Highlight Note

Hyperparameter Sampling

The specific values used in a hyperparameter tuning run depend on the type of *sampling* used.

Grid Sampling

Grid sampling can only be employed when all hyperparameters are discrete, and is used to try every possible combination of parameters in the search space.

For example, in the following code, grid sampling is used to try every possible combination of discrete *batch_size* and *learning_rate* value:

```
from azureml.train.hyperdrive import GridParameterSampling, choice

param_space = {
                '--batch_size': choice(16, 32, 64),
                '--learning_rate': choice(0.01, 0.1, 1.0)
              }

param_sampling = GridParameterSampling(param_space)
```

Random Sampling

Random sampling is used to randomly select a value for each hyperparameter, which can be a mix of discrete and continuous values:

```
from azureml.train.hyperdrive import RandomParameterSampling, choice, normal

param_space = {
                '--batch_size': choice(16, 32, 64),
                '--learning_rate': normal(10, 3)
              }

param_sampling = RandomParameterSampling(param_space)
```

Bayesian Sampling

Bayesian sampling chooses hyperparameter values based on the Bayesian optimization algorithm, which tries to select parameter combinations that will result in improved performance from the previous selection:

```
from azureml.train.hyperdrive import BayesianParameterSampling, choice, uniform

param_space = {
                '--batch_size': choice(16, 32, 64),
                '--learning_rate': uniform(0.5, 0.1)
              }

param_sampling = BayesianParameterSampling(param_space)
```

Note: You can only use Bayesian sampling with **choice**, **uniform**, and **quniform** parameter expressions, and you can't combine it with an early-termination policy.