Deep Learning Final Project

Jesse A. Comer — Mingni H. Cui — Omar Faruqi — Tan Xian Wei

1 Introduction

We present a vision & controller solution that, in local testing, yielded an average of 1-1 goals per game against AI, and in the August 8th tournament, we scored 1.75 goals on average against AI. Videos of one of our most successful 1000-steps games against AI is included in `writeup`: agent0.mp4[1] and agent2.mp4[2].

During the bulk of our development period, one team member worked on a controller using Reinforcement Learning (RL) and a defined policy. Another person worked on developing the controller using the true locations of key objects (x-dimension on a range of $[-45.6, 45.6]$, y-dimension on a range of $[-64.5, 64.5]$). A third person developed code to pull training data based on the controller, while the final person worked on an Fully Convolutional Network (FCN) that tried to predict true map coordinates.

During early development, we struggled to find the correct transformation for coordinates in the kart's field of view into actual map coordinates because the vision of the kart is not in bird's eye view. This hindered the accuracy of our FCN and with one week remaining, we began concentrating on a solution that relied on coordinates relative to the kart instead (herein, image coordinate).

After switching to image coordinates, we quickly realized the `_to_image` function given in Homework 5 to output image coordinates was insufficient for our final solution. When the puck is out of the kart's field of view, the projected point from `_to_image` became the coordinate mirrored along the x-axis. Thus, we decided to use a CNN to first identify if the puck is in the image. The controller only calls for the FCN predicted image coordinate if the CNN predicts that the puck is already in frame.

During final stages in development, we struggled with finding the balance in the trade-off between run-time and accuracy. Our most accurate FCN model was 10x slower than the Homework 5 master solution. To troubleshoot, one teammate attempted to reduce the two-model solution while another teammate created a model that would predict both presence and location of the puck by balancing two loss functions in a way similar to Homework 4. The latter was not selected due to low validation accuracy. We also grappled with finding a balanced set of training images for our CNN so that our model would have examples the puck at many different states and at many different distances from the kart. In the end, we introduced used AI at various difficulty levels and several controllers to pull the training data that yielded our final models.

2 Model and Strategy

Our submitted project uses a CNN trained on `render_data[i].instance` and the puck's in-game object ID to identify if the puck is in the field of view, and a FCN with 4 layers, 2 `conv2d-batchnorm-relu` blocks per layer used from Homework 5 to predict the $[-1, 1]$ image coordinate if the puck is in the driver's field of view.

The controller uses the x-coordinate of the predicted image coordinate to steer. The driver accelerates less if the puck is sufficiently close to him. If the puck is *very* close to the driver, he stops using the FCN-predicted aim point to steer, and instead, uses the angle between two vectors: the vector from the kart to the puck, and the vector from the kart to the goal. The angle between these two vectors is calculated using the following and

[1] `https://drive.google.com/file/d/1xrA3_yvkM8DrOWpGMjObvHlrTzD8W2b3/view?usp=sharing`
[2] `https://drive.google.com/file/d/1OadCLRZ_2bJweniyaXHfqKEJ9a5A7x4s/view?usp=sharing`

multiplied by -1 if the $v2$ is to the left of $v1$:

$$\theta = \frac{\pi}{2} - arcsin\left(\frac{v1 \cdot v2}{||v1|| \cdot ||v2||}\right)$$

If the puck is predicted to not be in the kart's field of view, the kart begins backing up and spinning around to find the puck. Also, when the puck is out of sight, rescue is implemented with probability following $Bin(p = 0.01)$ to avoid the kart getting stuck anywhere.

3 Other Attempted Strategies

Our main competing strategy was using reinforcement learning based on a defined policy.

The first implementation of reinforcement learning we attempted relies on an aim point generated using a FCN model which provides the relative position of the puck on the field-of-view image provided by the kart. The reinforcement learning agent is guided by a linear model which takes as input a tensor of shape 1×3 and includes the x and y coordinates along with the velocity of the kart. The model outputs the probability of braking and the probability of steering in a specific direction. The second implementation used a CNN model similar to the one provided to us as a solution to Homework 3. The third implementation used a FCN model similar to the one provided to us as a solution to Homework 4. Both models take as input the field-of-view image from the kart and similar to the aim-point implementation, both the CNN and FCN implementation output the probability of braking and the probability of steering in a specific direction.

All implementations were tested using the same number of episodes, maximum number of steps per episode, opposition agents and rewards function. The agent was encouraged to move towards the puck and move the puck towards the goal with a positive reward and discouraged to move away from the puck and move the puck away from the goal with a negative reward. Unfortunately, all implementations were unable to consistently score a goal. We attempted numerous techniques to enable faster learning for the agent. We removed the AI agent as the opposition in order to remove variance during the learning process. For each episode, we tried randomly initializing the location of the kart to help generalize the solution. We also trained the agent after removing the braking probability output and noticed improved results. The drawback of this solution was the kart's inability to escape when stuck next to a wall.

4 Future Steps

If we had more time, we would've like to develop our controller more. One issue that is unresolved is how to deal with the puck when it's *very* close to our own goal. We struggled to find an aim point that is between the puck and our own net. As a result, instead of driving to a point where we can push the puck away from our own net, we just aim directly at the puck and thus, cannot prevent our players from scoring on our own goal.

A future step to improve upon our current strategy is to employ a defense role: one main shooter player and supporting defense player. The supporting player can pick up items and try to sabotage the opposing team. This would required additional FCNs that identify enemy players and items to pick up. Having this supporting player could help crowd out the playing field when the main player is attempting to gain control of the puck. Unfortunately, we ran out of time before we could try to implement this supporting player.

Moving forward, we hope to continue to learn more about how the development of solutions in real life, like the self-driving car, integrate both vision and imitation learning into their models.