September summary

Work done up to September starting at “Error vs Residual Analysis”. In summary:

Looked at where loss decrease didn’t decrease error by plotting both side by side for the same problem.

Obtained curves of loss and error as the model trains, calculated locally (at the collocation points being used at each iteration) and globally (on a fine grid predicting u over the entire domain, and the pde loss and error at those points)

Checked how good the ability of PINNs to estimate gradient and curvature was based on standard PINN network at different steps before resampling and after a full simulation.

Lastly, looked at the error of using different sources of information to guide resampling, starting a small hyper-parameter study.

Contents

[Error vs Residual Analysis 3](#_Toc147223488)

[Loss and Error – At points vs globally 5](#_Toc147223489)

[Gradient and Curvature Estimation through PINNs 7](#_Toc147223490)

[Error of resampling-PINNs using different information. 10](#_Toc147223491)

**In September:**

Looked at hyperparameters for every case and found which work best for our proposed methods.

Wanted to look at error and loss vs iteration curves for different cases.

Investigated effect of network depth on solution.

Attempted use of GPU.

September

# Hyperparameter investigation

The default parameters work best for using residual information only, but aren't as good as Hyp1 for most of the geo-based settings.

Therefore using Hyperparameter set 1 (k=0.5, c=1) for geo based simulations from now on

# Depth Investigation

Depth of 5 gives error E-4 in some individual runs, but is very inconsistent, hence average error of simulations is much higher. Using Depth 4 makes residual methods inconsistent but using curvature still produces good consistent result. At depth 5 it remains accurate but is less consistent (one bad run).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Entry |  |  | Time (h) | Error |
| 1 | 2 layer | Residual | 4.23 | 2.22E-03 |
| 2 | 3 layer | Residual | 6.49 | 3.54E-04 |
| 3 | 4 layer | Residual | 8.81 | 1.35E-01 |
| 4 | 5 layer | Residual | 11.00 | 1.48E-01 |
| 5 | 2 layer | uxt | 4.23 | 1.87E-03 |
| 6 | 3 layer | uxt | 6.63 | 5.05E-04 |
| 7 | 4 layer | uxt | 8.96 | 3.64E-04 |
| 8 | 5 layer | uxt | 10.17 | 2.05E-02 |
| 9 | 2 layer | res-> uxt | 4.42 | 1.99E-03 |
| 10 | 3 layer | res-> uxt | 6.50 | 5.01E-04 |
| 11 | 4 layer | res-> uxt | 7.26 | 4.99E-03 |
| 12 | 5 layer | res-> uxt | 10.48 | 4.52E-02 |
| 13 | 2 layer | res-> uxut2 | 4.32 | 1.75E-03 |
| 14 | 3 layer | res-> uxut2 | 6.77 | 4.43E-04 |
| 15 | 4 layer | res-> uxut2 | 9.37 | 8.73E-02 |
| 16 | 5 layer | res-> uxut2 | 10.55 | 1.53E-01 |
| 17 | 2 layer | res->uxut3 | 4.42 | 2.16E-03 |
| 18 | 3 layer | res->uxut3 | 6.76 | 5.03E-04 |
| 19 | 4 layer | res->uxut3 | 9.21 | 1.44E-01 |
| 20 | 5 layer | res->uxut3 | 11.73 | 2.62E-01 |
| 21 | 2 layer | res->uxut4 | 4.38 | 2.14E-03 |
| 22 | 3 layer | res->uxut4 | 6.71 | 6.68E-04 |
| 23 | 4 layer | res->uxut4 | 9.07 | 1.16E-01 |
| 24 | 5 layer | res->uxut4 | 12.18 | 2.12E-01 |

Res = residual. U\_xt = curvature . uxut2 = avg(ux,ut). Uxut3 = max(). Uxut4 = rms()

# Error Curves

Used a finer FDM grid to evaluate local error better. Evaluating loss before resample leads to a smoother loss curve that’s easier to compare. For dual methods at 120,000 one information source is swapped for another.

Last month’s work

# Error vs Residual Analysis

Clamping helps observe that loss is quite blind to smaller errors in the range of 0 to 4, and so wouldn’t guide new points to those zones. They correspond to areas where gradients of u in t are highest. I want to follow this up with a similar map of curvature u\_xt at the same point and/or other geometric information used to guide resampling.

|  |
| --- |
| C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Residual_vs_uError_NotClamped.pngC:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Residual_vs_uError_ClampedAt4.png |
| Figure 1 – Comparison of PDE residual (the cost function value) vs actual error in U, averaged over 10 runs. |

# Loss and Error – At points vs globally

Train = at current training points (The local loss/error)

Test = Over test points (that cover the entire domain)

Much harder to evaluate loss as jumps frequently with optimiser changes every re-sample.

|  |
| --- |
|  |
|  |
|  |

# Gradient and Curvature Estimation through PINNs

15 000 Steps is the point at which the first re-sample is usually taken. Therefore it’s important that the model is capable of estimating the areas of high gradient / curvature correctly at 15 000 steps. This is shown in figure 2 and 3. All of this is based on a [2, 64, 64, 64, 1] tanh + glorot normal NN, using ADAM optimiser.

Figure 4 shows how well the model is capable of estimating the areas at the end of training (using the same PINNs but with adaptive resampling of points every 2000 steps based on residual information).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Ground Truth | 2k Steps | 15k Steps |
|  | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GroundTruth - u_x.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_2k_ux.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_15k_ux.png |
|  | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GroundTruth - u_t.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_2k_ut.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_15k_ut.png |
| Figure 2 – Gradients obtained from PINNs vs Ground Truth | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Ground Truth | 2k Steps | 15k Steps |
|  | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GroundTruth - u_xx.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_2k_uxx.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_15k_uxx.png |
|  | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GroundTruth - u_tt.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_2k_utt.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_15k_utt.png |
|  | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GroundTruth - u_tx.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_2k_uxt.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_15k_uxt.png |

Figure 3 – Curvature obtained through PINNs

|  |  |  |
| --- | --- | --- |
|  | Ground Truth | End Step |
|  | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GroundTruth - u_x.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_end_ux.png |
|  | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GroundTruth - u_t.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_end_ut.png |
|  | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GroundTruth - u_xx.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_end_uxx.png |
|  | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GroundTruth - u_tt.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_end_utt.png |
|  | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\GroundTruth - u_tx.png | C:\Users\mn17jilf\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gradients_end_uxt.png |

# Error of resampling-PINNs using different information.

In almost every case initialising with Hammersley yields better accuracy. This does not increase time by any significant amount so it is done for all cases moving forward.

The main interest was looking at the error of different sources of information for guiding resampling. The baseline to compare against was Wu’s loss driven resampling, utilising hyperparameters of c=1 and k=1. First those hyperparameters were used, keeping the amount and frequency of resampling the same.

|  |  |  |
| --- | --- | --- |
| Method | Time (h) | Error |
| Residual, Random Initialisation | 6.62 | 4.923E-04 |
| Residual, Hammersley | 6.50 | 5.151E-04 |

The sources of information looked at were curvature (differentiating u in the x and t directions) was highest, as well as combinations of gradients in x and t (Sum, Average, Max and RMS). Curvature seemed best:

|  |  |  |
| --- | --- | --- |
| Method | Time (h) | Error |
|  | 6.67 | 1.11E-03 |
| Sum( | 6.46 | 1.97E-03 |
| Avg( | 6.78 | 1.61E-03 |
| Max( | 5.60 | 1.51E-03 |
| RMS( | 5.60 | 1.85E-03 |

The next step was to investigate was how a combination of using residual and geometric information would do. A very simple first investigation involved simply using loss for half of the resampling, then switching to using geometric information. As from the first results using and max( did best, these were used.

|  |  |  |
| --- | --- | --- |
| Method | Time (h) | Error |
|  | 6.56 | 6.34E-04 |
| Max( | 7.06 | 1.21E-03 |

However, as all these used the hyper-parameters that were found to work best for residual, I also wanted to look at doing a quick investigation varying these and seeing whether which method was best changed as a result. (c controls random distribution, k controls spread of biased points), These factors were changed one at a time and tested for the more successful methods to save time (, max( and residual->.

|  |  |  |  |
| --- | --- | --- | --- |
| Method | hyper parameter choice | Time (h) | Error |
| Residual – Random | Default k=1,c=1 | 6.62 | 4.92E-04 |
| Residual - Hammersley | Default k=1,c=1 | 6.50 | 5.15E-04 |
|  | Default k=1,c=1 | 6.67 | 1.11E-03 |
| Max( | Default k=1,c=1 | 5.60 | 1.51E-03 |
|  | Default k=1,c=1 | 6.56 | 6.34E-04 |
| Residual - | k=0.5, c=1 | 6.73 | 5.41E-04 |
|  | k=0.5, c=1 | 6.75 | 4.91E-04 |
| Max( | k=0.5, c=1 | 7.08 | 6.84E-04 |
|  | k=0.5, c=1 | 6.72 | 3.83E-04 |
| Residual | k=1, c=5 | 6.04 | 1.50E-03 |
|  | k=1, c=5 | 6.86 | 4.67E-04 |
| Max( | k=1, c=5 | 6.51 | 3.92E-02 |
|  | k=1, c=5 | 6.82 | 4.46E-04 |
| Residual | k=2, c=5 | 7.06 | 5.01E-02 |
|  | k=2, c=5 | 6.64 | 8.31E-03 |
| Max( | k=2, c=5 | 6.65 | 3.05E-01 |
|  | k=2, c=5 | 6.78 | 5.44E-02 |

Yellow is better than residual on its own. It does involve residual. I need to check how it compares to the values Wu obtained, not just to my recreation of their method.

I also want to check the error history for these successful cases (would take too long to check for all of them). To see whether 100 rounds of resampling are really needed. Further on the main issue is that 100 resamples takes a very long time, and so checking accuracy using a much smaller number of resamples would be more relevant.

To Do:

Check error history.