# Gaussian Process Framework for Deep Neural Networks

Xiang Fu, Shengyuan Hu, Shangdi Yu

# Motivation

Find a GP representation for CNN

- poor uncertainty estimates
- many parameters

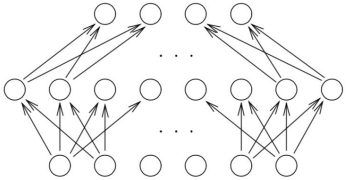
GP - posterior uncertainty, few parameters

# Background

- Give Gaussian priors to network parameter
- Get GP!

# Bayesian Neural Networks

- random NN with one hidden layer => GP
  (under certain conditions)
- e.g. zero-mean Gaussian priors for weights and biases



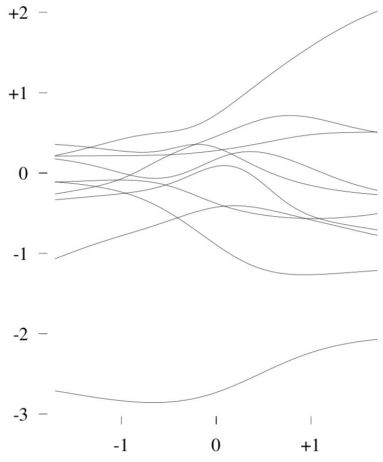

$$f_k(x) = b_k + \sum_{j=1}^{H} v_{jk} h_j(x)$$

$$h_j(x) = \tanh\left(a_j + \sum_{i=1}^{I} u_{ij} x_i\right)$$

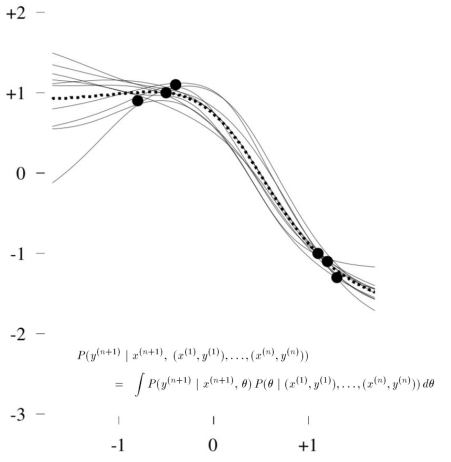
Zero mean Gaussian prior with $\sigma_b$, $\sigma_v$, $\sigma_a$, and $\sigma_u$

**=> GP prior for $f_k$**

**Prior**



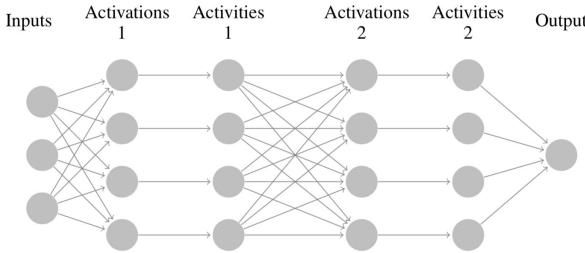

**Posterior**




$$P(y^{(n+1)} \mid x^{(n+1)}, (x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}))$$
$$= \int P(y^{(n+1)} \mid x^{(n+1)}, \theta)\, P(\theta \mid (x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)}))\, d\theta$$

# What if we have more than one layer?

GP



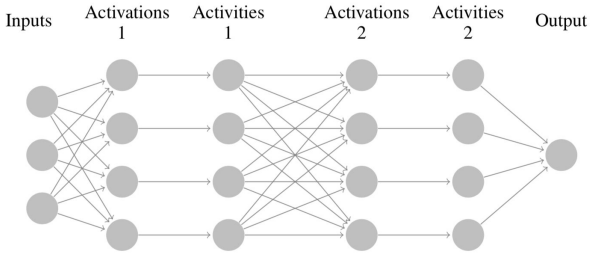

random wide **fully connected** feedforward networks **(1+ hidden layer)**

**Finite** deep networks => GP (Theorem 1)

A. G. d. G. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani, “Gaussian process behaviour in wide deep neural networks,” in International Conference on Learning Representations, 2018. [Online]. Available: https://openreview.net/forum?id=H1-nGgWC-

# What if we have more than one layer?

GP




$$f_i^{(1)}(x) = \sum_{j=1}^{M} w_{i,j}^{(1)} x_j + b_i^{(1)} .$$

$$g_i^{(\mu)}(x) = \phi(f_i^{(\mu)}(x)) \quad \text{(ReLU)}$$

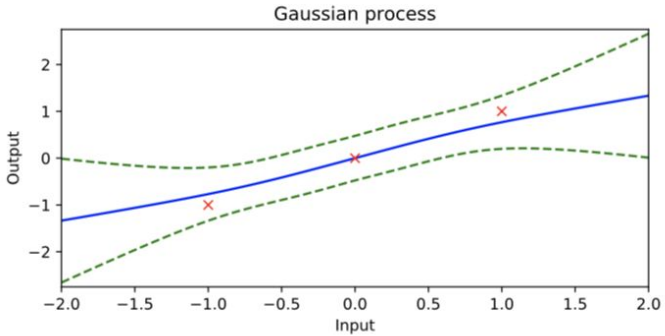
$$f_i^{(\mu+1)}(x) = \sum_{j=1}^{H_\mu} w_{i,j}^{(\mu+1)} g_j^{(\mu)}(x) + b_i^{(\mu+1)} ,$$

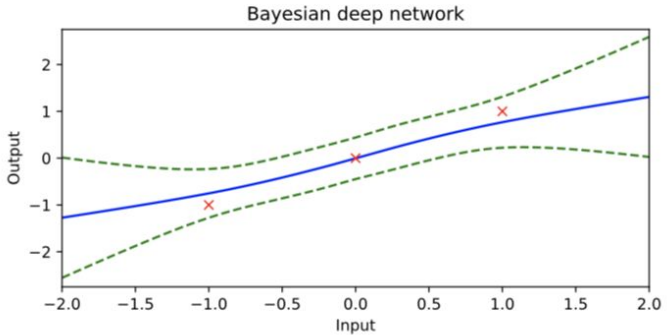
random wide **fully connected** feedforward networks **(1+ hidden layer)**

**Theorem 1.** *Consider a Bayesian deep neural network of the form in Equations (1) and (2) using ReLU activation functions. Then there exist strictly increasing width functions* $h_\mu : \mathbb{N} \mapsto \mathbb{N}$ *such that* $H_1 = h_1(n), \ldots, H_D = h_D(n)$*, and for any countable input set* $(x[i])_{i=1}^{\infty}$*, the distribution of the output of the network converges in distribution to a Gaussian process as* $n \to \infty$.

A. G. d. G. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani, “Gaussian process behaviour in wide deep neural networks,” in International Conference on Learning Representations, 2018. [Online]. Available: https://openreview.net/forum?id=H1-nGgWC-

# What if we have more than one layer?




**Finite** deep networks => GP

Bayesian deep networks from the literature can exhibit predictions that are close to Gaussian processes

A. G. d. G. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani, “Gaussian process behaviour in wide deep neural networks,” in International Conference on Learning Representations, 2018. [Online]. Available: https://openreview.net/forum?id=H1-nGgWC-

# Two Line Proof *Sketch*

- Berry-Esseen inequality: upper bound how far each layer is from a multivariate normal distribution

- Inductively propagate these inequalities through the network

# These fully-connected networks are rarely used in image classification tasks

# Whether state-of-the-art architectures such as CNNs and ResNets have equivalent GP representations?

A. G. d. G. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani, “Gaussian process behaviour in wide deep neural networks,” in International Conference on Learning Representations, 2018. [Online]. Available: https://openreview.net/forum?id=H1-nGgWC-

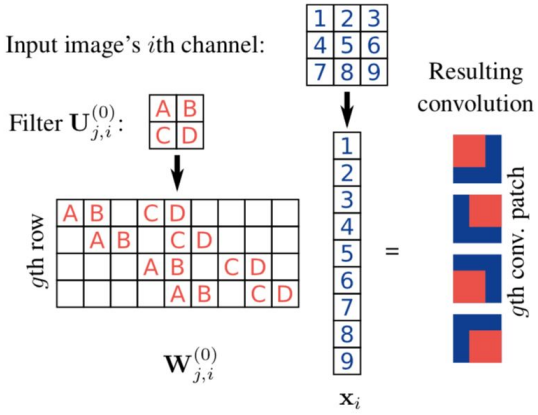
# Output of a (residual) CNN is a GP

- In the limit of infinitely many convolutional filters, with an appropriate prior over the weights and biases
- Very few parameters (only the hyperparameters of the original CNN)

Garriga-Alonso, A., Aitchison, L., and Rasmussen, C. E. Deep Convolutional Networks as shallow Gaussian Processes.
ArXiv e-prints, August 2018.

# GP Behaviour in A CNN

- Input image **X** of dimension $C^{(0)} \times H^{(0)} \times D^{(0)}$, flatten to $C^{(0)} \times (H^{(0)}\, D^{(0)})$
- Convolutional Filter $U^{(l)}_{j,i}$, transformed to $W^{(l)}_{j,i}$
- Bias $b^{(l)}_{j}$
- Make $U^{(l)}_{j,i}$ $b^{(l)}_{j}$ independent Gaussian RVs




$$U^{(\ell)}_{j,i,x,y} \sim \mathcal{N}\left(0, \sigma_w^2 / C^{(\ell)}\right)$$

$$b^{(\ell)}_{j} \sim \mathcal{N}\left(0, \sigma_b^2\right)$$

activation with non-linear function $\phi$

$$\mathbf{a}^{(\ell+1)}_{j}(\mathbf{X}) := b^{(\ell)}_{j} + \sum_{i=1}^{C^{(\ell)}} \mathbf{W}^{(\ell)}_{j,i}\, \phi\left(\mathbf{a}^{(\ell)}_{i}(\mathbf{X})\right)$$

# GP Behaviour in A CNN

- **Infinite Channels**
- Shared convolutional layers, makes covariance intractable
- only need the variance of the output
- covariances independent of the output channel j
- Iteratively calculate k(**X**, **X'**)

Activations:

$$A_{j,g}^{(\ell+1)}(\mathbf{X}) = b_j^{(\ell)} + \sum_{i=1}^{C^{(\ell)}} \sum_{h=1}^{H^{(\ell)}D^{(\ell)}} W_{j,i,g,h}^{(\ell)} \, \phi(A_{i,h}^{(\ell)}(\mathbf{X}))$$

Covariances:

$$v_g^{(\ell+1)}(\mathbf{X}, \mathbf{X}') = \mathbb{C}\left[A_{j,g}^{(\ell+1)}(\mathbf{X}), A_{j,g}^{(\ell+1)}(\mathbf{X}')\right] = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 \sum_{h \in g\text{th patch}} s_h^{(\ell)}(\mathbf{X}, \mathbf{X}'),$$

where

$$s_h^{(\ell)}(\mathbf{X}, \mathbf{X}') = \mathbb{E}\left[\phi(A_{i,h}^{(\ell)}(\mathbf{X}))\phi(A_{i,h}^{(\ell)}(\mathbf{X}'))\right]$$

With ϕ being ReLU:

$$s_g^{(\ell)}(\mathbf{X}, \mathbf{X}') = \frac{\sqrt{v_g^{(\ell)}(\mathbf{X}, \mathbf{X}) v_g^{(\ell)}(\mathbf{X}', \mathbf{X}')}}{\pi} \left(\sin\theta_g^{(\ell)} + (\pi - \theta_g^{(\ell)})\cos\theta_g^{(\ell)}\right)$$

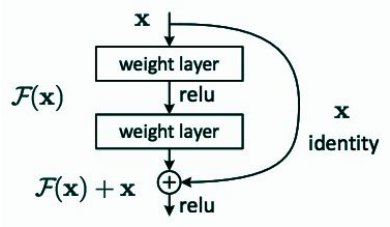
$$\text{where } \theta_g^{(\ell)} = \cos^{-1}\left(v_g^{(\ell)}(\mathbf{X}, \mathbf{X}') / \sqrt{v_g^{(\ell)}(\mathbf{X}, \mathbf{X}) v_g^{(\ell)}(\mathbf{X}', \mathbf{X}')}\right).$$

# Our Work

# Concatenation Operations and Dense Blocks

- Skip-Connection (ResNets): Addition of the identity function
- Dense-Connection (DenseNets): concatenation of feature-maps

## Addition (ResNets)

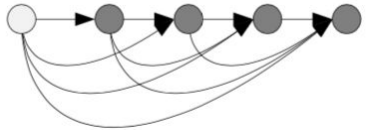



$$x_\ell = H_\ell(x_{\ell-1}) + x_{\ell-s}$$

$$a_j^{(\ell+1)}(X) = a_j^{(\ell-s)}(X) + b_j^{(\ell)} + \sum_{i=1}^{C^{(\ell)}} W_{j,i}^{(\ell)} \phi(a_i^{(\ell)}(X))$$

## Concatenation (DenseNets)

$$x_\ell = H_\ell([x_0, x_1, ..., x_{\ell-1}])$$



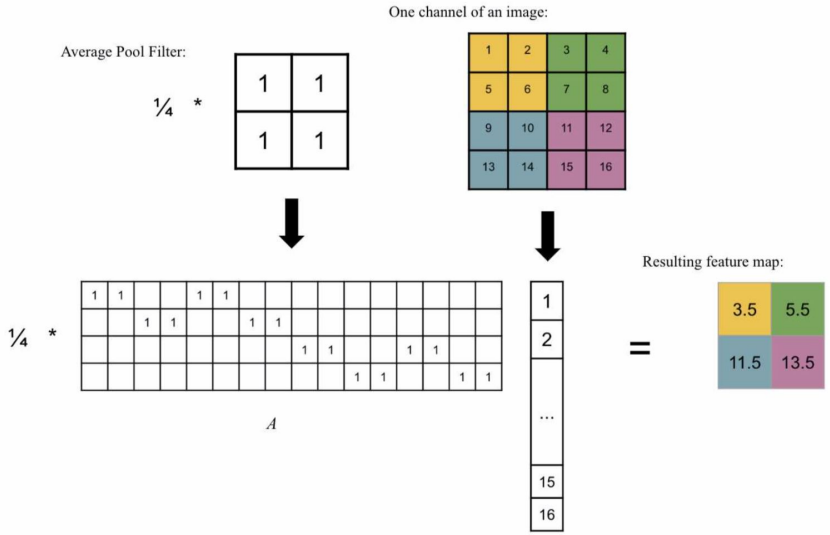
$$a_j^{(\ell+1)}(X) = b_j^{(\ell)} + \sum_{y=0}^{\ell} \sum_{i=1}^{C^{(y)}} W_{j,i}^{(y)} \phi(a_i^{(y)}(X))$$

Graph from : https://blog.waya.ai/deep-residual-learning-9610bb62c355

# Average Pooling

Average Pooling could be viewed as matrix multiplication!

Simply left multiply a constant on the weight matrix produces average pooling.




$$a_j^{(\ell+1)}(X) = b_j^{(\ell)} + \sum_{i=1}^{C^{(\ell)}} AW_{j,i}^{(\ell)} \phi(a_i^{(\ell)}(X))$$

# Experiments

**MNIST**

*Table 1.* GP Performance on different architecture

| SETTING | BASELINE | WITH AVG.POOL |
|---|---|---|
| 3 CONV LAYERS | **96.2** | **95.87** |
| 6 CONV LAYERS | 95.87 | 95.58 |
| 9 CONV LAYERS | 95.82 | 95.48 |
| 12 CONV LAYERS | 95.72 | 95.36 |
| 15 CONV LAYERS | 95.48 | 95.18 |
| 18 CONV LAYERS | 95.11 | 94.85 |

Accuracy inverse proportional to number of layers.

# Conclusions

Vanilla Convolutional Neural Network performs better than that with Average Pooling

Shallow Neural Network Kernel performs better than Deep Neural network Kernel

# What next?

- ➔ Scalable methods implementation in GPyTorch
- ➔ Experiments on larger datasets (CIFAR-10/100) and more complex models (DenseNet GP)
- ➔ Verify whether the relation between accuracy and number of layers on more complex and uncertain classification tasks
- ➔ Investigate the effects of hyperparameters

# Questions?

# References

Garriga-Alonso, A., Aitchison, L., and Rasmussen, C. E. Deep Convolutional Networks as shallow Gaussian Processes. ArXiv e-prints, August 2018.

Radford M. Neal. Bayesian Learning for Neural Networks. Springer, 1996.

A. G. d. G. Matthews, J. Hron, M. Rowland, R. E. Turner, and Z. Ghahramani, “Gaussian process behaviour in wide deep neural networks,” in International Conference on Learning Representations, 2018. [Online]. Available: https://openreview.net/forum?id=H1-nGgWC-