Meta Dropout
Learning to Perturb Features for Generalization

Hae Beom Lee, Taewook Nam, Eunho Yang, Sung Ju Hwang

Tobias Schmidt
Munich, 7th July 2021
Motivation – Generalization

- Low Variance
  - Low Bias
  - High Bias

- High Variance

\[ \lambda = 1 \]
\[ \lambda = 0 \]
\[ M = 9 \]
\[ \lambda = 10^{-5} \]
Motivation – Generalization

Common Solutions - Bias/Variance Trade-Off

Appropriate Priors
- Image Convolutions (Shift-Invariance)
- Graph Convolutions (Permutation-Invariance)
- …

Regularization
- Reducing model capacity
- Reducing information from inputs
- Smoothing loss surface
- Multi-task training
- Meta-Learning
Motivation – Train/Test Distribution Mismatch

Common Solutions - Bias/Variance Trade-Off

**Appropriate Priors**
- Image Convolutions (Shift-Invariance)
- Graph Convolutions (Permutation-Invariance)
- ...

**Regularization**
- Reducing model capacity
- Reducing information from inputs
- Smoothing loss surface
- Multi-task training
- Meta-Learning
Motivation – Train/Test Distribution Mismatch

Alternative Solutions - Simulate Test Samples

Data augmentations

Mixup
Motivation – Train/Test Distribution Mismatch

Alternative Solutions - Simulate Test Samples

Data augmentations

Mixup
Motivation – Train/Test Distribution Mismatch

Alternative Solutions - Simulate Test Samples

Data augmentations  Mixup
Idea: Learn to perturb the data for better generalization
Challenges

A training instance may need to cover multiple test instances

Meaningful directions differ from one task to another
Challenges

A training instance may need to cover multiple test instances

Meaningful directions differ from one task to another
Challenges

A training instance may need to cover multiple test instances

Meaningful directions differ from one task to another
Challenges

A training instance may need to cover multiple test instances

→ Noise Distribution

Meaningful directions differ from one task to another

→ Input-Dependent Noise
**Meta Learning Framework - MAML**

### Properties / Limitations of MAML

- Knowledge transfer via learned parameter $\theta$
- Parameter $\theta$ only implicitly captures test distributions
- Misses out on important knowledge about task distribution
Recap: MAML

Update over task distribution:
- **initial parameter** $\theta$

**Algorithm 1** Model-Agnostic Meta-Learning

Require: $p(T)$: distribution over tasks

Require: $\alpha$, $\beta$: step size hyperparameters

1: randomly initialize $\theta$

2: while not done do

3: Sample batch of tasks $T_i \sim p(T)$

4: for all $T_i$ do

5: Evaluate $\nabla_\theta \mathcal{L}_{T_i}(f_\theta)$ with respect to $K$ examples

6: Compute adapted parameters with gradient descent: $\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{T_i}(f_\theta)$

7: end for

8: Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{T_i \sim p(T)} \mathcal{L}_{T_i}(f_{\theta'_i})$

9: end while
Meta Dropout

**Algorithm 1 Meta-training**

1: **Input:** Task distribution $p(T)$, Number of inner steps $K$.
2: **while** not converged **do**
3: Sample $(D^r, D^{te}) \sim p(T)$
4: $\theta_0 \leftarrow \theta$
5: **for** $k = 0$ to $K - 1$ **do**
6: Sample $\bar{z}_i \sim p(z_i|x_i^r; \phi, \theta_k)$ for $i = 1, \ldots, N$
7: $\theta_{k+1} \leftarrow \theta_k + \alpha \nabla \theta_k \frac{1}{N} \sum_{i=1}^{N} \log p(y_i^{te}|x_i^{te}, \bar{z}_i; \theta_k)$
8: **end for**
9: $\theta^* \leftarrow \theta_K$
10: $\theta \leftarrow \theta + \beta \frac{1}{M} \sum_{j=1}^{M} \nabla \theta \log p(y_j^{te}|x_j^{te}, z_j = \bar{z}_j; \theta^*)$
11: $\phi \leftarrow \phi + \beta \frac{1}{M} \sum_{j=1}^{M} \nabla \phi \log p(y_j^{te}|x_j^{te}, z_j = \bar{z}_j; \theta^*)$
12: **end while**

Update over task distribution:
- initial parameter $\theta$
- parameters $\phi$ of noise $p(z)$

Perform Few-Shot Learning for each task perturbing the input $x_i^{tr}$ with multiplicative noise $\bar{z}_i$
Meta-Dropout: Model Architecture
Meta-Dropout: Implementation Detail

**Model Architecture**

**Reparameterization Trick**

\[
\log p(Y_{i|tr} | X_{i|tr}, \theta, \phi) \geq \sum_{i=1}^{N} \mathbb{E}_{z_i \sim p(z_i|X_{i|tr}, \theta, \phi)} [\log p(y_{i|tr} | x_{i|tr}, \theta, \phi)]
\]

\[
\approx \sum_{i=1}^{N} \sum_{s=1}^{S} \log p(y_{i|tr} | x_{i|tr}, z_i^{(s)}, \theta) \text{ with } z_i^{(s)} \sim p(z_i|X_{i|tr}, \theta, \phi)
\]

**Approximation via Monte-Carlo**

Impossible to calculate gradient!
**Meta-Dropout: Implementation Detail**

**Model Architecture**

**Form of the Noise:** 
\[ p \left( y_i^{tr} \mid x_i^{tr}, z_i^{(s)}, \theta \right) \] with \[ z_i^{(s)} \sim p(z_i \mid x_i^{tr}, \theta, \phi) \]

**Additive Noise**

- \( h^{(0)} = x_i^{tr} \)
- \( h^{(l)} = ReLU(f^{(l)}(h^{(l-1)}) + z^{(l)}) \)
- \( z_i^{(s)} \sim N(z^{(l)} \mid 0, \lambda^2 \mathbf{diag}(\sigma^2)) \)

**Multiplicative Noise**

- \( h^{(0)} = x_i^{tr} \)
- \( h^{(l)} = ReLU(f^{(l)}(h^{(l-1)}) \circ z^{(l)}) \)
- \( z^{(l)} = \text{Softplus}(a^{(l)}) \)
- \( a^{(l)} \sim N(a^{(l)} \mid \mu^{(l)}, \mathbf{I}) \)
Experiments - Datasets

Omniglot

- Handwritten character classification
- 20 instances of ~1600 characters from 50 alphabets

minImageNET

- Small version of ImageNET
- 100 classes with 600 samples
### Experiments

#### Meta-Learning Frameworks

<table>
<thead>
<tr>
<th>MAML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meta-SGD</td>
</tr>
<tr>
<td>Prototypical Networks</td>
</tr>
<tr>
<td>Matching Networks</td>
</tr>
</tbody>
</table>

- Reptile
- Amortized Bayesian ML
- Probabilistic MAML
- MT-NET
- CAVIA
Experiments

Perturbation Based Methods:

- Input & Manifold Mixup
- Variational Information Bottleneck
- Information Dropout

Variational Information Bottleneck

Information Dropout

Encode maximal information about Target $Y$ in latent stochastic encoding $Z$ measured by mutual information $I(Z, Y) < I_c$, where $I_c$ is information constraint.
Few-shot classification performance

<table>
<thead>
<tr>
<th>Models</th>
<th>Omniglot 20-way</th>
<th>miniImageNet 5-way</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-shot</td>
<td>5-shot</td>
</tr>
<tr>
<td>Meta-Learning LSTM (Ravi &amp; Larochelle, 2017)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Matching Networks (Vinyals et al., 2016)</td>
<td>93.8</td>
<td>98.7</td>
</tr>
<tr>
<td>Prototypical Networks (Snell et al., 2017)</td>
<td>95.4</td>
<td>98.7</td>
</tr>
<tr>
<td>Prototypical Networks (Snell et al., 2017) (Higher way)</td>
<td>96.0</td>
<td>98.9</td>
</tr>
<tr>
<td>MAML (our reproduction)</td>
<td>95.23±0.17</td>
<td>98.38±0.07</td>
</tr>
<tr>
<td>Meta-SGD (our reproduction)</td>
<td>96.16±0.14</td>
<td>98.54±0.07</td>
</tr>
<tr>
<td>Reptile (Nichol et al., 2018)</td>
<td>89.43±0.14</td>
<td>97.12±0.32</td>
</tr>
<tr>
<td>Amortized Bayesian ML (Ravi &amp; Beatson, 2019)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Probabilistic MAML (Finn et al., 2018)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MT-Net (Lee &amp; Choi, 2018)</td>
<td>96.2±0.4</td>
<td></td>
</tr>
<tr>
<td>CAVIA (512) (Zintgraf et al., 2019)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAML + Meta-dropout</td>
<td>96.63±0.13</td>
<td>98.73±0.06</td>
</tr>
<tr>
<td>Meta-SGD + Meta-dropout</td>
<td>97.02±0.13</td>
<td>99.05±0.05</td>
</tr>
</tbody>
</table>
## Few-shot classification performance

<table>
<thead>
<tr>
<th>Models (MAML +)</th>
<th>Noise Type</th>
<th>Hyper-parameter</th>
<th>Omniglot 20-way 1-shot</th>
<th>5-shot</th>
<th>miniImageNet 5-way 1-shot</th>
<th>5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>No perturbation</td>
<td>None</td>
<td></td>
<td>95.23±0.17</td>
<td>98.38±0.07</td>
<td>49.58±0.65</td>
<td>64.55±0.52</td>
</tr>
<tr>
<td>Input &amp; Manifold Mixup (Zhang et al., 2017)</td>
<td>Pairwise</td>
<td>γ = 0.2</td>
<td>89.78±0.25</td>
<td>97.86±0.08</td>
<td>48.62±0.66</td>
<td>63.86±0.53</td>
</tr>
<tr>
<td>(Verma et al., 2019)</td>
<td></td>
<td>γ = 1</td>
<td>87.00±0.28</td>
<td>97.27±0.10</td>
<td>48.24±0.62</td>
<td>62.32±0.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>γ = 2</td>
<td>87.26±0.28</td>
<td>97.14±0.17</td>
<td>48.42±0.64</td>
<td>62.56±0.55</td>
</tr>
<tr>
<td>Variational Information Bottleneck (Alemi et al., 2017)</td>
<td>Add.</td>
<td>β = 10⁻⁵</td>
<td>92.09±0.22</td>
<td>98.85±0.07</td>
<td>48.12±0.65</td>
<td>64.78±0.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>β = 10⁻⁴</td>
<td>93.01±0.20</td>
<td>98.80±0.07</td>
<td>46.75±0.63</td>
<td>64.07±0.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>β = 10⁻³</td>
<td>94.98±0.16</td>
<td>98.75±0.07</td>
<td>47.59±0.60</td>
<td>63.30±0.53</td>
</tr>
<tr>
<td>Information Dropout (ReLU ver.) (Achille &amp; Soatto, 2018)</td>
<td>Mult.</td>
<td>β = 10⁻⁵</td>
<td>94.49±0.17</td>
<td>98.50±0.07</td>
<td>50.36±0.68</td>
<td>65.91±0.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>β = 10⁻⁴</td>
<td>94.36±0.17</td>
<td>98.53±0.07</td>
<td>49.14±0.63</td>
<td>64.96±0.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>β = 10⁻³</td>
<td>94.28±0.17</td>
<td>98.65±0.07</td>
<td>43.78±0.61</td>
<td>63.36±0.56</td>
</tr>
<tr>
<td>Meta-dropout (See Appendix B for Add.)</td>
<td>Add.</td>
<td>0.1</td>
<td>96.55±0.14</td>
<td>99.04±0.05</td>
<td>50.25±0.66</td>
<td>66.78±0.53</td>
</tr>
<tr>
<td></td>
<td>Mult.</td>
<td>None</td>
<td>96.63±0.13</td>
<td>98.73±0.06</td>
<td>51.93±0.67</td>
<td>67.42±0.52</td>
</tr>
</tbody>
</table>
Ablation study on the noise type

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>95.23 ± 0.17</td>
<td>98.38 ± 0.07</td>
<td>49.58 ± 0.65</td>
<td>64.55 ± 0.52</td>
</tr>
<tr>
<td>Fixed Gaussian (✓)</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>95.44 ± 0.17</td>
<td>98.99 ± 0.06</td>
<td>49.39 ± 0.63</td>
<td>66.84 ± 0.54</td>
</tr>
<tr>
<td>Weight Gaussian</td>
<td>O</td>
<td>X</td>
<td>X</td>
<td>94.32 ± 0.18</td>
<td>98.35 ± 0.07</td>
<td>49.37 ± 0.64</td>
<td>64.78 ± 0.54</td>
</tr>
<tr>
<td>Independent Gaussian</td>
<td>O</td>
<td>O</td>
<td>X</td>
<td>94.36 ± 0.18</td>
<td>98.26 ± 0.08</td>
<td>50.31 ± 0.64</td>
<td>66.97 ± 0.54</td>
</tr>
<tr>
<td>MAML + More param</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>95.83 ± 0.15</td>
<td>97.85 ± 0.09</td>
<td>50.63 ± 0.64</td>
<td>65.20 ± 0.51</td>
</tr>
<tr>
<td>Determin. Meta-drop. (✓)</td>
<td>X</td>
<td>O</td>
<td>O</td>
<td>95.99 ± 0.14</td>
<td>97.78 ± 0.09</td>
<td>50.75 ± 0.63</td>
<td>65.62 ± 0.53</td>
</tr>
<tr>
<td>Meta-drop. w/ learned var.</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>95.98 ± 0.15</td>
<td>98.87 ± 0.06</td>
<td>50.93 ± 0.68</td>
<td>66.15 ± 0.56</td>
</tr>
<tr>
<td>Meta-dropout</td>
<td>O</td>
<td>O</td>
<td>O</td>
<td>96.63 ± 0.13</td>
<td>98.73 ± 0.06</td>
<td>51.93 ± 0.67</td>
<td>67.42 ± 0.52</td>
</tr>
</tbody>
</table>
Adversarial Robustness

(a) Omniglot 1-shot ($\ell_1$)  (b) Omniglot 1-shot ($\ell_2$)  (c) Omniglot 1-shot ($\ell_{\infty}$)  (d) Omniglot 5-shot ($\ell_{\infty}$)
Qualitative study on generalization capability

(a) MAML (miniImageNet)  (b) Meta-dropout (miniImageNet)  (c) Meta-dropout (Omniglot)
Conclusion

Main Claim:

“Using Meta-Dropout to perturb the latent features of training examples in a Meta-Learning Framework improves generalization capabilities”

Improves:
- Decision boundary
- Adversarial robustness
- Few-Shot learning performance
- Hypothesis supported by experiments across large variate of baseline models
- Code available

• Evaluated on only two datasets
• More Shots / More Ways
• Relatively small performance increase
• Discrepancy in the results
• Generalization across datasets domains not discussed
• No Computational Cost Reported
• Comparison noise $\phi$ for every layer or shared for all
References


