Imperial College London

Machine Learning models for more Green, Sustainable, Clean Renewable Energy

Application to Wave Energy Convertors (WECs)

Angela Maria Castañeda Oviedo

Under the Wave off Kanagawa*

Renewable Energy

- Wave Energy
- Tidal Energy
- Thermal Energy

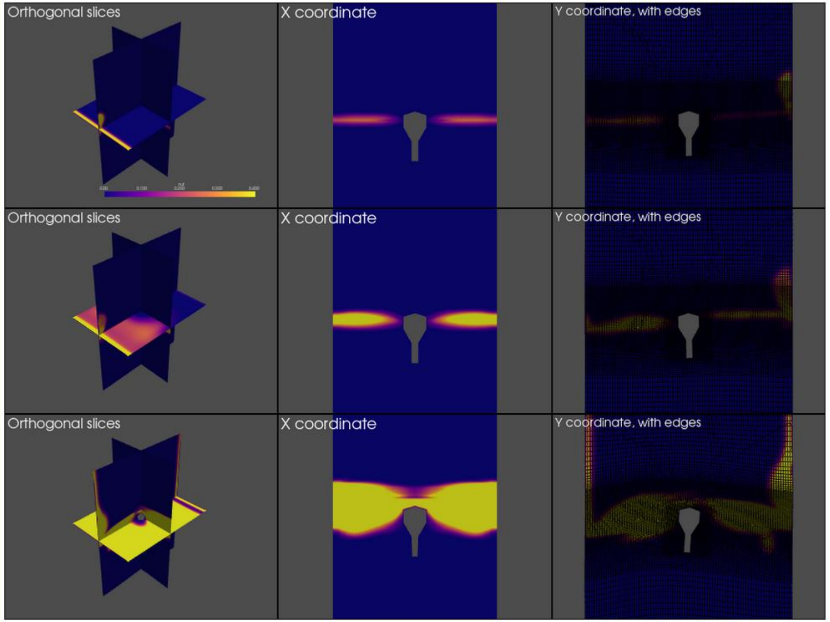
Wave Energy Converters (WECs)

Imperial College London

WavE-Suite project

Task: Discover numerical tools for assessing the survivability of WECs under extreme marine conditions.

- **mini-task:** apply Machine Learning Methods to the CFD simulations produced so far (WaveSuite dataset) .



WaveSuite dataset samples

WavE-Suite project

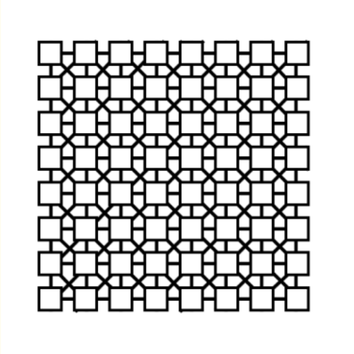
WaveSuite Dataset:

- Feature of interest: dynamic viscosity
- Entries are graphs
- 850 000 nodes per graph
- Same adjacency matrix

AutoEncoders

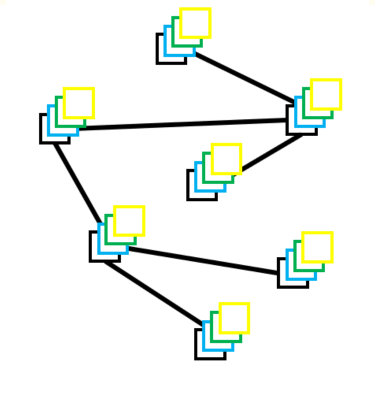
- Graph Convolutional Networks (GCN)
- Different combinations of activation functions

GCN: CNN vs GCN



Convolutional Networks (CNN)

Convolutional layer (or CNN layer)



Graph Convolutional Networks (GCN)

Graph Convolutional layer (or GCN layers)

$$H^{(k)} = AH^{(k-1)}W_{neigh}^{(k)} + H^{(k-1)}W_{self}^{(k)}$$ *

- $H^{(k)}$ is the feature matrix at step k.
- A is the adjacency matrix
- W matrices are the weights

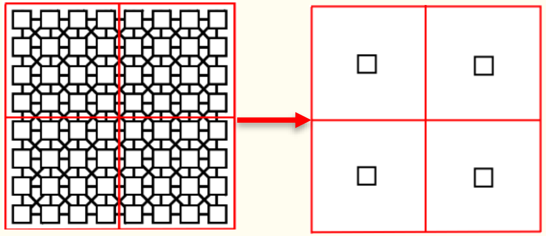
* William L. Hamilton. Graph representation learning. chapter 5. Synthesis Lectures on Artificial Intelligence and Machine Learning, 14(3):1–159.

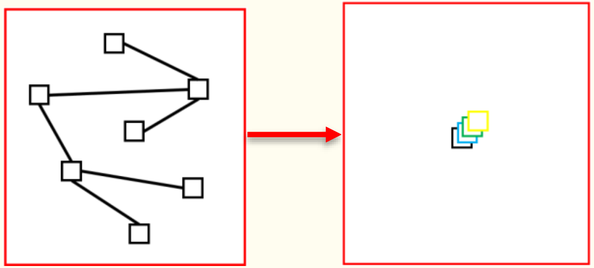
Other ways to generalize CNN to graphs

- Graph Convolutional layers (GCN)
- Graph Attention Networks (GAT)
- Message Passing Neural Network (MPNN)

$$^{*}H^{(k)} = AH^{(k-1)}W^{(k)}_{neigh} + H^{(k-1)}W^{(k)}_{self}$$

…pooling





* William L. Hamilton. Graph representation learning. chapter 5. Synthesis Lectures on Artificial Intelligence and Machine Learning, 14(3):1–159.

Graph AutoEncoders

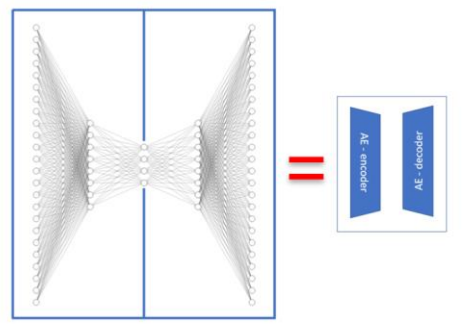
Graphs can have:

- node features
- adjacency matrix
- different number of nodes
- edges features.

WaveSuite graphs have the same adjacency matrix

- Instead of predicting adjacency matrix, keep a copy of one entry
- Predict node features instead

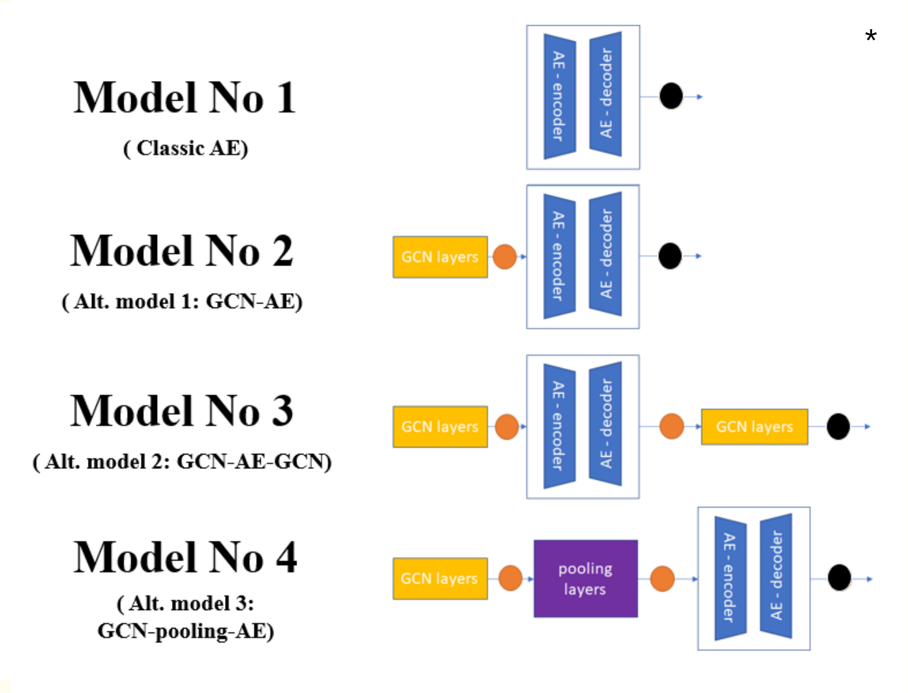
Proposed AutoEncoders



Classic AutoEncoder Module

Principles:

- Use powers of 2
- Decoder should be a **mirror** of the Encoder



Proposed AutoEncoder Designs

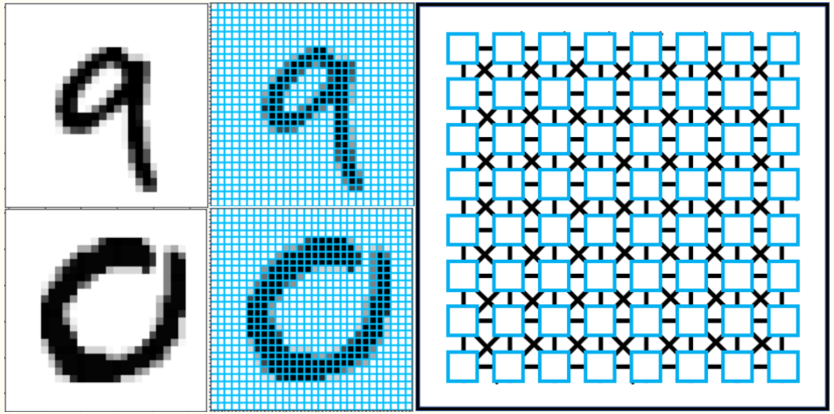
* None means No activation function at all

Smaller Datasets

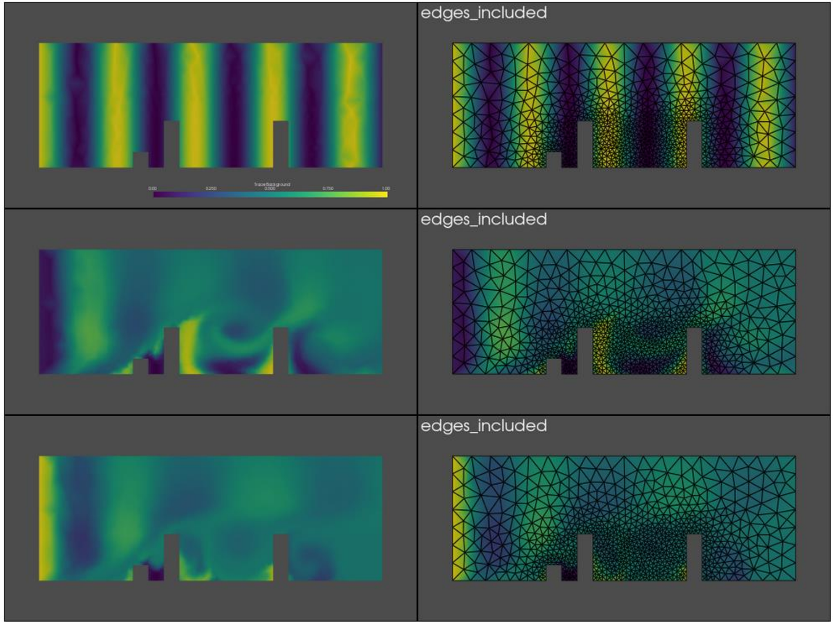
- WaveSuite dataset requires additional coding.
- It is not possible to generalize the behaviour of an AutoEncoder in one dataset to another dataset.

- Use smaller datasets to gain knowledge on how to design better AutoEncoders
- Apply that knowledge to the final AutoEncoder design

Smaller Datasets

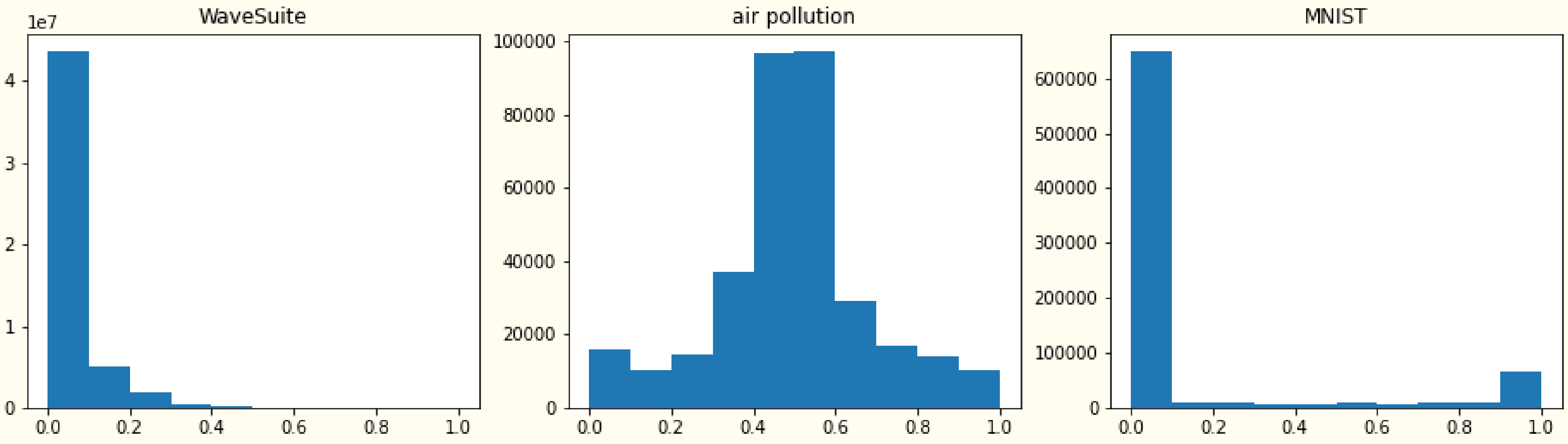


MNIST samples



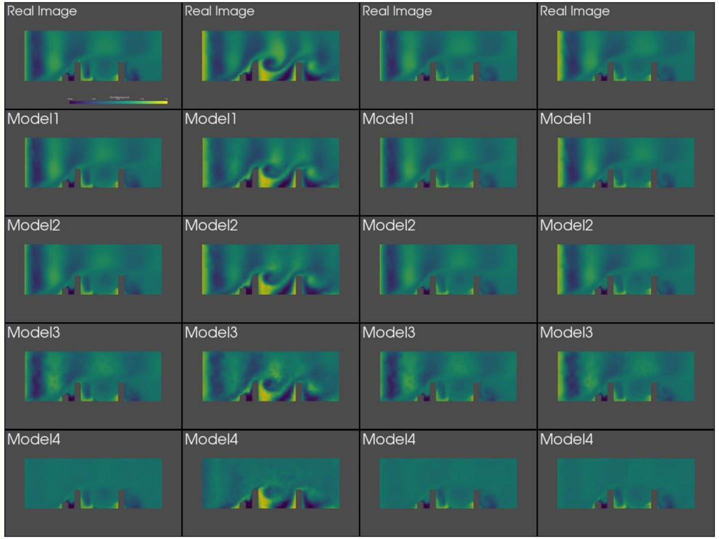
air-pollution samples

Smaller Datasets



Datasets Distributions

Smaller Datasets: Results

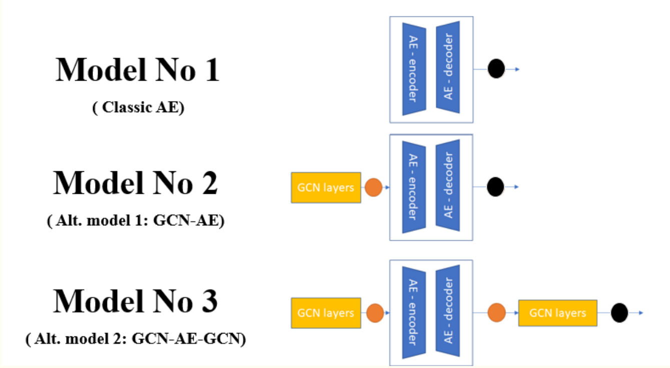


air-pollution results – Sigmoid function

MNIST results – Sigmoid function

Smaller Datasets: Results and Discussion

The activation functions at the output are very important, whereas the others not so much.



Good activation function → Good results no matter the model.

Best activation functions:

- air-pollution: Sigmoid
- MNIST: None *

Possible cause: datasets distributions?

* None is slightly better than Sigmoid, but Sigmoid also produces very good results

Smaller Datasets: Discussion

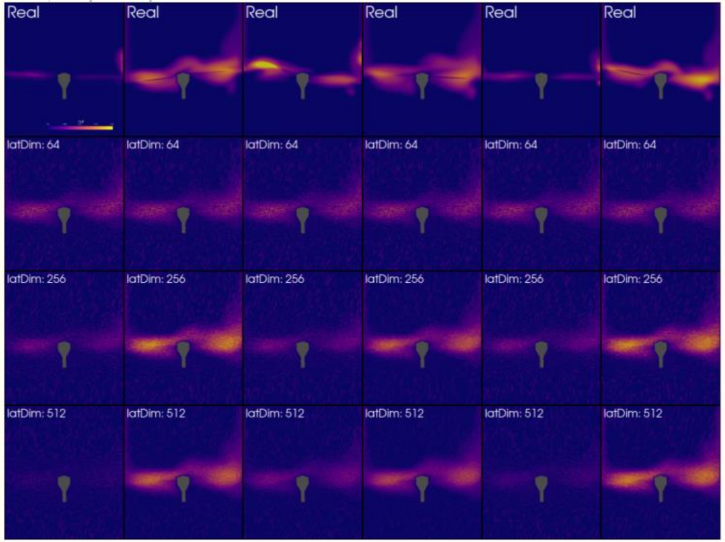
- Before trying to incorporate GCN layers, it is important to identify the best activation function at the output, for a particular dataset.

- ADAM optimizer is better than SGD

Hyperparameters:

- Latent space dimension
- Embedding sequence of the GCN layers (number of channels)
- Batch size

WaveSuite Dataset: Results



Model 1 with NO activation function at the output using different latent space dimensions

Model 1, NO act-func, 10 epochs:

- 16 points: 0.008728 MSE, 99 s
- 32 points: 0.008728 MSE, 116 s
- 64 points: 0.002214 MSE, 153 s
- 128 points: 0.004290 MSE, 217 s
- 256 points: 0.001160 MSE, 413 s
- 512 points: 0.000819 MSE, 3145 s.

WaveSuite Dataset: Results and Discussion

Latent space 256 points, 10 epochs

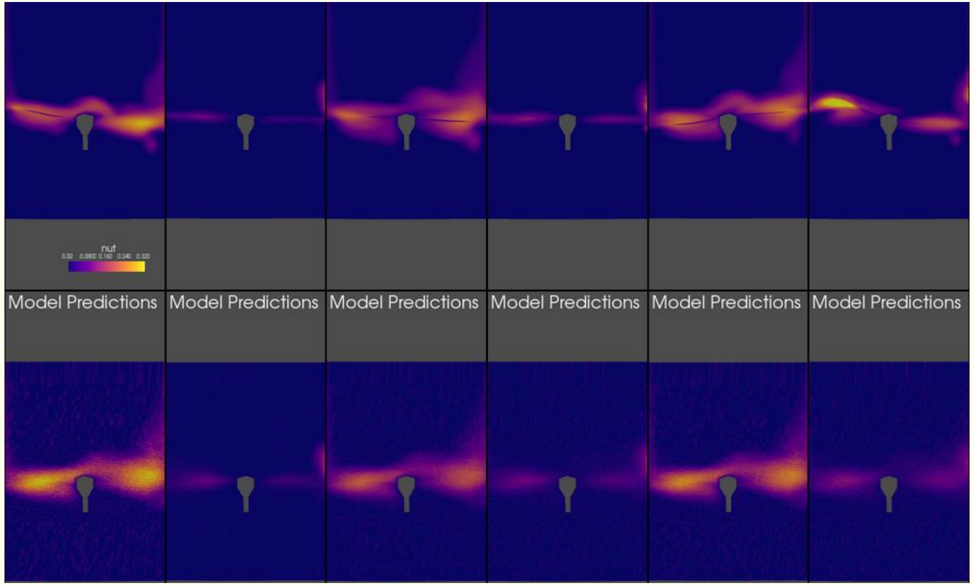
- Model 1: 0.00116 MSE, 413 s
- Model 2: 0.00278 MSE, 6526 s.
- Model 3: 0.000586 MSE, 12345 s.

Model 1, 256 latent space dimension:

- 10 epochs: 0.001160 MSE, 413 s
- 20 epochs: 0.000663 MSE, 800 s.
- 50 epochs: 0.000798 MSE, 2008 s.

- Extra hidden layer, 10 epochs: 0.000755 MSE, 2939 s.

WaveSuite Dataset: Results and Discussion



Model 1, 256 latent space dimension, using 50 epochs

Conclusions

- ADAM optimizer is better than SGD
- Finding the right dimension for the latent space is important.
- The activation function at the output is important.
- Adding GCN-layers to the AutoEncoders improves the results, but there is a tradeoff on execution time and computational resources.
- This project can be expanded to a Variational AutoEncoder to produce more samples for the WaveSuite dataset.

Thank You,

Rossella, Cesar,
Lluis, Marijan and Yves,

And anyone that helped me during this project.

Suggested Questions

- Why do you think the best activation function at the output of the AutoEncoder is related to the distribution of the dataset?

- Can you mention something about how you implemented the code?
- Why did you had to do more coding to make WaveSuite work compared to the smaller datasets?

Suggested Questions

Angela, can you…

- explain now the formula of GCN?
- explain now the other ways to generalize CNN to graphs? Do you think they can improve the results even more?
- tell us how many GCN layers are you using in each GCN module and why?
- comment something on the number of channels used in the GCN layers?