Journal of Interdisciplinary and Multidisciplinary Research (JIMR)
E-ISSN:1936-6264| Impact Factor: 8.886|
Vol. 19 Issue 02, February- 2024
Available online at: https://www.jimrjournal.com/
(An open access scholarly, peer-reviewed, interdisciplinary, monthly, and fully refereed journal.)

Literature survey on Attention based Neural Networks for Autonomous Driving Agents

Anand Tumma[1]
Assistant Professor(C)
Department of CSE
Telangana University

Prof. Arathi Chitla[2]
Department of CSE
Telangana University

Harshith Raj Boddu[3]
Big Data Python Developer
SAS2PY Pvt.Ltd.

Abstract: Autonomous driving is a very useful and important topic that reforms the car industry. To provide safety in this kind of driving we need the help of Machine learning where some good decisions can be made while driving. Neural networks are a subset of Machine learning. Neural networks allow us to train an agent in an efficient manner with a wide variety of its algorithms. This paper surveys and classifies algorithms and mechanisms that are useful in Autonomous Driving.

Keywords: Attention mechanism, Autonomous Driving, Driving agent, F1/10 race car, Machine learning, Neural network.

I. Introduction: Autonomous car is a vehicle that can drive on its own. This type of car is capable of sensing the environment so that it react to the upcoming events. The usage of an increasing number of sensors and actuators as well as deep neural networks for a car's automatic action are necessary to provide safe and stable driving. Classical approaches cannot capture all possible upcoming events and incidents hence a promising solution for this is the use of machine-learning.

ML algorithms are often classified under one of three broad categories: supervised learning, unsupervised learning and Reinforcement learning (RL). Supervised learning algorithms are based on inductive inference where the model is typically trained using labelled data to perform classification or regression. Whereas unsupervised learning encompasses techniques such as density estimation or clustering applied to unlabelled data. By contrast, in the RL paradigm an autonomous agent learns to improve its performance at an assigned task by interacting with its environment. Russel and Norvig define an agent as "anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators" [12].

RL agents [4] are not told explicitly how to act by an expert; rather an agent's performance is evaluated by a reward function R. For each state experienced, the agent chooses an action and receives an occasional reward from its environment based on the usefulness of its decision. The goal for the agent is to maximize the cumulative rewards received over its lifetime. Gradually, the agent can increase its long-term reward by exploiting knowledge learned about the expected utility (i.e. discounted sum of expected future rewards) of different state action pairs. One of the main challenges in reinforcement learning is managing the trade-off between exploration and exploitation. To maximize the rewards it receives, an agent must exploit its knowledge by selecting actions which are known to result in high rewards. On the other hand, to discover such beneficial actions, it has to take the risk of trying new actions which may lead to higher rewards than the current best-valued actions for each system state. In other words, the learning agent has to exploit what it already knows in order to obtain rewards, but it also has to explore the unknown in order to make better action selections in the future. Examples of strategies which have been proposed to manage this trade-off include €-greedy and softmax. When adopting the ubiquitous €-greedy strategy, an agent either selects an action at random with probability $0 < \epsilon < 1$, or greedily selects the highest valued action for the current state with the remaining probability 1- €. Intuitively, the agent should explore more at the beginning of the training process when little is known about the problem environment. As training progresses, the agent may gradually conduct more exploitation than exploration. The design of exploration strategies for RL agents is an area of active research.

The difficulties for this task arise from the scarce availability of datasets and specific literature [4]. Recent studies include attention as a mechanism to interpret and improve algorithms used for deep learning [3].

In virtual reality, attention already shows improvements for existing networks [6]. In correspondence to that, it is interesting to consider human attention to decide a vehicle's control for a specific state. In this context, it is hard to synthesize models and algorithms for driving behaviour without compromising safety. Since humans are able to cope with complex driving scenarios, a human inspired attention mechanism could improve autonomous driving.

The expected system should be able

- To describe how much is a neural network's training and prediction performance improved by the attention mechanism.
- To list the observable and measurable differences between the simulation and real car performance.
- To explain about the robustness of the trained agent to domain shift (e.g. different track)
- To develop a human inspired attention mechanism to improve autonomous driving.

II. Review of Literature:

For getting the necessary knowledge about autonomous driving and deep learning a review of the existing literature has to be done beforehand. F1/10 Autonomous car is a car which is exactly in 1/10 scale of original Formula One race car. These type of cars are widely used to do research on Autonomous driving. The main aim of using F1/10 cars is to get a deeper understanding of the F1Tenth- framework for the real car and the simulation.

In robotic simulation there are simulators like Gazebo are already available in the market. In this Robot operating system (ROS) [1] serves as an interface for robots while Gazebo is 3D simulator. But when it comes to F1/10 simulation apart from the simulator and its back end components, the F1/10 simulator supports racing metrics and strategies employed in popular motor sports like Formulae 1, Indianapolis 500 etc., using a dedicated telemetry system. Carla and Apollo are some examples of autonomous vehicle simulators. Simulators like Carla are open-source and relatively lightweight, but still focused on autonomous driving research in similar settings as Apollo. Autonomous racing differs from autonomous driving in certain key areas like higher speeds, lack of minimum safe following distance & lack of lanes, etc. The main limitations of the simulators above is the lack of a one-to-one correspondence with a physical testbed. The Amazon DeepRacer is a similar but less capable hardware platform compared to the F1/10 racecar [1]. It also has a simulated environment, but the entire DeepRacer platform is focused mostly on enabling Reinforcement Learning (RL). The DeepRacer is also a standalone platform with no out-of-box support for multi-vehicle operation. In addition to the complexity of the simulators themselves, the threshold to get started is very high both in terms of computer hardware and experience, and this is where F1/10 simulator tries to fill in the gap by being a lightweight autonomous racing simulator and providing an easy to learn Python API, F1/10 simulator make vehicular autonomy easily accessible.

The simulations of the car's behavior will be done with the mentioned F1Tenth-framework [1]. Therefore the specific model predictions will be integrated to ensure reasonable outcomes in simulation [4]. This has to be done for making sure that the models work properly before the deployment on the real car.

When simulation and models have reached the desired level of accuracy, it can be considered to drive the real car on a track. The analysis [5] of the performance on a race car will drive the additional refinements and possible iterations in simulation to ensure safety on the real track.

According to Mrinal R. Bachute and Javed M. Subhedar [2] the Tasks need to be performed by autonomous vehicles are

1. Perception
2. Motion planning
3. Pedestrian detection
4. Traffic sign detection
5. Road-marking detection
6. Self-localization
7. Automated parking
8. Motion control
9. Vehicle cyber security

Recently, the attention mechanism has been applied in natural language processing, image recognition, and bioinformatics. The attention mechanism assigns different importance scores to individual positions in its intermediate layer so that the model can focus on the most relevant information within the input. In 2D image analysis, the attention weights for individual positions on the contact map allow the visualization of critical regions that contribute to the final predictions. In addition, these weights are generated during the inference step, without requiring additional computation procedures after the prediction of a contact map. Hence, the attention mechanism is a suitable technique to facilitate the interpretation of protein contact prediction models [3].

Chen Chen et al. [3] proposed attention-equipped deep learning method for protein contact prediction, which adopts two different architectures of the attention targeted for interpreting 2D and 1D input features, respectively. The regional attention utilizes the n×n region around each position of its input 2D map while the sequence attention utilizes the whole range of its 1D input. The regional attention module is implemented with a specially designed 3D convolutional layer so that training and prediction on large datasets can be performed with high efficiency. The sequence attention is similar to the multi-headed attention mechanism applied in the NLP tasks. The author identified that their attention-based methods can achieve reliable accuracy while improving the interpretability of the model and providing potentially useful insights for the identification of residues that are critical in determining protein folds.

Alexander Makrigiorgos et al [6] uses an automated navigation expert provided by the CARLA developers to collect data for the training of our autonomous driving agents. This agent has access to privileged information about the state of the server at all times, such as a map of its environment and the exact positions of all other agents in the simulation. Using this information along with hard-coded rules about how to react in traffic situations (e.g. stop at red traffic lights or when pedestrians are ahead), the vehicle navigates from an initial random position to a randomly chosen destination, recording images from a camera mounted in the same position as in our gaze collection experiments along with the driving signals (throttle, brake, steering angle, current speed) used to train the networks and the high-level commands associated with each image. Upon reaching the destination, the episode is terminated and a new episode begins, randomizing the weather settings and the number and location of pedestrians and other vehicles. Noise is occasionally added to the agent's actions to increase the diversity of the training set, but this addition is not reflected in the recorded commands to ensure that trained agents do not learn noisy behaviour. In the rare case that the noise results in a collision, the episode is terminated and is not used for training purposes.

Reinforcement learning [7] provides a way to learn arbitrary policies considering specific goals. In recent years learning- based approaches have been used to tackle similar or related problems, like learning-based on human driving, inverse reinforcement learning, end-to-end algorithms that map sensed inputs (mainly visual, images) directly to control signals , and methods that understand the scene via learning to make tactical decisions.

While offline solutions are able to handle complex scenarios and compute a policy before implementation, but they are impractical because there exists a large number of presumable real-world situations. So, it is intractable to precompute a general policy that would be applicable in all possible cases. Online methods calculate a policy while they are experiencing the world, which makes them dominant compared to than their offline counterparts.

Single & Multi agent Framework:

The single agent reinforcement learning framework is based on the model where an agent interacts with the environment by selecting actions to take and then perceiving the effects of those actions, a new state and a reward signal indicating if it has reached some goal (or has been penalized, if the reward is negative). The objective of the agent is to maximize some measure over the rewards, like the sum of all rewards after a number of actions taken.

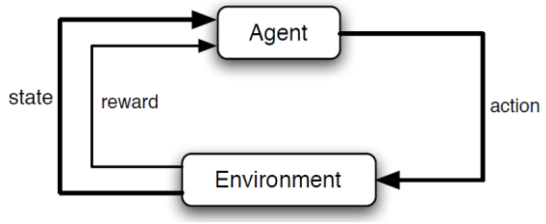

Fig 2: Single agent Framework

The multi-agent framework is based on the same idea of Single Agent but, this time, there are several agents deciding on actions over the environment. The big difference resides in the fact that all each agent probably has some effect on the environment and, so, actions can have different outcomes1 depending on what the other agents are doing. This is precisely the difference that poses problems when applying reinforcement learning techniques to the multi-agent domain. Usually, those are designed to solve stationary environments and, from the point of view of each agent, the environment is no longer stationary.

To model such a domain, the focus has turned to game theory, which is designed to solve multi-agent situations and in which the solutions involve compromises and cooperation. Particularly, the model most commonly used is that of stochastic games or the subclass of matrix games.

Deep Reinforcement Learning:

In this technique DRL [9] each driving agent as Autonomous Car (AC) receives partial input state observation of 84x84x3 dimension images through the front camera sensors. Cameras are mounted as part of the driving agents, and during each time step of the simulation environment, cameras capture the input state observations which serve as an input layer to the DRL model. The input layer is then passed to the convolutions and connected to hidden layers for extracting important features before they are passed to the output layer of the architecture.

ACs driving policies predict the control actions at the output layer based on the 3-dimensional input images at each time step.

III. Types of Attentions:

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. Ashish Vaswani et al [10] states that there are 2 types of Attentions.

1. Scaled Dot-Product Attention
2. Multi-Head Attention

1. Scaled Dot-Product Attention: In this kind of Attention the input consists of queries and keys of dimension d_k, and values of dimension d_v. We compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values. In practice, we compute the attention function on a set of queries simultaneously, packed together into a matrix Q. The keys and values are also packed together into matrices K and V. We compute the matrix of outputs as:

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

2. Multi-Head Attention: Multi-head attention (10) allows the model to jointly attend to information from different representation subspaces at different positions. With a single attention head, averaging inhibits this.

Topical Attention Model:

Xinyi Wang and Yi Yang [11] has proposed TAM model which takes two forms of inputs of documents: a bag-of-word representation for GSM and a sequence of word tokens for RNN. Neural topic model GSM is used to fit document generative process and estimate document-specific topic distribution t. Each sequential word tokens xt is encoded to hidden states ht via the GRU-based sequence encoder. Next, we need to use attention mechanism to bridge two components, so that both models can be jointly optimized. The attention mechanism is originally proposed by (Bahdanau, Cho, and Bengio, 2015) in machine translation. Attention mechanism calculates the similarity between a context vector (query) and each key (word) to obtain the attention score corresponding to the key. The trainable context vector can be seen as a high level representation of a fixed query "what is the informative word" in the sequence.

IV. Conclusion: This paper surveys the types of attention mechanisms that we can use in Autonomous driving. In reinforcement learning single and multi-agent frame works allows us to create an ideal environment for this Autonomous Cars.

V. References:

1. Varundev Suresh Babu and Madhur Behl. "f1tenth. dev-An Open-source ROS based F1/10 Autonomous Racing Simulator". In: 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE). IEEE. 2020, pp. 1614–1620.

2. Mrinal R. Bachute and Javed M. Subhedar. "Autonomous Driving Archi- tectures: Insights of Machine Learning and Deep Learning Algorithms". In: Machine Learning with Applications 6 (2021), p. 100164. ISSN: 2666-8270. DOI: https://doi.org/10.1016/j.mlwa.2021.100164.URL:https://www.sciencedirect.com/science/article/pii/S2666827021000827.

3. Chen Chen et al. Combination of deep neural network with attention mech- anism enhances the explainability of protein contact prediction. 2020.

4. Bangalore Ravi Kiran et al. "Deep Reinforcement Learning for Autonomous Driving: A Survey". In:CoRR abs/2002.00444(2020).arXiv:2002.00444.URL:https://arxiv.org/abs/2002.00444.

5. Liangkai Liu et al. "Computing Systems for Autonomous Driving: State- of-the-Art and Challenges". In: CoRR abs/2009.14349 (2020).arXiv:2009.14349.URL:https://arxiv.org/abs/2009.14349.

6. Alexander Makrigiorgos et al. "Human Visual Attention Prediction Boosts Learning & Performance of Autonomous Driving Agents". In: CoRR abs/1909.05003(2019). arXiv: 1909.05003. URL: http://arxiv.org/abs/1909.05003.

7. Arash Mohammadhasani et al. "Reinforcement Learning Based Safe Deci- sion making for Highway Autonomous Driving". In: CoRR abs/2105.06517 (2021). arXiv: 2105.06517. URL: https://arxiv.org/abs/2105.06517.

8. Gon¸calo Neto. From Single-Agent to Multi-Agent Reinforcement Learning: Foundational Concepts and Methods Learning Theory Course. 2005.

9. Aizaz Sharif and Dusica Marijan. Evaluating the Robustness of Deep Rein- forcement learning for Autonomous and Adversarial Policies in a Multi- agent Urban Driving Environment. 2021. arXiv: 2112.11947 [cs.AI].

10. Ashish Vaswani et al. "Attention is All You Need". In: 2017. URL: https://arxiv.org/pdf/1706.03762.pdf.

11. Xinyi Wang and Yi Yang. "Neural Topic Model with Attention for Super- vised Learning". In: Proceedings of the Twenty Third International Con- ference on Artificial Intelligence and Statistics. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108. Proceedings of Machine Learning Research. PMLR, Aug. 2020, pp. 1147–1156. URL: https://proceedings.mlr.pres s/v108/wang20c.html.

12. S. J. Russell and P. Norvig, Artificial intelligence: a modern approach (3rd edition). Prentice Hall, 2009.

13. Bahdanau, D.,Cho, K., and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In Proceedings of ICLR.