Input-gradient space particle inference for neural network ensembles

Trung Trinh[1], Markus Heinonen[1], Luigi Acerbi[2], Samuel Kaski[1,3]
[1]Aalto University, [2]Helsinki University, [3]University of Manchester
{trung.trinh, markus.o.heinonen, samuel.kaski}@aalto.fi, luigi.acerbi@helsinki.fi

Aalto-yliopisto HELSINGIN YLIOPISTO MANCHESTER 1824 The University of Manchester ICLR

Overview

TL;DR: We learn an ensemble of neural networks that is **diverse** with respect to their **input gradients**.

Repulsive deep ensembles (RDEs) [1]

Description: Train an ensemble $\{\boldsymbol{\theta}_i\}_{i=1}^{M}$ using Wasserstein gradient descent [2], which employs a kernelized repulsion term to diversify the particles to cover the Bayes posterior $p(\boldsymbol{\theta}|\mathcal{D})$

$$\boldsymbol{\theta}_i^{(t+1)} = \boldsymbol{\theta}_i^{(t)} + \eta_t \Bigg(\underbrace{\nabla_{\boldsymbol{\theta}_i^{(t)}} \log p(\boldsymbol{\theta}_i^{(t)} \,|\, \mathcal{D})}_{\text{Driving force}} - \underbrace{\frac{\sum_{j=1}^{N} \nabla_{\boldsymbol{\theta}_i^{(t)}} k(\boldsymbol{\theta}_i^{(t)}, \boldsymbol{\theta}_j^{(t)})}{\sum_{j=1}^{N} k(\boldsymbol{\theta}_i^{(t)}, \boldsymbol{\theta}_j^{(t)})}}_{\text{Repulsion force}} \Bigg)$$

- The driving force directs the particles towards high density regions of the posterior.
- The repulsion force pushes the particles away from each other to enforce diversity.

Problem: It is unclear how to define the repulsion term for neural networks:
- weight-space repulsion is ineffective due to overparameterization.
- function-space repulsion often results in underfitting.

Defining the input-gradient kernel k

Given a base kernel κ, we define the kernel in the input-gradient space for a minibatch of training samples $\mathcal{B} = \{(\mathbf{x}_b, y_b)\}_{b=1}^{B}$ as follows:

$$k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \boxed{\frac{1}{B}\sum_{b=1}^{B}} \boxed{\kappa(\nabla_{\mathbf{x}_b}\mathbf{f}(\mathbf{x}_b; \boldsymbol{\theta}_i)_{y_b}, \nabla_{\mathbf{x}_b}\mathbf{f}(\mathbf{x}_b; \boldsymbol{\theta}_j)_{y_b})}$$

Take the average over all samples in the mini-batch.

Compare the input gradients of two particles with respect to the same input.

We choose the RBF kernel on a unit sphere as the base kernel κ:

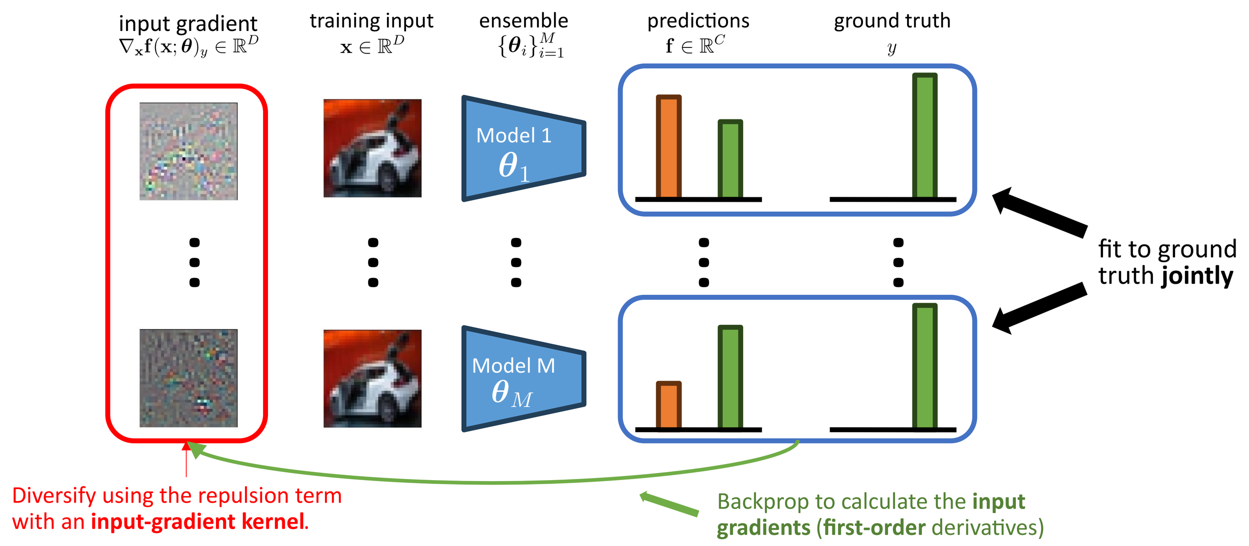
$$\kappa(\mathbf{s}_i, \mathbf{s}_j; \boldsymbol{\Sigma}) = \boxed{\exp\left(-\frac{1}{\boxed{2h}}(\mathbf{s}_i - \mathbf{s}_j)^\top \boxed{\boldsymbol{\Sigma}}^{-1}(\mathbf{s}_i - \mathbf{s}_j)\right)}, \quad \boxed{\mathbf{s}_i = \frac{\nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}; \boldsymbol{\theta}_i)_y}{||\nabla_{\mathbf{x}}\mathbf{f}(\mathbf{x}; \boldsymbol{\theta}_i)_y||_2}}$$

A scalar adaptively adjusted to prevent kernel vanishing.

Diagonal matrix containing the lengthscales.

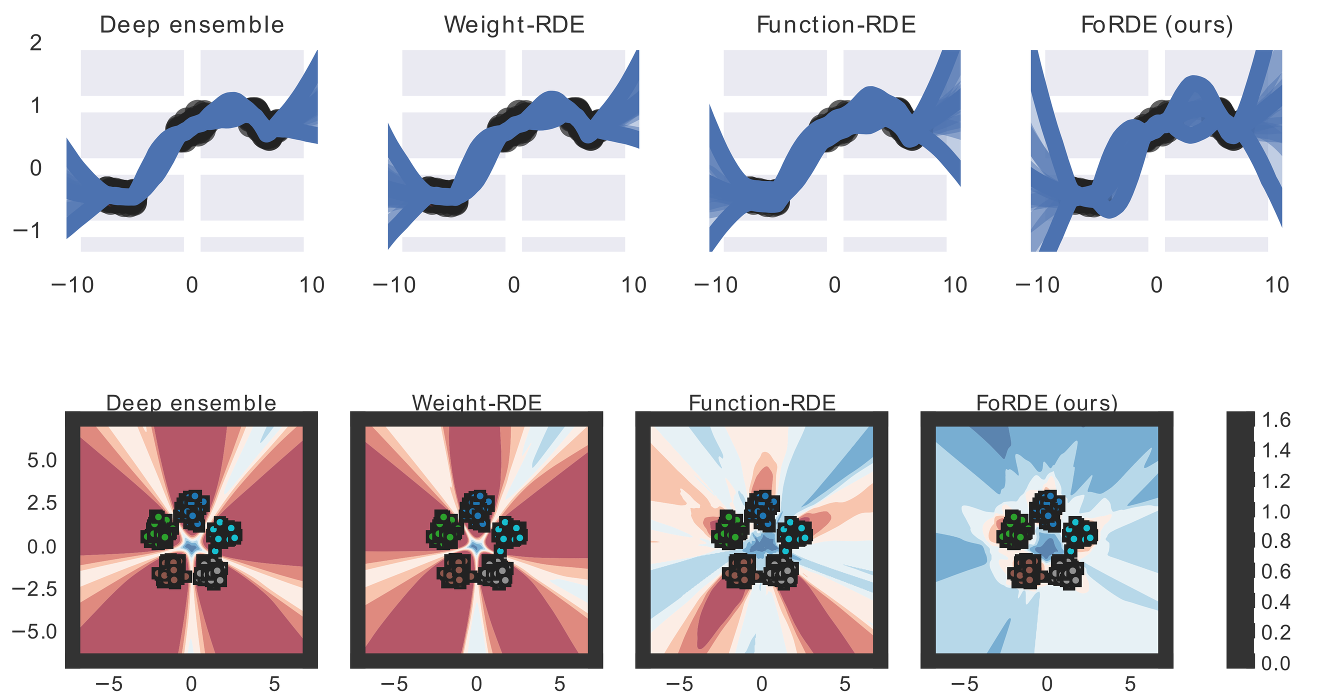
Normalize input gradients to unit vectors.

First-order Repulsive deep ensembles (FoRDEs)



Possible advantages:
- Each member is guaranteed to represent a different function;
- The issues of weight- and function-space repulsion are avoided;
- Each member is encouraged to learn different features, which can improve robustness.

Illustrative experiments



For a 1D regression task (above) and a 2D classification task (below), FoRDEs capture higher uncertainty than baselines in all regions outside of the training data. For the 2D classification task, we visualize the entropy of the predictive posteriors.

Main takeaways

1. Input-gradient-space repulsion can perform better than weight- and function-space repulsion.
2. Better corruption robustness can be achieved by configuring the repulsion kernel using the eigen-decomposition of the training data.

Benchmark comparison

Table 1: **FoRDE-PCA achieves the best performance under corruptions while FoRDE-Identity outperforms baselines on clean data. FoRDE-Tuned outperforms baselines on both clean and corrupted data.** Results of RESNET18 / CIFAR-100 averaged over 5 seeds. Each ensemble has 10 members. cA, cNLL and cECE are accuracy, NLL, and ECE on CIFAR-100-C.

Method	NLL ↓	Accuracy (%) ↑	ECE ↓	cA / cNLL / cECE
Deep ensembles	**0.70**±0.00	81.8±0.2	0.041±0.003	54.3 / 1.99 / 0.05
weight-RDE	**0.70**±0.01	81.7±0.3	0.043±0.004	54.2 / 2.01 / 0.06
function-RDE	0.76±0.02	80.1±0.4	0.042±0.005	51.9 / 2.08 / 0.07
FoRDE-PCA (ours)	0.71±0.00	81.4±0.2	0.039±0.002	**56.1 / 1.90 / 0.05**
FoRDE-Identity (ours)	**0.70**±0.00	**82.1**±0.2	0.043±0.001	54.1 / 2.02 / 0.05
FoRDE-Tuned (ours)	**0.70**±0.00	**82.1**±0.2	0.044±0.002	55.3 / 1.94 / 0.05

Table 2: **FoRDE-PCA achieves the best performance under corruptions while FoRDE-Identity has the best NLL on clean data. FoRDE-Tuned outperforms most baselines on both clean and corrupted data.** Results of RESNET18 / CIFAR-10 averaged over 5 seeds. Each ensemble has 10 members. cA, cNLL and cECE are accuracy, NLL, and ECE on CIFAR-10-C.

Method	NLL ↓	Accuracy (%) ↑	ECE ↓	cA / cNLL / cECE
Deep ensembles	0.117±0.001	96.3±0.1	0.005±0.001	78.1 / 0.78 / 0.08
weight-RDE	0.117±0.002	96.2±0.1	0.005±0.001	78.0 / 0.78 / 0.08
function-RDE	0.128±0.001	95.8±0.2	0.006±0.001	77.1 / 0.81 / 0.08
feature-RDE	0.116±0.001	96.4±0.1	**0.004**±0.001	78.1 / 0.77 / 0.08
FoRDE-PCA (ours)	0.125±0.001	96.1±0.1	0.006±0.001	**80.5 / 0.71 / 0.07**
FoRDE-Identity (ours)	**0.113**±0.002	96.3±0.1	0.005±0.001	78.0 / 0.80 / 0.08
FoRDE-Tuned (ours)	0.114±0.002	96.4±0.1	0.005±0.001	79.1 / 0.74 / 0.07

References
[1] F. D'Angelo and V. Fortuin, "Repulsive deep ensembles are Bayesian," Advances in Neural Information Processing Systems, vol. 34, pp. 3451–3465, 2021.
[2] C. Liu, J. Zhuo, P. Cheng, R. Zhang, and J. Zhu, "Understanding and Accelerating Particle-Based Variational Inference," in International Conference on Machine Learning, 2019.

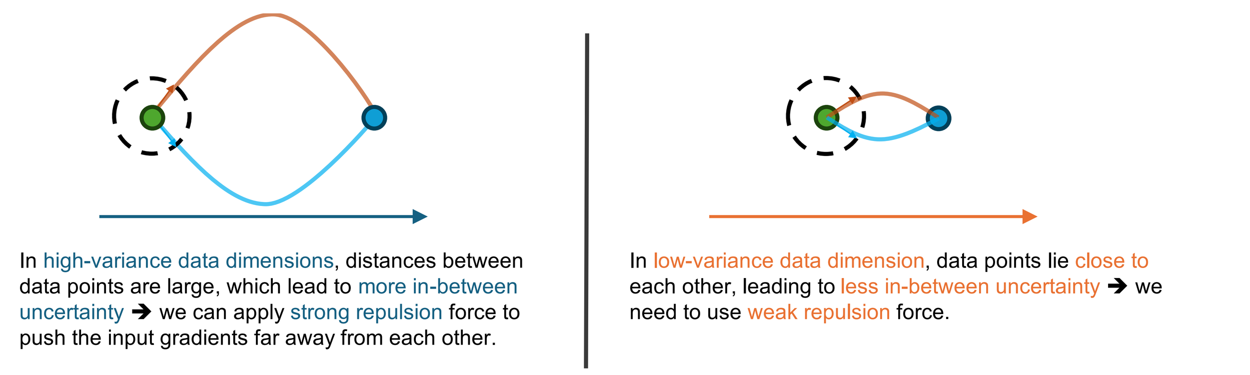
Tuning the lengthscales $\boldsymbol{\Sigma}$

Each lengthscale is inversely proportional to the strength of the repulsion force in the corresponding input dimension.

$$\underbrace{\frac{\partial}{\partial s_d}\kappa(\mathbf{s}, \mathbf{s}'; \boldsymbol{\Sigma})}_{\text{Repulsion force in the } d\text{-th dimension}} = -\frac{s_d - s'_d}{h\boldsymbol{\Sigma}_{dd}}\kappa(\mathbf{s}, \mathbf{s}'; \boldsymbol{\Sigma}) \propto \frac{1}{\boxed{\boldsymbol{\Sigma}_{dd}}}$$

lengthscale in the d–th dimension

Proposition: One should apply strong forces in high-variance dimensions (more in-between uncertainty) and weak forces in low-variance dimensions (less in-between uncertainty).

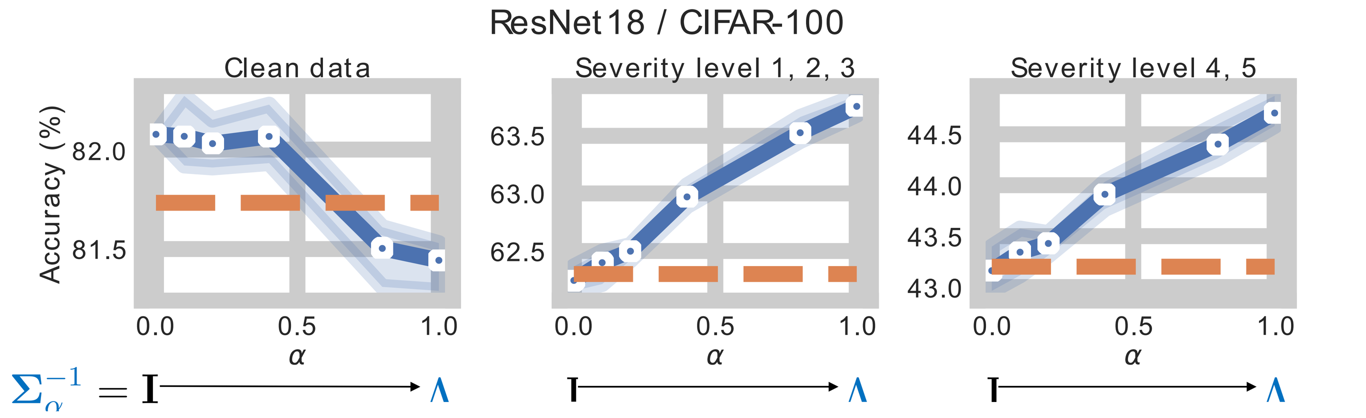


- Use PCA to get the eigenvalues and eigenvectors of the training data: $\{\mathbf{u}_d, \lambda_d\}_{d=1}^{D}$
- Define the base kernel:

$$\kappa_{\mathrm{PCA}}(\mathbf{s}, \mathbf{s}'; \boldsymbol{\Sigma}_\alpha) = \exp\left(-\frac{1}{2h}(\mathbf{U}^\top\mathbf{s} - \mathbf{U}^\top\mathbf{s}')^\top \boldsymbol{\Sigma}_\alpha^{-1}(\mathbf{U}^\top\mathbf{s} - \mathbf{U}^\top\mathbf{s}')\right)$$

  - $\mathbf{U} = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_D]$ is a matrix containing the eigenvectors as columns.
  - $\boldsymbol{\Sigma}_\alpha^{-1} = (1-\alpha)\mathbf{I} + \alpha\boldsymbol{\Lambda}$ where $\boldsymbol{\Lambda}$ is a diagonal matrix containing the eigenvalues.

Lengthscale tuning experiments



- Blue lines show accuracies of FoRDEs, while dotted orange lines show accuracies of Deep ensembles.
- When moving from the identity lengthscale $\mathbf{I}$ to the PCA lengthscales $\boldsymbol{\Lambda}$:
  - FoRDEs exhibit small performance degradations on clean images of CIFAR-100;
  - while becomes more robust against the natural corruptions of CIFAR-100-C.