Building neural networks that know what they don’t know

Balaji Lakshminarayanan
balajiln@

Google Research, Brain Team
Motivation
What do we mean by Uncertainty?

Return a distribution over predictions rather than a single prediction.

- **Classification**: Output label along with its confidence.
- **Regression**: Output mean along with its variance.

Good uncertainty estimates quantify *when we can trust the model’s predictions*. 

*Image credit: Eric Nalisnick*
What do we mean by Out-of-Distribution Robustness?

I.I.D. \[ p_{\text{TEST}}(y,x) = p_{\text{TRAIN}}(y,x) \]

(Independent and Identically Distributed)

O.O.D. \[ p_{\text{TEST}}(y,x) \neq p_{\text{TRAIN}}(y,x) \]

Examples of dataset shift:

- **Covariate shift.** Distribution of features \( p(x) \) changes and \( p(y|x) \) is fixed.
- **Open-set recognition.** New classes may appear at test time.
- **Subpopulation shift.** Frequencies of data subpopulations changes.
- **Label shift.** Distribution of labels \( p(y) \) changes and \( p(x|y) \) is fixed.
ImageNet-C: Varying Intensity for Dataset Shift

Increasing “OODness”

“I.I.D test set

“Distance” between $p_{\text{TEST}}(y,x)$ and $p_{\text{TRAIN}}(y,x)$ increases

Neural networks do not generalize under covariate shift

- **Accuracy drops** with increasing shift on Imagenet-C

- But do the models know that they are less accurate?
Neural networks **do not know when they don’t know**

- **Expected Calibration error (↓)**

  Calibration Error = |Confidence - Accuracy|

  - predicted probability of correctness
  - observed frequency of correctness

- **Quality of uncertainty degrades with shift -> “overconfident mistakes”**

Models assign high confidence predictions far away from training data.

Ideal desired behavior

Trust model when $x^*$ is close to $p_{\text{TRAIN}}(x,y)$

Deep neural networks assign high confidence predictions far away from $p_{\text{TRAIN}}(x,y)$

Image source: “Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness” Liu et al. 2020
Models assign high confidence predictions far away from training data

Ideal desired behavior

Trust model when \( x^* \) is close to \( p_{\text{TRAIN}}(x,y) \)

Deep neural networks assign high confidence predictions far away from \( p_{\text{TRAIN}}(x,y) \)

*Image source: “Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness”* Liu et al., 2020
Applications
Healthcare

- Use model uncertainty to decide when to trust the model or to defer to a human.
- Selective prediction, Cost-sensitive decision making

Diabetic retinopathy detection from fundus images

Gulshan et al, 2016
Self-driving cars

Dataset shift:

- Time of day / Lighting
- Geographical location (City vs suburban)
- Changing conditions (Weather / Construction)

*Image credit: Sun et al, Waymo Open Dataset*
Open Set Recognition

1. Take a photo

Model

Acne
Alopecia Areata
Cyst
Eczema
Psoriasis

Open Set Recognition

Melanoma

Unsupported condition

Test input may not belong to one of the K training classes.

High I.I.D. accuracy is not sufficient, need to be able to detect OOD inputs.

Image source: https://blog.google/technology/health/ai-dermatology-preview-io-2021/
Active Learning

- Use model uncertainty to improve data efficiency and model performance in blindspots.

*Image source: Active Learning Literature Survey, Settles 2010*
Bayesian Optimization and Experimental Design

- Hyperparameter optimization and experimental design
  - Used across large organizations and the sciences
- **Photovoltaics, chemistry experiments, AlphaGo, batteries, materials design**

*Image source: Attia et al. 2020 Closed-loop optimization of fast-charging protocols for batteries with machine learning*
Bandits and Reinforcement Learning

- Decision making with asymmetric losses
  \[ \ell(\mu) \neq \mathbb{E}_{z \sim N(\mu, \sigma^2)}[\ell(z)] \]

- Modeling uncertainty is crucial for **exploration vs exploitation** trade-off

- Non-stationarity

*Image source: David Silver's RL course*
All models are wrong, but some models that know when they are wrong, are useful.
Overview of Methods
Probabilistic Deep Learning

- Parametrize “base model”.
- Specify prior over functions.
- Capture model uncertainty by approximating the posterior.
- Average predictions over multiple functions (ensemble or Bayesian NN)

\[
p(y \mid x, \theta) \quad p(\theta) \quad p(\theta \mid D) \quad p(y \mid x) = \frac{1}{M} \sum_{m=1}^{M} p(y \mid x, \theta_m)
\]
Cartoon: Uncertainty/Robustness vs Compute frontier

Practitioners can pick “operating point” depending on constraints of application.

- Probabilistic framework gives an unifying view. Improving $p(y|x)$ improves performance on all downstream tasks (accuracy/calibration under shift, selective prediction, open set recognition, etc) as opposed to custom techniques for tasks.
Orthogonal ways of improving performance

Ensembles / Bayesian NNs

$$p(y | x) = \frac{1}{M} \sum_{m=1}^{M} p(y | x, \theta_m)$$

“Composable” toolkit

1. Improve the “base” model $p(y|x,\theta)$
2. Efficiently average predictions over diverse set of functions $\theta_1, \theta_2 \ldots \theta_M$
3. Better inductive biases for representations (e.g. pre-training or data augmentation)

Composing can further improve performance!
Improving Single Model Uncertainty via Distance Awareness

Jeremiah Liu (jereliu@) et al.
SNGP improves single model uncertainty with two simple changes: Spectral-normalization (SN) + Last-layer Gaussian Process (GP)
Why do vanilla DNNs assign high confidence predictions far away from training data?

1. Vanilla NNs can map inputs far away in input space to close points in latent space (cf. “shortcut learning”).

2. Confidence is a function of distance from boundary (and not the training data).
Idea 1: Enforce bi-Lipschitz smoothness via spectral normalization

Bi-Lipschitz smoothness discourages inputs far away in input space getting mapped close in latent space.
**Idea 1**: Enforce bi-Lipschitz smoothness

**Idea 2**: Replace last dense layer with “Gaussian process” layer

Bi-Lipschitz smoothness discourages inputs far away in input space getting mapped close in latent space.

For GP layer, confidence is a function of distance from the training data.
Spectral-normalized Neural Gaussian process (SNGP)

- SNGP improves
  - Accuracy under shift
  - Calibration under shift
  - OOD detection
- Simple to implement
- Low computational/memory overhead
- A building block for better ensembles

See also [van Amersfoort+ 2020].

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (†)</th>
<th>ECE (↓)</th>
<th>OOD AUROC (†)</th>
<th>OOD AUPR (†)</th>
<th>Latency (ms / example)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>96.5</td>
<td>0.023</td>
<td>0.8970</td>
<td>0.7573</td>
<td>10.42</td>
</tr>
<tr>
<td>MCD-GP DUQ</td>
<td>95.9</td>
<td>0.0146</td>
<td>0.9055</td>
<td>0.8030</td>
<td>88.38</td>
</tr>
<tr>
<td>MC Dropout Deep Ensemble</td>
<td>96.5</td>
<td>0.0210</td>
<td>0.9382</td>
<td>0.7997</td>
<td>85.62</td>
</tr>
<tr>
<td>SNGP</td>
<td>96.6</td>
<td>0.0115</td>
<td>0.9688</td>
<td>0.8802</td>
<td>17.36</td>
</tr>
</tbody>
</table>

Results on CIFAR-10 using Wide ResNet

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc (†)</th>
<th>ECE (↓)</th>
<th>Corrupted Acc/ECE</th>
<th>OOD SVHN</th>
<th>OOD CIFAR100</th>
<th>Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>96.0</td>
<td>0.023</td>
<td>72.9 / 0.153</td>
<td>0.7810</td>
<td>0.8352</td>
<td>3.91</td>
</tr>
<tr>
<td>MCD-GP DUQ</td>
<td>95.5</td>
<td>0.024</td>
<td>70.0 / 0.100</td>
<td>0.9599</td>
<td>0.8631</td>
<td>29.53</td>
</tr>
<tr>
<td>MC Dropout Deep Ensembles</td>
<td>96.0</td>
<td>0.024</td>
<td>70.0 / 0.116</td>
<td>0.9714</td>
<td>0.8320</td>
<td>27.10</td>
</tr>
<tr>
<td>SNGP (Ours)</td>
<td>95.9</td>
<td>0.018</td>
<td>74.6 / 0.090</td>
<td><strong>0.9901</strong></td>
<td><strong>0.9050</strong></td>
<td><strong>6.25</strong></td>
</tr>
</tbody>
</table>
Improving the quality of model uncertainty

Ensembles / Bayesian NNs

\[ p(y \mid x) = \frac{1}{M} \sum_{m=1}^{M} p(y \mid x, \theta_m) \]

“Composable” toolkit

1. Improve the “base” model \( p(y\mid x, \theta) \)
2. Efficiently average predictions over diverse set of functions \( \theta_1, \theta_2 \ldots \theta_M \)
   a. Diverse ensembles ↑
   b. Efficient ensembles ←
Surprisingly Simple Baseline: Deep Ensembles

Just re-run standard training but with different random seeds (initializations + SGD shuffling) & combine models.

Combine predictions of M models

Randomly Initialize & Train Net 1

Randomly Initialize & Train Net 2

....

Randomly Initialize & Train Net M

Randomly Shuffle Dataset M times

Inputs

[Lakshminarayanan+ 2016]
Deep Ensembles improve accuracy and calibration under dataset shift

Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift [Ovadia+ 2019]
Deep Ensembles: A Loss Landscape Perspective

Stanislav Fort*, Clara Huiyi Hu*, Balaji Lakshminarayanan
Motivation: Understand why deep ensembles work well

Variational methods capture local uncertainty around a mode.

Ensembles identify different modes but ignore local uncertainty and might not pick the best point from each mode.

Space of solutions

Validation

Training
Predictions similarity within and across trajectories
Subspace sampling methods

- Random
- Diagonal Gaussian
- Low-rank Gaussian
- Monte Carlo Dropout
From a bias-variance tradeoff perspective, we care about low bias as well as diversity.

We plot fraction of data points where top-1 prediction is different. Note that maximum disagreement depends on the accuracy (completely different mistakes).

Function diversity is higher across trajectories than within trajectories.
Can we combine uncertainty from subspace (within-mode) and ensembles (across-mode)?

Idea: run multiple trajectories and use subspace sampling within each trajectory (e.g. Ensemble + Diagonal Gaussian)
Best of both worlds - Results on CIFAR-10 using MediumCNN
Efficient ensembles lower inference memory and inference time

Idea: Create multiple input heads + multiple output heads (MIMO). Train each head on different SGD batch.

[Training Independent Subnetworks for Robust Prediction. Havasi+ 2020]
Better inductive biases for representations

Ensembles / Bayesian NNs

\[ p(y \mid x) = \frac{1}{M} \sum_{m=1}^{M} p(y \mid x, \theta_m) \]

1. Improve the “base” model \( p(y \mid x, \theta) \)
2. Efficiently average predictions over diverse set of functions \( \theta_1, \theta_2, \ldots, \theta_M \)
3. Better inductive biases for representations (e.g. pre-training or data augmentation)
Exploring the limits of OOD detection

Stanislav Fort*, Jie Ren*, Balaji Lakshminarayanan
Goal: Improve SOTA on hard OOD detection tasks

Far OOD, AUROC = 99%

Near OOD, AUROC = 85%
Improving near-OOD detection

CIFAR-100 $\rightarrow$ CIFAR-10

AUROC (%)

Hybrid models for open sets

85.6%
Benchmarking Human Performance (S.F. is the human)
Improving near-OOD detection

AUROC (%)

CIFAR-100 → CIFAR-10

Human: 95.9%

Hybrid models for open sets: 85.6%
Pre-trained Vision Transformers improve near-OOD detection
Pre-trained ViT improves near-OOD detection

Figure: 2D PCA projection of the space of embedding. Color coding shows Mahalanobis outlier score.
Pre-trained Vision Transformers improve near-OOD detection
Qualitative failure cases of ViT OOD detection

- Most false positives are due to mislabeling or ambiguity

(a) OOD images (CIFAR-10) closest to the in-distribution (CIFAR-100).

(b) The in-distribution (CIFAR-100) images with the closest embedding vector to images in Figure 11a.
Qualitative failure cases of ViT OOD detection

- SVHN digits are classified as CIFAR-100 “worms”

(a) OOD images (SVHN) closest to the in-distribution (CIFAR-100).

(b) The in-distribution (CIFAR-100) images with the closest embedding vector to images in Figure 11a.
The same story for genomics
(here pre-trained in a self-supervised way)

(a) BERT pre-training and fine-tuning for genomics.

<table>
<thead>
<tr>
<th>Model</th>
<th>Test Accuracy</th>
<th>Mahalanobis AUROC</th>
<th>MSP AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1D CNN [Ren et al., 2019]</td>
<td>85.93±0.11%</td>
<td>64.75±0.73%</td>
<td>65.84±0.46%</td>
</tr>
<tr>
<td>BERT pre-train and fine-tune</td>
<td>89.84±0.00%</td>
<td><strong>77.49±0.04%</strong></td>
<td>73.53±0.03%</td>
</tr>
</tbody>
</table>
Wrapping up
Uncertainty Baselines

High-quality implementations of baselines on a variety of tasks.

**Ready for use:** 7 settings, including:

- Wide ResNet 28-10 on CIFAR
- ResNet-50 and EfficientNet on ImageNet
- BERT on Clinic Intent Detection

14 different baseline methods.

Used across 10 projects at Google.

**Collaboration with OATML @ Oxford, unifying**
github.com/oatml/bdl-benchmarks.
Robustness Metrics

github.com/google-research/robustness_metrics

Lightweight modules to evaluate a model’s robustness and uncertainty predictions.

Ready for use:

- 10 OOD datasets
- Accuracy, uncertainty, and stability metrics
- Many SOTA models (TFHub support!)
- Multiple frameworks (JAX support!)

Enables large-scale studies of robustness
[Djolonga+ 2020].

Collaboration lead by Google Research, Brain Team @ Zurich.
Takeaways

● Uncertainty & robustness are critical problems in AI and machine learning.

● Best performance achieved by composing orthogonal techniques
  ○ Single model uncertainty
  ○ Ensembling multiple neural networks
  ○ Imposing inductive biases on representations

● Open questions
  ○ Understand relationship between different types of OOD shifts
  ○ What other tools do we need in the “composable” toolkit?

● Links to papers available in my webpage: http://www.gatsby.ucl.ac.uk/~balaji/
Not covered in this talk: Unsupervised anomaly detection

- Caveat: Using density $p(x)$ from deep generative model trained on $p_{\text{TRAIN}}(x)$ is not always reliable
- **Anomaly Detection using Deep Generative Models: Pitfalls and Promises** (overview talk that covers the following papers)

  - See also *A Simple Fix to Mahalanobis Distance for Improving Near-OOD Detection* where we extended this to density models of representations.
Thank you!

Dustin Tran  Jasper Snoek  Jeremiah Liu  Stanislav Fort  Clara Huiyi Hu  Jie Ren  Dan Hendrycks  Norman Mu

Kevin Murphy  Mike Dusenberry  Rodolphe Jenatton  Shreyas Padhy  Mario Lucic  Zelda Mariet  Ben Adlam  Zack Nado

Eric Nalisnick  Josip Djolonga  Jeremy Nixon  Ghassen Jerfel  Justin Gilmer  D. Sculley  Yeming Wen  Marton Havasi

& others!
References

This list is intended just as a starting point for exploring other related work using Google Scholar or Connected papers. Feel free to email me if you think there’s a reference that should be included here.

Survey papers

- A Survey of Uncertainty in Deep Neural Networks. J. Gawlikowski et al., arXiv 2107.03342.

Bayesian neural networks

- A Scalable Laplace Approximation for Neural Networks. H. Ritter, A. Botev, D. Barber ICLR 2018
References

Ensembles


Understanding marginalization


Gaussian processes and Neural Tangent Kernel

- Approximate Inference Turns Deep Networks into Gaussian Processes M. Emtiyaz Khan, Alexander Immer, Ehsan Abedi, M. Korzepa NeurIPS 2019
- Bayesian Deep Ensembles via the Neural Tangent Kernel B. He, B. Lakshminarayanan and Y.W. Teh NeurIPS 2020
- Exploring the Uncertainty Properties of Neural Networks’ Implicit Priors in the Infinite-Width Limit. B. Adlam, J. Lee, L. Xiao, J. Pennington and J. Snoek link
References

Practical guidance

- See the codebases! E.g. https://github.com/google/uncertainty-baselines

Data Augmentation and Invariances

- Using Self-Supervised Learning Can Improve Model Robustness and Uncertainty D. Hendrycks, M. Mazeika, S. Kadavath, D. Song NeurIPS 2019
- Combining Ensembles and Data Augmentation can Harm your Calibration. Y. Wen, G. Jerfel, R. Muller, M. Dusenberry, J. Snoek, B. Lakshminarayanan and D. Tran. ICLR 2021

Calibration

Proper Scoring Rules, Types of Uncertainty

- **Strictly Proper Scoring Rules, Prediction and Estimation**, Gneiting & Raftery, *JASA 2007*
- **What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?** A. Kendall, Y. Gal *NeurIPS 2017*
- **Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods** E. Hüllermeier, W. Waegeman *arXiv 1910.09457*

Deep Generative Models and Hybrid models

- **Do deep generative models know what they don’t know?** E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, B. Lakshminarayanan. *ICLR 2019.*

Detecting Out-of-Distribution Inputs

- **A Baseline for Detecting Misclassified & Out-of-Distribution Examples in Neural Networks** D. Hendrycks, K. Gimpel *ICLR 2017*
- **Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks** S. Liang, Y. Li, R. Srikant *ICLR 2018*
- **Deep Anomaly Detection with Outlier Exposure** D. Hendrycks, M. Mazeika, T. Dietterich *ICLR 2019*
- **Exploring the Limits of Out-of-Distribution Detection.** S. Fort, J. Ren, B. Lakshminarayanan *NeurIPS 2021.*