

Entrada e saída em um training job

Dentre os experimentos realizados, os experimentos com o dataset MNIST e o CIFAR com o download dos datasets sendo feitos dentro do treinamento. Já o experimento feito com o dataset seismic o download foi feito dentro do momento de download do training job, com os dados sendo pegos por um bucket. Nesse último caso, a imagem do docker espera que os dados se encontrem num diretório específico pré-determinado pela aws. Em relação à saída os dados de saída são, no caso da CNN e da ResNet50, os logs do treinamento em formato de txt num bucket determinado na criação do treinamento e

Configuração da AWS

Para que sejam feitos os testes é preciso que seja instalada a aws-cli 2 (<https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html>). Depois disso, é necessário que se configure as credenciais e a região padrão do usuário com o `aws configure`.

Execução do treinamento:

A execução dos training jobs pode ser feita de duas formas, ou iniciando pela interface gráfica do console da aws ou pela aws cli, inicialmente os treinamentos da FCN eram feitos com a interface gráfica mas depois a forma usada predominante foi a da aws cli, isso porque os usos pela interface gráfica às vezes retornavam erros de network e dificultavam a automatização de treinamentos. Pela cli conseguimos usar comandos de bash para facilitar os treinamentos.

Script de teste:

O comando principal utilizado é o `aws sagemaker create-training-job` (<https://docs.aws.amazon.com/cli/latest/reference/sagemaker/create-training-job.html>), que inicia um treinamento no sagemaker. Dentre os argumentos temos:

- `training-job-name` : Nome do training job
- `hyper-parameters` : Parâmetros do treinamento, separados com vírgula e com a sintaxe `[Parâmetro]=[valor]`
- `algorithm-specification` : Argumentos com a mesma sintaxe dos parâmetros, mas predefinidos como:
 - `TrainingImage` : Endereço da imagem usada no treinamento dentro da ECR
 - `TrainingInputMode` : Modo de entrada do treinamento (como geralmente baixamos o modelo dentro do treinamento só deixo o valor como `File` pois é o valor pré-definido). No caso da

FCN o valor também é File, mas temos a entrada dos dados por meio de um bucket.

- `output-data-config` : Configurações da saída do treinamento, sintaxe parecida com os parâmetros mas assim como o `algorithm-specification` também com valores pré-definidos:
 - `S3OutputPath` : Em que bucket a saída do treinamento será armazenada.
- `resource-config` : Configurações das instâncias que serão utilizadas para rodar o treinamento:
 - `InstanceType` : Tipo da instância, iniciando com `ml`. (Exemplo: `ml.p2.xlarge`).
 - `InstanceCount` : Número de instâncias utilizadas no treinamento.
 - `VolumeSizeInGB` : Tamanho do armazenamento da instância para o treinamento.
- `stopping-condition` : Condições para que o treinamento pare, por padrão é obrigatório que seja colocada uma:
 - `MaxRuntimeInSeconds` : O valor padrão é 86400 segundos, o valor de um dia, mas como tivemos problemas com instâncias que não pararam, é recomendado colocar valores menores, como 3600.

Exemplo de comando em bash:

Exemplo de comando que cria um training job com o nome `resnet50-p3-gpu4-exp1` com 512 de batch size e 5 épocas, usando uma `p3.8xlarge` com 16gb de armazenamento (geralmente deixo entre 16 e 32gb por causa do tamanho da imagem, pelo que vi isso não faz uma alteração no custo do treinamento). Utilizei o iam role `arn:aws:iam::245983579475:role/service-role/AmazonSageMaker-ExecutionRole-20180829T145526` em todos os training jobs e para a resnet usamos a imagem do docker `245983579475.dkr.ecr.us-east-1.amazonaws.com/resnet50-cifar:latest`. A saída é no bucket `s3://fcnccloudml/` e vai ficar armazenada num diretório como o mesmo nome do treinamento num arquivo txt de log. A condição de parada é caso o treinamento passe de uma hora.

```
aws sagemaker create-training-job \
  --training-job-name resnet50-p3-gpu4-b512-e2 \
  --hyper-parameters batch-size=512,epochs=5 \
  --role-arn arn:aws:iam::245983579475:role/service-role/AmazonSageMaker-ExecutionRole-
  --algorithm-specification TrainingImage=245983579475.dkr.ecr.us-east-1.amazonaws.com/
  --output-data-config S3OutputPath=s3://fcnccloudml/ \
  --resource-config InstanceType=ml.p3.8xlarge,InstanceCount=1,VolumeSizeInGB=16 \
  --stopping-condition MaxRuntimeInSeconds=3600
```

Retirada do log

No caso da CNN/MNIST e da Resnet50/Cifar, as imagens armazenam a saída num txt de log, mas inicialmente como não fazíamos isso os treinamentos da FCN/Seismic tem o log retirado pela CloudWatch. Nesse caso temos um script chamado `get-log.sh` dentro do diretório `seismic-`

`fcn/experimental_results/2020-06-FCN-on-sagemaker` que recebe o nome de um log da cloudwatch e retorna a saída inteira do log ou a salva em um arquivo. Mudamos essa forma de retirar um log pois para experimentos com muitas épocas a Cloudwatch possui um limite no tamanho do log.