Bayesian Hypernetworks

David Krueger*, Chin-Wei Huang*
Riahsat Islam, Ryan Turner, Aaron Courville
MILA and McGill RLLAB
What’s a Bayesian Hypernet?

Hypernet: a DNN that generates params of another DNN (the “primary net”)

Task: predict y from x

Think GAN / VAE / Real NVP
What is a Bayesian Neural Net?

Bayes Rule:

\[ p(\theta|D) \propto p(D|\theta)p(\theta) \]

Argmax

Predict using ensemble:

\[ p(y|x) = \int p(y|x, \theta)p(\theta|D)d\theta \]
What’s **special** about Bayesian Neural Nets?

Bayes Rule:

\[ p(\theta | D) \propto p(D | \theta)p(\theta) \]

“**That’s my best guess**”

“I’m 99% sure!”

“Knows what it knows”

“Calibrated confidence”

“Weight Uncertainty in Neural Networks” - Blundell et al 2015
Example: self-driving cars

Q: Is there a person in the road?
Car: No, and….

“That’s my best guess”

“I’m 51% sure!”
“I’m 99.999999% sure!”
Humans want?

AI Safety

“Is the default outcome doom?”

Existential risk

What Would Jesus Do?

Humans want?

KEEP CALM
AND
ASK A HUMAN

NICK BOSTROM
SUPERINTELLIGENCE
Paths, Dangers, Strategies
Concrete Problems in AI Safety

- Five “concrete problems”, calibrated confidence helps in 4/5
- **Avoiding Negative Side Effects:** How can we ensure that our cleaning robot will not disturb the environment in negative ways while pursuing its goals, e.g., by knocking over a vase because it can clean faster by doing so? Can we do this without manually specifying everything the robot should not disturb?

- **Avoiding Reward Hacking:** How can we ensure that the cleaning robot won’t game its reward function? For example, if we reward the robot for achieving an environment free of messes, it might disable its vision so that it won’t find any messes, or cover over messes with materials it can’t see through, or simply hide when humans are around so they can’t tell it about new types of messes.

- **Scalable Oversight:** How can we efficiently ensure that the cleaning robot respects aspects of the objective that are too expensive to be frequently evaluated during training? For instance, it should throw out things that are unlikely to belong to anyone, but put aside things that might belong to someone (it should handle stray candy wrappers differently from stray cellphones). Asking the humans involved whether they lost anything can serve as a check on this, but this check might have to be relatively infrequent – can the robot find a way to do the right thing despite limited information?

- **Safe Exploration:** How do we ensure that the cleaning robot doesn’t make exploratory moves with very bad repercussions? For example, the robot should experiment with mopping strategies, but putting a wet mop in an electrical outlet is a very bad idea.

- **Robustness to Distributional Shift:** How do we ensure that the cleaning robot recognizes, and behaves robustly, when in an environment different from its training environment? For example, heuristics it learned for cleaning factory workfloors may be outright dangerous in an office.
Technique
Variational Inference for Bayesian DNNs

- ELBO:

\[
\mathcal{L}(\phi) = \mathbb{E}_{q_\phi(\theta)}[\log p(\mathcal{D}|\theta)+\log p(\theta)–\log q_\phi(\theta)]
\]

\[
\log p(\mathcal{D}) = \mathcal{L}(\phi) + KL(q_\phi(\theta)||p(\theta|\mathcal{D}))
\]

constant maximize minimize

Encourages stochasticity!

- Examples:
  - Weight Uncertainty
  - Variational Dropout / MC dropout
Problem with Variational Inference: KL divergence

Variational inference can **underestimate** uncertainty!

\[ KL(p(\theta|D)||q_\phi(\theta)) \]

\[ KL(q_\phi(\theta)||p(\theta|D)) \]

\[ P = \text{true posterior} \]
\[ \text{(mixture of Gaussians)} \]

\[ Q = \text{variational approx} \]
\[ \text{(Gaussian)} \]
Are Bayesian Hypernets the solution?

- **Previous work:** approximate posterior is **factorial**:
  \[ q(\theta | \mathcal{D}) = \prod_i q(\theta_i | \mathcal{D}) \]
- **Use a DNN!**
  - \( \Rightarrow q(\theta | \mathcal{D}) \) can be **dependent, multimodal**
  - \( z \sim \mathcal{N}(0, 1) \)
  - \( y = f_\theta(x) \)
  - \( \theta = h(z) \)

**Note:** \( h \) must be invertible!
...but the image of \( h \) can be a subset of \( \mathbb{R}^{|\theta|} \), unlike with NICE (generative model)
Some Qualitative Results:

**Multimodality**

![Multimodality Diagram]

**Correlation**

![Correlation Diagram]
Background: Hypernetworks


“HyperNetworks” - Ha et al. 2016
Background: Weight Normalization

Reparameterization

- Express weights as function of new parameters
  \[ w = \frac{g}{||v||} v \]
- Minimize loss with respect to new parameters \( v, b, g \)
- Decouples direction and length of weight vector

“Weight Normalization” - Salimans and Kingma (slide from NIPS 2016 talk)
Background: Invertible Deep Generative Models

Key property: tractable likelihood (via change of variable):

\[ \frac{p_X(x)}{p_z(z)} \left| \det \left( \frac{\partial g(z)}{\partial z^T} \right) \right|^{-1} \]

(figure and equation: “Density Estimation via Real NVP” - Dinh et al. 2016)
Some results (5000 examples of MNIST):

<table>
<thead>
<tr>
<th>No. of Coupling Layers</th>
<th>Test Accuracy</th>
<th>No. of Coupling Layers</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>92.06%</td>
<td>0</td>
<td>90.91%</td>
</tr>
<tr>
<td>8</td>
<td>94.25%</td>
<td>8</td>
<td>96.27%</td>
</tr>
<tr>
<td>12</td>
<td>96.16%</td>
<td>12</td>
<td>96.51%</td>
</tr>
<tr>
<td>dropout</td>
<td>95.58%</td>
<td>dropout</td>
<td>95.52%</td>
</tr>
</tbody>
</table>

Table 2: Generalization results on subset (5000 training data) of MNIST. (A) MLP with 800 hidden nodes. (B) MLP with 1200 hidden nodes.
QUESTIONS?

Kill all humans??