HyperUCB: Hyperparameter Optimization using Contextual Bandits

Maryam Tavakol, Sebastian Mair, Katharina Morik
maryam.tavakol@tu-dortmund.de, mair@leuphana.de, katharina.morik@tu-dortmund.de

Summary
• HyperUCB is a contextual bandit extension for Hyperband
• Hyperparameters are pre-evaluated using a UCB strategy
• Only the \( k \)-best configurations are actually evaluated
⇒ The sampling is guided towards more promising area

Motivation
• Performance in ML highly depends on the hyperparameters
• Hyperparameters are usually tuned via grid/random search
• Hyperband (HB) speeds up random search while optimizing computational resources
• but HB does not leverage the information of previous runs

Contextual HyperUCB
HyperUCB extends HB to contextual bandit setting:
• Given a fixed budget \( B \)
• allocate \( B \) to \( s_{\text{max}} \) number of iterations
• in every iteration \( s \):
  - sample \( n \) configurations
  - sample \( n \) configurations and choose \( n \), with highest UCB
  - compute validation loss for configurations until \( B \) is exhausted
  - run successive halving to choose top-\( k \)
  - run contextual UCB to choose top-\( k \)
⇒ Learn a model to keep track of previous evaluations
⇒ Only execute promising hyperparameter configurations

Experiments
• MNIST data of handwritten digits (60k train, 10k test)
• Multi-layer perceptron with categorical cross entropy loss
• Validation loss is evaluated on the test data
• Minimum budget corresponds to 100 mini-batches of size 100
• The model has four hyperparameters:

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Range</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>learning rate</td>
<td>[0.0001, 1]</td>
<td>float</td>
</tr>
<tr>
<td># hidden layers</td>
<td>{1, 2, 3, 4, 5}</td>
<td>integer</td>
</tr>
<tr>
<td># neurons</td>
<td>{16, 32, \ldots, 512}</td>
<td>integer</td>
</tr>
<tr>
<td>activation</td>
<td>(relu, tanh, sigmoid)</td>
<td>categorical</td>
</tr>
</tbody>
</table>

⇒ HyperUCB outperforms HB for a budget greater than 19
⇒ HyperUCB is as fast or faster than Hyperband

Future Work
• Use a kernelized bandit to capture non-linearity
• Derive theoretical regret bounds
• Extend the experimental setup