Adversarial Uncertainty Quantification in Physics-Informed Neural Networks

Yibo Yang, Paris Perdikaris

University of Pennsylvania

26.05.2023

The model: UQPINN

$$\text{UQPINN} = \text{GAN} + \text{PINN}$$

1. **UQPINN** : Uncertainty Quantification Physics-Informed Neural Network
2. **GAN** : Generative Adversarial Network
3. **PINN** : Physics-Informed Neural Network

Novalty

”we will develop a flexible variational inference framework that will allow us to train such models directly from noisy input/output data, and predict outcomes of non-linear dynamical systems that are partially observed with quantified uncertainty”

– *Yibo Yang, Paris Perdikaris*

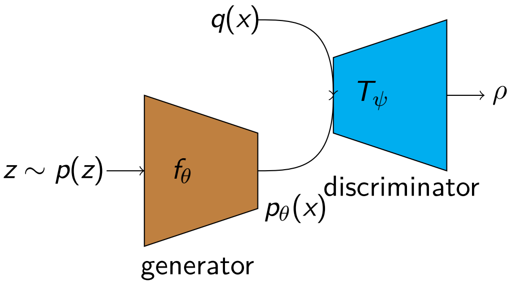
Novalty

"we will develop a flexible variational inference framework that will allow us to train such models directly from noisy input/output data, and predict outcomes of non-linear dynamical systems that are partially observed with quantified uncertainty"

– *Yibo Yang, Paris Perdikaris*

Using adversarial approach to handle randomness in observations.

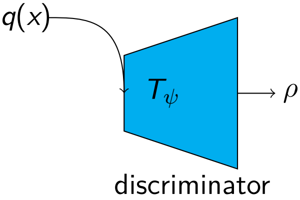
GAN



$$\max_{\psi} \mathcal{L}_{\mathcal{D}}(\psi)$$

$$\min_{\theta,\phi} \mathcal{L}_{\mathcal{G}}(\theta, \phi) + \beta\ \mathcal{L}_{PDE}(\theta)$$

GAN

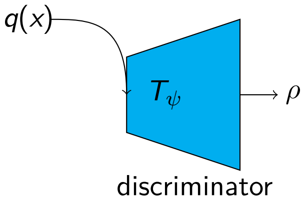


$$\underset{\psi}{argmin} \frac{\rho(y = +1|x, t, u)}{\rho(y = -1|x, t, u)}$$

$$p_\theta(x, t, u) = \rho(x, t, u|y = +1)$$

$$q(x, t, u) = \rho(x, t, u|y = -1)$$

GAN



$$\underset{\psi}{argmin} \frac{\rho(y=+1|x,t,u)}{\rho(y=-1|x,t,u)}$$

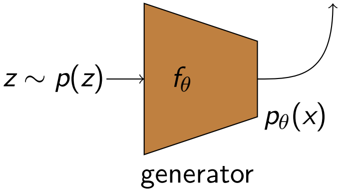
$$p_\theta(x,t,u) = \rho(x,t,u|y=+1)$$

$$q(x,t,u) = \rho(x,t,u|y=-1)$$

$$T_\psi \approx \rho(y=+1|x,t,u)$$

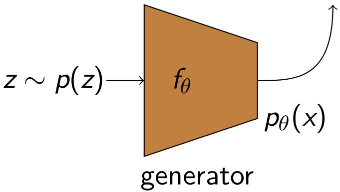
$$\mathcal{L}_D(\psi) = \mathbb{E}_{q(x,t)p(z)}[log\, \sigma(T_\psi(x,t,f_\theta))] + \mathbb{E}_{q(x,t,u)}[log(1-\sigma(T_\psi))]$$

GAN



$$\underset{\theta}{\mathrm{argmax}}\ \mathbb{KL}\left[p_\theta(x, t, u) \| q(x, t, u)\right]$$

GAN



$$\underset{\theta}{argmax}\ \mathbb{KL}\left[p_\theta(x, t, u) \| q(x, t, u)\right]$$

$$\begin{aligned}\mathcal{L}_G(\theta, \psi) = \mathbb{E}_{q(x,t)p(z)}[&T_\psi(x, t, f_\theta(x, t, z)) \\ &+ (1-\lambda) log(q_\phi(z|x, t, f_\theta(x, t, z)))]\end{aligned}$$

Experiment Setup

Author's Experiment Setup

GPU	NVIDIA Tesla P100(16GB)
DL framework	Tensorflow v1.10
Formula	1. pedagogical ODE 2. Burgers' equation 3. Darcy flow
Model	UQPINN

Experiment Setup

	Author's Experiment Setup	My Experiment Setup
GPU	NVIDIA Tesla P100(16GB)	MX450(2GB)
DL framework	Tensorflow v1.10	Pytorch v1.9.0
Formula	1. pedagogical ODE 2. Burgers' equation 3. Darcy flow	1. pedagogical ODE 2. Burgers' equation 3. Darcy flow
Model	UQPINN	1. UQPINN 2. PINN

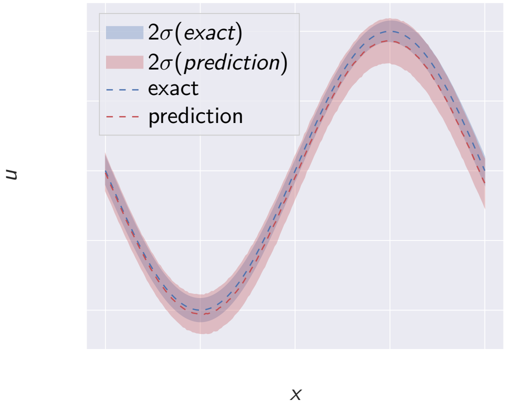
Experiment Setup

	Author's Experiment Setup	My Experiment Setup
GPU	NVIDIA Tesla P100(16GB)	MX450(2GB)
DL framework	Tensorflow v1.10	Pytorch v1.9.0
Formula	1. pedagogical ODE 2. Burgers' equation 3. Darcy flow	1. pedagogical ODE 2. Burgers' equation 3. Darcy flow
Model	UQPINN	1. UQPINN 2. PINN

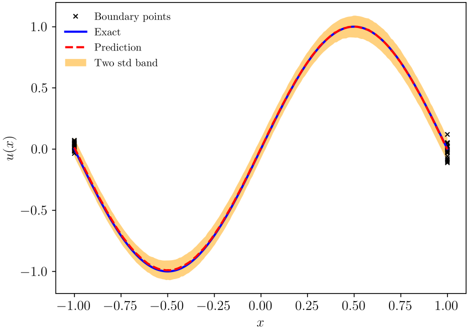
parameters are set to be the same as the author's

pedagogical ODE

$$
\begin{aligned}
&\frac{\partial^2 u}{\partial x^2} - u^2 \frac{\partial u}{\partial x} f(x) && x \in [-1, 1] \\
&f(x) = -\pi^2 sin(\pi x) - \pi cos(\pi x) sin(\pi x)^2 \\
&u(x) \sim \mathcal{N}(sin(\pi), noise) && x = \{-1, 1\}
\end{aligned}
$$



$2\sigma(exact)$
$2\sigma(prediction)$
exact
prediction
u
x



Boundary points
Exact
Prediction
Two std band
1.0
0.5
0.0
−0.5
−1.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
$u(x)$
x

Burgers Equation

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0 \qquad x \in [-1, 1], t \in [0, 1]$$

$$u(0, x) = -sin(\pi x)$$

$$u(t, x) = 0 \qquad x = \{-1, 1\}$$

$$\nu = \frac{0.01}{\pi}$$

Darcy

$$
\begin{aligned}
\nabla_{\vec{x}}(K(u)\nabla_{\vec{x}}u(\vec{x})) &= 0 & \vec{x} &\in [0, L_1] \times [0, L_2] \\
u(\vec{x}) &= 0 & x_1 &= L_1 \\
-K(u)\frac{\partial u(\vec{x})}{\partial x_1} &= q & x_1 &= 0 \\
K(u) &= K_s\sqrt{s(u)}\left(1 - (1 - s(u)^{\frac{1}{m}})^m\right)^2 & x_2 &= \{0, L_2\} \\
s(u) &= \left(1 + (\alpha(u_g - u))^{\frac{1}{1-m}}\right)^{-m}
\end{aligned}
$$