SageMaker Debugger Steps

Here are the steps you need to do in your model training script to use SageMaker Debugger:

Step 1: Import Amazon SageMaker Debugger client library, `SMDebug`.

```
import smdebug.pytorch as smd
```

Step 2a: In the `train()` function, add the `SMDebug` hook for PyTorch with `TRAIN` mode.

```
hook.set_mode(smd.modes.TRAIN)
```

Step 2b: In the `test()` function, add the `SMDebug` hook for PyTorch with `EVAL` mode.

```
hook.set_mode(smd.modes.EVAL)
```

Step 3a: In the `main()` function, create the `SMDebug` hook and register to the model.

```
hook = smd.Hook.create_from_json_file()
hook.register_hook(model)
```

Step 3b: In the `main()` function, pass the `SMDebug` hook to the `train()` and `test()` functions in the epoch loop.

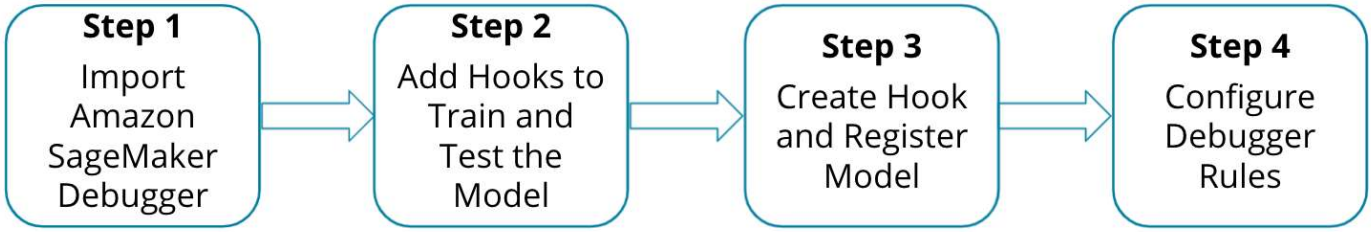
```
train(args, model, device, train_loader, optimizer, epoch, hook)
test(model, device, test_loader, hook)
```

Step 4: Configure Debugger Rules and Hook Parameters

```
from sagemaker.debugger import Rule, DebuggerHookConfig

rules = [
    Rule.sagemaker(rule_configs.vanishing_gradient()),
    Rule.sagemaker(rule_configs.overfit()),
    Rule.sagemaker(rule_configs.overtraining()),
    Rule.sagemaker(rule_configs.poor_weight_initialization()),
]

hook_config = DebuggerHookConfig(
    hook_parameters={"train.save_interval": "100", "eval.save_interval": "10"}
)
```



SageMaker Debugger Steps

Additional Resources

The following learning resources will help you better understand SageMaker's script mode.

- SageMaker Debugger documentation - Link
- Example of using Pytorch for model debugging - Link