Hyperparameter Optimization with Hypernets

Jonathan Lorraine, David Duvenaud

Hyperparameter optimization is bi-level optimization.

$$\underset{\lambda}{\operatorname{argmin}} \underset{\text{Valid.}}{\mathcal{L}} \left(\underset{\mathrm{w}}{\operatorname{argmin}} \underset{\text{Train}}{\mathcal{L}} (\mathrm{w}, \lambda) \right)$$

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

$$\mathrm{w}^*(\lambda) = \underset{\mathrm{w}}{\operatorname{argmin}} \underset{\text{Train}}{\mathcal{L}} (\mathrm{w}, \lambda)$$

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.

initialize $\phi, \hat{\lambda}$
for T_{joint} steps **do**
  $\mathbf{x} \sim$ Training data, $\lambda \sim p(\lambda|\hat{\lambda})$
  $\phi = \phi - \alpha \nabla_{\phi} \mathcal{L}_{\text{Train}}(\mathbf{x}, \mathrm{w}_{\phi}(\hat{\lambda}), \hat{\lambda})$
  $\mathbf{x} \sim$ Validation data
  $\hat{\lambda} = \hat{\lambda} - \beta \nabla_{\hat{\lambda}} \mathcal{L}_{\text{Valid.}}(\mathbf{x}, \mathrm{w}_{\phi}(\hat{\lambda}))$
return $\hat{\lambda}, \mathrm{w}_{\phi}(\hat{\lambda})$