Survey on Game AI based on Deep learning

Wangzhihui Mei 2019124044

CCNU-UOW JI

Abstract

This article provides an overview of the expected rapid development of deep learning technology and its application in-game artificial intelligence. It analyses the effectiveness of Deep learning techniques. Then the article gives analysis and comparison of methodologies of implementing game AI in four techniques: Convolutional Neural Network, Recurrent Neural Network, Supervised Learning and Unsupervised Learning.

1 Introduction

Great progress has been made in artificial intelligence technology in recent years. In 2016, Alphabet AlphaGo, a computer Go program developed by DeepMind, successfully defeated Li Shishi, who has been considered the world's top player for nearly 15 years. AlphaGo's success is extremely. This is mainly due to its use of deep learning algorithms[9]. This article introduces the application of deep learning in games from a broader perspective.

Deep learning is an machine learning algorithm showing effectiveness in recent years[16]. Its main method is to achieve better learning results by training multiple layers of neural networks. Researchers at the University of Montreal Analyze the help of unsupervised learning for deep structures [6] and the reasons for the failure of original training methods [2], and propose parameter initialization methods and activation functions [8] suitable for deep structures. Breaking through the bottleneck of training algorithms and computing power, deep learning is widely used in artificial intelligence-related Field[14, 19], and has made great progress on a number of research issues.

2 Deep Learning effectiveness

A brief overview of deep learning in game artificial intelligence is given in this section. In general, deep learning often performs well on prediction problems in the case of large scale dataset. We inferred 3 key points for the effectiveness of deep learning compared to traditional supervised learning models: non-linear learning, end-to-end learning and capability in the face of large-scale data.

2.1 Non-linear Learning

Deep learning builds a mapping function from the original data input to the final prediction target by stacking multiple layers of neural networks. Traditional machine learning models are good at finding linear transformations of data to output targets, but in reality, from data input to The mapping of target output is often complex and non-linear. In deep neural networks, each layer is a non-linear mapping from input to output. These multi-layer non-linear transformations form a hierarchical data depiction, As the number of layers increases, the representation is also more abstract and more invariant. For example, for image data, the deep learning structure first extracts the representation of the object boundary, and the subsequent network layers extract the relevant information from the representation of the boundary. The feature of the object part, and the feature of the object part is then used as the input of the next neural network layer(NNL) to obtain the feature vector of the object[12]; for text data, the deep learning structure first extracts the word feature representation, and then The network layer extracts the features of the sentence from the features of the word through non-linear transformation, and the features of the sentence are The feature is used as the input of the next NNL to obtain the feature vector of the article. Although theoretically, a neural network structure with more than two layers can express any function, but the deeper network structure is more efficient for a specific set of functions Expressive ability, which can express complex functions with fewer parameters[22]. This efficient expressiveness is necessary for complex artificial intelligence tasks. The expression of deep networks Advantages have obvious help for neural network learning: variables at the bottom and high levels can share statistical features, thereby improving learning efficiency. Because there is no such hierarchical feature representation in the two-layer neural network, we generally don't consider the two-layer The network is a deep learning structure. Similarly, non-linear models built directly on the original input space, such as decision trees and support vector machines (non-linear Kernel), are not considered deep learning models[1]

2.2 End-to-end Learning

The traditional machine learning process is often divided into multiple independent modules. Error propagation is not transmitted between these modules during the training process, so the previous module will not be adjusted according to the training results of the subsequent modules[15]. For example, in In a typical natural language processing problem, traditional methods will take separate steps such as tokenization, POS tagging, parsing, and semantic analysis. In such a process, Each step is an independent learning task, which requires massive training data with feature labeled. It is expensive to perform labelling these training samples for different learning tasks, and the wrong predictions made by each step (regardless of the cause) will also be Affects subsequent tasks. Even in a single learning task (such as syntactic analysis), feature extraction is often a preprocessing step independent of training. In deep learning processes, trainable hierarchical representations (Trainable hierarchical representation) replaces predefined features Representation, and the representation of each layer will be adjusted during training according to the error information transmitted by the subsequent layers, which is conducive to the optimization of the objective function[21]. On the other hand, in each layer of training in deep learning Does not depend on additional learning targets and training samples (Of course, deep learning can also use additional labels to provide a secondary objective function for the training of the middle layer). This not only allows limited resources to be preferentially used to label training samples for the final learning target , Also the flexibility to use ready-made labeled data sets. .

2.3 Scalability

Learning complex, non-linear functions, of course, requires more training samples. Too few training samples can lead to serious over-fitting problems. This not only makes how to obtain a massive labeled data a key, it is also important for machine learning. Faced with the scalability of a large number of training samples, a test was proposed. Although traditional machine learning algorithms can also be used to learn non-linear transformations, such computational complexity is very High: At least $O(N^2)$, N is the number of training data samples. When the amount of data is too large, this algorithm is obviously difficult. The training methods of deep learning are mostly based on stochastic gradient descent, not only need not be calculated The pairwise relationship of the training samples often does not even need to traverse all the training samples, so that a larger data set can be flexibly used. At the same time, the backpropagation algorithm can quickly calculate the gradient of the entire network and allow training errors Effectively propagate to the underlying feature space[11]. With the increase of training samples, the performance of traditional supervised learning algorithms The phenomenon of diminishing return appears. On the contrary, the effect of deep learning algorithms can be significantly improved with the increase of trainable data sets. This advantage comes from the flexible expression ability of deep network structure, so that deep learning can Express complex set of functions with simple structure.

3 Methodologies

3.1 Convolutional Neural Networks

Convolutional Neural Networks(CNN) are mainly used in computer vision problems. CNN handle the frames from video games. It consists of 3 layers: convolutional layer, pooling layer and fully connected layer. The convolutional layer is responsible for extracting local features in the image and is filtered by the convolution kernel to extract local features in the picture; the pooling layer is used to greatly reduce the parameter magnitude (dimensionality reduction); the fully connected layer is similar to the part of a traditional neural network and is used to output the desired result.

Before the emergence of CNN, images were a difficult problem for artificial intelligence for two reasons: The amount of data to be processed by the image is too large, resulting in high cost and low efficiency and it is difficult to retain the original features in the process of digitizing the image, resulting in low accuracy of image processing. The following explains these two issues in detail:

- Too much data to process;
- An image is made up of pixels, and each pixel is made up of color.

The first problem that CNN solves is "simplifying complex problems", reducing vast parameters into small-scale parameters, and then processing. More importantly: in most scenarios, the dimensionality reduction does not affect the results. For example, a 1000-pixel picture reduced to 200 pixels does not affect whether the picture is a cat or a dog, and the same is true of the machine.

Lungu et. al. [13] the demonstration of CNN that play games against human opponents in real-time with the result that the CNN achieved perfect performance. Stanescu et. al. [18] evaluated CNN's performance by comparing various search algorithms and found that CNN has better performance.

3.2 Recurrent Neural Network

Recurrent Neural Network(RNN) are "rememberable", they are mainly used in the domains of natural language processing (NLP) and speech. The RNN takes game states as inputs and outputs a probability value for each possible action. Unlike traditional neural networks, RNNs can use "sequence information". In theory, it can use information of any long sequence, but because of the "disappearing gradient" problem of the network structure, in practical applications, it can only use the information on time steps close to it.

The main idea of RNN is to introduce the concept of sequence. Here, the sequence can be either a time series or a spatial sequence, as long as it has a sequence relationship, it is applicable. Then, during forward propagation, the state of each moment is jointly determined by the previous state and the input of the current moment, thereby forming a network with a memory function, which has a significant effect in the field of natural language processing.

Sequence is the way RNN reads data, and it is also the most unique feature of RNN. Suppose we use the RNN network to build a "language model". The "language model" is actually to see if a sentence is smooth. It can be used in the field of machine translation and speech recognition. Generally, the length of each sentence is different. Each word is used as a training example, and a sentence is used as a minibatch. We give some derivations for the principle of the definition. According to the full probability formula, the probability of a sentence being "naturalized sentence" is:

$$p\left(w_1, w_2, \ldots, w_T\right) = p\left(w_1\right) \times p\left(w_2 | w_1\right) \times \ldots \times p\left(w_T | w_1, w_2, \ldots, w_{T-1}\right)$$

So the goal of a language model is to maximize $p\left(w_1, w_2, \ldots, w_T\right)$ and the loss function is usually a minimization problem, so you can define:

$$\operatorname{Loss}\left(w_1, w_2, \ldots, w_T | \theta\right) = -\log P\left(w_1, w_2, \ldots, w_T | \theta\right)$$

The training of model parameters of model through the gradient descend method of the loss function should be regarded as the most core step. When training the RNN model, the BPTT (backpropagation through time) algorithm is used. This algorithm uses the "chain rule" to solve the parameter gradient. The only difference is in each time step Parameter sharing. Xi et. al. [20] designed experiments on a game dataset with the results showing that it is suitable for long-term game playing.

3.3 Supervised Learning

Supervised Learning (SL) is a method in machine learning. It can learn or build a learning model from the training data, and assume new instances based on this model[23]. Training data consists of input objects (usually vectors) and expected output. function. The output of can be a continuous value (called regression analysis), or predict a classification label (insertion classification). It learn the model from samples. Large data sets are usually good for improving the performance of models. In the game, these data can come from game traces (that is, human records in the game), and use these data to train agents to map from input to output in a states space.

The resulting models are often fragile, the available data can be costly to produce and may lack key solutions that agents should handle. For game-playing, algorithms are also constraint with dataset, so new algorithm is not trainable. Hence, the usual scenarios of the application of supervised algorithms are games in combination with reinforcement learning algorithms and some additional training. State switching in games can also be an application of supervised learning. A neural network is able to give predictions for the next state rather than switching state directly.

Swiechowski et. al. [24] evaluated the effect of a supervised prediction model applying to game AI combined with Monte-Carlo Tree Search in a game of Hearthstone: Heroes of Warcraft. The experiment they designed showed that the model can evaluate game states properly. Chen et. al. [5] provided a vision-only model using deep supervised convolutional networks, which show that a supervised model has excellent performance. It demonstrates that supervised CNN models can be simple and effective. Solinas et. al. [17] trained on data from human gameplay sample likely worlds for evaluation, which use predictions about locations of individual cards made by a deep neural network. This technique performs well in the popular trick-taking card game of Skat combining with Perfect Information Monte Carlo search[7].

3.4 Unsupervised Learning

The goal of Unsupervised Learning(USL) is not to learn the mapping between data and its labels, but to find patterns in the data. The purpose is to classify the original data in order to understand the internal structure of the data. Unsupervised learning is a paradigm that prevents the creation of autonomous knights by rewarding agents that learn their observed data without considering specific tasks. In other words, agents learn for learning. Perhaps the simple goal of unsupervised learning is to train an algorithm to generate its own data instances. So-called generative models should not simply reproduce the data they are training on, this is an uninteresting memory behavior. The purpose of a generative model is to build a base class and a model that can extract data from it.

The autoencoder, which is a neural network that attempts to learn to make the input and output consistent is an important unsupervised learning technique. There are two parts in the network: an encoder that maps input x to a low-dimensional hidden vector h, and a decoder that attempts to reconstruct x from h. The core of the algorithm is that in order to keep h small, the network must learn to compress the data and thus learn a good representation. Researchers apply this unsupervised algorithm to games to extract more meaningful low-dimensional data from high-dimensional data. But this research direction is still at an early stage.

As a special technology, unsupervised learning can have very interesting and different applications in the field of game AI. They can not only assist supervised learning to complete some supervised learning tasks, but also can not get involved in some supervised learning The field is showing its strength. However, unsupervised learning also has its own shortcomings. They are often inefficient, difficult to train, and require some prior knowledge for dating. Many of these unsupervised learning algorithms are difficult to apply on a large scale.

Kulkarni et. al.[10] introduced a neural network architecture Transporter for representations of a concise geometric object, whose experiment executed on raw video frames through switching extracted features among frames in a fully unsupervised manner. Brown et. al.[3] presented an unsupervised learning tool to analyze big chess datasets, whose results show that chess does well in data mining in large datasets and machine learning evaluation experiments. In the domain of novelty-based highlight detection, Ringer et. al. [4] presented a multi-view unsupervised deep learning methodology, with results showing ideal results for game content generation.

3.5 Comparison

CNN can reduce the large data into relatively small data and keep the features without losing raw information, which means that it can be used in the game requiring frequent high-precision determination. The RNN can be used to generate the scene of games and bots. The SL learns the model from the given dataset, so it's better to apply to the scenario that large datasets are available. The SL can only learn known algorithms rather than unknown ones. SL can also be used to learn the transformation of game states. The USL learning the pattern of the data. It can reduce the dimension for high-dimension game data, so it can be promising in future research.

4 Conclusion

As one of the hot research directions recently, deep learning techniques has raised widespread popularity for its outstanding performance in Go and other video games. We mainly focus on how to use deep learning technology in the game industry to improve efficiency and change production methods. What we have listed are only a few typical applications of deep reinforcement learning in the game industry. Some of them have achieved substantial landings, bringing value to the industry, and some are still at the stage of exploration and experimentation. In fact, there are many complicated divisions of labor in the game industry. There are many and diverse scenarios far beyond this. Artificial intelligence technologies, including deep reinforcement learning, also have more possibilities to land, waiting to be tapped.

References

[1] **Bengio, Yoshua and Olivier Delalleau**, "On the expressive power of deep architectures," in "International Conference on Algorithmic Learning Theory" Springer 2011, pp. 18–36.

[2] **Bengio, Yoshua Glorot Xavier**, "Understanding the difficulty of training deep feedforward neural networks," in "Proceedings of the thirteenth international conference on artificial intelligence and statistics" 2010, pp. 249–256.

[3] **Brown, James A, Alfredo Cuzzocrea, Michael Kresta, Korbin DL Kristjanson, Carson K Leung, and Timothy W Tebinka**, "A Machine Learning Tool for Supporting Advanced Knowledge Discovery from Chess Game Data," in "2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)" IEEE 2017, pp. 649–654.

[4] **Charles, Ringer and Nicolaou Mihalis A.**, "Deep Unsupervised Multi-view Detection of Video Game Stream Highlights," in "Proceedings of the 13th International Conference on the Foundations of Digital Games" FDG '18 ACM New York, NY, USA 2018, pp. 15:1–15:6.

[5] **Chen, Zhao and Darvin Yi**, "The Game Imitation: Deep Supervised Convolutional Networks for Quick Video Game AI," 2017.

[6] **Dumitru, Manzagol Erhan**, "The difficulty of training deep architectures and the effect of unsupervised pre-training," in "Artificial Intelligence and Statistics" 2009, pp. 153–160.

[7] **Gaina, Raluca D., Simon M. Lucas, and Diego Perez-Liebana**, "Rolling horizon evolution enhancements in general video game playing," *2017 IEEE Conference on Computational Intelligence and Games, CIG 2017*, 2017, pp. 88–95.

[8] **Glorot, Xavier, Antoine Bordes, and Yoshua Bengio**, "Deep sparse rectifier neural networks," in "Proceedings of the fourteenth international conference on artificial intelligence and statistics" 2011, pp. 315–323.

[9] **Justesen, Niels, Philip Bontrager, Julian Togelius, and Sebastian Risi**, "Deep Learning for Video Game Playing," *IEEE Transactions on Games*, 2019, *PP* (c), 1–1.

[10] **Kulkarni, Tejas D, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih**, "Unsupervised learning of object keypoints for perception and control," in "Advances in Neural Information Processing Systems" 2019, pp. 10723–10733.

[11] **LeCun, Yann, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel**, "Handwritten digit recognition with a back-propagation network," in "Advances in neural information processing systems" 1990, pp. 396–404.

[12] **Lee, Honglak, Roger Grosse, Rajesh Ranganath, and Andrew Y Ng**, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in "Proceedings of the 26th annual international conference on machine learning" ACM 2009, pp. 609–616.

[13] **Lungu, I., F. Corradi, and T. Delbrück**, "Live demonstration: Convolutional neural network driven by dynamic vision sensor playing RoShamBo," in "2017 IEEE International Symposium on Circuits and Systems (ISCAS)" May 2017, pp. 1–1.

[14] **Mendes, Andre, Julian Togelius, and Andy Nealen**, "Hyper-heuristic general video game playing," *IEEE Conference on Computatonal Intelligence and Games, CIG*, 2016, *0*, 1–8.

[15] **Ringer, Charles and Mihalis A. Nicolaou**, "Deep unsupervised multi-view detection of video game stream highlights," *ACM International Conference Proceeding Series*, 2018.

[16] **Sironi, Chiara F., Jialin Liu, Diego Perez-Liebana, Raluca D. Gaina, Ivan Bravi, Simon M. Lucas, and Mark H.M. Winands**, "Self-adaptive MCTS for General Video Game Playing," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, *10784 LNCS* (January), 358–375.

[17] **Solinas, Christopher, Douglas Rebstock, and Michael Buro**, "Improving Search with Supervised Learning in Trick-Based Card Games," 2019.

[18] **Stanescu, M., N. A. Barriga, A. Hess, and M. Buro**, "Evaluating real-time strategy game states using convolutional neural networks," in "2016 IEEE Conference on Computational Intelligence and Games (CIG)" Sep. 2016, pp. 1–7.

[19] **Togelius, Julian and Georgios N. Yannakakis**, "General general game AI," *IEEE Conference on Computatonal Intelligence and Games, CIG*, 2016, *0*, 1–8.

[20] **Xi, Meng, Zhiling Luo, Naibo Wang, and Jianwei Yin**, "A Latent Feelings-aware RNN Model for User Churn Prediction with Behavioral Data," 2019.

[21] **Yann, Bengio LeCun**, "Deep learning," *nature*, 2015, *521* (7553), 436–444.

[22] **Yao, Andrew Chi-Chih**, "Separating the polynomial-time hierarchy by oracles," in "26th Annual Symposium on Foundations of Computer Science (sfcs 1985)" IEEE 1985, pp. 1–10.

[23] **Zhao, Chen and Yi Darvin**, "The Game Imitation: Deep Supervised Convolutional Networks for Quick Video Game AI," 2017.

[24] **wiechowski, M., T. Tajmajer, and A. Janusz**, "Improving Hearthstone AI by Combining MCTS and Supervised Learning Algorithms," in "2018 IEEE Conference on Computational Intelligence and Games (CIG)" Aug 2018, pp. 1–8.