Class Test: Deep and Reinforcement Learning

Module 1: A Neural Network from Scratch on Churn

Srikanth Pai, Asst. Professor, MSE

Instructions

Total: **5 marks**. Show all working. Keep four decimal places where the arithmetic is not exact. You may use $\sigma(3) = 0.9526$ and $\sigma(5) = 0.9933$.

The problem

A telco predicts customer churn from two standardised features, tenure x_1 and monthly charges x_2, with a 2–2–1 multilayer perceptron: ReLU on the hidden layer, sigmoid on the output, binary cross-entropy loss. Use the matrix convention from class ($\mathbf{z}^{(l)} = W^{(l)\top}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}$). Train by full-batch AdamW with learning rate $\eta = 0.1$, weight-decay coefficient $\lambda = 0.01$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, and all moment buffers initialised to zero.

Full batch.

Customer A: $\mathbf{x}_A = (1,\, 2),\ y_A = 1$ (churned); Customer B: $\mathbf{x}_B = (2,\, 1),\ y_B = 0$ (retained).

Initialisation. All biases start at zero:

$$W^{(1)} = \begin{pmatrix} 2 & -1 \\ 1 & 1 \end{pmatrix}, \qquad \mathbf{b}^{(1)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad W^{(2)} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \qquad b^{(2)} = 0.$$

Questions

1. **Forward pass.** Compute the predicted churn probability for each customer and report the full-batch loss. (1.5 marks)
2. **Backward pass.** Compute the batch-averaged gradients $\nabla_{W^{(1)}} L$, $\nabla_{b^{(1)}} L$, $\nabla_{W^{(2)}} L$ and $\nabla_{b^{(2)}} L$. (2 marks)
3. **AdamW update.** Perform one full-batch AdamW step ($t = 1$) and report the updated $W^{(1)}$, $b^{(1)}$, $W^{(2)}$ and $b^{(2)}$. (1.5 marks)

Reminder. Binary cross-entropy over a batch of B examples:

$$L = -\frac{1}{B}\sum_{i=1}^{B}\Big[\, y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)\,\Big],$$

where i indexes the B examples in the batch, $y_i \in \{0, 1\}$ is the true label (here 1 = churned), and $\hat{y}_i \in (0, 1)$ is the model's predicted probability that example i churns.