

Session 4 - Practical session

Jamie M. Taylor (CUNEF Universidad, Madrid, Spain)

`jamie.taylor@cunef.edu`

BCAM - In-Deep workshop

- ▶ In this session, we will consider a 2D PINN problem, and the idea is that you can see how tensorflow itself functions.
- ▶ The code at <https://github.com/jamie-m-taylor/In-Deep-examples> can help you here - You should be able to pull most necessary functionality out of the code there, making appropriate changes.
- ▶ Of course, the idea is to learn how to use the program, not to solve the problem itself - Feel free to “tweak” the problem I propose to your interests, or find a way to make it harder if you’re already familiar with tensorflow.

The problem

- ▶ We will do a 2D PINN inverse problem with a single, unknown parameter. We assume that

$$\Delta u(x, y) = k \left((100(x^2 + y^2) \sin(5x^2 + 5y^2 - 5) - 20 \cos(5x^2 + 5y^2 - 1)) \right)$$

for some k on the unit disc, with homogeneous Dirichlet condition on the boundary.

- ▶ The exact solution will be given by $k = 1$, $u(x, y) = \sin(5 - 5x^2 - 5y^2)$.
- ▶ We will have access to data, and try to recover k .

Ideas on how to start

- ▶ We can use a custom layer to impose the boundary condition, $1 - x^2 - y^2$ is a suitable cutoff function.
- ▶ The loss will consist of a weighted average of the loss for the PDE and the loss for the interpolation over data - By recording each separately, you can assess the quality of each component.
- ▶ When doing MC on a disk, to sample uniformly, use polar coordinates, sampling θ uniformly on $(0, 2\pi)$, and taking $r = \sqrt{u}$ with u uniform on $(0, 1)$ - feel free to try other strategies.
- ▶ Build a dataset of a small number of points, (x_j, y_j, u_j) , where $u_j = u^*(x_j, y_j) + \epsilon_j$ using random number generators.
- ▶ Once you have something that's running, try playing with the parameters to see how well you can approximate k .
- ▶ If you want to record k over training, use a custom callback.

Things to consider

- ▶ What architecture will you use? How big, what activation functions, how will you initialise your network?
- ▶ How will you weight your loss? It will have several components - try and experiment and see what works best.
- ▶ What kind of optimiser will you use? Tensorflow has a huge variety available in their documentation and they all have a variety of parameters, try experimenting to see how they affect convergence.