Bi-fidelity Training of Neural Networks

Subhayan De and Alireza Doostan

Aerospace Mechanics Research Center (AMReC)
Smead Department of Aerospace Engineering Sciences
University of Colorado Boulder

University of Colorado
Boulder

In collaboration with
Malik Hassanaly, Matthew Reynolds, and Ryan N. King
National Renewable Energy Laboratory

Acknowledgments: DOE's ASCR program and AFOSR

6th Annual Sandia Machine Learning and Deep Learning (MLDL) Workshop
26th July, 2022

Neural Network Surrogates

- Models, often in the form of partial differential equations, are used to describe the behavior of a physical system.

$$\text{Input, } x \;\Rightarrow\; \boxed{\begin{array}{c}\text{Model}\\ \mathcal{M}(x)\end{array}} \;\Rightarrow\; \text{QoI, } y$$

- The recent advances in computing power have enabled the training of large neural networks to learn this relationship.

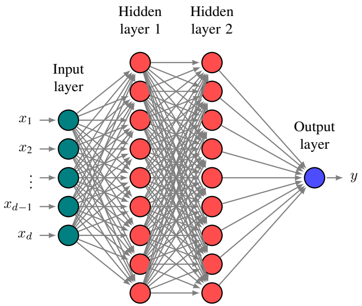
Neural Network Surrogates

Opportunities

- High degree of expressiveness of neural networks.
- Abundance of software tools.
- Training via low-order optimization schemes.

Challenges

- Often require a large amount of training data.
- Lots of parameters to tune.



Feed-forward Neural Network (FNN)

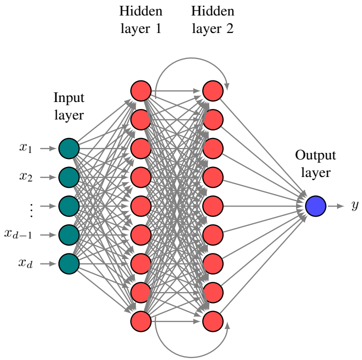
Neural Network Surrogates

Opportunities

- High degree of expressiveness of neural networks.
- Abundance of software tools.
- Training via low-order optimization schemes.

Challenges

- Often require a large amount of training data.
- Lots of parameters to tune.



Residual Neural Network (ResNet)

Bi-fidelity Models

- Multiple models are often available to describe a physical system.
- **High-fidelity models** require more computational resources but provide higher levels of accuracy. These models can be used to generate accurate data for training neural networks, but it is computationally expensive.
- Similarly, generating large experimental dataset can also be difficult.
- **Low-fidelity models** require small computational budget but generally provide lower levels of accuracy. They can be used to generate a large training dataset.

Question

- Can we efficiently train neural networks using a small dataset from the high-fidelity model but utilizing a large dataset from the low-fidelity model?

- To answer this question, we will look into
  - the use of transfer learning and
  - the use of ℓ_1-regularization.

Outline

1. Bi-fidelity Transfer Learning

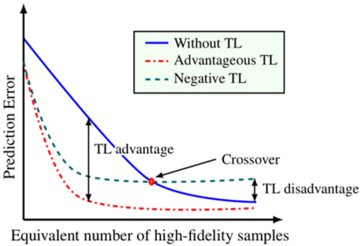
2. Bi-fidelity ℓ_1-regularization

Transfer Learning of Neural Networks

- Transfer learning (TL) uses the knowledge gained from solving one problem to solve another related problem, where accurate or labeled data may be missing or limited.
- Transfer learning can be used using a combination of low- and high-fidelity datasets.

Advantages:

- smaller initial training error,
- faster convergence of the optimization scheme, and
- similar (or smaller) validation error using smaller datasets.

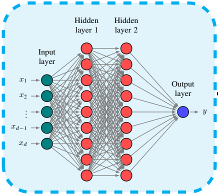


Pan and Yang (2009).

Bi-fidelity Weighted Learning (BFWL)

- A two-stage training procedure.
- The neural network is retrained using a modified step size that incorporates uncertainty information from the teacher $\mathcal{GP}$.

Student



LF Data $\{\mathbf{x}_i, y_l(\mathbf{x}_i)\}_{i=1}^{N_l}$

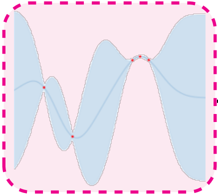
Network parameters $\boldsymbol{\theta}$

Update $\boldsymbol{\theta}$ using soft data

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \widehat{\eta}_k \frac{\widehat{\mathbf{m}}_k}{\sqrt{\mathbf{v}_k + \epsilon}}$$

$$\widehat{\eta}_k = \eta_k \exp\left[-\beta\Sigma(\mathbf{x}_{b,i_k})\right]$$

Teacher $\mathcal{GP}$



HF Data $\{\mathbf{x}_i, y_h(\mathbf{x}_i)\}_{i=1}^{N_h}$

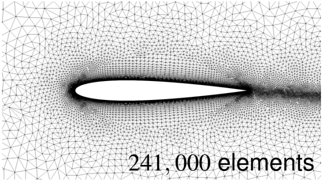
Soft Data $\{(\mathbf{x}_{b,i}, \mu(\mathbf{x}_{b,i}))\}_{i=1}^{N_l+N_h}$

De, Subhayan, et al. "On transfer learning of neural networks using bi-fidelity data for uncertainty propagation." *International Journal for Uncertainty Quantification* 10.6 (2020).

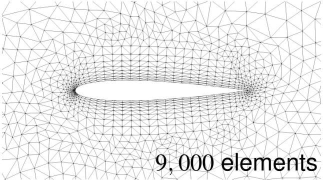
Example: Flow around a NACA Airfoil

- Flow around NACA 4412 airfoil at $Re = 1.52 \times 10^6$ and low angle-of-attack (AoA).
- **QoI:** the coefficient of pressure C_p at the surface of the airfoil.
- Four parameters of the airfoil are uncertain – maximum camber, location of maximum camber, maximum thickness, and AoA.

- 100 high-fidelity data from:



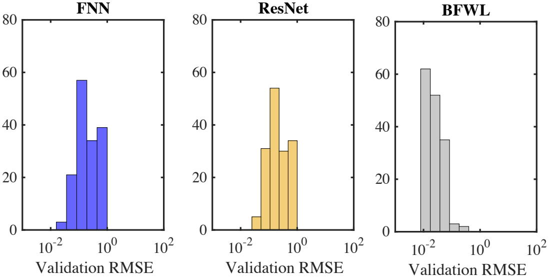
- 200 low-fidelity data from:



- The computational cost of the low-fidelity model is 498 times smaller than that of the high-fidelity one.
- Neural Network: 4 hidden layers @50 neurons each.

Example: Flow around a NACA Airfoil

- Histograms for 153 grid points along the surface of the airfoil.



Strategy	Mean validation error
FNN	3.01×10^{-1}
ResNet	2.92×10^{-1}
BFWL (FNN)	3.24×10^{-2}

- Bi-fidelity weighted learning can reduce the error by an order of magnitude.

Outline

1. Bi-fidelity Transfer Learning

2. Bi-fidelity ℓ_1-regularization

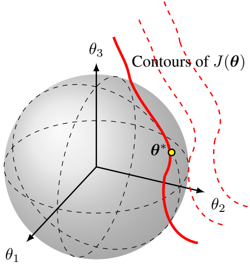
Regularization Techniques

- A regularization term is often added to the objective of the optimization to prevent overfitting.

- The most common is to use a Tikhonov regularization or standard ℓ_2-regularization.

$$\arg\min_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) := J(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_2$$

Strength of the regularization



Standard ℓ_2-regularization

- The ℓ_2-regularization term defines the surface of a hypersphere.

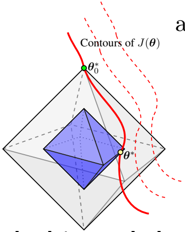
Regularization Techniques

- To induce sparsity in the parameters $\boldsymbol{\theta}$, the following optimization is used

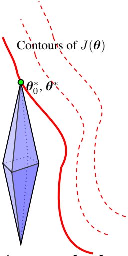
$$\arg\min_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) := J(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_1$$

- Bayesian interpretation: a maximum a posteriori (MAP) estimate of the parameters assuming independent zero-mean Laplace distributions for the priors.
- In standard ℓ_1-regularization, a different non-sparse solution $\boldsymbol{\theta}^*$ may exist.
- To avoid the solution $\boldsymbol{\theta}^*$ instead of the sparse solution $\boldsymbol{\theta}_0^*$, a weighted ℓ_1-regularization can be used.

$$\arg\min_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) := J(\boldsymbol{\theta}) + \lambda \|\mathbf{W}\boldsymbol{\theta}\|_1$$



Standard ℓ_1-regularization



Weighted ℓ_1-regularization

Training of Neural Networks using ℓ_1-Regularization Strategies

- **High-fidelity only strategies:**

Standard ℓ_1-regularization

$$\arg\min_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) := J(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_1$$

Weighted ℓ_1-regularization

$$\arg\min_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) := J(\boldsymbol{\theta}) + \lambda \|\mathbf{W}\boldsymbol{\theta}\|_1$$

$$w_{ii} = \left(|\theta_{k-1,i}| + \epsilon_w\right)^{-1}$$

- $\mathbf{W}$ is a diagonal matrix with entries $\{w_{ii}\}_{i=1}^{n_\theta}$.
- $\theta_{k-1,i}$ is the i-th parameter from previous iteration.
- ϵ_w is a small number to avoid division by zero.

Training of Neural Networks using ℓ_1-Regularization Strategies

- **Bi-fidelity strategies:**

Standard ℓ_1-regularization

$$\arg\min_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) := J(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathrm{LF}}\|_1$$

- $\boldsymbol{\theta}_{\mathrm{LF}}$ are parameters from a low-fidelity trained network.

Weighted ℓ_1-regularization

$$\arg\min_{\boldsymbol{\theta}} R(\boldsymbol{\theta}) := J(\boldsymbol{\theta}) + \lambda \|\mathbf{W}\boldsymbol{\theta}\|_1$$

$$w_{ii} = \left(|\theta_{\mathrm{LF},i}| + \epsilon_w\right)^{-1}$$

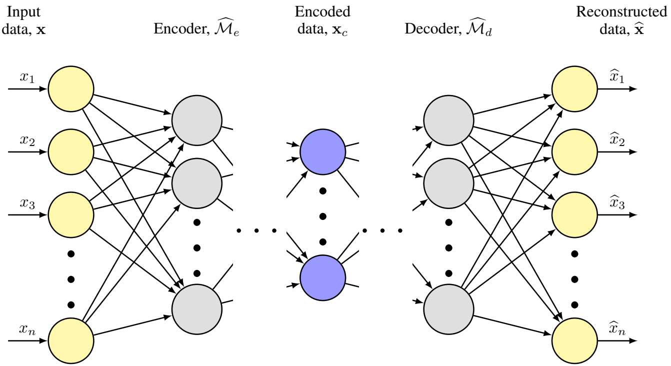
- $\mathbf{W}$ is a diagonal matrix with entries $\{w_{ii}\}_{i=1}^{n_\theta}$.
- ϵ_w is a small number to avoid division by zero.

De, Subhayan, and Alireza Doostan. "Neural network training using ℓ_1-regularization and bi-fidelity data." *Journal of Computational Physics* (2022): 111010.

Connection with Transfer Learning

- **Recall:** Transfer learning uses the knowledge gained from solving one problem to solve another related problem, where accurate or labeled data may be missing or limited.
- The bi-fidelity regularization strategies are generalizations of transfer learning of neural networks.
- However, instead of keeping the parameters of the first few layers fixed after training using the low-fidelity data, herein we use regularization to induce small variations in the high-fidelity trained network parameters.

Training of an Autoencoder



- Encoder network: $\mathbf{y} = \widehat{\mathcal{M}}_e$
- Decoder network: $\widehat{\mathbf{x}} = \widehat{\mathcal{M}}_d$
- Useful for dimension reduction, denoising, etc.

Example: Flow through a Dual Throat Nozzle

- **QoI:** Shock position
- Uncertain initial condition
- We use a simplified model based on Burger's equation given by

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = \frac{\partial}{\partial x}\left(\frac{\sin^2 u}{2}\right), \quad 0 \leq x \leq \pi, t > 0,$$
$$u(x, \delta, 0) = \delta \sin x, u(0, \delta, t) = u(\pi, \delta, t) = 0.$$

- The steady-state solution

Shock position

$$\lim_{t \to \infty} u(x, \delta, t) = \begin{cases} u^+ = \sin x, 0 \leq x \leq X_s \\ u^- = -\sin x, X_s \leq x \leq \pi \end{cases}$$

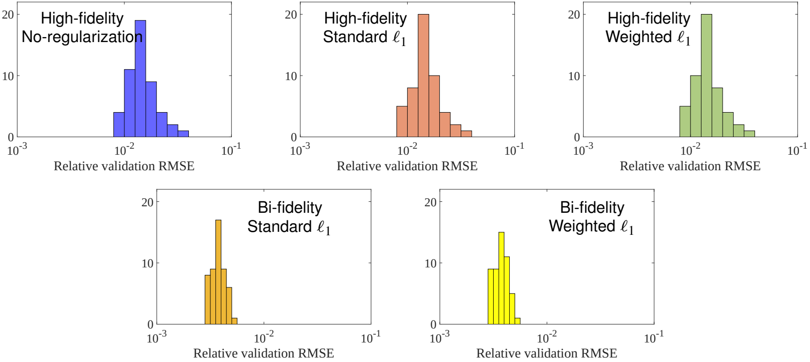
- The shock is at $x = X_s$

$$X_s = \begin{cases} \sin^{-1}\left(\sqrt{1-\delta^2}\right) < \frac{\pi}{2}, -1 < \delta \leq 0 \\ \pi - \sin^{-1}\left(\sqrt{1-\delta^2}\right) > \frac{\pi}{2}, 0 < \delta < 1 \end{cases}$$

- For the low-fidelity model, we use coarser discretization of x.
- *Encoder:* 3 hidden layers with 128, 64, and 16 neurons.
- *Decoder:* 3 hidden layers with 16, 64, and 128 neurons.

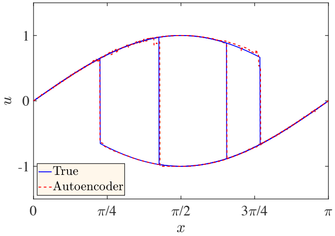
Example: Flow through a Dual Throat Nozzle

- Histograms for 50 replications of training and validation datasets.



Example: Flow through a Dual Throat Nozzle

Strategy	Mean validation error
HF (no regularization)	1.57×10^{-2}
HF Standard ℓ_1	1.57×10^{-2}
HF Weighted ℓ_1	1.56×10^{-2}
BF Standard ℓ_1	3.76×10^{-3}
BF Weighted ℓ_1	3.76×10^{-3}



- One order of magnitude improvement in validation error.
- Four reconstructed solutions.

Conclusions

- For large-scale engineering and scientific applications, one of the main obstacles in training neural networks is generating large datasets either using high-fidelity models or costly experimental setups.
- We propose training strategies using low-fidelity data and adapt it using a smaller high-fidelity dataset.
- Bi-fidelity weighted learning leads to considerable improvements in prediction error over standard training.
- The networks trained using bi-fidelity ℓ_1-regularization strategies have better predictive accuracy compared to networks trained using high-fidelity only strategies.

References

[1] De, S., Britton, J., Reynolds, M., Skinner, R., Jansen, K., & Doostan, A. (2020). On transfer learning of neural networks using bi-fidelity data for uncertainty propagation. *International Journal for Uncertainty Quantification*, 10(6).

[2] De, S., & Doostan, A. (2022). Neural network training using ℓ_1-regularization and bi-fidelity data. *Journal of Computational Physics*, 458, 111010.

[3] De, S., Hassanaly, M., Reynolds, M., King, R. N., & Doostan, A. (2022). Bi-fidelity Modeling of Uncertain and Partially Unknown Systems using DeepONets. *arXiv preprint arXiv:2204.00997*.

THANK YOU

Acknowledgments: This work was supported by funding from DOE's Advanced Scientific Computing Research (ASCR) program and by the AFOSR grant FA9550-20-1-0138.