

## Neural Network Based AI for Racing Games

### Work Done:

- **Coding Work:**

- I implemented John's suggestion using waypoint number as input, assuming that most of sharp turn that needs to use handbrake is in the waypoint. The result is better on average, but still I feel it does not mean that it is really better, since a lot of factor affects the car, but only a few factor considered in training which actually makes the training result seems random.
- I tweaked the code a bit, now the network gives out the output **after** the RaceNavigation function. Test result: Tweaked code result is named under Network II column on the table.

	Without Network	With Network	With Network II
1	67.144432	67.598412	69.90937
2	70.282356	68.239166	69.215881
3	70.202522	67.658318	68.06041
4	66.431091	71.152534	68.267876
5	69.680733	68.282349	69.119652
6	68.3592	67.020981	68.681244
7	67.5896	70.361359	66.728638
8	67.386101	67.338303	69.328011
9	68.749794	68.690628	67.240036
10	70.952805	68.204094	68.142059
11	69.588142	68.585396	68.215416
12	66.807945	68.984482	69.358612
13	68.887321	68.152542	71.021011
14	68.886612	66.484116	68.339691
15	67.638908	66.753563	70.624733
16	67.403976	64.915474	68.111298
17	65.645363	66.235954	68.017853
18	67.822258	67.46608	72.389481
19	66.308586	69.328087	66.542747
20	68.75473	66.749321	70.02018
21	68.593521	68.77198	67.997749
22	69.196724	68.539925	67.449059
23	69.907768	69.557793	68.778412
24	66.397911	69.559555	66.699333
25	68.216591	68.419701	69.235634

<b>AVG</b>	<b>68.2733996</b>	<b>68.12200452</b>	<b>68.69977544</b>
<b>BEST</b>	<b>65.645363</b>	<b>64.915474</b>	<b>66.542747</b>
<b>WORST</b>	<b>70.952805</b>	<b>71.152534</b>	<b>72.389481</b>

Table 1: Training Result: Waypoint as Input

Observation:

Training result is somehow not reliable due to same input may result in different outputs which the network won't be able to recognize. Possibility: need to go down to physics level for the network to recognize the pattern

- I completely replaced the handbrake system in the AI. Rewrote the code with Quang's old code with some modification. It is a rule base decision to decide to use or not to use the handbrake, and the handbrake time is fixed with some constant value that is achieved via trial and error for optimization. I further extended the hardcoded rule base method with Neural Network. The decision is still achieved via rule base method, but the handbrake time is decided by the network. Inputs of the network are speed error and heading error.
- I modified Track2.ini for Nevada map, increasing the target speed in all nodes (except those that are already at the maximum speed limit) by 10, hoping that by using faster target speed, if the new handbrake system is able to cope with that faster speed, the AI may race through the track faster as well.

- Test result:

- Original Nevada's Track2.ini

	Original	Hardcode	NN
1	131.533966	126.680634	124.623840
2	133.839661	128.841965	127.282021
3	129.091797	127.822289	128.929932
4	133.675888	126.381874	123.984398
5	134.735443	127.772354	124.188789
<b>Average</b>	<b>132.575351</b>	<b>127.499823</b>	<b>125.801796</b>

- Modified Nevada's Track2.ini

	Original	Hardcode	NN
1	129.887497	127.047714	122.173851
2	129.859756	123.509201	124.920464
3	127.432320	126.858231	123.439316
4	125.507607	128.209793	122.048103
5	128.580032	127.287064	122.676430
6	129.593674	127.000725	125.103714

7	131.031372	126.436211	134.557602
8	128.049149	125.269432	130.799744
9	129.169510	123.682404	122.322540
10	129.225754	130.632309	122.513412
<b>Average</b>	<b>128.833667</b>	<b>126.593308</b>	<b>125.055518</b>

Table 2: New Handbrake System Test Result

Observation:

Due to high speed and bad positioning before the very sharp turn, it may go out of the track and in some rare cases it may turn upside down and need to wait for the reset timer (case 7 and 8 in the training result table). The resulting handbrake system works quite badly when it was tested at Sedona's Track2.ini. It means that it may work for specific track only.

- Tested previous training result in many different tracks and recorded down the performance comparison between the original AI and AI with Neural Network Handbrake System. Test maps: Sedona (Activity 1-5) and Nevada (Activity 1-5).

Observation:

Out of 10 maps tested, Neural Network performed better in 6 maps. It performed on par in 1 map, and it performed worse in 3 maps. Possible reason: The braking system are only based on heading error and speed error, it won't know whether the turn is short or long, or whether the road is wide enough or not (e.g. if there are multiple short sharp turns consecutively or a sharp turn on the edge of the cliff). So sometimes the neural network will drift in the situation near the edge of the cliff which are not wide enough and then fall off the cliff.

**To Do:**

- To tweak to code so that the handbrake system that may work well for any kind of terrain and map and be able to overcome the cliff situation.