CpG Detector

1. **Loss Function - Mean Squared Error (MSE) :**
    a. The count of CpG pairs in DNA sequences is continuous in nature. MSE is an ideal loss function for regression problems like this
    b. In comparison to MAE, MSE penalizes larger errors more severely than smaller errors, making it effective for training models to minimize prediction errors. In context of CpG count we need to ensure diff/delta is contained within ‘0.35’

2. **Adam Optimizer:**
    a. It is well suited for training neural networks and while training, dynamically adjusts the learning rate, leading to faster convergence
    b. It can handle variety of data and model architectures effectively
    c. Since it computes adaptive learning rates automatically for each parameter, it reduces the need for manual tuning of hyperparameters

3. **learning_rate = 0.001**
    a. This value is in general chosen empirically based on its effectiveness in achieving convergence for a wide range of neural network architectures and datasets.
    b. This value is neither too slow nor too quick in terms of learning
    c. For this project have tried these LRs as well - 0.0025, 0.005 and 0.002

4. **LSTM_HIDDEN = 256**
    a. Hidden units help us to capture the temporal dependencies and patterns in the input sequences
    b. In general higher number of hidden units allows the model to learn more complex patterns and relationships
    c. By looking at the task in hand I wanted to start with a higher value (100+). Though I did try values like 16,32 and 64 to analyze and cross check
    d. CpG detection task was able to converge properly and generalize very well with value of 256, for given lr and epoch_num

5. **LSTM_LAYER = 2**
    a. The position of CpGs within the sequence and the surrounding nucleotides do not influence the count of CpGs

    b. While the task is non-linear in nature, it is in fact simpler compared to tasks involving complex dependencies like image data
    c. Because of a & b started with value ‘1’ and settled to ‘2’

6. **epoch_num = 100 for variable sequence and epoch_num = 50 for fixed**
    a. Started with epoch_num = 10, though it converged well to begin with for optimal lr, LSTM_LAYER and LSTM_HIDDEN, it wasn’t generalized well
    b. With epoch_num = 30, though performance was better than earlier, by looking at loss values across train & test and accuracy values, it seemed there’s room for learning still and model is not generalized fully
    c. I think training for even higher epoch number (than 50 & 100) might generalize the model slightly better and might improve accuracies further

7. Beyond evaluations on test set, code was written to compute accuracy on new set of samples by running predictions, to ensure model is properly generalized

8. Every time model was trained multiple iterations of no ‘7’ were performed to bring in randomness and to cross validate real world accuracies

9. While verifying variable length CpG detector, for a given number of samples - sequences of different length were tested to cross check model generalization