Overfitting, Underfitting

& Generalization — Practice Questions

10 Fully Solved Numericals with Step-by-Step Solutions

Topics: Classification Metrics · Training/Validation Loss · Bias-Variance · Early Stopping

Q1. Binary Classification — Overfitting Detection via Metrics

Problem

A binary classifier is trained and evaluated. The following metrics are recorded:

Set	Accuracy	Precision	Recall	F1-Score
Training	97.5%	98.0%	97.1%	97.54%
Test	72.3%	74.5%	69.8%	72.07%

(i) Calculate the generalization gap in accuracy.

(ii) Verify the reported Training F1-Score using Precision and Recall.

(iii) Classify the model and provide two metric-based justifications.

Step-by-Step Solution

Step 1 — Generalization Gap:

Generalization Gap = Training Accuracy − Test Accuracy

= 97.5% − 72.3% = 25.2%

Generalization Gap = 25.2%

Step 2 — Verify F1-Score:

F1 = 2 × (Precision × Recall) / (Precision + Recall)

= 2 × (0.980 × 0.971) / (0.980 + 0.971)

= 2 × 0.95138 / 1.951 = 1.90276 / 1.951 ≈ 0.9753 = 97.53%

Verified F1 ≈ 97.54% ✓ (matches reported value)

Step 3 — Classification:

Model is OVERFITTING.

Reason 1: Accuracy gap of 25.2% is far above acceptable (~5%). High training accuracy with much lower test accuracy is the hallmark of overfitting.

Reason 2: Recall drops sharply from 97.1% (train) to 69.8% (test), showing the model fails to generalize its ability to capture true positives.

Classification: OVERFITTING

Q2. Epoch-wise Loss Analysis — Identifying the Overfitting Onset

Problem

A neural network is trained with the following loss curve data:

Epoch	Training Loss	Validation Loss	Gap
20	0.45	0.48	0.03
60	0.20	0.22	0.02
100	0.10	0.28	0.18
140	0.04	0.51	0.47
180	0.02	0.79	0.77

(i) At which epoch does overfitting clearly begin?

(ii) By what % did the validation loss worsen from epoch 60 to 180?

(iii) Suggest a gap-based early stopping threshold.

Step-by-Step Solution

Step 1 — Overfitting Onset:

Between epoch 60 and 100, training loss continues to fall (0.20→0.10) but validation loss INCREASES (0.22→0.28).

The gap jumps from 0.02 to 0.18 — a 9× increase in just 40 epochs.

Overfitting clearly begins between epoch 60 and 100

Step 2 — Validation Loss Worsening (epoch 60 → 180):

% Worsening = [(New − Old) / Old] × 100

= [(0.79 − 0.22) / 0.22] × 100 = [0.57 / 0.22] × 100 = 259.1%

Validation loss worsened by 259.1% from epoch 60 to 180

Step 3 — Early Stopping Threshold:

At epoch 60, the gap is only 0.02 (healthy). By epoch 100 it's 0.18 (deteriorating).

A reasonable threshold: Stop training when gap > 0.05 for 3 consecutive checkpoints.

Suggested threshold: gap > 0.05 (numerical trigger between healthy gap of 0.02 and danger zone of 0.18)

Q3. Decision Tree Depth — Bias-Variance Tradeoff

Problem

Max Depth	Train Error (%)	Val Error (%)	Generalization Gap (%)
1	35.2	36.0	0.8
3	18.4	19.5	1.1
6	9.1	13.8	4.7
9	4.2	21.3	17.1
15	0.8	31.6	30.8

(i) At which depth does overfitting become significant?

(ii) Find the smallest generalization gap and the depth where it occurs.

(iii) What is the optimal depth? Expected test error range?

Step-by-Step Solution

Step 1 — Overfitting Onset:

From depth 3→6: val error rises from 19.5% to 13.8% (still improving).

From depth 6→9: train error drops 9.1→4.2% but val error JUMPS 13.8→21.3%.

The gap explodes from 4.7% to 17.1% — a 3.6× increase.

Overfitting becomes significant at Max Depth = 9

Step 2 — Smallest Generalization Gap:

Depth 1: gap = 36.0 − 35.2 = 0.8%

Depth 3: gap = 19.5 − 18.4 = 1.1%

Smallest gap = 0.8% at depth 1 (but high error = underfitting)

Smallest gap = 0.8% at depth 1

Step 3 — Optimal Depth:

Best validation error = 13.8% at depth 6.

Expected test error range ≈ 13% – 15% (assuming test ≈ validation error).

Optimal depth = 6 | Expected test error ≈ 13% – 15%

Q4. Multiclass Classification — Per-Class Metrics & Macro F1

Problem

A 3-class model reports per-class metrics on the test set:

Class	Precision	Recall
Cat	0.82	0.76
Dog	0.75	0.88
Bird	0.91	0.70

(i) Compute F1-Score for each class.

(ii) Compute Macro-Average F1.

(iii) If training accuracy is 96%, classify model behavior.

Step-by-Step Solution

Step 1 — Per-Class F1:

F1 = 2 × P × R / (P + R)

Cat: F1 = 2×0.82×0.76 / (0.82+0.76) = 1.2464/1.58 = 0.7888 ≈ 78.88%

Dog: F1 = 2×0.75×0.88 / (0.75+0.88) = 1.32/1.63 = 0.8098 ≈ 80.98%

Bird: F1 = 2×0.91×0.70 / (0.91+0.70) = 1.274/1.61 = 0.7913 ≈ 79.13%

F1: Cat=78.88%, Dog=80.98%, Bird=79.13%

Step 2 — Macro-Average F1:

Macro F1 = (78.88 + 80.98 + 79.13) / 3 = 239.0 / 3 ≈ 79.66%

Macro F1 = 79.66%

Step 3 — Model Behavior:

Training accuracy ≈ 96%, Macro Test F1 ≈ 79.66%. Gap ≈ 16.3%.

Model is OVERFITTING — high train accuracy but considerably lower test performance.

Classification: OVERFITTING (gap ≈ 16%)

Q5. Regularization Effect on Train vs Validation Loss

Problem

Two models are trained on the same dataset (1000 train, 250 val):

Model	Train Loss	Val Loss	L2 λ
A (no reg)	0.05	1.20	0
B (L2 reg)	0.18	0.24	0.01

(i) Compute the loss gap for both models.

(ii) Which model generalizes better? Justify numerically.

(iii) By what % did L2 regularization reduce the validation loss?

Step-by-Step Solution

Step 1 — Loss Gaps:

Model A gap = 1.20 − 0.05 = 1.15

Model B gap = 0.24 − 0.18 = 0.06

Gap: Model A = 1.15 | Model B = 0.06

Step 2 — Better Generalizer:

Model B: gap of only 0.06 vs Model A's 1.15 — 19× smaller gap.

Although Model B's training loss (0.18) is higher, its validation loss (0.24) is far closer to training, indicating far better generalization.

Model B generalizes better (gap ratio: 1.15 vs 0.06 — 19× improvement)

Step 3 — Validation Loss Reduction:

% Reduction = [(1.20 − 0.24) / 1.20] × 100 = [0.96 / 1.20] × 100 = 80%

L2 regularization reduced validation loss by 80%

Q6. Confusion Matrix to Derive All Classification Metrics

Problem

A binary classifier produces the following confusion matrix on the test set:

	Predicted: Positive	Predicted: Negative
Actual: Positive	TP = 420	FN = 80
Actual: Negative	FP = 60	TN = 440

(i) Calculate Accuracy, Precision, Recall, and F1-Score.

(ii) If training accuracy is 99.1%, compute the generalization gap.

(iii) Is the model overfitting, underfitting, or well-generalized?

Step-by-Step Solution

Step 1 — All Metrics:

Total = TP+FP+FN+TN = 420+60+80+440 = 1000

Accuracy = (TP+TN)/Total = (420+440)/1000 = 860/1000 = 86.0%

Precision = TP/(TP+FP) = 420/(420+60) = 420/480 = 0.875 = 87.5%

Recall = TP/(TP+FN) = 420/(420+80) = 420/500 = 0.840 = 84.0%

F1 = 2×0.875×0.840/(0.875+0.840) = 1.47/1.715 = 0.8571 = 85.71%

Accuracy=86.0%, Precision=87.5%, Recall=84.0%, F1=85.71%

Step 2 — Generalization Gap:

Gap = 99.1% − 86.0% = 13.1%

Generalization Gap = 13.1%

Step 3 — Classification:

A gap of 13.1% is substantial. The model achieves near-perfect training accuracy (99.1%) but drops to 86% on test — classic overfitting signature.

Classification: OVERFITTING

Q7. Underfitting Detection via High Bias Indicators

Problem

A logistic regression model is evaluated on a medical diagnosis task:

Set	Accuracy	Precision	Recall	F1
Training	61.3%	59.8%	63.2%	61.45%
Validation	59.7%	58.1%	61.4%	59.71%

Baseline (random classifier for balanced classes): 50% accuracy.

(i) Compute the generalization gap.

(ii) Verify the Training F1-Score.

(iii) Classify model behavior. What does the small gap indicate?

Step-by-Step Solution

Step 1 — Generalization Gap:

Gap = 61.3% − 59.7% = 1.6%

Generalization Gap = 1.6% (very small)

Step 2 — Verify F1:

F1 = 2 × 0.598 × 0.632 / (0.598 + 0.632) = 0.75594 / 1.230 = 0.6146 ≈ 61.46%

Verified Training F1 ≈ 61.45% ✓

Step 3 — Classification:

Although the gap (1.6%) is small (suggesting the model generalizes consistently), both train AND validation accuracy are low (~61%) — barely above the 50% baseline.

Small gap + low absolute performance = HIGH BIAS = UNDERFITTING.

The model hasn't learned enough from the training data itself.

Classification: UNDERFITTING (high bias, both errors are high)

Q8. Dropout Effect — Comparing Loss Trajectories

Problem

Two identical networks trained for 200 epochs, one with dropout (rate=0.4):

Model	Epoch 50 Train	Epoch 50 Val	Epoch 200 Train	Epoch 200 Val
Without Dropout	0.09	0.78	0.02	1.35
With Dropout	0.22	0.26	0.15	0.19

(i) Compute gap at epoch 200 for both models.

(ii) By what % did dropout reduce validation loss at epoch 200?

(iii) Identify which epochs to use as 'best checkpoint' for each model.

Step-by-Step Solution

Step 1 — Gap at Epoch 200:

Without Dropout: gap = 1.35 − 0.02 = 1.33

With Dropout: gap = 0.19 − 0.15 = 0.04

Gap Without Dropout = 1.33 | Gap With Dropout = 0.04

Step 2 — Validation Loss Reduction:

% Reduction = [(1.35 − 0.19) / 1.35] × 100 = [1.16/1.35] × 100 = 85.9%

Dropout reduced validation loss by 85.9% at epoch 200

Step 3 — Best Checkpoint:

Without Dropout: val loss increases throughout. Best checkpoint = epoch 50 (val=0.78, lower than 1.35).

With Dropout: val loss still decreasing (0.26→0.19). Could continue training or use epoch 200.

Best checkpoint — Without Dropout: Epoch 50 | With Dropout: Epoch 200 (or later)

Q9. Precision-Recall Tradeoff Under Threshold Shift

Problem

A fraud detection classifier is tested at different decision thresholds:

Threshold	Precision	Recall	F1-Score
0.3	0.62	0.94	?
0.5	0.81	0.76	?
0.7	0.93	0.51	?

(i) Compute F1-Score at all three thresholds.

(ii) Which threshold maximizes F1? Justify.

(iii) For fraud detection (missing fraud is costly), which threshold is preferred and why?

Step-by-Step Solution

Step 1 — F1 at Each Threshold:

F1(0.3) = 2×0.62×0.94/(0.62+0.94) = 1.1656/1.56 = 0.7472 = 74.72%

F1(0.5) = 2×0.81×0.76/(0.81+0.76) = 1.2312/1.57 = 0.7843 = 78.43%

F1(0.7) = 2×0.93×0.51/(0.93+0.51) = 0.9486/1.44 = 0.6588 = 65.88%

F1: Threshold 0.3 = 74.72% | Threshold 0.5 = 78.43% | Threshold 0.7 = 65.88%

Step 2 — Highest F1 Threshold:

Threshold 0.5 yields the highest F1 = 78.43%, balancing precision and recall.

Optimal F1 threshold = 0.5 (F1 = 78.43%)

Step 3 — Fraud Detection Preference:

In fraud detection, FALSE NEGATIVES (missing fraud) are far costlier than false positives.

Lower threshold → higher Recall. At threshold 0.3, Recall = 94% (catches 94% of fraud).

We accept lower precision (0.62) to maximize fraud capture rate.

Preferred threshold = 0.3 (Recall = 94%) — minimizes missed fraud cases

Q10. Full Generalization Analysis — Combined Question

Problem

A deep learning model is trained on an image classification task (5 epochs checkpoint):

Epoch	Train Loss	Val Loss	Train Acc	Val Acc
10	0.82	0.85	71%	70%
30	0.41	0.44	84%	83%
60	0.18	0.20	92%	91%
90	0.07	0.35	97%	81%
120	0.02	0.68	99%	74%

(i) Identify epoch ranges for underfitting, good fit, and overfitting.

(ii) By what % has validation loss worsened from epoch 60 to 120?

(iii) What is the best epoch to stop training? Justify with numbers.

(iv) Suggest a numerical early stopping rule based on this data.

Step-by-Step Solution

Step 1 — Epoch Region Classification:

Epoch 10–30: Both train and val accuracy are rising together (71→84%, 70→83%). Gap ≈ 1%. → UNDERFITTING (model still learning)

Epoch 30–60: Both losses falling (0.41→0.18, 0.44→0.20). Gap stable at ~0.02. → GOOD FIT

Epoch 60–120: Train loss falls to 0.02 but val loss rises 0.20→0.68. Val acc drops 91→74%. → OVERFITTING

Underfitting: Epoch 10–30 | Good Fit: Epoch 30–60 | Overfitting: Epoch 60–120

Step 2 — Validation Loss Worsening (60 → 120):

% Worsening = [(0.68 − 0.20) / 0.20] × 100 = [0.48/0.20] × 100 = 240%

Validation loss worsened by 240% from epoch 60 to 120

Step 3 — Best Stopping Epoch:

Epoch 60 has the lowest validation loss (0.20) AND highest validation accuracy (91%).

After epoch 60, val loss increases in every subsequent checkpoint.

Best stopping epoch = 60 (Val Loss = 0.20, Val Acc = 91%)

Step 4 — Early Stopping Rule:

At epoch 60 (good fit): val loss = 0.20, train loss = 0.18 → gap = 0.02

At epoch 90 (overfit starts): val loss = 0.35, gap = 0.28

Rule: Stop if val loss gap > 0.10 and val loss is increasing for 2+ checkpoints.

Early Stopping Rule: Trigger if (Val Loss − Train Loss) > 0.10 for 2 consecutive checkpoints