An Infinite-Feature Extension for Bayesian ReLU Nets That Fixes Their Asymptotic Overconfidence

Agustinus Kristiadi, Matthias Hein, Philipp Hennig

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

MAX PLANCK INSTITUTE
FOR INTELLIGENT SYSTEMS

imprs-is

SPONSORED BY THE
Federal Ministry
of Education
and Research

A brief summary

- NNs with ReLU activation are asymptotically overconfident

A brief summary

- NNs with ReLU activation are asymptotically overconfident
- Bayesian treatment mitigates this to some degree

A brief summary

- NNs with ReLU activation are asymptotically overconfident
- Bayesian treatment mitigates this to some degree
- But previous result:
  - only shows this for binary classification and
  - doesn't achieve the ideal maximum uncertainty even in the limit

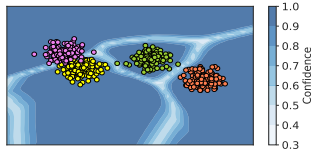
A brief summary

- NNs with ReLU activation are asymptotically overconfident
- Bayesian treatment mitigates this to some degree
- But previous result:
  - only shows this for binary classification and
  - doesn't achieve the ideal maximum uncertainty even in the limit
- We:
  - propose an extension to *multi-class* ReLU BNNs,
  - which is guaranteed to be maximally uncertain far from the data

A brief summary

- NNs with ReLU activation are asymptotically overconfident
- Bayesian treatment mitigates this to some degree
- But previous result:
  - only shows this for binary classification and
  - doesn't achieve the ideal maximum uncertainty even in the limit
- We:
  - propose an extension to *multi-class* ReLU BNNs,
  - which is guaranteed to be maximally uncertain far from the data
- Moreover, the method is beneficial for the *non-asymptotic* uncertainty, without affecting the prediction

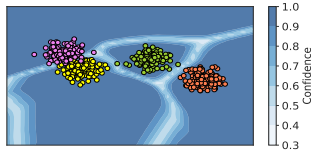
Problem with ReLU nets



- Point-estimation of a ReLU net f_θ is almost always *overconfident*[1]

[1] Hein et al., CVPR 2019

Problem with ReLU nets



- Point-estimation of a ReLU net f_θ is almost always *overconfident*[1]
- Because
  - Far from the data, $f_{\theta_{\text{MAP}}}$ is linear[2]
  - So, given an input x and a scalar $\alpha > 0$, there is a class c s.t. $\lim_{\alpha \to \infty} \text{softmax}\left(f_{\theta_{\text{MAP}}}(\alpha x)\right)_c = 1$

[1] Hein et al., CVPR 2019

[2] And non-constant

Bayesian ReLU nets

- Approximate instead the posterior $p(\theta \mid \mathcal{D}) \approx \mathcal{N}(\mu, \Sigma) =: q(\theta)$

Bayesian ReLU nets

- Approximate instead the posterior $p(\theta \mid \mathcal{D}) \approx \mathcal{N}(\mu, \Sigma) =: q(\theta)$
- Kristiadi et al. (ICML 2020) shows that[1]

$$\lim_{\alpha \to \infty} p(y = 1 \mid \alpha x, \mathcal{D}) \approx \lim_{\alpha \to \infty} \int \sigma\left(f_\theta(\alpha x)\right)\ q(\theta)\, d\theta < 1,$$

  and the exact value depends on the eigenvalues of Σ

[1] σ is the logistic-sigmoid function

Bayesian ReLU nets

- Approximate instead the posterior $p(\theta \mid \mathcal{D}) \approx \mathcal{N}(\mu, \Sigma) =: q(\theta)$
- Kristiadi et al. (ICML 2020) shows that[1]

$$\lim_{\alpha \to \infty} p(y = 1 \mid \alpha x, \mathcal{D}) \approx \lim_{\alpha \to \infty} \int \sigma\left(f_\theta(\alpha x)\right)\ q(\theta)\, d\theta < 1,$$

  and the exact value depends on the eigenvalues of Σ
- **But:**
  - This result is limited to binary classification
  - This bound can be loose—ideally we have the *uniform confidence* $1/C$.[2]

[1] σ is the logistic-sigmoid function

[2] Where C is the number of classes.

Achieving the uniform confidence

- Previous result uses[1]

$$p(y = 1 \mid \alpha x, \mathcal{D}) \approx \sigma\left(\frac{\alpha \mathbb{E}_q(f_\theta(x))}{\sqrt{1 + \pi/8 \operatorname{Var}_q(f_\theta(\alpha x))}}\right)$$

[1] Via linearization and the probit approximation

Achieving the uniform confidence

- Previous result uses[1]

$$p(y = 1 \mid \alpha x, \mathcal{D}) \approx \sigma\left(\frac{\alpha \mathbb{E}_q(f_\theta(x))}{\sqrt{1 + \pi/8 \operatorname{Var}_q(f_\theta(\alpha x))}}\right)$$

- No uniform confidence because:
  - The numerator is linear in α
  - The denominator is also linear in α

[1] Via linearization and the probit approximation

Achieving the uniform confidence

- Previous result uses[1]

$$p(y = 1 \mid \alpha x, \mathcal{D}) \approx \sigma\left(\frac{\alpha \mathbb{E}_q(f_\theta(x))}{\sqrt{1 + \pi/8 \operatorname{Var}_q(f_\theta(\alpha x))}}\right)$$

- No uniform confidence because:
  - The numerator is linear in α
  - The denominator is also linear in α
- So, to achieve the uniform confidence:
  - The numerator must stay the same
  - The variance must be in $\Theta(\alpha^d)$ for $d > 2$

[1] Via linearization and the probit approximation

Super-quadratic variance does exist

Just not for BNNs

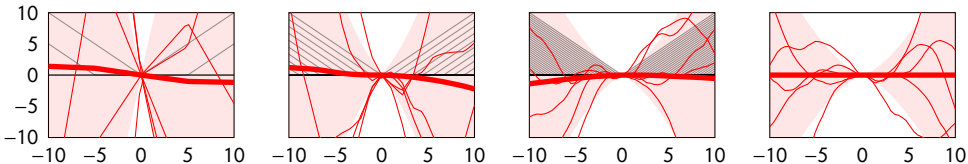
- In GPs, the *cubic spline kernel*[1] has cubic variance
  - "Placing" K ReLU features on the input space and take $K \to \infty$
  - But it is defined *only* on $\mathbb{R}_{>=0}$

[1] Wahba, 1990

Super-quadratic variance does exist

Just not for BNNs

- In GPs, the *cubic spline kernel*[1] has cubic variance
  - "Placing" K ReLU features on the input space and take $K \to \infty$
  - But it is defined *only* on $\mathbb{R}_{>=0}$
- We extend this kernel (***Double-Sided Cubic Spline (DSCS) kernel***)[2]



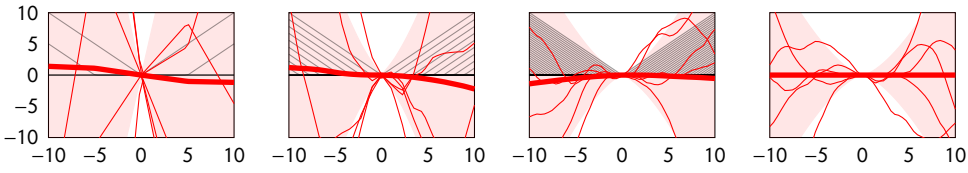
[1] Wahba, 1990

[2] For multi-dimensional inputs, average the kernel value of each component

Super-quadratic variance does exist

Just not for BNNs

- In GPs, the *cubic spline kernel*[1] has cubic variance
  - "Placing" K ReLU features on the input space and take $K \to \infty$
  - But it is defined *only* on $\mathbb{R}_{>=0}$
- We extend this kernel (***Double-Sided Cubic Spline (DSCS) kernel***)[2]



(a) $K = 6$ (b) $K = 20$ (c) $K = 60$ (d) $K = \infty$

- **Important properties:** $k(x, x) \approx 0$ for $x \approx 0$ and $k(\alpha x, \alpha x) \in \Theta(\alpha^3)$

[1] Wahba, 1990

[2] For multi-dimensional inputs, average the kernel value of each component

Extending ReLU BNNs

With the DSCS kernel

- The ReLU BNN $f_\theta(x)$ doesn't achieve the uniform confidence
  $\implies$ there is a *residual* predictive uncertainty

Extending ReLU BNNs

With the DSCS kernel

- The ReLU BNN $f_\theta(x)$ doesn't achieve the uniform confidence
  $\implies$ there is a *residual* predictive uncertainty
- Model this residual with the DSCS kernel k, call it ***ReLU-GP Residual (RGPR)***:

$$\widetilde{f}(x) := f_\theta(x) + \widehat{f}(x) \qquad \text{where} \qquad \widehat{f} \sim \mathcal{GP}(0, k)$$

Extending ReLU BNNs

With the DSCS kernel

- The ReLU BNN $f_\theta(x)$ doesn't achieve the uniform confidence
  $\implies$ there is a *residual* predictive uncertainty
- Model this residual with the DSCS kernel k, call it ***ReLU-GP Residual (RGPR)***:

$$\widetilde{f}(x) := f_\theta(x) + \widehat{f}(x) \qquad \text{where} \qquad \widehat{f} \sim \mathcal{GP}(0, k)$$

- Under posterior $q(\theta) = \mathcal{N}(\mu, \Sigma)$, we show that the *GP posterior* of $\widetilde{f}(x)$ is[1]

$$p(\widetilde{f}(x) \mid \mathcal{D}) \approx \mathcal{N}\left(\mathbb{E}_q(f_\theta(x)), \mathrm{Var}_q(f_\theta(x)) + k(x, x)\right)$$

[1] Under linearization and $k(x, x) \approx 0$ for $x \approx 0$

Extending ReLU BNNs

With the DSCS kernel

- The ReLU BNN $f_\theta(x)$ doesn't achieve the uniform confidence
  $\implies$ there is a *residual* predictive uncertainty
- Model this residual with the DSCS kernel k, call it ***ReLU-GP Residual (RGPR)***:

$$\widetilde{f}(x) := f_\theta(x) + \widehat{f}(x) \qquad \text{where} \qquad \widehat{f} \sim \mathcal{GP}(0, k)$$

- Under posterior $q(\theta) = \mathcal{N}(\mu, \Sigma)$, we show that the *GP posterior* of $\widetilde{f}(x)$ is[1]

$$p(\widetilde{f}(x) \mid \mathcal{D}) \approx \mathcal{N}\left(\mathbb{E}_q(f_\theta(x)), \mathrm{Var}_q(f_\theta(x)) + k(x, x)\right)$$

Thus, RGPR can be applied *post-hoc* cheaply, given a *pre-trained* ReLU BNN

[1] Under linearization and $k(x, x) \approx 0$ for $x \approx 0$

RGPR fixes asymptotic overconfidence

In multi-class classification

- Recall $p(\widetilde{f}(x) \mid \mathcal{D}) \approx \mathcal{N}\left(\mathbb{E}_q(f_\theta(x)), \mathrm{Var}_q(f_\theta(x)) + k(x,x)\right)$
  - Since $\mathbb{E}(\widetilde{f}(x)) = \mathbb{E}_q(f_\theta(x))$ we retain the BNN's prediction
  - But $\mathrm{Var}(\widetilde{f}(x)) = \mathrm{Var}_q(f_\theta(x)) + k(x,x) \in \Theta(\alpha^3)$!

RGPR fixes asymptotic overconfidence

In multi-class classification

- Recall $p(\widetilde{f}(x) \mid \mathcal{D}) \approx \mathcal{N}\left(\mathbb{E}_q(f_\theta(x)), \mathrm{Var}_q(f_\theta(x)) + k(x,x)\right)$
  - Since $\mathbb{E}(\widetilde{f}(x)) = \mathbb{E}_q(f_\theta(x))$ we retain the BNN's prediction
  - But $\mathrm{Var}(\widetilde{f}(x)) = \mathrm{Var}_q(f_\theta(x)) + k(x,x) \in \Theta(\alpha^3)$!
- It induces the predictive distribution:

$$p(y = c \mid x, \widetilde{f}, \mathcal{D}) = \int \mathrm{softmax}(\widetilde{f}(x))_c \, p(\widetilde{f}(x) \mid \mathcal{D}) \, d\widetilde{f}(x)$$

RGPR fixes asymptotic overconfidence

In multi-class classification

- Recall $p(\widetilde{f}(x) \mid \mathcal{D}) \approx \mathcal{N}\left(\mathbb{E}_q(f_\theta(x)), \mathrm{Var}_q(f_\theta(x)) + k(x,x)\right)$
  - Since $\mathbb{E}(\widetilde{f}(x)) = \mathbb{E}_q(f_\theta(x))$ we retain the BNN's prediction
  - But $\mathrm{Var}(\widetilde{f}(x)) = \mathrm{Var}_q(f_\theta(x)) + k(x,x) \in \Theta(\alpha^3)$!
- It induces the predictive distribution:

$$p(y = c \mid x, \widetilde{f}, \mathcal{D}) = \int \mathrm{softmax}(\widetilde{f}(x))_c \, p(\widetilde{f}(x) \mid \mathcal{D}) \, d\widetilde{f}(x)$$

- Under some approximations,[1] we can thus show:

$$\lim_{\alpha \to \infty} p(y = c \mid \alpha x, \widetilde{f}, \mathcal{D}) = \frac{1}{C} \qquad \text{for each class } c = 1, \ldots, C$$

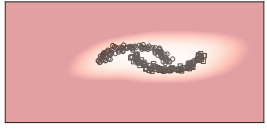
[1] Linearization and the generalized probit approximation (Gibbs, 1997)

Extending RGPR to *non-asymptotic* regimes

- Apply RGPR to not only the input, but also hidden layer $\{h_l(x)\}_{l=1}^{L-1}$ of the (point-estimated) NN[1]

$$p(\widehat{f}(x)) = \mathcal{N}\left(0, \sum_{l=0}^{L-1} \sigma_l^2 k(h_l(x), h_l(x))\right)$$

(a) Input only



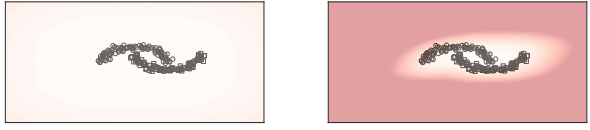
(b) Input & hiddens

[1] We assume that the GP for each hidden space is independent

Extending RGPR to *non-asymptotic* regimes

- Apply RGPR to not only the input, but also hidden layer $\{h_l(x)\}_{l=1}^{L-1}$ of the (point-estimated) NN[1]

$$p(\widehat{f}(x)) = \mathcal{N}\left(0, \sum_{l=0}^{L-1} \sigma_l^2 k(h_l(x), h_l(x))\right)$$



(a) Input only (b) Input & hiddens

- One can also optimize the hyperparameters $\{\sigma_l^2\}_{l=0}^{L-1}$

[1] We assume that the GP for each hidden space is independent

OOD detection

Table: OOD data detection in terms of FPR@95—lower is better. All values are in percent and averages over five OOD test sets and over 5 prediction runs. Prediction is done via Monte Carlo integration.

Methods	**MNIST**	**CIFAR10**	**SVHN**	**CIFAR100**
MAP	28.2	38.9	17.8	72.2
Temp. Scaling	28.4	34.9	17.6	71.9
Deep Ens.	23.0	51.0	11.3	74.7
Vanilla GP + DSCS	27.8	46.7	19.1	69.1
Last-layer Laplace	24.8	29.8	15.7	69.5
Last-layer Laplace + RGPR	**3.6**	**24.2**	**9.6**	**63.0**

Conclusion

- We extend the “ReLU kernel” of Wahba (1990) to the whole $\mathbb{R}$

Conclusion

- We extend the “ReLU kernel” of Wahba (1990) to the whole $\mathbb{R}$
- Using this kernel, we model the “uncertainty residual” of ReLU BNNs

Conclusion

- We extend the “ReLU kernel” of Wahba (1990) to the whole $\mathbb{R}$
- Using this kernel, we model the “uncertainty residual” of ReLU BNNs
- The resulting method, RGPR, can be applied *post-hoc* to ReLU BNNs

Conclusion

- We extend the “ReLU kernel” of Wahba (1990) to the whole $\mathbb{R}$
- Using this kernel, we model the “uncertainty residual” of ReLU BNNs
- The resulting method, RGPR, can be applied *post-hoc* to ReLU BNNs
- **Theoretically:** RGPR fixes the asymptotic overconfidence problem

Conclusion

- We extend the “ReLU kernel” of Wahba (1990) to the whole $\mathbb{R}$
- Using this kernel, we model the “uncertainty residual” of ReLU BNNs
- The resulting method, RGPR, can be applied *post-hoc* to ReLU BNNs
- **Theoretically:** RGPR fixes the asymptotic overconfidence problem
- **Practically:** RGPR can be extended to non-asymptotic regimes and gives good uncertainty quantification performance