Lab Report

The training loop for the pid- and the neural approach are identical with one notable exception. For the PID models to reach convergence, the number of time steps is increasing given as a function of the current epoch.

$$n_{timesteps} = (epoch + 1) \cdot timestep\ increase$$

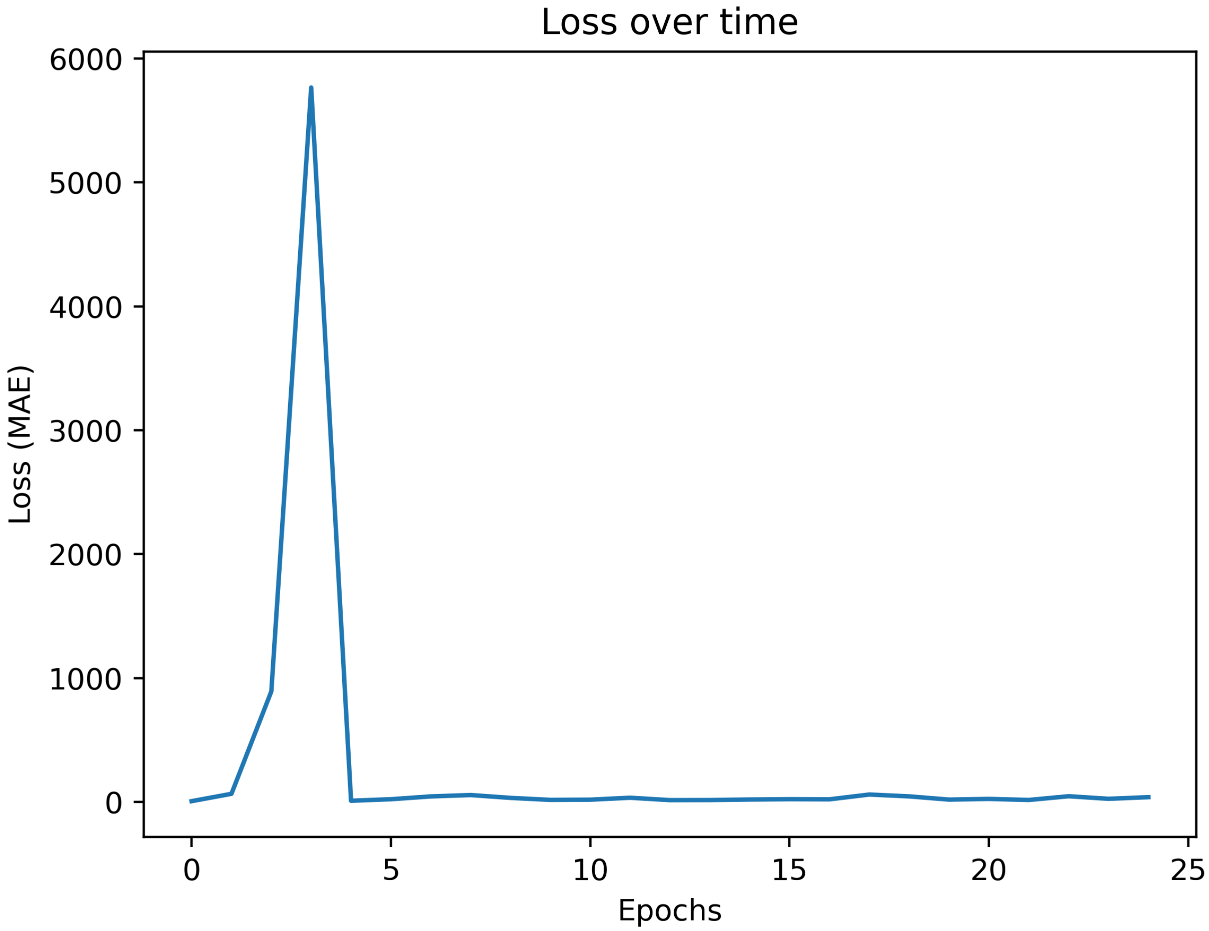
The neural approach runs the full epoch every time.

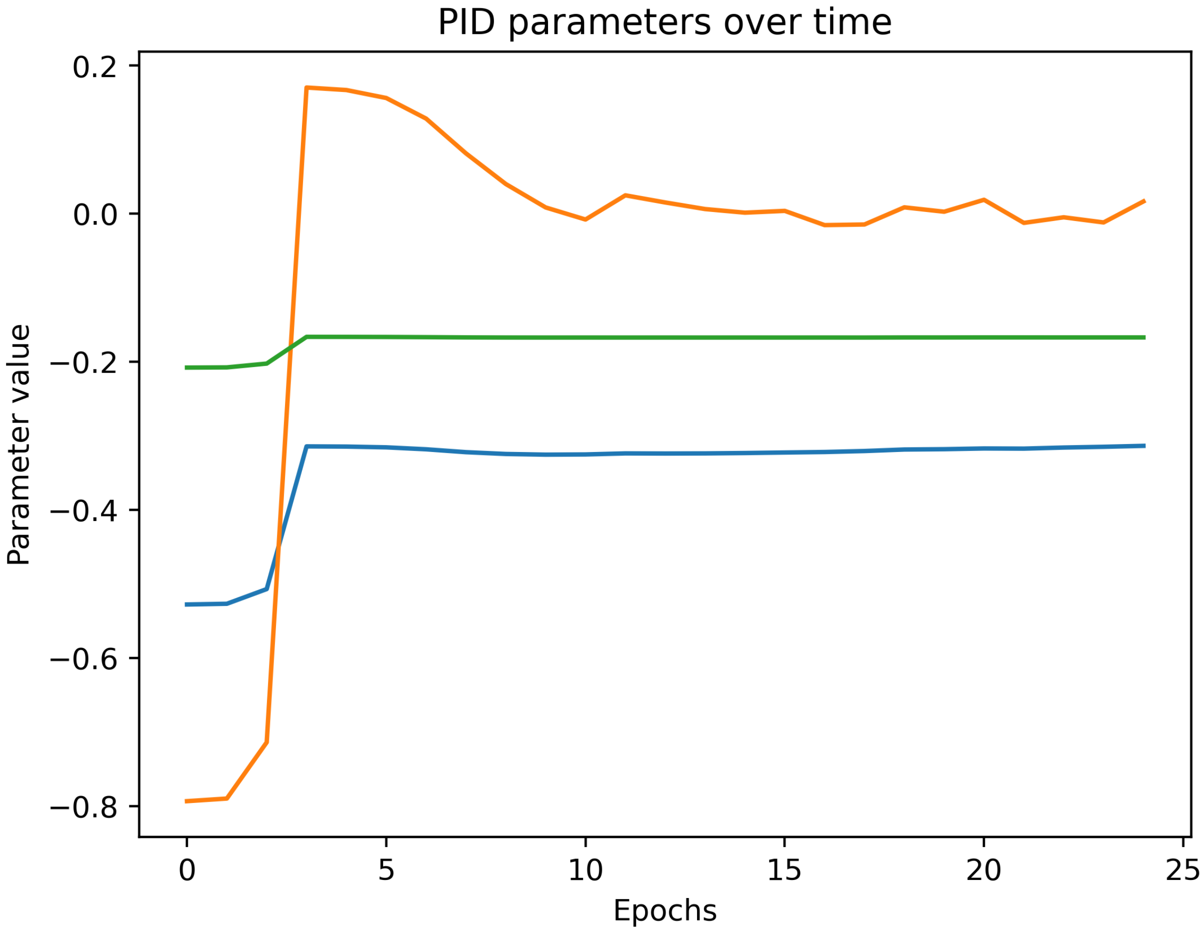
Hyperparameter matrix

Hyperparameter	Tub PID	Tub NN	Cournot PID	Cournot NN	Car PID	Car NN
Layer sizes		[3, 3, 1]		[3, 3, 1]		[3, 3, 1]
Activation		ReLU		ReLU		ReLU
Initial Params Max	1	1	1	1	1	1
Initial Params Min	-1	-1	-1	-1	-1	-1
Epochs	26	50	5	100	25	50
Max timesteps	$(e_{id} + 1) \cdot 10$	250	$(e_{id} + 1) \cdot 1$	1000	$(e_{id} + 1) \cdot 5$	250
Disturbance Max	0.1	1	0.1	0.1	0.1	1
Disturbance Min	-0.1	-1	-0.1	-0.1	-0.1	-1
Learning rate	0.00001	0.01	0.00001	0.01	0.00001	0.01
H init	100	100				
A	10	10				
C	0.1	0.1				
T			10	10		
P max			10	10		
cm			0.1	0.1		
Desired speed					100	100
μ					0.01	0.01
m					1	1
timestep increase	10		1		5	

Bathtub

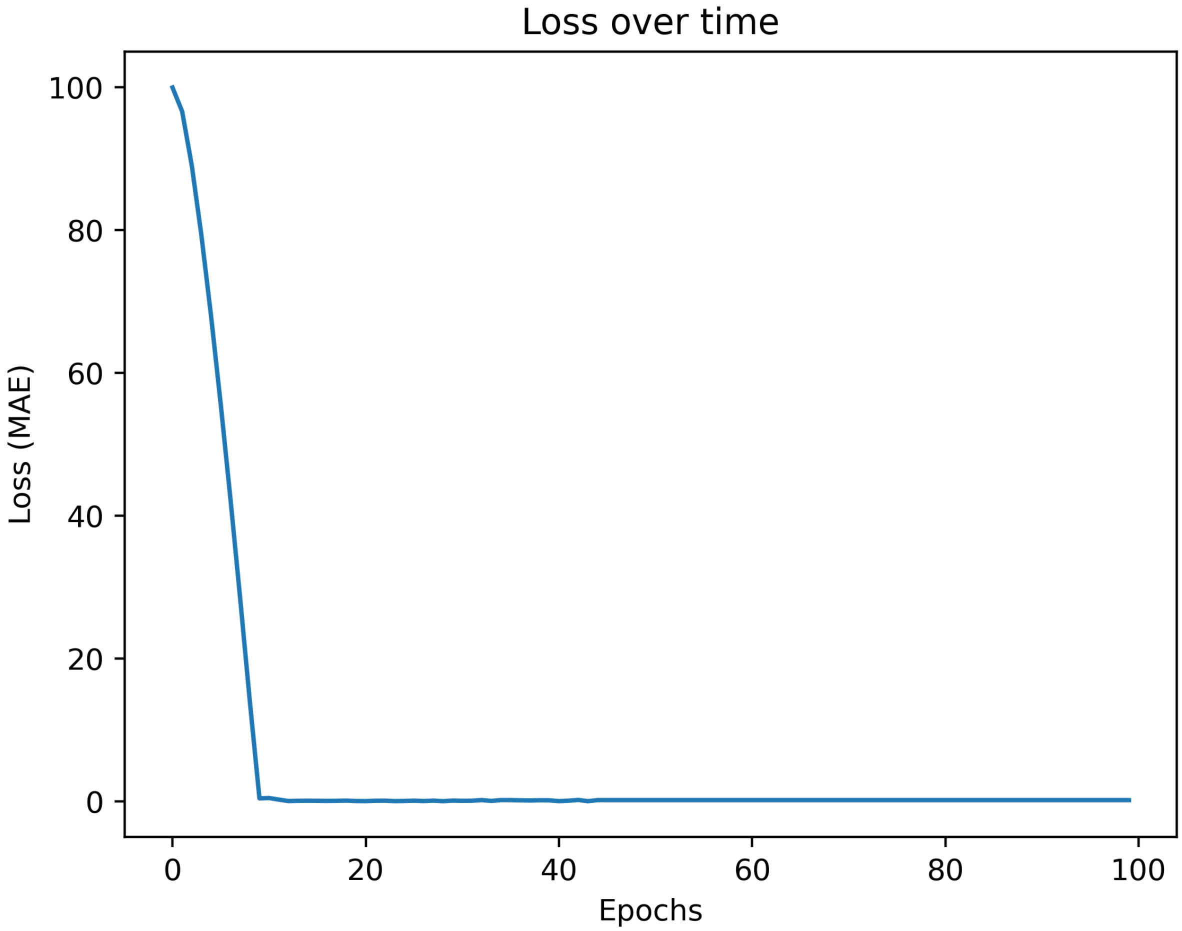
PID


Loss over time
Loss (MAE)
6000
5000
4000
3000
2000
1000
0
0
5
10
15
20
25
Epochs



After performing 100 epochs, the PID is able to make the water height oscillate around the correct water level.

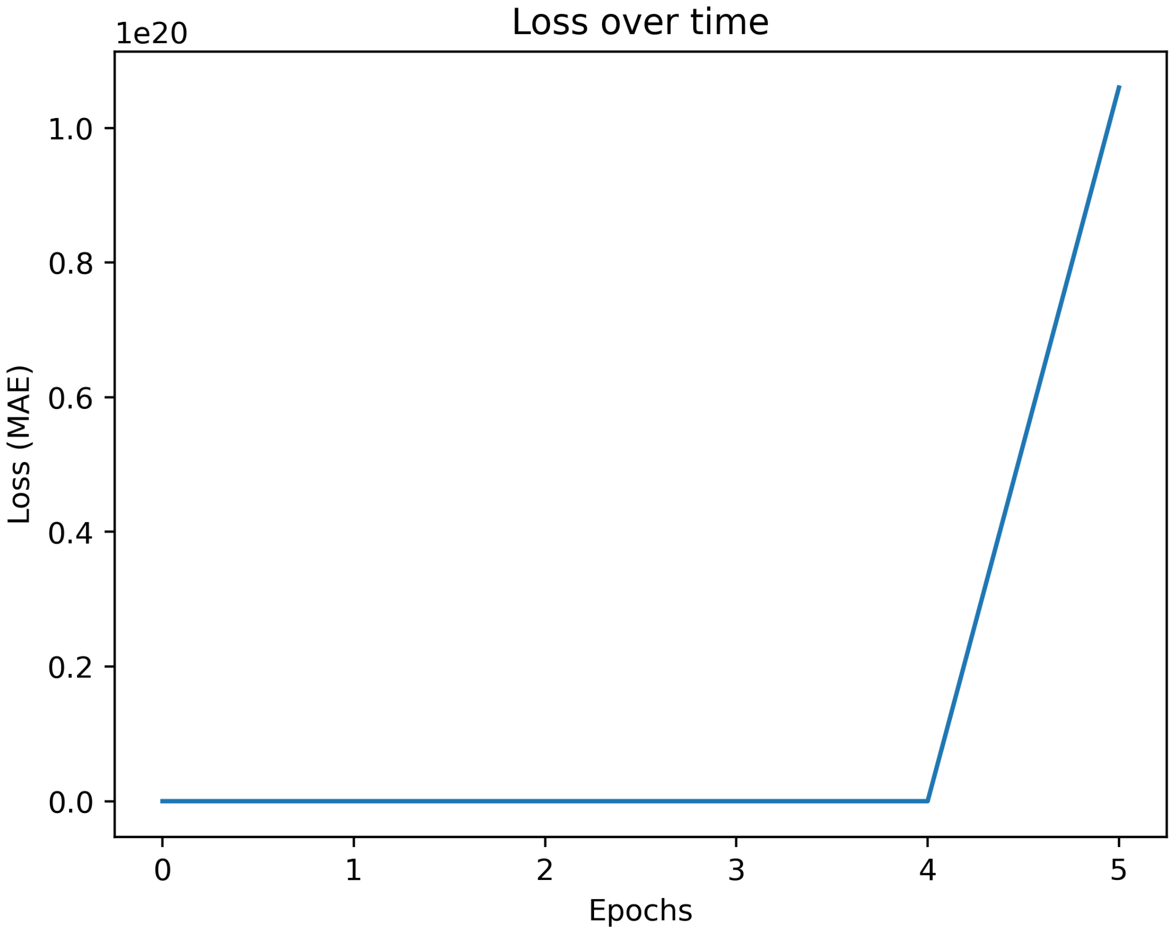
Neural Network

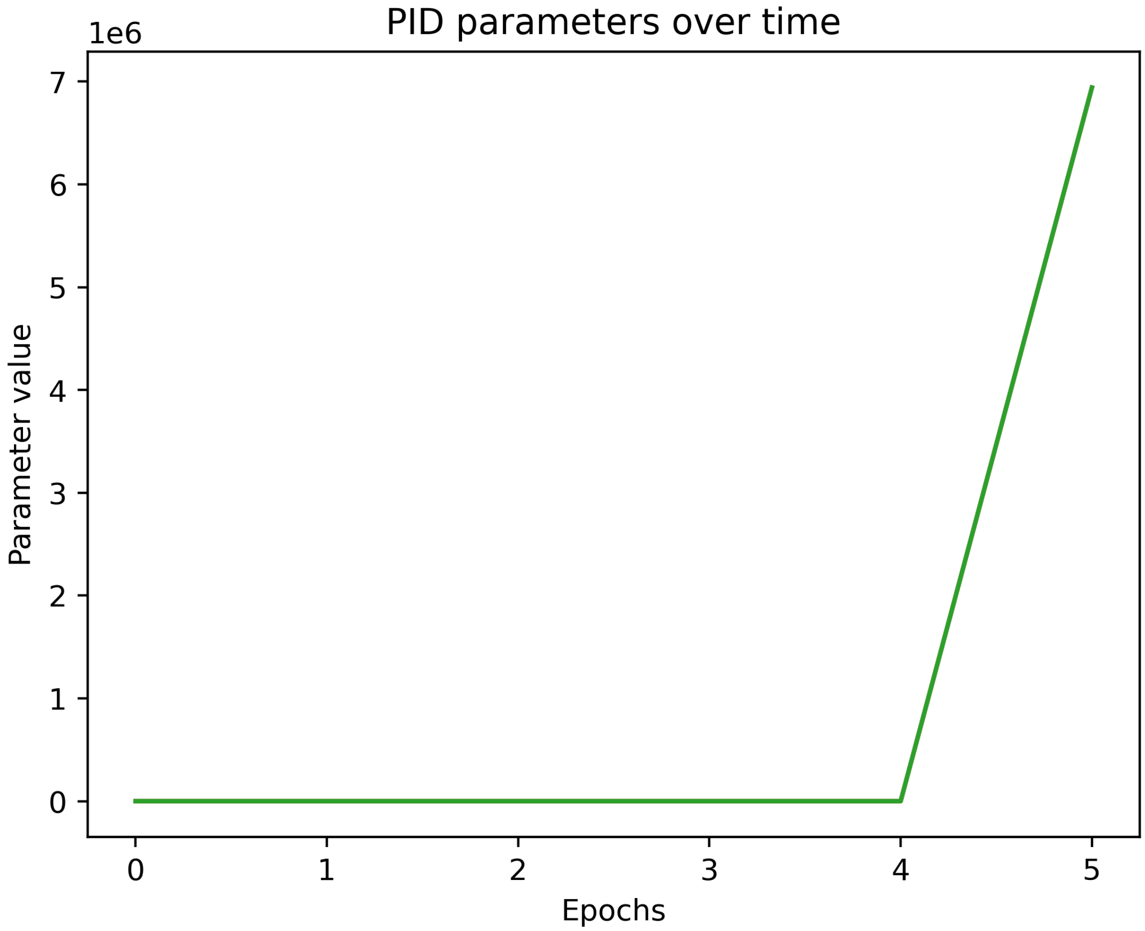


After about 10 epochs, the network converges at a set of parameters which keeps the loss very close to zero.

Cournot

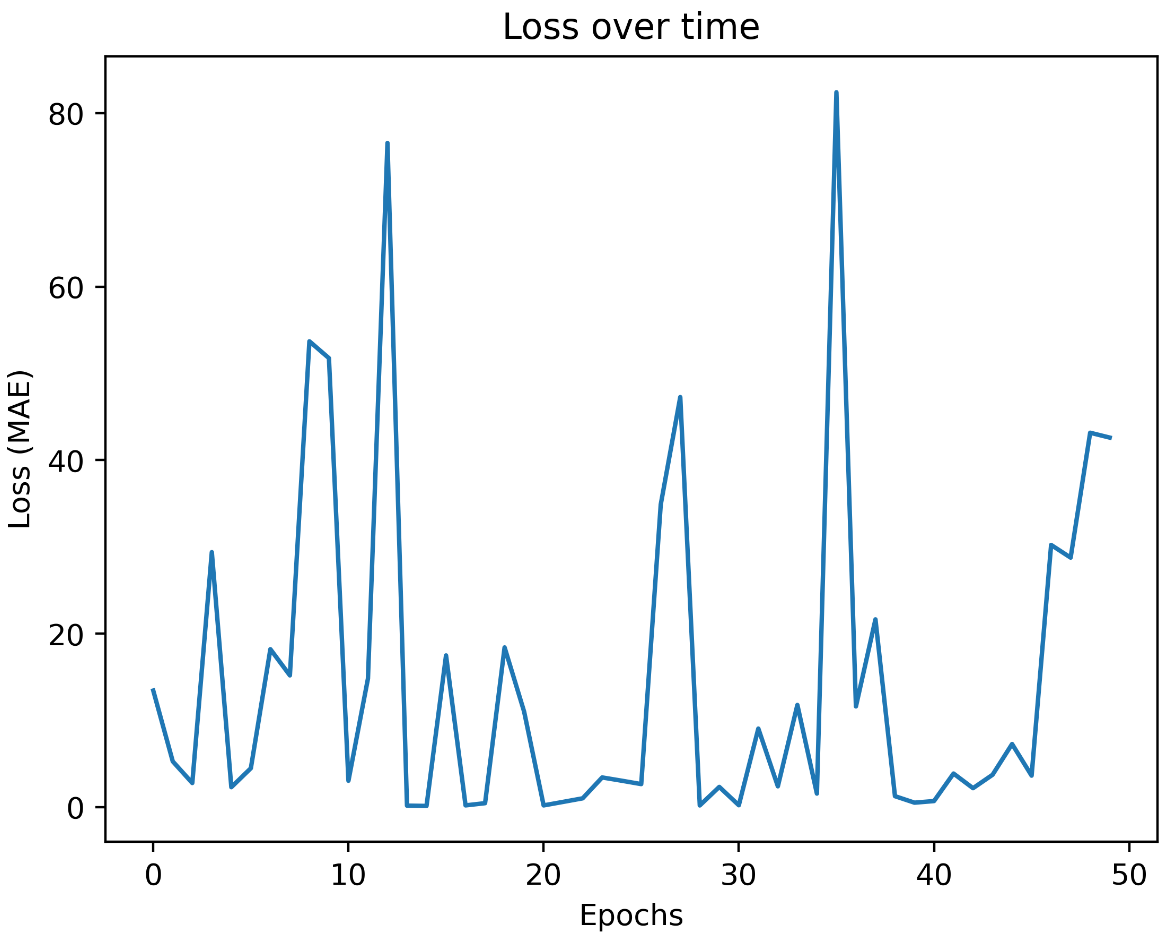
PID


Loss over time
1e20
1.0
0.8
0.6
0.4
0.2
0.0
Loss (MAE)
0
1
2
3
4
5
Epochs



With all combinations of hyperparameters attempted, and using incrementing timesteps, the PID's gradients explode during the Cournot model task. The parameters converge at infinite values.

Neural Network



Throughout the training process, the model is unable to consistently remain near zero loss.

Car

In the customised environment, the agent represents the driver of a car. The model output represents the forward facing force generated by the motor and the driver. The environment represents the car and its interaction with the environment. The car must counteract the ground's friction, as well as a random disturbance at every timestep. At each timestep, newton's second law is applied to determine the car's acceleration.

$$a = \frac{\sum F}{m} = \frac{U - F_{friction} + D}{m}$$

U is the model output.
Friction is given by the following formula.

$$F_{friction} = \mu mg \cdot speed$$

D is iid. from a uniform distribution

$$D \overset{\text{iid.}}{\sim} U(-1, 1)$$

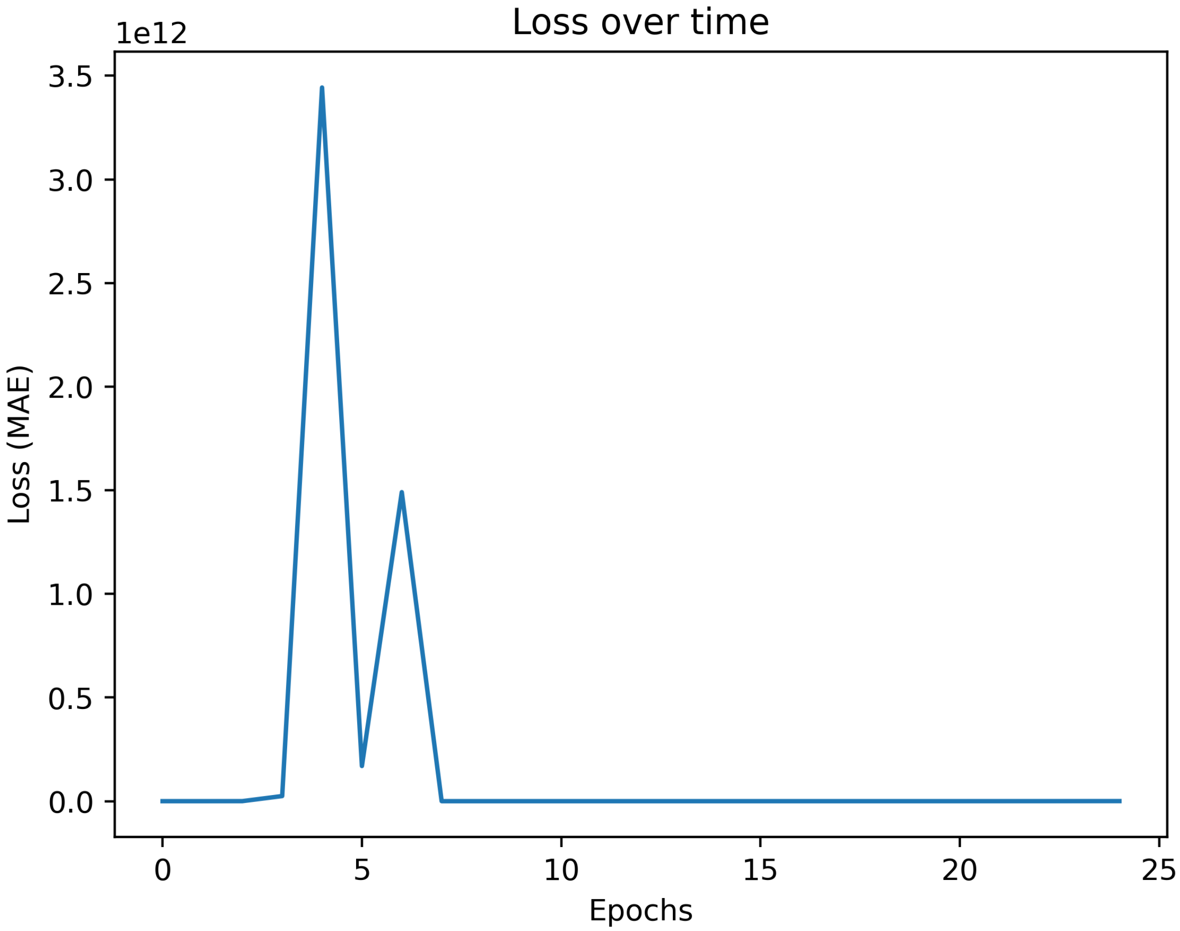
The car's speed is then updated according to the following function, as we make the assumption that one timestep equals one second.

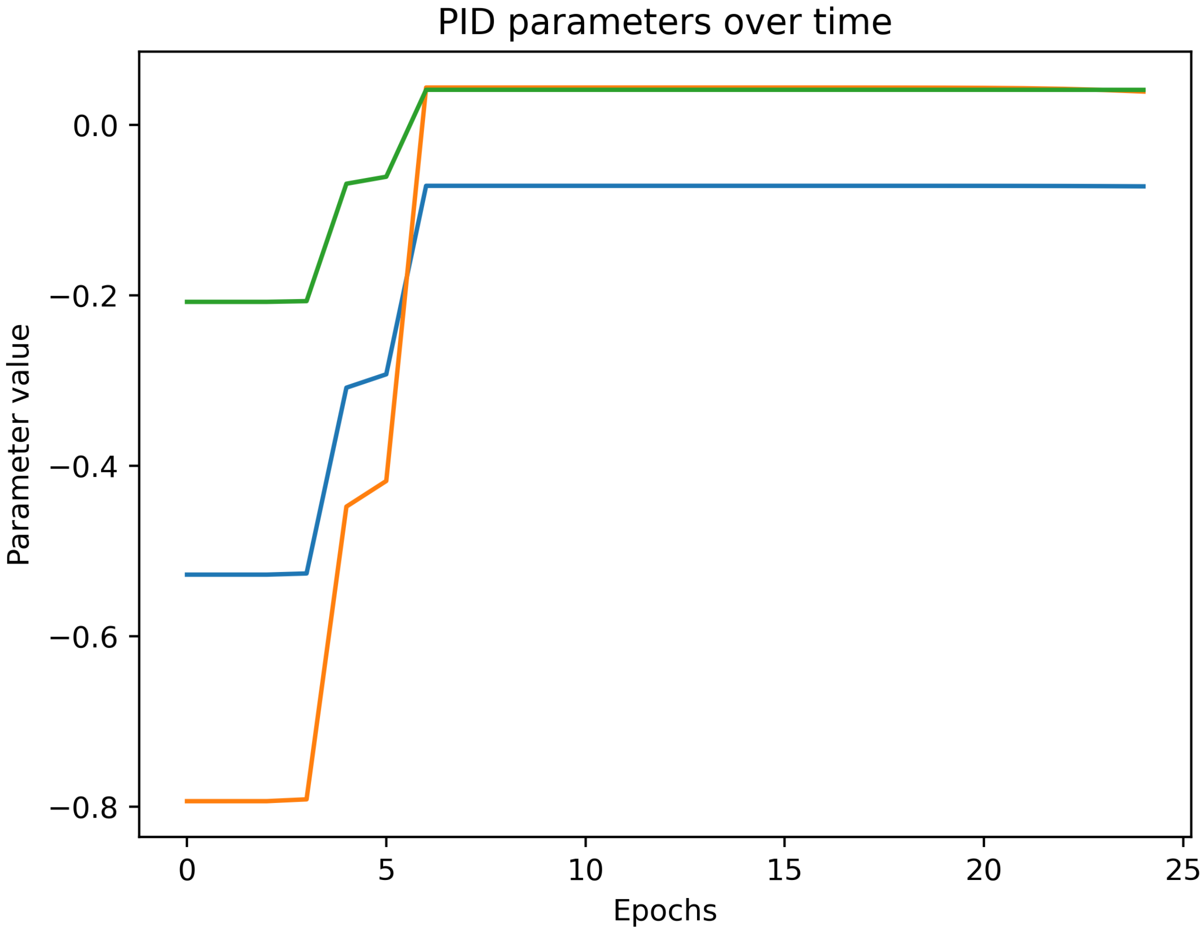
$$Speed_t = Speed_{t-1} + a \cdot 1$$

Loss is calculated according to the difference between the car's current speed and its desired speed.

The desired speed, the car's mass m, and the ground friction coefficient μ are given as hyperparameters.

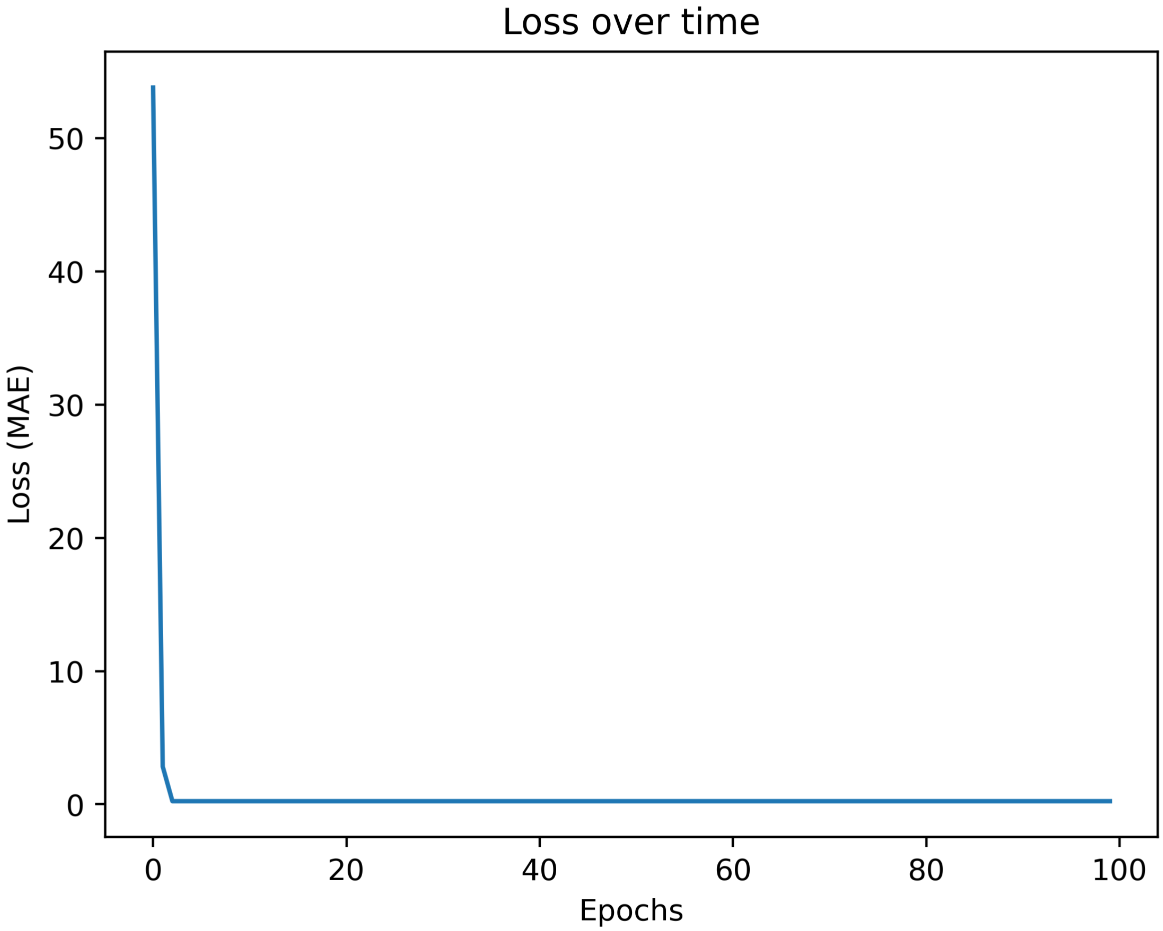
PID


Loss over time
1e12
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Loss (MAE)
0
5
10
15
20
25
Epochs



Though the PID encounters a few epochs with very high loss, it is able to converge at a set of parameters with near-zero errors after about 7 epochs.

Neural Network



After about 5 epochs, the network converges at a set of parameters which keeps the loss very close to zero.