



JOINT INSTITUTE
交大密西根学院

Deep Neural Network and Its Applications

VE370 Project 3

Bingcheng HU

516021910219

December 7, 2018

Contents

Deep Neural Network and Its Applications

VE370 Project 3

Contents

Abstract

1 Introduction

2 Background

2.1 Biological model and mathematical model of the neuron

2.2 Architectures: Restricted Boltzmann Machine

2.3 Architectures: Deep Belief Network

2.4 Architectures: Deep Convolutional Neural Networks

Convolution and sub-sampling

Pooling

3 Applications

3.1 Speech Recognition

3.2 Computer Vision and Pattern Recognition

3.3 Playing board games

3.4 Predicting game outcomes in Dota 2

4 Topics for future research

4.1 Design a depth model to learn from less training data

4.2 Optimization algorithms for adjusting network parameters

4.3 Unsupervised, semi-supervised and intensive learning methods applied to DNNs in complex systems

4.4 Implement deep learning algorithms on mobile devices

4.5 Deep Neural Network Stability Analysis

4.6 Application of deep neural networks in nonlinear networked control systems

5 Discussion

6 Conclusion

References

Abstract

Artificial Intelligence is a domain of computer science which is finding its applications in almost all domains of science and technology. Since the proposal of a fast learning algorithm for Artificial Neural Networks in 2006, deep learning technology has attracted more and more research interest because it overcomes the inherent ability of traditional algorithms to rely on the shortcomings of manual design features. Artificial neural network methods have also been found to be suitable for big data analysis and have been successfully applied to computer vision, speech recognition, playing board games and predicting game outcomes in Dota 2. In this article, we discuss some of the widely used artificial neural network architectures and their practical applications. This paper compared biological model and mathematical model of the neuron then lists three architectures: restricted Boltzmann machine, deep belief network and deep convolutional neural networks. Finally, a clear reason is given to a list of future research topics.

Key Words: Convolutional Neural Network, Deep Learning, Artificial neural network (ANN)

1 Introduction

The study of deep learning technology has attracted great attention, and the literature has reported a series of exciting results. Since 2009, ImageNet's competition has attracted numerous computer vision research groups from academia and industry. In 2012, the research team led by Hinton won the competition for ImageNet image classification through deep learning methods [3]. Hinton's team participated in the competition for the first time and their performance was 10% better than the second. Both Google and Baidu have updated their image search engine based on Hinton's deep learning architecture and have made significant improvements in search accuracy. Baidu also established the Deep Learning Institute (IDL) in 2013 and invited Stanford University Associate Professor Andrew Ng as the Chief Scientist. In March 2016, Google's deep learning program (called DeepMind) held a Go Game competition in South Korea between their AI player AlphaGo and Lee Se-dol [7], one of the world's most powerful players.

The purpose of this article is to provide a timely review and introduction on the deep learning technologies and their applications. It is aimed to provide the readers with a background on different deep learning architectures and also the latest development as well as achievements in this area.

2 Background

2.1 Biological model and mathematical model of the neuron

The brain is a highly complex, nonlinear and parallel information processing system. It has the capability to organize its structural constituents, known as neurons, so as to perform certain computations many times faster than the fastest digital computer in existence today, the image of it is shown at figure 1 (a).

A neural network is a machine that is designed to model the way in which the brain performs a particular task. Training inputs are applied to the input layer of the network, and the desired outputs are compared at the output layer. The difference between the output of the final layer and the desired output is back-propagated to the previous layers, just like figure 1(b) shows.

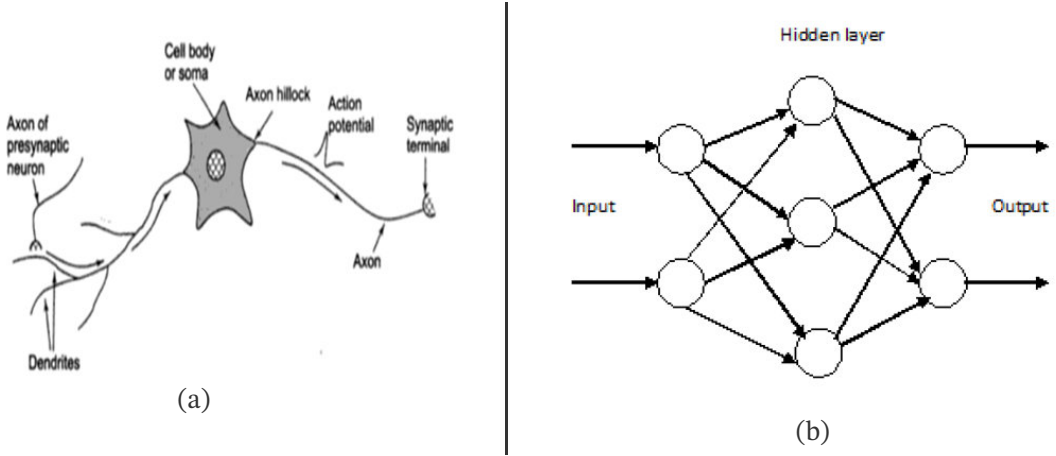


Fig. 1. Biological model of neuron(a) and architecture of Neural Network with hidden layers(b) [5]

2.2 Architectures: Restricted Boltzmann Machine

Algorithm 1 k-step contrastive divergence for RBMs

Input: $\text{RBM}(V_1, \dots, V_m, H_1, \dots, H_n)$, training period T , learning rate ϵ

Output: The RBM weight matrix ω , gradient approximation $\Delta\omega_{ij}$, Δa_i and Δb_j for $i = 1, \dots, m$, $j = 1, \dots, n$

Initialize ω with random values distributed uniformly in $[0, 1]$, $\Delta\omega_{ij} = \Delta a_i = \Delta b_j = 0$ for $i = 1, \dots, m$, $j = 1, \dots, n$

For $\forall t = 1 : T$ do,

 Sample $v_i^{(t)} \sim P(v_i | h^{(t)})$ when $i = 1, \dots, m$

 Sample $h_j^{(t+1)} \sim P(h_j | v^{(t)})$ when $j = 1, \dots, n$

 For $i = 1, \dots, m$, $j = 1, \dots, n$ do

$\Delta\omega_{ij} = \Delta\omega_{ij} + \epsilon \times [P(H_j = 1 | v^{(0)}) \times v_i^{(0)} - P(H_j = 1 | v^{(T)}) \times v_i^{(T)}]$

$\Delta a_i = \Delta a_i + \epsilon \times (v_i^{(0)} - v_i^{(T)})$

$\Delta b_j = \Delta b_j + \epsilon \times [P(H_j = 1 | v^{(0)}) - P(H_j = 1 | v^{(T)})]$

$\omega = \omega + \epsilon \times \Delta\omega_{ij}$

 End For

End For

Assuming that the difference between the model and the target distribution is small, we can use the samples generated by the Gibbs chain to approximate the negative gradient. Ideally, as the chain length increases, its contribution to the probability decreases and tends to zero. However, the estimate of the gradient does not represent the gradient itself. In addition, most CD components and corresponding log likelihood gradients have the same sign [9]. Therefore, a more practical algorithm called persistent contrast divergence is proposed in [3]. In this approach, the author suggests tracking the state of the persistent chain instead of searching for the initial value of the Gibbs Markov chain on a given data vector. After updating each weight, update the state of hidden and visible cells in the persistent chain. In this way, even a small learning rate does not create too much difference between the updated and persistent chain states, while at the same time leading to more accurate estimates.

2.3 Architectures: Deep Belief Network

As mentioned in the previous section, hidden and visible variables are not independent of each other [14]. To explore the dependencies between these variables, in 2006, Hinton built a DBN by stacking a set of RBMs. Specifically, the DBN consists of multiple layers of random and latent variables and can be considered a special form of the Bayesian probability generation model. DBNs are more efficient than artificial neural networks, especially when applied to problems with unlabeled data.

Algorithm 2 Greedy layer-by-layer algorithm for DBN

Input: Input visible vector v_{in} , training period T , learning rate ϵ , number of layers J .

Output: Weight matrix ω^i of layer $i, i = 1, 2, \dots, J - 2$.

Initialize ω with random values from 0 to 1, $h^0 = v_{in}$; where h^0 denotes the value of units in the input layer. h^i represents the units' value of the i th layer. layer = 1;

for $\forall t = 1 : T$ do,

 for $layer < L$, do gibbs sampling h^{layer} using $P(h^{layer}|h^{layer-1})$

 Computing the CD in Algorithm 1, $\omega(layer)$ is achieved using $\omega(t + 1) = \omega(t) + \epsilon \times \Delta\omega$

 End for

End for

The visible layer of the lowest RBM is first trained, with $h(0)$ as input. The values in the visible layer are then imported into the hidden layer, where the activation probability $P(h | \mu)$ of the hidden variable is calculated. The representation obtained in the previous RBM will be used as training data for the next RBM, and the training process continues until all layers are traversed. Since the approximation of the likelihood function is only required in one step in the algorithm, the training time has been significantly reduced. Bad configuration problems that typically occur in deep networks can also be overcome during pre-training. This pre-training algorithm is also called a greedy layer-by-layer unsupervised training algorithm. For the sake of clarity, we provide its implementation in Algorithm 2 [3].

2.4 Architectures: Deep Convolutional Neural Networks

In a TDNN, the weights are shared in a temporal dimension, which leads to reduction in computation.

The concept of CNNs is inspired by time-delay neural networks (TDNN). In CNNs, the convolution has replaced the general matrix multiplication in standard NNs. In this way, the number of weights is decreased, thereby reducing the complexity of the network. Furthermore, the images, as raw inputs, can be directly imported to the network, thus avoiding the feature extraction procedure in the standard learning algorithms. It should be noted that CNNs are the first truly successful deep learning architecture due to the successful training of the hierarchical layers. The CNN topology leverages spatial relationships so as to reduce the number of parameters in the network, and the performance is therefore improved using the standard backpropagation algorithms. Another advantage of the CNN model is that it requires minimal pre-processing.

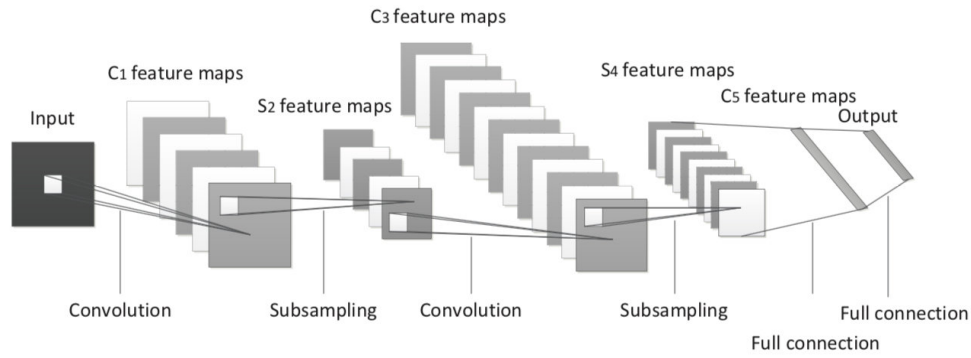


Fig. 2. Schematic structure of CNNs [3]

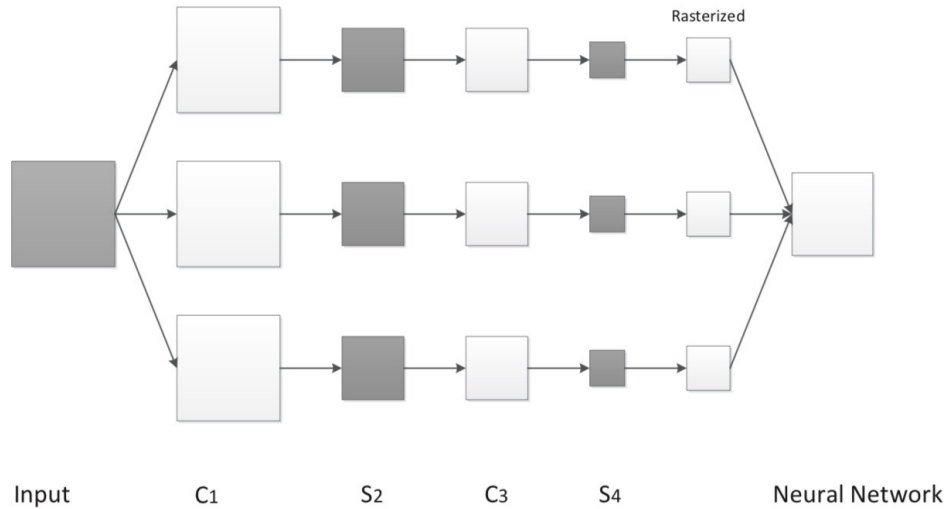


Fig. 3. Conceptual structure of CNNs Convolution [3]

Fewer parameters are required for CNN as compared to other traditional NN algorithms, which leads to reduction in memory and improvement in efficiency. The components of a standard CNN layer are shown in Figure 2, and a conceptual schematic diagram of a standard CNN is shown in Figure 3.

Convolution and sub-sampling

As shown in Figure 3, a CNN is a multi-layer neural network that consists of two different types of layers, i.e., convolution layers (c-layers) and sub-sampling layers (s-layers) [7]. C-layers and s-layers are connected alternately and form the middle part of the network. As Figure 4 shows, the input image is convolved with trainable filters at all possible offsets in order to produce feature maps in the first c-layer. A layer of connection weights are included in each filter. Normally, four pixels in the feature map form a group. Passed through a sigmoid function, these pixels produce additional feature maps in the first s-layer. This procedure carries on and we can thus obtain the feature maps in the following c-layers and s-layers. Finally, the values of these pixels are rasterized and displayed in a single vector as the input of the network [7]. Generally, c-layers are used to extract features when the input of each neuron is linked to the local receptive field of the previous layer. Once all the local features are extracted, the position relationship between them can be figured out. An s-layer is essentially a layer for feature mapping. These feature mapping layers share the weights and form a plane. Additionally, to achieve scale invariance, the sigmoid function is selected as the activation function due to its slight influence on the function kernel. It should also be noted that, the filters in this model are used to connect a series of overlapping receptive fields and transform the 2-D image batch input into a single unit in the output.

Pooling

Pooling is used to obtain invariance in image transformations. This process will lead to better robustness against noise. It is pointed out that the performance of various pooling methods depends on several factors, such as the resolution at which low-level features are extracted and the links between sample cardinalities. It is shown that better pooling performance can be achieved by learning receptive fields more adaptively. Specifically, utilizing the concept of over-completeness, an efficient learning algorithm is proposed to accelerate the training process based on incremental feature selection [7]. A stochastic pooling method to regularize large CNNs which is equivalent to introduce a stochastic pooling procedure in each convolutional layer. According to a multinomial distribution, the activation is randomly selected in each pooling region [7]. Moreover, since the selections in higher layers are independent of those in the lower ones, stochastic pooling is used to compute the deformations in a multi-layer model.

3 Applications

With rapid development of computation techniques, the GPU-accelerated computing techniques have been exploited to train CNNs more efficiently. Nowadays, CNNs have already been successfully applied to speech recognition, recommender systems, playing board games and predicting game outcomes in Dota 2.

3.1 Speech Recognition

During the past few decades, machine learning algorithms have been widely used in areas such as automatic speech recognition (ASR) and acoustic modeling [11]. The ASR can be regarded as a standard classification problem which identifies word sequences from feature sequences or speech waveforms. The standard architecture of an ASR system is given in Figure 4.

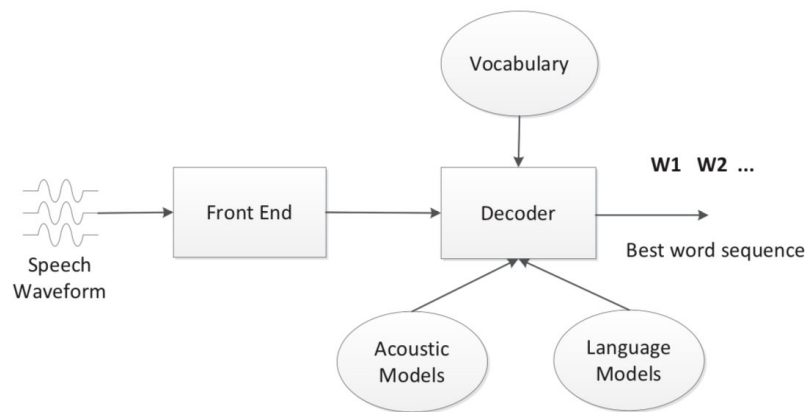


Fig. 4. Speech recognition system architecture [3]

Early applications of the deep learning techniques consist of large vocabulary continuous speech recognition (LVCSR) and phone recognition [11]. In 2015, the DNNs have been employed in automatic language identification (LID). Experiments have been carried out on short test utterance from two datasets: the Google 5 million utterances LID and the public NIST Language Recognition Evaluation dataset. More recently, a multi-task learning (MTL) method is proposed to improve the low-resource ASR using DNNs with no requirement on additional language resources [11].

3.2 Computer Vision and Pattern Recognition

During the past few years, deep learning techniques have achieved tremendous progress in the domains of computer vision and pattern recognition, especially in areas such as object recognition.

Computer vision aims to make computers accurately understand and efficiently process visual data like videos and images [8]. Conceptually, computer vision refers to the scientific discipline which investigates how to extract information from images in artificial systems.

Detection is one of the most widely known sub-domains in computer vision. It seeks to precisely locate and classify the target objects in an image. As Figure 5 shows, ANN can detect cars, roads, people, ect from the camera [13]. The usage of it is wide.

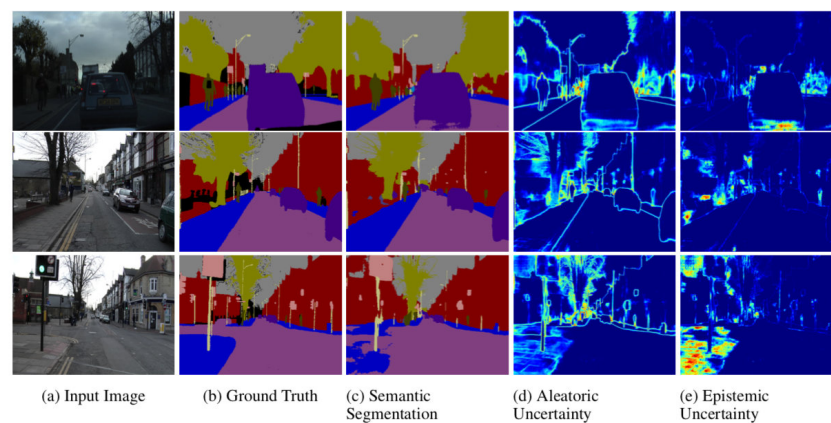


Fig. 5. Computer Vision and Pattern Recognition Usage [13]

DBNs are employed in the computer-aided diagnosis (CAD) systems for early detection of breast cancer [1]. And deep learning methods can also be applied to annotate genetic variants to identify pathogenic variants. ANN can not only predict the protein order/disorder regions but be used on remote sensing, medical diagnosis, disaster evaluation, and video surveillance. For traffic control and maritime security monitoring, ship detection on spaceborne images has been widely used [3].

3.3 Playing board games

Tic-tac-toe (also known as noughts and crosses or Xs and Os) is a pen and paper game for two players \times and \circ , which alternately mark space in a 3×3 grid. A player who successfully placed three markers on a horizontal, vertical or diagonal line wins the game [12].

In this experiment, each square in the large well was divided into 9 small squares and one small well. Initially, the first player selects 1 large square and places \times or \circ in one of the 9 small squares of the large square. Then the next player should choose a large square whose relative position is the same as the relative positive value of the small square of the well. The game is like this rule [6].

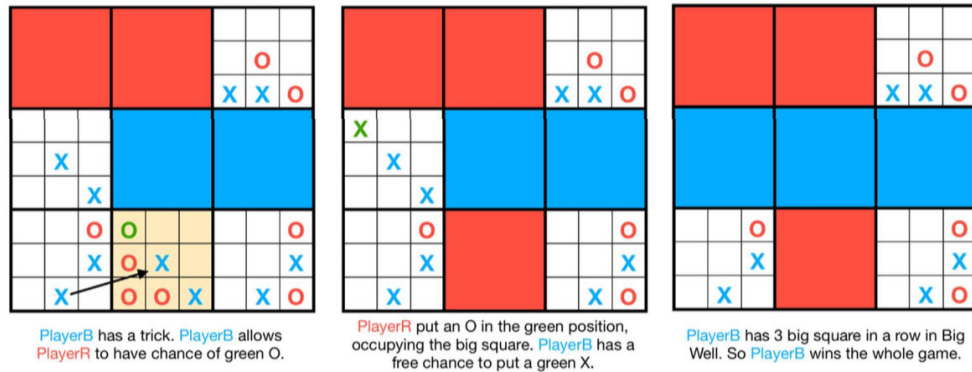


Fig. 6. Blue player wins a Tic-Tac-Toe in Big Well, and the whole game [6]

As shown in Figure 6, by entering the data set generated by the Monte Carlo tree search, the algorithm can automatically improve its board game skills. Because the architecture of the neural network is simple, the training efficiency is not very good. If more versions are developed, a better architecture can lead to better performance [6].

3.4 Predicting game outcomes in Dota 2

Dota 2 is an online strategy game, played in a five versus five format. Its multitude of selectable characters, each with a unique set of abilities and spells, causes every new match to be different from the last and picking the right characters can ultimately decide whether a team wins or loses a game [4].

For the ANN to be useful it needs to be able to learn. It does this with a technique called backpropagation that allows it to quickly update the weights between the neurons in the ANN. [8] The backpropagation algorithm utilizes the gradient ∂E of the loss function E in ∂w respect to any weight w . [4] After each epoch, the weights are modified so as to minimize the mean-squared error between the neural network's prediction and the actual target value.

Every ANN was trained for 5000 epochs using the training data. After each step of 1000 epochs we noted the current Mean Squared Error for the training set and tested the ANN. And the testing result is shown as Figure 7.

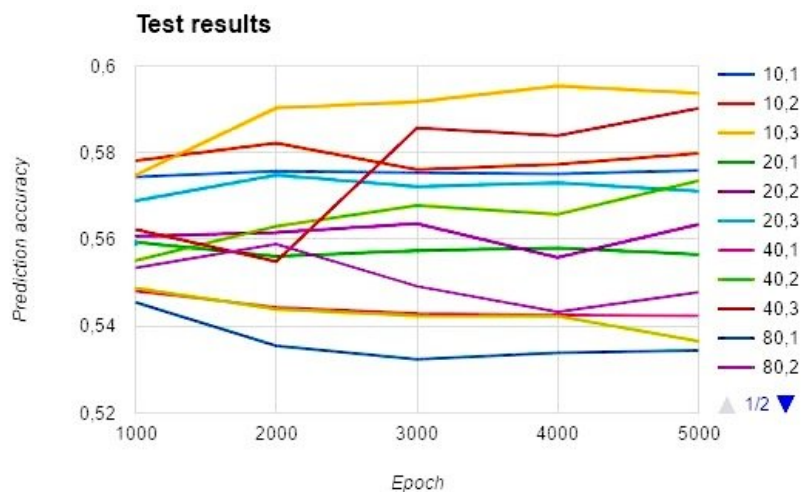


Fig. 7. Testing results after training for different models.[4]

The results show that all models have a prediction rate above 50%. The average accuracy for the models ranges between 53.44% and 59.54%.

4 Topics for future research

4.1 Design a depth model to learn from less training data

When only a limited amount of training data is available, a more powerful model is needed to achieve enhanced learning capabilities. Therefore, it is important to consider how to design a deep model to learn from less training data, especially for speech and visual recognition systems.

4.2 Optimization algorithms for adjusting network parameters

The method of adjusting parameters in machine learning algorithms is an emerging topic in computer science. In the DNN, a large number of parameters need to be adjusted. In addition, as the number of hidden nodes increases, the algorithm is more likely to fall into local optimum. [8]

4.3 Unsupervised, semi-supervised and intensive learning methods applied to DNNs in complex systems

Deep learning techniques have not achieved satisfactory results in NLP. With the development of deep unsupervised learning and deep reinforcement learning, we have more choices to train DNNs in complex systems.

4.4 Implement deep learning algorithms on mobile devices

It should be noted that deep learning methods, especially CNN, usually require a large computational burden. Recently, the idea of deep learning chips has emerged and has attracted a lot of research attention. Researchers at the Massachusetts Institute of Technology have proposed a chip for neural network implementation [3].

4.5 Deep Neural Network Stability Analysis

Dynamic neural networks have been widely used to solve optimization problems and are used in many engineering applications. Today, the stability analysis of deep neural networks has become a hot research topic because it brings many benefits to industry [10].

4.6 Application of deep neural networks in nonlinear networked control systems

Neural networks have been widely used to control engineering and signal processing to approximate nonlinear systems. On the other hand, NCS has been extensively studied so far. It is natural to apply deep neural networks to approximate nonlinear NCS with complex dynamics for better control/filtering performance [H. Chen].

5 Discussion

Since 2009, ImageNet's competition has attracted numerous computer vision research groups from academia and industry. In 2012, the research team led by Hinton won the competition for ImageNet image classification through deep learning methods [3]. Hinton's team participated in the competition for the first time and their performance was 10% better than the second. Both Google and Baidu have updated their image search engine based on Hinton's deep learning architecture and have made significant improvements in search accuracy. Baidu also established the Deep Learning Institute (IDL) in 2013 and invited Stanford University Associate Professor Andrew Ng as the Chief Scientist. In March 2016, Google's deep learning program (called DeepMind) held a Go Game competition in South Korea between their AI player AlphaGo and Lee Se-dol [7], one of the world's most powerful players.

Artificial Intelligence is a domain of computer science which is finding its applications in almost all domains of science and technology. Since the proposal of a fast learning algorithm for Artificial Neural Networks in 2006, deep learning technology has attracted more and more research interest because it overcomes the inherent ability of traditional algorithms to rely on the shortcomings of manual design features. Artificial neural network methods have also been found to be suitable for big data analysis and have been successfully applied to computer vision, speech recognition, playing board games and predicting game outcomes in Dota 2. In this article, we discuss some of the widely used artificial neural network architectures and their practical applications. This paper compared biological model and mathematical model of the neuron then lists three architectures: restricted Boltzmann machine, deep belief network and deep convolutional neural networks. Finally, a clear reason is given to a list of future research topics.

6 Conclusion

Human beings have experienced the same situation, starting from scratch, trying and summing up the experience to make better choices. However, Humans are not always strong enough to start from scratch and achieve further success. We must learn from the people of the past and walk farther on the shoulders of the giants. Since we can think for ourselves, we can create with previous experience. For a machine, it's powerful enough to explore on its own. Without human experience, it can achieve the same or even more achievements. Perhaps for a machine, deep neural network is a more appropriate approach to explore itself.

References

- [1] Kubat, M. (2015). Artificial neural networks. In *An Introduction to Machine Learning* (pp. 91-111). Springer, Cham.
- [2] Klerfors, D., & Huston, T. L. (1998). Artificial neural networks. St. Louis University, St. Louis, Mo.
- [3] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234, 11-26.
- [4] Widin, V., & Adler, J. (2017). On using Artificial Neural Network models to predict game outcomes in Dota 2.
- [5] Yadav, A., & Sahu, K. (2017). WIND FORECASTING USING ARTIFICIAL NEURAL NETWORKS: A SURVEY AND TAXONOMY. *International Journal of Research In Science & Engineering*, 3.
- [6] Chen, W. (2017). Using Neural Network and Monte-Carlo Tree Search to Play the Game TEN.
- [7] Deng, L. (2012). Three classes of deep learning architectures and their applications: a tutorial survey. *APSIPA transactions on signal and information processing*.
- [8] Zeng, N., Wang, Z., Zhang, H., & Alsaadi, F. E. (2016). A novel switching delayed PSO algorithm for estimating unknown parameters of lateral flow immunoassay. *Cognitive Computation*, 8(2), 143-152.
- [H. Chen] Chen, H., Liang, J., & Wang, Z. (2016). Pinning controllability of autonomous Boolean control networks. *Science China Information Sciences*, 59(7), 070107.
- [9] Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.
- [10] Hu, J., Chen, D., & Du, J. (2014). State estimation for a class of discrete nonlinear systems with randomly occurring uncertainties and distributed sensor delays. *International Journal of General Systems*, 43(3-4), 387-401.
- [11] Jaitly, N., & Hinton, G. (2011, May). Learning a better representation of speech soundwaves using restricted boltzmann machines. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on* (pp. 5884-5887). IEEE.
- [12] Kaplan, E. (1999). U.S. Patent No. 5,927,714. Washington, DC: U.S. Patent and Trademark Office.
- [13] Kendall, A., & Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision?. In *Advances in neural information processing systems* (pp. 5574-5584).
- [14] Yu, D., & Deng, L. (2011). Deep learning and its applications to signal and information processing [exploratory dsp]. *IEEE Signal Processing Magazine*, 28(1), 145-154.