Puzzle 1: *The Perfectly Wrong Model*

A neural network shows **99% training accuracy** and **98% validation accuracy**.
When deployed, accuracy drops to **52%**.

Question:
Which is the *most likely* hidden reason?

A. Learning rate too small
B. Activation function mismatch
C. Data leakage during validation
D. Vanishing gradient

Puzzle 2: *Activation Function Trap*

A deep network uses **Sigmoid activation in all layers**.
Loss decreases initially, then stagnates.

Which change improves learning WITHOUT changing architecture?

A. Increase epochs
B. Increase learning rate
C. Replace hidden layer activations with ReLU
D. Add more training data

Puzzle 3: *Gradient Descent Paradox*

Two models use the same dataset and architecture.

- Model A: learning rate = 0.01
- Model B: learning rate = 0.5

Model B's loss **oscillates**, Model A converges slowly.

Which statement is TRUE?

A. Model A is underfitting
B. Model B overshoots the minima

C. Model A is stuck in local minima
D. Model B has vanishing gradients

Puzzle 4: *The Perceptron Illusion*

A single-layer perceptron is trained to classify XOR data.

What will happen?

A. Perfect classification
B. Slow convergence
C. Random outputs
D. Model fails regardless of epochs

Puzzle 5: *Backpropagation Riddle*

During backpropagation, gradients become extremely small in early layers.

Which action directly addresses the root cause?

A. Increase dataset size
B. Change activation to ReLU
C. Reduce batch size
D. Increase epochs

Puzzle 6: *Overfitting vs Underfitting Trick*

Model behaviour:

- Training loss: low
- Validation loss: increasing
- Test loss: very high

Best corrective action?

A. Increase model complexity
B. Remove regularization

C. Apply dropout
D. Reduce training data

Puzzle 7: *Learning Rate Illusion*

A network converges fast but reaches a **sub-optimal minimum**.

Most effective fix?

A. Decrease learning rate
B. Increase learning rate
C. Add noise to gradients (momentum/optimizer change)
D. Reduce epochs

Puzzle 8: *Neuron Identity Crisis*

Two neurons have different weights but produce the same output for all inputs.

What does this indicate?

A. Overfitting
B. Model identifiability problem
C. Gradient explosion
D. Dead neurons

Puzzle 9: *Validation Set Trick*

Why should validation data **never** be used for weight updates?

A. Increases computation
B. Causes slow learning
C. Leads to biased performance estimation
D. Causes vanishing gradients

Puzzle 10: *Final Year Challenge*

Arrange the steps in **correct conceptual order**:

1Backpropagation
2Activation computation
3 Weight initialization
4 Loss calculation