Semantic Deep Learning

Hao Wang

September 29, 2015

Abstract

Artificial intelligence and machine learning research is dedicated to building intelligent artifacts that can imitate or even transcend the cognitive abilities of human beings. To emulate human cognitive abilities with intelligent artifacts, one must first render machines capable of capturing critical aspects of sensory data, with adequate data representations and performing reasoning and inference with formal knowledge representations. In recent years, the research in deep learning and knowledge engineering has made wide impact on the two problems of data and knowledge representations. Deep learning is a set of machine learning algorithms that attempt to model data representations through many layers of non-linear transformations. Hierarchical, distributed, and efficient data representations can be learned through deep learning models with proper training algorithms. The learned data representation can disentangle the hidden explanation factors and variations in the input data that are critical for further artificial intelligence and machine learning tasks. Additionally, the research in knowledge engineering has frequently focused on modeling the high level human cognitive abilities, such as reasoning, making inferences, and validation. The formal knowledge representation facilitates knowledge reusing and sharing in a machine processable way. It also promotes many advances in the field of semantic data mining which refers to the data mining tasks that systematically incorporate domain knowledge, especially formal semantics, into the data mining process. Empirical studies have attested that formal knowledge representations can make positive influences in all stages of both the data mining and machine learning processes. Inspired by the success of both deep learning and semantic data mining, we hypothesize that formal knowledge representations have the potential to assist in the deep learning process as well. In this report, we summarize the advances in both deep learning and semantic data mining in recent years. We illustrate how learning models with deeper architectures are capable of constructing better data representations for further artificial intelligence and machine learning tasks. We also demonstrate how formal knowledge representation can assist in data mining process at all data mining stages, from various perspectives. At last, we present our thoughts and intuitions on semantic deep learning, which addresses the topic of learning deep data representation with the assistance of formal knowledge representation.

1 Introduction

Human beings have long been picturing the dreaming of building intelligent artifacts that can comprehend and imitate human cognitive intelligent behaviors. Ancient Greeks passed down tales of conceived intelligent objects, such as intelligent anvils and hammers that can manufacture weapons and animated statues that can play chess. Science fictions envision our future with pervasive artificial intelligences that could automate routine labor, understand speech or images, make diagnoses

in medicine, and support basic scientific research. Since the very first devising of a computer, people have started to conceive the idea that it could become intelligent. Using computers to model our world and to proceed with intelligent tasks has been the main focus of artificial intelligence (AI) [137] and machine learning (ML) [115] research over the past century. Today, AI and ML are thriving fields with many practical applications and active research topics.

Using computers to transcend human limits and build artifacts and solutions to problems is an indeed breathless undertaking. However, such tasks have never been trivial ones. Many progresses in AI and ML research have been made to understand and improve learning algorithms. However, human beings exhibits such exceptionally complex intelligent abilities that contemporary research has not yet obtained satisfactory solutions for most of AI and ML tasks. We do not yet have algorithms that can understand scenes and describe them in natural language, except in very limited settings. We lack of methods that can infer enough semantic concepts to interact with most humans using these concepts. If we consider image recognition, one of the most fundamental AI tasks, we found that we do not yet have algorithms that can discover the semantic concepts from image that are necessary to interpret most of them.

The complexity of human intelligent behaviors mostly derives from the sophisticated yet delicate structure of the cognitive center of human being, the human brain. The human brain is a complicated network contains around eighty-six billion neurons. Contemporary neural science research has demonstrated that such neural networks of the human brain should account for almost all of the cognitive activities of human being. Science and technology often treat natural inspirations with great caution; so is artificial intelligence. When computer scientists first start to conceive the idea of building intelligent artifacts with ability of learning and reasoning, an artificial neural network (ANN) [18, 115] seems to be the most natural and rational counter part of human brain.

1.1 The importance of depth: From Neural Network to Deep Learning

Early designs of ANNs are usually feedforward networks with large amounts of connected, non-linear processing units, called artificial neurons, organized in layers. While ANNs have made pronounced progresses in many AI and ML applications, many challenges persist in ANNs, and their performances are often far from satisfactory [54]. The deficiencies of ANNs come from multiple facets, of which the most important one should account for the difficulty to build an effective ANN with deep-architecture (non-linear transformation with more than 3 layers) [19, 54]. Previous empirical studies have shown that deeper neural networks, with larger quantities of processing units and greater depth, were generally found to be not better, and often worse, than neural networks with one or two hidden layers [169]. The reason for such roughness should mostly account for the difficulty in training; i.e., classical training methods that proved effective for shallow-architectures are not as effective when adapted to deeper models.

For example, back-propagation (BP) [135] was the first and most popular computational model for the training of feed-forward ANNs. Training ANN with BP requires labeled data, which are, in practice, hard to obtain with a great amount. Adding more layers to a neural network also diminishes the error back-propagated to the lower layers, which makes the training process hard to converge [176, 64]. Overfittings are pervasive when training complex models with a large number of parameters. The training of BP usually accounts for the local gradient information with a random initial point. As the feature space of ANN is often highly non-linear, the training gets trapped in poor local optimals and plateaus easily [16]. Such severity increases drastically along with the

depths of the network. Therefore, for many similar reasons similar to the ones above, in the past two decades, AI and ML researchers have often preferred to limit the depth of machine learning models to within only one or two layers. The trends of machine learning models once shifted from neural network models to shallow models with a mostly convex loss function.

The depth issue of machine learning models persisted until 2006, when the first break through was made in designing and training with many layers of Restricted Boltmann Machines (RBM) [66, 64]. RBM is a probabilistic graphic model that can be interpreted as both a generative model and a stochastic neural network. This special property renders RBM the ability to make good use of both unlabeled data and labeled data in the training process. As a generative model, RBM learns a representation of data, instead of classifying them as in the traditional neural networks. Training with unlabeled data proceeds in a greedy layer-wise pre-training process that has a time complexity linear to the size of the network. Through the pre-training process, the DBN parameters are initialized to a point that is closer to the global optimum. This alleviates the pervasive local optimal problems that occur in the traditional training of neural networks. As a discriminative model, fine-tuning through back-propagation further adjusts the model parameters with labeled data as a stochastic neural network. While it is still possible for DBNs to fall into local optimal, they now have much larger probabilities to stay closer with the global optimum, due to the pre-training.

In the following years, RBM based deep networks demonstrated exceptional performance in many AI and ML tasks [34, 52, 117]. The importance of depth in learning models raised wide attentions in AI and ML research communities. The term *deep learning* is formally defined as the learning model with many layers of non-linear transformations. The machine learning model in deep learning is often called *deep-architecture*, while previous machine learning machine models with only transformation of only one of two layers are often called *shallow-architecture*. Later on, AI and ML communities started to ponder about deep learning models in various other forms. The deep convolution neural networks (CNN) were designed to effectively train data with topological structures and strong local correlations, such as image and speech [88, 70]. Deep conditional RBM were proposed to model time-series data, especially data of human motion [117]. Natural language processing often uses deep recurrent neural networks, in which dependencies from previous inputs could well assist in the prediction of the next word [150, 151]. In the domain of speech recognition, artificial neural network based Hidden Markov Model (ANN-HMM) replaced the traditional Gaussian Mixture Hidden Markov Model (GMM-HMM) with better recognition accuracy [38, 63]. Many researcher have also detected performance gain by stacking shallow-architectures with proper designed architecture and training algorithm. Popular stacked shallow-architecture models include, deep support vector machine (SVM) network [30, 43], deep conditional random field network [44], deep sparse coding network [93], so on and so forth. Learning with deep-architecture is now a hot topic in many fields of computer science research.

Deep learning now has made significant impacts on a wide range of scopes, including key aspects of machine learning and artificial intelligence. Empirical studies of many deep learning algorithms have demonstrated its success in diverse applications for traditional AI and ML practices, including computer vision [144, 46, 88, 164, 166], speech and phonetic recognition [89, 34, 42, 143] and signal processing [118, 36, 63], object recognition [64, 16, 31, 88], information retrieval [140, 65], natural language processing [163, 177, 111], multi-task and multi-modal learning [155, 121], robotics [145, 95], and many other domains. Pervasive successes of deep learning algorithms have lead to deep speculations and wide discussions in AI and ML communities. Theoretical researcher have identified many key characteristics by which deep architecture and related algorithms can

maximize their performances for distinct tasks [13, 11], such as deep-architecture, distributed representation, and making use of unlabeled data. Many of these characteristics in deep learning were found to coincide with the architecture of the biological neural network, especially the visual and auditory cortexes of human brain. Comparably, deep learning methods have demonstrated exceptional performance in processing low-level sensory data, such as image and speech data.

1.2 Modeling the high-level cognitive ability: Knowledge Engineering, Ontology, and Semantic Data Mining

Although the deep-architectures have demonstrated exceptional performances in processing low-level sensory data, few evidences have shown the association of deep learning with high-level cognitive abilities of human being, such as reasoning, making inferences, comprehending, and interpreting human knowledge. Previous AI researchers have devoted many efforts to addressing the such abilities through a subfield of AI, the knowledge engineering (KE) [137]. KE is a research field that is dedicated to developing techniques to build and reuse formal knowledge in a systematic way. In the past few decades, the proliferation of knowledge engineering (KE) has remarkably enriched the family of formal knowledge representation. Ontology is one of the successful knowledge engineering advances, which is the explicit specification of a conceptualization [58, 156]. Normally, an ontology is developed to specify a particular domain (e.g., genetics). Such an ontology, often known as a domain ontology, formally specifies the concepts and relationships in that domain. The encoded formal semantics in ontologies are primarily used for effectively sharing and reusing of knowledge and data. Prominent examples of domain ontologies include the Gene Ontology (GO [170]), Unified Medical Language System (UMLS [97]), and more than 300 ontologies in the National Center for Biomedical Ontology (NCBO [2]). The ontologies that formally represent domain knowledge, including structured collection of prior information, inference rules, knowledge-enriched datasets, etc., could thus develop frameworks for systematic incorporation of domain knowledge in an intelligent data mining environment.

The formal knowledge representation facilitates the knowledge reusing and sharing in a machine processable way. Advances in semantic data mining have also attested that formal knowledge representation could well assist in the data mining and machine learning process. *Semantic Data Mining* refers to data mining tasks that systematically incorporate domain knowledge, especially formal semantics, into the process. The effectiveness of domain knowledge in data mining has been attested in past research efforts, in both empirical and theoretical studies. Fayyad et al. [51] claimed that formally encoded domain knowledge can play an important role in all stages of data mining including, data transformation, feature reduction, algorithm selection, post-processing, model interpretation, and so forth. Russell and Norvig [137] believed that an intelligent agent (e.g., a data mining system) must have the ability to obtain background knowledge and should learn knowledge more effectively with the background knowledge.

The formal knowledge representations play such a role in semantic data mining that the encoded rich semantics could fundamentally promote the performance of data mining from many perspectives. The formal knowledge can bridge the semantic gap between the data, applications, data mining algorithms, and data mining results. It has have been shown that formal knowledge can reduce semantic gap through semantic aware preprocessing [83, 126, 165], semantic data annotation [48, 85]. Formal knowledge can also provide data mining algorithms with a priori knowledge which either guides the mining process or reduces/constrains the search space. It could influence

the semantic similarities of entities in the search space, in the form of graph or hypergraph; [98] or it could incorporate ontology as consistency constraints into multiple related classification tasks [9] and information extractors in information extraction tasks [26]. It could prune the data mining results with consistency checking, presented post-processing of the association rule mining [104, 105]. The formal knowledge can provide a formal way for representing the data mining flow, from data postprocessing of mining results. In ontology-based information extraction (OBIE) [119, 179], the extracted information is a set of annotated terms from the document, with the relations defined in the ontology. It is therefore straight-forward to represent the extracted information with ontology.

Empirical studies of semantic data mining research have attested the positive influence of domain knowledge on data mining. The data preprocessing can benefit from domain knowledge that it can help filter out the redundant or inconsistent data [83, 126]. During the searching and pattern generating processes, domain knowledge can work as a set of prior knowledge of constraints to help reduce search space and guide the search path [9, 10]. Further more, the discovered patterns can be cleaned out [104, 103] or made more visible by encoding them in the formal structure of knowledge engineering [179].

1.3 Bridging the semantic gap: Semantic Deep Learning

Based on the success of both deep learning and semantic data mining, we have more reasons to hypothesize that the formal knowledge representations have the potential to assist in the deep learning process as well. The study of semantic data mining has attested the positive influences of formal knowledge on data mining and machine learning: that they often present results with better precision, recall, consistency, and richer semantics. However, in most semantic data mining research, the potential of the rich semantics encoded in formal knowledge representations was usually not fully explored. In knowledge engineering, domain knowledge is usually encoded in a highly formal and abstract way, for example, formal logic, that in many scenarios it is often impractical to apply abstract semantic directly on raw data. Researchers commonly realize that there is often a large semantic gap between the raw data and formal knowledge representation. For such reasons, many semantic data mining practices tend to transform the formal knowledge representation into a form with reduced semantics, for example, a graph of connected entities, to bridge the gap between formal semantics and raw data. Such transformation will result in a loss of information and many useful aspects of formal knowledge representation. The reasoning abilities of formal knowledge, such as consistency checking and inferencing, were mostly applied on the data pre-processing, post-processing of the data, or data mining results rather than the key stages of the data mining process, including model design and training.

Fortunately, previous studies of deep learning have identified one key characteristic of using deep-architecture in machine learning and data mining, that is the data representations learned often corresponds to more abstract human cognitive concepts. Due to the large depth of the deep learning architecture, features and data representations can be learned in increasing levels of abstractions that higher level of features and concepts were often attested to have encoded closer semantics with human cognitive concepts [70, 94, 84], such as image scenes [46], sentiments [164, 152, 151], semantics of speech and natural language [35, 36]. Based on such fact, it is nature to speculate that deep learning can be a better formalism to incorporate the formal knowledge into the machine learning process through the such reduced semantic gap [13].

On the other hand, even though the deep-architectures have demonstrated the potential of reduc-

ing the semantic gap between data and formal knowledge representation, evidences are still scarce that current deep learning can fully explored such limit of data mining without the involvement of priori knowledge. For example, no deep learning technique have made use of the taxonomy constraints between labels in classification and recognition tasks, or potential semantic relations in feature space of object recognition tasks. Further more, no deep learning model has ever addressed the problem of modeling the high level cognitive intelligent abilities, such as reasoning, inferencing or validation, all of which have been well studied in the many year of the research of knowledge engineering.

The advances in both deep learning and semantic data mining have gave raise the hope to compensate their deficiencies for each other. With the abstract representation build by deep-architecture, it is reasonable to expect that formal knowledge representation has better potential to be applied on the data representation learned by deep-architecture in semantic rich way. With the assistant of formal knowledge, it is promising to expect the deep learning process to obtain the ability to model not only the representation of data, but also high level cognitive abilities of human being. We hypothesize that the formal knowledge representation could well assist in many aspects of deep learning process in a similar way as in semantic data mining, including deep-architecture designing, parameter tunning, representation explaining and result interpretation. We formally define the term, *semantic deep learning*, as the deep learning technique with the assistant of formal knowledge representation.

In this report, we exploit the possibility of semantic deep learning with the most popular form of formal knowledge, ontology. We start by making a brief introduction to the deep learning techniques in section 2 and some popular deep learning architectures and variants in section 3; in section 4 we describe few key characteristics of deep learning techniques, especially the ones that have the potential to assist in the semantic deep learning process; in section 5, we introduce the current advances in semantic data mining, specifically the ontology based approaches; in section 6 we present the major applications of semantic data mining and in section 7 we summarize the common ways that ontologies assist in semantic data mining, i.e. the roles that ontologies usually play in the semantic data mining process; finally in section 8 we present the thoughts and intuitions we currently have of the ontology based semantic deep learning.

2 Introduction to Deep Learning

Deep learning [66, 11, 64] refers to a set of machine learning algorithms that can learn the data representation and feature extraction with many layers of non-linear transformations. As shown in figure 1, a typical deep learning architecture, *deep-architecture*, resembles an artificial neural network, yet has many more layers of non-linear processing units. Deep learning is often called *representation learning* when it is necessary to highlight its importance in automatic feature and representation learning from data. A hierarchical, distributed, efficient data representation can be learned with deep-architecture in which higher level features and representations are defined in terms of the ones from lower layers. Features and data representations are learned in increasing levels of abstractions that higher layers are often found to have encoded representations with closer semantics with human cognitive concepts.

Deep learning is also often called *deep structure learning* in order to emphasize the distributed natural of its learned representation. The higher level features and data representations are constructed from a distributed subset of lower level components. Such distributed data representation

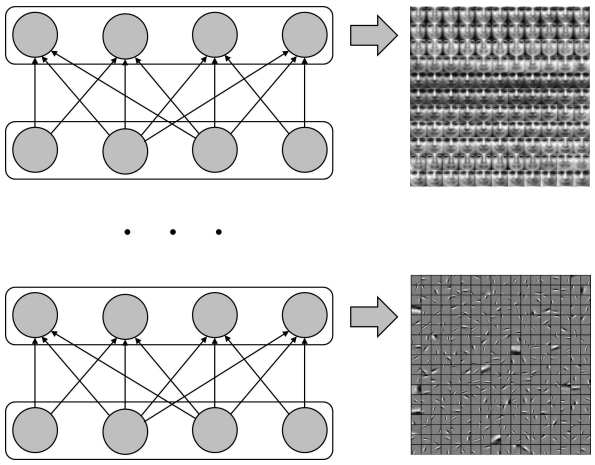


Figure 1: A deep-architecture usually consists of many layers of non-linear transformations. Features and data representations in a deep-architecture are often learned in increasing level of abstraction. Processing units in low level layers often learn low level features and data representations, e.g. edges of objects, while high level layers can learn more abstract representations from lower layer processing units, e.g. human face.

promotes the reuse of low-level features so that representations are learned in a succinct and efficient way. Bengio et al. claimed that the distributed representations in deep learning are often exponentially more efficient than many machine learning models [17]. Such efficiency potentially allows the deep learning techniques to get around with the pervasive curse of the dimensionality problem [15] in many previous machine learning techniques, especially for the ones with local smooth priori.

Deep learning is now a thriving field with many practical applications and active research topics. Empirical studies of many deep learning algorithms have demonstrated its success in diverse range of traditional AI and ML practices, including computer vision [144, 46, 88, 164, 166], speech and phonetic recognition [89, 34, 42, 143] and signal processing [118, 36, 63], object recognition [64, 16, 31, 88], information retrieval [140, 65], natural language processing [163, 177, 111], multi-task and multi-modal learning [155, 121], robotics [95], and many others. Excellent surveys on recent deep learning research progresses can be found in [12, 13, 11, 17, 96, 14, 141].

In this section, we briefly introduce the major variants of deep architectures proposed in recent years. Due to space limitation, we focus on summarizations of the major deep learning variants that have potential to assist the goal of our research, semantic deep learning. We first introduce one important family of deep-architecture, the Restricted Boltzmann Machine (RBM) based deep-architecture in section 2.1; we present the deep convolutional neural networks (CNN), which have wide applications in computer vision and speech, in section 2.3; we introduce the deep autoencoder in section 2.4; and finally, we briefly summarize other popular deep learning architecture variants in section 2.5.

2.1 RBM based deep-architecture

RBM based deep learning was first introduced by G. E. Hinton et al. in 2006 [66, 64] in his remarkable work of RBM based deep autoencoder. RBM is a probabilistic graphic model that can be

interpreted as both a generative model and a stochastic neural network. This special property renders RMB based deep-architecture the ability to make use of both unlabeled data and labeled data in the training process. As a generative model, RBM learns a representation of data distribution instead of classifying the data as in the traditional neural networks. Training with unlabeled data proceed through a greedy level-wise unsupervised pre-training process in which weight optimization of parameters has a time complexity linear to the size of the network. The unsupervised pre-training initializes the parameters to a point closer to the global optimum. As a stochastic neural network, the model is further fine-tuned with BP after pre-training with labeled data. Such optimization process alleviates the pervasive local optimal problem occurred in the traditional neural networks. While the model is still possible to fall into a local optimal, it has a much better probability to stay closer with the global optimum with the pre-training and fine-tunning. Based on the success of RBM based deep autoencoder, many other RBM based deep-architectures were proposed in the following years. In this section, we will give a brief introduction to important family of the RMB based deep learning method. Note that as RBM can be used as building block for many other categories of deep-architecture, such as deep autoencoder and deep sparse coding, many of the important RBM based deep learning advances are also detailed in next few sections.

2.2 Restricted Boltzman Machine

Restricted Boltzmann machine (RBM) is a probabilistic graphic model that serves as building blocks for many deep learning models. RBM is a simplified version of the Boltzmann Machine (BM) with a bipartite connection restriction.

2.2.1 Boltzmann Machine

Boltzmann machine (BM) is a bidirectionally connected network with binary stochastic processing units. A global energy function $\boldsymbol{E}$, which indicates the degree of harmony of the network, is usually defined on the state of the network,

$$E = -\sum_{i<j} w_{ij} s_i s_j + \sum_i \theta_i s_i, \tag{1}$$

in which w_{ij} is weight of the connection between unit i and j, $s_i \in \{0, 1\}$ is the state of unit i, θ_i is a bias which indicate the threshold of activation for unit i. Boltzman machine consists of two types of units, the visible units v and hidden units h. The v and h correspond to input data and hidden variation factors respectively. A probabilistic distribution function is defined on each network state by the engergy function,

$$p(v, h) = \frac{1}{Z} e^{-E(v,h)}, \tag{2}$$

where $Z = \sum_{v,h} e^{-E(v,h)}$ is the partition function to normalize the distribution. To train a BM, we usually compute the gradient of the log-likelihood given a single training example v,

$$\frac{\partial lnL(\theta \mid v)}{\partial \theta} = -\sum_h p(h \mid v) \frac{\partial E(v, h)}{\partial \theta} + \sum_{v,h} p(v, h) \frac{\partial E(v, h)}{\partial \theta}. \tag{3}$$

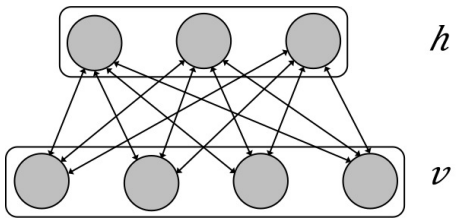


Figure 2: Restricted Boltzmann Machine

Note that this expression leads to a computation that runs over all values of the variables which makes the computational complexity intractable. Gibbs sampling based techniques are usually used to approximate such gradient.

2.2.2 Restricted Boltzmann Machine as probabilistic graphic model

As shown in figure 2, RBM is a simplified BM with restriction that variables in the same layer share no connections between each other. The energy function of a RBM can be rewritten from equation 1

$$E(v,h) = -\sum_{i \in visible} a_i v_i - \sum_{i \in hidden} b_i h_i - \sum_{i,\ j} v_i h_j w_{ij}. \tag{4}$$

Under such constraint, the conditional distribution $p(h|v)$ and $p(v|h)$ factorize nicely:

$$p(h|v) = \prod_{i=1}^{n} p(v \mid h_i) \tag{5}$$

and

$$p(v|h) = \prod_{j=1}^{m} p(h \mid v_i) \tag{6}$$

Training of a RBM is can be done by gradient ascent from training data set $S = \{x_1, ..., x_l\}$. Using equation 3, the ascent from training data set can be written as,

$$\begin{aligned}
\frac{\partial log p(S)}{\partial w_{ij}} &= \sum_{x \in S} \frac{\partial log p(x)}{\partial w_{ij}} \\
&= \sum_{x \in S} \left[-\sum_{h} p(h \mid v) \frac{\partial E(v,h)}{\partial \theta} + \sum_{v,h} p(v,h) \frac{\partial E(v,h)}{\partial \theta} \right] \\
&= \sum_{x \in S} \left[\mathbb{E}_{p(h|v)}[v_i h_j] - \mathbb{E}_{p(h,v)}[v_i h_j] \right] \\
&= \sum_{x \in S} < v_i h_j >_{p(h|v)} - < v_i h_j >_{p(h,v)},
\end{aligned} \tag{7}$$

where the first term is the expectation of $\frac{\partial E(v,g)}{\partial w_{ij}}$ when v is set to the input data, and h are sampled according to $p(h|x)$. The second term is the expectation of $\frac{\partial E(v,g)}{\partial w_{ij}}$ when u and g are sampled according to the joint distribution of $p(u, g)$.

2.2.3 Restricted Boltzmann Machine as stochastic neural network

Another important property of RBM is that it can be interpreted as a stochastic neural network. In the RBM, the conditional property of a single variable being one is

$$p(h_i = 1|v) = sigm(\sum_{j=1}^{m} w_{ij}v_j + c_i) \tag{8}$$

and

$$p(h_i = 1|h) = sigm(\sum_{i=1}^{n} w_{ij}h_j + b_i), \tag{9}$$

where sigm(x) = 1 / (1 + exp(-x)) is the logistic function. A RBM can thus be interpreted as a standard feed-forward neural network with one layer nonlinear processing units. The observation is mapped to the expected value of the hidden neuron given the observation.

2.2.4 RBM based deep-architecture

Following the sucess of RBM in non-linear dimensional reduction, many other variants of RBM based deep learning techniques were proposed. Gaussian-Bernoulli RBM (GRBM) [66, 118], Gaussian Gated Boltzmann Machine (GBM) [106, 60] and mean covariance RBM (mcRBM) [34, 131, 87] were often used as the input layer for continuous data. Deep Conditional RBM (CRBM) were proposed to model time-series data, especially the data of human motion [117, 167]. Deep sparse DBN [22] follows the biological inspiration of sparse coding in biological neural networks that it further restrict the RBMs with sparse activation constraint. Convolutional RBM [122] processes the image data with translation invariant feature learning. Recurrent temporal RBM [161, 116] models the sequential data in a similar way as recurrent neural network. RBMs were often used as building blocks for deep autoencoder [174], which will be introduced in detail in section 2.4.

2.3 Deep Convolutional Neural Networks

Convolutional neural network (CNN, ConvNet) is an important family of feed-forward ANN where each neuron is responsible for local and overlapping receptive fields from lower layer inputs. The notion of CNN can be traced back to the early 80's [91] when object recognition tasks were inspired from biological visual cortex organization. The cells in visual cortex were found to be arranged in a way that they are only sensitive to small sub-regions of the visual field [72], called *local receptive field*. Local receptive fields are tiled to cover the whole visual fields. Typical design of neural network enables full connections between neurons in adjacent layers. Such design is, however, inefficient when modeling data with topological structures, such as images and speech (with time-frequency representation). A general assumption of image and speech data is their 2D structure and locality of dependencies, i.e. inputs (pixels or speech spectrum) that are spatially or temporally near by are strongly correlated. CNNs explore such assumption by restricting the lower layer input to a local area. They have, therefore, much fewer connections and parameters so that they are easier to train while with theoretically slightly worse performance.

As shown in figure 3, the convolutional layer in CNN learns a set of N feature maps $F = \{f_1, ..., f_N\}$ through convolution transformation $C = \{c_1, ..., c_N\}$.

$$f_k = c_k \otimes x, \tag{10}$$

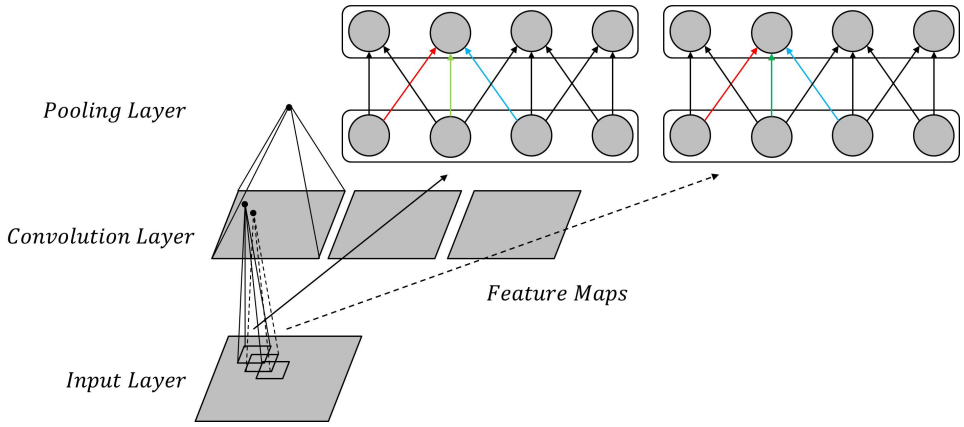


Figure 3: Two layers in a convolutional neural network. Each neuron in higher level layer, a local receptive field, only connects to a set of local neurons.

in which $\otimes$ is the convolution operator. Typical CNN design often contains other layers, including *weight sharing*, *pooling*, and *drop out*, to address the problem of translation invariants and overfitting. Shift invariance is obtained by forcing the replication of weight configurations across space. As shown in figure 3, the weights of local filter enforce to be the same across all local receptive fields in a feature map. Then pooling layer summarizes a small region R of a local receptive fields in a feature map in the way that it output the maximum or average these receptive fields.

$$p_R = \max_{i \in R} f_i. \tag{11}$$

Dropout [67, 154] is usually used to reduce overfitting by randomly setting the output of few neurons to zero. It was proved to successfully reduce complex co-adaptations of neurons, since a neuron cannot rely on the presence of particular other neurons [56]. Typical CNNs design usually contains few fully connected neural network work layers before they output to the final classifier.

2.4 Deep Autoencoder

Autoencoder (AE) is a special type of DNN that learns to reconstruct the data with minimal error, noise or distortion. Figure 4 shows the typical structure of an autoencoder which usually consists of two parts, an encoder $f(\cdot)$ and a decoder $g(\cdot)$. An autoencoder learns to reconstruct the input data x by minimizing a loss function $L(r(x), x)$ between input and reconstruction, where $r(x) = g(f(x))$ is the learned reconstruction function. An autoencoder transforms the input data to a desired representation (also known as the code or the feature vector) at the output layer of the encoder $c = f(x)$, the code layer. The decoder, on the other hand, is trained to reconstruct the input from the representation c.

Autoencoder was first introduced in the 1980s by Rumelhart et al. [134] as a dimensionality reduction technique via unsupervised learning of error propagation. The dimensionality reduction was enforced though a code layer with a dimension d_c smaller than the input dimension d_x with minimized reconstruction error. Linear auto-encoders can be solved analytically while it were proved to learn the same subspace as PCA. Like neural networks, non-linear autoencoders are often trained

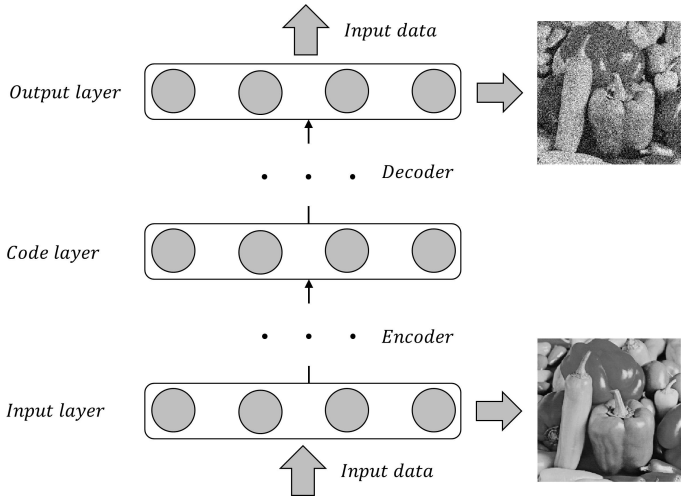


Figure 4: A example of deep autoencoder. Deep autoencoder transform the input data into a representation in the code layer and recovery the data from the code layer with minimum error.

via variants of back-propagations, such as gradient descent and contrastive descent. Training of non-linear autoencoders thus suffer from all the deficiencies of the training of neural networks. Moreover for deep autoencoders with more layers of non-linear transformations, the back-propagated error becomes minuscule when it reaches the input layer. Training of deep autoencoder are thus often ineffective with poor solutions. These problems were alleviated when RBM based deep autoencoder for non-linear dimension reduction was proposed by G.E. Hinton in 2006 [66]. The deep autoencoder is composed by stacking RBMs. Level-wised greedy pre-training with unlabeled data reduces the training problems that often occur in autoencoders. A significant performance gain were observed when it was compared to PCA and logistic PCA.

Deep autoencoders nowadays play an important role in unsupervised learning and transfer learning. Besides the its impressive application in dimension reduction, the learned representation were used to boost the performance of subsequent machine learning algortihms. It is then become necessary that the code layer has a dimension larger than the input dimension inorder to learn a richer set of variations from the input data.It is then become necessary to add regularizations to the autoencoder to prevent a identical mapping. Such autoencoders are called regularized autoencoders. Primary variants of regularized autoencoder include denoising autoencoder, contrastive autoencoder, and sparse autoencoder. Denoising autoencoder [174, 175] tries to improve the generalization by learning a robust reconstruction from a noisy input x. Contrastive autoencoder [133] improves the sensitivity to the input data. It learns to penalize the sensitivity through the Frobenius norm of the Jacobian $J_f(x)$ of the non-linear mapping to encourages the encoder to be contractive in the neighborhood of the training data. Other variants of deep autoencoder includes autoencoder with regularizations applied in order to avoid learning a identical mapping, such as weight decay [17] and sparse coding [93] [22]. Applications of deep autoencoder spreads through many fields natural language processing [149], speech processing [151] [42].

2.5 Other deep-architectures

There are many other variants of deep-architectures which target different applications with various optimization goals and constraints. Stacking shallow-architectures is one of the popular ways to obtain a deep-architecture. Popular stacked shallow-architecture models include, deep support vector machine (SVM) network [30, 43], deep conditional random field network [44], deep sparse coding network [127, 93], so on and so forth. Deep recurrent neural network explores the limitation of the context length that information can cycle in the neural network for arbitrary long time [112]. Sparse coding [124, 125] also called dictionary learning, learns an over-complete set of basis with sparse activations. The representation of input data is a linear superposition of the basis functions $\phi_i(x, y)$ and the parameters a_i under sparse constraints. The input data is represented by

$$I(x, y) = \sum_i a_i \phi_i(x, y). \tag{12}$$

The learning of sparse code usually involves an optimization of a loss function with two components, information loss and sparse constraint:

$$E = -[Information\ loss] - \lambda[sparse\ constraint]. \tag{13}$$

3 Application of Deep Learning

3.1 Natural Language Processing

Natural language processing (NLP) [102] addresses the problem of building a machine processable formal representation of human language for further applications, such as information extraction, machine translation, search, and summarization. Before recent prosper of deep learning in NLP, NLP systems and techniques often treat words as atomic units - there is no notion of similarity between words, as these are represented as indices in a vocabulary. This choice has several good reasons - simplicity, robustness and the observation that simple models trained on huge amounts of data outperform complex systems trained on less data.

”” Many current NLP systems and techniques treat words as atomic units - there is no notion of similarity between words, as these are represented as indices in a vocabulary. This choice has several good reasons - simplicity, robustness and the observation that simple models trained on huge amounts of data outperform complex systems trained on less data. An example is the popular N-gram model used for statistical language modeling - today, it is possible to train N-grams on virtually all available data (trillions of words [3]).

Large language models in machine translation. ””

In other practices of NLP, similar as the prevalence of feature engineering practices in computer vision and speech, the overwhelming majority of the state of the art NLP systems relies on applying task specific features engineering on linear statistical models, such as n-gram language model. The performance improvements on these benchmarks were often a result of applying knowledge of linguistic, which is often very task dependent. Applications of deep learning technique on NLP were often proved to have avoided task specific engineering. The deep-architecture was used as a single learning system to discover adequate internal representations for multiple natural language processing tasks [33], including part-of-speech tagging (POS), chunking (CHUNK), named entity recognition (NER) and semantic role labeling (SRL).

Deep-architectures often benefits natural language models from its distributed natural of the learned representation. Many deep learning based NLP models adopts deep-architectures to learn continuous vector space model [173] of the word representation. The words and phrases were often randomly mapped to a vector representation through one-hot vector representation as the input of the deep-architecture. The deep learning algorithms then train the deep-architecture to map the word, predicate, or sentence space into high dimensional continuous vector space. Such deep learning models largely reduct the curse of dimensionality problem embedded in the rich semantic and syntax relations in the NLP corpus.

Successful NLP applications of the deep vector space model includes, parsing [150], sentiment analysis [151, 100, 55], word similarity [110, 149, 32]. The representation learned in NLP were often attested to better capture the semantics of the natural language. Each word in NLP is represented as a real-valued feature vector such that the inner product well addresses their similarity. One recent finding in NLP attested that the mapping learnt by deep architecture can capture both syntactic and semantic regularities in language, and also relationship by a relation-specific vector offset [113]. For example, they observe the plural syntactic regularity $x_{apples} - x_{apple} = x_{cars} - x_{car}$, and the male/female semantic regularity $x_{king} - x_{man} + x_{woman} = x_{queen}$. The learned deep-architecture were also attested to be able to assist in multiple NLP task with one architecture, including part-of-speech tags, named entity tags, semantic roles, semantic similarities [32]. As the deep-architecture can learn a vector representation for multiple data sources, the continuous vector space model were often combined with other data sources, for example, image to assist in scene parsing [150] or interpret sentiment and semantic predictions of natural images [172].

Deep neural tensor network (DNTN) has been another deep neural network model for NLP tasks. Socher et. al. [152] use DNTN model to solve a sentiment classification prediction tasks over the movie review context. The DNTN learns a vector representation both the single word and the rhetorical relationships between adjacent phrases and sentences. The DNTN captures both the semantics and syntax structure from the context. It is therefore successful the negative transition in natural language which is hard to manage by any previous methods. As a related topic with semantic deep learning, DNTNs are used to learn the vector representation of knowledge bases and knowledge graphs as well. Boardes et.al [21] use DNTN from both WordNet [114] and Freebase [20]. Chen et. al. [28] uses DNTN to learn the representation of WordNet [114] to predict new relationship entries that can be added to the WordNet database. Socher et. al. [148] use DNTN to model the reasoning ability of knowledge base to predict and classify the unseen relationship in knowledge base.

3.2 Speech Recognition

Automatic speech recognition refers to the task of translating speech signals into text. It is still a challenging task due to high viability of the speech signal. Speech data can be affected by many task irrelevant factors such as accents, intonations, pronunciations, speaking style, speaking rate, and even recording devices and background environment. To disentangle the critical factor, the semantics from the speech signal is the primary task for most speech recognition systems. Speech recognition has been dominated by the Gaussian Mixture Hidden Markov Model (GMM-HMM) [61, 40, 39] before the prevalence of deep learning based method. The dominance of GMM-HMM in speech recognition mostly due to the piecewise stationary signal or a short-time stationary signal characteristic of the speech signal. Speech signals can usually be approximated as a stationary

process in a short time-scale. Therefore, the random process model of speech can be natural thought of as a Markov model for many stochastic purposes. In each HMM state the spectral representation of the sound wave is models by a mixture Gaussian model. HMMs-based speech recognition models are simple and computationally feasible to use. However, one important deficiency of Gaussian mixture models should still attribute to assumption of the statistical properties of the speech data. HMMs are usually statistically inefficient for modeling data with non-linear manifold in the search space.

DNN started to influent the speech recognition since 2010 and soon became the mainstream technology of speech recognition in recent years. Reviews of deep learning on speech recognition can be found on [38, 63]. Early work of deep learning on speech recognition present a hybrid view of DNN architecture and traditional speech recognition methods. Specifically, deep neural network hidden markov model (DNN-HMM) architectures were attested to be a success in which DNN replaces the role of GMM to estimate the observation probability [118, 35, 36, 184]. Except for the state-of-the-art performance gain in many speech recognition benchmarks, DNN-HMM often demonstrate the ability to handle tasks with large vocabulary [35, 36] which promotes the practical applications in speech industry including Microsoft [41, 29] IBM [139, 138], and Google [62, 75]. Comparing with previous popular Mel-frequency cepstral coefficients (MFCC) features [183], deep speech recognition methods often applied raw spectral [41] and temporal data [147] while obtaining significantly lower speech recognition error.

Hiton et. al. [63] proposed a deep neural network method which use coefficients in previous frames to predict the posterior probability of the HMM. Graves et. al. [57] uses deep recurrent neural network (RNN) with long short-term architecture to obtain the flexibility to model long ranges context.

3.3 Computer Vision

Over the past decades, tremendous progress has been made in computer vision by the research of deep learning. Distinguished advances include advances in hand writing recognition [31], object detection localization [144, 46], image classification [88], face recognition [164], 3D video [166] [76]. Early works of deep learning on computer vision often adopts generative feature learning, in which RBM is used to train with the unsupervised data. Hinton et al. [64, 66] first explored the power of RBM in DBN and deep autoencoder on hand writing recognition [64] and dimension reduction tasks [66]. Variations of DBN were also applied on 3D object recognition [120] and human motion modeling [168]. The largest so far DBN for image processing were built by Le et al. in 2012 with a nine-layer sparse autoencoder with one billion connections, trained on the dataset with 10 million images [90]. They explored the possibility of detecting objects, such as human faces, with only unlabeled data and large scale distributed DBN.

In following years, the advent of CNN divert the trends in image processing with deep learning while pre-training were often attested to be less important when large amount of labeled data is available. CNN based supervised feature learning method starts to be prevalent in many computer vision task since 2012. Large scale benchmarks were setup for research of computer vision tasks includes PASCAL VOC challenge [49] and ImageNet Large Scale Visual Recognition (ILSVRC) [136]. These challenges have been running annually since 2005 and 2010. CNN achieved the state-of-the-art in most of the recent computer vision challenge benchmarks. Specifically, deep-learning based image recognition methods won ILSVRC classification contest from 2012 to 2014 [88, 185,

162], and the ILSVRC object localization and detection contest in 2013 [144] and 2014 [77].

Due to the complexity of the human face, face recognition tasks has been explored by more as a standalone tasks by multiple deep learning architectures, including Deep Convolutional Neural Network [142, 159, 50, 189, 160], Deep Convolutional Belief Network [70], Deep Neural Network [188, 189], CNN-RBM hybrid model [158], Deep Independent Subspace Analysis Network [24]. Representative success of face recognition application in deep learning includes facial point hallmark detection [157, 187], facial recognition and identity detection [159, 70, 164], face alignment and view reconstruction [188, 189]. Current state-of-art performance of face recognition on LFW [71], and YouTube Faces DB [180] was achieved by FaceNet [142] with an identification accuracy of 99.63% and 95.12%. Deep learning also demonstrate the ability to building high-level features that could identify human face without labeled data, i.e., using large scale unsupervised learning Quoc V. Le [90]., and traffic sign identification [146].

4 Characteristics of deep learning

Deep learning has demonstrated exceptional performance in many fundamental AI and ML practices in the past decade. The pervasive successes of deep learning techniques has intrigued the AI and ML communities to ponder over the theoretical groundings of deep learning. Previous research have identified several key common characteristics for deep-learning techniques, including *learning data representation*, *deep architecture and abstraction*, *distributed representation*, *disentangling factors of variation*, *utilizing unlabeled data*, *unsupervised pre-training and transfer learning* etc [13, 11, 17]. In this section, we summarize several key advantages of deep learning that favors our proposal of data driven approach of formal semantics, including:

- Representation learning: learning feature representations from data
- Deep architecture: composing representation of complex high order functions by many weakly nonlinear transformations.
- Distributed representation: the appeal of hierarchical distributed representations for more efficient data representations.

More importantly, we will see how these feature will help better integrate the formal semantics into the data mining process, i.e., the semantic data mining.

4.1 Learning Feature Representation from Data

Bengio et al. argued that to make AI fundamentally understand the world around us, it must be able to learn to process and identify the explanatory factor from the low-level sensory data [13]. For many complex machine learning tasks, such as computer vision and speech recognition, the data are well known for their the variability and richness. For those tasks, generic machine learning algorithms were often found to be very difficult to extract discriminative information while handle the translation and trasformation of data. For such reason, much of the efforts in image and speech processing went into the hand-crafting of features and data representations. Such process is often called *feature engineering* which aims at taking advantage of priori knowledge and human ingenuity to promote the machine learning tasks.

Hand-crafted features were known to have made profound progresses in many fields, such as SIFT [99] and HOG [37] in computer vision, LBP for face detection [5] [123] and MFCC [183] for speech recognition. However, they were often proved to have many deficiencies, including inaccuracy, intensive, domain and task-dependent and labor intensive. The design of features is often more subjective trial and error process which would rely heavily on the experiences and knowledge of the designer. The features are often either over-specified or incomplete depending rather than a accurate reflex of the underlying factors of the data. The feature design process is strongly task dependent which has to be redone for every new task [33]. Most hand-crafted features were proven to be only able to capture low-level information from data, such as edges in images or senones in speech, while capturing high-level representation such as object parts is often more difficult.

Previous research has long contented that better pattern recognition systems can be built by relying more on automatic learning, and less on human ingenuity [92]. Deep learning is such a fundamental method which is known to take advantage of the priori in the data to compensate this labor-intensive. In order to highlight the role of automatic learning representation from data, in many deep learning literatures, the deep learning technique is often called representation learning. We argue that the learned representation can assist the application of formal semantics on the data mining and machine learning process in a fundamental way. in semantic data mining higher level features with more abstract concepts are constructed using combination of lower level features. In practice, such representation learning through deep-architecture were often used to obtain higher level representations with correpondent to higher level and more abstract representations. Such representation closer the semantic gaps.

4.2 Power of representation: Deep Architecture

One of the long term goal of AI and ML research is to develop methods that are capable of highly complex intelligent tasks, such as perception, reasoning, and intelligent control. To achieve such goals, the machine learning community must endeavor to discover algorithms that are capable of expressing complex behaviors that require highly varying mathematical functions, i.e. mathematical functions that are highly non-linear in terms of raw sensory inputs [11]. While many previous algorithms have endeavored to do so, theoretical and empirical evidence have suggested that *shallow-architectures* are fundamentally limited in modeling high-dimensional complex functions. Although theoretical research show that some shallow architectures can represent functions with arbitrary precision, the efficiency of learning such representation would be usually too low in terms of number of computational elements and examples [17]. Bengio and leCun [17] have argued that it usually requires exponentially more parameters and components for a shallow-architecture to represent function with the same precision as the architecture with more layers, i.e. the deep-architecture. More parameter usually not only means more training time, but also fundamentally more data is required to achieve the learning accuracy at the same level. The fact that the deep learning community established a distinction between *shallow-architecture* and *deep-architecture* highlights the recent discovery of the importance of depths with regarding to model the complex functions and profound limitation of the shallow architectures.

4.3 Distributed Representation: explore the representation efficiency

Deep learning techniques are capable of learning a distributed data representation. High level features in a deep-architecture are composed by distributed sets of components. Each low level feature

is often reused to represent multiple feature representation. The distributed representation which promotes the notion of feature reuse is at the heart of theoretical advantages of deep learning. The idea behind the distributed representation is that the feature reuse can represent exponentially large number of concepts by composing many features. One example of the distributed representation is the binary representation of numbers, in which n independent bits can represent 2^n numbers, an exponential size of the size of representation. The distributed representation reuses local representations at diverse levels and representation schemas which could potentially make a much more succinct representation of the input data. Such representation is proved to be able to better get around the curse of dimensionality problem in many other machine learning algorithms.

The distributed representation benefits the learning process that it has the strength of modeling well the correlations and dependencies in the data. Such ability largely relax the restriction on input data that the input data of deep learning can often be raw data samples. In speech recognition, deep learning was proved to have the best performance on the time series data or speech spectrum. In natural language processing, the input data is often the random indexing of the vocabulary. The distributed representation learns to model the best representation of dependencies batten input data through the learned data representation.

5 Introduction to semantic Data Mining

While the deep learning models have demonstrated exceptional performance for the data representation and transformation tasks, they have shown little evidences of the ability to associate with the high level cognitive behaviors such as reasoning, understand and interpreting the knowledge. Such behaviors were addressed by previous AI researches from a *knowledge driven* perspective, i.e. using formal knowledge representation. The formal knowledge representation not only facilitated the knowlede sharing and resuing in a formal and effective way, but was also attested to be able to assist in the machine learning and data mining process, through the technique called semantic data mining.

Semantic Data Mining refers to data mining tasks that systematically incorporate domain knowledge, especially formal semantics, into the process. The effectiveness of domain knowledge in data mining has been attested in past research efforts. Previous theoretical and empirical semantic data mining research has attested the positive influence of domain knowledge on data mining. For example, the preprocessing can benefit from domain knowledge that can help filter out the redundant or inconsistent data [83, 126]. During the searching and pattern generating process, domain knowledge can work as a set of prior knowledge of constraints to help reduce search space and guide the search path [9, 10]. Further more, the discovered patterns can be cleaned out [104, 103] or made more visible by encoding them in the formal structure of knowledge engineering [178]. As a formal specification of domain concepts and relationships, ontology can assist in the data mining process in various perspectives. It is reasonable to expect a performance gain in ontology-based approaches compared with the data mining approaches without using ontologies or other forms of domain knowledge. Many semantic data mining research efforts have attested such improvements. With well designed algorithms, previous research either reports performance improvement or accomplishments of data mining tasks that could not be achieved without using ontologies. In the following sections, we give a brief summarization of the performance improvement in ontology-based approaches and their applications.

6 Application of Semantic Data Mining

Empirical results from previous research have attested the potential of ontology to assist in various data mining tasks. In this section, we summarize semantic data mining algorithms designed in several important tasks, including association rule mining, classification, clustering, recommendation, information extraction, and link prediction.

6.1 Ontology-based Association Rule Mining

Association rule mining is a fundamental data mining task that finds the associations of frequent item sets. As the item found in association mining often corresponds to concepts in the ontology, it is very convenient to provide constraint or auxiliary information using the ontology relations. Ontology can provides pruning constraints and abstraction constraints for the association rule mining task [10]. The pruning constraints are used for filtering a set of non-interesting items while the abstraction constraints promote the generalization of item into more general concepts in the ontology. Ontology can also assist in the post-processing of the association rule mining results using an ontology for the consistency checking. Invalid or inconsistent association rules are pruned and filtered out with the help of ontology and an inference engine [104, 105]. In [98], Liu et al. use ontology as auxiliary information to discover latent associations in the data. They built the connections between ontology and data using a bipartite hypergraph model.

6.2 Ontology-based Classification

In semantic data mining, one typical use of ontology is to annotate the classification labels with the set of relations defined in the ontology. With the ontology annotated classification labels, the semantics encoded in the classification task was often proved to have the potential not only to influence the labeled data in the classification task but also to handle large number of unlabeled data [9]. Ontology can serve as consistency constraints into multiple related classification tasks. These tasks classify multiple categories in parallel. An ontology specifies the constraints between the multiple classification tasks.An unlabeled error rate is defined as the probability the classifier assigns a label for the unlabeled data that violates the ontology. This classification task produces the classification hypothesis with the classifiers that produce the least unlabeled error rate and thus most classification consistency. In other classification tasks, ontology often provides a similarity measure for terms and concepts in the data, for example documents. In [8] semantic graph of connected entities are constructed from the set of relations from DBpedica-based ontology. HITS algorithm [86] is used to identify the core entities in the semantic graph for the further identification of dynamic topics. The classification of documents is based on calculating the similarity of document's semantic graph to define ontological context (topics).

6.3 Ontology-based Clustering

Clustering [74] is a data mining task that grouping a set of objects in the same cluster which are similar to each other. Early work of ontology-based clustering includes using ontology in the text clustering task for the data preprocessing [68], enriching term vectors with ontological concepts [69], and promoting distance measure with ontology semantics [79].

In recent works, ontologies were often used to annotate the data in the text clustering with an enriched conceptual similarities [153], [153]. It also helps to re-weight the vectors in knowledge-based vector space for text clustering [78] and the terms in the medical documents [186]. Fodeh [53] used the onotology to prune the feature space in document clustering. He claimed that ontology can be used to greatly reduce the number of features needed. In gene clustering task, gene ontology (GO) assisted the similarity measure between genes with graph structure (GS) and information content (IC) based measures.

6.4 Ontology-based Information Extraction

Information extraction (IE) refers to the task of retrieving certain types of information from natural language text by processing them automatically. IE is closely related to text mining. Ontology-based information extraction (OBIE) is a subfield of information extraction, which uses formal ontologies to guide the extraction process [82, 179]. Because of this guidance in the extraction process, OBIE systems have mostly implemented following a supervised approach [178]. Although very few semi-supervised IE systems are considered as ontology-based [181, 182], they rely on instances of known relationships [4, 132]. Therefore those semi-supervised systems can also be considered as OBIE systems.

Early work of OBIE includes knowledge extraction from web documents [6] and data-rich unstructured documents [47]. Ontology can provide consistency checking for the extracted information in the IE system [81] and constraints and exclusions for different categories and relations [27]. As a way to promote the adoption of OBIE, Ontology-based Components for Information Extraction (OBCIE) [178] aims to encourage re-usability by modeling the components of the IE system as modular as possible using ontology. Gutierrez et al. [59] extended the OBCIE architecture by incorporating hybrid configurations (e.g., different implementations and different functionalities).

6.5 Ontology-based Recommendation System

Recommender systems or recommendation systems [3, 23] are the systems that dedicate to predict the preference or ratings that a user would give to an item. In a good recommendation system, heterogeneous information from multiple sources is usually required. Ontology can integrate the use of heterogeneous information and guide the recommendation preference.

Early work of ontology-based recommendation system uses ontology for user profiling [109], personalized search [128], and web browsing [108, 107]. In recent works, ontology helps to generate and recommend tags automatically for web resources [129]. The web documents are annotated and matched by terms in the ontology first. Then ontology-based reasoning is conducted to infer the new knowledge from the annotated terms. This inference is made by finding the common ancestor nodes for them and possibly all the nodes in the path between the matched nodes with ontological concepts. In other works, ontology is used to encode the long term and short term user preference information [80, 25]. The user preference ontology is constructed from the concepts of the general domain ontology together with the documents that the user visited. Ontology can also help to store concepts and relationships to the web items, for example, in a news feed recommendation system [73, 25]. A news ontology can be used to build a news personalization service to provide the concept framework for new contents and determine the semantic relations between terms and concepts.

6.6 Ontology-based Link Prediction

Link prediction for social networks becomes a very active research area in data mining due to the success of online social networks such as Twitter, Facebook, and Google+. As link prediction is often closely related with graph structures between social entities, the graph structure of entities and relations in ontologies plays an important role in link prediction. Aljandal et al. [7] presented a link prediction framework with ontology-enriched numerical graph features. The authors claimed that in previous social network research flat representation of interest taxonomies limited the improvement of link prediction. Ontology aggregated distance measure is proposed to encode the interest taxonomies in ontology into the distance measure to more accurately describe the shared user interests. In other works, ontology often helps to annotate the data with rich semantics [171]. The annotation links between the data and predicates in ontology form an annotation graph. Semantic information in the ontology is used in the sequential pattern mining algorithm to prune the search space and partially relieve the algorithm from support counting [101]. Amakrishnan [130] proposed a method to discover the informative connection subgraphs that relate two entities in the graph. They proposed heuristics for edge weighting that depend indirectly on the semantics of entity and property types in the ontology and on characteristics of the instance data.

7 Role of Ontologies in Semantic Data Mining

The question why domain knowledge is helpful in the data mining process has been long discussed in previous semantic data mining research. The perspective and mechanism of utilizing ontologies in semantic data mining often varies across different systems and applications. Of the many ways that ontologies assist in semantic data mining, we have identified three major ways that have the potential to assist in the goal of our research. In this section we introduce the three major roles ontology could play in semantic data mining, *bridging the semantic gap*, *provide priori knowledge*, and *provide a formal data mining flow representation*.

7.1 Bridging the semantic gap

The application of ontology in semantic data mining was often attested that it could bridge the semantic gap between the data, applications, data mining algorithms, and data mining results. Researchers claim that there exists a knowledge gap between the data, data mining algorithm, and mining results in all stages of data mining including preprocessing, algorithm execution, and result generation [45]. Previous research practices have identified many senarious where there exist semantic gaps in data preprocessing. Data preprocessing usually contains data cleaning, normalization, transformation, feature extraction and selection. Without considering formal semantics, data mining practices usually deploy ad-hoc or empirical methods to determine the quality of the data. For example, scarcity and nearest neighbor rules are the dominant method to determine the outliers and missing values. In the normalization and transformation step, it is important to determine the correlation between features and attributes of the data when performing data normalization. Strongly correlated attributes could be reduced into one combined attribute. In practice, semantic gaps are usually filled manually by domain experts. However, ontologies have been shown to be beneficial in many data preprocessing tasks [83, 126, 165].

We are easy to observe semantic gap between the data mining algorithm and data as well. Data

mining algorithms are usually generic methods that designed for data from different domains and scenarios. Data from a specific domain usually carry domain specific semantics. The generic data mining algorithms lack the ability to identify and make use of semantics across different domains and applications. Ontologies are useful to specify domain semantics and can reduce the semantic gap by annotating the data with rich semantics. Semantic annotation aims at assigning the basic element of information links to formal semantic descriptions [48, 85]. Semantic annotation is crucial in realizing semantic data mining by bringing formal semantics to data. The annotated data are very convenient for the later steps of semantic data mining because the data are promoted to the formal and structured format that connects ontological terms and relations.

Other research efforts have dedicated to bridge the semantic gap between data mining results and users. The data mining results can be represented by ontologies in the semantic rich format which help sharing and reuse. For example, information extraction (IE) is the task of automatically extracting structured information from text. The data/text mining results are sets of structured information and knowledge with regarding to the domain. To represent the structured and machine-readable information, it is natural to represent the information with ontology. Ontology Based Information Extraction (OBIE) [179] has extensively used this representation. With OBIE, the information extracted is not only well structured but also represented by predicates in the ontology which are easy for sharing and reuse. In other semantic data mining research, ontology is used for the post pruning and filtering of the association rule mining results [104, 105, 103].

7.2 Providing prior knowledge and constraints

Ontology could provide data mining algorithms with a priori knowledge, which either guides the mining process or reduces/constrains the search space. The definition and reuse of prior knowledge is one of the most important problems for semantic data mining. As a formal specification of concepts and relationships, ontology is a natural way to encode the formal semantics of prior knowledge. The encoded prior knowledge has the potential to guide and influence all stages of the data mining process, from preprocessing to result filtering and representation. For example, Liu et al. [98] developed a RDF hypergraph representation to capture information from both ontologies and data. Ontologies are incorporated into the graph representation of the data as the priori knowledge to bias the graph structure and also representing the distances between terms and concepts in the graph. The approach transforms the hypergraph and weighted hyperedges into a bipartite graph to represent both the data and ontology in a uniformed structure. Random walk with restart over the bipartite graph is performed to generate semantic associations. Whenever the random walk goes through the ontology-based edges, the domain knowledge encoded in ontologies bridges the latent semantic relations underneath the data with rich semantics.

As a collection of concepts and predicates, ontology has the ability to perform logic reasoning and thus make consistent inference for those predicates. In semantic data mining, the ability to make consistent inference is usually represented as constraints. The set of constraints powered by the ontology have the ability to detect inconsistent data and results in the preprocessing stage, the algorithm execution stage, and the result filtering and generation stage. For example, Balcan et al. [9] incorporated ontology as consistency constraints into multiple related classification tasks. The ontology specifies the constraints between multiple classification tasks. Carlson et al. [26] presented a semi-supervised information extraction algorithm that couples the training of many information extractors. Using ontology as constraints on the set of extractors, it yields more accurate results.

Claudia Marinica et al. [104, 105] presented post-processing of the association rule mining results using ontology for consistency checking. Invalid or inconsistent association rules are pruned and filtered out with the help of ontology and an inference engine.

7.3 Formally representing data mining results

Ontology could provide a formal way for representing the data mining flow, from data preprocessing to mining results. The well designed data mining systems should present results and discovered patterns in a formal and structured format, so that data mining results are capable to be interpreted as domain knowledge and to further enrich and improve current knowledge bases. Ontology is one of the way to represent the data mining results in a formal and structured way. As a formal definition of concepts and relationships, ontology can encode rich semantics for different domains. The data mining results from different domains and tasks conform naturally with the representation of ontology, for example, information extraction and association rule mining. Specifically, in ontology-based information extraction (OBIE) [119, 179], the extracted information are a set of annotated terms from the document with the relations defined in the ontology. It is therefore straight forward to represent the extracted information with ontology.

Wimalasuriya and Dou [179] claimed that ontology is a valid form to represent the OBIE results in a semantic rich format. Encoding OBIE results in the formal structure of ontology could streamline the data mining process of other data mining tasks that need to make use of the current result. The inference engines which was designed in the field of knowledge engineering could perform consistency checking that validate the data mining results and clean out the inconsistent results. OBIE systems can extract information with higher recall and accuracy compared with traditional IE systems. The ontology in OBIE provides the function as a conceptual framework and consistency checking. It also organizes the extracted information in a formal and structured way using explicit ontology representation. Similarly, ontology-based association pattern mining method [98] can represent latent semantic associations.

8 Semantic Deep Learning

Machine learning and data mining tasks were found to involve with rich data semantics more often than not. Identifying key semantics from data is often the primary goal for many important machine learning and data mining tasks, such as image recognition and information extraction. In other scenario, such as clustering, classification and recommendation, data semantics can play important roles as well, by either promoting the task performances or producing semantic rich results. Unfortunately, fully exploiting the data semantics is often a nontrivial task. Machine learning and data mining algorithms are usually generic methods that were designed for data from different domains and scenarios. Data from a specific domain usually carries domain specific semantics. The generic algorithms lack the ability to identify and make use of semantics across different domains and applications. For such reason, researchers have claimed that there exists semantic gaps between the data, algorithm, and results in all stages of machine learning and data mining including preprocessing, algorithm execution, and result generation [45].

As we have shown in previous sections, both deep learning and semantic data mining were proved to have the abilities in reducing the semantic gaps from either a data driven or a knowledge driven perspective. We have shown in section 2 and section 3, deep-architectures have demonstrated

their exceptional abilities in learning efficient, hierarchical, distributed data representations which could further facilitate many other machine learning tasks.

Increasing level of abstractions are often observed along with the increasing level of layers in deep-architectures, especially the ones in semantic rich machine learning and data mining tasks. For example, in deep learning based image processing, lower level layer representations were often found to correspond to low level image abstractions, such as edges and object parts, through which abstractions in high level layers can often encode representations of human face and automobiles. We have also shown in section 4 one key benefit of such incremental data abstractions which is the higher level data representations in a deep-architecture could often be related with more abstract human cognitive concepts and activities [70, 94, 84], such as image scenes [46], sentiments [164, 152, 151], semantics of speech and natural language [35, 36]. On the other hand, as we have shown in section 7.1, the formal knowledge can bridge the semantic gap between the data, applications, data mining algorithms, and data mining results. Formal knowledge can often reduce semantic gap from data through semantic aware preprocessing [83, 126, 165], semantic data annotation [48, 85], and producing semantic rich data mining results [179].

Nevertheless, evidence is still scarce that both deep learning and semantic data mining techniques have fully exploited the limits of many machine learning tasks. Few evidences have shown the association of deep learning with high level cognitive abilities of human being, such as reasoning, making inferences, comprehending and interpreting human knowledge. On the other hand, although the study of semantic data mining have reported data mining results with better precision, consistency and more importantly, richer data semantic, it is often observed that most contemporary semantic data mining methods could only make use of very limited aspects of the rich semantics encoded in the formal knowledge representation. As both semantic data mining and deep learning has the potential to bridge the semantic gaps between low level sensory data and high level cognitive concepts, it is intuitive to speculate that it is now more promising to further bridge the semantic gaps between domain knowledge, machine learning algorithms and data through both semantic data mining and deep learning. We hypothesize that the formal knowledge representation could assist in many facets of deep learning technique in a similar way as in semantic data mining, including deep-architecture designing, parameter tunning, feature explaining and result interpretation. We formally define the term, *semantic deep learning*, as the deep learning techniques with the assistant of formal knowledge representation. In this report, we specifically expoit the possibility of semantic deep learning with the most popular form of knowledge representation, ontology. We will present the next few sections our study on how to perform semantic aware classification task by exploring the rich semantics from the ontology on the domain of the data labels. The goal of this study is to

- Develop a deep-architecture design that is guided by ontology,
- Promote the classification task in a semantic rich paradigm by exploiting the relations between the data labels and their supper concepts using ontology,
- Improve the classification accuracy with the reasoning ability of ontology.

8.1 Deep Learning Ontology

We start by introducing our deep learning ontology (DLO), which is constructed to formally encode the concepts and relations in the domain of the data label. It contains specifications of data label,

their super concepts and three types of relations between these concepts, (1) subclass, which defines the subsumption relations (e.g. bird and farm bird), (2) disjoint, which defines relations between disjoint or contradictory entities (e.g. duck and chicken), (3) coexists, which defines the relation of closely related concepts (e.g. butterfly and flower). In our ontology-based deep learning framework, DLO plays multiple roles through all stages of the machine learning process, including guidance of the deep network structure, consistency checking, and semantic enrichment results with multi-level output. The subsumption relation in the DLO presents a tree structure of the concepts, in which the all leaves concepts and some internal concepts correspond to classification labels of the data. The concepts of ontology design should cover all the labels that will appear in this classification task. Other ontology concepts are the super concepts of the classification labels. In figure 5, we present one example of deep learning ontology for the domain of birds, in which chicken, duck, sparrow, corresponds the labels of the data. The other concepts farm bird, wild bird, and bird are defined through the domain knowledge of the bird domain.

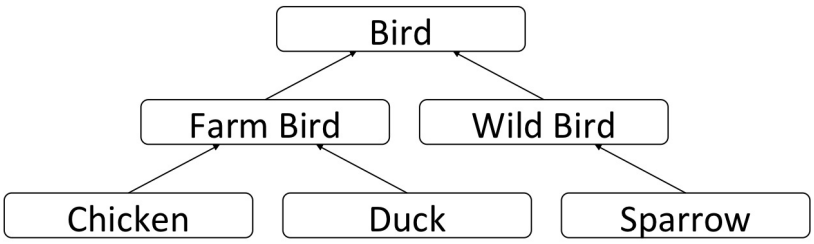
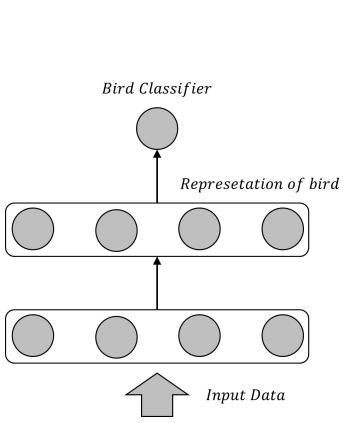


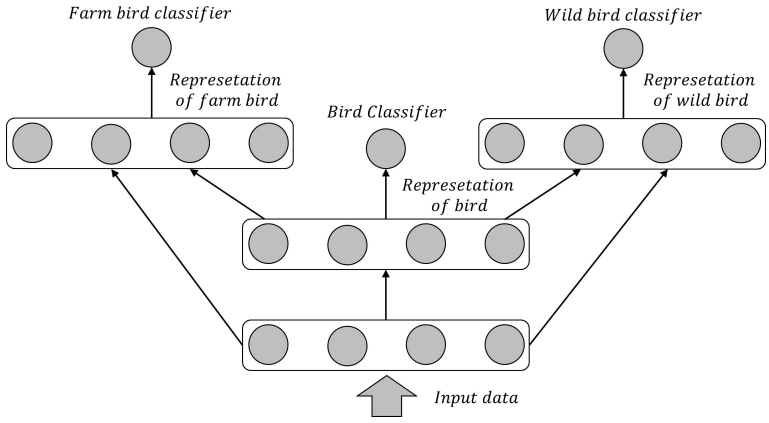
Figure 5: Semantic Deep Learning Ontology

8.2 Deep-architecture from Deep Learning Ontology

The deep learning ontology specifies the relations between the classification labels and their super concepts. We further design a deep-architecture for the ontology based classification task with the assistant of the deep learning ontology. The design of deep-architecture follows the subsumption relation hierarchy of the deep learning ontology by proceeding the following steps:



(a) Representation of bird concept



(b) Representation of farm bird and wild bird concept

Step zero, initialization. For the top level concept of deep learning ontology, such as *object* for

a general ontology or *animal* for the animal domain ontology, build a one layer RBM with input variable v_0 corresponds to input data and hidden variables h_0 as the representation of the top level concept. For example, figure 6(a) shows the deep architecture at current step for the bird concept of the ontology in figure 5. After the initialization, Go to step two.

Step one, extend deep-architecture with subconcepts. For each representation layer built in the last iteration or at initialization, identify the corresponding concept A in the deep learning ontology. For each B *is_a* A subsumption relation in the deep learning ontology, build a RBM with input variable $v_A = \{v_{IA}, v_{RA}\}$, in which V_{IA} accepts the raw input data, v_{RA} is the representation of concept A built in the last iteration. The hidden variables of this RBM h_B corresponds to the representation of concept B.

Step two, unsupervised pre-training. We train the new RBM layer by unsupervised pre-training with unlabeled data first. For each input v_R, v_{IR} accepts direct input of the raw data, v_R is generated through the RBM layers of its super concepts.

Step three, supervised fine tuning. For each representation learned at step two for concept B, attach a softmax layer from h_B. We then train the network by supervised training using using labled data. Each data label is promoted to the corresponding concept of the current layer using subsumption inference of the deep learning ontology. For example, the data with label *chicken* is promoted as the label *bird* during the supervised training of the bird representation. Finally, if there still exists concepts in the deep ontology that not do correspond to a representation in the deep-architecture, go to step one.

Note that our deep-architecture design induced from the ontology adopts a top-down paradigm with regarding to the subsumption structure of ontology. The data representations and features learned from a super concept were used to assist in the deep-learning process for its subconcepts. Our design paradigm is different from the current prevalent deep-architecture designs which mostly adopt a bottom up feature learning paradigm. For example, in DBN, through greedy level-wised pre-training, lower level features, such as edges in image data, are combined to construct high level features and representations, such as objects. It is obvious that the bottom up feature learning is promising in capturing the *part-of* relation between features, however, it is not capable to model feature space that is dominated by other relations, such as the *is_a* relation in label space. Our deep-architecture design follows the ontology subsumption architecture, thus could model the *part-of* relations between concepts and features. Note that our deep-architecture will result in a design that more abstract concepts corresponds to smaller number of non-linear transformations as in contrast with other deep-architectures. We argue that such design in fact follows the biological intuition for object recognition and classification. It has been found that biological neural network responds much fast for identification of general concepts, such as if this is an automobile, than concrete concepts, such as if this is a Honda or Ford.

8.3 Large scale semantic rich unsupervised learning by deep inference

We argue that a good learned representation is necessary capture not only the variations of the data but also the key factors that is necessary for our further machine learning tasks. In real world machine learning practice, it is usually hard to tell how much the learned representation is capable of supporting the learning task. We usually expect the data representation should be learned from the unlabeled data, since the labeled data is luxurious to obtain to satisfy the needs of a good data representation. One would be hard to expect the data representation learned could be related with

our machine learning tasks.

The ontology deep learning architecture build a framework for the representation of the data which can produce multiple-hierarchical output. As each output of such deep-architecture corresponds to one concept in the deep learning ontology, inferences can be performed on these outputs using the deep learning ontology. As the deep learning ontology defines a clear taxonomy subsumption relation architecture for all concepts, the inconsistency found is usually corresponds to a path from the top concept to one leaf concept. For example, if the classification output from the deep-architecture produces, $bird, Y$, $penguin, N$, $emperopenguin, Y$, inconsistency can be found in the subsumption path of $emperorpenguin \rightarrow penguin \rightarrow bird$. Further training through unlabeled data can proceed through the consistency checking and back propagation process. Such process contains the following steps,

Step one, identify the inconsistency and inconsistency path. We first identify the inconsistency in the output label, for instance, for the $emperorpenguin \rightarrow penguin \rightarrow bird$ example, we can identify the inconsistency $emperorpenguin, Y \rightarrow penguin, N$ using the inference checking ability of ontology. The inconsistency path is defined as the path from the top concept to the concepts where inconsistency happens. It contains all the concepts related with the inconsistency if we like to fix and back propagate the error we found.

Step two, identify the inconsistency correcting schema. For each inconsistency path, we further identify few inconsistency correcting schemas that would fix the inconsistency. For example, for the $bird, Y$, $penguin, N$, $emperopenguin, Y$ output on the inconsistency path. Two possible inconsistency correcting schemas would be,

- Schema A: penguin, $N \rightarrow penguin, Y$
- Schema B: bird , $Y \rightarrow bird$, N and emperor penguin, $Y \rightarrow emperorpenguinN$

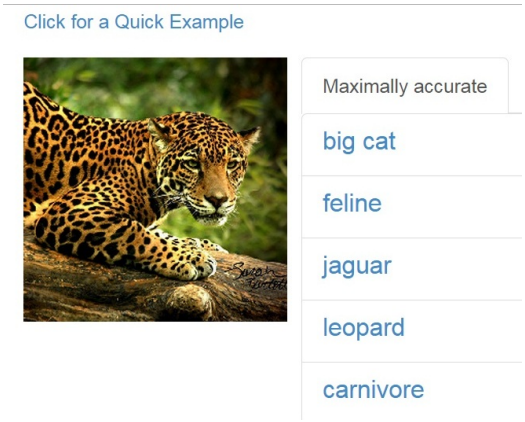
For each inconsistency correcting schema, we compute a target function to estimate the cost of correcting. We define $L_S = \{e|e \in correctedlabelsinschemaS\}$. We can estimate the cost of schema S by

$$C_S = \sum_{e \in L_S} \|J_e(x)\|_F^2 \quad = \sum_{e \in L_S} \sum_{ij} \left(\frac{\partial h_e(x)}{\partial x_i} \right) \tag{14}$$

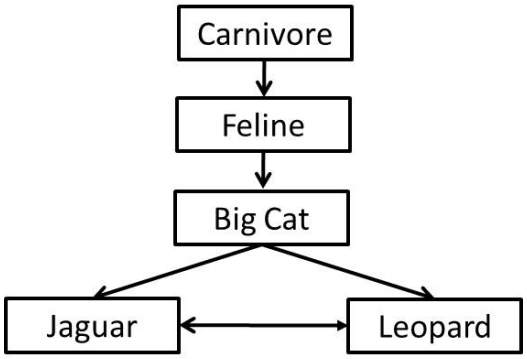
, in which $J_f(x)$ is the Frobenius norm of the Jacobian $J_x(x)$ of the non-linear mapping. $J_f(x)$ measures the robustness of the representation $f_e(x)$ by estimating the sensitivity to the input, i.e. the level of contrast to the neighbour of the training data [133]. We further train the deep-architecture by correcting the according the in the schema S with largest contrast, i.e. the best robustness of representation.

8.4 Multi-level output of classification result

Without the knowledge of nature relations between labels, the output could be either vague or redundant. For example, as shown in figure 6(c) an output of jaguar from the demo [1] of Caffe [77], the state-of-art online image classification neural network framework. The top five output from a image of jaguar are big cat, feline, jaguar, leopard, carnivore respectively. It is easy to observe that there exists obvious structural relations between the five output labels. With the help of the deep learning ontology, the relations could be defined shown in figure 6(d).



(c) Object detection output from Caffe



(d) Relations between output labels from Caffe

By specifying using the deep learning ontology, the outputs in figure 6(d) are largely redundant. For example, for the output *jaguar* and *leopard*, as they are defined as similar concept in the deep learning ontology, we can produce one label *jaguar* (*leopard*) output instead of two. Further more, if the two output labels A and B are disjoint, it is very likely the current system does not have enough ability to distinguish the two concepts. For example, as the *jaguar* and *leopard* shares very little difference, even most human does not equip with the knowledge to identify from one and another. As the object detection system would be very likely to produce output with large error rate, we can output the higher level concepts such as *bigcat* or $feline$ instead which largely reduces the classification error.

References

[1] The Caffe Demos. http://demo.caffe.berkeleyvision.org/.

[2] The National Center for Biomedical Ontology. `http://www.bioontology.org/`.

[3] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.

[4] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the 5th ACM International Conference on Digital Libraries*, pages 85–94, 2000.

[5] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *the 8th European Conference on Computer Vision*, pages 469–481. Springer, 2004.

[6] H. Alani, S. Kim, D. E. Millard, M. J. Weal, W. Hall, P. H. Lewis, and N. R. Shadbolt. Automatic ontology-based knowledge extraction from web documents. *Intelligent Systems, IEEE*, 18(1):14–21, 2003.

[7] W. Aljandal, V. Bahirwani, D. Caragea, and W. H. Hsu. Ontology-aware classification and association rule mining for interest and link prediction in social networks. In *AAAI Spring Symposium: Social Semantic Web: Where Web 2.0 Meets Web 3.0*, pages 3–8, 2009.

[8] M. Allahyari, K. J. Kochut, and M. Janik. Ontology-based text classification into dynamically defined topics. In *IEEE International Conference on Semantic Computing*, pages 273–278, 2014.

[9] N. Balcan, A. Blum, and Y. Mansour. Exploiting ontology structures and unlabeled data for learning. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1112–1120, 2013.

[10] A. Bellandi, B. Furletti, V. Grossi, and A. Romei. Ontology-driven association rule extraction: A case study. *Contexts and Ontologies Representation and Reasoning*, page 10, 2007.

[11] Y. Bengio. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

[12] Y. Bengio. Deep learning of representations: Looking forward. In *Statistical Language and Speech Processing*, pages 1–37. Springer, 2013.

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

[14] Y. Bengio and O. Delalleau. On the expressive power of deep architectures. In *Algorithmic Learning Theory*, pages 18–36. Springer, 2011.

[15] Y. Bengio, O. Delalleau, and N. L. Roux. The curse of highly variable functions for local kernel machines. In *Advances in Neural Information Processing Systems 18*, page 2006. MIT Press, 2006.

[16] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems*, 19:153, 2007.

[17] Y. Bengio, Y. LeCun, et al. Scaling learning algorithms towards AI. *Large-scale Kernel Machines*, 34(5), 2007.

[18] C. M. Bishop et al. *Pattern recognition and machine learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[19] A. L. Blum and R. L. Rivest. Training a 3-node neural network is NP-complete. *Neural Networks*, 5(1):117–127, 1992.

[20] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[21] A. Bordes, J. Weston, R. Collobert, and Y. Bengio. Learning structured embeddings of knowledge bases. In *Conference on Artificial Intelligence*, number EPFL-CONF-192344, 2011.

[22] Y.-l. Boureau, Y. L. Cun, et al. Sparse feature learning for deep belief networks. In *Advances in Neural Information Processing Systems*, pages 1185–1192, 2008.

[23] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.

[24] X. Cai, C. Wang, B. Xiao, X. Chen, and J. Zhou. Deep nonlinear metric learning with independent subspace analysis for face verification. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 749–752. ACM, 2012.

[25] I. Cantador, A. Bellogín, and P. Castells. Ontology-based personalised and context-aware recommendations of news items. In *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pages 562–565, 2008.

[26] A. Carlson, J. Betteridge, E. R. Hruschka Jr, and T. M. Mitchell. Coupling semi-supervised learning of categories and relations. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, pages 1–9, 2009.

[27] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka Jr, and T. M. Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pages 101–110, 2010.

[28] D. Chen, R. Socher, C. D. Manning, and A. Y. Ng. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. *arXiv preprint arXiv:1301.3618*, 2013.

[29] X. Chen, A. Eversole, G. Li, D. Yu, and F. Seide. Pipelined back-propagation for context-dependent deep neural networks. In *Annual Conference of the International Speech Communication Association*, 2012.

[30] Y. Cho and L. K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*, pages 342–350, 2009.

[31] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649. IEEE, 2012.

[32] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167. ACM, 2008.

[33] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[34] G. Dahl, A.-r. Mohamed, G. E. Hinton, et al. Phone recognition with the mean-covariance restricted Boltzmann machine. In *Advances in Neural Information Processing Systems*, pages 469–477, 2010.

[35] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Large vocabulary continuous speech recognition with context-dependent DBN-HMMs. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4688–4691. IEEE, 2011.

[36] G. E. Dahl, D. Yu, L. Deng, and A. Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):30–42, 2012.

[37] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 886–893. IEEE, 2005.

[38] L. Deng, G. Hinton, and B. Kingsbury. New types of deep neural network learning for speech recognition and related applications: An overview. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8599–8603. IEEE, 2013.

[39] L. Deng, P. Kenny, M. Lennig, V. Gupta, F. Seitz, and P. Mermelstein. Phonemic hidden markov models with continuous mixture output densities for large vocabulary word recognition. *Signal Processing, IEEE Transactions on*, 39(7):1677–1681, 1991.

[40] L. Deng, M. Lennig, F. Seitz, and P. Mermelstein. Large vocabulary word recognition using context-dependent allophonic hidden markov models. *Computer Speech & Language*, 4(4):345–357, 1990.

[41] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, et al. Recent advances in deep learning for speech research at Microsoft. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8604–8608. IEEE, 2013.

[42] L. Deng, M. L. Seltzer, D. Yu, A. Acero, A.-R. Mohamed, and G. E. Hinton. Binary coding of speech spectrograms using a deep auto-encoder. In *Annual Conference of the International Speech Communication Association*, pages 1692–1695. Citeseer, 2010.

[43] L. Deng, G. Tur, X. He, and D. Hakkani-Tur. Use of kernel deep convex networks and end-to-end learning for spoken language understanding. In *Spoken Language Technology Workshop*, pages 210–215. IEEE, 2012.

[44] T. Do, T. Arti, et al. Neural conditional random fields. In *International Conference on Artificial Intelligence and Statistics*, pages 177–184, 2010.

[45] P. Domingos. Toward knowledge-rich data mining. *Data Mining and Knowledge Discovery*, 15(1):21–28, 2007.

[46] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[47] D. W. Embley, D. M. Campbell, R. D. Smith, and S. W. Liddle. Ontology-based extraction and structuring of information from data-rich unstructured documents. In *Proceedings of the*

seventh international conference on Information and knowledge management, pages 52–59. ACM, 1998.

[48] M. Erdmann, A. Maedche, H.-P. Schnurr, and S. Staab. From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. In *Proceedings of the COLING-2000 Workshop on Semantic Annotation and Intelligent Content*, pages 79–85, 2000.

[49] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[50] H. Fan, Z. Cao, Y. Jiang, Q. Yin, and C. Doudou. Learning deep face representation. *arXiv preprint arXiv:1403.2802*, 2014.

[51] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37, 1996.

[52] A. Fischer and C. Igel. An introduction to restricted Boltzmann machines. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 14–36. Springer, 2012.

[53] S. Fodeh, B. Punch, and P.-N. Tan. On ontology-driven document clustering using core semantic features. *Knowledge and Information Systems*, 28(2):395–421, 2011.

[54] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

[55] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on Machine Learning*, pages 513–520, 2011.

[56] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

[57] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

[58] T. R. Gruber. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 43(5):907–928, 1995.

[59] F. Gutierrez, D. Dou, A. Martini, S. Fickas, and H. Zong. Hybrid Ontology-Based Information Extraction for Automated Text Grading. In *the 12th International Conference on Machine Learning and Applications*, volume 1, pages 359–364. IEEE, 2013.

[60] T. Hao, T. Raiko, A. Ilin, and J. Karhunen. Gated boltzmann machine in texture modeling. In *Artificial Neural Networks and Machine Learning*, pages 124–131. Springer, 2012.

[61] X. He, L. Deng, and W. Chou. Discriminative learning in sequential pattern recognition. *Signal Processing Magazine, IEEE*, 25(5):14–36, 2008.

[62] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean. Multilingual acoustic models using distributed deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8619–8623. IEEE, 2013.

[63] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.

[64] G. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[65] G. Hinton and R. Salakhutdinov. Discovering binary codes for documents by learning deep generative models. *Topics in Cognitive Science*, 3(1):74–91, 2011.

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

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

[68] A. Hotho, A. Maedche, and S. Staab. Ontology-based text document clustering. *Knstliche Intelligenz*, 16(4):48–54, 2002.

[69] A. Hotho, S. Staab, and G. Stumme. Ontologies improve text document clustering. In *Proceedings of the third IEEE International Conference on Data Mining*, pages 541–544, 2003.

[70] G. B. Huang, H. Lee, and E. Learned-Miller. Learning hierarchical representations for face verification with convolutional deep belief networks. In *IEEE Conferene on Computer Vision and Pattern Recognition*, pages 2518–2525. IEEE, 2012.

[71] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, Technical Report 07-49, University of Massachusetts, Amherst, 2007.

[72] D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

[73] W. IJntema, F. Goossen, F. Frasincar, and F. Hogenboom. Ontology-based news recommendation. In *Proceedings of the 2010 EDBT/ICDT Workshops*, page 16. ACM, 2010.

[74] A. K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[75] N. Jaitly, P. Nguyen, A. W. Senior, and V. Vanhoucke. Application of Pretrained Deep Neural Networks to Large Vocabulary Speech Recognition. In *Annual Conference of the International Speech Communication Association*. Citeseer, 2012.

[76] S. Ji, W. Xu, M. Yang, and K. Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.

[77] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[78] L. Jing, M. K. Ng, and J. Z. Huang. Knowledge-based vector space model for text clustering. *Knowledge and Information Systems*, 25(1):35–55, 2010.

[79] L. Jing, L. Zhou, M. K. Ng, and J. Z. Huang. Ontology-based distance measure for text clustering. In *Proceedings of SIAM SDM workshop on text mining*, 2006.

[80] J. Kang and J. Choi. An ontology-based recommendation system using long-term and short-term preferences. In *2011 International Conference on Information Science and Applications*, pages 1–8. IEEE, 2011.

[81] S. Kara, Ö. Alan, O. Sabuncu, S. Akpınar, N. K. Cicekli, and F. N. Alpaslan. An ontology-based retrieval system using semantic indexing. *Information Systems*, 37(4):294–305, 2012.

[82] V. Karkaletsis, P. Fragkou, G. Petasis, and E. Iosif. Ontology based information extraction from text. In *Knowledge-driven Multimedia Information Extraction and Ontology Evolution*, pages 89–109. Springer, 2011.

[83] N. Khasawneh and C.-C. Chan. Active user-based and ontology-based web log data preprocessing for web usage mining. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 325–328, 2006.

[84] G. Kiran, R. Shankar, and V. Pudi. Frequent itemset based hierarchical document clustering using Wikipedia as external knowledge. In *Knowledge-Based and Intelligent Information and Engineering Systems*, pages 11–20. Springer, 2010.

[85] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1):49–79, 2004.

[86] J. M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. S. Tomkins. The web as a graph: Measurements, models, and methods. In *Computing and Combinatorics*, pages 1–17. Springer, 1999.

[87] A. Krizhevsky, G. E. Hinton, et al. Foactored 3-way restricted boltzmann machines for modeling natural images. In *International Conference on Artificial Intelligence and Statistics*, pages 621–628, 2010.

[88] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[89] H.-S. Le, I. Oparin, A. Allauzen, J. Gauvain, and F. Yvon. Structured output layer neural network language models for speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(1):197–206, 2013.

[90] Q. V. Le. Building high-level features using large scale unsupervised learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8595–8598. IEEE, 2013.

[91] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.*, 1(4):541–551, Dec. 1989.

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

[93] H. Lee, C. Ekanadham, and A. Y. Ng. Sparse deep belief net model for visual area V2. In *Advances in Neural Information Processing Systems*, pages 873–880, 2008.

[94] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 609–616. ACM, 2009.

[95] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *arXiv preprint arXiv:1301.3592*, 2013.

[96] D. Li and D. Yu. *Deep Learning: Methods and Applications* . Now Publishers Inc, Delft,The Netherlands, 2014.

[97] D. Lindberg, B. Humphries, and A. McCray. The Unified Medical Language System. *Methods of Information in Medicine*, 32(4):281–291, 1993.

[98] H. Liu, D. Dou, R. Jin, P. LePendu, and N. Shah. Mining biomedical Ontologies and data using RDF hypergraphs. In *the 12th International Conference on Machine Learning and Applications*, volume 1, pages 141–146. IEEE, 2013.

[99] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157. IEEE, 1999.

[100] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

[101] N. R. Mabroukeh and C. I. Ezeife. Using domain ontology for semantic web usage mining and next page prediction. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1677–1680. ACM, 2009.

[102] C. D. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

[103] G. Mansingh, K.-M. Osei-Bryson, and H. Reichgelt. Using ontologies to facilitate post-processing of association rules by domain experts. *Information Sciences*, 181(3):419–434, 2011.

[104] C. Marinica and F. Guillet. Knowledge-based interactive postmining of association rules using ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 22(6):784–797, 2010.

[105] C. Marinica, F. Guillet, and H. Briand. Post-processing of discovered association rules using ontologies. In *Data Mining Workshops, 2008. ICDMW'08. IEEE International Conference on*, pages 126–133, 2008.

[106] R. Memisevic and G. E. Hinton. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural Computation*, 22(6):1473–1492, 2010.

[107] S. E. Middleton, H. Alani, and D. C. De Roure. Exploiting synergy between ontologies and recommender systems. *arXiv preprint cs/0204012*, 2002.

[108] S. E. Middleton, D. C. De Roure, and N. R. Shadbolt. Capturing knowledge of user preferences: ontologies in recommender systems. In *Proceedings of the 1st International Conference on Knowledge Capture*, pages 100–107. ACM, 2001.

[109] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure. Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, 22(1):54–88, 2004.

[110] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[111] T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. Cernockỳ. Empirical Evaluation and Combination of Advanced Language Modeling Techniques. In *Annual Conference of the International Speech Communication Association*, pages 605–608, 2011.

[112] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur. Recurrent neural network based language model. In *Annual Conference of the International Speech Communication Association*, pages 1045–1048, 2010.

[113] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic Regularities in Continuous Space Word Representations. In *Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, pages 746–751, 2013.

[114] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[115] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[116] R. Mittelman, B. Kuipers, S. Savarese, and H. Lee. Structured Recurrent Temporal Restricted Boltzmann Machines. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1647–1655, 2014.

[117] V. Mnih, H. Larochelle, and G. E. Hinton. Conditional restricted boltzmann machines for structured output prediction. *arXiv preprint arXiv:1202.3748*, 2012.

[118] A.-r. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):14–22, 2012.

[119] H.-M. Müller, E. E. Kenny, and P. W. Sternberg. Textpresso: an ontology-based information retrieval and extraction system for biological literature. *PLoS biology*, 2(11):e309, 2004.

[120] V. Nair and G. E. Hinton. 3D object recognition with deep belief nets. In *Advances in Neural Information Processing Systems*, pages 1339–1347, 2009.

[121] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning*, pages 689–696, 2011.

[122] M. Norouzi, M. Ranjbar, and G. Mori. Stacks of convolutional restricted Boltzmann machines for shift-invariant feature learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2735–2742. IEEE, 2009.

[123] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[124] B. A. Olshausen et al. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

[125] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research*, 37(23):3311–3325, 1997.

[126] D. Perez-Rey, A. Anguita, and J. Crespo. Ontodataclean: Ontology-based integration and preprocessing of distributed data. In *Biological and Medical Data Analysis*, pages 262–272. Springer, 2006.

[127] C. Poultney, S. Chopra, Y. L. Cun, et al. Efficient learning of sparse representations with an energy-based model. In *Advances in Neural Information Processing Systems*, pages 1137–1144, 2006.

[128] A. Pretschner and S. Gauch. Ontology based personalized search. In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, pages 391–398, 1999.

[129] N. Pudota, A. Dattolo, A. Baruzzo, F. Ferrara, and C. Tasso. Automatic keyphrase extraction and ontology mining for content-based tag recommendation. *International Journal of Intelligent Systems*, 25(12):1158–1186, 2010.

[130] C. Ramakrishnan, W. H. Milnor, M. Perry, and A. P. Sheth. Discovering informative connection subgraphs in multi-relational graphs. *ACM SIGKDD Explorations Newsletter*, 7(2):56–63, 2005.

[131] M. Ranzato and G. E. Hinton. Modeling pixel means and covariances using factorized third-order Boltzmann machines. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2551–2558. IEEE, 2010.

[132] S. Riedel, L. Yao, and A. McCallum. Modeling relations and their mentions without labeled text. In *Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases*, pages 148–163, Berlin, Heidelberg, 2010. Springer-Verlag.

[133] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning*, pages 833–840, 2011.

[134] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.

[135] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5, 1988.

[136] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.

[137] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[138] T. N. Sainath, B. Kingsbury, A.-r. Mohamed, and B. Ramabhadran. Learning filter banks within a deep neural network framework. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 297–302. IEEE, 2013.

[139] T. N. Sainath, B. Kingsbury, B. Ramabhadran, P. Fousek, P. Novak, and A.-r. Mohamed. Making deep belief networks effective for large vocabulary continuous speech recognition. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 30–35. IEEE, 2011.

[140] R. Salakhutdinov and G. Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, 2009.

[141] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[142] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *arXiv preprint arXiv:1503.03832*, 2015.

[143] F. Seide, G. Li, and D. Yu. Conversational Speech Transcription Using Context-Dependent Deep Neural Networks. In *Annual Conference of the International Speech Communication Association*, pages 437–440, 2011.

[144] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[145] P. Sermanet, R. Hadsell, M. Scoffier, U. Muller, and Y. LeCun. Mapping and planning under uncertainty in mobile robots with long-range perception. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2525–2530. IEEE, 2008.

[146] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *Neural Networks , The 2011 International Joint Conference on*, pages 2809–2813. IEEE, 2011.

[147] S. M. Siniscalchi, D. Yu, L. Deng, and C.-H. Lee. Exploiting deep neural networks for detection-based speech recognition. *Neurocomputing*, 106:148–157, 2013.

[148] R. Socher, D. Chen, C. D. Manning, and A. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, pages 926–934, 2013.

[149] R. Socher, E. H. Huang, J. Pennin, C. D. Manning, and A. Y. Ng. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, pages 801–809, 2011.

[150] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th International Conference on Machine Learning*, pages 129–136, 2011.

[151] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics, 2011.

[152] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, volume 1631, page 1642. Citeseer, 2013.

[153] W. Song, C. H. Li, and S. C. Park. Genetic algorithm for text clustering using ontology and evaluating the validity of various semantic similarity measures. *Expert Systems with Applications*, 36(5):9095–9104, 2009.

[154] N. Srivastava. *Improving neural networks with dropout*. PhD thesis, University of Toronto, 2013.

[155] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems*, pages 2222–2230, 2012.

[156] R. Studer, V. R. Benjamins, and D. Fensel. Knowledge engineering: Principles and methods. *Data and Knowledge Engineering*, 25(1):161–197, 1998.

[157] Y. Sun, X. Wang, and X. Tang. Deep convolutional network cascade for facial point detection. In *Computer Vision and Pattern Recognition , IEEE Conference on*, pages 3476–3483. IEEE, 2013.

[158] Y. Sun, X. Wang, and X. Tang. Hybrid deep learning for face verification. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1489–1496. IEEE, 2013.

[159] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *Computer Vision and Pattern Recognition , IEEE Conference on*, pages 1891–1898. IEEE, 2014.

[160] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. *arXiv preprint arXiv:1412.1265*, 2014.

[161] I. Sutskever, G. E. Hinton, and G. W. Taylor. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2009.

[162] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[163] A. D. Szlam, K. Gregor, and Y. L. Cun. Structured sparse coding via lateral inhibition. In *Advances in Neural Information Processing Systems*, pages 1116–1124, 2011.

[164] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708. IEEE, 2014.

[165] D. Tanasa and B. Trousse. Advanced data preprocessing for intersites web usage mining. *Intelligent Systems, IEEE*, 19(2):59–65, 2004.

[166] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *the 11th Euronpean Conference on Computer Vision*, pages 140–153. Springer, 2010.

[167] G. W. Taylor and G. E. Hinton. Factored conditional restricted Boltzmann machines for modeling motion style. In *Proceedings of the 26th annual international conference on machine learning*, pages 1025–1032. ACM, 2009.

[168] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. In *Advances in Neural Information Processing Systems*, pages 1345–1352, 2006.

[169] G. Tesauro. *Practical issues in temporal difference learning*. Springer, 1992.

[170] The_gene_ontology_consortium. Creating the gene ontology resource: design and implementation. *Genome Res.*, 11(8):1425–1433, August 2001.

[171] A. Thor, P. Anderson, L. Raschid, S. Navlakha, B. Saha, S. Khuller, and X.-N. Zhang. Link prediction for annotation graphs using graph summarization. In *the 10th International Semantic Web Conference*, pages 714–729. Springer, 2011.

[172] J. Turian, L. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, 2010.

[173] P. D. Turney, P. Pantel, et al. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37(1):141–188, 2010.

[174] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103. ACM, 2008.

[175] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, 2010.

[176] M. P. Wellman and M. Henrion. Explaining explaining away. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):287–292, 1993.

[177] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: Learning to rank with joint word-image embeddings. *Machine Learning*, 81(1):21–35, 2010.

[178] D. C. Wimalasuriya and D. Dou. Components for information extraction: Ontology-based information extractors and generic platforms. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management*, pages 9–18, 2010.

[179] D. C. Wimalasuriya and D. Dou. Ontology-based information extraction: An introduction and a survey of current approaches. *Journal of Information Science*, 36(3):306–323, 2010.

[180] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In *Computer Vision and Pattern Recognition , IEEE Conference on*, pages 529–534. IEEE, 2011.

[181] F. Wu and D. S. Weld. Autonomously semantifying wikipedia. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, CIKM '07, pages 41–50, 2007.

[182] F. Wu and D. S. Weld. Automatically refining the wikipedia infobox ontology. In *Proceedings of The 17th International Conference on World Wide Web*, pages 635–644. ACM, 2008.

[183] M. Xu, L.-Y. Duan, J. Cai, L.-T. Chia, C. Xu, and Q. Tian. HMM-based audio keyword generation. In *Advances in Multimedia Information Processing*, pages 566–574. Springer, 2005.

[184] D. Yu, L. Deng, and G. Dahl. Roles of pre-training and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2010.

[185] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[186] X. Zhang, L. Jing, X. Hu, M. Ng, J. Xia, and X. Zhou. Medical document clustering using ontology-based term similarity measures. *International Journal of Data Warehousing and Mining*, 4(1):62–73, 2008.

[187] Z. Zhang, P. Luo, C. C. Loy, and X. Tang. Facial landmark detection by deep multi-task learning. In *Computer Vision–ECCV 2014*, pages 94–108. Springer, 2014.

[188] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity-preserving face space. In *Computer Vision , IEEE International Conference on*, pages 113–120. IEEE, 2013.

[189] Z. Zhu, P. Luo, X. Wang, and X. Tang. Multi-view perceptron: a deep model for learning face identity and view representations. In *Advances in Neural Information Processing Systems*, pages 217–225, 2014.