heterogeneous networks provide us with vast opportunities to mine the intrinsic knowledge of heterogeneous networks from multiple aspects. These data pose vast challenges on traditional multimodal data mining methods due to their high volume, velocity, variety, and veracity. Some pioneering multimodal deep learning models were presented for data fusion. In this survey, we summarized several multimodal data fusion deep learning models, all built on the current representative deep learning architectures: DBN, SAE, CNN, and RNN. We summarize the models in four groups of multimodal data deep learning models based on DBN, SAE, CNN, and RNN. These pioneering models have made some progress; however, the models are still in the preliminary stage, so there are still challenges.

First, there are a great number of free weights in the multimodal data fusion deep learning models, especially, redundant parameters that have little effect on the task of interest. To train these parameters capturing feature structures of data, large amounts of data are fed into the multimodal data fusion deep learning models based on the backpropagation algorithm, which is computing intensive and time-consuming. To increase weight-learning efficiency, some parallel variants of the backpropagation algorithm have been executed on computation-intensive architectures: CPU cluster, GPU, and cloud platforms. In turn, the scale of multimodal data fusion deep learning models greatly depends on the computing capability of the training devices. However, the increased speed of the computing capability of the current high-performance devices falls behind that of the multimodal data. The multimodal data fusion deep learning models trained on high-performance computing devices of the current architecture may not learn feature structures of the multimodal data of increasing volume well. Therefore, one future research possibility of deep learning on the fusion feature learning of multimodal data is to design new learning frameworks with more powerful computing architectures. In addition, the compression of free parameters, an effective way to enhance training efficiency in deep learning for single-modality data feature learning has made great progress. Thus, how to combine the current compression strategy to design new compression methods of multimodal deep learning is also a potential research direction.

Second, multimodal data contain not only intermodality information but also abundant cross-modality information. To learn the abundant intermodality and crossmodality information of multimodal data, most existing deep learning models for multimodal data fusion first use a deep model to capture the private features from each modality, transforming the modality-specific raw representation to a high-abstraction representation in a certain global space. Then these high abstraction representations are further concatenated into a vector that represents the global representation of the multimodal. Finally, a deep model is used to model high-abstract representations from the concatenated vectors. However, by using this method, the multimodal deep learning models cannot capture the fully semantic knowledge of the multimodal data. There are no clear explanations why these single intermodality features, the representations of the same semantic space, which can give rise to the combination of features of different semantic levels, lose cross-modality information. Also, the intermodality representations are concatenated in a linear fashion that cannot fit the complex relationships over multiple modalities. With the exploration of the multimodal data, three or more modalities are combined to mine the intermodality and crossmodality knowledge. The current multimodal data fusion deep learning models may not achieve the desired results. Thus, new deep learning models for multimodal data that take semantic relationships into consideration are urgently needed. In addition, some semantic fusion strategies—for example, multiview fusion, transfer learning fusion, and probabilistic dependency fusion—have made some progress in the semantic fusion of the multimodal data. Thus, the combination of deep learning and semantic fusion strategies may be a way to solve the challenges posed by the exploration of multimodal data.

Third, multimodal data are collected from dynamic environments, indicating that the data are uncertain. That is, these data are dynamic, which means that the distribution of data is not unchanged. The traditional method of multimodal deep learning to learn dynamic multimodal data is to train a new model when the data distribution changes. However, it takes too much time to train a new deep learning model, and it cannot satisfy online multimodal data applications. Online learning and incremental learning are the representative real-time strategies that learn the new knowledge of the new data without much loss of historical knowledge. Thus, with the explosion of the dynamic multimodal data, the design of online and incremental multimodal deep learning models for data fusion must be addressed. Also, the multimodal data are low quality and contain noise, incomplete data, and outliers. Currently, several deep learning models are focusing only on single-modality noisy data. With the explosion of low-quality multimodal data, a deep learning model for low-quality multimodal data needs to be addressed urgently.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grants 61602083 and 61672123, the Doctoral Scientific Research Foundation of Liaoning Province 20170520425, the Dalian University of Technology Fundamental Research Fund under grant DUT15RC(3)100, and the China Scholarship Council.

References

Abdulnabi, A. H., Shuai, B., Zuo, Z., Chau, L., & Wang, G. (2018). Multimodal recurrent neural networks with information transfer layers for indoor scene labeling. *IEEE Transactions on Multimedia*, *20*(7), 1656–1671.

Al-Waisy, A. S., Qahwaji, R., Ipson, S., & Al-Fahdawi, S. (2018). A multimodal deep learning framework using local feature representations for face recognition. *Machine Vision and Applications*, *29*, 35–54.

Amer, M. F., Shields, T., Siddiquie, B., & Tamrakar, A. (2018). Deep multimodal fusion: A hybrid approach. *International Journal of Computer Vision*, *126*(2–4), 440–456.

Angshul, M. (2019). Blind denoising autoencoder. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(1), 312–317.

Ashfahani, A., Pratama, M., Lughofer, E., & Ong, Y. S. (2019). *DEVDAN: Deep evolving denoising autoencoder*. arXiv:1910.04062v1.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, *2*(1), 1–127.

Bengio, Y., Courville, A. C., & Vincent, P. (2013). Representation learning: A review and new Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828.

Biessmann, F., Plis, S., Meinecke, F. C., Eichele, T., & Muller, K. (2011). Analysis of multimodal neuroimaging data. *IEEE Reviews in Biomedical Engineering*, *4*, 26–58.

Bramon, R., Boada, I., Bardera, A., Rodriguez, J., Feixas, M., Puig, J., & Sbert, M. (2012). Multimodal data fusion based on mutual information. *IEEE Transactions on Visualization and Computer Graphics*, *18*(9), 1574–1587.

Bronstein, M. M., Bronstein, A. M., Michel, F., & Paragios, N. (2010). Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3594–3601). Washington, DC: IEEE Computer Society.

Cao, Y., Xu, J., Lin, S., Wei, F., & Hu, H. (2019). *GCNet: Non-local networks meet squeeze excitation networks and beyond*. arXiv:1904.11492v1.

Chen, X. W., & Lin, X. (2014). Big data deep learning: Challenges and perspectives. *IEEE Access*, *2*, 514–525.

Chen, Y., & Zaki, M. J. (2017). KATE: K-competitive autoencoder for text. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 85–94). New York: ACM.

Chen, Z., Zhang, N. L., Yeung, D. Y., & Chen, P. (2017). Sparse Boltzmann machines with structure learning as applied to text analysis. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (pp. 1805–1811). Palo Alto, CA: AAAI.

Christian, S., Sergey, I., Vincent, V., & Alexander, A. A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (pp. 4278–4284). Palo Algo: AAAI.

Erhan, D., Bengio, Y., Courville, A. C., Manzagol, P. A., Vincent, P., & Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, *11*, 625–660.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A., & Mikolov, T. (2013). DeViSE: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems, 26* (pp. 2121–2129). Red Hook, NY: Curran Associates, Inc.

Gao, J., Li, P., & Chen, Z. (2019). A canonical polyadic deep convolutional computation model for big data feature learning in Internet of Things. *Future Generation Computer Systems*, *99*, 508–516.

Gao, J., Li, J., & Li, Y. (2016). Approximate event detection over multimodal sensing data. *Journal of Combinatorial Optimization*, *32*(4), 1002–1016.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., . . . Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, *27* (pp. 2672–2680). Red Hook, NY: Curran.

Graves, A., & Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *Proceedings of the 31st International Conference on Machine Learning* (pp. 1764–1772).

Graves, A., & Schmidhuber, J. (2008). Offline handwriting recognition with multidimensional recurrent neural networks. In D. Koller, D. Schumann, Y. Bengio, & L. Bottou (Eds.), *Advances in neural information processing systems*, *21* (pp. 545–552). Cambridge, MA: MIT Press.

Groves, A. R., Beckmann, C. F., Smith, S. M., & Woolrich, M. W. (2011). Linked independent component analysis for multimodal data fusion. *NeuroImage*, *54*(3), 2198–2217.

Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., . . . Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition 77*, 354–377.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27–48.

Han, D., Kim, J., & Kim, J. (2017). Deep pyramidal residual networks. In *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6307–6315). Washington, DC: IEEE Computer Society.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778). Washington, DC: IEEE Computer Society.

Hermans, M., & Schrauwen, B. (2013). Training and analyzing deep recurrent neural networks. In C. J. C. Burges, L. Bottou, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, *26* (pp. 190–198). Red Hook, NY: Curran.

Hinton, G. E. (2012). A practical guide to training restricted Boltzmann machines. In G. Montavon, G. B. Orr, & K.-R. Müller (Eds.), *Neural networks: Tricks of the trade* (pp. 599–619). Berlin: Springer.

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hong, C., Yu, J., Wan, J., Tao, D., & Wang, M. (2015). Multimodal deep autoencoder for human pose recovery. *IEEE Transactions on Image Processing*, *24*(12), 5659–5670.

Hou, J., Wang, S., Lai, Y., Chang, H., & Wang, H. (2018). Audio-visual speech enhancement using multimodal deep convolutional neural networks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, *2*(2), 117–128.

Hou, X., Sun, K. D., Shen, L., & Qiu, G. (2019). Improving variational autoencoder with deep feature consistent and generative adversarial training. *Neurocomputing*, *341*, 183–194.

Hu, B., Lu, Z., Li, H., & Chen, Q. (2014). Convolutional neural network architectures for matching natural language sentences. In Z. Ghahramani, M. Welling,

C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, *27* (pp. 2042–2050). Red Hook, NY: Curran.

Jang, M., Seo, S., & Kang, P. (2019). Recurrent neural network-based semantic variational autoencoder for sequence-to-sequence learning. *Information Sciences*, *490*, 59–73.

Jia, C., Shao, M., Li, S., Zhao, H., & Fu, Y. (2018). Stacked denoising tensor autoencoder for action recognition with spatiotemporal corruptions. *IEEE Transactions on Image Processing*, *27*(4), 1878–1887.

Jie, H., Li, S., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7132–7141). Piscataway, NJ: IEEE.

Ju, F., Sun, Y., Gao, J., Antolovich, M., Dong, J., & Yin, B. (2019). Tensorizing restricted Boltzmann machine. *ACM Transactions on Knowledge Discovery from Data*, *13*(3), 30:1–16.

Karpathy, A., & Li, F. F. (2017). Deep visual-semantic alignments for generating image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(4), 664–676.

Kettenring, J. R. (1971). Canonical analysis of several sets of variables. *Biometrika*, *58*(3), 433–451.

Khaleghi, B., Khamis, A. M., Karray, F., & Razavi, S. N. (2013). Multisensor data fusion: A review of the state-of-the-art. *Information Fusion*, *14*(1), 28–44.

Khattar, D., Goud, J. S., Gupta, M., & Varma, V. (2019). MVAE: Multimodal variational autoencoder for fake news detection. In *Proceeding of 2019 the World Wide Web Conference* (pp. 2915–2921). New York: ACM.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, *25* (pp. 1106–1114). Red Hook, NY: Curran.

Lahat, D., Adali, T., & Jutten, C. (2015). Multimodal data fusion: An overview of methods, challenges, and prospects. *Proceedings of the IEEE*, *103*(9), 1449–1477.

LeCun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Lei, T., Zhang, Y., Wang, S. I., Dai, H., & Artzi, Y. (2018). Simple recurrent units for highly parallelizable recurrence. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 4470–4481). Stroubsburg, PA: Association for Computational Linguistics.

Li, E., Xia, J., Du, P., Lin, C., & Samat, A. (2017). Integrating multilayer features of convolutional neural networks for remote sensing scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(10), 5653–5665.

Li, P., Chen, Z., Yang, L. T., Zhang, Q., & Deen, M. J. (2018). Deep convolutional computation model for feature learning on big data in Internet of Things. *IEEE Transactions on Industrial Informatics*, *14*(2), 790–798.

Li, Y., Yang, M., & Zhang, Z. (2019). A survey of multi-view representation learning. *IEEE Transactions on Knowledge and Data Engineering*, *31*(10), 1863–1883.

Lv, Z., Song, H., Val, P. B., Steed, A., & Jo, M. (2017). Next-generation big data analytics: State of the art, challenges, and future research topics. *IEEE Transactions on Industrial Informatics*, *13*(4), 1891–1899.

Ma, L., Lu, Z., Shang, L., & Li, H. (2015). Multimodal convolutional neural networks for matching image and sentence. In *Proceedings of 2015 IEEE International Conference on Computer Vision* (pp. 2623–2631). Washington, DC: IEEE Computer Society.

Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(2), 645–657.

Makhzani, A., & Frey, B. (2013). *K-sparse autoencoders*. arXiv:1312.5663v2.

Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., & Yuille, A. (2014). *Deep captioning with multimodal recurrent neural networks (m-RNN)*. arXiv:1412.6632.

Martens, J., & Sutskever, I. (2011). Learning recurrent neural networks with Hessian-free optimization. In *Proceedings of the 28th International Conference on Machine Learning* (pp. 1033–1040). Madison, WI: Omnipress.

Martinez-Montes, E., Valdes-Sosa, P. A., Miwakeichi, F., Goldman, R. I., & Cohen, M. S. (2004). Concurrent EEG/fMRI analysis by multiway partial least squares. *NeuroImage*, *22*(3), 1023–1034.

Meng, W., Li, W., Zhang, & Zhu, L. (2019). Enhancing medical smartphone networks via blockchain-based trust management against insider attacks. *IEEE Transactions on Engineering Management*. doi:10.1109/TEM.2019.2921736

Michael, T., Olivier, B., & Mario, L. (2018). *Recent advances in autoencoder-based representation learning*. arXiv:1812.05069v1

Mulder, W. D., Bethard, S., & Moens, M. F. (2015). A survey on the application of recurrent neural networks to statistical language modeling. *Computer Speech and Language*, *30*(1), 61–98.

Narayanan, A., Siravuru, A., & Dariush, B. (2019). *Temporal multimodal fusion for driver behavior prediction tasks using gated recurrent fusion units*. arXiv:1910.00628.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. In *Proceedings of 28th International Conference on Machine Learning* (pp. 689–696). Madison, WI: Omnipress.

Nguyen, T., Kavuri, S., & Lee, M. (2019). A multimodal convolutional neuro-fuzzy network for emotion understanding of movie clips. *Neural Networks*, *118*, 208–219.

Ning, L., Pittman, R., & Shen, X. (2018). LCD: A fast contrastive divergence based algorithm for restricted Boltzmann machine. *Neural Networks*, *108*, 399–410.

Ouyang, W., Chu, X., & Wang, X. (2014). Multi-source deep learning for human pose estimation. In *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2337–2344). Washington, DC: IEEE Computer Society.

Poria, S., Cambria, E., Bajpai, R., & Hussain, A. (2017). A review of affective computing: From unimodal analysis to multimodal fusion. *Information Fusion*, *37*, 98–125.

Qiu, T., Chen, N., Li, K., Atiquzzaman, M., & Zhao, W. (2018). How can heterogeneous Internet of things build our future: A Survey. *IEEE Communications Surveys and Tutorials*, *20*(3), 2011–2027.

Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the 2018*

IEEE Conference on Computer Vision and Pattern Recognition (pp. 4510–4520). Piscataway, NJ: IEEE.

Sano, A., Chen, W., Lopez-Martinez, D., Taylor, S., & Picard, R.W. (2019). Multimodal ambulatory sleep detection using LSTM recurrent neural networks. *IEEE Journal of Biomedical and Health Informatics*, *23*(4), 1607–1617.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681.

Srivastava, N., & Salakhutdinov, R. (2012). Multimodal learning with deep Boltzmann machines. In P. L. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, *25* (pp. 2231–2239). Red Hook, NY: Curran.

Sui, J., Adali, T., Yu, Q., Chen, J., & Calhoun, V. D. (2012). A review of multivariate methods for multimodal fusion of brain imaging data. *Journal of Neuroscience Methods*, *204*(1), 68–81.

Suk, H. I., Lee, S. W., Shen, D., & Alzheimer's Disease Neuroimaging Initiative. (2014). Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. *NeuroImage*, *101*, 569–582.

Sun, M., Zhang, X., Hamme, H. V., & Zheng, T. F. (2016). Unseen noise estimation using separable deep auto encoder for speech enhancement. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, *24*(1), 93–104.

Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning* (pp. 1017–1024). Madison, WI: Omnipress.

Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, *105*(12), 2295–2329.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceeding of the 25th International Conference on Machine Learning* (pp. 1096–1103). New York: ACM.

Wagner, J., Andre, E., Lingenfelser, F., & Kim, J. (2011). Exploring fusion methods for multimodal emotion recognition with missing data. *IEEE Transactions on Affective Computing*, *2*(4), 206–218.

Wang, C. Y., Wang, J. C., Santoso, A., Chiang, C. C., & Wu, C. H. (2018). Sound event recognition using auditory-receptive-field binary pattern and hierarchical-diving deep belief network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(8), 1336–1351.

Wang, S., Zhang, J., & Zong, C. (2018). Associative multichannel autoencoder for multimodal word representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 115–124). Stroudsburg, PA: Association for Computer Language.

Wang, Y., Yao, H., & Zhao, S. (2016). Auto-encoder based dimensionality reduction. *Neurocomputing*, *184*, 232–242.

Weng, R., Lu, J., Tan, Y., & Zhou, J. (2016). Learning cascaded deep auto-encoder networks for face alignment. *IEEE Transactions on Multimedia*, *18*(10), 2066–2078.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., . . . Bengion, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the 32nd International Conference on Machine Learning* (pp. 2048–2057).

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Proceedings of the 13th European Conference on Computer Vision* (pp. 818–833). Zurich: Springer.

Zhang, H., Wang, Z., & Liu, D. (2014). A comprehensive review of stability analysis of continuous-time recurrent neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *25*(7), 1229–1262.

Zhang, N., Ding, S., Zhang, J., & Xue, Y. (2018). An overview on restricted Boltzmann machines. *Neurocomputing*, *275*, 1186–1199.

Zhang, Q., Yang, L. T., & Chen, Z. (2016). Deep computation model for unsupervised feature learning on big data. *IEEE Transactions on Services Computing*, *9*(1), 161–171.

Zhang, Q., Yang, L. T., Chen, Z., & Li, P. (2018). A survey on deep learning for big data. *Information Fusion*, *42*, 146–157.

Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018) ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6848–6856). Piscataway, NJ: IEEE.

Zhang, Z., Patras, P., & Haddadi, H. (2019). Deep learning in mobile and wireless networking: A survey. *IEEE Communications Surveys and Tutorials*, *21*(3), 2224–2287.

Zheng, S., Jayasumana, S., Paredes, B. R., Vineet, V., Su, Z., Du, D., . . . Torr, P. H. S. (2015). Conditional random fields as recurrent neural networks. In *Proceedings of 2015 IEEE International Conference on Computer Vision* (pp. 1529–1537). Washington, DC: IEEE Computer Society.

Received August 17, 2019; accepted December 11, 2019.