✓ 100 XP

Configuring early termination

3 minutes

With a sufficiently large hyperparameter search space, it could take many iterations (child runs) to try every possible combination. Typically, you set a maximum number of iterations, but this could still result in a large number of runs that don't result in a better model than a combination that has already been tried.

To help prevent wasting time, you can set an early termination policy that abandons runs that are unlikely to produce a better result than previously completed runs. The policy is evaluated at an *evaluation_interval* you specify, based on each time the target performance metric is logged. You can also set a *delay_evaluation* parameter to avoid evaluating the policy until a minimum number of iterations have been completed.

Note

Early termination is particularly useful for deep learning scenarios where a deep neural network (DNN) is trained iteratively over a number of *epochs*. The training script can report the target metric after each epoch, and if the run is significantly underperforming previous runs after the same number of intervals, it can be abandoned.

Bandit policy

You can use a bandit policy to stop a run if the target performance metric underperforms the best run so far by a specified margin.

Python Copy

```python
from azureml.train.hyperdrive import BanditPolicy

early_termination_policy = BanditPolicy(slack_amount = 0.2,
                                        evaluation_interval=1,
                                        delay_evaluation=5)
```

This example applies the policy for every iteration after the first five, and abandons runs where the reported target metric is 0.2 or more worse than the best performing run after the same number of intervals.

You can also apply a bandit policy using a slack *factor*, which compares the performance metric as a ratio rather than an absolute value.

Median stopping policy

A median stopping policy abandons runs where the target performance metric is worse than the median of the running averages for all runs.

Python Copy

```python
from azureml.train.hyperdrive import MedianStoppingPolicy

early_termination_policy = MedianStoppingPolicy(evaluation_interval=1,
                                                delay_evaluation=5)
```

Truncation selection policy

A truncation selection policy cancels the lowest performing *X*% of runs at each evaluation interval based on the *truncation_percentage* value you specify for *X*.

Python Copy

```python
from azureml.train.hyperdrive import TruncationSelectionPolicy

early_termination_policy = TruncationSelectionPolicy(truncation_percentage=10,
                                                     evaluation_interval=1,
                                                     delay_evaluation=5)
```

Next unit: Running a hyperparameter tuning experiment

Continue >