AI Research Labs

Deep Learning Foundations · Practice test 6 · 20 questions

Loss Functions and Output Layers

How to use this test. Answer all 20 questions without notes, then mark yourself with the answer key at the back. Each question has exactly one best answer. Score 75% or more and you've mastered the material; below that, reread the lessons for the questions you missed.

6.1 Training as Minimising a Loss

Q1 calculate Data $(1,2),(2,4),(3,5)$; line $\hat{y}=x$ (so $m=1$, $c=0$). What is the MSE J?

A 1.67

B 9

C 3

D 5

Q2 calculate For the same data and line, $\partial J/\partial c = -\frac{2}{N}\sum(y-mx-c)$. What is its value?

A $+10/3 \approx 3.33$

B $-5/3 \approx -1.67$

C $-10/3 \approx -3.33$

D $-22/3 \approx -7.33$

Q3 interpret At $m=1$, $c=0$ both gradients are negative. What will a gradient descent step do?

A Increase m but decrease c, rotating the line about its centre

B Increase both m and c, moving the line up towards the points

C Decrease both m and c, moving the line further down

D Leave m and c unchanged, since the line already passes nearby

Q4 concept What are the two decisions that define how a network is trained?

A The loss to measure error, and the optimiser to minimise it

B The number of layers, and the number of units in each one

C The activation function, and the size of the output layer

D The training set, and the test set used for final evaluation

Q5 calculate After one step with $\eta=0.05$, the line becomes $\hat{y}=1.367x+0.167$ and the MSE drops to about 0.66. What was the MSE before?

A 9

B 1.5

C 3

D 0.66

6.2 MSE, MAE and Asymmetric Losses

Q6 `calculate` Targets $1, 2, 3, 4, 100$. Which single prediction minimises MSE, and which minimises MAE?

- A MSE: 3 (the median). MAE: 22 (the mean).
- B MSE: 22 (the mean). MAE: 3 (the median).
- C Both are minimised by the mean, 22.
- D Both are minimised by the median, 3.

Q7 `concept` Why does one huge outlier affect an MSE-trained model much more than an MAE-trained one?

- A MSE counts each outlier twice in its average
- B MSE's gradient grows with the error; MAE's stays ±1
- C MAE ignores every error larger than one entirely
- D MAE uses the median of the inputs, not the outputs

Q8 `concept` MAE is not differentiable at an error of exactly zero. How is it handled?

- A A subgradient (any value in [−1, 1], usually 0) is used
- B MAE is replaced by MSE whenever the error becomes zero
- C Those examples are dropped from the training set entirely
- D Training stops, since the minimum has then been reached

Q9 `apply` A shop forecasts demand. Running out costs three times as much as overstocking. Which loss fits?

- A A pinball loss that weights under-prediction 3×
- B Cross-entropy, because the demand is uncertain
- C MAE, because it is robust to the outlying days
- D MSE, because it punishes the large errors much more

Q10 `apply` You use a sigmoid output for a regression network predicting prices in thousands. What goes wrong?

- A Nothing: sigmoid is the standard regression output
- B Every prediction is trapped between 0 and 1
- C The network can only learn linear functions
- D The loss becomes negative for large prices

6.3 Softmax and Cross-entropy

Q11 `calculate` Logits $(2, 1, 0)$. What probability does softmax give the first class?

- A About 0.880
- B About 0.667
- C About 0.333
- D About 0.665

Q12 calculate The correct class receives probability 0.5. What is the cross-entropy loss (natural log)?

A About 1.386
B About 0.693
C 0.5
D About 0.301

Q13 concept What is $\partial H / \partial s_j$ for softmax followed by cross-entropy?

A $-z_j / o_j$
B $o_j(1 - o_j)$
C $(o_j - z_j)\, o_j(1 - o_j)$
D $o_j - z_j$

Q14 calculate Softmax outputs $(0.665, 0.245, 0.090)$ and the correct class is the first. What is the gradient with respect to the logits?

A $(0.665,\ 0.245,\ 0.090)$
B $(-0.335,\ 0.245,\ 0.090)$
C $(-0.335,\ 0,\ 0)$
D $(0.335,\ -0.245,\ -0.090)$

Q15 apply Your softmax overflows with logits like $(1000, 999, 998)$. What's the standard fix?

A Subtract the largest logit from all of them first
B Divide every logit by the sum of all the logits
C Use base-10 exponentials instead of base e
D Clip each logit at 1 before the exponential

6.4 Binary, Multi-class and Logits in PyTorch

Q16 calculate The true label is $y = 1$ and the model predicts $\hat{y} = 0.1$. What is the binary cross-entropy?

A About 2.303
B About 0.105
C 0.9
D About 0.693

Q17 apply Your PyTorch classifier ends with a softmax layer and you train with `nn.CrossEntropyLoss`. What's wrong?

A CrossEntropyLoss only works for two classes, so use MSELoss instead
B The loss applies softmax again; the model should return raw logits
C Nothing: CrossEntropyLoss expects probabilities, so softmax is required
D The softmax must come after the loss is computed, not before it

Q18 `concept` Why isn't accuracy used as the training loss?

- A It is far too expensive to compute on every single mini-batch
- B It always gives the same value for every possible model
- C It can only be measured on the test set, not on the training data
- D It's piecewise constant, so its gradient is zero almost everywhere

Q19 `calculate` Class A is uniform on $[0, 5]^2$, class B on $[0, 10]^2$, equal priors. What is $P(A \mid x)$ for a point inside $[0, 5]^2$?

- A 0.5
- B 0.8
- C 1.0
- D 0.25

Q20 `interpret` In that example the best possible accuracy is 0.875. A colleague's network reaches 0.87. What does that suggest?

- A It must be overfitting, since 0.87 is suspiciously high
- B It is near the Bayes limit; little more can be gained
- C It is badly underfitting and needs a much larger network
- D The test set must be leaking into the training data

Answer key

Each answer comes with the reasoning, including why the most tempting wrong option is wrong.

Q1 **C** · lesson 6.1

Residuals $y - \hat{y}$ are $1, 2, 2$; squares $1, 4, 4$; mean $9/3 = 3$. 9 is the sum, not the mean; 1.67 averages the residuals instead of their squares.

Q2 **C** · lesson 6.1

The residuals sum to $1 + 2 + 2 = 5$, so $-\frac{2}{3} \times 5 = -3.33$. Dropping the 2 gives -1.67; -7.33 is $\partial J/\partial m$, which weights each residual by x.

Q3 **B** · lesson 6.1

The update subtracts the gradient: $\theta \leftarrow \theta - \eta \nabla J$, so negative gradients increase the parameters. The points lie above the line (positive residuals), so moving up is right.

Q4 **A** · lesson 6.1

Training means choosing a loss $J(\theta)$ that captures the task's error, then minimising it, usually with gradient descent. Architecture and data splits matter, but they're separate decisions from how training works.

Q5 **C** · lesson 6.1

The starting line $\hat{y} = x$ had MSE 3 (squared residuals $1, 4, 4$, mean 3). One gradient step cut it to about 0.66. 9 is the unaveraged sum of squares.

Q6 **B** · lesson 6.2

Squared error is minimised by the mean, $110/5 = 22$; absolute error by the median, 3. The single outlier (100) drags the MSE answer far from the typical value, which is why MAE is called robust.

Q7 **B** · lesson 6.2

The derivative of e^2 is $2e$, so a residual of 100 pulls 100 times harder than a residual of 1. The derivative of $|e|$ is ± 1, the same size for every error. MAE doesn't ignore large errors; it just doesn't amplify them.

Q8 **A** · lesson 6.2

At zero, $|e|$ has a corner; any slope between −1 and 1 is a valid subgradient, and frameworks use 0. Examples aren't dropped, the loss isn't swapped, and one zero error doesn't mean the overall minimum.

Q9 **A** · lesson 6.2

The pinball loss charges αr for over-prediction and $-\beta r$ for under-prediction; with $\beta = 3\alpha$ the model learns to aim high. MSE and MAE are symmetric, and cross-entropy is for classification.

Q10 **B** · lesson 6.2

A sigmoid outputs values in (0, 1), so prices in the hundreds are unreachable (and the unit saturates). Use a linear output for regression. The loss stays non-negative and the hidden layers are still non-linear.

Q11 **D** · lesson 6.3

$e^2/(e^2 + e^1 + e^0) = 7.389/11.107 \approx 0.665$. 0.667 is $2/3$, i.e. normalising the raw logits without exponentiating; 0.880 is $\sigma(2)$, the sigmoid of one logit.

Q12 **B** · lesson 6.3

$-\ln 0.5 \approx 0.693$. 0.301 uses base-10 logs, 1.386 is $-\ln 0.25$, and 0.5 is the probability itself.

Q13 **D** · lesson 6.3

The softmax derivative cancels the $1/o_j$ from the log, leaving predicted minus target. $o_j(1 - o_j)$ is only part of the softmax derivative, the third option wrongly adds an activation factor, and $-z_j/o_j$ is the derivative with respect to o_j, not s_j.

Q14 **B** · lesson 6.3

$o - z$ with $z = (1, 0, 0)$ gives $(-0.335, 0.245, 0.090)$: push the correct logit up, the others down. The wrong classes still get non-zero gradients, and the sign convention is predicted minus target.

Q15 **A** · lesson 6.3

Subtracting the maximum gives $(0, -1, -2)$; the common factor cancels in the ratio, so the probabilities are unchanged and nothing overflows. Dividing or clipping changes the probabilities; base 10 overflows too.

Q16 **A** · lesson 6.4

For $y = 1$ the loss is $-\ln \hat{y} = -\ln 0.1 \approx 2.303$: a confident mistake. 0.105 is $-\ln 0.9$, what you'd get if y were 0; 0.693 is $-\ln 0.5$.

Q17 **B** · lesson 6.4

nn.CrossEntropyLoss applies log-softmax internally, so a softmax in the model gets applied twice: training runs but the gradients are distorted. Return logits and apply softmax only for reporting probabilities. It handles any number of classes.

Q18 **D** · lesson 6.4

Accuracy only changes when a prediction crosses the threshold, so small weight changes leave it unchanged: no gradient to follow. It's cheap and can be measured on any set; train on cross-entropy and report accuracy.

Q19 **B** · lesson 6.4

Densities there are $1/25$ for A and $1/100$ for B: $P(A \mid x) = \frac{1/25}{1/25 + 1/100} = 0.8$. 0.5 ignores the densities, 0.25 is the area ratio of the squares, and 1.0 ignores that B also covers the small square.

Q20 **B** · lesson 6.4

Where the classes overlap, no classifier can beat the Bayes-optimal accuracy, 0.875 here. A model at 0.87 is essentially there; a bigger network can't push past the ceiling, and nothing about 0.87 suggests leakage.