Physics informed machine learning

With a focus on PINN

Himanshu Chaudhary[1]

[1]PMA
California Institute of Technology

Journal Club Talk
May 16th, 2022

Caltech

Table of Contents

Disclaimer

I do not work in this field.

Table of Contents

Caltech

What is Machine Learning?

Machine Learning

Roughly speaking algorithms that can analyze data to identify patterns and/or make predictions.

A very broad definition! There is a high chance that you are using something that can be called Machine learning.

What is Machine Learning?

Machine Learning

Roughly speaking algorithms that can analyze data to identify patterns and/or make predictions.

A very broad definition! There is a high chance that you are using something that can be called Machine learning.

The term itself is quite old, coined in 1959 by Arthur Samuel. Started out as a branch of AI research.

What is Machine Learning?

Machine Learning

Roughly speaking algorithms that can analyze data to identify patterns and/or make predictions.

A very broad definition! There is a high chance that you are using something that can be called Machine learning.

The term itself is quite old, coined in 1959 by Arthur Samuel. Started out as a branch of AI research.

Very closely related to optimization, curve fitting, model selection, classification etc..

Success of Neural Networks

Neural Networks became popular in the last decade.
Best performing algorithms in fields related to speech, NLP(Natural Language Processing), image analysis, recommender systems etc..

Reasons behind the success of Neural Networks

- Fast and efficient hardware.
- Large amount of data.
- Easy to use software*.

Hardware Lottery [1]

"when a research idea wins because it is suited to the available software and hardware and not because the idea is universally superior to alternative research directions"

Caltech

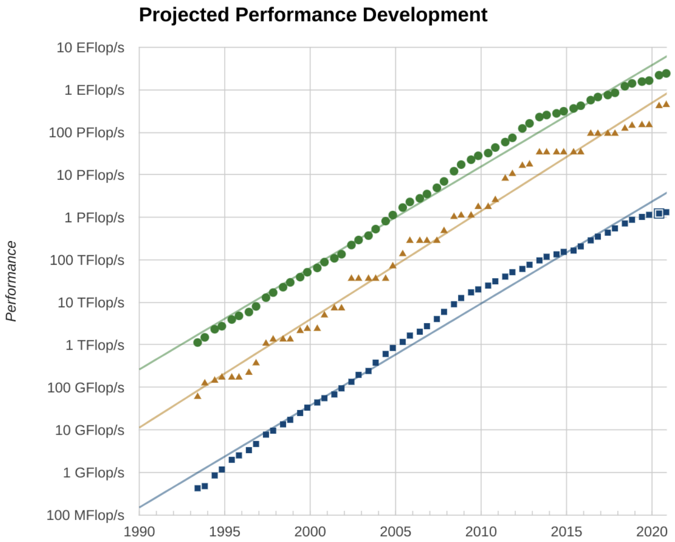
[1] https://hardwarelottery.github.io/

Nivdia H100 specs https://www.nvidia.com/en-us/data-center/h100/

Form Factor	H100 SXM
FP64	30 teraFLOPS
FP64 Tensor Core	60 teraFLOPS
FP32	60 teraFLOPS
TF32 Tensor Core	1,000 teraFLOPS* | 500 teraFLOPS
BFLOAT16 Tensor Core	2,000 teraFLOPS* | 1,000 teraFLOPS
FP16 Tensor Core	2,000 teraFLOPS* | 1,000 teraFLOPS
FP8 Tensor Core	4,000 teraFLOPS* | 2,000 teraFLOPS
INT8 Tensor Core	4,000 TOPS* | 2,000 TOPS

Caltech

Top 500 supercomputers https://www.top500.org/statistics/perfdevel/



Caltech

Table of Contents

Caltech

What is a Neural Network?

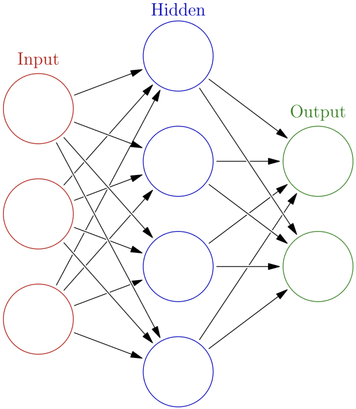


Figure: A simple feed forward neural network (By Glosser.ca - Own work, Derivative of File:Artificial neural network.svg, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=24913461)

The arrows have numeric values associated with them called weights w^l_{jk}.

The circles are called nodes and their value(a^l_j) is calculated using

$$a^l_j = \sum_k w^l_{jk} a^{l-1}_k + b^l_j \quad .$$

An appropriate cost function that tells us how close are we to the desired result.

l: layer number; j,k: node numbers; : activation function

Training a Neural Network

Find the values of w and b that give the desired output. Basically, a optimization problem which can be solved very efficiently on GPUs.

Different type of Neural Networks in use

The "simple" feed forward neural network is hardly ever used in the production settings.

Different type of Neural Networks in use

- The "simple" feed forward neural network is hardly ever used in the production settings.
- In image related tasks CNN(convolution neural networks) and its derivatives perform the best.
  - Good at dealing with spatial dependencies.
  - Also called Shift Invariant or Space Invariant Artificial Neural Networks .

Different type of Neural Networks in use

The "simple" feed forward neural network is hardly ever used in the production settings.

In image related tasks CNN(convolution neural networks) and its derivatives perform the best.

- Good at dealing with spatial dependencies.
- Also called Shift Invariant or Space Invariant Artificial Neural Networks .

In speech and language related tasks RNN(recurrent neural network) and its derivatives perform the best.

- Good at dealing with temporal dependencies.
- Have an internal state(memory) giving them the capacity to deal with temporal dependencies and analyze variable length data.

Different type of Neural Networks in use

- The "simple" feed forward neural network is hardly ever used in the production settings.
- In image related tasks CNN(convolution neural networks) and its derivatives perform the best.
  - Good at dealing with spatial dependencies.
  - Also called Shift Invariant or Space Invariant Artificial Neural Networks .
- In speech and language related tasks RNN(recurrent neural network) and its derivatives perform the best.
  - Good at dealing with temporal dependencies.
  - Have an internal state(memory) giving them the capacity to deal with temporal dependencies and analyze variable length data.

Crux of the story

Not enough to throw data at a NN and hope for the best. Most of the successful use cases of NN depend on these specialized architectures.

Table of Contents

Caltech

Using Machine learning in Physics

Maybe you already are!!

Using Machine learning in Physics

Maybe you already are!!

As a glorified interpolator. Maybe good for noisy, high dimensional data??

Using Machine learning in Physics

Maybe you already are!!

As a glorified interpolator. Maybe good for noisy, high dimensional data??

For data analysis. Classification clustering type stuff.

Using Machine learning in Physics

Maybe you already are!!

As a glorified interpolator. Maybe good for noisy, high dimensional data??

For data analysis. Classification clustering type stuff.

Solving differential equations.

Using Machine learning in Physics

Maybe you already are!!

As a glorified interpolator. Maybe good for noisy, high dimensional data??

For data analysis. Classification clustering type stuff.

Solving differential equations.

System identification, Reduced order modeling.

Why do we want to use NNs for solving differential equations? [4]

Noisy data can not be easily incorporated in the current algorithms.

Caltech

[4] Ref: Physics-informed machine learning, `https://doi.org/10.1038/s42254-021-00314-5`

Why do we want to use NNs for solving differential equations? [4]

Noisy data can not be easily incorporated in the current algorithms.

- Can you use RK4 if your initial data is noise?

[4] Ref: Physics-informed machine learning, https://doi.org/10.1038/s42254-021-00314-5

Why do we want to use NNs for solving differential equations? [4]

Noisy data can not be easily incorporated in the current algorithms.

- Can you use RK4 if your initial data is noise?
- What if your boundary data comes from different source and is again noisy?

Caltech

[4] Ref: Physics-informed machine learning, `https://doi.org/10.1038/s42254-021-00314-5`

Why do we want to use NNs for solving differential equations? [4]

Noisy data can not be easily incorporated in the current algorithms.

- Can you use RK4 if your initial data is noise?
- What if your boundary data comes from different source and is again noisy?

ML models are more robust to incomplete data and outliers. Once trained ML models can deal with outliers much more easily.

[4] Ref: Physics-informed machine learning, https://doi.org/10.1038/s42254-021-00314-5

Why do we want to use NNs for solving differential equations? [4]

Noisy data can not be easily incorporated in the current algorithms.

- Can you use RK4 if your initial data is noise?
- What if your boundary data comes from different source and is again noisy?

ML models are more robust to incomplete data and outliers. Once trained ML models can deal with outliers much more easily.

No complicated mesh generation required!

Caltech

[4] Ref: Physics-informed machine learning, https://doi.org/10.1038/s42254-021-00314-5

Why do we want to use NNs for solving differential equations? [4]

- Noisy data can not be easily incorporated in the current algorithms.
  - Can you use RK4 if your initial data is noise?
  - What if your boundary data comes from different source and is again noisy?
- ML models are more robust to incomplete data and outliers. Once trained ML models can deal with outliers much more easily.
- No complicated mesh generation required!
- NNs can overcome the curse of dimensionality.

Caltech

[4] Ref: Physics-informed machine learning, `https://doi.org/10.1038/s42254-021-00314-5`

Why do we want to use NNs for solving differential equations? [4]

Noisy data can not be easily incorporated in the current algorithms.

- Can you use RK4 if your initial data is noise?
- What if your boundary data comes from different source and is again noisy?

ML models are more robust to incomplete data and outliers. Once trained ML models can deal with outliers much more easily.

No complicated mesh generation required!

NNs can overcome the curse of dimensionality.

Once trained NN can be very fast in making predictions.

Caltech

[4] Ref: Physics-informed machine learning, https://doi.org/10.1038/s42254-021-00314-5

Table of Contents

Caltech

How to solve differential equations using a NN a.k.a. PINN.[5]

We want to solve Burgers equations with appropriate boundary conditions.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}$$

Rough idea is that we want to approximate $u(t, x)$ using a NN with a loss functions that forces the solution to satisfy the differential equation and boundary conditions.

$$L = w_{\text{data}}\, L_{\text{data}} + w_{\mathrm{PDE}} L_{\mathrm{PDE}},$$

where

$$L_{\text{data}} = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \left(u\left(x_i, t_i\right) - u_i\right)^2 \quad \text{and}$$

$$L_{\mathrm{PDE}} = \frac{1}{N_{\mathrm{PDE}}} \sum_{j=1}^{N_{\mathrm{PDE}}} \left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2} \right)^2_{(x_j, t_j)}$$

[5]Ref: Physics-informed machine learning, `https://doi.org/10.1038/s42254-021-00314-5`

How to solve differential equations using a NN a.k.a. PINN.[6]

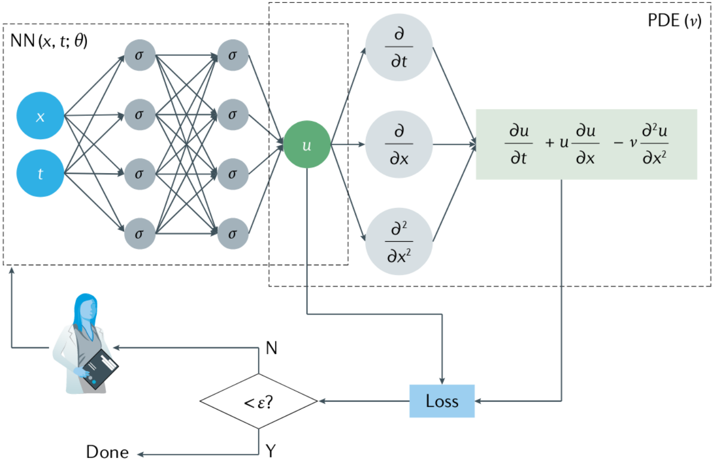


Figure: Physics Informed Neural Network(PINN)

PINNs in action

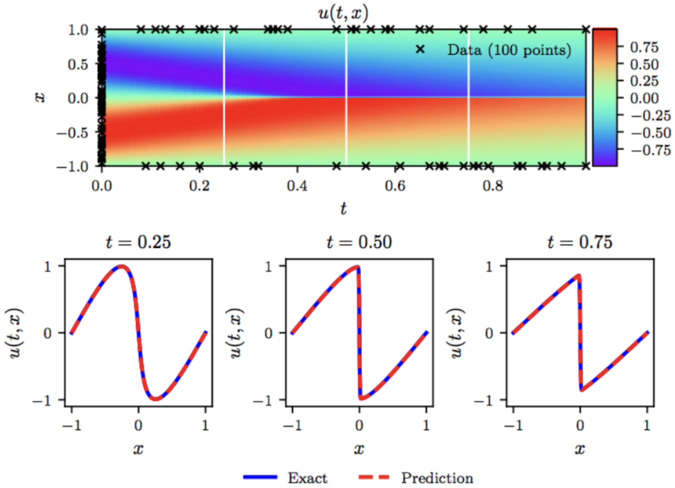


Figure: Solution of burgers equation using PINN https://maziarraissi.github.io/PINNs/

Caltech

PINNs in action

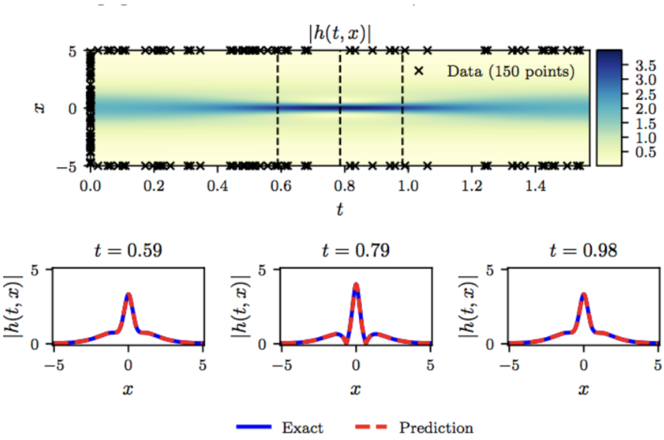


Figure: Solution of Shrödinger equation using PINN https://maziarraissi.github.io/PINNs/

Table of Contents

Three ways we can make machines learn physics[7]

Making the training of NNs simpler.

- Learning bias, already saw an example.
- Observational bias
- Inductive bias

In practice a combination of these is used.

Caltech

[7]Ref: Physics-informed machine learning, `https://doi.org/10.1038/s42254-021-00314-5`

Observational bias

Observational bias

Augment your data with symmetries about what to learn.

New auxiliary variables.

$f(x; y) = x^y + x$, then make x^y a new variable and provide as input.

Observational bias

Observational bias

Augment your data with symmetries about what to learn.

New auxiliary variables.

$f(x; y) = x^y + x$, then make x^y a new variable and provide as input.

Scale, rotate, translate etc. in a way that reflects the underlying physics, instead of making the NN learn these.

Observational bias

Observational bias

Augment your data with symmetries about what to learn.

New auxiliary variables.

$f(x; y) = x^y + x$, then make x^y a new variable and provide as input.

Scale, rotate, translate etc. in a way that reflects the underlying physics, instead of making the NN learn these.

Take Fourier transform if your data is periodic.

Inductive bias

Inductive bias

Embed physics in your NN architecture, so that the whole NN exactly satisfies the desired physical laws.

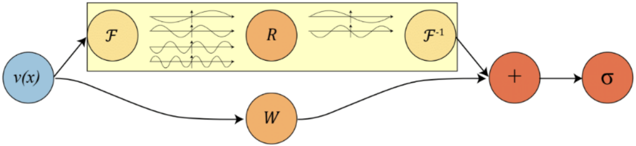


Figure: Fourier Layer `https://zongyi-li.github.io/blog/2020/fourier-pde/`

Need to do on per problem basis. Hard to design architecture that satisfy non-trivial laws.
May not be possible to design architectures that scale well!

Table of Contents

Caltech

Why are we not using PINNs to solve everything?

NN have trouble in learning high frequency functions. They might not converge at all!

Why are we not using PINNs to solve everything?

NN have trouble in learning high frequency functions. They might not converge at all!

Solution accuracy is bad. This may be fine in some cases but not for typical numerical simulations.

Why are we not using PINNs to solve everything?

NN have trouble in learning high frequency functions. They might not converge at all!

Solution accuracy is bad. This may be fine in some cases but not for typical numerical simulations.

No bound on errors! We have no idea how accurate a given prediction is.

Why are we not using PINNs to solve everything?

NN have trouble in learning high frequency functions. They might not converge at all!

Solution accuracy is bad. This may be fine in some cases but not for typical numerical simulations.

No bound on errors! We have no idea how accurate a given prediction is.

Different problems may require different architectures.

Why are we not using PINNs to solve everything?

NN have trouble in learning high frequency functions. They might not converge at all!

Solution accuracy is bad. This may be fine in some cases but not for typical numerical simulations.

No bound on errors! We have no idea how accurate a given prediction is.

Different problems may require different architectures.

Most of the time a complete black box.

Hype Hype Hype?

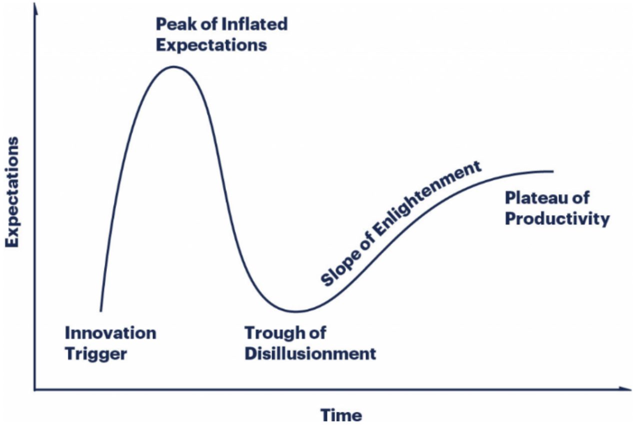


Figure: Gartner Hype Cycle [8]

[8] https://www.bmc.com/blogs/gartner-hype-cycle/

Thanks