



AWS DeepRacer

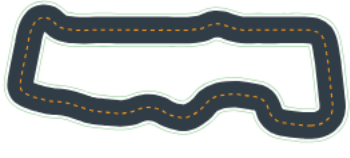
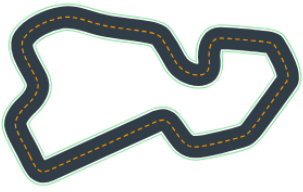



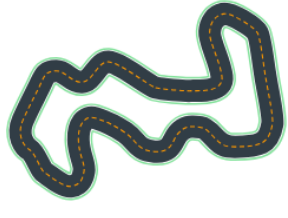
Elena Sanz Espada
IABD 24-25

Índice




Cosas Generales	3
Circuitos	3
Evaluaciones	3
Números en los nombres	4
Notas	4
Sabiduría gptdiana	4
Algoritmos	4
PPO	4
SAC	4
Hiperparámetros	4
Modelos	6
Modelo-PreDef	6
Modelo2	7
Modelo22	8
Modelo23	9
Modelo611	10
Modelo62	11
ModeloRecompensa	12
ModeloRecompensa-velocidad-aumentada-32	14
ModeloRecompensa-33	16
ModeloRecompensa-34	18
ModeloRecompensa-35	20
ModeloRecompensa-36	22
Modelo4	24
Modelo41	26
Modelo42	28

Cosas Generales

Circuitos

Forewer Raceway	Cumulo Turnpike	Jennens Super Speedway -
<ul style="list-style-type: none"> - Clockwise - Counterclockwise 	<ul style="list-style-type: none"> - Clockwise 	<ul style="list-style-type: none"> - Clockwise - Counterclockwise
<ul style="list-style-type: none"> - Ns 	<ul style="list-style-type: none"> - Length: 60 m (197') - Width: 106 cm (42") 	<ul style="list-style-type: none"> - Length: 62.07
		
2022 re:Invent Championship	Po-Chun Super Speedway	Asia Pacific Bay Loop
<ul style="list-style-type: none"> - Clockwise - Counterclockwise 	<ul style="list-style-type: none"> - Clockwise - Counterclockwise 	<ul style="list-style-type: none"> - Counterclockwise
<ul style="list-style-type: none"> - Length:35.87m 	<ul style="list-style-type: none"> - Length:82.24 m 	<ul style="list-style-type: none"> - Length:60 m - Width: 135 cm
		

Evaluaciones

Penalizaciones	salida	2 segundos
	choque	5 segundos
Gráfica	línea verde 	Valor de la recompensa
	línea roja 	Porcentaje recorrido bien del circuito en evaluación
	línea azul 	Porcentaje recorrido bien en entrenamiento

Números en los nombres

Los números en los nombres de los modelos están para reflejar de dónde han sido clonados o qué estoy probando con ellos y porque no soy muy original con los nombres. Lo he empezado a hacer a partir del modelo recompensa y el modelo2. Por ejemplo:

El 1 sería el modelo por defecto, el modelo2 para experimentos con los parámetros y el modelo Recompensa sería el 3 por lo que su clon Recompensa-velocidad-aumentada-32 es el 3 versión 2. El 611 se llama así porque lo hicimos todos en clase y le copié el nombre a Aitzol. Si hago pruebas de coches head-to-head les pondré el 6.

Notas

El entrenamiento se acumula si clonas los modelos.

Según Xabi es mejor entrenar el modelo una vez en un circuito que varias veces menos tiempo en el mismo circuito.

Las variables de las gráficas de SAC aunque parecen caóticas y tienen muchos picos se comportan muy parecido. Cuando hay un pico en la recompensa también los hay el entrenamiento y evaluación aunque sean más bajos o menos bruscos.

Sabiduría gptdiana

Algoritmos

PPO

Más adecuado para problemas simples o cuando necesitas un entrenamiento más rápido. Los hiperparámetros afectan principalmente la estabilidad frente a la velocidad del aprendizaje.

SAC

Más poderoso para entornos complejos con funciones de recompensa sofisticadas. Los hiperparámetros controlan el equilibrio entre exploración, explotación y estabilidad del entrenamiento.

Hiperparámetros

Learning Rate (Tasa de aprendizaje)

- Controla la velocidad con la que el modelo ajusta los pesos de la red durante el entrenamiento.
- Aumentar: El modelo aprende más rápido, pero puede ser inestable o no converger.
- Disminuir: El aprendizaje es más estable, pero lento; riesgo de quedar atrapado en mínimos locales.

Entropy Coefficient (Coeficiente de entropía):

- Controla cuánto explora la política nuevas acciones en lugar de explotar las conocidas.
- Aumentar: Más exploración; útil al inicio del entrenamiento, pero puede generar inestabilidad.
- Disminuir: Menos exploración; fomenta acciones conocidas, útil en etapas finales.

Batch Size (Tamaño del lote):

- Número de muestras usadas para calcular una actualización de gradiente.
- Aumentar: Menos ruido en las actualizaciones; más estabilidad pero mayor consumo de recursos.
- Disminuir: Actualizaciones más rápidas pero más ruidosas; riesgo de inestabilidad.

Epochs (Épocas):

- Número de veces que se procesa el mismo conjunto de datos por actualización.
- Aumentar: Mejora el ajuste, pero puede sobreentrenar los datos en memoria.
- Disminuir: Menos ajuste por iteración, pero más rápido; puede necesitar más episodios.
- Solo de PPO

Clip Range (Rango de recorte):

- Limita cuánto puede cambiar la política entre actualizaciones.
- Aumentar: Más flexibilidad; puede causar inestabilidad.
- Disminuir: Cambios más controlados; puede ralentizar el aprendizaje.
- Solo de PPO

Replay Buffer Size (Tamaño del buffer de repetición):

- Almacena experiencias pasadas para mejorar la estabilidad del entrenamiento.
- Aumentar: Más diversidad de datos, útil para pistas complejas.
- Disminuir: Datos recientes se sobrescriben más rápido, útil para pistas simples.
- Solo de SAC

Tau (Factor de actualización suave):

- Controla la velocidad con la que las redes objetivo (critic targets) se actualizan.
- Aumentar: Actualización más rápida; útil para aprendizaje dinámico, pero menos estabilidad.
- Disminuir: Actualización más lenta; mejora la estabilidad pero ralentiza el aprendizaje.
- Solo de SAC

Train Frequency (Frecuencia de entrenamiento):

- Número de pasos entre actualizaciones del modelo.
- Aumentar: Menos actualizaciones por episodio; puede ahorrar tiempo pero reduce la capacidad de ajuste.
- Disminuir: Más actualizaciones por episodio; mejora el ajuste, pero mayor consumo de recursos.
- Solo de SAC

Modelos

Modelo-PreDef

Modelo predefinido creado sin cambiar nada más que la velocidad a 0.8-2.2

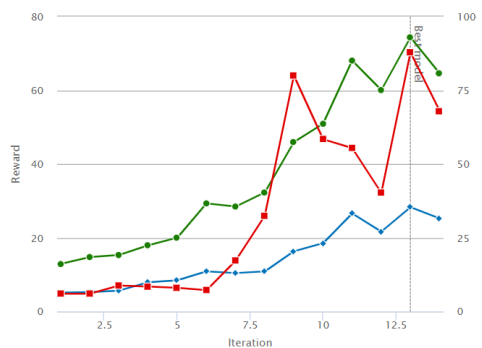
Parámetros

- Velocidad: [0.8,2.2]
- El resto por defecto

Funcion Recompensa

La función de ejemplo para ir por medio

Entrenamiento



- Pista: Forewer Raceway
- Clockwise
- Modo: Contrarreloj
- Tiempo: 60 mins

Evaluaciones

	Tipo prueba	Circuito	Dirección	Salidas	Choques	Tiempo
Eval1	Contrareloj	Forever Raceway	Clockwise	1		00:21,688
				3		00:26,871
				0		00:19,666
Eval2	Contrareloj	Forever Raceway	No Clockwise	4		00:30,067
				5		00:32,670
				5		00:33.075
Eval3	Contrareloj	Cumulo Turnpike	Clockwise	2		00:50,447
				1		00:47,461
				4		00:54,992
Observaciones						

Modelo2

Modelo creado con velocidad 0.7 a 2.4 para intentar ver la diferencia al cambiar el parámetro learning rate

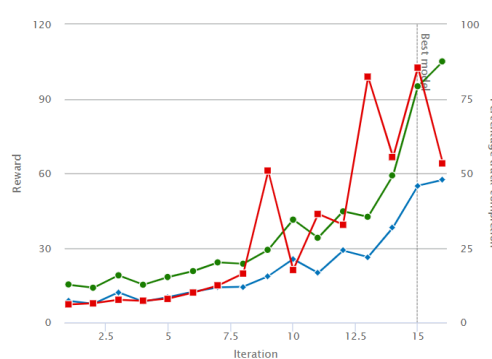
Parámetros

- Velocidad: [0.7 , 2.4]
- Learning rate: 0.0005
- El resto por defecto

Funcion Recompensa

La función de ejemplo para ir por medio

Entrenamiento



- Pista: Forewer Raceway
- Clockwise
- Modo: Contrarreloj
- Tiempo: 60 mins

Evaluación

	Tipo prueba	Circuito	Dirección	salidas	choques	Tiempo
Eval1	Contrareloj	Forever Raceway	Clockwise	1		00:22.011
				3		00:21.002
				0		00:22.397
Eval2	Contrareloj	Forever Raceway	No Clockwise	4		00:31.252
				5		00:30.734
				5		00:26.801
Eval3	Contrareloj	Cumulo Turnpike	Clockwise	2		01:34,320
				1		01:24,341
				4		01:30,748
Observaciones						
<p>Va parecido pero peor que el modelo por defecto.</p> <p>Recibe menos recompensa y el recorrido en evaluación (línea roja) es menos estable.</p> <p>Cuando sale del circuito de entrenamiento es bastante peor</p>						

Modelo22

Prueba del parámetro learning rate parte2. He bajado el learning rate con respecto a Model2 y por defecto a ver si hay diferencia. Le he dado más tiempo para entrenar también. Me he dado cuenta de que le estoy entrenando en un circuito diferente asique la diferencia... a ver qué sale.

Clon de: [Modelo2](#)

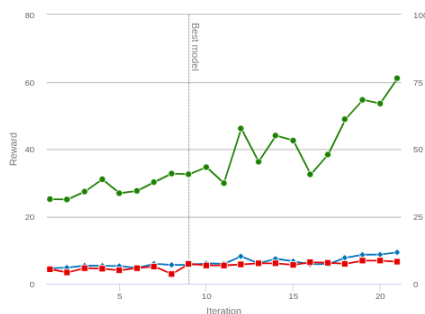
Parámetros

- Velocidad: [0.7 , 2.4]
- Learning rate: 0.00004
- El resto por defecto

Funcion Recompensa

La función de ejemplo para ir por medio

Entrenamiento



- Pista: Cumulo Turnpike
- Clockwise
- Modo: Contrarreloj
- Tiempo: 75 mins

Evaluación

	Tipo prueba	Circuito	Dirección	salidas	choques	tiempo
Eval1	Contrarreloj	Forever Raceway	Clockwise	0		00:20.797
				1		00:22.633
				3		00:27.870
Eval2	Contrarreloj	Cumulo Turnpike	Clockwise	14		01:17.455
				12		01:12.473
				12		01:15.067

Observaciones

En la grafica del entrenamiento da la sensación que no aprende pero los tiempos y salidas de Eval1 no son muy diferentes de Modelo2 puede que sea porque el circuito de evaluación es el mismo que el entrenamiento de Modelo2. En Eval2 el circuito de entrenamiento es el mismo que el de evaluación y por lo mal que ha ido creo que puedo afirmar que no ha aprendido

Modelo611

Modelo creado en clase con Aitzol para probar el head to head

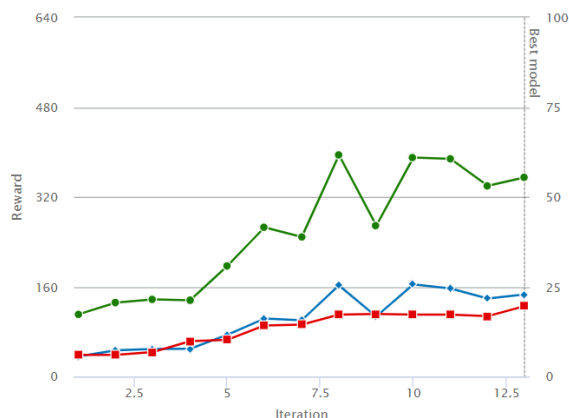
Parámetros

- Velocidad: [0.8,2.2]
- El resto por defecto

Funcion Recompensa

Recompensa de ejemplo head-to-head

Entrenamiento



- Pista: Forewer Raceway
- Clockwise
- Modo: Head-to-head
- los bots cambian de carril v:[0.9]
- Tiempo: 45 mins

Evaluación

	Tipo prueba	Circuito	Dirección	Salidas	Choques	Tiempo
Eval1	Head-to-head	Forever Raceway	Clockwise	1	2	00:34.388
				3	2	00:32.677
				0	3	00:34.908
Eval2	Head-to-head	Asia Pacific Bay Loop	No Clockwise	10	0	01:18.985
				8	1	01:18.972
				9	1	01:20.241
Observaciones						
No está mal pero estaría bien probar en un circuito más ancho. Adelanta y esquiva coches pero la parte de mantenerse en el circuito no lo ha aprendido bien. También va muy lento, habrá que penalizar por lento o recompensar por correr.						

Modelo62

Head-to-head con hiperparametros

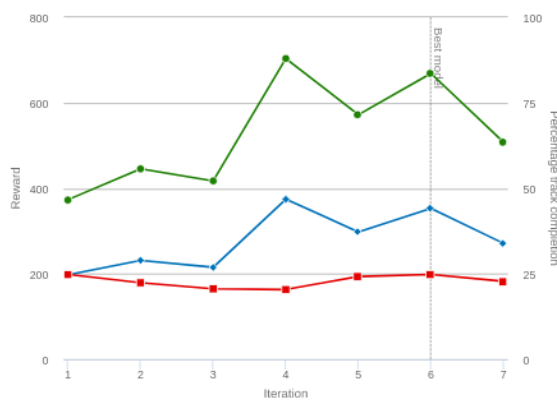
Parámetros

- Velocidad: [0.8,2.2]
- Learning rate: 0.00045
- Entropy: 0.02
- El resto por defecto

Funcion Recompensa

Recompensa de ejemplo head-to-head

Entrenamiento



- Pista: Forewer Raceway
- Clockwise
- Modo: Head-to-head
- Tiempo: 45 mins

Evaluación

	Tipo prueba	Circuito	Dirección	Salidas	Choques	Tiempo
Eval1	Contrareloj	Forever Raceway	Clockwise	0	0	00:18.597
				0	0	00:18.934
				0	0	00:18.936
Eval2	Head-to-head	Forever Raceway	Clockwise	0	2	00:30.112
				1	2	00:32.739
				0	2	00:29.471
Observaciones						
Lo ha hecho muy bien a contrareloj para haberlo entrenado para head-to-head... me confundí 😊 Lo ha hecho bien en el head to head pero no ha conseguido adelantar. Necesito un circuito más ancho						

ModeloRecompensa

Modelo creado con velocidad 0.8-2.2

Parámetros

- Velocidad: [0.8,2.2]
- El resto por defecto

Funcion Recompensa

```
def reward_function(params):  
    reward = 0  
  
    # Read input parameters  
    track_width = params['track_width']  
    distance_from_center = params['distance_from_center']  
    abs_steering = abs(params['steering_angle']) # Only need the absolute steering angle  
    # Calculate 3 markers that are at varying distances away from the center line  
    marker_1 = 0.1 * track_width  
    marker_2 = 0.25 * track_width  
    marker_3 = 0.5 * track_width  
  
    # Give higher reward if the car is closer to center line and vice versa  
    if distance_from_center <= marker_1:  
        reward = 1.0  
    elif distance_from_center <= marker_2:  
        reward = 0.5  
    elif distance_from_center <= marker_3:  
        reward = 0.1  
    else:  
        reward = 1e-3 # likely crashed/ close to off track  
  
    #Zig-zag  
    # Steering penalty threshold, change the number based on your action space setting  
    ABS_STEERING_THRESHOLD = 15  
    # Penalize reward if the car is steering too much  
    if abs_steering > ABS_STEERING_THRESHOLD:  
        reward *= 0.8  
  
    #salida de pista  
    # recompensa por mantenerse dentro del recorrido  
    if params['all_wheels_on_track']:  
        reward += 0.3  
  
    # si se ha salido penalizamos  
    if params['is_offtrack']:  
        reward += -2  
  
    # recompensamos segun progreso del circuito  
    progress = round(params['progress'],2)
```

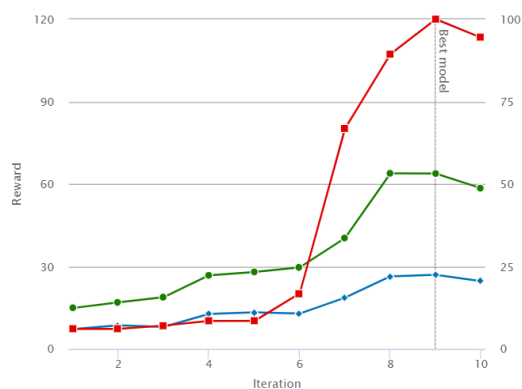
```

if progress == 0.25:
    reward += 1
elif progress == 0.5:
    reward += 2
elif progress == 0.75:
    reward += 3
elif progress == 0.95:
    reward += 4
else:
    reward += 0

return float(reward)

```

Entrenamiento



- Pista: Forever Raceway
- Clockwise
- Modo: Contrarreloj
- Tiempo: 45 mins

Evaluación

	Tipo prueba	Circuito	Dirección	Salidas	Choques	Tiempo
Eval1	Contrarreloj	Forever Raceway	Clockwise	1		00:20.475
				3		00:20.601
				0		00:20.805
Eval2	Contrarreloj	Forever Raceway	No Clockwise	4		00:19.405
				5		00:18.937
				5		00:19.273
Eval3	Contrarreloj	Cumulo Turnpike	Clockwise	2		00:54.943
				1		00:59.399
				4		00:53.315
Observaciones						

Me gusta como ha quedado la gráfica del entrenamiento.
Aunque le va bien en la pista donde ha entrenado cuando sale no tanto.
Quiero probar qué pasa si le aumento la velocidad

ModeloRecompensa-velocidad-aumentada-32

ModeloRecompensa pero con velocidad aumentada de [0.8,2.2] a [0.9-3.0]

Clon de: [ModeloRecompensa](#)

Parámetros

- Velocidad: [0.9 , 3.0]
- El resto por defecto

Funcion Recompensa

```
def reward_function(params):
    reward = 0
    # Read input parameters
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']
    abs_steering = abs(params['steering_angle']) # Only need the absolute steering angle
    # Calculate 3 markers that are at varying distances away from the center line
    marker_1 = 0.1 * track_width
    marker_2 = 0.25 * track_width
    marker_3 = 0.5 * track_width

    # Give higher reward if the car is closer to center line and vice versa
    if distance_from_center <= marker_1:
        reward = 1.0
    elif distance_from_center <= marker_2:
        reward = 0.5
    elif distance_from_center <= marker_3:
        reward = 0.1
    else:
        reward = 1e-3 # likely crashed/ close to off track

    #Zig-zag
    # Steering penalty threshold, change the number based on your action space setting
    ABS_STEERING_THRESHOLD = 15

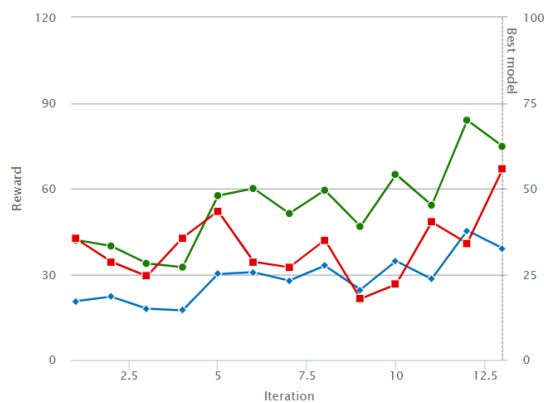
    # Penalize reward if the car is steering too much
    if abs_steering > ABS_STEERING_THRESHOLD:
        reward *= 0.8

    #salida de pista
    if params['all_wheels_on_track']:
        reward += 0.3
    if params['is_offtrack']:
        reward += -2

    # recompensamos segun progreso del circuito
    progress = round(params['progress'],2)
    if progress == 0.25:
        reward += 1
```

```
elif progress == 0.5:
    reward += 2
elif progress == 0.75:
    reward += 3
elif progress == 0.95:
    reward += 4
else:
    reward += 0
return float(reward)
```

Entrenamiento



- Pista: Forewer Raceway
- Clockwise
- Modo: Contrarreloj
- Tiempo: 55 mins

Evaluación

	Tipo prueba	Circuito	Dirección	Salidas	Choques	Tiempo
Eval1	Contrareloj	Forever Raceway	Clockwise	1		00:21.032
				3		00:24.336
				0		00:25.940
Eval2	Contrareloj	Forever Raceway	No Clockwise	5		00:28:224
				7		00:37.881
				9		00:33.471
Eval3	Contrareloj	Cumple Turnpike	Clockwise	2		00:49.571
				4		00:53.465
				2		00:49.591
Observaciones						
<p>No ha mejorado con respecto al anterior.</p> <p>Si descontamos las penalizaciones lo ha hecho más rápido pero se ha salido demasiado y mirando el video zigzaguea y parece fuera de control. Voy a probar a cambiarle la recompensa</p>						

ModeloRecompensa-33

Intento de mejora ModeloRecompensa32. La velocidad mínima vuelve a 0.8. Voy a entrenarlo en un circuito diferente con rectas más largas y curvas más cerradas para que tenga más variedad. Voy a cambiar la recompensa para evitar más el zigzag.

Clon de: [ModeloRecompensa-velocidad-aumentada-32](#)

Parámetros

- Velocidad: [0.8 , 3.0]
- El resto por defecto

Funcion Recompensa

```
def reward_function(params):
    reward = 0
    # Read input parameters
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']
    abs_steering = abs(params['steering_angle']) # Only need the absolute steering angle
    # Calculate 3 markers that are at varying distances away from the center line
    marker_1 = 0.1 * track_width
    marker_2 = 0.25 * track_width
    marker_3 = 0.5 * track_width

    # Give higher reward if the car is closer to center line and vice versa
    if distance_from_center <= marker_1:
        reward = 1.1
    elif distance_from_center <= marker_2:
        reward = 0.4
    elif distance_from_center <= marker_3:
        reward = 0.1
    else:
        reward = 1e-5 # likely crashed/ close to off track

    #Zig-zag
    # Steering penalty threshold, change the number based on your action space setting
    ABS_STEERING_THRESHOLD = 15

    # Penalize reward if the car is steering too much
    if abs_steering > ABS_STEERING_THRESHOLD:
        reward *= 0.7

    #salida de pista
    if params['all_wheels_on_track']:
        reward += 0.3

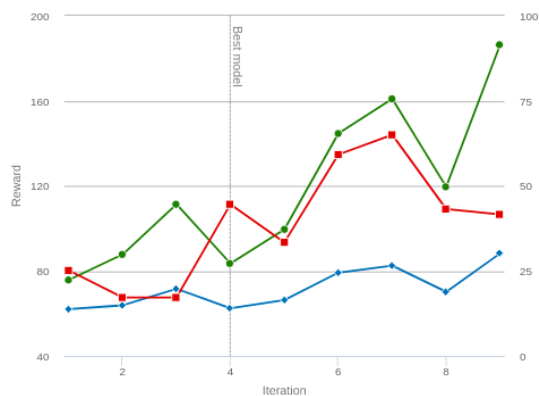
    if params['is_offtrack']:
        reward += -2

    # recompensamos segun progreso del circuito
```



```
progress = round(params['progress'],2)
if progress == 0.25:
    reward += 1
elif progress == 0.5:
    reward += 2
elif progress == 0.75:
    reward += 3
elif progress == 0.95:
    reward += 4
else:
    reward += 0
return float(reward)
```

Entrenamiento



- Pista: Jennens Super Speedway
- Clockwise
- Modo: Contrarreloj
- Tiempo: 75 mins

Evaluación

	Tipo prueba	Circuito	Dirección	Salidas	Choques	Tiempo
Eval1	Contrareloj	Jennens Super Speedway	Clockwise	1		00:48.811
				1		00:46.679
				2		00:49.666
Eval2	Contrareloj	Jennens Super Speedway	No Clockwise	5		00:56.612
				5		00:55.403
				5		00:56.128
Observaciones						
Aunque la velocidad está limitada a 3 rara vez pasa de 2. Tendré que cambiar la recompensa para que vaya más rápido. El modelo ha empeorado, se sale más y aun descontando las penalizaciones va más lento. Queda descartado, ha sido un paso atrás						

ModeloRecompensa-34

Intento de mejora ModeloRecompensa32 y 33. La velocidad mínima vuelve a 0.8. Voy a entrenarlo en un circuito diferente con rectas más largas y curvas más cerradas para que tenga más variedad. Voy a cambiar la recompensa para que vaya más rapido y con menos zig-zag

Clon de: [ModeloRecompensa-velocidad-aumentada-32](#)

Parámetros

- Velocidad: [0.8 , 3.0]
- El resto por defecto

Funcion Recompensa

```
##### recompensa-34 #####
def reward_function(params):
    reward = 0
    # Read input parameters
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']
    abs_steering = abs(params['steering_angle']) # Only need the absolute steering
angle
    speed = params['speed']
    # Calculate 3 markers that are at varying distances away from the center line
    marker_1 = 0.1 * track_width
    marker_2 = 0.25 * track_width
    marker_3 = 0.5 * track_width

    # Give higher reward if the car is closer to center line and vice versa
    if distance_from_center <= marker_1:
        reward = 1.0
    elif distance_from_center <= marker_2:
        reward = 0.5
    elif distance_from_center <= marker_3:
        reward = 0.1
    else:
        reward = 1e-3 # likely crashed/ close to off track

    #Zig-zag
    # Steering penalty threshold, change the number based on your action space
setting
    ABS_STEERING_THRESHOLD = 15

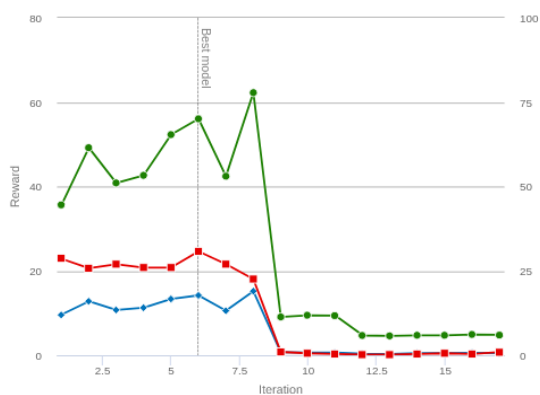
    # Penalize reward if the car is steering too much
    if abs_steering > ABS_STEERING_THRESHOLD:
        reward *= 0.6
    else:
        if speed < 1.8:
            reward -= (speed - 1.8) * 0.2
        if speed > 1.5: # premia por velocidad
            reward += (speed - 1.5) * 0.3 # Increase reward based on speed
```

```
#salida de pista
# recompensa por mantenerse dentro del recorrido
if params['all_wheels_on_track']:
    reward += 0.2

# si se ha salido penalizamos
if params['is_offtrack']:
    reward += -2

return float(reward)
```

Entrenamiento



- Pista: 2022 re:Invent Championship
- Clockwise
- Modo: Contrarreloj
- Tiempo: 55 mins

Entrenamiento interrumpido. Llegado un momento solo gira a la izquierda, siempre el mismo giro y no avanza.

Hay que revisar donde interviene el ángulo de giro y la proporción de recompensas

Modelo **descartado**

ModeloRecompensa-35

ModeloRecompensa34 pero con la recompensa mejor proporcionada.

Clon de: Ninguno porque laboratorio nuevo 😊

Parámetros

- Velocidad: [0.8 , 3.0]
- El resto por defecto

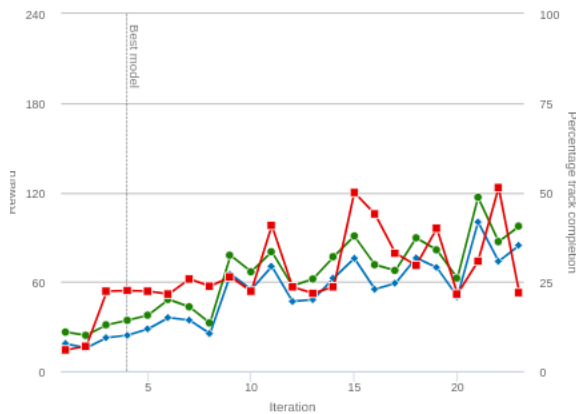
Funcion Recompensa

```
def reward_function(params):  
    reward = 0  
  
    # Read input parameters  
    track_width = params['track_width']  
    distance_from_center = params['distance_from_center']  
    abs_steering = abs(params['steering_angle']) # Absolute steering angle  
    speed = params['speed']  
  
    ### Centrado  
    # Calculate markers at varying distances from the center line  
    marker_1 = 0.1 * track_width  
    marker_2 = 0.25 * track_width  
    marker_3 = 0.5 * track_width  
  
    # Reward based on distance from center  
    if distance_from_center <= marker_1:  
        reward = 1.5  
    elif distance_from_center <= marker_2:  
        reward = 0.8  
    elif distance_from_center <= marker_3:  
        reward = 0.2  
    else:  
        reward = 1e-3 # Likely off track  
  
    ### Zigzag (steering stability)  
    ABS_STEERING_THRESHOLD = 15 # Threshold for penalizing steering  
    if abs_steering > ABS_STEERING_THRESHOLD:  
        reward *= 0.5 # Penaliza por girar muy cerrado  
  
    ### Velocidad  
    # Encourage higher speeds  
    if speed >= 2.0:  
        reward += (speed - 2.0) * 1.0 # Incentivo para que corra  
    elif speed < 1.4:  
        reward -= (1.4 - speed) * 0.5 # Penalizacion por lento  
  
    ### Pista (track adherence)  
    # Premia por estar dentro  
    if params['all_wheels_on_track']:  
        reward += 0.4  
    if params['is_offtrack']:
```

```
reward -= 3.0 # Penaliza por salir

return float(reward)
```

Entrenamiento



- Pista: Forewer Raceway
- Clockwise
- Modo: Contrarreloj
- Tiempo: 75 mins

Evaluación

	Tipo prueba	Circuito	Dirección	Salidas	Choques	Tiempo
Eval1	Contrarreloj	Forewer Raceway	Clockwise	4		00:22.754
				3		00:20.597
				3		00:19.866
Eval2	Contrarreloj	Forewer Raceway	No Clockwise	11		00:40.948
				5		00:26.072
				10		00:39.140
Eval3	Contrarreloj	Asia Pacific Bay Loop	No Clockwise	9		00:51.452
				10		00:57.922
				11		00:57.032

Observaciones

Viendo solo el entrenamiento pensaba que necesitaba más tiempo para entrenar. Las evaluaciones dejan que desear. Los tiempos de la primera están bien para lo que se ha salido pero en las otras 2 ha sido un poco desastre, se ha salido demasiado. Pensaré si lo re-entreno en un circuito más variado y si le ajusto la recompensa referente a la velocidad. No se como pero de vez en cuando derrapa un poco

ModeloRecompensa-36

Modelo creado igual que Recompensa-35 pero con hiperparametros cambiados para intentar mejorar.

Parámetros

- Velocidad: [0.8 , 3.0]
- Gradient batch size: 128
- Learning rate: 0.004
- Entropy: 0.3
- Number experience episodes: 15
- Number epochs: 10 (por defecto)

Funcion Recompensa

```
def reward_function(params):  
    reward = 0  
    # Read input parameters  
    track_width = params['track_width']  
    distance_from_center = params['distance_from_center']  
    abs_steering = abs(params['steering_angle']) # Absolute steering angle  
    speed = params['speed']  
  
    # Calculate markers at varying distances from the center line  
    marker_1 = 0.1 * track_width  
    marker_2 = 0.25 * track_width  
    marker_3 = 0.5 * track_width  
  
    # Reward based on distance from center  
    if distance_from_center <= marker_1:  
        reward = 1.5  
    elif distance_from_center <= marker_2:  
        reward = 0.8  
    elif distance_from_center <= marker_3:  
        reward = 0.2  
    else:  
        reward = 1e-3 # Likely off track  
  
    ### Zigzag (steering stability)  
    ABS_STEERING_THRESHOLD = 15 # Threshold for penalizing steering  
    if abs_steering > ABS_STEERING_THRESHOLD:  
        reward *= 0.5 # Penaliza por girar muy cerrado  
  
    ### Velocidad  
    # Encourage higher speeds  
    if speed >= 2.0:  
        reward += (speed - 2.0) * 1.0 # Incentivo para que corra  
    elif speed < 1.4:  
        reward -= (1.4 - speed) * 0.5 # Penalizacion por lento  
  
    # Premia por estar dentro
```

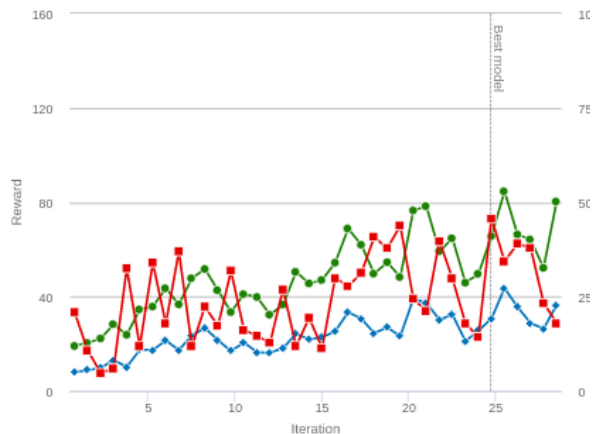
```

if params['all_wheels_on_track']:
    reward += 0.4
if params['is_offtrack']:
    reward -= 3.0 # Penaliza por salir

return float(reward)

```

Entrenamiento



- Pista: Forewer raceway
- Clockwise
- Modo: Contrarreloj
- Tiempo: 75 mins

Evaluación

	Tipo prueba	Circuito	Dirección	Salidas	Choques	Tiempo
Eval1	Contrarreloj	Forewer Raceway	Clockwise	2		00:20.058
				4		00:26.206
				3		00:29.003
Eval2	Contrarreloj	Forewer Raceway	No Clockwise	8		00:36.674
				7		00:32.608
				9		00:38.334
Eval3	Contrarreloj	Asia Pacific Bay Loop	No Clockwise	14		01:14.417
				12		01:20.217
				12		01:15.562

Observaciones

Con respecto a recompensa-35, se nota la entropía durante el entrenamiento por los picos de la gráfica. También me ha parecido que necesitaba más tiempo para entrenar y viendo cuánto se ha salido en las evaluaciones creo que lo entrenaré en otro circuito con más variedad en curvas porque si derrapa, unas veces sale bien y otras sale de más. Subir gradient batch size a 256 y bajar la entropía un poquito sería buena idea para ver si se estabiliza.

Modelo4

Intento de SAC. Voy a poner la recompensa para evitar más el zigzag, que vaya demasiado lento y premie por ir centrado. Voy hablar con GPT para tener una idea de qué poner en los hiperparámetros y por qué

Parámetros

- Velocidad: [1.0 , 3.0]
- Angulo giro: [22,-22] reduzco el ángulo para que no haga demasiadas pruebas
- Learning rate: 0.0004
- Gradient batch size: 128. 256 le daría más estabilidad pero por el tiempo 128 es mejor.
- Discount factor: 0.99 (por defecto)

Funcion Recompensa

```
def reward_function(params):  
    # Leer parámetros de entrada  
    track_width = params['track_width']  
    distance_from_center = params['distance_from_center']  
    steering_angle = abs(params['steering_angle'])  
    speed = params['speed']  
    all_wheels_on_track = params['all_wheels_on_track']  
  
    # Inicializar la recompensa  
    reward = 1e-3 # Evitar cero para no interrumpir el entrenamiento  
  
    ### Evitar zigzag ###  
    # Penalización para grandes ángulos de giro  
    ABS_STEERING_THRESHOLD = 15 # Ángulo límite para penalizar  
    if steering_angle > ABS_STEERING_THRESHOLD:  
        reward *= 0.7 # Penaliza si el giro es demasiado cerrado  
  
    ### Premiar ir centrado ###  
    # Definir marcadores basados en la anchura de la pista  
    marker_1 = 0.1 * track_width # Cercano al centro  
    marker_2 = 0.25 * track_width # Moderadamente cerca  
    marker_3 = 0.5 * track_width # Lejos del centro  
  
    # Recompensa basada en la distancia desde el centro  
    if distance_from_center <= marker_1:  
        reward += 1.5 # Máxima recompensa por estar cerca del centro  
    elif distance_from_center <= marker_2:  
        reward += 0.8 # Recompensa moderada  
    elif distance_from_center <= marker_3:  
        reward += 0.3 # Recompensa baja  
    else:  
        reward = 1e-3 # Penalización fuerte por estar fuera de pista  
  
    # Recompensa basada en la velocidad  
    MIN_SPEED = 1.5 # Velocidad mínima deseada
```



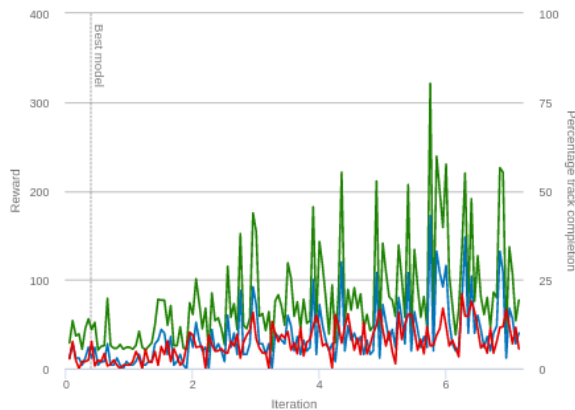
```

TARGET_SPEED = 2.5 # Velocidad ideal
if speed < MIN_SPEED:
    reward *= 0.5 # Penaliza velocidades demasiado lentas
elif speed >= MIN_SPEED and speed <= TARGET_SPEED:
    reward += (speed - MIN_SPEED) * 0.5
elif speed > TARGET_SPEED:
    reward += 0.3 # Incentiva velocidades mas altas
# Recompensa por mantener estabilidad
if all_wheels_on_track:
    reward += 0.5

return float(reward)

```

Entrenamiento



- Pista:Asia Pacific Bay Loop
- Clockwise
- Modo: Contrarreloj
- Tiempo: 3h 20mins (200mins)

Evaluación

	Tipo prueba	Circuito	Dirección	Salidas	Choques	Tiempo
Eval1	Contrarreloj	Asia Pacific Bay Loop	No Clockwise	8		00:45.804
				9		00:48.212
				7		00:42.847
Eval2	Contrarreloj	Forewer Raceway	Clockwise	36		01:39.769
				40		01:52.598
				41		01:55.007

Observaciones

No da bien las curvas. Le ha faltado tiempo de entrenamiento. Voy a volver a entrenarlo pero voy a subirle el gradient batch a 256 ahora que no empieza de cero. Me daba curiosidad que tal lo haría en un circuito con menor curvas y no sabe girar a la derecha, casi todas las curvas son a la izquierda en el circuito de entrenamiento asique tendre que volver a entrenarlo en uno contrario

Modelo41

Mejora Modelo4 de SAC. El modelo4 se quedó corto en tiempo de entrenamiento así que este va a entrenar 4 h y media, con 256 de gradient y subo 2 grados. También voy a añadir una penalización por salirse y modificar un poco la parte relacionada con la velocidad.

Clon de: [Modelo4](#)

Parámetros

- Velocidad: [1.0 , 3.0]
- Angulo giro: [24,-24]
- Learning rate: 0.0004
- Gradient batch size: 256
- Discount factor: 0.99 (por defecto)

Funcion Recompensa

```
def reward_function(params):  
    # Leer parámetros de entrada  
    track_width = params['track_width']  
    distance_from_center = params['distance_from_center']  
    steering_angle = abs(params['steering_angle'])  
    speed = params['speed']  
    all_wheels_on_track = params['all_wheels_on_track']  
  
    reward = 1e-3 # Evitar cero para no interrumpir el entrenamiento  
    # Penalización para grandes ángulos de giro  
    ABS_STEERING_THRESHOLD = 15 # Ángulo límite para penalizar  
    if steering_angle > ABS_STEERING_THRESHOLD:  
        reward *= 0.65 # Penaliza si el ángulo de giro es demasiado alto  
  
    # Definir marcadores basados en la anchura de la pista  
    marker_1 = 0.1 * track_width # Cercano al centro  
    marker_2 = 0.25 * track_width # Moderadamente cerca  
    marker_3 = 0.5 * track_width # Lejos del centro  
  
    # Recompensa basada en la distancia desde el centro  
    if distance_from_center <= marker_1:  
        reward += 1.5 # Máxima recompensa por estar cerca del centro  
    elif distance_from_center <= marker_2:  
        reward += 0.8 # Recompensa moderada  
    elif distance_from_center <= marker_3:  
        reward += 0.3 # Recompensa baja  
    else:  
        reward = 1e-3 # Penalización fuerte por estar fuera de pista  
  
    # Recompensa basada en la velocidad  
    MIN_SPEED = 1.2 # Velocidad mínima deseada  
    TARGET_SPEED = 2.3 # Velocidad ideal  
    if speed < MIN_SPEED:  
        reward *= 0.5 # Penaliza velocidades demasiado lentas  
    elif speed >= MIN_SPEED and speed <= TARGET_SPEED:
```

```
reward += (speed - MIN_SPEED) * 0.5 # Recompensa por velocidades dentro
del rango deseado
elif speed > TARGET_SPEED:
    reward += 0.3 # Incentiva velocidades superiores, pero no demasiado

### Bonificación por mantener todas las ruedas en la pista ###
if all_wheels_on_track:
    reward += 0.5 # Recompensa adicional por mantener estabilidad
# si se ha salido penalizamos
if params['is_offtrack']:
    reward += -2

return float(reward)
```

Modelo42

Mejora Modelo4 de SAC. El modelo4 se quedó corto en tiempo de entrenamiento así que este va a entrenar otras 3 h y media pero subo 2 grados el giro. También dejo a 128 el gradient para ver un poco la diferencia con el Modelo41 También voy a añadir una penalización por salirse y modificar un poco la parte relacionada con la velocidad.

Clon de: [Modelo4](#)

Parámetros

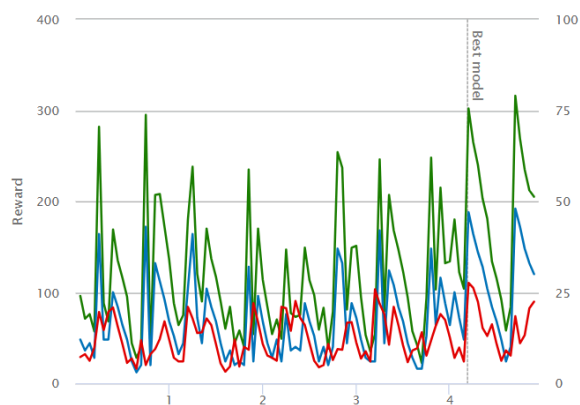
- Velocidad: [1.0 , 3.0]
- Angulo giro: [24,-24]
- Learning rate: 0.00032
- Gradient batch size: 128
- Discount factor: 0.99 (por defecto)

Funcion Recompensa

```
def reward_function(params):  
    # Leer parámetros de entrada  
    track_width = params['track_width']  
    distance_from_center = params['distance_from_center']  
    steering_angle = abs(params['steering_angle'])  
    speed = params['speed']  
    all_wheels_on_track = params['all_wheels_on_track']  
  
    # Inicializar la recompensa  
    reward = 1e-3 # Evitar cero para no interrumpir el entrenamiento  
  
    ### Evitar zigzag ###  
    # Penalización para grandes ángulos de giro  
    ABS_STEERING_THRESHOLD = 15 # Ángulo límite para penalizar  
    if steering_angle > ABS_STEERING_THRESHOLD:  
        reward *= 0.7 # Penaliza si el giro es demasiado cerrado  
  
    ### Premiar ir centrado ###  
    # Definir marcadores basados en la anchura de la pista  
    marker_1 = 0.1 * track_width # Cercano al centro  
    marker_2 = 0.25 * track_width # Moderadamente cerca  
    marker_3 = 0.5 * track_width # Lejos del centro  
  
    # Recompensa basada en la distancia desde el centro  
    if distance_from_center <= marker_1:  
        reward += 1.5 # Máxima recompensa por estar cerca del centro  
    elif distance_from_center <= marker_2:  
        reward += 0.8 # Recompensa moderada  
    elif distance_from_center <= marker_3:  
        reward += 0.3 # Recompensa baja  
    else:  
        reward = 1e-3 # Penalización fuerte por estar fuera de pista  
  
    # Recompensa basada en la velocidad
```

```
MIN_SPEED = 1.5 # Velocidad mínima deseada
TARGET_SPEED = 2.5 # Velocidad ideal
if speed < MIN_SPEED:
    reward *= 0.5 # Penaliza velocidades demasiado lentas
elif speed >= MIN_SPEED and speed <= TARGET_SPEED:
    reward += (speed - MIN_SPEED) * 0.5
elif speed > TARGET_SPEED:
    reward += 0.3 # Incentiva velocidades mas altas
# Recompensa por mantener estabilidad
if all_wheels_on_track:
    reward += 0.5
return float(reward)
```

Entrenamiento



- Pista:Asia Pacific Bay Loop
- Clockwise
- Modo: Contrarreloj
- Tiempo: 3h 20mins (200mins)

Evaluación

	Tipo prueba	Circuito	Dirección	Salidas	Choque	Tiempo
Eval1	Contrareloj	Asia Pacific Bay Loop	No Clockwise	7		00:41.922
				7		00:43.131
				6		00:39.455
Eval2	Contrareloj	Forewer Raceway	Clockwise	33		01:35.198
				21		01:04.662
				37		01:45.329
Observaciones						
Lo ha hecho mejor que el modelo4 pero aun asi cuando lo pruebo en otro circuito diferente es desastroso, parece que toma decisiones semi-aleatoriamente						