IJISET - International Journal of Innovative Science, Engineering & Technology, Vol.5 Issue 4, April 2018
ISSN (Online) 2348 – 7968
www.ijiset.com

Investigating Deep Learning and Self Play in Artificial Intelligent Games

Adamu Adamu Habu[1]

[1] Department of Mathematics and Computer Science, Federal University of Kashere,
Gombe State, P.M.B 00182, Nigeria

Abstract
Deep learning, a technique used for learning in a more complex neural network with many layers of abstraction. It was able to produce best solutions in computer science problems such as computer vision, image recognition, speech recognition, and natural language processing. A deep learning network contains multiple layers of abstraction and as such tends to be more powerful than shallower networks. On the other hand, self play is an artificial game that plays itself repeatedly in order to acquire training and skills. It is enforced through a technique called, reinforcement learning. In this work, we investigate deep learning and self play and answer some promising questions. Moreover, we provide an analysis and conclusions for both deep learning and self play that address symbolic Artificial Intelligence look-ahead search, machine learning mechanisms, and uncertain reasoning.

Keywords: *Deep Learning, Self Play, Artificial Intelligence, Reinforcement Learning*

1. Introduction

Deep learning, a machine learning technique that employs many layers of hierarchy in the exploitation of information [1], has been used by many researchers to improve the state of art in areas that include but not limited to speech recognition [2], visual object recognition [3], and image recognition [4]. A deep learning network contains multiple layers of abstraction and as such tends to be more powerful than shallower networks. For example, in a network that recognise pattern of objects. The first layer can recognise edges; the second layer can recognise shapes like circle, triangle etc. while the third layer can recognise complex objects. The multiple layers give deep network advantage in learning. However, deep networks are hard to train because different layers are learning at different speeds [5]. Self play, on the other hand is an artificial game that plays itself repeatedly in order to acquire training and skills. For example, a set of 50 neural networks can be played and the one that produces the best possible result is selected and used to train the network. In such a situation where a network trains itself, is regarded as self play. However, self play can be enforced through a technique called reinforcement learning [6].

In reinforcement learning, the agent learns what to do in the absence of training example. As an example, in a game of chess, in supervised learning the agent would be told what to do and what not to be done. However, in the absence of feedback from a teacher the agent has to decide the correct move to defeat the opponent. It will be difficult for the agent to decide on which move to be executed if he has no idea about what is right or wrong. This kind of information is refers to what is called reward or reinforcement. In some games like chess, the reward comes at the end of the game while in others, the reward comes more frequently. The reinforcement learning is used to observe reward and suggest the best solution for the environment by rewarding for better play and penalising for loses. Reinforcement learning is the best way to train a program because; human cannot predict accurate and lack consistent evaluation of some large positions [6].
In this work, we investigate deep learning and self play and provide an account of how a single layered feed-forward network learns and produce output using back propagation. Second, we investigate the extent to which self play can be applied to variety of games and answer the question; if self play can allow Artificial Intelligent, AI game player to acquire expertise beyond the rules of the game? Third, we investigate if deep learning has been successfully applied to self play? Furthermore, we provide analysis that address symbolic look ahead search, machine learning mechanism, and uncertain reasoning for both deep learning and self play.

2. Literature Review

2.1 Deep learning

Neural network, a network that mimics the human brain have been used for many years to solve problems in areas such as speech and image recognition, natural language processing, game theory to mention but a few. It uses high level of computation, instead of using program of instructions as in Von Neumann computers. It employs parallel nets connected via links with variable weights. The weighted input passes through any of the three non linearities; hard limiters, threshold logic elements and sigmoid non linearities and may include temporal integration or other mathematical operations depending on the complexity of the network. Neural network is characterized by network topology, node properties and training rules that specify initial state of the weight and how they can be adapted to ensure improved performance. Furthermore, it has the ability to learn and adapt to environmental challenges [6][7][8].

Until 2006, deep multi layered neural networks were not successfully trained, instead focus was on shallower architectures. Complex computations that involve human information processing mechanisms such as speech and image recognition suggest the need for deep architectures. It is believed that if efficient and effective deep learning algorithms were developed there would be significant development in these areas. The concept of deep learning was originated from artificial neural network research. Feed-forward network with more than one hidden layer is a good example of deep architecture. Single feed-forward architecture are normally trained using back propagation, unfortunately, back propagation has some limitations as regard to deep learning. However, there is currently high level research in the area of deep learning [9].

2.2 Feed Forward Network

A feed-forward network with a single hidden layer consists of three layers and like every other feed-forward network, it has no feedback loop. The layers are as follows:

Input Layer: In this layer, no processing takes place. However, it supply input signals to the second layer.

Hidden layer: Processing starts from the hidden layer. For this kind of neural network, the hidden layer consists of only one layer. The output signals of the input layer are the inputs to the hidden layer. Both the input and the output of the hidden layer cannot be seen from the outside world.

Output Layer: The output layer constitutes the overall decision of the network. It receives its input from the hidden layer.

A feed-forward network is said to be fully connected when every neuron in a particular layer is connected to every neuron in the preceding layer. The weight would be represented by W with subscript 1 for input layer, 2 for Hidden layer and 3 for Output layer. The weight, W_i would be multiplied by the input, X_i and the Summation of all the products and an external bias, b_k on each neuron are parsed through an activation function, which is either hard limiters, threshold or sigmoid. The activation function, defines the output of the neuron. Different algorithms exist for training neural networks, for this paper the back propagation algorithm was adopted [7].

The Back propagation consists of two phases. In the forward phase, the input is propagated from layer to layer until it reaches the output, the weight of the network are fixed. However, the activation potential and the output of the neuron are affected by the changes in this phase. In the backward phase, the output of the network is compared with a desirable response. The difference is taken as the error, which is propagated across the network from layer to layer in backward direction. The weights are adjusted based on some calculation procedure [10].

2.3 Back Propagation Training Algorithm

To adjust weight's in a neural network. Its effect to the output and the error has to be known, which is calculated using the derivative of the error with respect to the weight. A non-linear output function that is differentiable is used; in this case we used the function called the sigmoid function.

The following is the algorithm that trains the neural network by minimizing the mean square error between the actual output and the desired output [7].

Step1: set weights and nodes to small random values.

Step2: inputs and desirable outputs are presented.

Step3: using sigmoid function, the actual output is calculated

Step4: weights are adjusted.

Step5: repeat Step2 until the algorithm converges using set criteria.

2.4 Similarities and Differences between Deep Learning Networks and Feed Forward Networks

On the one hand, both feed-forward networks with a single hidden layer and deep learning neural networks acquire knowledge through learning algorithms from their environment. They modify the weights in the network to acquire desirable results. Secondly, they can change their network topology similar to that of the human brain. And can adapt to their environment by changing their weight to deal with minor environmental conditions. Additionally, since they are built based on neuron biological analogy, they both have fault tolerant parallel processing capabilities [5].

On the other hand, while, the feed-forward network with a single hidden layer consist of only one hidden layer. The deep learning network can have multiple layers of abstraction. Additionally, the deep learning network encounters the vanishing gradient problem. When there are many layers, an unstable situation occurs, in which the earlier neurons tend to learn much slower than the latter layers. Therefore, in order to remedy the problem the standard gradient based learning techniques should not be used. The feed-forward network with single hidden layer does not experience inconsistency in neurons. Therefore, standard gradient based learning techniques can be used [5]. Lastly, sigmoid activation functions can be used in feed-forward network with single hidden layer. As argued by Glorot & Bengio [11], in deep learning network, sigmoid activation function can cause training problems, because it causes the final hidden layer to saturate at 0, which is the beginning of the training.

3. Analysis

Wiering [12] experiment Backgammon game with temporal difference learning and uses three different methods of learning namely: (1). learning by self play, (2). learning by playing against an expert program and (3). learning from viewing experts play against themselves. The results shows that learning from viewing experts play against each other presents better results at the start as when compared with the initial random games from self play. Second, if a game involves the use of database, learning from self play would not be more advantageous. Although, the third options has its own disadvantages. Kwasnicka & Spirydowicz [13] designed a program called checkers, which is based on the game checkers. The program contains neural network and is trained by playing against itself. They argue that the program can play checker at intermediate level and it is easier to get better program that can play checker without learning possibilities. Considering the result of the two experiments above, in this work, it is our position that self play can be applied to train games but learning experts play against themselves should be considered if the game contains database because it will be easier for the training since the game can easily learn how to manipulate the database. Secondly, in classic deterministic games learning by playing against an expert would be better option since the game can monitor the moves of the expert and easily adapt the skills.

On a second note, the AI game cannot get continuous expertise using self play in the above stated games because knowledge is power, if you don't have it you don't have it and even if the game can acquire skills it will take longer period of time to train as in comparison to when other learning methods were used.

We argue that analysis and conclusions of different types of games should be different because some games uses self play while others are complex to implement self play. Additionally, different games use different AI techniques. Look ahead search for example are implemented in games like chess. With the recent development in deep learning, it will be useful to compare the performance of deep learning as applied to self play. Van De Steeg et al. [14] compared five different structures of multilayer perceptrons for learning a Tic-Tac-Toe 3D game when trained using self play and when trained using fixed opponent. Three of the perceptrons have different number of hidden layers while the two are structured ones. It was found that the deep structured neural network produced the strongest output as compared to the others both through the self play and the fixed opponent. Reinforcement learning was used to learn from the experience of the agents. After some training, the agent should be able to choose actions that can maximize its future intake. The reinforcement learning was implemented using TD-learning, an algorithm that uses temporal difference error. The algorithm updates the value of states before the opponents turn. At the end of the game, the player receives a 1 if it wins, -1 if it lose and 0 for draw. Two experiments were performed for each of the five multilayer perceptrons. In experiment one, the multilayer perceptron was trained using self play and was tested against a fixed bench mark player. In the second experiment, the network was trained by playing against the same bench mark player without learning the moves of the opponent.

4. Conclusions

Deep learning uses the concept of neural networks to process hierarchical architectures. In the case of look ahead search, I will argue that the deep learning don't use such concept because the idea of deep learning is to provide solutions through learning process. Machine learning concept is fully applied in deep learning because the neural network is trained using reinforcement learning. It gets a reward when it wins and penalise when it loses. After training, the deep learning acquires skills and is able to maximize its reward. Several layers of nodes exist in deep leaning, the network has to decide which node or layer to be expanded, uncertainty is used to decide on the node to expand. The deep learning algorithm has to use probabilistic theorems to decide the best node that maximize its reward. Additionally, uncertainty is used to solve the problems of settling on local minima rather than global minimum in a Hopfield network [7].

A neural network for classic deterministic games such as Go, Chess can use look-ahead search towards the end of the game to maximize the chances of winning. Machine learning concept can be used to train the network either by studying the moves of the opponent or when experts play against themselves. The program can learn by playing against itself but the speed of the training will be low as compare to the other methods.

Classical deterministic games don't use uncertainty. Classical games like Backgammon cannot use look-ahead search because uncertainty is used to predict their moves. However, uncertainty is applicable to this kind of games. Machine learning techniques are used to train the network by allowing it to play itself.

References

[1]. Li Deng and Dong Yu (2014), "Deep Learning: Methods and Applications", Foundations and Trends® in Signal Processing: Vol. 7: No. 3–4, pp 197-387. http://dx.doi.org/10.1561/2000000039

[2]. Deng L, Hinton G, Kingsbury B. New types of deep neural network learning for speech recognition and related applications: An overview. InAcoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on 2013 May 26 (pp. 8599-8603). IEEE.

[3]. Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, Darrell T. Decaf: A deep convolutional activation feature for generic visual recognition. InInternational conference on machine learning 2014 Jan 27 (pp. 647-655).

[4] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 770-778).

[5]. Nielsen MA. Neural networks and deep learning (2015). Also available at: http://neuralnetworksanddeeplearning. com. 2016.

[6]. Russell SJ, Norvig P. Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited,; 2016.

[7]. Picton P. Introduction to neural networks. Macmillan Publishers Limited; 1994.

[8]. Kröse B, Krose B, van der Smagt P, Smagt P. An introduction to neural networks.

[9]. Deng L. A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing. 2014 Jan;3.

[10]. Haykin SS. Neural networks and learning machines. Upper Saddle River, NJ, USA:: Pearson; 2009.

[11]. Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics 2010 Mar 31 (pp. 249-256).

[12].Wiering MA. Self-Play and Using an Expert to Learn to Play Backgammon with Temporal Difference Learning. JILSA. 2010 May 1;2(2):57-68.

[13]. Kwasnicka H, Spirydowicz A. CHECKERS: TD (λ) LEARNING APPLIED FOR DETERMINISTIC GAME.

[14]. Van De Steeg M, Drugan MM, Wiering M. Temporal difference learning for the game tic-tac-toe 3d: Applying structure to neural networks. InComputational Intelligence, 2015 IEEE Symposium Series on 2015 Dec 7 (pp. 564-570). IEEE.

First Author Adamu Adamu Habu holds a Bachelors degree of Technology from Abubakar Tafawa Balewa University, Bauchi, Nigeria (2010) and a Masters degree in Computing and IT from the University of St Andrews, Scotland, United Kingdom (2017). He worked at the technology group of Sterling Bank Plc, Kano regional office from 2011 to 2012 as a national youth corps member. Since 2012 he works as an academic staff at the department of Mathematics and Computer Science, Federal University of Kashere. He is a member of Internal Association of Engineers (Membership number: 213118) and a recipient of 2015 overseas scholarship award by the Nigerian, National Information Technology Development Agency (NITDA). Research interest include; Artificial Intelligence and Human Computer Interaction.