Iran Journal of Computer Science
https://doi.org/10.1007/s42044-018-00027-6

REVIEW ARTICLE

A review of various semi-supervised learning models with a deep learning and memory approach

Jamshid Bagherzadeh[1] · Hasan Asil[1]

Received: 4 April 2018 / Accepted: 27 November 2018

Abstract

Based on data types, four learning methods have been presented to extract patterns from data: supervised, semi-supervised, unsupervised, and reinforcement. Regarding machine learning, labeled data are very hard to access, although unlabeled data are usually collected and accessed easily. On the other hand, in most projects, most of the data are unlabeled but some data are labeled. Therefore, semi-supervised learning is more practical and useful for solving most of the problems. Different semi-supervised learning models have been introduced such as iterative learning (self-training), generative models, graph-based methods, and vector-based techniques. In addition, deep neural networks are used to extract data features using a multilayer model. Various models of this method have been presented to deal with semi-supervised data such as deep generative, virtual adversarial, and Ladder models. In semi-supervised learning, labeled data can contribute significantly to accurate pattern extraction. Thus, they can result in better convergence by having greater effects on models. The aim of this paper was to analyze the available models of semi-supervised learning with an approach to deep learning. A research solution for future studies is to benefit from memory to increase such an effect. Memory-based neural networks are new models of neural networks which can be used in this area.

Keywords Semi-supervised learning · Deep neural networks · Ladder · SemiBoost · RegBoost · ADGM

1 Introduction

Creating data models is useful in various sciences and engineering fields. Basically, such models can be used to obtain knowledge from data, make predictions, or both. Extracting a general model of events is referred to as learning in which a plethora of data is dealt with. Such data are obtained inexpensively, although the knowledge of such data is not acquired simply at a low cost. In fact, machine learning seeks to discover and codify algorithms by which machines become capable of learning [1]. For instance, it is desirable that spams should be identified. In other words, a prediction model should be able to show whether a message is a spam by regarding every message as an input. Here a learning algorithm is responsible for finding a function which can allocate a Yes/No label to every email message. Machine learning has different applications in data mining, bioinformatics, computer games, the Internet data processing, etc. [2].

Learning problems are divided into four groups: supervised, unsupervised, semi-supervised, and reinforcement [3]. Supervised learning is a type of machine learning in which inputs and outputs are clear. Based on the information provided for a learner, the system tries to present a function from the input to the output. In this type of learning, a supervisor should provide labeled data, meaning the data used for learning. Such data possess the input and output (target) values. Moreover, a set of labeled training data is observed by the system [4]. The aim of this method is to obtain rules and latent relationships of data to predict the labels of unobserved data after learning. On the contrary, there are no specific data in advance in unsupervised learning. The goal is not to establish a relationship between the input and the output. Instead, a learner seeks a specific structure. As a matter of fact, data classification is sought in this type of learning in which a set of unlabeled data is observed by the system. The aim of unsupervised learning is to organize data to cluster data, detect outlier data, and decrease data dimensions in a way that the main characteristics of data sets are

✉ Hasan Asil
h.asil@iauazar.ac.ir

[1] Department of Computer Engineering, Faculty of Electrical and Computer Engineering, Urmia University, Urmia, Iran

Published online: 06 December 2018

maintained [5]. Another method is semi-supervised learning used in problems including both labeled and unlabeled types of data. In semi-supervised learning, both types of data are employed at the same time to increase learning precision. Semi-supervised classification promises an appropriate prediction of label function with high precision in addition to human effort to label data. Thus, it is highly valuable both in theory and practice. Semi-supervised classification can be used for voice recognition, data mining, video surveillance, and prediction. Another method is the reinforcement learning in which a feedback is given as a positive comment (reward) or a negative comment (penalty) to a learning agent. In fact, the currently stored knowledge is reinforced or weakened with reward or penalty signals. Unlike supervised learning, an agent is never told what the right action is in every situation in reinforcement learning. Instead, a criterion is used to tell an agent how good or bad an action is [3]. In this method, the system observes data iteratively and takes a specific action on a piece of data. Then the system receives a specific reward for that action. The final goal of such a system is to select an action which can receive the maximum reward in the future [6].

In different applications of machine learning, labeled data are very hard to access, whereas unlabeled data are easy to collect and access. For instance, webpages are easy to access; however, there are a limited number of classified or labeled webpages because semi-supervised learning is more efficient in classification due to much use of unlabeled data. Generally, unlabeled data are used in this method to change labeled data hypotheses [7]. Due to the limited use of labeled data in semi-supervised learning, this method is of high importance both in theory and practice [8].

Semi-supervised learning is considered as an appropriate approach in the lack of labeled data and the use of unlabeled data in the training phase. Finding a method that makes use of unlabeled data is of utmost importance practical applications. In the early stages of training the semi-supervised classifier, the selection of proper and reliable unlabeled data is very important. In some cases, the collection of labeled data by assigning a label to each of them is a time-consuming process while unlabeled data are readily available. Therefore, finding a method that utilizes unlabeled data more is strikingly valuable in practical applications. Some of the usages in text-processing applications are: spam detection from normal messages, classification of documents and web pages and recommendation rating of pages based on user's interest. In video monitoring, the procedure of recognizing and labeling all the faces shown in the pictures by humans is really time-consuming. Also, the prediction of protein structure in bioinformatics requires several months of laboratory work. Hence, the need for a semi-supervised method that can compensate for the lack of a number of labeled data using the unlabeled data is completely felt. However, the importance of supervised learning is slightly greater than the application mentioned above. So that most of the learnings in humans and animals are done in a semi-supervised manner and this increases the importance of supervised learning [2, 4].

A question that arises from the use of semi-supervised methods is that whether any improvement is achieved in classification by considering the unlabeled data? Or what happens that semi-supervised methods become useful? It is clear that the improvement in classification is obtained regarding the distribution of input data. It is remarkable to note that, sometimes semi-supervised learning does not make any changes in classification.

The main question in the field of semi-supervised learning is that under what conditions these approaches should be employed? In other words, for what issues or on which data sets, the use of this method will improve the performance of the learner agent or the separator. More generally, does utilizing unlabeled data really have effect on the betterment of the performance? In fact, regarding the investigation of the articles presented in this field, the answer of the mentioned question is yes, but there is a fundamental condition which is the suitability of sample distribution for the isolation problem estimated using unlabeled data. Indeed, unlabeled data help to gain the basic knowledge about the data distribution.

One of the most substantial challenges of learning is the high dimensionality problem. If the dimension of the data is increased, the number of training data necessary for statistical work such as distribution estimation and data density raises exponentially. This is a problem that occurs in generative methods. In the discrimination approach, high dimensions also lead to the reduction of the dimensional effect on the separation of samples and the actual distance between data is not considered. The learning algorithm can operate in the same condition and the mentioned problem is not created for it [7].

According to the importance of the issue, various algorithms based on the semi-supervised learning have been provided. In fact, semi-supervised algorithms are continually developed. Nowadays, applying other human features in learning is considered as another technique for machine learning and artificial intelligence. The human brain has the ability to learn and they have a glimpse of the far and near past in decision-making. Positive and negative memories and experiences can have a huge impact on decision-making. Scientists have sought to implement this idea in their algorithms. Some of the introduced algorithms in the field of machine learning have been based on this approach. Learning methods that make their decisions according to the past are called memory-based learning. Indeed, this learning has been developed and has had a great deal of impact on the achieved results [8].

The aim of this paper is to introduce the diverse models presented in the semi-supervised learning and then compare them due to their application. In Sect. 2, the various models of semi-supervised learning and their working procedure are generally reviewed. Then the advantages and disadvantages of each algorithm are examined. Semi-supervised approaches which are based on deep-learning techniques are investigated in Sect. 3. To determine the development process of memory-based algorithms in machine learning, Sect. 4 considers the memory and different memory-based learning approaches. Today, these methods are developed based on deep learning and we compare them in this article. Finally, some of the future works on memory-based learning and deep learning are proposed.

2 Semi-supervised learning algorithms

Indeed, the use of unlabeled data is equivalent to data distribution learning and any learning process requires a prior knowledge for convergence [5]. Accordingly, different models are presented for the semi-supervised learning. These models are different due to the applied topics and data types. The goal of proposing a semi-supervised learning method is to improve the learning results and solve the various problems based on data types. Over the years, each model has its own characteristics, benefits and drawbacks. An overview of these methods can help to identify their features as well as strengths and weaknesses. On the other hand, investigating and comparing the approaches can be remarkably helpful in solving future issues.

Recently, many algorithms have been developed for semi-supervised learning. Some of such methods were powered by supervised algorithms such as iterative methods, margin classifiers, graph-based methods, and aggregation methods.

2.1 Self-training methods

Self-training is the first iterative method for semi-supervised learning. In self-training, a class is trained with a small labeled data set first. Then a classifier is used to classify unlabeled data. After that, the most reliable unlabeled points are added to the training set along with predicted labels. The classifier is retrained. This process is iterated until the procedure meets termination conditions. Then the final classifier is given in the output. A problem of the self-training algorithm is that incorrect labeled samples are spread to the next iterations with great effects on results. Therefore, self-training procedures are required in every iteration for finding a criterion (metric) to select a set of highly reliable predictions [7].

Self-training is one of the primary patterns of repetitive methods for semi-supervised learning. In self-training, a classifier is trained with labeled data at first. Then this classifier is employed to assign a label to each of the unlabeled data and most trusted unlabeled points together with their predicted labels are added to the training set. The classifier is retrained and these procedures are repeated until it reaches the stop conditions. The last classifier is considered as output. One of the significant issues in the self-training algorithms is about the samples published to the subsequent repetitions with incorrect labels and has a strong impact on the results. Therefore, in each of the repetition of self-training processes, finding a metric for selecting a set with highly reliable predictions is necessary. If the reliability of prediction falls below a threshold, some algorithms try to avoid this problem by “de-learning” the unlabeled points [3]. Self-training has been applied to several natural language-processing tasks. For example, a self-training algorithm with a process involving two classifiers is proposed to classify conversations as “emotional” or “non-emotional”. The major problem of self-training is how to choose a set of highly reliable predictions.

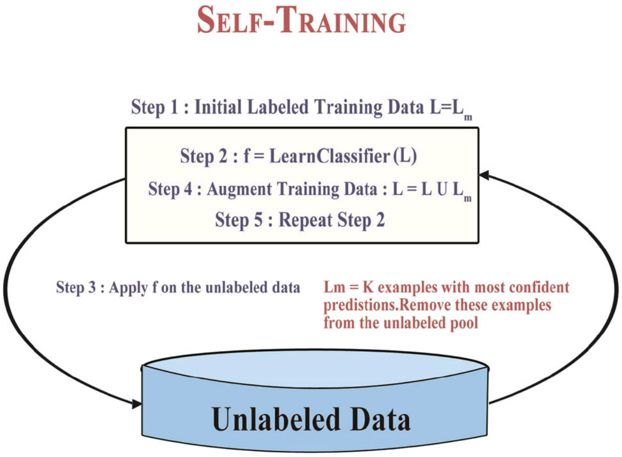

SELF-TRAINING
Step 1 : Initial Labeled Training Data L=L$_m$
Step 2 : f = LearnClassifier (L)
Step 4 : Augment Training Data : L = L U L$_m$
Step 5 : Repeat Step 2
Step 3 : Apply f on the unlabeled data
Lm = K examples with most confident predistions.Remove these examples from the unlabeled pool
Unlabeled Data

Fig. 1 Self-training structure

Figure 1 shows the structure of this method. From the figure, it is obvious that labeled data are first used for training. Then, unlabeled data are applied for learning step by step. At this stage, a label is assigned to each unlabeled data owing to the predictions and then they participate in learning. This process is repeated until the favorable condition is created [2].

Classification models perform better than the models trained only on labeled data. Joint training is one of the most successful iterative approaches to semi-supervised learning [9].

2.2 Generative models

This method employs unlabeled data for more accurate evaluations. Different models have been introduced for semi-supervised learning. Generative models include mixed Gaussian distribution, the EM algorithm, Bayesian distribution, hidden Markov model, and the Baum–Welch algorithm

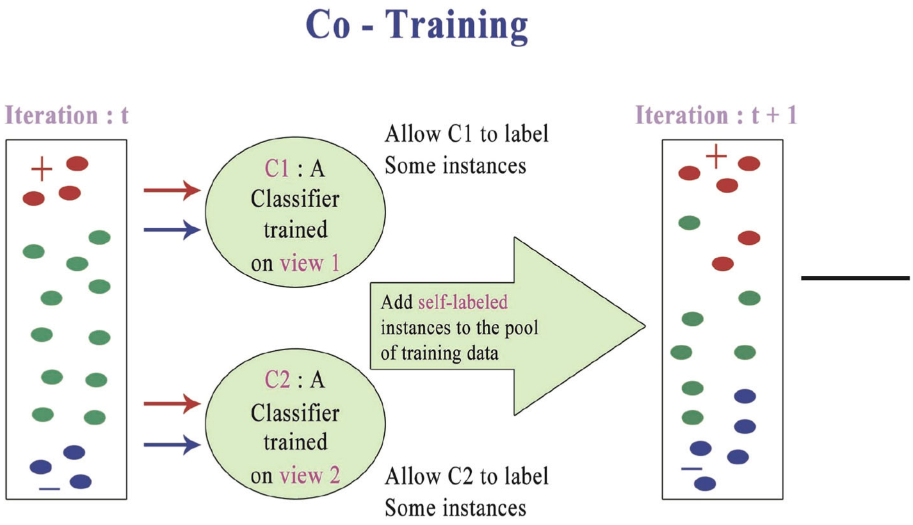

Co - Training
Iteration : t
C1 : A Classifier trained on view 1
C2 : A Classifier trained on view 2
Allow C1 to label Some instances
Allow C2 to label Some instances
Add self-labeled instances to the pool of training data
Iteration : t + 1
+
–

Fig. 2 The general co-training algorithm

[10]. Generative models are based on repetitive approaches. In this model, each category is estimated by a Gaussian distribution model and unlabeled samples are utilized to estimate distribution parameters. Nigam et al. [11]. employ EM algorithms with a mixture of polynomials using a repetitive approach to classify text. They demonstrated that the obtained classification models have more acceptable performance compared to the models which are trained only on labeled data.

Mutual training is one of the most successful repetitive approaches for semi-supervised learning [8]. Mutual training includes two preferable independent data views, both of which are adequate individually to train a classifier. Each classifier predicts a label and a degree of confidence for labeled data. Unlabeled samples which are labeled by a high reliable classifier are employed as training data for other samples. This procedure is repeated until any of the classifiers change. In mutual training, different learning algorithms can be used instead of the various data views that are rare in many domains. The main difficulty of repetitive methods is how to choose a set of high-reliability predictions. The agreement between classifiers is selected for mutual training which is not always useful and some of the incorrectly labeled examples are published for subsequent repetitions.

2.3 Co-training methods

Co-training is a machine learning algorithm used when a small amount of data are labeled, and a plethora of data are unlabeled. Co-training is a semi-supervised learning technique with two views. It is assumed that every sample is described using two sets of various features presenting different information on the sample. In an ideal case, these two views are conditionally independent (for instance, two feature sets of every sample are conditionally independent classifications). Every point of view is sufficient. Co-training learns a separate classification for every view using labeled samples. In this method, predictions of unlabeled data are used to iterate the creation of labeled training information [2]. The structure of this model is illustrated in Fig. 2. This model allows data to be labeled with data classification and start to learn by creating a powerful training data set.

2.4 Margin-based methods

Supervised margin-based methods are successful techniques for classification. Numerous studies were conducted to develop these methods in semi-supervised learning. Many of the margin-based methods are usually the expansions of a support vector machine (SVM) for semi-supervised learning [11]. An SVM uses a function minimizing training data errors and margin costs. To expand this semi-supervised learning ability; a waste function should be defined for unlabeled samples. The transductive support vector machine is one of the first attempts to expand SVMs on semi-supervised learning. In a standard SVM, there are only labeled data available for training, and the goal is to find a linear decision-making boundary with maximized margins to regenerate the Kernel Hilbert space [2]. Support vectors are a collection of points in the n-dimensional data space that specify the boundaries of the categories. The demarcation and classification of the data are conducted based on these vectors and by moving one of them, the output of the classification may change. In fact, support vectors are a kind of line in two-dimensional space, a page in three-dimensional space and they will form a hyper-plane in n-dimensional space. In SVM, only the data contained in the support vectors are the basis for learning the machine and making the model and this algorithm is not sensitive to other data points. Also, its goal is to find the best boundary between data in such a way that it has the greatest possible distance from all categories (their supporting vectors). This approach is one of the relatively new methods that in recent years has shown better performance over older methods of classification such as perceptron neural networks. The basis of SVM classifier is a data linear classification and in linear division of data [12], we try to select a line that has a more reliable margin. The optimal line for data is found by QP methods which are well-known techniques for solving constrained problems. Before the linear division, the machine can classify data of high complexity by transferring the data to a much higher dimension using the phi function [11]. To solve the high dimensionality problem utilizing these methods, we apply Lagrange duality theorem. In this theory, to convert the considered minimization problem into its dual form, a simpler function called kernel function is employed instead of a complex phi function which is a vector multiplication of phi function. Various kernel functions can be applied including exponential, polynomials and sigmoid.

In a standard SVM, only the labeled data are employed for training and finding a linear boundary with a maximum margin in the production of the Hilbert Kernel space is considered as a main goal. Owing to the fact that, a large data set of unlabeled data is used in semi-supervised learning, a TSVM must be considered to force the decision boundary to be placed in the low-density region in feature space by maximizing the boundary of the labeled and unlabeled data. Indeed, SVM is a kind of pattern recognition algorithm. SVM can be applied in pattern recognition and object classification problems. In Fig. 3 an example is depicted. A straightforward technique to do this and build an optimal classifier is to calculate the distance between the obtained boundaries and support vectors of each category (most borderline point of each category or class) and ultimately select the boundary that has totally the highest distance with available categories. In Fig. 3, the midline is an appropriate approximation of this boundary that has a high distance with both categories. This action of determining the boundary and selecting the optimal line can easily be accomplished by performing not-so-complicated mathematical calculus [2, 11].

2.5 Graph-based methods

Graph-based semi-supervised learning methods are based on the string theory. These methods define a graph in which nodes are samples (labeled or unlabeled), and edges indicate the similarities of samples. These methods usually assume the label evenness in the graph. Many of the graph-based methods are intended to estimate a function on a graph. Such a function should minimize waste on labeled samples and evenness on the entire graph. For this purpose, a match phrase is defined between class titles and their similarities. There are many methods for graph-based semi-supervised learning such as Markov random steps, label distribution, and Laplacian SVNs [13].

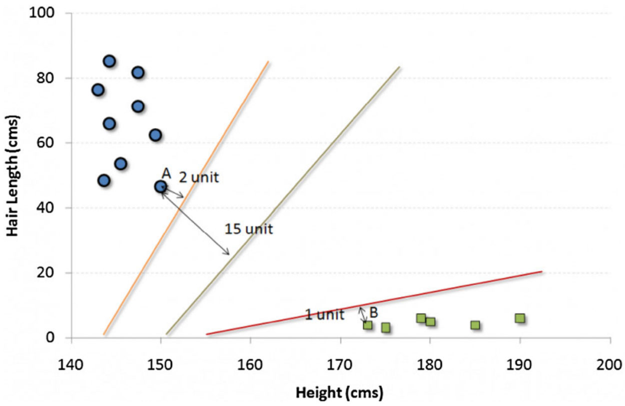


Fig. 3 Sample of SVM

2.6 Semi-supervised boosting

Boosting is considered as a supervised learning method with many applications. The goal of boosting is to minimize marginal costs. This method has also been developed for semi-supervised learning. One of such algorithms is COBoost, an expanded method based on boosting and co-training. The SemiBoost and RegBoost algorithms are other methods expanded in this area. These methods were mainly designed for binary classification problems; however, they cover many scientific scopes such as voice, object, and text recognitions in more than two classes [14].

2.7 Examples of semi-supervised algorithms

Section 2 presents various methods as well as different examples of semi-supervised learning algorithms developed based on those methods. These methods are mainly autonomous. Table 1 shows the algorithms. The algorithms that are superior to their previous algorithms are marked in red [15].

3 Deep learning

As a subset of machine learning, deep learning is based on a set of algorithms designed to model high-level abstract concepts in databases. The modeling process is completed using a deep graph with several processing layers including several linear and nonlinear transformations. To wit, it is based on learning the representation of knowledge and features in different model layers [16]. Deep learning has various applications in image classification, object identification, image extraction, semantic segmentation, gesture estimation, etc. [17].

Table 1 Examples of semi-supervised algorithms

Row	Name	Row	Name
1	Standard self-training	12	Co-training by committee: bagging
2	Standard co-training	13	Co-training committee: RSM
3	Statistical co-training	14	Co-training committee: tree-structured ensemble
4	Assemble	15	Co-training with relevant random subspaces
5	Democratic co-learning	16	Classification algorithm based on local clusters centers
6	Self-training with editing	17	Ant-based semi-supervised classification
7	Tri-training	18	Self-training nearest neighbor rule using cut edges
8	Tri-training with editing	19	Robust co-training
9	Co-forest	20	Adaptive co-forest editing
10	Random subspace method for co-training	21	Co-training with NB and SVM classifiers
11	Co-training by committee: AdaBoost	22	Co-training committee: tree-structured ensemble

Deep learning has been studied widely in recent years. Hence, numerous deep learning methods have been developed. These methods are generally classified into the following categories by their roots.

3.1 Convolutional method

Convolutional neural networks (CNNs) are among the most important deep learning methods, whereby several layers are trained with a powerful approach. This method is highly effective as one of the most common methods of computer vision. A CNN network is generally composed of three layers: convolutional layer, pooling layer, and fully connected layer. Each layer delivers a particular function. Moreover, in each convolutional neural network there are two stages of training: feed-forward and back propagation. In stage one the input image is fed to the network through the dot product of the input and each neuron parameter and the subsequent application of convolution to each layer. The network output is calculated thereafter. In the next step, back propagation starts based on the calculated error rate. In this phase, the gradient of each parameter is calculated based on the chain rule and all parameters are modified in proportion to their contribution to the network error. Afterwards, the feed-forward phase starts after updating the parameters. Network training completes after adequate iterations of these steps [18].

Table 2 lists an example of the algorithms developed based on this method [19].

3.2 Restricted Boltzmann machines (RBMs)

An RBM is a Boltzmann machine suffering from restrictions that arise from the formation of a bipartite graph by visible and hidden units [20]. This limitation leads to the development of more optimal training algorithms, especially the gradient-based contrastive divergence algorithm. Since this model is a bipartite graph, its hidden and visible units are also conditionally independent.

The deep Boltzmann machine (DBM) is another deep learning algorithm, wherein the processing units are wrapped in layers. As compared to the DBNs, in which two upper layers comprise an undirected graph and the lower layers form a directed generator model, DBM benefits from connections all over its structure. Similar to DBM, RBMs belong to the family of Boltzmann machines. The difference between these machines is that DBMs are composed of several layers of hidden units. The units in odd-numbered layers are conditionally independent of the even-numbered layers, vice versa [12].

3.3 Autoencoder

An autoencoder is an artificial neural network (ANN) designed for learning efficient coding. Each autoencoder serves to learn the compressed representation of a data set. In other words, autoencoders are generally used for dimensionality reduction. Each autoencoder is also composed of three (or more) layers [21].

Input layer: for example, in a face recognition operation, the neurons in the input layer can map the data to image pixels.

Hidden layer: several very smaller latent layers that account for encryption.

Output layer: each neuron in this layer is the same as each neuron in the input layer.

Autoencoders are generally used for dimensionality reduction or feature extraction. In a new structure, the symmetric multilayer autoencoder, which is different from the conventional autoencoders, is used to reduce dimensionality. This new structure has reduced the number of the required weights, diminishing the calculation costs significantly.

Autoencoders are, in fact, a special type of artificial neural networks used for optimal learning encoding. The autoencoder is trained to regenerate its input X instead of training the network and predicting target Y for input X. Hence, the output vectors have the same dimensions as the input vector.

Table 2 List of an example of the CNN algorithms

Row	Method	Year	Rank	Configuration	Advantages
1	AlexNet	2012	First	Five convolutional layers + three fully connected layers	Its important architecture directed the attention of many researchers to computer vision.
2	Clarifai	2013	First	Five convolutional layers + three fully connected layers	It made all internal network events and processes visible
3	SPP	2014	Third	Five convolutional layers + three fully connected layers	The limitation on the image size was removed through spatial pyramid pooling
4	VGG	2014	Second	13–15 convolutional layers + 3 fully connected layers	A thorough network assessment with incremental depth
5	GoogLeNet	2014	First	21 convolutional layers + 1 fully connected layers	Increased network depth and width without increasing the computational requirements
6	ResNet	2015	First	152 convolutional layers + 1 fully connected layer	Increased network depth and introduction of a method of preventing gradient saturation

3.4 Sparse autoencoder

Sparse autoencoders seek to extract sparse features from raw data. The sparsity of representations can be determined either by penalizing the hidden unit biases or by directly penalizing the hidden unit outputs. Sparse representations have several possible advantages [22]:

1. Similar to the SVM theory, the use of representations with large dimensions increases the odds of easy separation of different classes.
2. Sparse representations provide a simple segmented interpretation of the sophisticated input data.
3. Biological vision uses sparse representations in the basic areas of vision. A highly well-known example of sparse autoencoders is a nine-layer model with a local link to the pooling layer and contrast normalization. This model enables the system to train a face detector without labeling images as “with face” and “without face”. The resulting feature detector is significantly capable of scaling, translation, and out-of-plane rotation.

4 Semi-supervised learning models based on deep learning

There has been an increase in the use of deep learning to solve problems in recent years. In fact, it is used to analyze different levels of abstraction. Deep learning is used widely in different areas such as image classification (image recognition), text classification, and voice recognition [23]. In recent years, different models of this learning method have been presented to solve problems. The instances are as follows:

4.1 Deep learning model based on PixelRNN and DCGAN

A deep learning model was introduced by Daniel Fritz et al. for image classification. This model has been presented to solve semi-supervised problems. It combines a deep network with PixelRNN and DCGAN models for image recognition. Most of previous projects were based on labeled or unlabeled data in this area. PixelRNN and DCGAN models were implemented to classify handwritten pictures. This project benefited from transfer learning and semi-supervised learning to solve the problem. Transfer learning tries to obtain more features from labeled data with the help of deep learning. However, semi-supervised learning was employed to label unlabeled data. Another semi-supervised learning method is based on labeled and then unlabeled data [24].

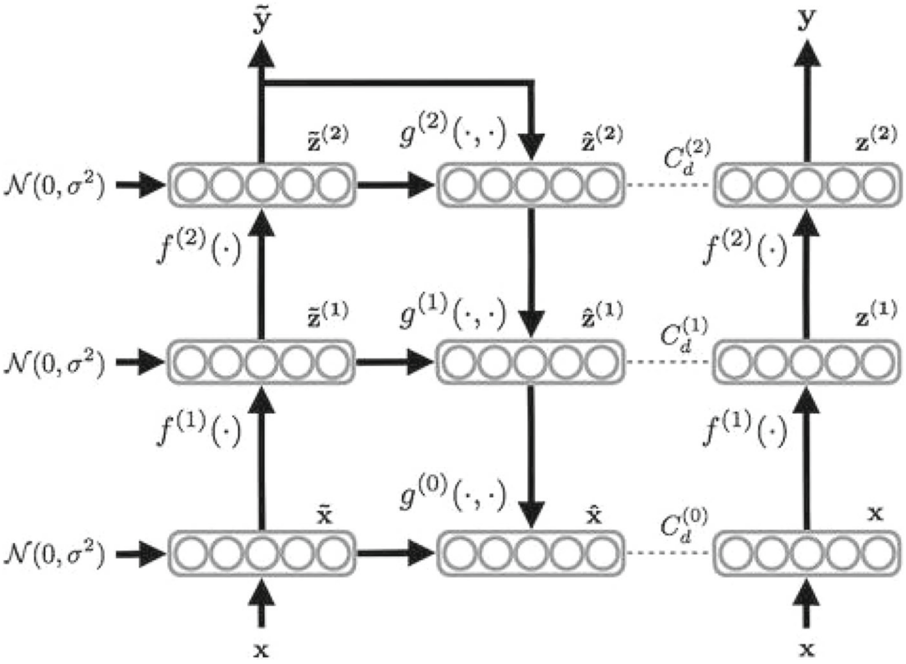
4.2 GAN model

This model is a generative and reparative network. This method is based on generative models in semi-supervised learning combined with deep learning. It is used to classify images. It has been designed to classify K classes. Salmeniz et al. designed this project based on a relatively semi-supervised environment able to identify fake data and classify them as one group [25].

4.3 Ladder model

This model is a hybrid of semi-supervised learning and deep learning. It was presented to avoid pretraining and minimize the costs of supervised and unsupervised learning. In

Fig. 4 The ladder model



this model, the most important technique is to delete the members of relatively low importance with fewer connections. This network consists of two coders and one decoder [26].

Figure 4 indicates the structure of this network compared with a feedback network. According to the results, reconstruction costs of this method are lower. A principle of this method is side connection. The hybrid method can have better effects on performance and reduce the error rate. Comparing different selections of architecture helps learning improve its meaning [27, 28].

4.4 ADGM model

The auxiliary deep generative model (ADGM) is another project conducted on semi-supervised learning. In this model, a set of various coders use additional random models to create a decoder. These variables result in more flexibility and can improve classification. Two generators are defined in this method. One is meant for feature extraction, and the other one is responsible for relative supervision. Then the method starts classification. This project was evaluated with MNIST dataset. According to the results, AtlasRBF was 1.5 times more convergent than DGM, Virtual Adversarial, and Ladder [29, 30].

Another model of hierarchical variables was introduced in 2015. In that model, auxiliary variables helped predict the system greatly. Kingma et al. introduced a probabilistic approach to semi-supervised learning by collecting probable features.

5 Models based on MBNNs

Neural networks can solve problems according to their learning ability without having to write the program [31]. On the other hand, one of the strategies in solving problems is based on memory-based recovery [32]. This structure is formed using feedback and time delay. The concept of memory in neural networks is used in two forms. In the first form as the weight combination raises the neural network as the producer of different outputs for different inputs, the neural network can be designed as a memory element. The second mode is the place of dynamic problems where the previous states of a system are needed to determine the status of a system.

So far, several models have been presented based on these networks. Among these approaches the following are mentioned:

5.1 Recurrent neural network (RNN)

The common neural networks machine-learning specialists used did not have memory capability and could not work as human beings, i.e., they had no knowledge of the past. It was a major disadvantage for these networks. RNNs were designed to address this problem [33]. In fact, RNNs have a recursive loop that makes it possible to use the information we have obtained from the previous moments in the network [34]. The figure below shows the structure of this network.

These networks have many applications in various areas, including voice recognition, modeling language, translation, auto explanation insertion for image, and so on.

5.2 LSTM network

RNNs have the ability to learn based on short-term time. Various models were proposed to solve this problem. One of the most powerful networks in this regard is LSTM network. Overall, this network is used where the distance between relevant information and where this information is needed is high. The LSTM networks—"Long Short Term Memory"—are a special type of RNNs capable of learning long-term dependencies [35]. These networks were first introduced by Hochreiter and Schmidhuber [36]. In fact, the purpose of designing LSTM networks was to solve the long-term dependency problem. Note that remembering information for long periods is the normal and default behavior of ordinary LSTM networks, and their structure is such that they learn very far information well, which lies in their structure. All RNNs are in the form of repetitive sequences of neural network modules. In standard RNNs, these repeatable modules have a simple structure [36].

5.3 Neural turing machines

Neural Turing Machine (NTM) architecture has two basic components: a controller neural network and a memory. Figure 1 provides a high-level graph of NTM. As many neural networks, the controller communicates with the outside world through the input and output vector. The difference between these networks and standard ANNs is based on interaction with a memory matrix [37]. Reading and writing operations interact with the help of a head of memory elements. As the interaction with memory is very scattered, data storage is unconcentrated to the memory location [37]. In these networks, the heads can focus on one location of memory or have less focus.

5.4 Meta learning

Despite the recent advances concerning neural networks, one of the challenges is "one-shot learning." Old traditional gradient networks need a lot of data to learn [38].

When they face new data, the models must relearn their own parameters. New architectures based on Turing machines present this ability to eliminate abnormalities by speeding up new data collection and retrieval. This is done with the help of a memory added to the network. In fact, this method seeks a solution to strengthen the memory of a nerve to accelerate data collection and data-based prediction [39]. This method focuses on an external memory management focused on memory content: methods that consider the mechanism of memory storage besides storage location.

5.5 Differential neural computer (DNC) external memory learning

The position of ANNs in the sensory processing, sequential learning and reinforcement learning has been significantly stabilized, but their efficiency has always been limited due to the inability to display variables, data structure storage, and the long-term storage of data due to lack of memory. Like a regular computer, it can use its memory to display and manipulate complex data structures, but can at the same time learn from data like the neural network [40]. While we learn, with monitoring, we show that DNC can successfully respond to artificial questions designed to emulate natural reasoning and deduction in the natural language. This model can perform tasks such as finding the shortest path between specific points and find the links lost in randomly generated graphs, and then extend these tasks to specific graphs such as transport networks and the school. During reinforcement learning, DNC can complete the moving parts puzzle where the change of objectives is determined by the sequence of symbols. Overall, the results show that DNC is capable of performing complex and constructed tasks that cannot be accessed by neural networks without external read–write memory [35].

6 An analysis of semi-supervised learning models

Regarding machine learning, labeled data are very hard to access, and unlabeled data are usually collected and accessed easily. On the other hand, most of data are unlabeled in many projects, and only some data are labeled. Therefore, semi-supervised learning is more practical in most of the problems.

6.1 Comparing semi-supervised learning models

No comprehensive model is presented in machine learning that can provide suitable answer for all fields, and the existing models are presented based on the type of the problem. Table 1 shows advantages and disadvantages of each model separately, but direct comparison is not possible because of the types of models. Many models have been presented for machine learning. Table 3 shows the advantages and disadvantages of these methods separately.

6.2 Comparing semi-supervised learning models

Deep neural networks have performed well in a wide range of tasks. An instance of such projects is image detection. Table 4 indicates semi-supervised models based on deep learning.

According to Table 4, most of the proposed models were supposed to obtain better features from the data of labeled semi-supervised data.

Table 3 Comparing the advantages and disadvantages of semi-supervised learning methods

No.	Method	Advantages	Disadvantages
1	Self-training (iterative method)	It is the simplest semi-supervised learning algorithm It can be used in most of the classifications	Errors can strengthen It cannot provide much information on convergence
2	Generative models	They can make good predictions of models which are close to solutions They can provide a knowledge of data structures or problems	They are not good for classification problems They encounter problems in balancing labeled and unlabeled data when there is a small number of labeled data The local optimization algorithm is of EM type They are prone to errors Unlabeled data can damage model detection
3	Co-training	It can be used in different methods of classification The error rate is lower than that of the self-training method	It may not be able to separate indices.
4	Graph-based methods	It is based on a mathematical framework It will perform well if the graph matches It can be applied to directed graphs	It will produce the worst output if the graph does not match The performance is vulnerable to the graph structure and edge
5	Vector-based method	It is highly validated.	The optimization of local optimum can be problematic

Table 4 The features of deep semi-supervised learning models

No.	Deep semi-supervised models	Features
1	PixelRNN, DCGAN	Image detection With two methods based on transfer learning and deep learning Labeling unlabeled data
2	GAN	Classifying K-class images Identifying fake data and classifying them as one group
3	Ladder	This model avoids pretraining It deletes the members of relatively low importance with few connections
4	ADGM	It uses additional random variables It increases flexibility and improves classification It is 1.5 times more convergent than Ladder Image application
5	Hierarchical variables	It is based on auxiliary variables It is based on probability

6.3 Comparison of memory-based neural networks models

NTM is from the primary DNC generation. This machine used architecture similar to a neural network controller and read–write access to the memory matrix, but it differed in terms of the mechanism of access to the interface memory [41]. In NTM, content-based addressing is combined with position-based addressing and allows the network to repeat through its memory locations and their indices (e.g., position n, then $n+1$, and so on). Thus, the network can store and retrieve time sequences in continuous memory blocks. However, this method has many disadvantages. First of all, NTM has no mechanism to ensure that the allocated memory blocks do not overlap and interfere, which is one of the major problems in computer memory management. There is no interference in the allocation of DNC memory, and this also provides free opportunities at a time, which is not associated with index or inventory and does not need interconnect blocks. Secondly, NTM has no way to free up written positions and thus has no way to reuse memory after processing long sequences. This problem has been solved in DNCs with free paths to free allocation. Thirdly, the ordinal data in the NTMs are maintained as long as the repetition continues through successive positions; when writing head goes

Table 5 Comparing the advantages and disadvantages of memory-based methods

No	Network type	Advantages	Disadvantages
1	RNN	Using the information of the moments before	Lack of access to the distant past
2	LSTM	The distance between relevant information and where this information is needed is high Simple architecture	Need for high memory requirement
3	NTM	Based on Turing machines Use of matrix	Not return to the back Focus on location
4	Meta learning	Recognizing and retrieving new information Eliminating abnormalities	Focus on data
5	DNC	Solving complex problems Return to previous memory weights Memory separation Focus on content and location	Implementation complexity

to a different section of memory, the order of writing before and after the jump cannot be retrieved by reading head [30]. When used by DNC, link matrix does not have this problem because it scans the writing sequence. Table 5 shows the advantages and disadvantages of these methods.

7 Ensemble learning

Ensemble learning classifiers are multi-component classifiers aiming at providing a better performance than a single-component one. To achieve better results, combined classifications are used in these methods. If the same functions are used in these classifications, indeed, their combination will not have an effect on the improvement of the performance.

Therefore, the combined techniques differ in how they create or make various classifiers, and how they combine basic classifiers with regard to their weights [2].

A combined technique for the classification problems consists of the following:

- Training set: a training set of labeled samples that are used for training.
- Basic learner: a basic learner is a learning algorithm used to learn the training set.
- Generator: this component is used to create different classifications.
- Combiner: it is used to combine the classification techniques.

There are basically two types of hybrid frameworks for creating combinations: dependent (ordinal) and independent (parallel). In a dependent framework, the output of a classifier is used to create the next classifier. Therefore, the knowledge created in the previous repetitions can be used to guide the learning in the future repetitions. Boosting is an example of this approach. In the second framework, i.e., dependent, each classifier is created independently and the outputs of all classifiers are combined with the voting methods. Bagging is a well-known example [2, 42].

7.1 Bagging

Bagging is a combined method that uses random subsets of the training set to provide hypotheses for its combinations in an independent manner. For each classification, a training set is created with random draft. In the re-sampling process, many of the original samples may be repeated in the obtained training set. Each independent classifier is created in combination with a different random sample of the training set. Since bagging makes a re-sampling with the substitution of the training set, each sample can be sampled several times [43]. Braiman showed that the bagging algorithm is effective for the “unstable” learning algorithms, i.e., small changes to the training set lead to bigger changes in predictions. Braiman claims that the neural networks and decision trees are examples of unstable learning algorithms [2]. Figure 5 shows the bagging model.

7.2 Boosting

The independent combined frameworks create various samples of classifiers by assigning weight to the training samples. These frameworks, such as boosting [2], are general frameworks for combined learning that create a combination of basic classifiers in an ordinal manner. Boosting is a general method to improve the performance of weak classifiers like classification rules or decision trees. This method works by running a linear learner repeatedly on the weighted training

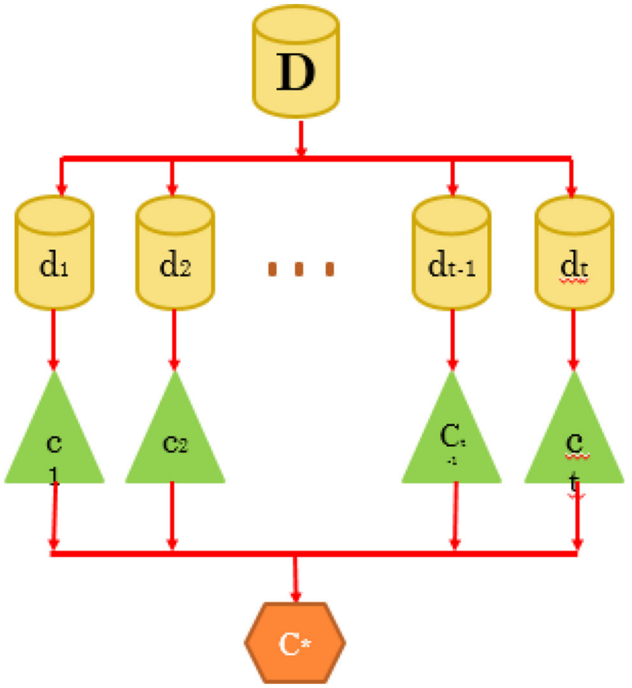


Fig. 5 Bagging model

samples. After many classifications, the created classifiers are combined with a final compound classification, usually making a better performance than a single classification. AdaBoost (adaptive boosting) is one of the first boosting algorithms that was introduced by Freund and Schapire [42]. The combined methods are generally among the most successful types of machine learning techniques. Research has shown that these methods are better than other algorithms in large-scale and high-dimensional problems [44, 45]. Figure 6 shows the general structure of this method.

7.3 Semi-supervised learning and ensemble techniques

Supervised learning methods are effective when there are sufficient labeled examples. However, since labeled examples require experimental research or annotations of an experienced human being, many applications, such as object recognition, classification of documents and web pages, and obtaining these samples are difficult, costly, or time-consuming. The semi supervised learning algorithms are used to create a classifier not only from the labeled data, but also from the non-labeled ones. The semi supervised learning algorithms aim at using non-labeled samples and also, combining non-labeled samples with the classification data of labeled samples to improve classification performance.

Although SemiBoost and RegBoost perform suitably in many areas, these methods are essentially designed for binary classification problems. However, many practical areas, such

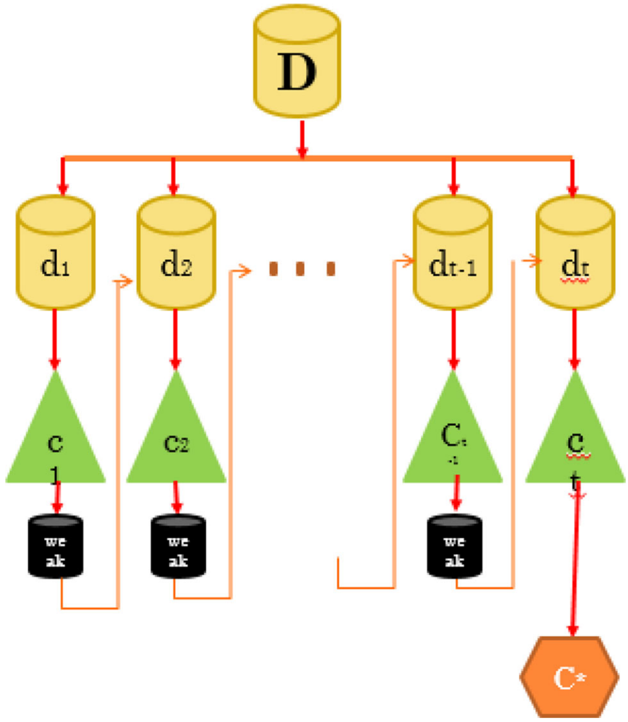


Fig. 6 Boosting model

as recognition of voice, objects, and text, include more than two classes. Recent algorithms for a semi-supervised multi-class learning problem are half-margin-based algorithms like SemiBoost and RegBoost to manage the multi-class case using the algorithm against all or similar meta-algorithms. This approach can have various problems, such as unbalanced class distribution, increased complexity, lack of guarantee for an optimal comprehensive classification, estimation of the probability and various scales of the outputs of the binary classifications, which make their combination more complex [2].

The major difficulties of a multi-class classification problem are:

- How to define the margin.
- How to design the loss function for the margin cost.
- How to determine the weight factor for data and classifications.

8 Semi-supervised learning frameworks

Framework is a software frame. A framework is a set of programming libraries and probably, a set of rules for programming. Various softwares and frameworks have been presented for semi-supervised learning up to now [46]. Some of these software have special applications or purposes. Some of these softwares are not self-confirmed but they are used by

Table 6 Semi-supervised learning frameworks

Row	Frameworks	Application
1	Aleph	Aleph is a multi-platform framework with different adjustments, various applications in graph and regularization
2	DUALIST	This tool is developed by Burr Settles and it is a utility for active learning with semantic terms capability
3	Gaussian process learning	Developed by Neil Lawrence and has applications in Gaussian processors
4	Harmonic function	Developed by Xiaojin Zhu and using MATLAB. It is a function for graph-based learnings
5	HSSR Hessian	Developed by Kwang In Kin, Florian Steinke and Matthias Hein in MATLAB and has applications in regression and dimensionality reduction
6	Java implementation	Developed by Smly in Java and has various algorithms
7	Junto	Developed by ParthaPratim in Java and it is applied in Gaussian random fields
8	Naive Bayes EM Algorithm 1.0.0	Developed by Rui Xia in C++ and it is applied in Nave Bayes classifier
9	David Andrzejewski	This algorithm is written by David Andrzejewski in C and it is applied in parallel semi-supervised learning

public because they are acceptable. Table 6 introduces some instances of these frameworks [45, 47, 48].

Each of the above-mentioned tools have different applications. Their differences are in the development language, application type and etc.

9 Data sets

The key to the development of genuine expertise in machine learning is practicing various machine learning approaches with different data sets. This is because in machine learning, every problem is considered a unique problem per se, which calls for a different unique strategy [15]. Some of these data sets are as follows.

- Mnist

It is a simple data set for machine vision. This data set consists of English handwritten images. It contains 70,000 image records classified in 10 groups. It also has 10,000 test data items with an average prediction rate of 92.20%.

- Wine Quality Data Set

The Wine Quality Data Set contains predictions about the regular wind qualities and it assesses the quality of each wind type against the chemical criteria.

This data set represents a multi-class classification problem. The distribution of the observations is not balanced in the data set. In addition, a total of 4898 observations with 11 feature columns and 1 label column comprise this data set. Finally, the mean RMSE error rate corresponding to this data set is 0.148.

- Pima Indians Diabetes Data Set

Pima Indians Diabetes Data Set consists of predictions made about diabetes in Indians within 5 years.

This data set is a two-class classification problem. The distribution of the observations is not balanced in the data set. Moreover, a total of 768 observations with 8 inputs and a label column (as the output) form this data set. It also contains the missing data marked with “zero”. The classification precision in this problem is approximately 65%, while the highest precision reported has been 77%.

- Sonar Data Set

The word “sonar” refers to a tracking device that works with sound waves. The sonar data set contains the information on the strength of the waves reflected from objects after they are radiated at different angles on objects.

This data set is a two-class classification problem and the distribution of the observations is not balanced in the data set. A total of 208 observations, 60 inputs, and one label column comprise this data set. The inputs consist of waves reflected at different angles, while the class column consists of two values: M (for mine) and R (for rock). The average prediction precision is 53%, while the best resulting level of precision has been 88%.

- Banknote Data Set

This data set is designed to detect fake notes. Hence, it is a two-class classification problem, and the distribution of the observations is not balanced in the data set.

It consists of 1372 observations, 4 input columns, and 1 output column. The mean precision of this problem is 50% (because a coin is flipped to determine whether the note is fake or not). Hence, you can increase the model precision as compared to the coin flip odds.

- Iris Flowers

This data set is one of the most well-known machine-learning data sets. It contains the information on various flower species. It is designed for three-class problems, and the distribution of observations is balanced species-wise.

A total of 150 observations with 4 input columns and 1 output column form this data set. The average precision of this problem is also 26% per class.

- Abalone Data Set

Abalone data set contains predictions about the age of clams. It is a multi-class classification problem, which can be converted into a regression problem through transformation, mirroring the beauty of this technique. There is a lack of class balance. It contains 4177 observations, with 8 input classes and 1 output class.

The precision corresponding to the largest class is 16%.

- Ionosphere Data Set

The ionosphere data set is designed to predict the ionosphere structure. The predictions are based on the reflection of radar waves that hit the free electrons in the ionosphere.

It is also an unbalanced two-class problem. It consists of 351 observations, 34 inputs, and 1 output column. In the class column, g denotes the “good state” and b denotes the “bad state”. Finally, the average prediction precision is 64%, while the best precision rate has been 94% so far.

- Wheat Seeds Data Set

The Wheat Seeds data set, as a balanced two-class classification problem, is used to predict various types of wheat seeds. It contains 210 observations, 7 inputs, and 1 output.

The prediction precision per label variable is also 28%.

- Boston House Price Data Set

The Boston House Price Data set consists of predictions about the house prices and their neighboring house prices expressed in thousand dollars.

It is a regression problem with 506 observations, 13 inputs, and 1 output column.

The mean RMSE for this data set is also 9.21 thousand dollars.

- Swedish Auto Insurance Data Set

The Swedish Auto Insurance Data Set contains the insurance information of automobiles in Sweden, which are expressed in terms of korona.

This data set suits regression problems. It contains 63 observations with a named input showing the number of insurance claims and a named output presenting the total prices paid for the insurance claims (in terms of thousand korona). The average standard deviation is approximately 72.251 thousand korona based on the RMSE values.

10 Conclusion

Today, diverse techniques and algorithms are introduced for machine learning. In fact, the main goal in learning is to provide better results. On the other hand, this type of learning has been modeled based on the human behaviors. Generally, people use their previous results and experiences in their decisions to achieve more desirable results. Indeed, employing memory for solving the problems is the main goal of learning techniques. Scientists are now seeking for a solution to implement memories in the system and improve the results. The proposed methods in this field are referred to as memory-based neural networks. Different solutions have been provided to address this challenge. In this article, a variety of solutions and their features are investigated. There is, of course, a lot to do to obtain the original purpose. In the future, better results can be achieved by combining strategies and developing existing methods.

A massive amount of data is dealt with in learning. It is not costly to obtain such data, although the knowledge of data is not obtained simply at a low cost. Four types of learning (supervised, semi-supervised, unsupervised, and reinforcement) were introduced to extract patterns from data. Regarding machine learning, labeled data are very hard to access, and unlabeled data are usually collected and accessed easily. On the other hand, most of data are unlabeled in many projects in which only some data are labeled. Therefore, semi-supervised learning is more practical in solving many of the problems. Different semi-supervised learning models are self-training, generative models, graph-based methods, and vector-based methods.

On the other hand, deep neural networks are used to extract more features from data using their multilayer structures. Different models were presented for this method of supervised data such as deep generative models, virtual adversarial, and ladder.

To empower these networks, various solutions have been used in recent years. One of the methods is providing memory for the network. This giving memory was done to learn more about networks. The first method presented in this topic was RNNs. After this, a solution was developed to further store LSTM algorithms and Turing machine-based networks. NTM architecture has two basic components: a controller neural network and a memory. The algorithm used a matrix network with forward storage. DNC was presented to solve the problem of this approach. The machine used architecture similar to a neural network controller and read–write access to the memory matrix, but differed in terms of the mechanism of access to the memory interface. Given the development of deep networks, it is possible to combine these methods with deep methods to strengthen those networks.

In semi-supervised learning, labeled data can greatly help extract patterns accurately. Therefore, they can result in more convergence by having more effects on models. A research strategy for future studies is to benefit from the memory to increase such an effect. The memory-based neural networks are new types of neural networks which can be used in this area.

References

1. Zhu, X.: Semi-Supervised Learning Literature Survey. Department of Computer Sciences University of Wisconsin, ICML, Madison (2008)
2. Tanha, J.: Ensemble Approaches to Semi-Supervised Learning. Library of the University of Amsterdam, Amsterdam (2013)
3. Wang, Y., Xu, X., Zhoa, H., Hua, Z.: Semi-supervised learning based on nearest neighbor rule and cut edges. Knowl. Based Syst. **23**(6), 547–554 (2010)
4. Sathya, R., Abraham, A.: Comparison of supervised and unsupervised learning algorithms for pattern classification. Int. J. Adv. Res. Artif. Intell. **2**(2), 34–38 (2013)
5. Alpaydin, E.: Introduction to Machine Learning. MIT Press, Cambridge (2004)
6. Seeger, M.: Learning with labeled and unlabeled data. Technical report, University of Edinburgh (2001)
7. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: bagging, boosting, and variants. Mach. Learn. **36**(1), 105–139 (1999)
8. Rezende, D.J., Mohamed, S., Danihelka, I., Gregor, K., Wierstra, D.: One-shot generalization in deep generative models. In: Proceedings of 33rd International Conference on Machine Learning (2016)
9. Nigam, K., Mccallum, A.K.: Text Classification from Labeled and Unlabeled Documents Using EM. Kluwer Academic Publishers, Boston (1998). **(manufactured in The Netherlands)**
10. Adiwardana, D.D., Matsukawa, A., Whang, J.: Using generative models for semi-supervised learning. Stanford reports (2017)
11. Basu, A., Watters, C., Shepherd, M.: Support vector machines for text categorization. In: Proceedings of the 36th Hawaii International Conference on System Sciences (2002)
12. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. Pattern Anal. Mach. Intell. IEEE Trans. **35**(8), 1798–1828 (2013)
13. Latouche, P., Rossi, F.: Graphs in machine learning: an introduction. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, 22–24 April 2015. arXiv:1506.06962v1
14. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors
15. Triguero, I., García, S., Herrera, F.: Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowl. Inf. Syst. **42**(2), 245–284 (2015)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR. arXiv:abs/1512.03385 (2015)
18. Zeiler, M.D.: Hierarchical Convolutional Deep Learning in Computer Vision. New York University, New York (2013)
19. Long, J., Shelhamer, E., Darrell, T. (eds.): Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2015)
20. Hinton, G., Sejnovski, T.: Learning and Re-learning in Boltzmann Machines, PDP. MTI Press, Cambridge (1986)
21. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A. (eds.) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning. ACM (2008)
22. Simoncelli, E.P.: 4.7 statistical modeling of photographic images. In: Handbook of Video and Image Processing (2005)
23. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXivpreprint arXiv:1207.0580 (2012)
24. van den Oord, A., Kalchbrenner, N.: Pixel recurrent neural networks. In: International Conference on Machine Learning, New York, NY, USA, 2016. arXiv:1601.06759v3 (2016)
25. Odena, A.: Semi-supervised learning with generative adversarial networks. In: Data Efficient Machine Learning Workshop, ICML 2016. arXiv:1606.01583v2 (2016)
26. Valpola, H.: From neural PCA to deep unsupervised learning. In: Advances in Independent Component Analysis and Learning Machines, pp. 143–171. Elsevier. arXiv:1411.7783 (2015)
27. Sadarangani, A., Jivani, A.: A survey of semi-supervised learning. Int. J. Eng. Sci. Res. Technol. (2016). https://doi.org/10.5281/zenodo.159333
28. Rasmus, A., Valpola, H., Honkala, M., Berglund, M., Raiko, T.: Semi-supervised learning with ladder networks, neural and evolutionary computing (2015). arXiv:1507.02672v2 [cs.NE]
29. Maaløe, L., Sønderby, C.K., Sønderby, S.K., Winther, O.: Auxiliary deep generative models. In: Proceedings of the 33rd International Conference on Machine Learning (2016)
30. Jensen, R., Shen, Q.: New approaches to fuzzy-rough feature selection. IEEE Trans. Fuzzy Syst. **17**(4), 824–838 (2009)
31. Arefiyan, F., Eftekhari, M., Shen, Q.: The 8th symposium on advances in science and technology (8thSASTech), 2013, Mashhad, Iran (2013)
32. Gallistel, C.R., King, A.P.: Memory and the Computational Brain: Why Cognitive Science will Transform Neuroscience, vol. 3. Wiley, New York (2009)

33. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6645–6649. IEEE (2013)
34. Das, S., Giles, C.L., Sun, G.-Z.: learning context-free grammars: capabilities and limitations of a recurrent neural network with an external stack memory. In: Proceedings of the Fourteenth Annual Conference of Cognitive Science Society. Indiana University (1992)
35. Gers, F., Schraudolph, N., Schmidhuber, J.: Learning precise timing with LSTM recurrent networks. J. Mach. Learn. Res. **3**, 115–143 (2002)
36. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997
37. Graves, A., Wayne, G., Danihelka, I.: Neural Turing machines. arXiv:1410.5401v2 [cs.NE] (2014)
38. Santoro, A., Bartunov, S., Botvinick, M.: Meta-learning with memory-augmented neural networks. In: Proceedings of the 33rd international conference on machine learning, New York, NY, USA (2016)
39. Jankowski, N., Duch, W., Grabczewski, K.: Meta-Learning in Computational Intelligence, vol. 358. Springer Science & Business Media, Berlin (2011)
40. Graves, A., Wayne, G., Reynolds, M.: Hybrid Computing Using a Neural Network with Dynamic External Memory. Publishers Limited, part of Springer Nature, Berlin (2016). https://doi.org/10.1038/nature2010
41. Jensen, R., Shen, Q.: Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches. IEEE Press and Wiley, New York (2008)
42. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
43. Kuncheva, L.I., Whitaker, C.J.: Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Mach. Learn. **51**(2), 181–207 (2003)
44. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to Boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997)
45. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using Gaussian fields and harmonic functions. In: The 20th International Conference on Machine Learning (ICML) (2003)
46. Settles, B.: Closing the loop: fast, interactive semi-supervised annotation with queries on features and instances. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1467–1478. ACL (2011)
47. Zhu, X., Lafferty, J., Ghahramani, Z.: Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions. In: ICML 2003 Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining (2003)
48. Zhu, X., Ghahramani, Z.: Learning from labeled and unlabeled data with label propagation. Technical report CMU-CALD-02-107, Carnegie Mellon University (2002)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.