SNN ASSIGNMENT

NAME	Yash Jaiswal
PROJECT	SpikeVerse

#Task 1 : Input Encoding and Spike Visualization

Objective: Convert MNIST images into spike trains and visualize these spike patterns.

Steps:

- Loaded the MNIST dataset and used `ToTensor()` to normalize pixel values to [0, 1].
- Implemented `poisson_encoding(image, time_steps)` to convert an image into a spike train over a given number of time steps.
    - Each pixel spikes randomly over time depending on its intensity (brighter pixel = higher spike chance).
- Selected one random image and encoded it into 50 time steps.
- Visualized spike maps across time using a 5x10 grid of subplots.

Key Observation:

- The spike patterns over time still preserve the structure of the digit.

Conclusion: Poisson rate coding creates realistic spike patterns from static images and is an essential first step in simulating how SNNs process visual information over time.

#Task 2 : SNN Model Building and Training

Objective: Build a two-layer SNN using LIF neurons and train it to classify MNIST digits using spike input.

Steps:

- Defined a custom `LIFNeuronLayer` to simulate integrate-leak-fire behavior.

    - Neurons integrate input spikes and leak over time.
    - They fire if membrane potential crosses a threshold and reset after firing.
    - Used a sigmoid-based surrogate function to allow backpropagation.
- Built **`SNNModel`** using two LIF layers: input → hidden → output.
- During the forward pass:
    - Simulated spike activity over 30 time steps.
    - Collected spike counts from the output neurons.
- Used CrossEntropyLoss on total spike counts to train the model.
- Encoded each image into a spike train using Poisson encoding before feeding it to the network.
- Trained on 2000 MNIST samples with batch size 8 for 10 epochs using Adam optimizer.

Key Observation:

- The model learns to generate more spikes from the correct output neuron.
- Spike count acts like a class score, and prediction is based on the neuron that spikes most.

Conclusion: This task helped simulate biologically inspired learning using SNNs. The LIF neuron model captures time based spiking behavior, and surrogate gradients enable effective training using traditional optimization methods.

#Task 3 : Evaluation and Accuracy Testing

Objective: Evaluate the trained SNN model on the MNIST test dataset and measure its classification accuracy.

Steps:

- Loaded the MNIST test set with a batch size of 16.
- Defined an `evaluate()` function to:
    - Encode each test image into spike trains using Poisson encoding.
    - Simulate the model over 30 time steps.
    - Predict the digit by counting spikes in the output layer.
- Computed the total number of correct predictions.
- Printed the overall test accuracy.

Key Observation:

- The final model achieved high test accuracy using only spike counts for classification.

- The spike-based system generalized well to unseen test data, showing that learning through spikes is effective.

Conclusion: This task showed that our trained SNN can accurately classify digits using time based spike inputs. The approach mimics how the brain processes visual information and confirms that SNNs can learn effectively using biologically inspired mechanisms.