Hyperparameter Tuning Steps

These are the steps you need to do to perform hyperparameter tuning in Sagemaker:

Step 1: Imports Hyperparameters can have continuous values like the learning rate, categorical values like the batch size, and integer values like the number of epochs. To work with all these kinds of values, import them like so:

```
from sagemaker.tuner import (
    IntegerParameter,
    CategoricalParameter,
    ContinuousParameter,
    HyperparameterTuner,
)
```

Step 2: Specify Hyperparameter Ranges Next, you need to specify the ranges of hyperparameters that you want Sagemaker to search through. Remember that in your training script, you will need to read command-line arguments with the same name as the key in the hyperparameter dictionary below:

```
hyperparameter_ranges = {
    "lr": ContinuousParameter(0.001, 0.1),
    "batch-size": CategoricalParameter([32, 64, 128, 256, 512]),
}
```

Step 3: Logging Training Results To let Sagemaker know how well the model is training for different values of hyperparameters, you will need to log the training results. You will also need to tell Sagemaker what to look for in the logs and whether Sagemaker needs to maximize or minimize that metric:

```
objective_metric_name = "average test loss"
objective_type = "Minimize"
metric_definitions = [{"Name": "average test loss", "Regex": "Test set: Averag
```

In the snippet above, you can see that I want to minimize my average test loss and I have specified that Sagemaker needs to search for the term `Test set: Average loss:` to look for the metric

Step 4: Creating your `HyperparameterTuner` You will also need to create a tuner object, where you need to pass your estimator, the objective metric name and definition, and the hyperparameter ranges. You also need to specify the max number of jobs to run and how many of those jobs can be run in parallel.

```
tuner = HyperparameterTuner(
    estimator,
    objective_metric_name,
    hyperparameter_ranges,
    metric_definitions,
    max_jobs=4,
    max_parallel_jobs=2,
    objective_type=objective_type,
)
```

Step 5: Training and Getting Results Finally, you can start training by calling the `fit()` method. You can also retrieve the best estimator and its hyperparameter as follows:

```
tuner.fit(wait=True)
tuner.best_estimator()
tuner.best_estimator().hyperparameters()
```

Additional Resources

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

- Docs for the types of Parameters supported by `HyperparameterTuner` in SageMaker: link
- Base `HyperparameterTuner` class documentation - Link
- Example of Hyperparamter Tuning using TensorFlow - Link
- Example of Hyperparameter Tuning using Pytorch - Link