Research Journal Based On A. M. Turing, Computing Machinery and Intelligence

COMP110 - Research Journal

1700522

November 17, 2017

1 Introduction

In 1950 Alan Turing posed the question "Can machines think?"[1] a question that to this day has not truly been answered. Artificial Intelligence or AI research has a number of focuses exploring Turing's initial question utilising multiple methods. This journal will be looking into the use of AI in video games to create what Bridger calls a "brain vs brain"[2] scenario. It is important in the context of this journal to understand the difference between machine learning and deep learning as well as what the Turing Test is.

2 The Imitation Game

The Imitation Game is often referred to as the 'Turing Test' and was set out by Alan Turing in Computing Machinery and Intelligence[1]. The Imitation Game tests an AI to see if it can evolve and adapt to a conversation the same as a human would. The imitation game requires an AI to be able to work without perfect information, this is the only way the "interrogator" will not be able to tell if it is a person or computer. As it currently stands no AI has ever truly passed Imitation Game. This delves into the field of artificial human interaction as this is what the Turing Test is based on. While the Imitation Game tests if a human can be fooled by an AI is this the best place to focus our research? It is the position of this journal that Turing's ideas for AI were in fact flawed and the best focus for AI research is applying AI to problem solving as to best apply AI to benefit society.

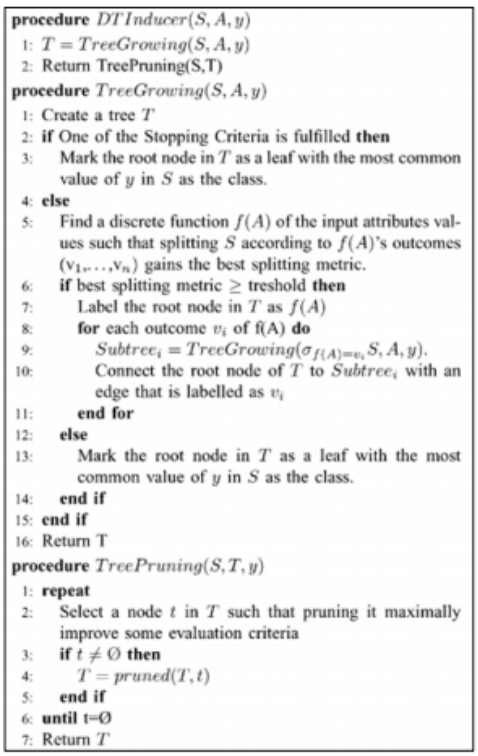
3 Machine Learning vs Deep Learning

In AI research one of the focuses is the training of these AI units to be able to reliably complete tasks and there are 2 main ways to do this, Machine Learning and Deep Learning.

3.1 What is Machine Learning?

Machine Learning can be defined as "The capacity of a computer to learn from experience, i.e. to modify its processing on the basis of newly acquired information." according to the Oxford dictionary[3]. This means that machine learning works through the adapting to new information that is presented to it and that it can 'learn' through completing the same task multiple times. This is how AIs learned to play checkers[4] and chess[5]. This research lead to the 'Deep Blue' defeating the world chess champion[6].

Machine Learning is split into 3 types of algorithms; Supervised learning, unsupervised learning and reinforcement learning. Supervised Learning algorithms can be described as "The algorithm consist of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables). Using these set of variables, we generate a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data."[7].



procedure $DTInducer(S, A, y)$
1: $T = TreeGrowing(S, A, y)$
2: Return TreePruning(S,T)

procedure $TreeGrowing(S, A, y)$
1: Create a tree T
2: **if** One of the Stopping Criteria is fulfilled **then**
3: Mark the root node in T as a leaf with the most common value of y in S as the class.
4: **else**
5: Find a discrete function $f(A)$ of the input attributes values such that splitting S according to $f(A)$'s outcomes $(v_1,..,v_n)$ gains the best splitting metric.
6: **if** best splitting metric $\geq$ treshold **then**
7: Label the root node in T as $f(A)$
8: **for** each outcome v_i of f(A) **do**
9: $Subtree_i = TreeGrowing(\sigma_{f(A)=v_i}S, A, y)$.
10: Connect the root node of T to $Subtree_i$ with an edge that is labelled as v_i
11: **end for**
12: **else**
13: Mark the root node in T as a leaf with the most common value of y in S as the class.
14: **end if**
15: **end if**
16: Return T

procedure $TreePruning(S, T, y)$
1: **repeat**
2: Select a node t in T such that pruning it maximally improve some evaluation criteria
3: **if** $t \neq \emptyset$ **then**
4: $T = pruned(T, t)$
5: **end if**
6: **until** t=Ø
7: Return T

Figure 1: Pseudo code for Decision Tree [8]

Figure 1 above shows the algorithm for a decision tree search, this is a type of supervised learning algorithm.

Unsupervised Learning algorithms can be described as "This algorithm, we do not have any target or outcome variable to predict / estimate. It is used for clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention."[7].

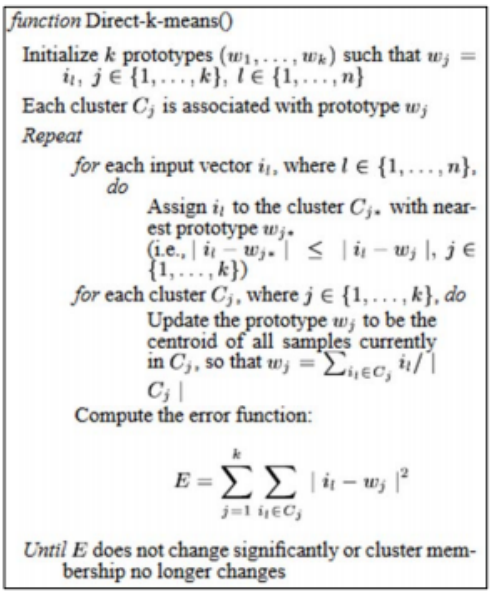


Figure 2: Pseudo code for k-means clustering [8]

Figure 2 shows the algorithm for k-means clustering, this is a type of unsupervised learning algorithm.

Reinforcement Learning algorithms can be described as "This algorithm, the machine is trained to make specific decisions. It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions. "[7]. Temporal Difference Learning, (or TD learning) is a reinforcement learning algorithm that was used in 1995 by Tesauro to play backgammon [9]. TD-Gammon was the attempt to make use of TD learning to play the game backgammon at a high level. The results of Tesauro's research showed promise for reinforcement learning which could then be applied to other AI research in later years.

Algorithm 1: TD(λ) for approximate policy approximation:

Data: a simulation model for a proper policy π in MDP $\mathcal{M}$.
a feature function $\phi : \mathcal{S} \to \mathbb{R}^K$, mapping states to feature vectors, $\phi(T) \stackrel{\text{def}}{=} 0$;
a parameter $\lambda \in [0, 1]$; and
a sequence of step sizes $\alpha_1, \alpha_2, \ldots$ for incremental coefficient updating.

Output: a coefficient vector $\boldsymbol{\theta}$ for which $V^\pi(s) \approx \boldsymbol{\theta}^T \boldsymbol{\phi}(s)$.

Set $\boldsymbol{\theta} := 0$ (or an arbitrary initial state), $t := 0$.
for $n := 1,2,\ldots$ **do**
  Set $\boldsymbol{\delta} := 0$.
  Choose a start state $s_t \in S$.
  Set $\boldsymbol{e}_t := \boldsymbol{\phi}(s_t)$.
  while $s_t \neq T$ **do**
    Simulate one step of the process, producing a reward r_t and next state s_{t+1}.
    Set $\boldsymbol{\delta} := \boldsymbol{\delta} + \boldsymbol{e}_t(r_t + (\boldsymbol{\phi}(s_{t+1}) - \boldsymbol{\phi}(s_t))^T \boldsymbol{\theta})$.
    Set $\boldsymbol{e}_{t+1} := \lambda \boldsymbol{e}_t + \boldsymbol{\phi}(s_t)$.
    Set $t := t + 1$.
  end
  Set $\boldsymbol{\theta} := \boldsymbol{\theta} + \alpha_n \boldsymbol{\delta}$.
end

Figure 3: Pseudo code for temporal difference learning
[10]

Figure 3 shows the algorithm for TD learning, this is a common refinforcement learning algorithm. The Markov decision processes is a mathematical framework used within reinforcement learning. The MDP framework can be used in combination with dynamic programming to solve optimisation type problems the algorithm is as follows:

$$(S, A, P.(\cdot, \cdot), R.(\cdot, \cdot), \gamma)$$

- S a potentially finite set of states
- A a potentially finite set of actions (A_s is the set of actions from the state of s)
- $P_a(s, s')$ is the probability that action a in the state of s at a given time will result in state s' at the next time step
- $R_a(s, s')$ is the expected immediate reward from the change to s' from s as a direct result of a
- $\gamma \in [0, 1]$ is the discount factor. This is the difference in future and present rewards.

The aim of MDP is to chose a policy (π) that maximises the function for rewards with the problem:

$$\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_t + 1)$$

(where we choose $a_t = \pi(s_t)$) [11] This is done through a combination of mathematical operations which allow the AI to reflect and build up its database through trial and error.

3.1.1 Monte Carlo Tree

In more recent years a method known as 'Monte Carlo Tree Search' has evolved within machine learning, specifically within supervised learning. "Monte Carlo Tree Search (MCTS) is a method for finding optimal decisions in a given domain by taking random samples in the decision space and building a search tree according to the results."[12]. MCTS evaluates what the best option for the AI to take is and then takes it. Machine learning algorithms are used to allow the MCTS algorithm to calibrate itself so that it can correctly evaluate the options available in order to correctly select the optimal choice each time. MCTS has been applied to a number of games in order to test its effectiveness. MCTS has been applied to 2 games that this journal will explore Dou Di Zhu, (a popular Chinese card game)[13] and Hearthstone, (an online collectable card game)[14]. In these examples Powley and Santos both explore how MCTS can be used in card games but with a very different set of rules and parameters. In his paper Santos defines MCTS as "a de facto standard in game AI"[14] making this area of AI research incredibly relevant to the appliance of AI for problem solving rather than the mimicking of human interactions as Turing had in mind. This aligns with the views of this journal as it is for the application of AI for problem solving tasks.

3.2 What is Deep Learning?

Deep Learning is a method of training an AI as though it was a human brain, making use of artifical neurons to simulate human learning as described by Philbin[15]. Neural networks have recently. Deep Learning has been applied to research tasks across the world, one such example was in 2016 at Chungnam Nationality University in Korea, [16]. Lee's team discovered that they could use Deep Learning to work out how to reduce the distraction in smartwatch users. This application of artificial intelligence is very different to what has been seen so far in this journal. Most of the AIs this journal has looked at have been dedicated AIs, designed to to a single task well. Deep Learning AIs make use of artificial neural nets[15] and as such have the potential to be more like the Artificial Intelligence that Turing was referring to in 1950[1]. This use of Deep Learning for simulating people in research was also used in Taipei at Tamkang University to assess the sentiment in reviews on Google Play[17]. This is what appears to be the use of Deep Learning, the simulation of human activity. This is further reinforced by Huawei making use of Deep Learning to aid in the design of their phones[18].

4 How does Machine Learning & Deep Learning Compare?

Machine and Deep Learning algorithms appear to be designed for 2 very different tasks. A Machine Learning algorithm follows the more traditional path artificial intelligence has taken since the 1950's. As such Machine Learning has evolved to perform dedicated tasks such as play games like chess and Dou Di Zhu[4, 5, 6, 13, 14]. Deep learning on the other hand seems like it is more likely to pass Turing's Imitation Game[1] as the algorithms are designed to act more like human brains to the extent where they can be used for research [15, 16, 17, 18]. Deep Learning makes use of incredibly expensive and complex hardware and as such is greatly under developed by comparison. It is the position of this journal that Machine Learning is greatly superior for dedicated tasks but Deep Learning is much closer to the general purpose AI that Turing was envisioning. Turing's envisionment of AI however in the author's did lack a question; Should we create an AI that can simulate humanity? As there is no knowing what a Strong Deep Learning AI would be capable of once technology reaches a point where it is possible. It is however the author's belief that a strong Deep Learning AI would beat the Imitation Game and pass the Turing Test.

References

[1] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. 236, pp. 433–460, 1950.

[2] B. C. Bridger and C. S. Groskopf, "Fundamentals of artificial intelligence in game development," in *Proceedings of the 38th Annual on Southeast Regional Conference*, ser. ACM-SE 38. New York, NY, USA: ACM, 2000, pp. 51–55.

[3] *Oxford English Dictionary*. Oxford University Press, 2017.

[4] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 44, no. 1.2, pp. 206–226, 1959.

[5] C. E. Shannon, "Programming a computer for playing chess," *Philosophical Magazine*, vol. 41, 7th Series, pp. 256–275, 1950.

[6] M. Campbell, "Knowledge discovery in deep blue," *Commun. ACM*, vol. 42, no. 11, pp. 65–67, Nov. 1999.

[7] "Essentials of machine learning algorithms (with python and r codes)," https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/, accessed: 2017-11-13.

[8] A. Dey, "Machine learning algorithms: A review," *International Journal of Computer Science and Information Technologies*, vol. 7, no. 3, pp. 1174–1179, 2016.

[9] G. Tesauro, "Temporal difference learning and td-gammon," *Commun. ACM*, vol. 38, no. 3, pp. 58–68, Mar. 1995.

[10] F. Kunz, "An introduction to temporal difference learning," in *Seminar on Autonomous Learning Systems*, 2000.

[11] E. Altman, *Constrained Markov decision processes*. CRC Press, 1999, vol. 7.

[12] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[13] E. J. Powley, D. Whitehouse, and P. I. Cowling, "Determinization in monte-carlo tree search for the card game dou di zhu," *Proc. Artif. Intell. Simul. Behav*, pp. 17–24, 2011.

[14] A. Santos, P. A. Santos, and F. S. Melo, "Monte carlo tree search experiments in hearthstone," in *Computational Intelligence and Games (CIG), 2017 IEEE Conference on*. IEEE, 2017, pp. 272–279.

[15] C. A. Philbin, "Machine learning & artificial intelligence: Crash course computer science #34)," https://www.youtube.com/watch?v=z-EtmaFJieY, PBS Digital Studios, accessed: 2017-11-11.

[16] J. Lee, J. Kwon, and H. Kim, "Reducing distraction of smartwatch users with deep learning," in *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, ser. MobileHCI '16. New York, NY, USA: ACM, 2016, pp. 948–953.

[17] Y.-D. L. Min-Yuh Day, "Deep learning for sentiment analysis on google play consumer review." IEEE, 2017.

[18] H. Guo, R. Tang, Y. Ye, and X. He, "Holistic neural network for ctr prediction," in *Proceedings of the 26th International Conference on World Wide Web Companion*, ser. WWW '17 Companion. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2017, pp. 787–788.