Optimizing Millions of Hyperparameters with Implicit Differentiation
Jonathan Lorraine, Paul Vicl, David Duvenaud

Motivation and Contributions
- Tuning hyperparameters can be critical for generalization, but difficult when hyperparameters are high-dimensional.
- If we can tune as many hyperparameters as weights, we have a new paradigm of learned, flexible regularization.
- Learned data augmentation
- Data distillation
- Tuning millions of regularization hyperparameters
- We scale gradient-based hyperparameter optimization to high dimensions by combining implicit differentiation with efficient & stable inverse-Hessian approximations.

Hyperparameter Optimization is Nested Optimization
\[ \lambda^* := \arg \min_{\lambda} C(\lambda) \]
where \( C(\lambda) := C(\lambda, w(\lambda)) \) and \( w(\lambda) := \arg \min_w C(\lambda, w) \)

Hypergradients with Implicit Differentiation
\[
\frac{\partial C(\lambda)}{\partial \lambda} = \frac{\partial C(\lambda)}{\partial w} \frac{\partial w}{\partial \lambda} + \nabla_w C(\lambda, w(\lambda)) \frac{\partial w}{\partial \lambda}
\]

Inverse-Hessian approximations via Neumann series
\[
\left( \frac{\partial^2 C(\lambda)}{\partial w^2} \right)^{-1} = \lim_{i \to \infty} \sum_{i=0}^{\infty} \left( - \frac{\partial^2 C(\lambda)}{\partial w^2} \right)^i
\]

Calculating Hypergradients Efficiently
1. while not converged do
2. for \( k = 1 \ldots N \) do
3. \( \lambda^T \leftarrow \frac{\partial C(\lambda)}{\partial \lambda} \lambda^T \)
4. \( \lambda^T \leftarrow \text{hypergradient}(C_{\lambda}, C_{\lambda^T}, \lambda^T, \lambda^T) \)
5. return \( \lambda^T, \lambda^T \)

Data Distillation
Plane Car Bird CIFAR-10 Distillation
Cat Deer Frog Horse Ship Truck
Chemistry
MNIST Distillation

Learned Data Augmentation
Original Sample 1 Sample 2 Pixel Std.

Tuning Regularization Parameters

<table>
<thead>
<tr>
<th>Method</th>
<th>Validation</th>
<th>Test</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid Search</td>
<td>97.32</td>
<td>94.58</td>
<td>100k</td>
</tr>
<tr>
<td>Random Search</td>
<td>84.81</td>
<td>81.46</td>
<td>100k</td>
</tr>
<tr>
<td>Bayesian Opt.</td>
<td>72.13</td>
<td>69.29</td>
<td>100k</td>
</tr>
<tr>
<td>STN</td>
<td>70.30</td>
<td>67.68</td>
<td>25k</td>
</tr>
<tr>
<td>Ours, Many</td>
<td>68.18</td>
<td>66.40</td>
<td>18.5k</td>
</tr>
</tbody>
</table>

For LSTMs on PTB, we tune the same 7 (and millions of) hyperparameters faster, with comparable memory, and to a lower perplexity.