

Tutorial Deepracer-for-cloud

1 - Conexão com servidor

Obs.: Se a solução está sendo executada localmente, essa seção pode ser desconsiderada.

Obs2.: O Inteli possui um servidor chamado Shrek2 e ele será o alvo da nossa conexão. Sendo assim, as instruções indicadas funcionaram apenas nesse host.

Para iniciar a conexão com o servidor, existe a necessidade de que o sistema local possui o OpenSSH instalado. Devido o fato de que esse software é bastante comum nas distribuições dos sistemas operacionais mais famosos, o seu guia de instalação não será contemplado nesse documento. Para mais informações, acesse: <https://www.openssh.com/>

Com o OpenSSH instalado, a conexão será iniciada com o seguinte comando: `ssh deepracer@shrek2.inteli.local`. Esse comando deve ser executado em um emulador de terminal e a máquina local precisará estar dentro da rede interna da faculdade. Logo após isso, o programa irá solicitar senha do usuário "deepracer". Por fins de segurança, essa senha também não será fornecida nesse documento.

Por fim, a interface do terminal do servidor estará visível no emulador de terminal e ela será utilizada para rodar os comandos nas seções seguintes.

2 - Iniciando o Deepracer-for-cloud usando a pista padrão

Para iniciar o deepracer-for-cloud na pista padrão, é necessário ter como parâmetro um número de um grupo. Cada grupo possui a sua própria série de arquivos para realizar o seu treinamento. Como exemplo, os arquivos do grupo 3 serão utilizados.

Como cada grupo possui um nome de arquivo diferente e o deepracer-for-cloud possui

uma nomenclatura padrão, a criação de soft links será necessária para dar prosseguimento com o treino. O comando para fazer esses links é: `cd ~/deepracer-for-cloud && ln -s run3.env run.env && ln -s system3.env system.env && ln -s custom_files3 custom_files`. O “3” simboliza o número do grupo que está sendo utilizado.

Com os links devidamente criados, o próximo passo é executar o script “[activate.sh](#)” no shell que está sendo utilizado. Para isso, deve-se rodar o comando `cd ~/deepracer-for-cloud && source bin/activate.sh`, isso fará com que os próximos comandos de atalho sejam reconhecidos pelo shell.

Com o shell apropriadamente ajustado. As configurações do ambiente devem ser atualizadas para que o treinamento ocorra da maneira adequada. Para isso, o comando `cd ~/deepracer-for-cloud && dr-update && dr-update-env && dr-upload-custom-files` deve ser executado.

Depois da atualização dos arquivos, a única instrução restante é o `dr-start-training -w`, que iniciará um treino do zero na sessão de terminal que está sendo utilizada.

Para finalizar um treino, basta executar o comando `dr-stop-training`

Obs3.: Rodar o comando de inicio de treino com a flag ‘-w’ irá reiniciar todo o aprendizado do carrinho. Antes de executá-lo, é recomendado que todos os arquivos de resultado (cobertos em uma seção posterior do documento) sejam devidamente armazenados em outro local.

Obs4.: Para visualizar o treinamento em uma interface gráfica, uma ponte deve ser feita durante a conexão SSH. Para isso, é necessário voltar para uma sessão de terminal na máquina host e conectar com o servidor com o seguinte comando: `ssh -L <PORTA_LOCAL>:localhost:<PORTA_REMOTA> deepracer@shrek2.inteli.local`, substituindo a porta da interface gráfica do grupo. Quando o treino está em andamento, é possível rodar o comando `dr-start-viewer`, que mostrará as instruções na tela para abrir a visualização. Para finalizar a visualização, é necessário usar o comando `dr-stop-viewer`.

Obs5.: Para conseguir dar prosseguimento com o mesmo treinamento após a troca de parâmetros, é possível resumir o treino após um update usando `dr-increment-`

```
training && dr-start-training .
```

3 - Alteração da pista

Para alterar a pista do simulador para o ambiente do Intel, foi primeiramente necessário fazer a modificação da imagem do deepracer-simapp. Para fins de objetividade, esse procedimento não será descrito nesse documento. Após todo esse processo, foi modificada a variável `DR_SIMAPP_SOURCE` no arquivo "system3.env" com a imagem docker customizada e a variável `DR_WORLD_NAME` no arquivo "run3.env" para o nome da nova pista. Depois dessas alterações, o comando de update `cd ~/deepracer-for-cloud && dr-update && dr-update-env && dr-upload-custom-files` foi executado para alterar o ambiente de execução do simulador.

4 - Alteração de hiper parâmetros e função de recompensa

Para fazer ajustes no hiper parâmetros do treino ou na função de recompensa, dois arquivos que devem ser alterados, respectivamente, são o "hyperparameter.json" e o "reward_function.py". O algoritmo de treino também pode ser alterado no arquivo "model_metadata.json".

5 - Avaliação do treinamento

Após parar o treinamento, é possível compilar todos os resultados e o agente resultante usando o comando `dr-start-evaluation`. Todos os arquivos resultantes poderão ser encontrados no bucket minio do setup.

6 - Demonstração

Caso tenham ocorrido questionamentos durante a leitura do documento. Existe um vídeo demonstrativo no repositório

```
SM_HP_NUM_EPOCHS=10
SM_HP_LEARNING_RATE=0.0005
SM_HP_DISCOUNT_FACTOR=0.95
SM_HP_EXPLORATION_STRATEGY=epsilon-greedy
SM_HP_EPSILON_START=1.0
SM_HP_EPSILON_END=0.2
SM_HP_EPSILON_DECAY=0.99
SM_HP_NUM_HIDDEN_LAYERS=3
SM_HP_HIDDEN_UNITS_PER_LAYER=128
SM_HP_LOSS_FUNCTION=mean_squared_error
PYTHONPATH=/opt/ml/code:/usr/local/bin:/usr/lib/python3.8:/usr/lib/python3.8/lib-dynload:/usr/local/li
h/python3.8/dist-packages:/usr/lib/python3/dist-packages

Invoking script with the following command:

/bin/sh -c './start.sh --aws_region us-east-1 --batch_size 64 --discount_factor 0.95 --epsilon_decay 0.99 --epsilon_end 0.2
--epsilon_start 1.0 --exploration_strategy epsilon-greedy --hidden_units_per_layer 128 --learning_rate 0.0005 --loss_functio
n mean_squared_error --model_metadata_s3_key s3://bucket/g3_custom_files/model_metadata.json --num_epochs 10 --num_hidden_la
yers 3 --s3_bucket bucket --s3_prefix g3_rl-deeptracer-sagemaker'

2024-12-09 06:37:55.911 sagemaker-training-toolkit INFO      Exceptions not imported for SageMaker Debugger as it is not inst
alled.
2024-12-09 06:37:56.284770: I tensorflow/core/util/port.cc:118] oneDNN custom operations are on. You may see slightly differ
ent numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the en
vironment variable 'TF_INPLACE_ONEDNN_OPTS=0'.
2024-12-09 06:37:56.333044: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use
available CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 AVX512F AVX512_VNNI AVX512_BF16 AVX_VNNI AMX_TILE AMX_INT8 AMX_BF16 FMA, in other
operations, rebuild TensorFlow with the appropriate compiler flags.

[0] | xC> ssh [redacted] 06:38 AM | Dec 09 - 2024 | x
```