Challenge 6 - Perceptron Implementation

Tasks I did:

1. Built a simple neuron (simpleneuron.py):
    a. Two inputs, one bias, sigmoid activation.
    b. Verified truth-table wiring in a quick Pandas printout.
2. Implemented training loop (neurontrain.py):
    a. Forward pass: y=σ($w{\cdot}x$+b)
    b. Backprop step: Δ= $(t-y)$·σ′(y), update $w \leftarrow w$+η·Δ·x, $b \leftarrow b$+η·Δ
    c. Trained for 10 000 epochs with η=0.1
3. Trained on two logic functions:
    a. NAND targets [1,1,1,0] -> converged to outputs ≈[0.99,0.98,0.98,0.02].
    b. XOR targets [0,1,1,0] -> outputs stalled around ≈[0.50,0.50,…], never approaching correct labels.

What did I learn:

1. The perceptron successfully learns linearly-separable functions (NAND) using the simple update rule.
2. It fails on XOR because XOR is not linearly separable, no single weighted sum plus bias can carve out its “checkerboard” decision boundary.
3. Hyperparameters matter: learning rate too high can overshoot; too low slows convergence. η=0.1 was a good compromise for NAND.