# Wids Endterm Report

Rucha Ranade (24B1530)

December 2025

## 0.1 Week 1: PINNs

PINNs stand for physics involved neural networks. This is applied when the available dataset is less in proportion and it is hard to train a model accurately on that dataset. In such cases a term related to physics is also considered while defining the loss function. This term is nothing but the differentiation equation for the given variable. Such consideration in loss function enables us to train a model with more precision on smaller proportion of data.

### 0.1.1 Exponential Decay

- For training exponential decay PINN, the equation is:

$$y(x) = e^{-x}.$$

- Taking the derivative with respect to $x$, we obtain:

$$\frac{d}{dx}\left(e^{-x}\right) = -e^{-x}.$$

- Substituting into the governing equation gives

$$\frac{d}{dx}\left(e^{-x}\right) + e^{-x} = 0.$$

- Thus, the problem reduces to the first-order ordinary differential equation

$$\frac{dy}{dx} + y = 0,$$

  with the initial condition

$$y(0) = 1.$$

- In the Physics-Informed Neural Network (PINN) framework, the physics residual is defined as

$$\mathcal{R}(x) = \frac{dy}{dx} + y.$$

- The model consisted of five fully connected (dense) layers with `tanh` activation functions, and was trained using the Adam optimizer with an adaptive learning rate.
- The network was trained for 200 epochs, during which all trainable parameters were optimized.

### 0.1.2 SHM

- The equation for SHM is:

$$a = -w^2 A sinwt$$

- This equation in differential form can be written as:

$$\frac{d^2y}{(dt)^2} = -w^2 y$$

- The boundary conditions were specified as:

$$(x, y) = (0, 0) \quad , \quad \left(\frac{\pi}{2}, 1\right) and \quad (\pi, 0),$$

  with the angular frequency chosen as:

$$\omega = 1\ rad/s.$$

- In the Physics-Informed Neural Network (PINN) framework, the physics residual was defined as:

$$\mathcal{R}(x) = \frac{d^2y}{(dt)^2} + w^2 y$$

- Again for this task also Adam optimizer was used and trainable weights were optimized over 3000 epochs and physics loss and boundary condition loss both were considered.

Colab link for Week 1: `https://colab.research.google.com/drive/1zdMyQvXM7XhL_shmOThjGgHOHZ-qKHDX?usp=sharing`

## 0.2 Week 2: SoH(Pybamm)

State of Health (SoH) is a measure of the overall condition of a battery and indicates its ability to store and deliver energy relative to its original capacity. It reflects battery degradation due to aging, cycling, and operating conditions. The C-rate of a battery represents the rate at which it is charged or discharged and is inversely related to the time required for full discharge; for example, a 1C rate corresponds to a discharge time of one hour. In this work, the Single Particle Model (SPM) is employed to analyze battery behavior, as it provides a simplified yet effective electrochemical representation of lithium-ion batteries by modeling each electrode as a single spherical particle. The SPM enables efficient estimation of battery performance and health-related parameters while maintaining reasonable computational complexity.

### 0.2.1 Li-Concentration

- Used PyBaMM (Python Battery Mathematical Modelling) and implemented Single particle model on lithium ion
- Simulated the performance for as hour and plotted concentration of negative particle with repect to radius of cell
- Concentration is dependent on variables like time,radial distance and x-axis, so for a particular time mean with respect to x axis was taken and then plotted
- Timings considered in second were:
$$(0, 900, 1800, 2700, 3600)$$

### 0.2.2 OCV vs SOC

- The Single Particle Model (SPM) for a lithium-ion battery was employed and simulated using PyBaMM with its default parameter set.
- The battery behavior was simulated over the full state-of-charge (SOC) range from 0% to 100%, and the corresponding battery open-circuit voltage (OCV) in volts was obtained by solving the model.
- The OCV–SOC curve generated from the PyBaMM simulation was approximated using a fifth-degree polynomial by applying the `NumPy polyfit` function.

Colab Link for Week 2: `https://colab.research.google.com/drive/1NCOi2WyHIuYQ2f22gXSpqR7e4vNnyrWc?usp=sharing`

## 0.3 Week 3: SoH prediction by physics constraints

Coulomb counting is a method to estimate the state of charge by integrating the measured battery current over time. It relies on the principle of charge conservation, making it a simple and physically consistent approach. In physics-informed models, it is used as a constraint to ensure realistic SOC evolution during discharge.

### 0.3.1 Process of training model

- The same NASA battery dataset was used, and discharge cycles were preprocessed to compute State of Charge (SOC) and State of Health (SoH) values.
- Coulomb counting was applied using measured current and time intervals to estimate the theoretical SOC evolution during each discharge cycle.
- A physics loss term was constructed by penalizing deviations between the measured SOC and the Coulomb-counted SOC, enforcing charge conservation.
- An additional voltage-based constraint was introduced by enforcing monotonic decrease of discharge voltage with time.
- The physics loss was computed using the resampled high-resolution physics dataset, while the main dataset was used for supervised learning.
- A Transformer encoder was trained on binned cycle data to predict cycle-level SoH values.
- The total training loss was defined as a weighted sum of data loss (MSE between predicted and true SoH) and physics loss.
- The inclusion of physics-based constraints guided the model toward physically consistent predictions and reduced overfitting.
- Incorporating the physics loss resulted in approximately 72% reduction in training loss, demonstrating improved convergence and stability.

*Python file uploaded on Github

## 0.4 Week 4: SoH prediction with SPM model

Transformer: A Transformer is a neural network architecture that uses self-attention to learn relationships across an entire input sequence at once, allowing it to model long-range dependencies without relying on recurrence.
Backpropagation: Backpropagation is the training process where the prediction error is propagated backward through the network to compute gradients, which are then used to update model parameters and minimize the loss.

### 0.4.1 Process followed to train model

- The NASA lithium-ion battery dataset from Kaggle was used, and incompatible battery types were removed to ensure uniform discharge behavior during training.
- Raw cycle data was preprocessed and feature engineering was performed to compute State of Charge (SOC) and State of Health (SoH) values for each discharge cycle.
- The processed data was separated into a main dataset for machine learning training and a physics dataset for enforcing physical constraints.
- Each discharge cycle was divided into 20 time bins, forming a sequential representation of the full cycle.
- A Transformer encoder was used to model the entire discharge cycle at once and predict a single SoH value per cycle.
- The Single Particle Model (SPM) was implemented using PyBaMM to generate physics-based voltage trajectories for the discharge cycles.
- A physics loss was computed by comparing measured voltage with scaled SPM-predicted voltage, where the scaling factor $\alpha$ is a learnable parameter accounting for aging and modeling mismatch.
- The Transformer model and $\alpha$ parameter were jointly optimized using backpropagation by minimizing a combined loss consisting of data loss and weighted physics loss.

*Python file uploaded on Github.

### 0.4.2 Comparison between model developed by physics constraints and by SPM model

- The physics-constraint model enforces general physical laws directly from measured data, whereas the PyBaMM-SPM approach uses a mechanistic electrochemical model to generate physics-based voltage predictions.
- The Coulomb counting and voltage constraint method is computationally lightweight, while the PyBaMM-SPM model is more computationally expensive but provides deeper electrochemical insight.
- The PyBaMM-SPM approach improves physical fidelity through model-based voltage estimation, whereas the constraint-based model improves learning efficiency by enforcing fundamental conservation principles.