AMATH 445/645 – Practice Problems for Lectures 1–4

1. You train a model and achieve:

   - Training accuracy = 99%
   - Test accuracy = 58%

   (a) What is happening?
   (b) Name two ways to fix this issue.

   (a) Overfitting; the model has memorized the training data but fails to generalize to unseen data.
   (b) Possible fixes:

   - Get more training data
   - Add regularization (L1/L2)
   - Implement early stopping
   - Reduce model complexity
   - Apply data augmentation
   - Add dropout (for neural networks)

2. Match each scenario to the correct diagnosis (underfitting, overfitting, good fit):

   (a) High training error and high validation error
   (b) Low training error, but validation error starts increasing after some epochs
   (c) Both training and validation errors are low and close to each other

   (a) Underfitting; model too simple to capture patterns.
   (b) Overfitting; model memorizing training data.
   (c) Good fit; model generalizes well.

3. Identify the machine learning paradigm for each scenario:

   (a) Predicting house prices from features like size, bedrooms, location
   (b) Detecting credit card fraud without any labeled transactions
   (c) AlphaGo learning to play Go through self-play
   (d) A language model predicting the next word in a sentence
   (e) Medical imaging with 100 labeled MRI scans and 10,000 unlabeled scans

   (a) Supervised learning (regression)
   (b) Unsupervised learning (anomaly detection)
   (c) Reinforcement learning
   (d) Self-supervised learning
   (e) Semi-supervised learning

4. Explain the difference between classification and regression. Provide one real-world example of each.

   Classification predicts discrete categories or class labels (e.g., spam vs. not spam, digit recognition 0-9).
   Regression predicts continuous numerical values (e.g., temperature prediction, stock prices).

5. What key assumption about the covariance matrices in Linear Discriminant Analysis (LDA) leads to a linear decision boundary?

   LDA assumes that all classes share the same covariance matrix ($\Sigma_k = \Sigma$ for all classes k). This causes the quadratic terms in the discriminant function to cancel out when comparing classes, resulting in a linear boundary.

6. Compare and contrast LDA and logistic regression in terms of what they model and their assumptions.

   LDA models the class-conditional densities $P(x|y)$ (generative approach) and assumes Gaussian distributions with shared covariance.
   Logistic regression models the posterior probability $P(y|x)$ directly (discriminative approach) and makes no assumptions about the distribution of x.

7. How does Quadratic Discriminant Analysis (QDA) differ from LDA, and what effect does this have on the decision boundary?

   QDA allows each class to have its own covariance matrix (Σ_k), while LDA assumes a shared covariance matrix (Σ). This makes QDA's decision boundary quadratic rather than linear, allowing it to capture more complex relationships but requiring more parameters to estimate.

8. You split your data: 70% training, 15% validation, 15% test.

   (a) What is the purpose of each split?

   (b) Why must the test set never be used during training or model selection?

   (a) We have,

   - Training set: Used to learn model parameters (weights)
   - Validation set: Used to tune hyperparameters and select the best model
   - Test set: Used only once at the end to evaluate final model performance

   (b) Using the test set during training would leak information about test data into model development, giving an overly optimistic (biased) estimate of how the model will perform on truly unseen data.

9. What is early stopping in neural network training? How does it help with generalization?

   Early stopping halts training when validation error begins to increase, even if training error continues to decrease. This prevents overfitting by stopping before the model starts memorizing noise in the training data, preserving its ability to generalize to unseen data.

10. Let the true data distribution be $P(x, y)$. Show that the classifier minimizing the probability of misclassification is

$$\hat{y}(x) = \arg\max_k P(y = k|x).$$

The probability of error at a point x when predicting class $\hat{y}$ is

$$P(\text{error}|x) = 1 - P(y = \hat{y}|x).$$

The overall probability of error is $\mathbb{E}_x[P(\text{error}|x)]$. This expectation is minimized when $P(y = \hat{y}|x)$ is maximized for each x. Therefore the optimal classifier chooses the class with largest posterior:

$$\hat{y}(x) = \arg\max_k P(y = k|x).$$

This is called the Bayes optimal classifier.

11. In Linear Discriminant Analysis (LDA), the decision boundary between two classes can be written as

$$\omega^T x + b = 0,$$

which defines a hyperplane in $\mathbb{R}^d$. Show that ω is orthogonal (normal) to this hyperplane.

Let the hyperplane be

$$H = \{x \in \mathbb{R}^d : \ \omega^T x + b = 0\}.$$

Pick any two points $x_1, x_2 \in H$. Then

$$\omega^T x_1 + b = 0, \qquad \omega^T x_2 + b = 0.$$

Subtracting gives

$$\omega^T (x_1 - x_2) = 0.$$

But $x_1 - x_2$ is an arbitrary direction vector lying *within* the hyperplane (it is tangent to H). Since its dot product with ω is zero for any such direction, ω is orthogonal to every direction in the hyperplane. Hence ω is a normal (orthogonal) vector to the hyperplane.

12. Logistic regression models the probability of class 1 as:

$$P(y = 1|x) = \sigma(\beta^T x) = \frac{1}{1 + e^{-\beta^T x}}$$

Show that maximizing the likelihood is equivalent to minimizing the binary cross-entropy loss.

For binary classification with $y \in \{0, 1\}$, the likelihood for N samples is:

$$L(\beta) = \prod_{i=1}^{N} P(y_i|x_i) = \prod_{i=1}^{N} (\sigma(\beta^T x_i))^{y_i} (1 - \sigma(\beta^T x_i))^{1-y_i}$$

Take the log:

$$\ell(\beta) = \sum_{i=1}^{N} \left[y_i \log \sigma(\beta^T x_i) + (1 - y_i) \log(1 - \sigma(\beta^T x_i)) \right]$$

The negative log-likelihood is:

$$-\ell(\beta) = -\sum_{i=1}^{N} \left[y_i \log p_i + (1 - y_i) \log(1 - p_i) \right]$$

where $p_i = \sigma(\beta^T x_i)$. This is exactly the binary cross-entropy loss. Therefore maximizing likelihood is equivalent to minimizing cross-entropy.

13. True or False? Justify your answer.

(a) If training error is low and validation error is high, the model is underfitting.

(b) Logistic regression assumes the data in each class follows a Gaussian distribution.

(c) In unsupervised learning, the model is trained on unlabeled data to find hidden patterns.

(d) The test set can be used multiple times during model development to fine-tune performance.

(a) False. This describes overfitting. Underfitting would show high error on both sets.
(b) False. That's LDA's assumption. Logistic regression makes no assumptions about feature distributions.
(c) True. Unsupervised learning finds patterns (clusters, anomalies, latent structure) without labels.
(d) False. Test set should only be used once at the very end. Multiple uses would lead to overfitting to test data.

14. You're building a system to classify skin lesions as benign or malignant. You have 200 labeled images from one hospital and 50,000 unlabeled images from multiple hospitals. Which machine learning paradigm would you use and why?

    Semi-supervised learning would be most appropriate. The 200 labeled images provide initial supervision, while the 50,000 unlabeled images can help the model learn the underlying data distribution and improve generalization. This approach leverages the abundance of unlabeled data while using the limited labeled data effectively.

15. Linear regression has a closed-form solution (normal equations) but logistic regression does not. Explain why.

    Linear regression minimizes a quadratic loss function $J(\beta) = \|X\beta - y\|^2$, which is convex and quadratic in β. Setting $\nabla J(\beta) = 0$ yields linear equations $(X^T X)\beta = X^T y$ that can be solved analytically.
    Logistic regression minimizes the cross-entropy loss, which contains the sigmoid function $\sigma(\beta^T x)$. This makes $J(\beta)$ nonlinear and non-quadratic in β, so $\nabla J(\beta) = 0$ cannot be solved analytically. We must use iterative optimization methods like gradient descent.

16. The gradient descent update rule is

$$\beta^{(k+1)} = \beta^{(k)} - \eta \nabla J(\beta^{(k)}).$$

    Explain geometrically why we move in the negative gradient direction.

    The gradient $\nabla J(\beta)$ points in the direction of steepest increase of the function at point β. Therefore, moving in the opposite direction $-\nabla J(\beta)$ gives the direction of steepest decrease, which brings us toward a local minimum most efficiently.

17. Describe what happens if the learning rate η is:

    (a) too small
    (b) too large
    (c) adaptive (changing during training)

    (a) Too small: Convergence is very slow; may get stuck in plateaus.
    (b) Too large: The algorithm overshoots the minimum, may oscillate or diverge.
    (c) Adaptive: Starts with larger steps for faster progress, then reduces step size to fine-tune near the minimum (e.g., learning rate schedules).

18. Compare batch gradient descent, stochastic gradient descent (SGD), and mini-batch gradient descent.

- **Batch GD**: Uses all N samples to compute gradient each step. Accurate but computationally expensive for large datasets.
- **SGD**: Uses one random sample per update. Fast, noisy updates that can escape local minima but may oscillate.
- **Mini-batch GD**: Uses a small random subset. Balances efficiency and stability; most common in practice.

19. Suppose $y_i = 1$ but $\sigma(\beta^T x_i) = 0.1$. What direction does the gradient update move β relative to x_i?

The error is: $\sigma(\beta^T x_i) - y_i = 0.1 - 1 = -0.9$ (negative).
Gradient descent update: $\beta \leftarrow \beta - \eta \nabla J(\beta)$
Since ∇J contains the error term $(-0.9)x_i$, subtracting the gradient adds $(0.9\eta)x_i$ to β.
Thus β moves in the positive direction of x_i, increasing $\beta^T x_i$ and raising the predicted probability toward 1.

20. Logistic regression satisfies

$$\log \frac{P(y=1|x)}{P(y=0|x)} = \beta_0 + \sum_{j=1}^{d} \beta_j x_j.$$

Interpret the meaning of coefficient β_j.

β_j measures the change in the log-odds of class 1 when feature x_j increases by one unit, holding all other features fixed.
If $\beta_j > 0$: Increasing x_j increases the probability of class 1.
If $\beta_j < 0$: Increasing x_j decreases the probability of class 1.
If $\beta_j = 0$: Feature x_j has no effect on the classification.

21. Show that logistic regression predicts class 1 exactly when $\beta^T x \geq 0$.

Class 1 is predicted when $P(y=1|x) \geq 0.5$.

$$P(y=1|x) = \sigma(\beta^T x) = \frac{1}{1+e^{-\beta^T x}} \geq 0.5$$

Multiply both sides by $1 + e^{-\beta^T x} > 0$:

$$1 \geq 0.5(1+e^{-\beta^T x}) \implies 2 \geq 1 + e^{-\beta^T x} \implies 1 \geq e^{-\beta^T x}$$

Taking natural log (monotonic):

$$0 \geq -\beta^T x \implies \beta^T x \geq 0$$

Thus the decision boundary is the hyperplane $\beta^T x = 0$.

22. Given one datapoint:

$$x = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad y = 0, \quad \beta^{(0)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \eta = 0.1$$

(a) Compute $\beta^T x$, the predicted probability, and the error.
(b) Compute the gradient and perform one gradient descent update.
(c) Has the probability moved in the correct direction?

(a) $\beta^T x = 0$, $\sigma(0) = 0.5$, error $= 0.5 - 0 = 0.5$

(b) Gradient $= (0.5 - 0) \cdot \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1.0 \\ 1.5 \end{bmatrix}$ Update:

$$\beta^{(1)} = \beta^{(0)} - 0.1 \begin{bmatrix} 1.0 \\ 1.5 \end{bmatrix} = \begin{bmatrix} -0.1 \\ -0.15 \end{bmatrix}$$

(c) New $\beta^T x = -0.1(2) + -0.15(3) = -0.2 - 0.45 = -0.65$, $\sigma(-0.65) \approx 0.34$
Since $y = 0$, we want probability near 0. The probability decreased from 0.5 to 0.34, so yes, moving in the correct direction.

23. Why does logistic regression struggle to classify spin configurations near the critical temperature T_c in the Ising model?

Near T_c, the system exhibits critical fluctuations. Ordered and disordered phases become indistinguishable at local scales, causing significant overlap between classes. Logistic regression is a linear classifier that finds a linear decision boundary, but near T_c the true decision boundary becomes highly nonlinear due to complex spin correlations. Therefore logistic regression cannot separate the classes effectively near the phase transition.

24. A data scientist is building a regression model to predict house prices. She considers two different loss functions:

$$L_1(t, y) = |t - y|, \qquad L_2(t, y) = (t - y)^2.$$

(a) Suppose you observe residuals:

$$[-1,\ 0,\ 2,\ -4].$$

Compute the total L_1 and L_2 losses.

(b) Briefly explain the main difference in how L_1 and L_2 losses handle outliers.

(a) We have

$$L_1 = |-1| + |0| + |2| + |-4| = 1 + 0 + 2 + 4 = 7$$
$$L_2 = (-1)^2 + 0^2 + 2^2 + (-4)^2 = 1 + 0 + 4 + 16 = 21$$

(b) L_2 squares the residuals, so large errors are penalized much more heavily, making it sensitive to outliers. L_1 grows linearly and therefore is more robust to outliers.

25. For a point x, the signed distance to the decision boundary $w^T x + b = 0$ is

$$\frac{w^T x + b}{\|w\|}.$$

Using this, show that the margin (distance between the two supporting hyperplanes) equals $\frac{2}{\|w\|}$.

The supporting hyperplanes for a hard-margin SVM are defined by $w^T x + b = \pm 1$. For support vectors (points on these boundaries), we have $|w^T x_i + b| = 1$.

The distance from the decision boundary ($w^T x + b = 0$) to the positive supporting hyperplane ($w^T x + b = 1$) is:

$$\frac{1}{\|w\|}$$

Similarly, the distance to the negative supporting hyperplane ($w^T x + b = -1$) is:

$$\frac{|-1|}{\|w\|} = \frac{1}{\|w\|}$$

Therefore the total margin (distance between the two supporting hyperplanes) is:

$$\frac{1}{\|w\|} + \frac{1}{\|w\|} = \frac{2}{\|w\|}$$

26. Explain geometrically why maximizing the margin $\frac{2}{\|w\|}$ is equivalent to minimizing $\frac{1}{2}\|w\|^2$.

    The margin is inversely proportional to $\|w\|$: margin $= \frac{2}{\|w\|}$.
    To maximize the margin, we want $\|w\|$ as small as possible. Minimizing $\frac{1}{2}\|w\|^2$ is a convenient convex formulation that achieves this goal while simplifying the optimization (the $\frac{1}{2}$ is added for mathematical convenience when taking derivatives).

27. The hard-margin SVM solves the constrained optimization problem:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{subject to } y_i(w^T x_i + b) \geq 1 \text{ for } i = 1, \ldots, N.$$

    Explain why minimizing $\|w\|$ (maximizing the margin) leads to better generalization on unseen data.

    A larger margin reduces the model's sensitivity to small perturbations in the input data. The decision boundary is placed as far as possible from the training points, making it more robust to noise and less likely to overfit. This inductive bias toward simplicity helps the model perform better on unseen data.

28. From the Lagrangian optimization of the hard-margin SVM, we obtain:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

    where $\alpha_i \geq 0$ are Lagrange multipliers. Explain why only support vectors matter for the final classifier.

    The Karush-Kuhn-Tucker (KKT) conditions require complementary slackness:

$$\alpha_i [y_i(w^T x_i + b) - 1] = 0$$

    For points strictly inside the margin ($y_i(w^T x_i + b) > 1$), the term in brackets is positive, forcing $\alpha_i = 0$. Only for support vectors, which lie exactly on the margin ($y_i(w^T x_i + b) = 1$), can $\alpha_i > 0$.
    Thus w is a linear combination only of the support vectors. Non-support vectors have $\alpha_i = 0$ and contribute nothing to the classifier.

29. Why do we typically solve the dual formulation of the SVM optimization problem rather than the primal?

(a) The dual formulation expresses the problem entirely in terms of dot products $x_i^T x_j$, which allows us to apply the kernel trick.

(b) The constraints in the dual ($\alpha_i \geq 0$, $\sum \alpha_i y_i = 0$) are simpler than the primal constraints.

(c) The dual is a convex quadratic programming problem with a unique solution.

(d) Many support vectors have $\alpha_i = 0$, making the solution sparse.

30. Soft-margin SVM introduces slack variables $\xi_i \geq 0$ to handle non-separable data:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0.$$

Interpret the meaning of:

(a) $\xi_i = 0$

(b) $0 < \xi_i < 1$

(c) $\xi_i = 1$

(d) $\xi_i > 1$

(a) $\xi_i = 0$: Point is correctly classified and lies on or outside the margin boundary.

(b) $0 < \xi_i < 1$: Point lies inside the margin (between the decision boundary and the margin) but is still on the correct side — a "margin violation" but not misclassified.

(c) $\xi_i = 1$: Point lies exactly on the decision boundary.

(d) $\xi_i > 1$: Point is misclassified (on the wrong side of the decision boundary).

31. The soft-margin SVM objective is:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i$$

Explain the role of the regularization parameter C. What happens when C is very large? Very small?

C controls the trade-off between maximizing the margin and minimizing training errors.

- **Large** C: High penalty for margin violations. The model focuses on classifying all training points correctly, resulting in a narrower margin and potential overfitting.
- **Small** C: More tolerance for violations. The model prioritizes a wider margin over training accuracy, leading to better generalization but potentially higher training error.
- $C \to \infty$: Approaches hard-margin SVM (if data is separable).
- $C \to 0$: Ignores misclassifications entirely, meaningless model.

32. When would you choose hard-margin SVM over soft-margin SVM, and vice versa?

- **Hard-margin**: Only appropriate when data is perfectly linearly separable AND you are certain there is no noise. Rarely used in practice because real data almost always has overlaps or outliers.

- **Soft-margin**: The practical choice for most real-world problems. Handles noisy data, outliers, and non-separable cases by allowing some misclassifications in exchange for better generalization.

33. Explain the purpose of the kernel trick in SVMs. Why is it so powerful?

    The kernel trick allows SVMs to create nonlinear decision boundaries while still solving a convex optimization problem. It works by implicitly mapping input data to a higher-dimensional feature space where a linear separator exists, without ever computing the transformation explicitly.
    Key advantages:

    (a) Computationally efficient; avoids working in high-dimensional space directly

    (b) Can use very high (even infinite) dimensional feature spaces

    (c) Same optimization algorithm works, just replace dot products with kernel functions

34. The dual SVM optimization involves dot products $x_i^T x_j$. When applying the kernel trick, we replace this with $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$. Why does this substitution work?

    The dual formulation and the final decision function depend on data only through dot products:

    $$f(x) = \sum_{i=1}^{N} \alpha_i y_i (x_i^T x) + b$$

    When we replace x with $\phi(x)$, the dot product becomes $\phi(x_i)^T \phi(x_j)$. By defining a kernel function $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$, we can compute this dot product in feature space without ever constructing $\phi(x)$. This "kernel trick" works as long as K satisfies Mercer's condition (positive semidefinite).

35. Name three common kernel functions and briefly describe when you might use each.

    (a) **Linear kernel**: $K(x_i, x_j) = x_i^T x_j$
    - Use when data is already linearly separable or very high-dimensional (e.g., text classification with many features)
    - Fastest, most interpretable

    (b) **Polynomial kernel**: $K(x_i, x_j) = (x_i^T x_j + c)^d$
    - Use for moderately nonlinear problems
    - Degree d controls flexibility (higher d = more complex)

    (c) **Gaussian (RBF) kernel**: $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$
    - Most popular default choice
    - Can model complex boundaries, universal approximator capability
    - Parameter γ controls influence radius of each point