Studies on PINNs

The PINN Module and Future Studies

A. Salih Taşdelen

METU

March 21, 2023

Outline

The Progress

- Reproduced the laminar flow over a cylinder **rao˙physics-informed˙2020**.
- Tried to understand the implementation and how it could be generalized. Since the source is in Tensorflow 1 (**abadi˙tensorflow˙2016**), the user is the one who almost directly implements the computation graph.
- Read other papers that implements PINNs in TF2:
  - **DeepXDE** by **lu˙deepxde˙2021**. (Residual Based Adaptive Refinement)
  - **Elvet** by **araz˙elvet˙2021**. (Gradient Stack)
  - **dNNSolve** by **guidetti˙dnnsolve˙2021**. (Fourier Neural Nets)
  - **IDRLnet** by **peng˙idrlnet˙2021**. (Constraint Importance is proportional to its domain area)
  - **PyDEns** by **koryagin˙pydens˙2019**. (Deep Galerkin)
  - **TensorDiffEq** by **mcclenny˙tensordiffeq˙2021**. (`tf.gradients` instead of `tf.GradientTape`)

Problem in Mathematical Terms I

The Domain

- Suppose we are given a problem in D number of space time dimensions. Namely, we have time t, and $D-1$ number of spatial dimensions. Then, the domain can be represented as:
  - $\mathrm{D} = [t_0, t_1] \times \Omega \subset \mathbb{R}^D$ where,
  - $\Omega \subseteq \mathbb{R}^{D-1}$, is the spatial domain for which,
    - $\vec{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{D-1} \end{bmatrix}^T$ is an element of Ω.
  - We can represent the elements of space-time domain as a column vector $\varphi = \begin{bmatrix} t & x_1 & x_2 & \cdots & x_{D-1} \end{bmatrix}^T \in \mathbb{R}^D$.
  - With the above definitions, let us say that $\vec{u}(\varphi)$ is the desired solution of the problem.
  - Also let $\hat{u}(\varphi)$ be the solution approximated by the Neural Network.
  - We generalize the desired solution to be in any dimension, so that it could be the measure of various quantities like; velocity, pressure etc.

Problem in Mathematical Terms II

Governing Equations - PDEs

Let us look at the governing equations as well:

- Partial Differential Equations:
  - Suppose the PDEs have the form $f_i(\varphi, \vec{u}(\varphi)) := 0$, $\varphi \in T_\Omega \subseteq \mathrm{D}$.
  - If we have n_Ω number of PDEs their representation would be:

$$\vec{f}(\varphi, \vec{u}(\varphi)) := \begin{bmatrix} f_1 & f_2 & \cdots & f_{n_\Omega} \end{bmatrix}^T = \vec{0} \tag{1}$$

  - Let us denote the number of collocation points sampled for the PDEs as N_Ω. Which is equivalent to say, $|T_\Omega| = N_\Omega$.

Problem in Mathematical Terms III

Governing Equations - ICs

- Initial Conditions:
  - Suppose the ICs have the form $g_i(\varphi, \vec{u}(\varphi)) := 0$, $\varphi \in T_{0_i} \subseteq \mathrm{D}$.
  - $|T_{0_i}| = N_{0_i}$, number of collocation points samples for the i^{th} IC.
  - For the initial condition we now that $t = t_0$, hence $T_{0_i} = \{t_0\} \times \mathrm{I}_i$.
  - The spatial domain of IC is $\mathrm{I}_i \subseteq \Omega$.
  - The spatial domains of initial conditions cannot coincide thus, $\mathrm{I}_i \cap \mathrm{I}_j = \varnothing$, $i \neq j$.
  - Let us define the set for all the sets for IC domains, $\Gamma_0 = \{T_{0_i}\}_{1 \leq i \leq n_0}$.
  - Since IC domains do not intersect we have the total number of IC collocation points as:

$$N_0 = \sum_{i=1}^{n_0} |T_{0_i}|$$

Problem in Mathematical Terms IV

Governing Equations - BCs

- Boundary Conditions:
  - Suppose the BCs have the form $h_i(\varphi, \vec{u}(\varphi)) := 0$, $\varphi \in T_{0_i} \subseteq \mathrm{D}$.
  - $|T_{\partial\Omega_i}| = N_{\partial\Omega_i}$, number of collocation points samples for the i^{th} BC.
  - For the boundary conditions, $T_{\partial\Omega_i} = [t_0, t_1] \times \mathrm{B}_i$.
  - The spatial domain of BC is $\mathrm{B}_i \subseteq \partial\Omega$.
  - The spatial domains of boundary conditions cannot coincide thus, $\mathrm{B}_i \cap \mathrm{B}_j = \varnothing$, $i \neq j$.
  - Let us define the set for all the sets for BC domains, $\Gamma_{\partial\Omega} = \{T_{\partial\Omega_i}\}_{1 \leq i \leq n_{\partial\Omega}}$.
  - Since BC domains do not intersect we have the total number of BC collocation points as:

$$N_{\partial\Omega} = \sum_{i=1}^{n_{\partial\Omega}} |T_{\partial\Omega_i}|$$

Problem in Mathematical Terms V

The Loss Functions

Suppose we use MSE to compute the Neural Networks Loss.

$$\mathrm{L}_0 := \mathrm{L}_0(\Gamma_0, \theta) = \sum_{T_{0_i} \in \Gamma_0} \frac{1}{|T_{0_i}|} \sum_{\varphi \in T_{0_i}} ||g_i(\varphi, \hat{u}(\varphi))||_2^2 \tag{2}$$

$$\mathrm{L}_{\partial\Omega} := \mathrm{L}_{\partial\Omega}(\Gamma_0, \theta) = \sum_{T_{\partial\Omega_i} \in \Gamma_{\partial\Omega}} \frac{1}{|T_{\partial\Omega_i}|} \sum_{\varphi \in T_{\partial\Omega_i}} ||h_i(\varphi, \hat{u}(\varphi))||_2^2 \tag{3}$$

$$\mathrm{L}_{\Omega} := \mathrm{L}_{\Omega}(T_{\Omega}, \theta) = \frac{1}{|T_{\Omega}|} \sum_{\varphi \in T_{\Omega}} ||f_i(\varphi, \hat{u}(\varphi))||_2^2 \tag{4}$$

$$L = \alpha_0 \mathrm{L}_0 + \alpha_{\partial\Omega} \mathrm{L}_{\partial\Omega} + \mathrm{L}_{\Omega} \tag{5}$$

Residuals

Notice that the functions f_i, g_i,and h_i's are the residuals that we are trying to minimize. In other words we want them to converge to 0. Thus NNs loss can be written in terms of $\mathrm{L}(f_i(\varphi), \vec{0})$, $\mathrm{L}(g_i(\varphi), \vec{0})$, and $\mathrm{L}(h_i(\varphi), \vec{0})$.

Generalization

Instead of separately handling PDEs, BCs, and ICs we can further generalize the residuals. We can think of each equation as a constraint, then each constraint has a domain and a residual. This way we will have only a single summation as a loss function.

$$\mathrm{L} := \mathrm{L}(\Gamma, \theta) = \sum_{T_i \in \Gamma} \frac{1}{|T_i|} \sum_{\varphi \in T_i} C(g_i(\varphi, \hat{u}(\varphi)), \vec{0}) \tag{6}$$

Loss

Here $C(y_{pred}, y_{true})$ represents any loss function.

How it Works I

Example

Let us consider the 1D Heat Conduction. The PDE, BCs, and IC are as follows:

$$\frac{\partial u}{\partial t} - 0.05\frac{\partial^2 u}{\partial x^2} = 0$$
$$u(0, x) = sin(3\pi x)$$
$$\left.\frac{\partial u}{\partial x}\right|_{\partial\Omega} = 0$$

The analytical solution is: $u(t, x) = cos(3\pi x)e^{-.05(3\pi)^2 t}$

How it Works II

Defining Constraints

How it Works III

Defining Model and Training

How it Works IV

Gradient Computation Algorithm

How it Works V

Training Algorithm

New Features

- A seperate module for domain generation.
- Efficient gradient calculation.
- Ready to use constraints, Neumann BCs, Dirichlet BCs, etc.
- Numeric derivative computation.

Future Work

- Residual based adaptive refinement.
- Constraint weights proportional to its domain area.
- Fourier Networks in combination to standard networks.
- Dimensionless inputs and outputs.

References I