# Quiz 4: Neural Network Backpropagation & Optimization

## Instructions

Select the best answer for each question. These questions cover backpropagation concepts and gradient descent optimization techniques.

## Questions

### 1. Efficiency of Backpropagation

What is the main efficiency advantage of the backpropagation algorithm compared to calculating the gradient of the loss with respect to each parameter independently? A) It uses less memory during the forward pass. B) It works better with non-linear activation functions. C) It requires fewer training examples to converge. D) It avoids redundant calculations by reusing intermediate values computed during the forward and backward passes.

### 2. Foundation of Backpropagation

Backpropagation relies fundamentally on which mathematical rule to compute gradients layer by layer, propagating the error signal backward through the network? A) The Chain Rule B) Bayes' Theorem C) The Law of Large Numbers D) Taylor's Theorem

### 3. Calculating Batch Gradients

During mini-batch gradient descent, the gradients $dW^{[l]} = \frac{\partial J}{\partial W^{[l]}}$ and $db^{[l]} = \frac{\partial J}{\partial b^{[l]}}$ used for the update step are typically computed how? A) Using only the example with the highest loss in the batch. B) Summing the gradients calculated for each example in the mini-batch. C) Averaging the gradients calculated for each example in the mini-batch. D) Selecting the gradient from a randomly chosen example in the mini-batch.

### 4. Meaning of the Error Term

In the standard backpropagation derivation, what does the term $\delta^{[l]} = \frac{\partial \mathrm{L}}{\partial z^{[l]}}$ (or its batch equivalent $\frac{\partial J}{\partial Z^{[l]}}$) represent? A) The final activation output of layer $l$. B) The error signal or gradient of the loss with respect to the pre-activation values $z^{[l]}$ (or $Z^{[l]}$) of layer $l$. C) The activation function's derivative at layer $l$. D) The contribution of layer $l$ to the network's prediction accuracy.

### 5. Automatic Differentiation

Modern deep learning frameworks (PyTorch, TensorFlow, JAX) implement automatic differentiation. What is the primary practical benefit of this for the developer? A) It automatically computes the necessary gradients, eliminating the need for manual derivation and implementation of backpropagation. B) It automatically selects the best network architecture. C) It automatically chooses the optimal learning rate. D) It performs the forward propagation faster than manual coding.

### 6. Gradient Descent Variants

Which gradient descent variant offers a practical balance by using a small subset of data for each update, allowing for vectorization benefits and more stable convergence than SGD, while being less computationally expensive than Batch GD? A) Batch Gradient Descent (BGD) B) Stochastic Gradient Descent (SGD) C) Mini-batch Gradient Descent D) Newton's Method

### 7. Learning Rate Issues

If, during training, the loss function value starts oscillating wildly or increasing instead of decreasing, what is a likely cause related to the learning rate $\alpha$? A) The learning rate is too small. B) The learning rate is exactly zero. C) The batch size is too large. D) The learning rate is too large.

### 8. Momentum Optimizer

What is the primary benefit of using the Momentum optimization algorithm compared to standard Mini-batch Gradient Descent? A) It guarantees finding the global minimum of the loss function. B) It helps accelerate convergence, especially in directions of consistent gradient, and dampens oscillations. C) It automatically adjusts the batch size during training. D) It eliminates the need for a learning rate.

### 9. Adam Optimizer

Conceptually, the Adam (Adaptive Moment Estimation) optimizer combines the key ideas of which two other optimization algorithms? A) Momentum and RMSProp B) AdaGrad and SGD C) Batch Gradient Descent and Momentum D) RMSProp and Newton's Method

### 10. Backpropagation in Deep Networks (Harder)

On a scale of 1 (I already forgot what backpropagation means) to 10 (I can write the algorithm of backpropagation for a two layer neural network without autodiff), how A) I already forgot what backpropagation means B) I think I understand what backpropagation means, but I'm glad autodiff handles it C) I understand the importance of backpropagation and could re-derive some steps alone, but I'm glad autodiff handles it D) I can implement the algorithm of backpropagation for a two layer neural network without autodiff

## Answers

1. D
2. A
3. C

4. B
5. A
6. C
7. D
8. B
9. A
10. D

## Explanations

1. **D)** Backpropagation efficiently computes gradients by reusing intermediate values (activations from forward pass, error terms from backward pass), avoiding recalculating shared sub-expressions multiple times.
2. **A)** The Chain Rule is the core mathematical principle allowing the gradient of the final loss to be related back to the parameters in earlier layers by multiplying local derivatives.
3. **C)** In mini-batch GD, gradients are computed for each example in the batch, and then averaged to get a single update direction for the parameters.
4. **B)** $\delta^{[l]}$ represents how much the final loss changes for a small change in the pre-activation output $z^{[l]}$ of the neurons in layer $l$. It's the error signal used to compute the gradients for $W^{[l]}$ and $b^{[l]}$.
5. **A)** Autodiff systems build a computational graph and automatically apply the chain rule, freeing the developer from manually calculating and coding the complex gradient formulas.
6. **C)** Mini-batch GD strikes a balance: faster and less memory-intensive than BGD, more stable and vectorization-friendly than SGD.
7. **D)** A learning rate that is too large can cause the updates to overshoot the minimum, leading to oscillations or divergence (increasing loss).
8. **B)** Momentum adds a fraction of the previous update vector to the current gradient step, helping to speed up progress along shallow ravines and dampen oscillations across steep directions.
9. **A)** Adam combines the adaptive learning rates per parameter (like RMSProp) with the concept of momentum (using moving averages of both the gradient and its square).
10. **D)** Duh.