Aristotle University of Thessaloniki
Faculty of Sciences
School of Informatics

MSc in Artificial Intelligence

Master Thesis

"Exploring New Techniques for Solving Differential Equations using Neural Networks"

Palaskos Achilleas (113)

Supervisor Professor: Vlahavas Ioannis

March 2024

CONTENTS

Partial Differential Equations (PDEs)

1. (Informal) Definition

- An equation that combines:
  - an unknown function
  - its partial derivatives
  - Boundary/Initial conditions

2. Why need a new method?

- Traditional methods (e.g. FEMs, FDMs):
  - Still highly effective in many scenarios
  - Provide accurate & efficient solutions
- BUT:
  - Have their own limitations
  - Not suitable for all kinds of problems

3. Limitations of traditional methods

- Require ***mesh generation*** (often computationally expensive)
- Difficulty with PDEs with **unknown or varying coefficients**
- Problems in ***PDEs derived from observational data***
- Struggle with problems exhibiting ***multiple scales***
- Challenges for ***rapidly changing solutions***

How a PINN works?

PDE:

B.C:

I.C:

1. Forward pass

2. Backpropagate

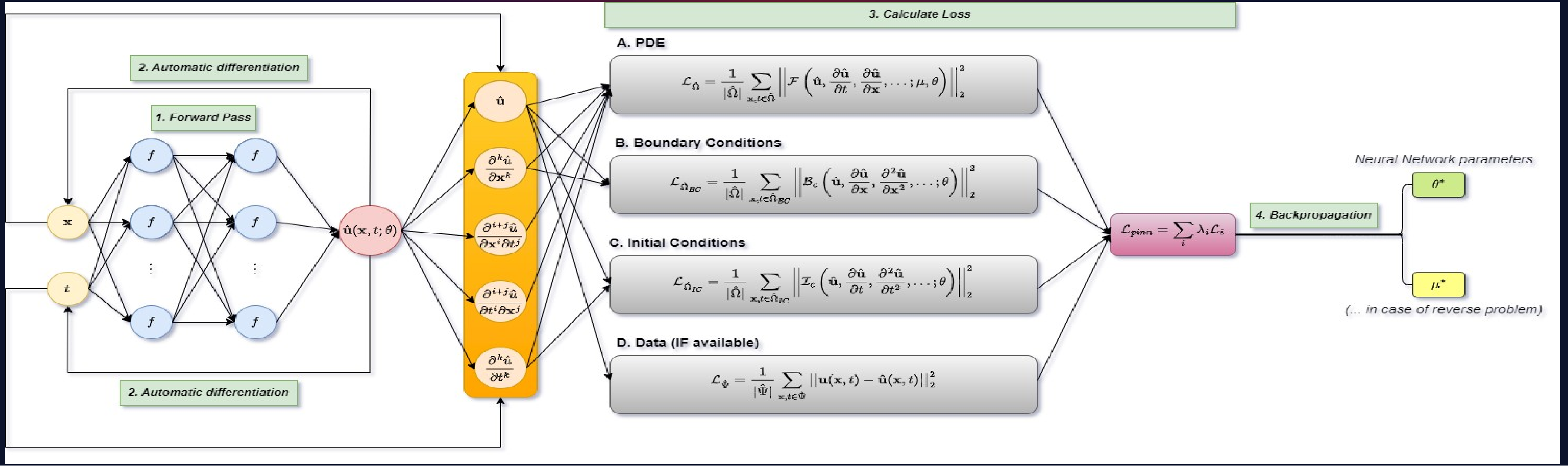
Calculate gradients of w.r.t. inputs

3. Calculate Loss

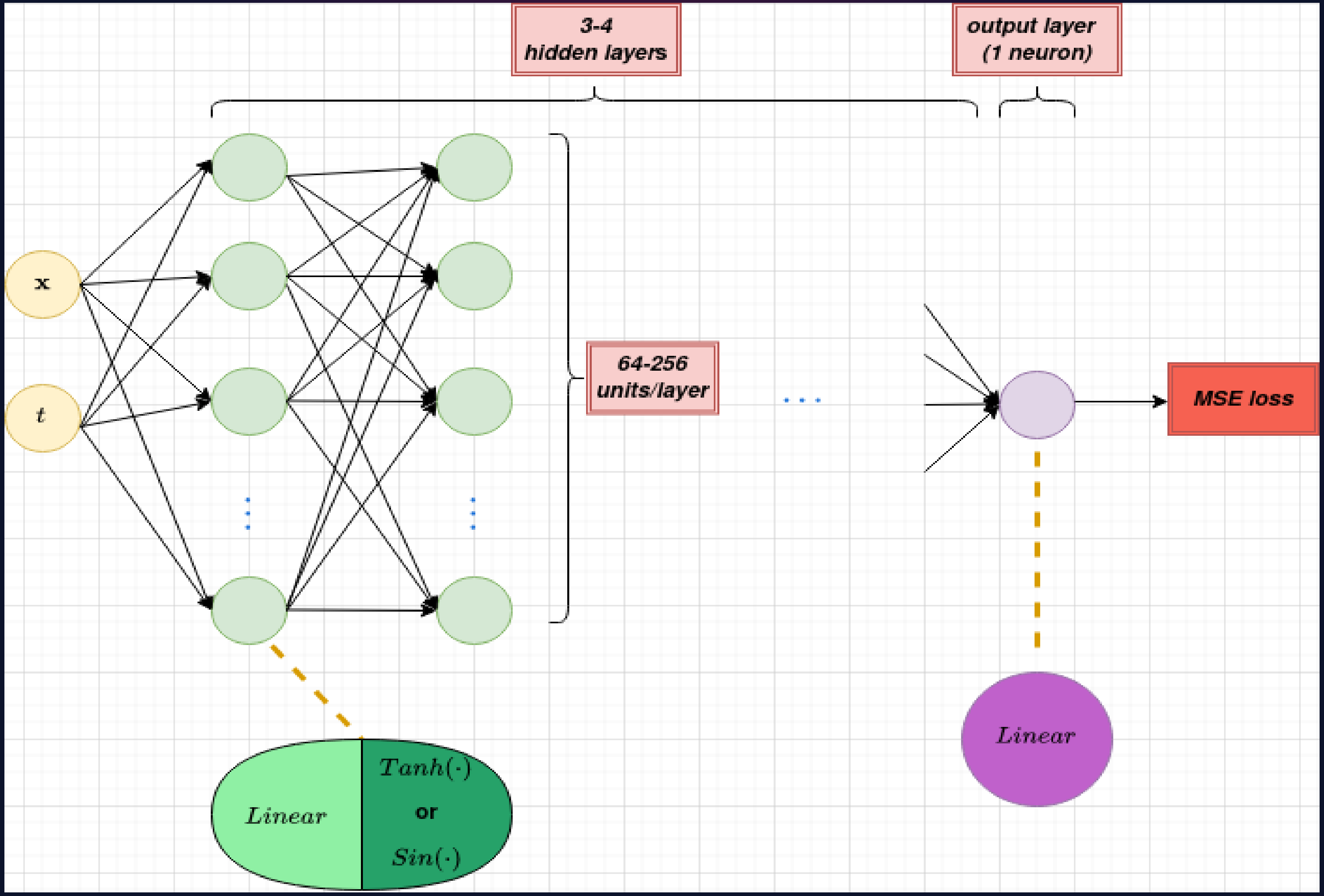
4. Optimize

Calculate gradients of

w.r.t. weights



PINN Architecture



Experimental Setup

1. Data

- Normalization ALL spatio-temporal points in
- Train set
  - points
  - randomly sampled
- Validation/Test set
  - points
  - uniformly sampled
  - Validation NOT always utilized

2. Performance metrics

- Normalized Mean Square Error (NMSE)
- Mean Absolute Error (MAE)

Important NOTE!

- In the following experiments
- ALL hyper-parameters were chosen with TUNING!

1. Homogeneous 1D wave equation (pulsating cord)

2nd Experiment

where: , and , with

subject to:

1. *Boundary Conditions (Dirichlet):*
2. *Initial Conditions:*

<u>Analytical Solution</u>:

- L-BFGS optimizer ()

(Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm)

- Duration:
- SUCCESSFUL … but too slow!

Why L-BFGS preferred to Adam ?

1. Nature of the Problem

- complex, non-convex optimization landscapes
- quasi-Newton more effective than 1st order methods

2. Convergence to Better Minima

- due to 1 more informative updates about curvature
- more accurate solutions

3. Memory Efficiency

- Hessian matrix NOT calculated BUT approximated
- suitable when Hessian too large to compute or store

4. Stability & Robustness:
better later at training

reliable & consistent steps

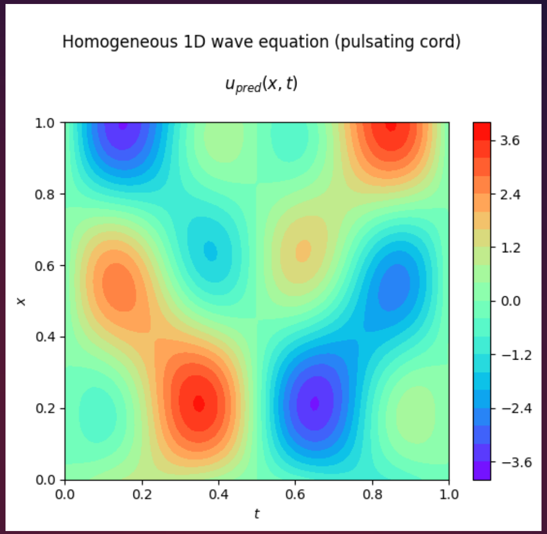
1st Experiment

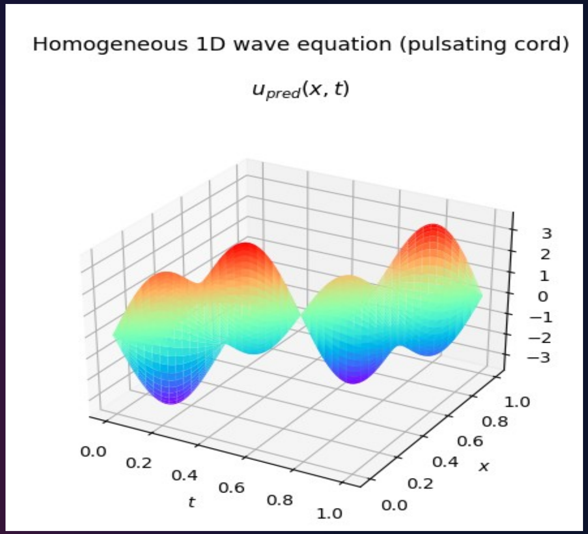
- ADAM optimizer ()
- UNSUCCESSFUL

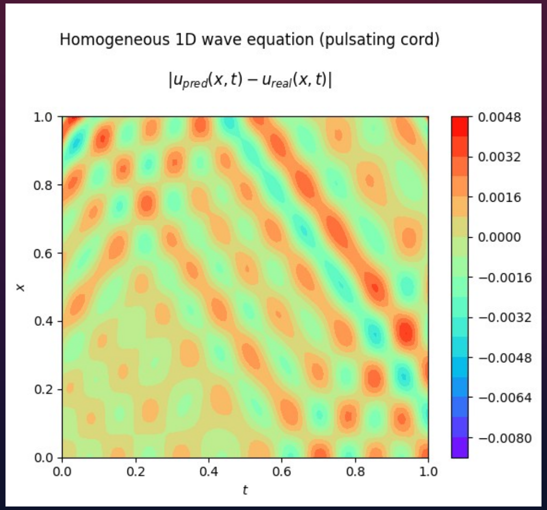
1. Homogeneous 1D wave equation (pulsating cord)

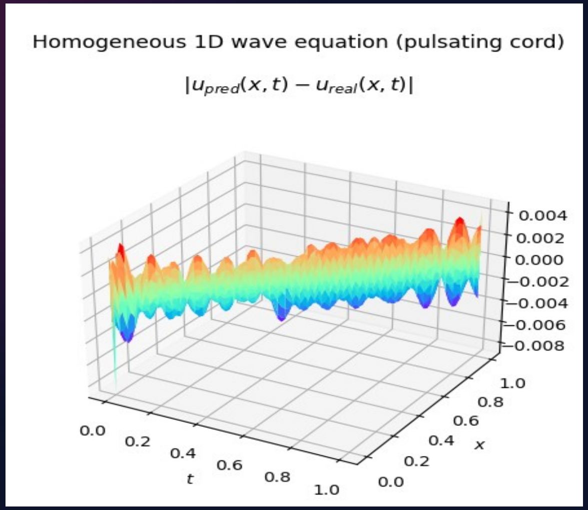
3rd Experiment

- **L-BFGS** optimizer ()
- ***RESAMPLING!***
  1. Choose new random samples
  2. Find greatest errors
  3. Create corresponding subdomain
  4. Randomly sample points ONLY in the subdomain
  5. Add new points in dataset
  6. Train new dataset ()
  7. Repeat until , then train with samples
- Duration:
- SUCCESSFUL … and QUICK!









2. Inhomogeneous 1D wave equation (source)

Experiment

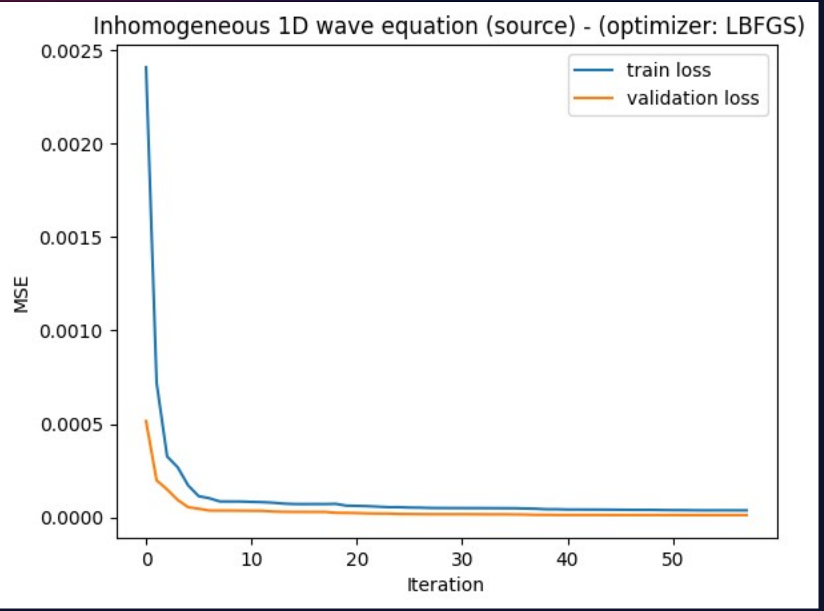
where: , and , with

subject to:

1. *Boundary Conditions (Dirichlet):*
2. *Initial Conditions:*

<u>Analytical Solution</u>:

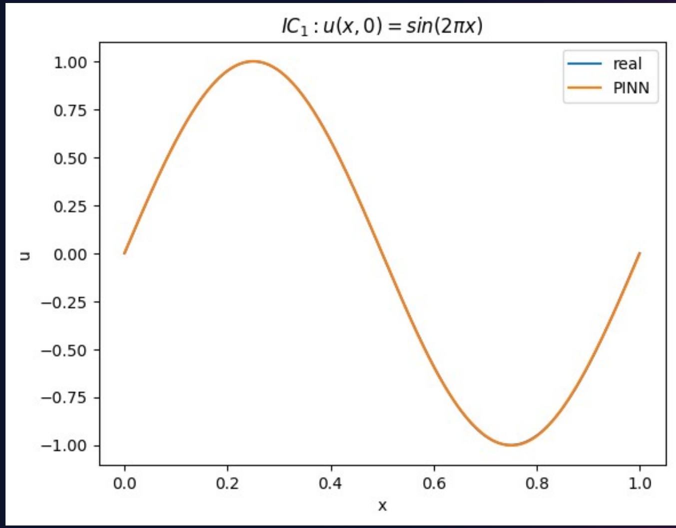
- L-BFGS optimizer
- Resampling ()
- Use of **Validation set** (points)
- Duration:
- SUCCESSFUL … and QUICK!

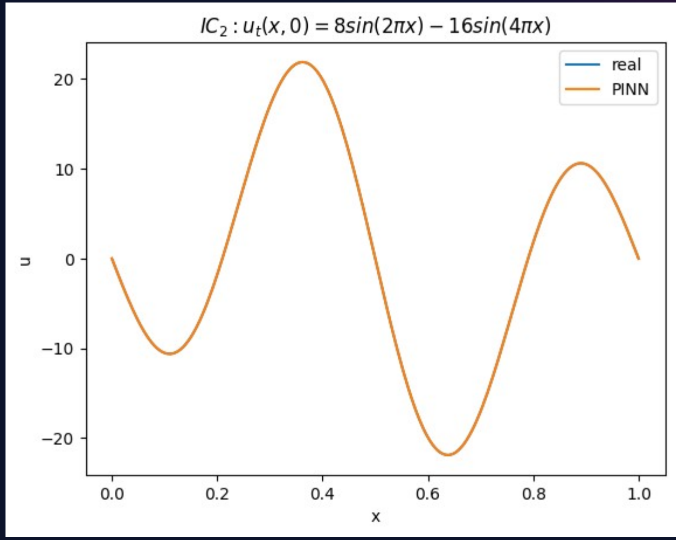


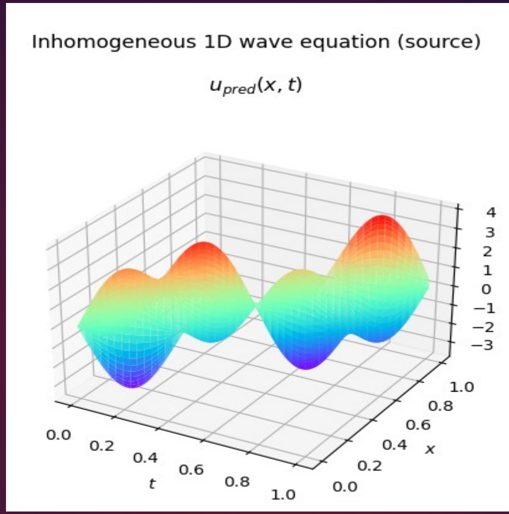
NOTE!

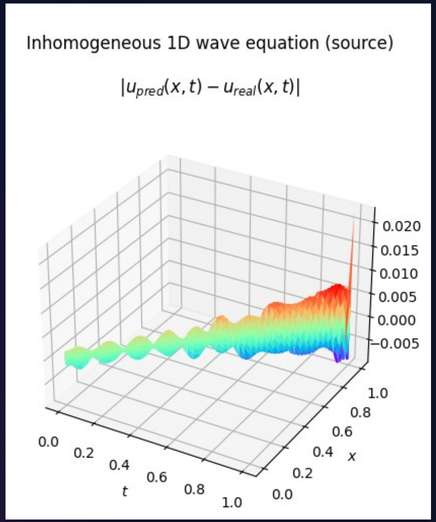
- Validation set NOT A MUST (in most cases) with PINNs
- It can make training even faster!

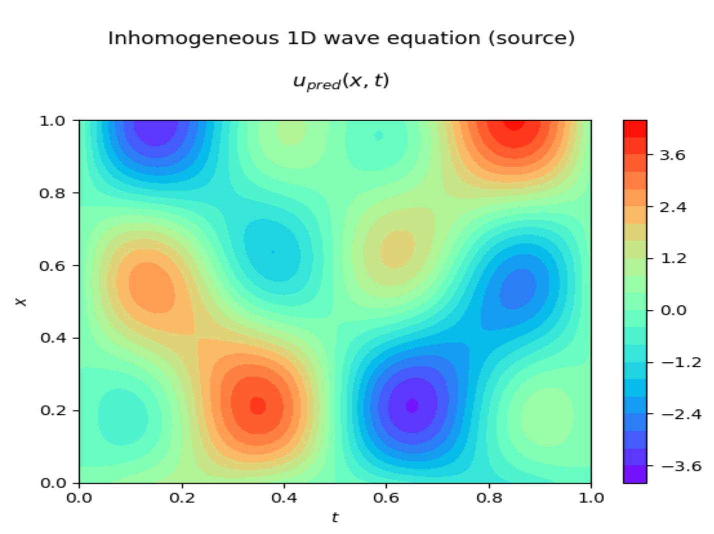
2. Inhomogeneous 1D wave equation (source)

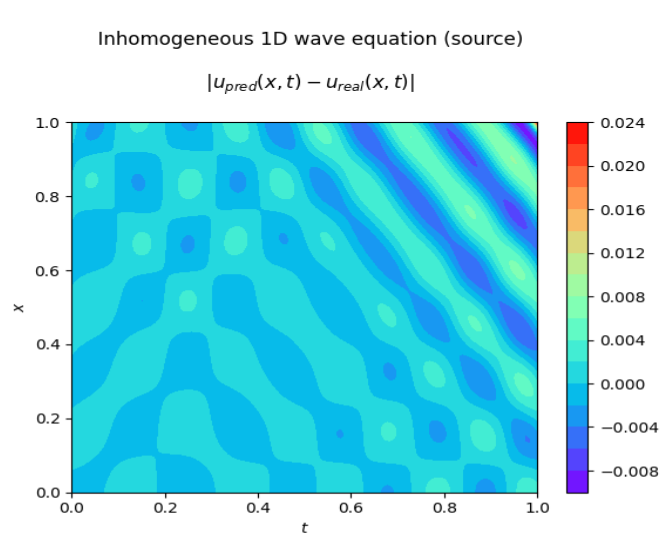












3. Homogeneous 2D wave equation

where: , and with

subject to:

Boundary Conditions (Dirichlet):

Analytical Solution:

a. Multiple Boundary Sampling

- Every epoch sample boundary points multiple times ()
- not too small no effect at all
- not too large too much focus on boundary points
- is a good choice
- … but equation NOT SOLVED!

b. Fourier Features Networks - FFNs

- Input data
- Choose
- Create
- Create

- Choose
- Adam + LBFGS
- … BUT …
- NOT SOLVED!

Initially…

- L-BFGS optimizer + Resampling
- UNSUCCESSFUL …
- Especially at boundary points ()

3. Homogeneous 1D wave equation

c. ReLoBRaLo
(Relative Loss Balancing with Random Lookback)

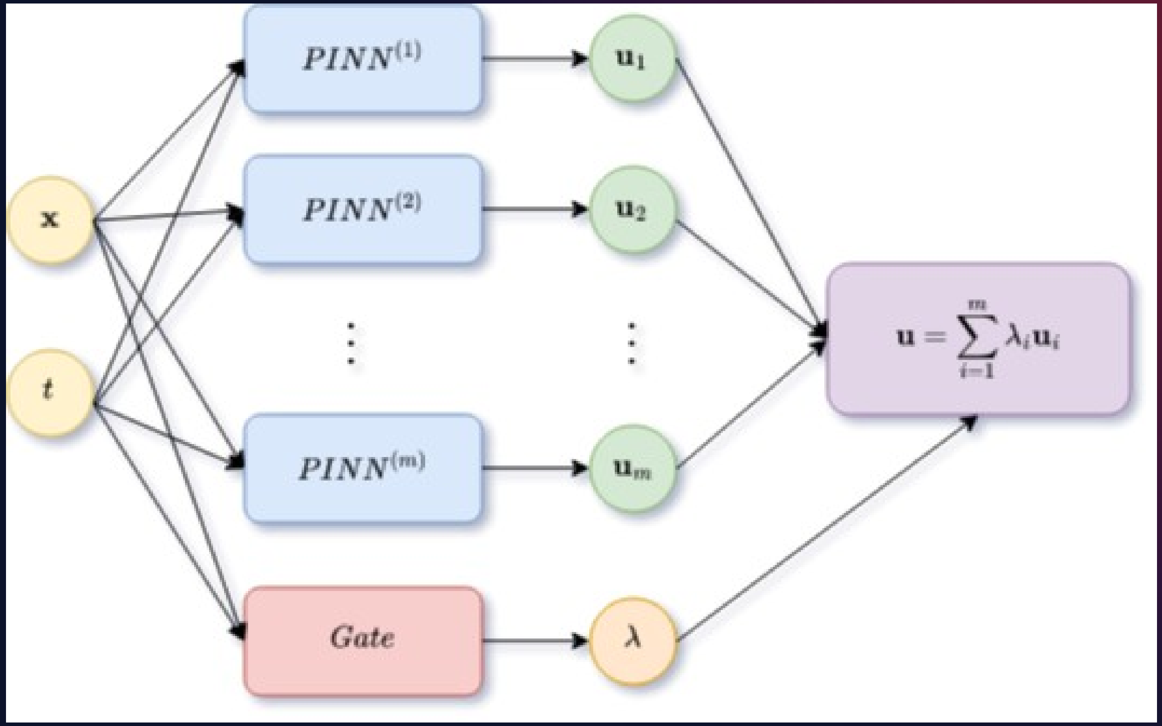
- **AIM**: dynamically balance coefficients of terms in Loss function
- **Equations**:
- **Hyper-parameters**
  1. : percentage of past losses to use
  2. how often to use ALL past losses
  3. regulates the «softness» of softmax
     - softmax returns uniform values
     - softmax argmax

- **ADVANTAGES**
  - No gradients calculation involved
  - Better than: *Learning Rate Annealing, GradNorm*, etc…
  - Very flexible many hyper-parameters
- **DISADVANTAGES**
  - No mathematical proof for convergence

- **2 experiments:**
  - Adam/LBFGS + Resampling
  - more quickly to better solution BUT problem NOT SOLVED

3. Homogeneous 1D wave equation

d. MoE-PINNs
(Mixture of Experts PINNs)



- **IDEA**: combine PINNs with different hyper-parameters (take advantage of all of them simultaneously)
- **AIM**: each PINN focuses on a different part of the problem
- **EXPERIMENT**: L-BFGS + Resampling
  - , different #layers, #units, activation functions
  - NOT SOLVED (quite large)
- **POSSIBLE REASON**: No PINN architecture inherently unable to solve it, even under complex combinations

3. Homogeneous 1D wave equation

SOLUTION…
function transformation!

- **AIM**: transform unknown function B.C. non-zero + no more complexity
- **HOW ???** careful observation of problem formulation
  - Intuitive! – No mathematical proof (does not always work!)
  - Inspired by the dual term

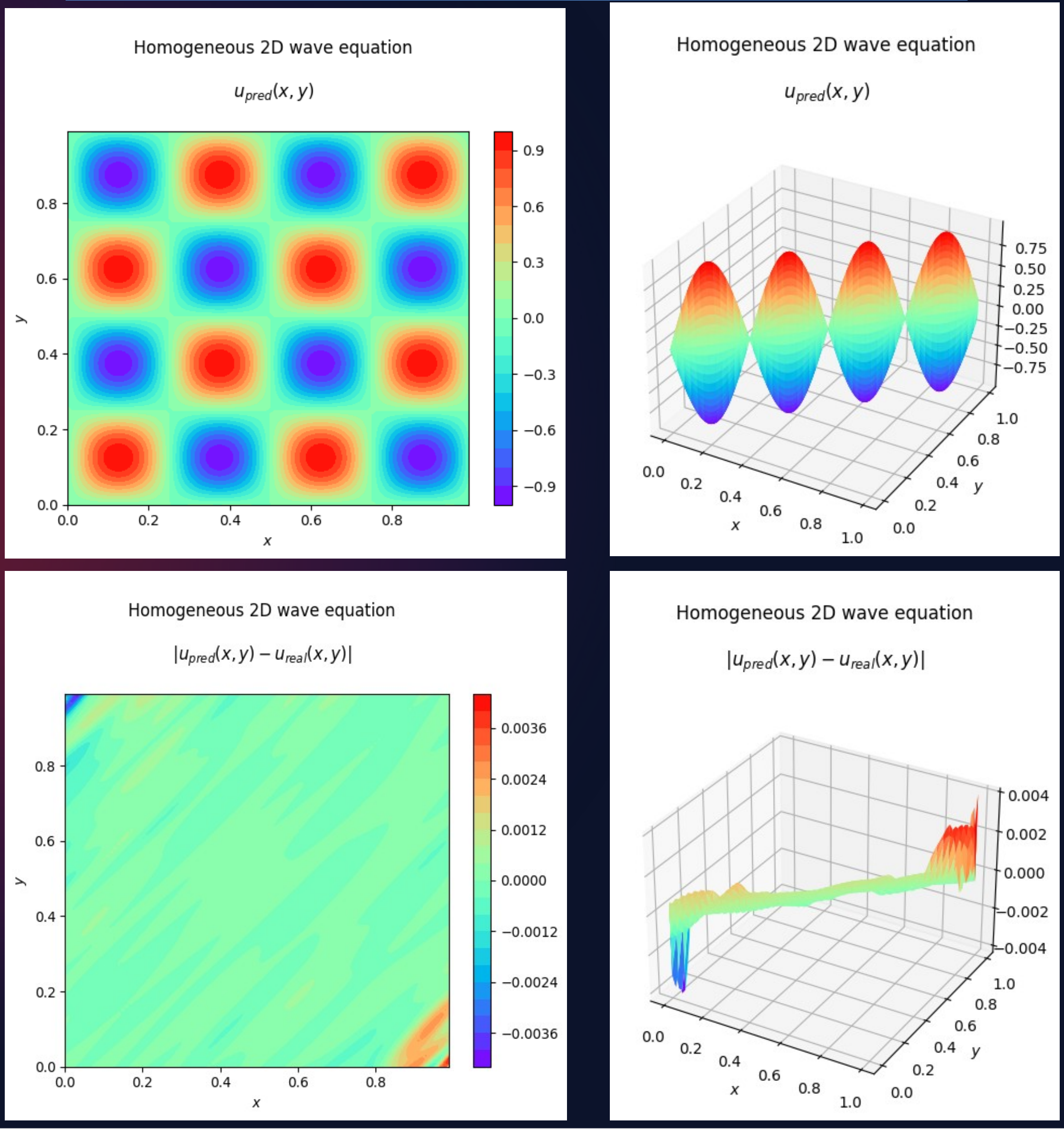
where: , and with

subject to:

B.Cs:

Analytical Solution:

Adam + L-BFGS (No resampling)



4. Homogeneous 2D Diffusion Equation

where: and with

subject to:

1. *Boundary Conditions (Dirichlet):*
2. *Initial Conditions:*

<u>Analytical Solution</u>:

Problem Transformation…

- Use of "**Method of separable variables**"
- (1)
- (I.C.):
  - and and (2)
  - Under the above conditions it can be proven that B.Cs. are also met

- Substitute (1) in: and use (2) …
- After many operations:
- For

Simplified Problem!

where: with

subject to:

1.

<u>Analytical Solution</u>:

- **ReLU** activation! here possible (only 1st order derivatives)
- Adam () + Resampling
- SOLVED!

Conclusions & Future Research

1. Advantages of PINNs

- solution at unknown points SIMPLE!
  - ONLY forward pass needed
  - No need for re-training
  - Significant decrease of inference time

2. Limitations of PINNs

- Limited **pre-processing** techniques
- **Dimensionality** increases exponentially with #independent variables
- **Activation functions**
  - only Tanh and Sin seem to work well in general
  - they encounter the problem of vanishing gradients
- Limited **optimizers**: only Adam and L-BFGS seem to work
- **Generalization** problem

3. Suggestions

- Focus on PINNs' drawbacks, NOT solving specific equations
- More efficient techniques for data sampling
- Novel NN architectures
- Reinforcement Learning for:
  - tuning hyperparameters
  - Loss function optimization
- Combinations of independent methods
- Find ways for Generalization

4. Conclusion

- PINNs deal with a difficult problem
- Research still in early stages & limited
- Future is promising!

Thank you !!!