QF624-2025-W8

Number of participants: 19

1. In the forward pass, the pre-activation result z_2^1 (i.e.\ before applying ReLU) of the hidden neuron f_2^1 is:

1 correct answer
out of 11 respondents

✓	$z_2^1 = f_1^1 w_2^1 + f_1^2 w_2^3$	9%	1 vote
	$z_2^1 = f_1^1 w_2^2 + f_1^2 w_2^3$	0%	0 votes
	$z_2^1 = x^1 w_2^2 + x^2 w_2^4$	27%	3 votes
	$z_2^1 = f_1^1 w_2^3 + f_1^2 w_2^4$	64%	7 votes

2. **Given $f_3 = \mathrm{ReLU}\left(f_1^1 w_3^1 + f_1^2 w_3^2\right)$, which expression equals $\frac{\partial f_3}{\partial w_3^2}$? Here, $z_3 = f_1^1 w_3^1 + f_1^2 w_3^2$ and $\mathbf{1}\{\cdot\}$ is the indicator of the ReLU's "active" region.**

3 correct answers
out of 10 respondents

	$f_1^1 \, \mathbf{1}\{z_3 > 0\}$	0%	0 votes
✓	$f_1^2 \, \mathbf{1}\{z_3 > 0\}$	30%	3 votes
	$(f_1^1 + f_1^2) \, \mathbf{1}\{z_3 > 0\}$	70%	7 votes
	$\mathbf{1}\{z_3 > 0\}$	0%	0 votes

3. Suppose we replaced ReLU with a sigmoid activation in every hidden node. Which phenomenon becomes most severe as the network depth grows?

7 correct answers
out of 10 respondents

✓	Vanishing gradients, because $\sigma'(z) \leq \frac{1}{4}$ everywhere.	70%	7 votes
	Exploding gradients, because $\sigma'(z)\lvert 1-\sigma(z)\rvert$ can exceed 1.	10%	1 vote
	Both vanish and explode equally.	20%	2 votes
	Neither; sigmoid has unit-scale gradients on average.	0%	0 votes

4. In vanilla gradient descent, the weight update rule for a single scalar weight w minimizing loss $L(w)$ is $w \leftarrow w - \eta \frac{dL}{dw}$. Which of the following best describes why we subtract $\eta \frac{dL}{dw}$?

10 correct answers
out of 10 respondents

	To move w in the direction that increases the loss most rapidly.	0%	0 votes
✓	To move w in the direction that decreases the loss most rapidly.	100%	10 votes
	To set the gradient to zero immediately.	0%	0 votes
	To ensure the loss stays constant.	0%	0 votes

5. Suppose your current learning rate η is too large. What behavior would you likely observe during training?

7 correct answers
out of 7 respondents

	The loss decreases smoothly to the minimum.	0%	0 votes
	The loss decreases too slowly and stalls.	0%	0 votes
✓	The loss bounces around or diverges (goes up).	100%	7 votes
	The gradient becomes zero at every step.	0%	0 votes

6. Sometimes we reduce η over time (e.g.\ halve it every 100 steps). Why might this help?

8 correct answers
out of 9 respondents

✓	To avoid overshooting as we approach the minimum.	89%	8 votes
	To make the gradients larger later in training.	11%	1 vote
	To change the objective function.	0%	0 votes
	To guarantee we find a global minimum in non-convex problems.	0%	0 votes

7. Vanilla (batch) gradient descent computes the gradient over the entire training set before each update. Which is a direct consequence?

8 correct answers
out of 8 respondents

	Every update is noisy and high-variance.	0%	0 votes
✓	Each update exactly follows the true loss surface but can be slow per step.	100%	8 votes
	It only works for linear models.	0%	0 votes
	It always converges in one step.	0%	0 votes