Hyperparameter optimization is bi-level optimization.

The optimal weights are a best-response function to the hyperparameters.

Can learn a differentiable approximation to continuous best-response’s using hypernets without seeing labeled tuples of (hyperparameter, optimized weights).

Can optimize thousands of hyperparameters with joint updates to hyperparameters and weights using best-response.

\[
\argmin_{\lambda} \min_{w} \mathcal{L} \left( \argmin_{w} \mathcal{L}_{\text{Train}}(w, \lambda) \right)
\]

\[
w^*(\lambda) = \argmin_{w} \mathcal{L}_{\text{Train}}(w, \lambda)
\]

1. initialize \(\phi, \hat{\lambda}\)
2. for \(T_{\text{joint}}\) steps do
3. \(x \sim \text{Training data}, \lambda \sim p(\lambda|\hat{\lambda})\)
4. \(\phi = \phi - \alpha \nabla_{\phi} \mathcal{L}_{\text{Train}}(x, w_{\phi}(\hat{\lambda}), \hat{\lambda})\)
5. \(x \sim \text{Validation data}\)
6. \(\hat{\lambda} = \hat{\lambda} - \beta \nabla_{\hat{\lambda}} \mathcal{L}_{\text{Valid.}}(x, w_{\phi}(\hat{\lambda}))\)
7. return \(\hat{\lambda}, w_{\phi}(\hat{\lambda})\)