Introduction

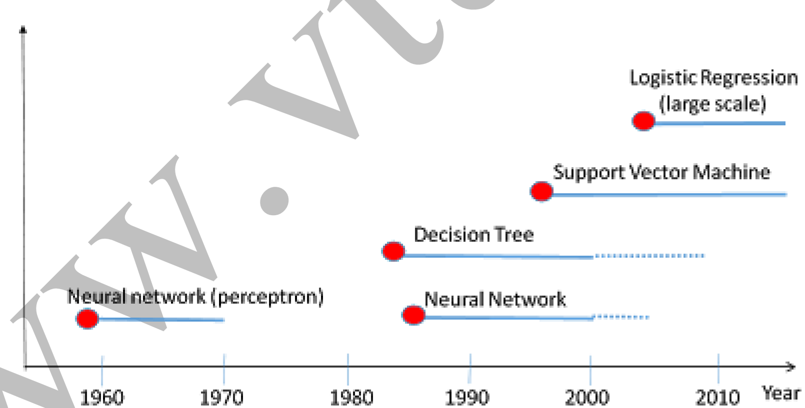
Artificial intelligence (AI) is the intelligence exhibited by machines or software, and the branch that develops machines and software with human-like intelligence. The goal of AI is to invent a machine which can sense, remember, learn, and recognize like a real human being. Perceptron is the first machine which can sense and learn but has fundamentally limited learning abilities. The later neural network with multiple hidden layers can learn more complicated functions but it lacks a good learning algorithm. The appearance of SVM enlightens people within a short time since it facilitates the learning procedures and performs well in many practical problems, but SVM also encounters its bottlenecks due to its shallow architectures. Deep learning is a learning method with the deep architecture and the good learning algorithms, which can perform the intellectual learning like learning the features. This ability, together with the efficient learning algorithms that can ensure this ability, point out a new direction toward.

In statistical machine learning, a major issue is the selection of an appropriate feature space where input instances have desired properties for solving a particular problem. For example, in the context of supervised learning for binary classification, it is often required that the two classes are separable by a hyper plane. In the case where this property is not directly satisfied in the input space, one is given the possibility to map instances into an intermediate feature space where the classes are linearly separable. This intermediate space cans either be specified explicitly by hand-coded features, be defined implicitly with a so-called kernel function, or be automatically learned. In both of the first cases, it is the user's responsibility to design the feature space. This can incur a huge cost in terms of computational time or expert knowledge, especially with highly dimensional input spaces, such as when dealing with images.

History of Machine Learning

The development of machine learning is an integral part of the development of artificial intelligence. In the early days of AI, people were interested in building machines that mimic human brains. The perceptron model was invented in 1957, and it generated over optimistic view for AI during 1960s. After Marvin Minsky pointed out the limitation of this model in expressing complex functions, researchers stopped pursuing this model for the next decade.

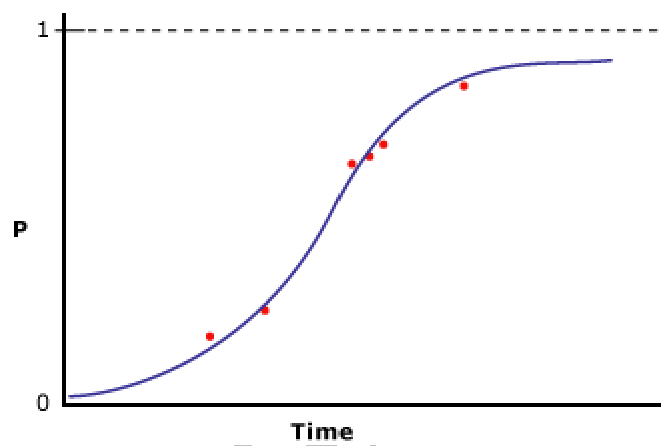
In 1970s, the machine learning field was dormant, when expert systems became the mainstream approach in AI. The revival of machine learning came in mid-1980s, when the decision tree model was invented and distributed as software. The model can be viewed by a human and is easy to explain. It is also very versatile and can adapt to widely different problems. It is also in mid 1980s multi-layer neural networks were invented, with enough hidden layers; a neural network can express any function, thus overcoming the limitation of perceptron.



Both decisions trees and neural networks see wide application in financial applications such as loan approval, fraud detection and portfolio management. They are also applied to a wide-range of industrial process and postal office automation (address recognition).

Machine learning saw rapid growth in 1990s, due to the invention of World-Wide-Web and large data gathered on the Internet. The fast interaction on the Intern called for more automation and more adaptively in computer systems. Around 1995, SVM was proposed and have become quickly adopted. SVM packages like libSVM, SVM light make it a popular method to use.

After year 2000, Logistic regression was rediscovered and re-designed for large scale machine learning problems. In the ten years following 2003, logistic regression has attracted a lot of research work and has become a practical algorithm in many large-scale commercial systems, particularly in large Internet companies.



We discussed the development of 4 major machine learning methods. There are other method developed in parallel, but see declining use today in the machine field: Naive Bayes, Bayesian networks, and Maximum Entropy classifier (most used in natural language processing).

Shallow Network

Shallow network is the network which contains few limited number of hidden layers. In Shallow network the degree of freedom is limited.

Eg: SVM

SVM is a type of non-linear training functions. Number of features considered for training is known as Degree of freedom. In SVM number of rows is considered instead of number of columns. Here the main disadvantage is that even though the degree of freedom is high, if the number of rows are less then the degree of freedom is also reduced. SVM is a non-recursive network.

Deep Architecture

In contrast to the shallow architectures like kernel machines which only contain a fixed feature layer(or base function) and a weight-combination layer(usually linear), deep architectures refers to the multi-layer network where each two adjacent layers are connected to each other in some way. “Deep architectures are compositions of many layers of adaptive non-linear components, in other words, they are cascades of parameterized non-linear modules that contain trainable parameters at all levels”.

According to the definition, people might intuitively tell that the greatest difference between the shallow architectures and the deep architectures are the number of layers. Then, people might ask several related questions as the followings:

1. How does deep architecture relate to the two core problems in Artificial Intelligence, representation and learning problems?
2. How do we compare the deep architecture with the shallow architecture? Or equivalently, what are the criteria’s of judging a certain kind of architecture?

Why do we need the deep architectures?

There’s an interesting history about people’s changes in their attitudes toward the deep architectures and the shallow architectures.

Perceptrons

In around 1960, the first generation of neural network was born(by Frank Rosenblatt). At that time, it was named Perceptron with one hand-crafted feature layer, trying to implement object recognition by learning a weight vectors combining all the features in some way. In brief, this early Perceptron only consists of an input layer, an output layer and a fixed hand-crafted feature in the middle. It’s capability of classifying some basic shapes like triangles and squares let people see the potential that a real intelligent machine which can sense, learn, remember and recognize like human-beings can be invented with this trend, but its fundamental limitations soon broke people's dreams. One of the apparent reasons is that the feature layer of this Perceptron is fixed and crafted by human beings, which is absolutely against the definition of a real "intelligent" machine. Another reason is its single-layer structure limits the functions it can learn.

Eg: an exclusive-or function is out of its learning ability.

Neural networks with hidden layers

In around 1985, based on the Perceptrons, Geoffrey Hinton replaced the original single fixed feature layer with several hidden layers, creating the 2nd-generation neural network. This neural network can learn the more complicated functions compared with the first Perceptions, via a famous learning algorithm called Back-propagation which back-propagates the error signal computed at the output layer to get derivatives for learning, in order to update the weight vectors until convergence is reached. Although the learning scope of these neural networks with the hidden layers is extended due to the multi-layer structures, it still has four main disadvantages as the followings:

1. Lacks the ability to train the unlabeled data while in practice most data is unlabeled;
2. The correcting signal will be weakened when it passes back via multiple layers;
3. Learning is too slow across multiple hidden layers;
4. It can get stuck in poor local optima.

When people were trying to make improvements to Hinton's neural networks with respect to those advantages, like increasing the training data set and estimating the initial weight values, another group of people made improvements on the original Perceptrons, creating a new family called Support Vector Machines(SVM). The kernel machines attracted most researchers' attentions, which slowed down the developments of the neural network.

Support Vector Machines(SVM)

SVM was first raised by Vladimir N. Vapnik and his co-workers in 1995, which adopts the core of statistical learning theory, turning the hand-crafted feature layer in the original Perceptrons into a feature layer following a fixed recipe. This recipe is called the kernel function(usually denoted as $K(x, x_i)$ or $\Phi(x)$), whose job is to map the input data into another high-dimensional space. Then, a clever optimization technique will be adopted to learn the weights combining the feature and the data, corresponding to the output. SVM makes learning fast and easy, due to its simple structures. For some certain kind of data with simple structures,

Eg, with a small number of features or the data which doesn't contain hierarchical structures,

SVM works well in many different AI problems, like the pattern recognitions. However, for the data which itself contains complicated features, SVM tends to perform worse because of its simple structure. Even if the kernel function converts the input data into a more complicated high-dimensional feature space, this procedure is still "fixed" because the kernel functions have already determined the mapping method, the information contained in the data structure is not fully used.

One way to solve this problem is to add a prior knowledge to the SVM model in order to obtain a better feature layer. This approach involves human intervention and is highly dependent on the prior knowledge we add, which takes us away from the road to a real intelligent machine. There are three reasons:

1. The prior knowledge is told to the model but not learnt by the model;
2. It's hard to find a general set of prior knowledge which can be applied to even the same kind problem, Eg, the model for hand-written digit recognition and the model for chemical molecule structure recognition need the different prior knowledge;
3. We cannot guarantee that the prior knowledge added to the model is correct, which might mislead the learning procedures and result in a poor model.

Although SVM uses the kernel functions instead of the hand-crafted features, although it uses a cleverer optimization technique instead of Back propagation which can deal with the unlabeled data, SVM is still a kind of Perceptron where the features are directly obtained but not learnt from the data itself. That is to say, despite the fact that SVM can work really well in solving many AI problems, it is not a good trend to AI due to its fatal deficiency, shallow architecture. Therefore, in order to move toward AI, we need an architecture which is capable of learning the features from the data given to it, as well as dealing with the unlabeled data.

An efficient and general learning algorithm must be able to apply to this architecture. Additionally, this architecture should be able to be used for solving different kinds of AI problems. With the purpose of finding an architecture that meets the requirements above, some researchers started to look back to the multi-layer neural network, trying to exploit its advantages related to "deep" and overcome the limitation.

Comparing the deep architectures with the shallow architectures

- A highly flexible way to specify prior knowledge, hence a learning algorithm that can function with a large repertoire of architectures.
- A learning algorithm that can deal with deep architectures, in which a decision involves the manipulation of many intermediate concepts, and multiple levels of non-linear steps.
- A learning algorithm that can handle large families of functions, parameterized with millions of individual parameters.
- A learning algorithm that can be trained efficiently even, when the number of training examples becomes very large. This excludes learning algorithms requiring storing and iterating multiple times over the whole training set, or for which the amount of computations per example increases as more examples are seen. This strongly suggests the use of on-line learning.
- A learning algorithm that can discover concepts that can be shared easily among multiple tasks and multiple modalities (multi-task learning), and that can take advantage of large amounts of unlabeled data (semi-supervised learning). "

From these features that a "good" architecture should have, we can tell that most of them are related to the learning algorithms, which doesn't contradict our efforts of designing a reasonable structures or modules of the architecture since learning algorithms are related to the architectures. To summarize, our goal is to design an architecture which

1. can learn prior knowledge from the input data;
2. have multiple layers where every two adjacent layers can be connected via non-linear steps;

3. have various trainable parameters;
4. can scale well;
5. can support different learning paradigms like multi-task learning, semi-supervised learning.

With these criterions, Bengio and Lecun's work made these three contributions:

- prove that "shallow architectures can be very inefficient in terms of required number of computational elements and examples";
- analyze a limitation of kernel machines with a local kernel;
- prove that deep architectures have the potential to generalize in non-local ways.

Deep Learning

Deep learning is a set of algorithms in machine learning that attempt to model high-level abstractions in data by using architectures composed of multiple non-linear transformations. Since 2006, deep structured learning, or more commonly called deep learning or hierarchical learning, has emerged as a new area of machine learning research (Hinton etc., 2006; Bengio, 2009). During the past several years, the techniques developed from deep learning research have already been impacting a wide range of signal and information processing work within the traditional and the new, widened scopes including key aspects of machine learning and artificial intelligence; see overview articles in (Bengio, 2009; Arel et al., 2010; Yu and Deng, 2011; Deng, 2011, 2013; Hinton et al., 2012; Bengio et al., 2013a), and also the media coverage of this progress in (Markoff, 2012; Anthes, 2013). A series of workshops, tutorials, and special issues or conference special sessions in recent years have been devoted exclusively to deep learning and its applications to various signal and information processing areas. These include:

- 2008 NIPS Deep Learning Workshop;
- 2009 NIPS Workshop on Deep Learning for Speech Recognition and Related Applications;
- 2009 ICML Workshop on Learning Feature Hierarchies;
- 2011 ICML Workshop on Learning Architectures, Representations, and Optimization for Speech and Visual Information Processing;
- 2012 ICASSP Tutorial on Deep Learning for Signal and Information Processing;
- 2012 ICML Workshop on Representation Learning;
- 2012 Special Section on Deep Learning for Speech and Language Processing in IEEE Transactions on Audio, Speech, and Language Processing (T-ASLP, January);
- 2010, 2011, and 2012 NIPS Workshops on Deep Learning and Unsupervised Feature Learning;
- 2013 NIPS Workshops on Deep Learning and on Output Representation Learning.
- 2013 Special Issue on Learning Deep Architectures in IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI, September).

- 2013 International Conference on Learning Representations;
- 2013 ICML Workshop on Representation Learning Challenges;
- 2013 ICML Workshop on Deep Learning for Audio, Speech, and Language Processing;
- 2013 ICASSP Special Session on New Types of Deep Neural Network Learning for Speech Recognition and Related Applications.

The authors have been actively involved in deep learning research and in organizing or providing several of the above events, tutorials, and editorials. In particular, they gave tutorials and invited lectures on this topic at various places. Part of this book is based on their tutorials and lecture material. Before embarking on describing details of deep learning, let's provide necessary definitions. Deep learning has various closely related definitions or high-level descriptions:

Definition 1: A class of machine learning techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification.

Definition 2: "A sub-field within machine learning that is based on algorithms for learning multiple levels of representation in order to model complex relationships among data. Higher-level features and concepts are thus defined in terms of lower-level ones, and such a hierarchy of features is called a deep architecture. Most of these models are based on unsupervised learning of representations."

Definition 3: "A sub-field of machine learning that is based on learning several levels of representations, corresponding to a hierarchy of features or factors or concepts, where higher level concepts are defined from lower-level ones, and the same lower-level concepts can help to define many higher-level concepts. Deep learning is part of a broader family of machine learning methods based on learning representations. An observation (Eg. an image) can be represented in many ways, but some representations make it easier to learn tasks of interest (e.g., is this the image of a human face?) from examples, and research in this area attempts to define what makes better representations and how to learn them."

Definition 4: "Deep learning is a set of algorithms in machine learning that attempt to learn in multiple levels, corresponding to different levels of abstraction. It typically uses artificial neural networks. The levels in these learned statistical models correspond to distinct levels of concepts, where higher-level concepts are defined from lower-level ones, and the same lower-level concepts can help to define many higher-level concepts."

Definition 5: "Deep Learning is a new area of Machine Learning research, which has been introduced with the objective of moving Machine Learning closer to one of its original goals: Artificial Intelligence. Deep Learning is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text."

Common among the various high-level descriptions of deep learning above are two key aspects:

1) models consisting of multiple layers or stages of nonlinear information processing; and
2) methods for supervised or unsupervised learning of feature representation at successively higher, more abstract layers.

Deep learning is in the intersections among the research areas of neural networks, artificial intelligence, graphical modeling, optimization, pattern recognition, and signal processing. Three important reasons for the popularity of deep learning today are the drastically increased chip processing abilities (e.g., general-purpose graphical processing units or GPGPUs), the significantly lowered cost of computing hardware, and the recent advances in machine learning and signal/information processing research. These advances have enabled the deep learning methods to effectively exploit complex, compositional nonlinear functions, to learn distributed and hierarchical feature representations, and to make effective use of both labeled and unlabeled data.

Active researchers in this area include those at University of Toronto, New York University, University of Montreal, Stanford University, Microsoft Research (since 2009), Google (since about 2011), IBM Research (since about 2011), Baidu (since 2012), Facebook (since 2013), UCBerkeley, UC-Irvine, IDIAP, IDSIA, University College London, University of Michigan, Massachusetts Institute of Technology, University of Washington, and numerous other places. These researchers have demonstrated empirical successes of deep learning in diverse applications of computer vision, phonetic recognition, voice search, conversational speech recognition, speech and image feature coding, semantic utterance classification, natural language understanding, handwriting recognition, audio processing, information retrieval, robotics, and even in the analysis of molecules that may lead to discovery of new drugs as reported recently by Markoff (2012).8

Three Classes of Deep Learning Networks

As described earlier, deep learning refers to a rather wide class of machine learning techniques and architectures, with the hallmark of using many layers of non-linear information processing that are hierarchical in nature. Depending on how the architectures and techniques are intended for use, e.g., synthesis/generation or recognition/classification, one can broadly categorize most of the work in this area into three major classes:

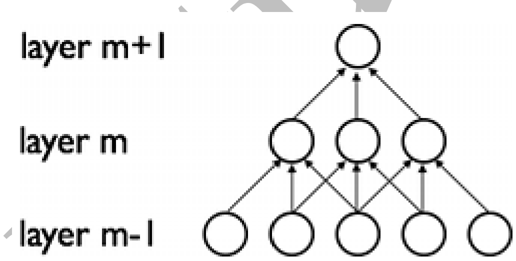
1) Deep networks for unsupervised or generative learning, which are intended to capture high-order correlation of the observed or visible data for pattern analysis or synthesis purposes when no information about target class labels is available. Unsupervised feature or representation learning in the literature refers to this category of the deep networks. When used in the generative mode, may also be intended to characterize joint statistical distributions of the visible data and their associated classes when available and being treated as part of the visible data. In the latter case, the use of Bayes rule can turn this type of generative networks into a discriminative one for learning.
2) Deep networks for supervised learning, which are intended to directly provide discriminative power for pattern classification purposes, often by characterizing the posterior distributions of classes conditioned on the visible data. Target label data are always available in direct or indirect forms for such supervised learning. They are also called discriminative deep networks.
3) Hybrid deep networks, where the goal is discrimination which is assisted, often in a significant way, with the outcomes of generative or unsupervised deep networks. This can be accomplished by better optimization or/and regularization of the deep networks in category 2. The goal can also be accomplished when discriminative criteria for supervised learning are used to estimate the parameters in any of the deep generative or unsupervised deep networks in category above.

Algorithms of Deep Learning

In this section, we will discuss the learning algorithms for the deep architectures. Since CNNs takes use of the methods which are classical in the field of signal processing, we will focus on the DBNs learning algorithms since it is more recent and leaves more space to be explored and improved.

Convolutional Neural Networks(CNNs)

CNNs adopt the Bacak-propogation to update the weights between every two adjacent layers. Therefore, one entire CNNs procedure helps to calculate the weight update. We should run this procedure several times until convergence is reached. Other improvements like using Fast Fourier Transform(FFT) algorithms to filter the input data or using max-poolig to sub sampling are related to either the convolution process or subsampling proces s. Most of them are based on the classical methods in signal processing.



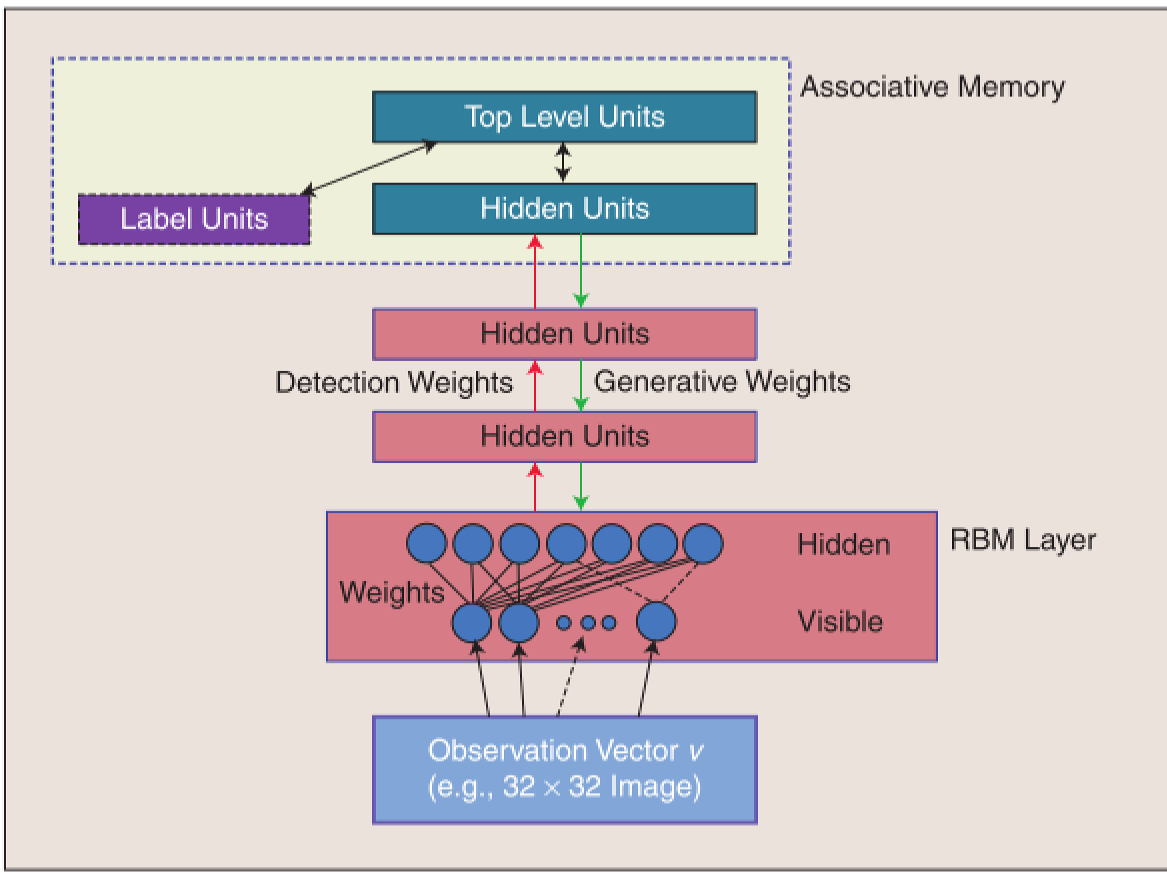
Deep Belief Networks(DBNs)

Hinton and his group raised five learning strategies for the multi-layer networks, where the last strategy is used in training a DBNs which was designed to allow higher-level feature detectors to communicate their needs to lower-level ones whilst also being easy to implement in layered networks of stochastic, binary neurons that have activation states of 1 or 0 and turn on with a probability that is a smooth non-linear function of the total input they received. The learning procedures of DBNs consist of the pre-training phase and fine-tuning phase.

We will discuss both of them respectively.

Pre-training

The pre-training phase is to treat the DBNs as a stack of Restricted Boltzmann Machines(RBM). That is to say, in figure the top two layers(associative memory) are also regarded as a RBM where the input layer is comprised of the hidden layer and the label unit layer.



As figure shows, RBM is an input layer(visible layer, denoted as v) plus a hidden layer(denoted as h) together with a weight vector(W) that links them. Learning RBM is to learn the W that maximize the log likelihood, which is approximately to maximize log p(v). We use a gradient ascent method to solve this maximization problem.

With contrastive divergence method and Gibbs-sampling, we can learn the weight vectors in one layer of RBM. Now, we need an algorithm to learn multi-layer RBM. Greedy Layer-Wise Training of Deep Networks .This learning approach is easy. We first start to learn one RBM(v, h1), then stack another RBM(h1 ,h2) where the sampled h1 via the learned weight W1 are treated as the visible input data in the second RBM, and use the same approach to learn the second RBM. This procedure goes on until all the layers are learned.

To summarize, the pre- training is a greedy layer-by-layer learning procedure. When we learn one single layer, we use a gradient ascent method to get the optimum weight vector, where the each weight update is calculated via Contrastive divergence and Gibbs-sampling. This pre-training is an unsupervised learning.

Fine-tuning

After the pre-training, the weights between every adjacent layer have the values that reflect the information contained in the data structure. In order to get a better performance, these weights need to be fine-tuned according to the model types.

Discriminative model If the pre-trained DBNs is used as a discriminative model, Back propagation is used to adjust the detection weights by supervised learning using the labeled data. There’s nothing new about the Back-propagation, the only thing we should pay attention to is that we should carefully set the learning rate since a too-big value will change the pre-trained weights a lot, but a too-small value will lead to a slow tuning procedure.

Generative model

If the pre-trained DBNs is used as a generative model, a wake-sleep algorithm is used to tune the generative weights. Therefore, wake stage is a “bottom-up” procedure tuning the generative weights. Tuning only once is not enough, so we also need the sleep stage which is similar to wake, but is “top-down” and tunes the detection weights. We iteratively apply the wake and sleep stages until the convergence is reached, which indicates the end of fine-tuning.

Applications

Deep learning has been applied to solve different kinds of problems. In this section, we list some of the application examples and summarize their work. In the field of visual document analysis [17], MNIST benchmark is used. Researchers in Microsoft Research claimed that by two policies, they can achieve the best performance for MNIST benchmark in image recognition problems. These two policies,

1. use elastic distortions to expand the MNIST to have more training examples.
2. use convolutional neural networks, are widely adopted in the image recognition later.

Karnowski and Arel apply deep spatiotemporal inference network (DeTSIN) to an image classification problem (MNIST benchmark, hand-written digit recognition). Their DeTSIN can learn both spatial information (relationship between adjacent pixels) and temporal information (same image at different time point). DeTSIN has good performance on MNIST benchmark. It is an interesting application about face recognition. They want to see if using unsupervised learning on deep networks will learn some very high level feature detectors for face recognition. They sampled tons of pictures (frames) from YouTube, which only small portions include faces. And then they designed a special deep architecture for image learning. After training on these unlabeled data, they test if any single neuron learned can be used as a good detector for face. Surprisingly, the best neuron can achieve 81.7% accuracy! After that, they apply a logistic classifier on top of the deep net they learned for supervised learning and find their results better than the state-of-art, which shows that the unsupervised learning of deep net can provide good high level features to improve the classification accuracy.

In the field of Human Action Recognition, developed a 3D convolutional neural network. The data are vedio pictures and there are three different actions: CellToEar, ObjectPut and Pointing. This 3D CNNs is compared with 2D CNNs and another two methods. For CellToEar and ObjectPut , 3D CNNs performs best. In pointing, 3D CNNs performed a bit worse. However, in average, 3D CNNs perform the best. All these applications solve the AI tasks well to some degree, compared with the traditional learning model. Therefore, we can say that deep learning has a good performance in solving complicated problems.

Future

From the analysis above, we know that deep learning represents a more intellectual behavior (learning features) compared with the other traditional machine learning. Architectures and the related learning algorithms are the two main components of deep learning. From the analysis above, we know that deep architectures like CNNs and DBNs perform well in many AI tasks.

- But is it true that only deep architectures can implement deep learning?
- Is it possible to implement deep learning without the deep architectures?

A recent work from by Cho and Saul who come from UCSD shows that kernel machines can also be used for deep learning. The approach they use is to apply multiple times of feature mapping to mimic the computation of deep learning. They apply this method to solve the image recognition problem, which performs better than the SVM with Gaussian kernel as well as the DBNs. This work gives us a new direction in exploring deep learning, which also indicates the fact that the deep architecture is proved to be a good model for the deep learning, but not the best one. There might be many surprises waiting for us to explore in this amazing field.

Conclusion

In the process of finding a road to AI, deep architectures like neural networks and shallow architectures like SVM respectively played the different roles in the different time periods.

SVM is attractive since there're many easy-to-implement optimization techniques to use, but it lacks the ability of learning features due to its shallow architecture; deep architectures like CNNs and DBNs can learn the features and perform better than SVM in complicated AI tasks like compute vision problem, but the learning procedures need to deal with many parameters.

- Deep architectures help deep learning by trading a more complicated space for better performance, in some cases, even for less computation time.
- Deep architectures are good models for deep learning, but can't be proved to be the best one. There're still many possibilities in the architectures and learning algorithms that can carry out better performances.

Although deep learning works well in many AI tasks, it works equally poorly in some areas as the other learning methods. Natural Language Processing(NLP) is a typical example; deep learning cannot understand a story, as well as a general request to an expert system. So there's still a long way to go before we can implement the real intelligent machine. But deep learning indeed provides a direction to implement the more intellectual learning; therefore it can be regarded as a small step toward AI.

REFERENCES

[1] Geoffrey E Hinton. A Fast Learning Algorithm for Deep Belief Nets. 1554:1527–1554, 2006.

[2] Yoshua Bengio and Yann Lecun. Scaling Learning Algorithms towards AI To appear in Large-Scale Kernel Machines,. (1):1–41, 2007.

[3] Yoshua Bengio. Learning Deep Architectures for AI, volume 2. 2009.

[4] Fabien Lauer. Incorporating Prior Knowledge in Support Vector Machines for Classification : a Review. 9(April 2007):1578–1594, 2008.

[5] Y LeCun and Y Bengio. Convolutional networks for images, speech, and time series. handbook of brain theory and neural networks, 1995.

[6] Itamar Arel, Derek C Rose, and Thomas P Karnowski. Deep Machine Learning A New Frontier in Artificial Intelligence Research. (November):13–18, 2010.

[7] Geoffrey Hinton. To Recognize Shapes, First Learn to Generate Images. 2006.

[8] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D Convolutional Neural Networks for Human Action Recognition. IEEE transactions on pattern analysis and machine intelligence, March 2012.

[9] Quoc V Le, Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang Wei Koh, and Andrew Y Ng. Tiled convolutional neural networks. Pages 1–9.

[10] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations.