SIG Math

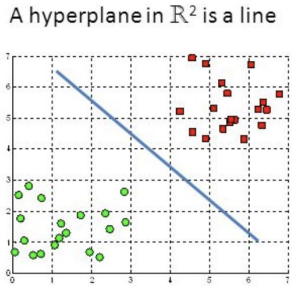
Physics-Informed Neural Networks

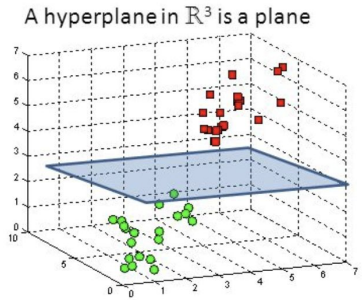
White papers

https://doi.org/10.1016/j.jcp.2018.10.045

Perceptron

Mathematically, it is a hyperplane in space along with a normal vector that that helps to decide points in space





“Easy” for computer to compute and understand since as function it can be encoded as

$$\mathbf{L}(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$$

Neural Networks

While there are many Neural Networks architecture out there, we will mainly focus on the Multi-Layered Perceptron (MLE) network

- Essentially, this is a network of perceptrons that comes together to classify points in space

Q: Given a *discrete* set of data in the plane. How can you pickout a circle using only a set of hyperplanes?

MLE

Rationale: A series of well-placed perceptrons can be use to decide (or approximate) data, geometries, functions,...

Neural Network training is essentially tuning the placement of the perceptrons via the updating the perceptrons weights and biases.

The training is an iterative guess-and-check:

- Make initial guess
- Compute how off we are (the loss)
- Update the weights based on this loss to have a better educated guess (backpropagation)
- Repeat until the loss is minimized.

Differential Equations

A (partial) differential equations is an algebraic equation relating a differentiable function $u(x_1, \dots x_n)$ with its derivatives

$$F\left(u, \frac{\partial u}{\partial x_1}, \dots, \frac{\partial^{k_1} u}{\partial x_1^{k_1}}, \cdots \frac{\partial u}{\partial x_n}, \dots, \frac{\partial^{k_n} u}{\partial x_n^{k_n}}\right) = 0$$

Often times, these will not have known closed-form or analytic solutions.

Differential Equations

Even if a differential equation is well-posed on a region $D \subseteq \mathbb{R}^n$ it does not mean that we can obtain a unique solution. You can see this with just simple calculus.

To make sure that we have a unique solution, we will need to impose condition

1. On the Boundaries (Boundary conditions)
2. On all lower derivatives order
3. (If time-dependent) then we need to know what it initial state looks like (initial conditions).

Physics-Informed Neural Networks (PINNs)

The purpose of PINN is to learn the dynamics of the PDE which can be done via

- Initialize your MLE Neural Networks
- For each iteration:
    - Sample domain points, boundary points, initial states,...
    - Make the MLE predict based on the sample
    - Compute loss
    - Backpropage to update
    - Repeat until the Mean-squared-error (MSE) or loss is minimized.

CODE