Performance Evaluation of Attention Based Neural Network Models through Loss Function

Anand Tumma
Assistant Professor
Department Of CSE
Telangana University

Prof. Arathi Chitla
Department Of CSE
Telangana University

Abstract: Neural Networks has different type of attention mechanisms to improvise the autonomous driving. Due to the existence of various attention mechanisms, a comprehensive evaluation of the performance of each model is required to determine their effectiveness. This paper describes few attention mechanisms and evaluates them based on the loss function.
Key Words: Attention, Brake, Epoch, Loss Function, MAE, MSE, Speed, Throttle.

1. Introduction: Autonomous driving is a very actual and important topic that reforms the car industry. 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 is the use of Neural networks. Neural networks [2] allow us to train an agent in an efficient manner with a wide variety of its mechanisms. One of such mechanisms is Attention. Since humans are able to cope with complex driving scenarios, a human inspired attention mechanism could improve autonomous driving.

Attention mechanism [7] is an Encoder-Decoder kind of neural network architecture that allows the model to focus on specific sections of the input while executing a task. It dynamically assigns weights to different elements in the input, indicating their relative importance or relevance.

- The encoder processes the input sequence and represents it as a fixed-length vector (context vector), which is then passed to the decoder.
- The decoder uses this fixed-length context vector to generate the output sequence.

2. Attention mechanisms: There are various types of attention mechanisms like Self-attention, scaled dot-product attention and multi-head attention.

i) Self-attention: Self-attention is categorised into two types i.e. Single Head attention and Multi-Head Attention. Self-Attention is a neural network mechanism that models intra-sequence dependencies by computing attention weights between all pairs of elements within the same input sequence using learned **Query**, **Key** and **Value** projections.

For an input matrix **M**:

$Q = MW_Q,\ K = MW_K,\ V = MW_V$

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

Attention based on Softmax function [1]

ii) Scaled Dot-product attention: In Scaled Dot-Product attention Q, K, V values can be taken from same input frame or different. It computes attention scores using a scaled dot product. Scaled Dot-Product Attention [9] computes attention using Q = K = V without learned projections. It is included as a lightweight baseline to compare the effect of learnable attention parameters.

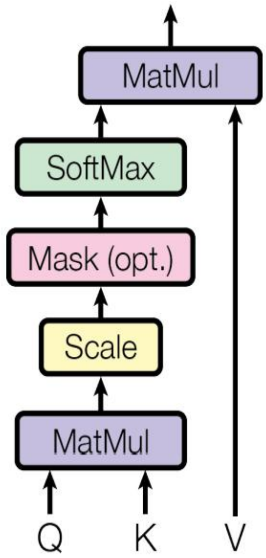


Figure: Scaled Dot-Product Attention [1]

Scaled dot-product attention is the mathematical method used to compute attention weights. It defines how similarity between elements is calculated. It works as follows:

1. The query vector is compared with key vectors using a dot product.
2. The result is divided by the square root of the key dimension(d_k) (this is the “scaling” part).
3. A softmax function [1] converts these scores into probabilities.
4. These probabilities are used to compute a weighted sum of value vectors.

iii) Multi-Head Attention: Multi-Head Attention consists of several attention layers running in parallel. It is the extension of single Head

Attention. Single Attention calculates Attention once but Multi-Head Attention calculates it multiple times based on number of Heads.

The input is projected into multiple sets of heads. Every Head consists of Queries (Q), Keys (K), Values (V). Each head performs scaled dot-product attention independently. The outputs of all heads are Concatenated, linearly transformed, Combined into one final output.

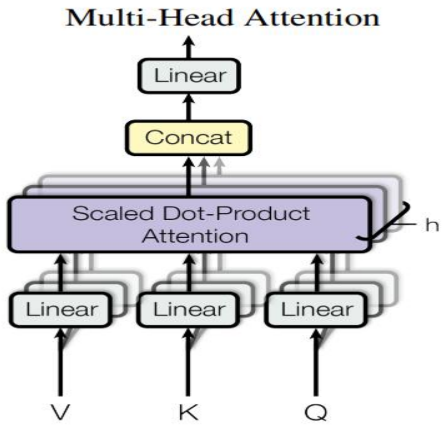


Figure: Multi-Head Attention [1]

3. Loss Function: A loss function is a mathematical function which calculates the deviation between model's predictions and the actual (true) values. It is the function used to tell the model **how wrong it is**. The main purpose of calculating loss function during training is to **minimize this loss** so the model becomes more accurate while predicting something.

Training a Neural Network model is as follows:

1. The model makes a prediction.
2. The loss function compares the prediction with the true label.
3. It produces a numerical error value.
4. This error is used to update the model weights (via back propagation).

Without a loss function, the model would not know how to improve.

Loss function is also used regression related problems. Two widely used Loss functions are MSE and MAE.

MSE (Mean Squared Error): MSE [16] is the average of the squared differences between the predicted and actual values. It is calculated with the following formula.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathrm{V_a} - \mathrm{V_P})^2$$

$\mathbf{V_a}$ is the Actual value and $\mathbf{V_p}$ is the Predicted value. The number of observations used is represented with 'n'.

MAE (Mean Absolute Error): MAE [17] is a Loss function which deals with absolute values. MAE mathematically represented as follows.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\mathrm{V_a} - \mathrm{V_P}|$$

4. Evaluation: To assess the best Attention mechanism some simulation software must be used. One of the best Simulation software available as open source is CARLA [2]. The environment of CARLA should be captured to evaluate the three attention mechanisms (Self-Attention, Scaled Dot-Product attention and Multi-Head attention) so that the trained model can react to the upcoming events in effective manner.

Loss function can be categorised into two types based on the dataset on which it is calculated.

a. Train Loss: The difference between the actual value and the expected value that is calculated on the training dataset during training session is called Train loss.

$$Train\ Loss = MSE_{speed} + MSE_{throttle} + MSE_{brake}$$

The sum of errors of speed, throttle and brake gives train loss. The model tries to reduce this loss.

b. Validation Loss: The value of the loss function calculated on validation dataset during validation is called as Validation loss.

$$Validation\ Loss = MSE_{speed} + MSE_{throttle} + MSE_{brake}$$

The validation loss is the sum of errors of speed, throttle and brake during validation session.

While calculating Train loss and validation loss the following cases may arise.

I. **Underfitting:** In this case both the train and validation loss will be high which says that the model is not properly trained and not working well on new data.

II. **Generalized:** When the model gives minimal train loss and minimal validation loss then the model is said as generalised model (robust or best model).

III. **Overfitting:** It is the case which gives lesser train loss and high validation loss. Overfitting means model trained properly but not working well on unseen data.

Autopilot is a built-in feature of CARLA [15] which acts as an expert driver and it can be used for research-oriented activities.

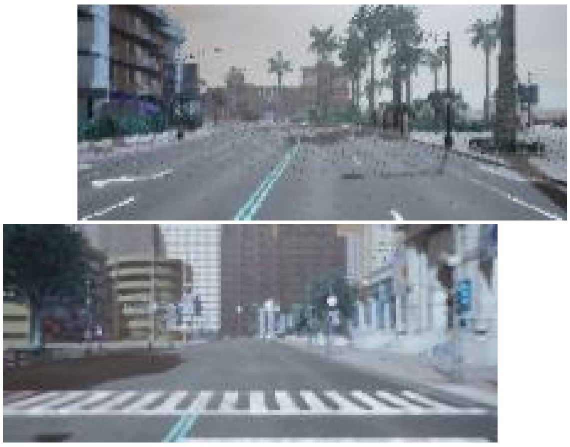
When autopilot is enabled, the vehicle can:

- Follow lanes
- Obey traffic lights
- Avoid collisions
- Navigate through roads
- Interact with other vehicles and pedestrians

The following code is used to enable autopilot mode.

vehicle.set_autopilot(True)

The images captured from CARLA are split into two parts. One is for training and another is used for evaluation. From CARLA environment by using RGB camera 8000 images are captured from which 6400(80%) are used for training and remaining 1600(20%) are used to evaluation.



Sample images captured from CARLA environment.

For every attention mechanism 20 epochs are done. The best model is saved whenever the validation loss is lesser than the train loss or whenever the present epoch's validation loss is lesser than the previous epoch's validation loss. By doing this the best model of every attention mechanism is saved.

Model evaluated	evaluated for	**MSE**	**MAE**
Self-Attention	**Steer**	0.0008	0.0154
	Throttle	0.0038	0.0318
	Brake	0.0027	0.0161
Multi-Head Attention	**Steer**	0.0008	0.0136
	Throttle	0.0078	0.0453
	Brake	0.0041	0.0215
Scaled Dot-Product Attention	**Steer**	0.0008	0.0174
	Throttle	0.0052	0.0362
	Brake	0.0046	0.0283

Table: Observed MSE, MAE values

The above values are obtained after performing 20 epoches for every attention model and evaluating them. Multi-Head attention has higher error values compared with the remaining mechanisms. Especially the throttle of Multi-Head attention has highest error as 0.0078. The steering MSE is almost same for all the mechanisms. Self-Attention showed very less error values in predicting steering, Throttle and Brake when compared with Multi-Head and Scaled Dot-Product Attention. So the Self-Attention(Learned Q,K,V) is the best model in terms of Steer, Throttle and Brake.

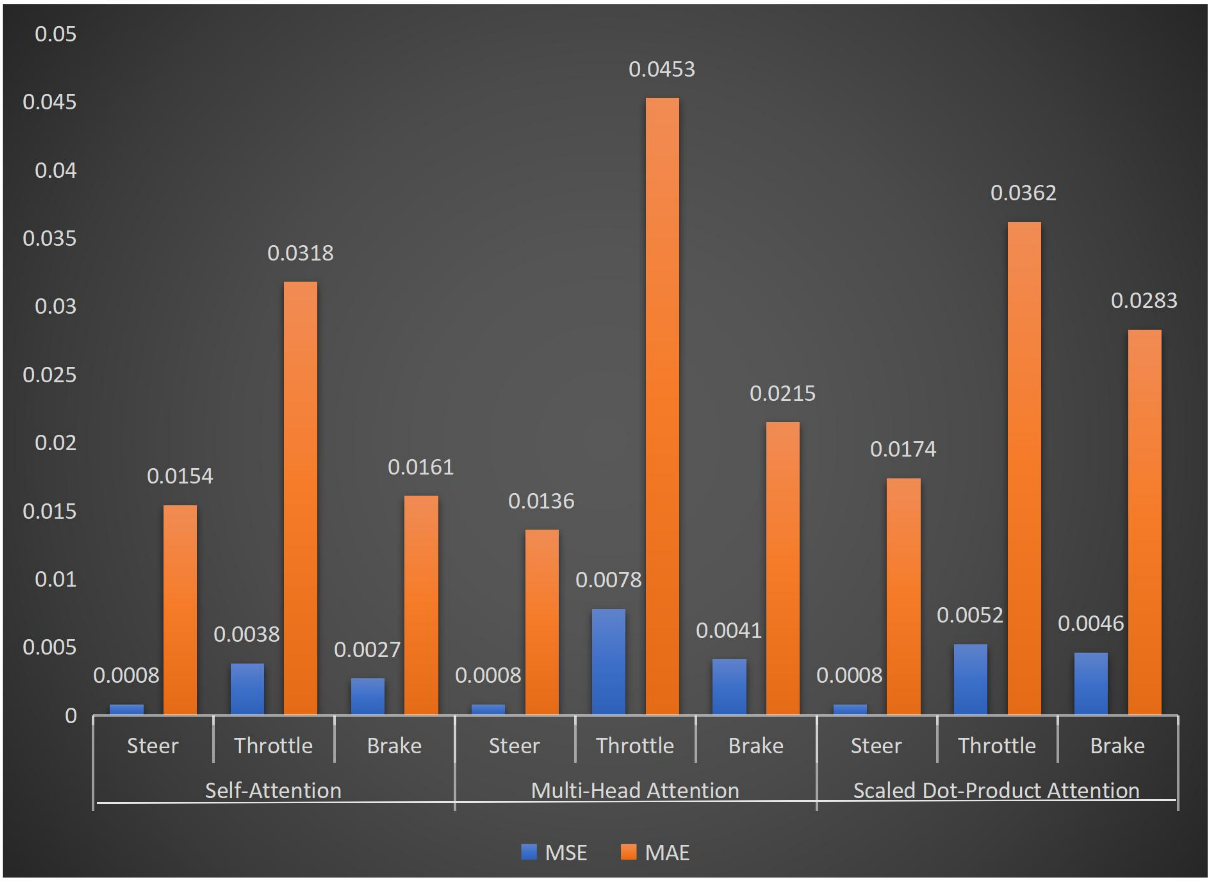


Chart: Graphical representation of Loss Function values

5. Conclusion: Attention based Neural Networks is an efficient approach to conduct autonomous driving. Amongst all attention Neural Network models Self-Attention with learned Q, K, V projections [1] show relatively good performance. It possesses less error values (MSE and MAE values) while predicting the upcoming events.

6. References:

[1]. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin “Attention Is All You Need” arXiv-2017, arXiv:1706.03762.

[2]. Anand Tumma, Arathi Chitla “Autonomous Driving Simulators: The advantages of CARLA” International Journal of Computer Applications (0975 – 8887) Volume 187 – No.19, June 2025.

[3]. 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.
[4]. Mrinal R. Bachute and Javed M. Subhedar. "Autonomous Driving Archi- tectures: Insights of Machine Learning and Deep Learning Algorithms". In: Machine Learning with Applications volume-6 (2021),p.100164.ISSN:2666-8270. DOI:https://doi.org/10.1016/j.mlwa.2021.100164.
[5]. Chen Chen et al. Combination of deep neural network with attention mechanism enhances the explainability of protein contact prediction. 2020.
[6]. 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.
[7]. Alexander Makrigiorgos et al. "Human Visual Attention Prediction Boosts Learning & Performance of Autonomous Driving Agents". In: CoRRabs/1909.05003(2019).arXiv:1909.05003.URL:http://arxiv.org/abs/1909.05003.
[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 Reinforcement Learning for Autonomous and Adversarial Policies in a Multi- agent Urban Driving Environment. 2021. arXiv: 2112.11947 [cs.AI].
[10]. Xinyi Wang and Yi Yang. "Neural Topic Model with Attention for Super- vised Learning". In: Proceedings of the Twenty Third International Conference 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.
[11]. Mohsin Raza "Autonomous Vehicles: Levels, Technologies, Impacts and Concerns" International Journal of Applied Engineering Research ISSN 0973 4562 Volume 13, Number 16 (2018) pp. 12710-12714.
[12]. Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez and Vladlen Koltun "CARLA: An Open Urban Driving Simulator" arXiv:1711.03938v1 [cs.LG] 10 Nov 2017.
[13]. Sumbal Malika, Manzoor Ahmed Khana, Hesham El Sayed "CARLA: Car learning to Act-An inside out" International Workshop on Smart

Communication and Autonomous Driving (SCAD 2021) November1-4, 2021, Leuven, Belgium.

[14]. Symphorien Karl Yoki Donzia, Haeng-Kon Kim "A Study on Autonomous Driving Simulation Using a Deep Learning Process Model" International Journal of Software Innovation Volume 10 Issue 1.

[15]. Emily Barbour, Kevin McFall "AUTONOMOUS VEHICLE SIMULATION USING OPEN-SOURCE SOFTWARE CARLA" Journal of UAB ECTC Volume 18, 2019.

[16]. Li, C.; Liu, K.; Liu, S. "A Survey of Loss Functions in Deep Learning". Mathematics 2025, 13, 2417. https://doi.org/10.3390/math13152417.

[17].Lorenzo Ciampiconi, Adam Elwood, Marco Leonardi, Ashraf Mohamed, Alessandro Rozza "A survey and taxonomy of loss functions in machine learning" arXiv-2023, arXiv:2301.05579.