A Journey with Artificial Neural Network from Classification to Control

Ezra Castillo, DesCyPhy Lab
Benjamin Franklin High School, Class of 2024
USC Viterbi Department of Electrical Engineering, SHINE 2023

Introduction

Neural network-based techniques have demonstrated exceptional effectiveness in acquiring task-specific knowledge, surpassing human performance in challenging classification problems [1]. However, integrating them into autonomous driving systems remains challenging due to concerns about their unrobust nature [2]. This paper focuses on constructing neural networks tailored for the Car Racing environment, aiming to explore their potential for achieving human-level performance and providing insights for autonomous driving applications in the future.

Architecture of a Neural Network

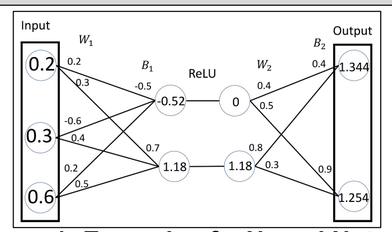

Figure 1: Example of a Neural Network

A neural network comprises an input layer, one or more hidden layers, and an output layer. The input layer receives data, which is processed by neurons in the network. Neurons use learnable weights to extract crucial features and capture complex relationships from the input data. Ultimately, the output layer generates predictions based on this transformed information, representing acquired knowledge and facilitating task fulfillment.

In the provided example, we start with inputs X = [0.2, 0.3, 0.6]. These inputs are multiplied by the weight matrix W_1 of the first layer and summed with the bias vector $B1$ as follows:

$$W_1X + B_1 = \begin{bmatrix} 0.2 & -0.6 & 0.2 \\ 0.3 & 0.4 & 0.5 \end{bmatrix} \begin{bmatrix} 0.2 \\ 0.3 \\ 0.6 \end{bmatrix} + \begin{bmatrix} -0.5 \\ 0.7 \end{bmatrix} = \begin{bmatrix} -0.52 \\ 1.18 \end{bmatrix}$$

The resulting values are passed through the Rectified Linear Unit (ReLU) activation function, denoted as α (x)=max(x,0), which outputs positive values directly and sets negative values to zero. The outputs of the ReLU activation are then multiplied by the weight matrix W_2 of the second layer, summed with the bias vector B_2 and produce the final output of the neural network as follows:

$$W_2\alpha(W_1X + B_1) + B_2 = \begin{bmatrix} 0.4 & 0.8 \\ 0.5 & 0.3 \end{bmatrix} \begin{bmatrix} 0 \\ 1.18 \end{bmatrix} + \begin{bmatrix} 0.4 \\ 0.9 \end{bmatrix} = \begin{bmatrix} 1.344 \\ 1.254 \end{bmatrix}$$

Empirical Environments

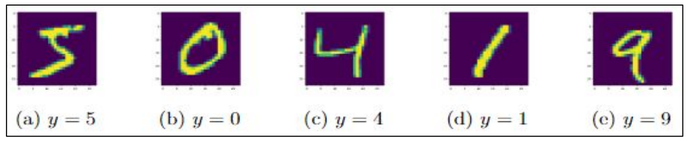

Figure 2: First Five MNIST Dataset Figures

The dataset used for the project is the MNIST dataset [3], imported from PyTorch [4]. The dataset consists of handwritten digits, and the first five figures along with their corresponding labels, denoted as *y,* are shown in Figure 2.

The Car-Racing [5] environment is utilized in the project. Figure 3 depicts an illustration of the Car-Racing scenario.

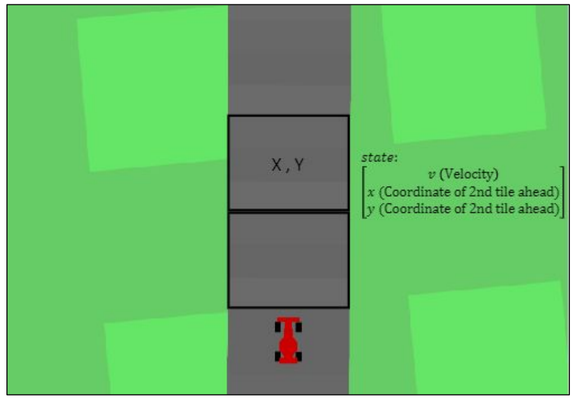

Figure 3: Car-Racing

Original State Representation

The original state representation in the Car-Racing environment refers to the image extracted from the current state at a specific time. This image serves as the default setting in the Car-Racing environment.
However, it is important to note that the accuracy and reliability of image state representations can be influenced by the robustness of perception networks.

Compact State Representation

The compact state representation refers to a concise method used to represent the state of the car. As shown in Figure 3, the state is represented as $[v, x, y]^T$, where *v* denotes velocity and *x* and *y* represent the car's coordinates.

Case Study & Results

Case Study 1: MNIST

In the Mnist experiment, we tested different coefficients to observe their influence on the outputted data.
Specifically, we experimented with different layers, neurons, epochs, batch sizes, and learning rates, denoted as *lr*.

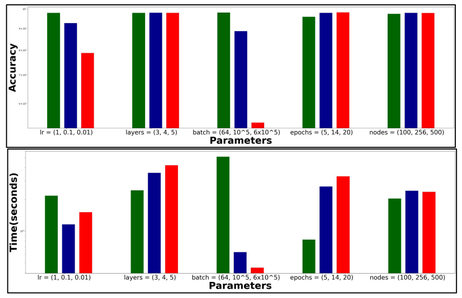

Figure 4: MNIST Experiment Results

Here are the key observations:

- Learning Rate: Higher learning rates generally lead to better training progress and improved accuracy. However, excessively high learning rates can hinder convergence.
- Number of Layers: The number of layers does not have a significant impact on accuracy, but more layers increase the training time.
- Number of Epochs: Increasing the number of epochs generally improves the accuracy of the output. However, longer training times are required as more epochs are used.
- Batch Size: Larger batch sizes result in faster training times, but they may lead to slightly less accurate results compared to smaller batch sizes.
- Number of Nodes: Increasing the number of nodes in the network generally improves both accuracy and training times.

Case Study 2: Car-Racing Experiment

After playing the Car-Racing game and testing the model that achieved the best average score after 100 episodes (consisting of 1,000,000 steps with an architecture of [3, 256, 256, 5]), I have identified that car speed and accuracy in staying within the lanes of the road are critical factors influencing the player's final score. To achieve a high score, players must maintain a high speed while also ensuring control to prevent the car from spinning out of the lane.

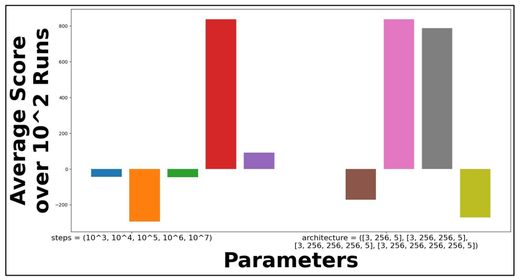

Figure 5: Car-Racing Experiment Results

Results Analysis

Upon observing the models performance in three episodes, I have determined the following

- The model achieved an average score of 829.59, whereas my average score was 878.17.
- The model performed similarly but not as well as my own performance in the driving simulator.
- The model exhibited reasonable accuracy in turning, but it tended to lose control and drive off the track more frequently.

These results indicate that while the model performed reasonably well, there is still room for improvement to match or surpass human performance in the Car-Racing game.

Next Steps for You OR Advice for Future SHINE Students

During my first couple weeks in the lab, being exposed to PhD level research was very intimidating. I was exposed to topics that I had no prior experience of, forcing me to become familiar with them on the spot. My advice for future SHINE students is to keep yourself from feeling discouraged and instead try to learn as much as you can throughout the program. Make sure to ask lots of questions and prioritize the help that your mentors can provide to you. Now that I am exposed to machine learning and programming, I plan to continue my own research in the future to create something innovative using the valuable skills I gained throughout the summer.

Acknowledgements

I would like to express my deepest appreciation to my lab mentor Kevin Chang and center mentors Matthew Ai and Marcus Gutierez for guiding and supporting me throughout my research. I would also like to thank Prof. Nuzzo, TELACU UB, and the SHINE program for giving me the opportunity to come conduct research at USC this summer.

Citations

[1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, 2016.
[2] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions,"*Journal of big Data*, vol. 8, pp. 1–74, 2021.
[3] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE signal processing magazine*, vol. 29, no. 6, pp. 141–142, 2012.
[4] S. Imambi, K. B. Prakash, and G. Kanagachidambaresan, "Pytorch," *Programming with TensorFlow: Solution for Edge Computing Applications*, pp.87–104, 2021
[5] C. Li, "Challenging on car racing problem from openai gym," *arXiv preprint arXiv:1911.04868*, 2019.