Instructor Solutions Guide

Discovery Worksheet: Introduction to ML & AI

Expected Answers, Common Responses, and Discussion Prompts

Instructor Notes

Purpose: This guide provides expected answers, common student misconceptions, and discussion prompts for each discovery.

Time Allocation:

- Discovery 1 (Overfitting): 12-15 minutes
- Discovery 2 (K-Means): 15-18 minutes
- Discovery 3 (Boundaries): 15-18 minutes
- Discovery 4 (Gradient): 10-12 minutes
- Discovery 5 (GANs): 10-12 minutes
- Discovery 6 (PCA): 10-12 minutes
- Reflection: 5 minutes

Total: 75-90 minutes

Discovery 1: The Overfitting Paradox - SOLUTIONS

Expected Answers

Task 1: Training Errors

- Model A (red line): ∼45 (high - constant prediction misses variation)
- Model B (green curve): ∼12 (medium - follows trend)
- Model C (purple wiggly): ∼0 (near perfect - hits every point)
- Lowest training error: **Model C**

Task 2: Test Errors

- Model A: ∼48 (similar to training)
- Model B: ∼15 (slight increase from training)
- Model C: ∼67 (huge increase - wildly wrong predictions)
- Lowest test error: **Model B**

Task 3: The Paradox

Expected discovery: "Model C memorizes the training data instead of learning the pattern. It fits noise, not signal. When new data comes, the noise is different, so predictions are terrible."

Common student responses:

- "Model C is trying too hard" → Good intuition! Connect to "overfitting"
- "Model C doesn't generalize" → Excellent! This is the key term
- "Model C is cheating" → Interesting framing, but clarify it's not intentional

Task 4: Trade-off Plot

Students should plot points approximately at:

- Model A: (45, 48) - high bias corner
- Model B: (12, 15) - sweet spot
- Model C: (0, 67) - high variance corner

Pattern: U-shaped relationship - lowest training error does NOT mean lowest test error.

Task 5: Model D Prediction

- Training error: ∼0 (even more complex, still memorizes)
- Test error: >67 (even worse than Model C - more complex = worse generalization)

Key Insights (Expected)

- Model A: "underfitting" or "high bias" (too simple)
- Model C: "overfitting" or "high variance" (too complex)
- Model B: "balanced" or "just right" (optimal complexity)

Common Misconceptions

1. **"More complex is always better"**
   Address: Show that Model C fails dramatically on test data. Complexity must match data structure.
2. **"Training error is what matters"**
   Address: Emphasize: We care about predictions on NEW data, not memorizing old data.
3. **"Model B just got lucky"**
   Address: This is systematic - balanced complexity consistently wins.

Discussion Prompts

- "If you only saw training error, which model would you choose? Why is that dangerous?"
- "In real life, do you memorize facts or learn patterns? Which is more useful?"
- "Where else have you seen the principle: 'simple enough to generalize, complex enough to capture reality'?"

Discovery 2: The Moving Centers Algorithm - SOLUTIONS

Expected Answers

Task 1: Center Movements

- Red cluster point count in Step 1: 8 points
- Approximate center in Step 2: $(1.0, 6.0)$ (mean of assigned points)
- Movement distance: $\sqrt{(2-1)^2 + (7-6)^2} \approx 1.4$ units

Task 2: Within-Cluster Variance
Sample calculations (will vary by which points students pick):

- Point 1 at $(0.8, 6.2)$ to center $(1, 6)$: distance ≈ 0.28
- Point 2 at $(1.2, 5.8)$ to center: distance ≈ 0.28
- Point 3 at $(1.5, 6.3)$ to center: distance ≈ 0.58
- Average squared distance: $(0.28^2 + 0.28^2 + 0.58^2)/3 \approx 0.17$

Task 3: Variance Reduction
From chart (approximate values):

- Step 0: 156.3
- Step 1: 89.2
- Step 2: 78.4
- Step 5: 78.4 (converged)

Variance is **decreasing** - algorithm is optimizing!
Task 4: Convergence Detection
Expected answer: "The centers stop moving. When centers don't change position between iterations, the algorithm has converged."
Alternative good answers:

- "Variance stops decreasing"
- "Point assignments stop changing"
- "The distance centers move becomes very small (near zero)"

Task 5: Discover the Rules
Rule 1: Each point **joins the nearest center** (or "chooses closest center")
Rule 2: Each center **moves to the average position of its points** (or "becomes the mean of its cluster")
Task 6: Optimization Objective
The algorithm minimizes: **total within-cluster variance** (or "sum of squared distances from points to their centers")

Key Insights (Expected)

- K = number of **clusters** or **groups**
- Means = **average** or **centroid** position
- Minimizes **variance** or **distance** within clusters

Common Misconceptions

1. **"Centers are data points"**
   Address: Centers can be anywhere in space, not necessarily at existing points.
2. **"K-means always finds the best solution"**
   Address: Different random starts can give different results (local optima).
3. **"You must know K in advance"**
   Address: True limitation! Need domain knowledge or validation methods.

Discussion Prompts

- "What would happen if we started with different random centers?"
- "How would you choose K for a new problem?"
- "Can you think of real-world applications where finding groups automatically would be useful?"

Discovery 3: The Impossible Separation - SOLUTIONS

Expected Answers

Task 1: Linear Boundary Errors

- Dataset A: $0/30 = 0\%$ (perfect separation possible)
- Dataset B: $3/33 = 9\%$ (a few outliers)
- Dataset C: $\sim 8/30 = 27\%$ (circular pattern, linear fails)
- Dataset D: $\sim 15/30 = 50\%$ (XOR, cannot do better than random)

Datasets A and B can be (nearly) perfectly separated with straight line.

Task 2: Mathematical Proof

Point classifications:

- (1,1) Red: $a + b + c > 0$
- (1,9) Blue: $a + 9b + c < 0$
- (9,1) Blue: $9a + b + c < 0$
- (9,9) Red: $9a + 9b + c > 0$

Key insight: Adding the two blue inequalities gives $10a + 10b + 2c < 0$, which contradicts the red requirements. Mathematical impossibility proven.

Task 3: Nonlinear Solutions

Dataset C:

- Boundary type: **Circle**
- Equation: $x^2 + y^2 = 9$ (or similar radius)

Dataset D:

- Line 1: $x = 5$
- Line 2: $y = 5$
- Combined rule: "Red if ($x < 5$ AND $y < 5$) OR ($x > 5$ AND $y > 5$)"

Task 4: When Linearity Fails

Expected answers:

- Linear works when: "Classes can be separated by a straight line/plane"
- Linear fails when: "Data has curved boundaries, circular patterns, or XOR structure"
- Solution: "Use multiple lines/boundaries, nonlinear functions, or neural networks"

Common Misconceptions

1. **"XOR is just hard, not impossible"**
   Address: Show the mathematical proof - it's provably impossible for single line.
2. **"We should just try more lines"**
   Address: Correct intuition! This leads to neural networks (multiple layers).
3. **"Nonlinear models are always better"**
   Address: No - they can overfit (connects to Discovery 1). Use simplest model that works.

Discussion Prompts

- "Why is XOR called the 'impossible problem' for perceptrons?"
- "If one line fails, how could we combine TWO lines to solve XOR?"
- "Where in the real world might you encounter non-linearly separable data?"

Discovery 4: The Optimization Landscape - SOLUTIONS

Expected Answers

Task 1: Read the Terrain

- Path A starts: $(2, 8)$
- Path A ends with error: ~ 5.2 (local minimum)
- Path B starts: $(7, 8)$
- Path B ends with error: ~ 6.1 (different local minimum)
- Same minimum? **No** (different valleys)
- Global minimum error: ~ 3.8

Task 2: Calculate Gradients
Reading from contours:

- $E(3.0, 7) \approx 6.5$
- $E(3.5, 7) \approx 6.3$
- $E(3, 7.5) \approx 6.7$

Gradients:

- $\frac{\partial E}{\partial x} \approx \frac{6.3-6.5}{0.5} = -0.4$
- $\frac{\partial E}{\partial y} \approx \frac{6.7-6.5}{0.5} = 0.4$
- Descent direction: $(+0.4, -0.4)$ (opposite of gradient)

Task 3: Step Size Experiments
Too LARGE:

- Problem: "Overshoot the minimum, bounce around"
- Risk: "Divergence, never converges, unstable"

Too SMALL:

- Problem: "Very slow convergence, takes forever"
- Risk: "Gets stuck in local minimum, computationally expensive"

Optimal: "Start with larger steps, decrease over time" or "adaptive learning rate"
Task 4: Local vs Global
Expected answer: "Path A followed the gradient downhill and got trapped in the nearest valley. The gradient always points to the nearest minimum, not necessarily the best one."
Escape strategies:

- "Random restart from different location"
- "Momentum to jump over small hills"
- "Simulated annealing (occasionally accept uphill moves)"
- "Multiple initializations and pick best result"

Task 5: Optimization Strategy
Parameters = **model weights, coordinates**
Learning rate = **step size, how far to move**
Gradient = **slope, direction of steepest ascent**
If gradient positive: move **left** (decrease parameter)
If gradient negative: move **right** (increase parameter)

Common Misconceptions

1. **"Gradient descent always finds the best solution"**
   Address: No - only finds local minimum. Global minimum not guaranteed.
2. **"Bigger learning rate is always faster"**
   Address: Too big causes overshooting. Need balance.
3. **"All optimization landscapes are smooth"**
   Address: Real problems can have plateaus, saddle points, discontinuities.

Discussion Prompts

- "Imagine hiking down a mountain in fog - you can only see your feet. What strategy would you use?"
- "Why might machine learning need to train the same model multiple times with different starting points?"
- "What's the connection between this landscape and Discovery 1's overfitting problem?"

Discovery 5: The Two-Player Game - SOLUTIONS

Expected Answers

Task 1: Quality Tracking

- Epoch 1: 12% (noise blob)
- Epoch 10: 35% (improvement: 23%)
- Epoch 50: 68% (improvement: 33%)
- Epoch 100: 94% (improvement: 26%)
- Total improvement: 82%

Task 2: Loss Dynamics

- Generator winning: Epochs 50-100 (loss decreasing faster)
- Discriminator winning: Epochs 1-20 (loss stable while G struggles)
- Equilibrium: Around epoch 60-70
- At equilibrium: Both losses $\approx 2 - 3$, roughly equal

Task 3: Game Theory Table

G	D	G success	D success	Winner
0.2	0.8	0.16 (16%)	0.64 (64%)	Discriminator
0.5	0.5	0.25 (25%)	0.25 (25%)	Tie (equilibrium)
0.8	0.2	0.64 (64%)	0.16 (16%)	Generator
0.9	0.1	0.81 (81%)	0.09 (9%)	Generator

Nash equilibrium: $G = 0.5$, $D = 0.5$ (both equally successful)

Task 4: Training Evolution

- Steepest improvement: Epochs 1-30
- Slows after: Epoch 50
- Reach 100%? **No** - discriminator improves too, making task harder

Task 5: Adversarial Insight

Why both improve: "Generator gets better by trying to fool discriminator. Discriminator gets better by learning to detect fakes. Each improvement forces the other to improve. Competition drives mutual learning."

Generator alone: "No feedback, no improvement. Generator needs discriminator to tell it what's wrong."

Analogy: Generator = **art student**, Discriminator = **art teacher/critic**

Common Misconceptions

1. **"One player should win completely"**
   Address: Equilibrium is the goal - both at 50/50 means generator creates perfect fakes.
2. **"Training is competitive, so one fails"**
   Address: Both improve! Competition drives mutual growth (cooperative-competitive).
3. **"Generator loss should reach zero"**
   Address: At equilibrium, discriminator is random (50/50) on real vs fake.

Discussion Prompts

- "Why is this called 'adversarial' if both players benefit?"
- "Can you think of other situations where competition leads to improvement?"
- "What happens if discriminator trains much faster than generator?"

Discovery 6: The Dimensionality Revelation - SOLUTIONS

Expected Answers

Task 1: Variance Calculations

From 3D plot:

- Var(X) ≈ 4.2
- Var(Y) ≈ 4.1
- Var(Z) ≈ 1.05
- Total ≈ 9.35

From 2D projection:

- Var(PC1) ≈ 8.3 (89% of total)
- Var(PC2) ≈ 0.9 (10% of total)
- Total retained: 99%

Information lost: 1%

Task 2: Reconstruction Error

Example point:

- Original: $(2.0, 2.0, 1.0)$
- Projected: $(2.8, 0.2)$
- Reconstructed: $(2.0, 2.0, 1.0)$
- Error: ≈ 0 (very small)

Average error from chart: ~ 0.12

Reconstruction is **excellent** - only 1% information loss.

Task 3: Compression Analysis

Dimensions	Info Retained	Storage Saved	Good?
3	100%	0%	N/A
2	99%	33%	YES (great trade-off)
1	89%	67%	Maybe (depends on use)

Compression for 2D:

- Original: 150 numbers
- Compressed: 100 numbers
- Ratio: $100/150 = 67\%$ (33% reduction)

Task 4: When PCA Works

Expected answer: "Points lie near a 2D plane in 3D space. Most variation is along two diagonal directions, very little variation perpendicular to the plane. Data has intrinsic low-dimensional structure."

Random scatter: "PCA would not compress well - would need all 3 dimensions to represent the data accurately."

Task 5: Principal Components

From scree plot:

- PC1: 89%
- PC2: 10%
- PC3: 1%
- Sum: 100%

Elbow suggests keeping: **2 components**

Key Insights (Expected)

- PCA finds directions of **maximum** variance
- Data near lower-dimensional **subspace/plane** compresses well
- Trade-off: Storage vs **information loss/accuracy**

Common Misconceptions

1. **"PCA creates new features from nothing"**
   Address: No - PCA finds existing structure. It rotates axes to align with variance.
2. **"First PC is always the best"**
   Address: Depends on data. For random data, no PC is significantly better.
3. **"PCA always compresses to 2D for visualization"**
   Address: Can keep any number of components based on variance retained threshold.

Discussion Prompts

- "Why is this data compressible from 3D to 2D with almost no loss?"
- "If you had 100 features, how would you decide how many PCs to keep?"
- "Can you think of applications where reducing dimensions would be useful?"

Assessment Rubric

Use this rubric to gauge student understanding:

Excellent Understanding (90-100%)

- Correctly calculates numerical answers
- Explains patterns in own words
- Makes connections across discoveries
- Predicts outcomes for new scenarios
- Asks sophisticated "what if" questions

Good Understanding (75-89%)

- Most calculations correct
- Identifies main patterns
- Answers conceptual questions adequately
- Makes some cross-discovery connections

Developing Understanding (60-74%)

- Some calculation errors
- Recognizes patterns with prompting
- Struggles with conceptual explanations
- Limited connections across topics

Needs Support (¡60%)

- Frequent calculation errors
- Cannot articulate patterns independently
- Requires significant guidance
- Use 1-on-1 discussion to build foundation

Class Discussion Guide

Opening (5 minutes)

"Before we start the lecture, let's share discoveries. Turn to your neighbor and compare answers for Discovery 1, Task 3 - why does Model C fail on test data?"

Listen for: students using words like "memorization," "overfitting," "doesn't generalize"

Mid-Lecture Checkpoints

After introducing each formal concept, connect to worksheet:

When introducing bias-variance:

"Who discovered the paradox in Chart 1? You already found the bias-variance tradeoff before I named it!"

When introducing K-means:

"In Discovery 2, what two rules did you discover? [Assignment and Update] Exactly - that's the K-means algorithm."

When introducing neural networks:

"Discovery 3 showed you XOR is impossible for single line. How did you solve it? [Two lines] That's precisely what neural networks do - combine multiple simple boundaries."

When introducing optimization:

"Chart 4 showed different starting points leading to different valleys. What's the solution? [Random restarts] Used in practice constantly."

When introducing GANs:

"Your Nash equilibrium table showed equilibrium at 50/50. What does that mean for the generator? [Perfect fakes] Exactly!"

When introducing PCA:

"Discovery 6 showed 99% info retained with 33% storage savings. When is that worth it? [When storage/speed matters, small info loss OK]"

Common Questions and Answers

Q: "How do we know which model complexity is right?"

A: Cross-validation (split data, test on held-out portion) - systematic version of Discovery 1.

Q: "Does K-means always find the same clusters?"

A: No - depends on initialization. Run multiple times, pick best result (lowest variance).

Q: "Can any nonlinear problem be solved with enough lines?"

A: Yes! Universal approximation theorem - enough neurons can approximate any function.

Q: "Why not always use the most complex model?"

A: Discovery 1 showed this fails! Overfitting, computational cost, interpretability.

Q: "Is Nash equilibrium always 50/50?"

A: For this game formulation, yes. Other GAN variants have different equilibria.

Q: "How much variance should PCA retain?"

A: Common thresholds: 90-95%, but depends on application. Visualization: 2-3 PCs. Compression: as low as tolerable.

Closing (5 minutes)

"Look at your three most important insights from the final reflection. Turn to your neighbor - did you write similar things or different? Why might different people discover different patterns in the same charts?"

This reinforces: Multiple valid interpretations, discovery is personal, formal lecture codifies shared understanding.

End of Instructor Guide