Mathematical Fundamentals for Electrochemical Energy Storage Systems

Exercise 4

Task 1: Matrix Multiplication

a. Implement matrix multiplication in python using for loops.

i. Use your implementation to multiple two randomly generated matrix consisting of 100 rows and 100 columns. Use the random module that is available as a part of numpy

ii. Use the time module to calculate the time taken to execute your multiplication

iii. Compare the time taken to do your multiplication vs the time taken by numpy. What do you think can be improved?

Task 2: Application of Matrix Multiplications

b. (Bonus: This exercise is to understand the power of matrix manipulation. Use the random module that is available as part of numpy to generate the necessary variables)

i. Implement a function in python that takes an input x, a weight value W, and a bias value b. It should then calculate $wx + b$ and return this as output. Assume that size of x, W, b are (1,1)

ii. Change the function above so that it can perform the same calculation as a matrix multiplication. Assume that you now have a matrix W that has both the weight and bias value and the matrix X that has the input point.

iii. Now assume that you have your input X of size (10000,1). You want to define a separate w and b for each of your 10000 points and perform the calculation $wx + b$. Implement this calculation as a matrix multiplication

iv. Implement the question 3 as a function in python but instead of returning $wx + b$, please return tanh ($wx + b$). (Fun fact: You have just implemented one neuron of a neural network. Celebrate, you are a machine learning master now