Viva Questions & Answers - Core + Basic/Medium/Deep

Q1. Why choose this dataset?
Because it satisfies assignment requirements (>=500 samples, >=5 features), is well-known, and suitable for binary classification in medical diagnosis.

Q2. Why F1-score as primary metric?
F1 balances precision and recall which is important in medical diagnosis to reduce false negatives.

Q3. What preprocessing steps did you perform?
Loaded CSV, assigned headers, converted labels M/B to 1/0, dropped ID, custom train-test split (80/20), and custom standardization (mean/std).

Q4. Why ReLU for hidden layers?
ReLU reduces vanishing gradients, is computationally cheap, and usually speeds up convergence.

Q5. Why sigmoid on output?
Sigmoid gives probability for binary classification and pairs naturally with binary cross-entropy loss.

Q6. Why might logistic perform better than MLP?
Dataset may be almost linearly separable; logistic regression is optimal and simpler, MLP may add variance.

Q7. How did you implement gradient descent?
Computed gradients manually in NumPy and updated weights with W = W - lr * dW; MLP used mini-batch SGD.

Q8. What loss function and why?
Binary Cross-Entropy (BCE) because it measures difference between predicted probabilities and labels for binary tasks.

Q9. What regularization did you use?
L2 weight decay to reduce overfitting.

Q10. How did you validate your pipeline?
Used an 80/20 train-test split and evaluated on held-out test set with accuracy, precision, recall, and F1.

Q11. What challenges did you face?
CSV header issues, label mapping, and ensuring from-scratch implementations matched expected behavior.

Basic Questions

B1. What is a learning rate?
A hyperparameter that determines the size of weight updates during training.

B2. What is a batch size?
Number of samples processed before the model's parameters are updated.

B3. Why scale features?
Scaling centers features to similar ranges, improving gradient descent convergence.

Medium Questions

M1. What is vanishing gradient?
When gradients become very small in deep networks, slowing or preventing learning in earlier layers.

M2. Why use weight initialization schemes?
Proper initialization (He/Xavier) helps preserve variance and speed up convergence.

M3. How to choose number of hidden units?

Start with modest sizes and validate; avoid too many parameters on small datasets to prevent overfitting.

Deep Questions

D1. Explain backpropagation mathematically.

Backpropagation applies chain rule to compute gradients of loss w.r.t weights layer-by-layer, using local derivatives and upstream deltas.

D2. Why use mini-batch SGD vs full-batch?

Mini-batch provides a balance between noisy gradients (stochastic) and computation efficiency; often faster convergence.

D3. How does L2 regularization affect gradients?

Adds term proportional to weights in gradient (lambda * W), shrinking weights and penalizing large values.