Solving Partial Differential Equations with Uncertainties Using Neural-Networks (A surrogate forward model)

Sajed N. Zarrinpour

Dr Khadijeh Nedaiasl, Dr Parvin Razaghi

sa.zarrinpour@iasbs.ac.ir

Institute for Advanced Studies in Basic Sciences

May 2020

Outline

“The key is in not spending time, but in investing it”
- Stephen R. Covey.

Use Case Example

Breast Cancer

Use Case: Breast Cancer

MRI-guided biopsy

Figure: https://healthmanagement.org/products/view/breast-biopsy-mri-coil-sentinelle-hologic

Use Case: Breast Cancer

MRI-guided biopsy sample image

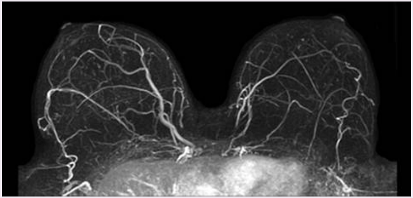

Figure: https://www.cedars-sinai.edu/Patients/Programs-and-Services/Imaging-Center/For-Patients/Exams-by-Procedure/MRI/MR-Guided-Breast-Biopsy/

Use Case: Breast Cancer

Ultra sound imaging

Figure: https://www.philips.com/a-w/about/news/archive/standard/news/press/2018/20181025-philips-debuts-integrated-breast-ultrasound-solution-to-make-exams-easier-and-faster-for-patients-and-clinicians.html

Use Case: Breast Cancer

Ultra sound sample image

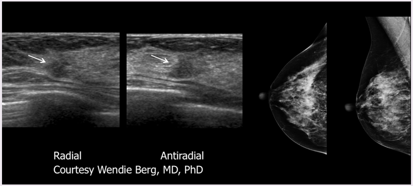

Figure: https://www.radiology.pitt.edu/node/225

Use Case: Breast Cancer

FEM Model $\sim$ 2 hours

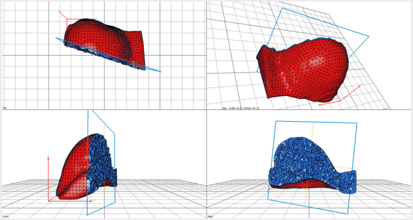

Figure: https://aapm.onlinelibrary.wiley.com/doi/10.1002/mp.12673

Use Case: Breast Cancer

FEM - ML Model $\sim$ 0.2 seconds

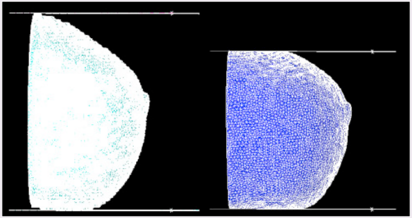

Figure: https://www.sciencedirect.com/science/article/abs/pii/S0010482517303177

What are the bottlenecks?

Finite Element Method

Poisson Equation on Unit Square - Strong Form

$$
\begin{aligned}
-\Delta u = & \quad f(x) \quad ,\text{in } \Omega = [0 \times 1] \times [0 \times 1] - [0,1) \\
u = & \quad 0 \quad ,\text{on } \partial\Omega
\end{aligned}
$$

Finite Element Method

Poisson Equation on Unit Square - Weak Form

$\sum_{i=1}^{M} c_i < \phi_i, \phi_j >=< f, \phi_j > \qquad i = 1, ..., M,$
where
$u_h = \sum_{i=1}^{M} c_i\phi_i \; and \in V_h = span\{\phi_i\}_{i=1}^{M}$

Removing the bottleneck.

Is It Possible?

Possibility

Universal Approximation Theorem

Let $\phi : \mathbb{R} \to \mathbb{R}$ be a non-constant, bounded, and continuous function. Let I_m denote the m-dimensional unit hypercube $[0, 1]^m$. The space of real-valued continuous functions on I_m is denoted by $C(I_m)$. Then, given any $\epsilon > 0$ and any function $f \in C(I_m)$, there exist an integer N, real constants $v_i, b_i \in \mathbb{R}$ and real vectors $w_i \in \mathbb{R}^m$ for i=1,...,N, such that we may have define : $F(x) = \sum_{i=1}^{N} v_i \phi(w_i^T x + b_i)$ as an approximate realization of the function f; that is, $|F(x) - f(x)| < \epsilon$ for all $x \in I_m$.

Possibility

Universal Approximation Theorem

For any Lebesgue-integrable function $f\colon \mathbb{R}^n \to \mathbb{R}$ and any $\epsilon > 0$, there exists a fully-connected ReLU network $\mathcal{A}$ with width $d_m \leq n + 4$, such that the function $F_{\mathcal{A}}$ represented by this network satisfies:
$\int_{\mathbb{R}} \|f(x) - F_A(A)\| dx < \epsilon$

Alright. Where to begin?!

Neural-Network

Artificial Neuron

- **W**eight, **b**ias

Neural-Network

Artificial Neuron

- **W**eight, **b**ias
- Activation function (ψ)

Neural-Network

Artificial Neuron

$$\psi(WX + b)$$

Different Combinations $\sim$ Different Networks

Let's see it in action!

Deep Ritz Method

Poisson Equation - Neural Networks Form

$u_\theta(x) = a.f_n(x) \circ \ldots \circ f_1(x) + b$
where
$f_i(x) = \phi(W_i2.\phi(W_i1.x + b_{i1}) + b_{i2}) + x$ and $a \in \mathbb{R}^m, b \in \mathbb{R}$
and $\phi(x) = \max\{x^3, 0\}$

IASBS

Deep Ritz Method

(a) DRM - 811 parameter, (b) FDM - 1681 parameter

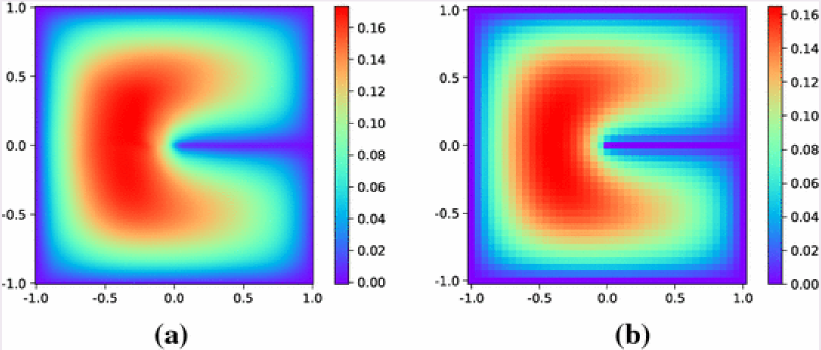

Figure: https://arxiv.org/abs/1710.00211

Neural-Network

How It Works

- structure

Neural-Network

How It Works

- structure
- loss function

Are You READY?!

Let's dive deep!

Linear Case

effective conductance in inhomogeneous media

Linear Case

Effective Conductance

$$\begin{aligned} A_{\text{eff}}(a) &= \min_{u(x)} \int_{[0,1]^d} a(x)||\nabla u(x) + \xi||_2^2 \mathrm{d}x. \quad &,\text{in } \Omega = [0 \times 1] \\ u(0) &= u(n) &,\text{on } \partial\Omega \end{aligned}$$

Final result

$mean(A_{\text{eff}}(a)) = 0.76800650$,
With 1.021×10^{-3} L^2-Error

Steps

- Data Set Generation

Linear Case

Minimization Form

$$\begin{aligned} A_{\mathrm{eff}}(a) = &\quad \min_{u(x)} \int_{[0,1]^d} a(x)||\nabla u(x) + \xi||_2^2 \mathrm{d}x. &\quad ,\text{in } \Omega = [0 \times 1] \\ u(0) = &\quad u(n) &\quad ,\text{on } \partial\Omega \end{aligned}$$

Equivalent PDE Form

$$-\nabla \cdot (a(x)(\nabla u(x) + \xi)) = 0$$

Linear Case

Calculate u_a

$$(L_a u)_i := \sum_{k=1}^{d} \frac{-a_{i+\frac{1}{2}e_k} u_{i+e_k} + (a_{i-\frac{1}{2}e_k} + a_{i+\frac{1}{2}e_k}) u_i - a_{i-\frac{1}{2}e_k} u_{i-e_k}}{h^2}$$

$$(b_a)_i := \sum_{k=1}^{d} \frac{\xi_k (a_{i+\frac{1}{2}e_k} - a_{i-\frac{1}{2}e_k})}{h}$$

Effective Conductance

$$A_{\mathrm{eff}}(a) = h^d (u_a^T L u_a - 2 u_a^T b_a + a^T 1)$$

Steps

- Data Set Generation
- Neural-Network Model

Linear Case

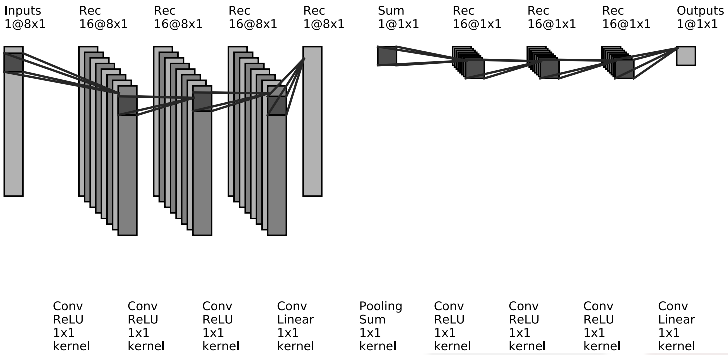

Figure: Neural-Network architecture

Linear Case

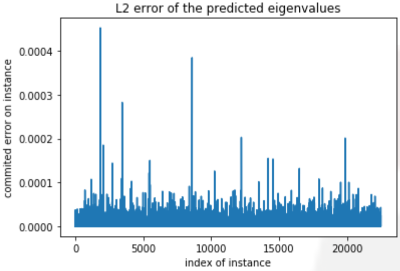

Figure: committed error per sample over the prediction set

Linear Case

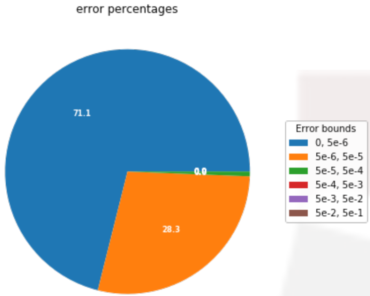

Figure: committed error per sample over the prediction set distribution

Nonlinear Case

Nonlinear Shrödinger equation with inhomogeneous background potential

Nonlinear Case

Shrödinger Equation

$$\begin{aligned} -\Delta u(x) + a(x)u(x) + \sigma u(x)^3 = & \quad E_0 u(x) \qquad \text{,in } \Omega = [0 \times 1]^2 \\ \int_{[0,1]^2} u(x)^2 \mathrm{d}x = & \quad 1 \\ u(0) = & \quad u(n) \qquad \text{,on } \partial\Omega \end{aligned}$$

Final result

$mean(E) = 10.17474556$,
With 7.235×10^{-5} L^2-Error

Steps

- Data Set Generation

Nonlinear Case

FDM

$$(Lu)_i + a_i u_i + \sigma u_i^3 = E_0 u_i,$$

$$\sum_{i=1}^{n^2} u_i^2 h^2 = 1,$$

$$(lu)_i := \sum_{k=1}^{d} \frac{-u_{i+e_k} + 2u_i - u_{i-e_k}}{h^2}$$

Steps

- Data Set Generation
- Neural-Network Model

Nonlinear Case

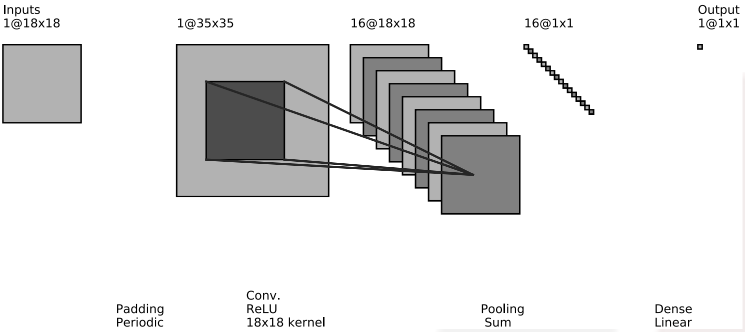

Figure: Single convolutional layer.

Nonlinear Case

Periodic Padding Psudo Code

```
class PeriodicPadding2D(layers.Layer):
  ...
  def wrap_pad(self, input, size):
    M1 = tf.concat([
           input[:,:, -size:],
           input, input[:,:, 0:size]
           ], 2)
    M1 = tf.concat([
           M1[:,-size:, :],
           M1,
           M1[:,0:size, :]
           ], 1)
    return M1
  ...
```

Nonlinear Case

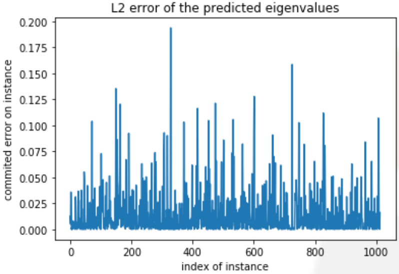

Figure: committed error per sample over the prediction set

Nonlinear Case

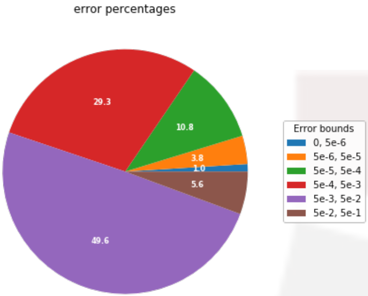

Figure: committed error per sample over the prediction set

Conclusion

- Bypassing the calculation of the basis

Conclusion

- Bypassing the calculation of the basis
- Better accuracy in less cost

Conclusion

- Bypassing the calculation of the basis
- Better accuracy in less cost
- Almost the same for Linear and Nonlinear

Future Works

- Studying methods for label scaling/normalization

Future Works

- Studying methods for label scaling/normalization
- Incorporating more physical knowledge to the model

Future Works

- Studying methods for label scaling/normalization
- Incorporating more physical knowledge to the model
- Using this method for simulation of soft tissue deformation

Q&A

"Ask, and it shall be given you!"

Matthew 7:7

References

- Solving PDE problems with uncertainty using neural-networks
- The Deep Ritz Method
- A finite element-based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real-time
- Universal Approximation Theorem
- Quotes about the time