Adding and changing functions in PDE-Net

Arthur Grundner

5. April 2019

Let's say we want to use PDE-Net, that is, infer a PDE of a type that differs from our examples or we want to generate data by other methods. Then we have to follow these steps:

1. Create a new folder and best copy the files *common_methods.py*, *generate_data.py*, *inferring_the_pde.py*, *main.py* and *more_methods.py* from the folder *non-linear_pde* into it.

2. To generate different data for $t > 0$, we modify *generate_data.py* as described in the file. In order to generate different initial data, we modify the method *initgen* in *common_methods.py*.

3. In *main.py* we can adjust the options in the **options-dictionary**: 'Mesh-size', 'layers' and 'dt' are set according to our given data. With 'batch_size' and 'downsample_by' we can sub-divide the data into multiple samples, with which training takes place. Usually we keep the 'noise_level' at 0.0. It may be interesting to increase this value for testing purposes (cf. *common_methods.py/addNoise* for how we add the noise). According to the expected maximal order 'max_order' of the unknown PDE, we might have to increase the size of the filters 'filter_size', which naturally also increases the amount of learnable parameters. Repeating the warmup-step often with different initial values for the coefficients has a high impact on the performance. We can set the amount of repeats with the 'iterations'-parameter. Given data that behaves nicely, that is, it wraps around on the boundary (periodic boundary conditions), we can set 'boundary_cond' to 'PERIODIC'. By this, we can pad the input before each convolution step and the amount of layers we can have is unbounded.

4. Finally we get to *inferring_the_pde.py*:

   Line 42: In case there are multiple additional parameters (not derivative-coefficients) in F $(= u_t)$ to be discovered, we have to adjust the type of *self.param*. The same should then be done in the print-statement in Line 245.

   Line 122: Here we can modify how the parameters should be initialized.

   Line 158: In case that we have

$$F = u_t = \sum_{0 \leq i+j \leq N} c_{ij} \frac{\partial^{i+j} u}{\partial x^i \partial y^j} + f(u),$$

   we can simply adjust the function f in *more_methods.py* to fit our function. Otherwise we adjust 'out' + 'f' so that it matches u_t.