STAT 479: Study Guide for Exam

Exam will be 6 questions, taken from the following question bank:

My Favorite HW Questions

1. **From HW1: Deriving Backpropagation from MLE**

   Consider a neural network with one hidden layer. The network's output is given by:

   $$\hat{y} = \sigma(w_2 \cdot h),$$

   where:

   - $h = \sigma(w_1 \cdot x)$,
   - $\sigma(z)$ is the sigmoid activation function defined as $\sigma(z) = \frac{1}{1+e^{-z}}$,
   - w_1 and w_2 are weights,
   - x is the input.

   Assume that the training data (x, y) are drawn i.i.d. from a distribution, and the network is trained using Maximum Likelihood Estimation (MLE). For binary classification, the likelihood is given by:

   $$P(y|x, w_1, w_2) = \hat{y}^y (1 - \hat{y})^{1-y},$$

   where $\hat{y}$ is the predicted probability for the positive class.

   (a) What is the **negative** log-likelihood $\mathcal{L}$? You may leave $\hat{y}$ unexpanded here.

   (b) What is the gradient of $\mathcal{L}$ with respect to w_2?
   Hint: The derivative of the sigmoid function is:

   $$\sigma'(z) = \sigma(z)(1 - \sigma(z)).$$

   (c) What is the gradient of $\mathcal{L}$ with respect to w_1?

   (d) **Bonus:** The gradient computations you derived for w_1 and w_2 can be directly applied across all training examples in a mini-batch of size n.

   1. **5 points:** Express the gradient updates for w_1 and w_2 in matrix form, assuming that the input batch is represented as a matrix X of shape (n, d) and output Y of shape $(n, 1)$.
   2. **5 points:** Explain why this matrix formulation is computationally more efficient than computing gradients individually for each training example.

2. **From HW2: MLE for Gaussian Naive Bayes**
Naive Bayes assumes that features $X = (X_1, X_2, \ldots, X_d)$ are conditionally independent given the class Y, and that $X_i|Y \sim \mathcal{N}(\mu_{i,Y}, \sigma^2_{i,Y})$, i.e.

$$P(X|Y) = \prod_{i=1}^{d} P(X_i|Y),$$
$$P(X_i|Y = k) = \frac{1}{\sqrt{2\pi\sigma^2_{i,k}}} \exp\left(-\frac{(X_i - \mu_{i,k})^2}{2\sigma^2_{i,k}}\right).$$

   (a) **Likelihood Function**: Write the likelihood of the observed data $\{X_{i,j}\}_{j:Y_j=k}$, assuming $X_i|Y = k$ is Gaussian. Select the correct expression:

   A. $\prod_{j:Y_j=k} \prod_{i=1}^{d} P(X_{i,j}|Y = k)$

   B. $\prod_{j:Y_j=k} \prod_{i=1}^{d} \frac{1}{2\pi\sigma^2_{i,k}} \exp\left(-\frac{(X_{i,j}-\mu_{i,k})^2}{\sigma^2_{i,k}}\right)$

   C. $\prod_{j:Y_j=k} \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi\sigma^2_{i,k}}} \exp\left(-\frac{(X_{i,j}-\mu_{i,k})^2}{2\sigma^2_{i,k}}\right)$

   D. $\prod_{j:Y_j=k} \prod_{i=1}^{d} P(Y = k) \cdot P(X_{i,j})$

   (b) **Log-Likelihood**: Write the log-likelihood for $\mu_{i,k}$ and $\sigma^2_{i,k}$. For notation convenience, let N_k be the number of samples with label k: $N_k = \sum_{j=1}^{n} \mathbb{I}(Y_j = k)$. Select the correct expression:

   A. $-\frac{N_k}{2} \log(2\pi\sigma^2_{i,k}) - \frac{1}{2\sigma^2_{i,k}} \sum_{j:Y_j=k}(X_{i,j} - \mu_{i,k})^2$

   B. $-\frac{1}{2} \log(2\pi\sigma^2_{i,k}) - \frac{1}{\sigma^2_{i,k}} \sum_{j:Y_j=k}(X_{i,j} - \mu_{i,k})^2$

   C. $\sum_{j:Y_j=k} \log P(X_{i,j}|Y = k) + \log P(Y = k)$

   D. $-\frac{N_k}{2} \log(2\pi) + \frac{\sum_{j:Y_j=k}(X_{i,j}-\mu_{i,k})^2}{2\sigma^2_{i,k}}$

   (c) **MLE for $\mu_{i,k}$**: Derive the MLE for $\mu_{i,k}$. Select the correct estimator:

   A. $\widehat{\mu_{i,k}} = \frac{\sum_{j:Y_j=k} X_{i,j}}{N_k}$

   B. $\widehat{\mu_{i,k}} = \frac{\sum_{j=1}^{n} X_{i,j}}{n}$

   C. $\widehat{\mu_{i,k}} = \frac{\sum_{j:Y_j=k} X^2_{i,j}}{N_k}$

   D. $\widehat{\mu_{i,k}} = \sum_{j:Y_j=k} X_{i,j}$

3. **From HW3: MLE for Bayesian Network**
Consider a simple Bayesian network $A \to B$ with binary variables $A, B \in \{0, 1\}$ and joint distribution $P(A, B) = P(A)P(B|A)$. You have a complete dataset of n observations, where $n_{a,b}$ denotes the number of times $(A = a, B = b)$ occurs.

   Let $\theta_1 = P(A = 1), \theta_2 = P(B = 1 \mid A = 0), \theta_3 = P(B = 1 \mid A = 1)$.

   (a) **Log-Likelihood**: Select the correct log-likelihood $\ell(\theta_1, \theta_2, \theta_3)$:

A. $\ell(\theta_1, \theta_2, \theta_3) = n_{0,0} \ln(1-\theta_1) + n_{0,1} \ln(1-\theta_1)\theta_2 + n_{1,0} \ln \theta_1(1-\theta_3) + n_{1,1} \ln \theta_1\theta_3$

B. $\ell(\theta_1, \theta_2, \theta_3) = n_{0,0} \ln[(1-\theta_1)(1-\theta_2)] + n_{0,1} \ln[(1-\theta_1)\theta_2] + n_{1,0} \ln[\theta_1(1-\theta_3)] + n_{1,1} \ln[\theta_1\theta_3]$

C. $\ell(\theta_1, \theta_2, \theta_3) = n \ln \theta_1 + n_{0,1} \ln \theta_2 + n_{1,1} \ln \theta_3$

D. $\ell(\theta_1, \theta_2, \theta_3) = n_{0,0} \ln \theta_1 + n_{0,1} \ln \theta_2 + n_{1,0} \ln \theta_3$

(b) **MLE for** $P(B = 1|A = 0)$: Derive the MLE for $P(B = 1|A = 0)$ by maximizing the log-likelihood. Select the correct estimator:

A. $\hat{\theta}_{2,MLE} = \frac{n_{0,1}}{n_{0,0}+n_{0,1}}$

B. $\hat{\theta}_{2,MLE} = \frac{n_{0,1}}{n}$

C. $\hat{\theta}_{2,MLE} = \frac{n_{0,0}+n_{0,1}}{n}$

D. $\hat{\theta}_{2,MLE} = \frac{n_{1,1}}{n_{1,0}+n_{1,1}}$

4. **From HW4: MRFs**

Consider a Markov Random Field (MRF) with the following structure:

- Nodes: X_1, X_2, X_3, X_4
- Edges: $(X_1, X_2), (X_2, X_3), (X_3, X_4), (X_4, X_1), (X_1, X_3)$
- The joint probability distribution is given by:

$$P(X_1, X_2, X_3, X_4) \propto \exp\left(\sum_{(i,j)\in E} \theta_{ij} X_i X_j\right)$$

(a) Which of the following is a correct factorization of this MRF?

A. $P(X_1, X_2, X_3, X_4) = \phi(X_1, X_2)\phi(X_2, X_3)\phi(X_3, X_4)\phi(X_4, X_1)$

B. $P(X_1, X_2, X_3, X_4) = \phi(X_1, X_2, X_3)\phi(X_3, X_4)$

C. $P(X_1, X_2, X_3, X_4) = \phi(X_1, X_2)\phi(X_2, X_3)\phi(X_3, X_4)\phi(X_4, X_1)\phi(X_1, X_3)$

D. $P(X_1, X_2, X_3, X_4) = \phi(X_1, X_2, X_3, X_4)$

(b) Why is computing the partition function Z difficult for large MRFs?

A. The partition function requires summing over an exponential number of terms.

B. The partition function depends on the parameters θ_{ij}, which are unknown.

C. The partition function does not exist for undirected graphical models.

D. The partition function is always equal to 1.

(c) Removing edge (X_1, X_3) changes which independence property?

A. $X_1 \perp X_3 \mid \{X_2, X_4\}$

B. $X_1 \perp X_3 \mid X_2$

C. $X_1 \perp X_4 \mid X_2$

D. $X_1 \perp X_2 \mid X_4$

(d) A researcher is using Graphical Lasso to learn the structure of a Markov Random Field (MRF) from data. However, they observe that small changes in the regularization parameter result in large differences in the learned graph structure, with many edges appearing or disappearing unpredictably.

Which of the following is the most likely reason for this instability?

A. The sample size is too small relative to the number of variables, leading to an unstable covariance estimate.

B. The true MRF is not connected, so regularization causes disjoint components to form.

C. Graphical Lasso is not a consistent estimator and always leads to instability.

D. The data is non-Gaussian, violating the assumptions of Graphical Lasso.

5. **From HW5: Deriving EM Updates for a Gaussian Mixture Model (GMM)**

A Gaussian Mixture Model assumes that data points are generated from a mixture of K Gaussian distributions, each with a mean μ_k, covariance Σ_k, and a mixing weight π_k. As introduced in class, the Expectation-Maximization (EM) algorithm iteratively estimates these parameters.

(a) **Setting Up the Model**

We model the data as being drawn from K Gaussian components:

$$p(x|\theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$$

where π_k are the mixing weights summing to 1, and $\mathcal{N}(x|\mu_k, \Sigma_k)$ is a Gaussian density function. Which of the following best describes the latent variables in the GMM framework?

A. The mixing weights π_k that determine the prior probability of each Gaussian component.

B. The covariance matrices Σ_k, which control the shape of each Gaussian distribution.

C. The component assignments z_n, which indicate which Gaussian component generated each data point.

D. The observed data points x_n, which follow a mixture of Gaussians.

(b) **Expectation Step (E-Step)**

In the E-step, we compute the posterior responsibility γ_{nk}, which represents the probability that data point x_n was generated by component k:

$$\gamma_{nk} = p(z_n = k|x_n, \theta^{(t)}) = \frac{\pi_k^{(t)} \mathcal{N}(x_n|\mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_{j=1}^{K} \pi_j^{(t)} \mathcal{N}(x_n|\mu_j^{(t)}, \Sigma_j^{(t)})}$$

What is the main role of γ_{nk} in the EM algorithm?

A. It represents the maximum likelihood estimate of the Gaussian parameters.

B. It updates the mixing weights to reflect the proportion of data points assigned to each cluster.

C. It acts as a ”soft” assignment of each data point to the Gaussian components.

D. It maximizes the log-likelihood function directly.

(c) **Maximization Step (M-Step)**

In the M-step, we update the parameters of the Gaussians by maximizing the expected complete-data log-likelihood. The updates are:

$$\pi_k^{(t+1)} = \frac{1}{N} \sum_{n=1}^{N} \gamma_{nk}$$

$$\mu_k^{(t+1)} = \frac{\sum_{n=1}^{N} \gamma_{nk} x_n}{\sum_{n=1}^{N} \gamma_{nk}}$$

$$\Sigma_k^{(t+1)} = \frac{\sum_{n=1}^{N} \gamma_{nk} (x_n - \mu_k^{(t+1)})(x_n - \mu_k^{(t+1)})^T}{\sum_{n=1}^{N} \gamma_{nk}}$$

What does the M-step accomplish?

A. It reassigns each data point to a single Gaussian component.

B. It updates the model parameters to maximize the likelihood given the current soft assignments.

C. It computes the posterior probability of each data point belonging to a Gaussian component.

D. It eliminates one Gaussian component per iteration to simplify the model.

(d) **Convergence and Likelihood Maximization**

The EM algorithm repeats the E-step and M-step iteratively until convergence, typically when the log-likelihood:

$$\log p(X|\theta) = \sum_{n=1}^{N} \log \sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)$$

stabilizes.

Which of the following statements is **true** regarding the convergence properties of EM?

A. EM always finds the global maximum of the likelihood function.

B. EM maximizes a lower bound on the likelihood at each iteration, ensuring non-decreasing likelihood.

C. EM can decrease the likelihood in some iterations.

D. EM requires computing second-order derivatives to estimate parameter updates.

New Potential Questions

1. **Conceptual Challenges and Solutions**

   For each of the following questions, provide a concise explanation (1-2 sentences) that clearly articulates the fundamental challenge or the corresponding solution.

   (a) What is the fundamental challenge of learning parameters in undirected graphical models as compared to learning parameters in a directed graphical model?

   (b) What is the fundamental solution that both Iterative Proportional Fitting (IPF) and Generalized Iterative Scaling (GIS) use?

   (c) What is the fundamental challenge of learning parameters when some of the variable values are not observed (i.e., the GM is only partially-observed)?

   (d) What is the fundamental solution that Expectation-Maximization (EM) uses?

   (e) Why can an independence-equivalence (I-equivalence) class contain multiple graphs, and what does this imply for structure learning?

   (f) Why is structural regularization often needed when learning general graphical models, whereas Chow-Liu trees can be learned via likelihood maximization without additional regularization?

2. **Monty Hall as a PGM**

   The Monty Hall problem is a classic probability puzzle based on a game show scenario. There are three doors, behind one of which is a car, and behind the other two are goats. A contestant picks a door. Then, the host, Monty Hall, who knows what is behind each door, always opens a door that the contestant did not pick, revealing a goat. The contestant is then given the choice to either stay with their initial choice or switch to the remaining unopened door.

   To analyze this using **probabilistic graphical models**, consider the following **random variables**:

   - $C \in \{1, 2, 3\}$: The door hiding the car (chosen uniformly at random).
   - $P \in \{1, 2, 3\}$: The door initially picked by the contestant (assumed uniform).
   - $M \in \{1, 2, 3\}$: The door Monty opens (determined based on C and P).

   (a) Draw a **Bayesian Network (BN)** representing the relationships between these variables. Clearly specify **which edges exist** in the BN and explain why.

   (b) Write the **prior probabilities** for C and P.

   (c) Construct the **conditional probability table** for $P(M|C, P)$, ensuring it accounts for Monty's behavior.

   (d) Given that Monty has opened **door 2** and the contestant initially picked **door 1**, compute:

- $P(C = 1 \mid M = 2, P = 1)$ (i.e., probability that the car is behind the originally chosen door).
- $P(C = 3 \mid M = 2, P = 1)$ (i.e., probability that the car is behind the other unopened door).

Show your calculations explicitly using **Bayes' Theorem** and the **conditional independence** properties of the Bayesian Network.

(e) Based on the inferred probabilities, should the contestant **switch** or **stay** to maximize their chance of winning the car? Justify your answer using probabilistic reasoning derived from your Bayesian network.

3. **Bayesian Network Theory**

State **TRUE** or **FALSE** for each of the following questions. In parts (f)-(g), P is a distribution,$\mathcal{G}$ is a BN structure, and $\mathcal{I}$ is an independence set.

(a) For all strictly positive joint distributions of A, B, C, if $A \perp B \mid C$ and $A \perp C \mid B$, then $A \perp B$ and $A \perp C$.

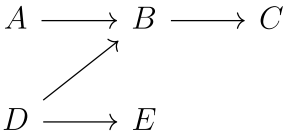


Figure 1: A Bayesian network.

(b) In Figure 1, $E \perp C \mid B$.

(c) In Figure 1, $A \perp E \mid C$.

(d) The BN structure in Figure 1 could be used to learn a distribution that matches

$$P(A, B, C, D, E) = P(A)P(B \mid A)P(C \mid B)$$

by appropriately setting its parameters.

(e) If a BN is converted to an undirected Markov Random Field (MRF) with the same node/edge skeleton, could the set of distributions that factorize according to the graph ever be smaller?

(f) If distribution P factorizes over graph $\mathcal{G}$, then $\mathcal{I}(\mathcal{G}) \subseteq \mathcal{I}(P)$.

(g) If $\mathcal{G}$ is an I-map for P, then P may have extra conditional independencies than $\mathcal{G}$.

4. **Approximate Inference Theory** Let's compare Variational Inference (VI) with Markov Chain Monte Carlo (MCMC), the two most popular methods for approximate inference. Below is a list of algorithmic properties or problem settings. For each item, link to either **VI** or **MCMC** (you only need to write **VI** or **MCMC**).

(a) Inference results are generally closer to target distributions.

(b) Non-parametric.

(c) Amenable to batched computation using GPUs.

(d) Transform inference into optimization problems.

(e) Easier to integrate with back-propagation.

(f) Involves more stochasticity.

(g) Easier to set the termination condition for the computational loop.

(h) Higher variance under limited computational resources.

(i) Problem case: Estimating a topic model with online streaming text data.

(j) Problem case: Estimating a topic model from a very small text corpus.

5. **Linear-chain CRF**

Part-of-speech (POS) tagging is a supervised learning problem with sequential inputs and sequential outputs. Instead of using a Hidden Markov Model (HMM), a generative model, we can directly model the conditional distribution $\mathbb{P}(\mathbf{y} \mid \mathbf{x})$ using the following form:

$$\mathbb{P}(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, \mathbf{x}), \tag{1}$$

where $\Psi_t(y_t, y_{t-1}, \mathbf{x})$ is a non-negative potential function that depends on a pair of tags y_t, y_{t-1} and a sequence of words $\mathbf{x}$, and $Z(\mathbf{x})$ is the normalizing constant.

(a) **Connecting HMMs to CRFs** Show that the conditional distribution $\mathbb{P}(\mathbf{y} \mid \mathbf{x})$ of an HMM can be written as a special case of a linear-chain CRF. Specifically, write down the complete likelihood for an HMM and express it in a log-linear form similar to Equation (1).

(b) **Feature Engineering in CRFs** A key advantage of CRFs over HMMs is the ability to incorporate rich, overlapping features. Define a **log-linear** model for the potential function $\Psi_t(y_t, y_{t-1}, \mathbf{x})$ using the following features:

1. identity of the given word, x_t, and the current tag, y_t,
2. identity of the previous word, x_{t-1}, and the current tag, y_t,
3. identity of the next word, x_{t+1}, and the current tag, y_t,
4. identity for whether x_t contains a capital letter, and the current tag, y_t,
5. identity of the previous tag, y_{t-1}, and the current tag, y_t.

Explicitly define the feature functions $f_k(y_t, y_{t-1}, \mathbf{x})$ and express the potential function $\Psi_t(y_t, y_{t-1}, \mathbf{x})$ in terms of these feature functions and their corresponding weights.

(c) **Maximum Likelihood Learning in CRFs** Derive the log-likelihood function for your linear-chain CRF and outline the steps for parameter estimation using gradient-based optimization. Briefly discuss the computational challenges involved in computing gradients and normalizing constants.

Assume $\mathbf{x}$ are binary vectors. In other words, $p_{\boldsymbol{\theta}}(\mathbf{x} \mid \mathbf{z})$ can be modeled with a sigmoid belief net, so the likelihood is of the form $p_\theta(\mathbf{x}|\mathbf{z}) = \text{Bernoulli}(f_\theta(\mathbf{z}))$.

Unused Questions from Quiz Study Guide

1. **MLE for Exponential Distribution**

   Suppose we observe data $x_1, x_2, \dots, x_n$ drawn from an exponential distribution with PDF:

   $$f(x; \lambda) = \lambda e^{-\lambda x}, \quad x \geq 0.$$

   (a) **Log-Likelihood Function**: Which of the following correctly represents the log-likelihood function $\ell(\lambda)$ for this dataset?

   A. $\ell(\lambda) = n \log \lambda - \lambda \sum_{i=1}^{n} x_i$

   B. $\ell(\lambda) = n \log \lambda + \lambda \sum_{i=1}^{n} x_i$

   C. $\ell(\lambda) = n \log \lambda - \lambda \prod_{i=1}^{n} x_i$

   D. $\ell(\lambda) = n\lambda - \lambda \sum_{i=1}^{n} x_i$

   (b) **Gradient of the Log-Likelihood**: What is the gradient of the log-likelihood $\ell(\lambda)$ with respect to λ?

   A. $\frac{\partial \ell}{\partial \lambda} = \frac{n}{\lambda} - \sum_{i=1}^{n} x_i$

   B. $\frac{\partial \ell}{\partial \lambda} = \frac{n}{\lambda} + \sum_{i=1}^{n} x_i$

   C. $\frac{\partial \ell}{\partial \lambda} = n\lambda - \sum_{i=1}^{n} x_i$

   D. $\frac{\partial \ell}{\partial \lambda} = \lambda \prod_{i=1}^{n} x_i$

   (c) **MLE for** λ: Which of the following is the Maximum Likelihood Estimator (MLE) for λ?

   A. $\hat{\lambda} = \frac{n}{\sum_{i=1}^{n} x_i}$

   B. $\hat{\lambda} = \frac{\sum_{i=1}^{n} x_i}{n}$

   C. $\hat{\lambda} = \frac{n}{\prod_{i=1}^{n} x_i}$

   D. $\hat{\lambda} = \frac{1}{\sum_{i=1}^{n} x_i}$

2. **Conditional vs Joint Models**

   Let's consider two different probabilistic models for a categorical outcome Y given feature variables $X = (X_1, X_2, \dots, X_n)$.

   Model 1 (Conditional Model) The probability of Y given X is modeled as:

   $$P(Y = k|X) = \frac{\exp(\theta_k^T X)}{\sum_j \exp(\theta_j^T X)}$$

   Model 2 (Joint Model) A joint probability distribution over (Y, X) is defined using an unnormalized score:

   $$P(Y, X) = \frac{\psi(Y, X)}{Z}, \quad \text{where } \psi(Y, X) = \exp(\theta_Y^T X)$$

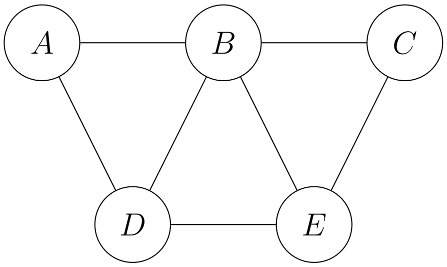
   and Z is the normalization constant.

(a) Derive an explicit expression for Z in Model 2.

(b) Express $P(Y|X)$ in Model 2.

(c) Show that Model 1 can be derived from Model 2 by choosing an appropriate form of Z.

3. **Identifying Cliques and Matching to Data Distributions** Consider the following undirected probabilistic graphical model, where nodes represent random variables, and edges represent direct dependencies between variables:



(a) Identify all **maximal cliques** in this graph. Write in the form (x, y, z) if nodes x, y, z are a maximal clique.

(b) Below is a conditional probability table (CPT) describing joint probabilities of some of the variables:

B	C	$P(B\|C)$	$P(C)$
0	0	0.7	0.5
0	1	0.4	0.5
1	0	0.3	0.5
1	1	0.6	0.5

B	C	E	$P(E\|B, C, D)$
0	0	0	0.8
0	0	1	0.2
0	1	0	0.5
0	1	1	0.5
1	0	0	0.4
1	0	1	0.6
1	1	0	0.1
1	1	1	0.9

Table 1: Conditional probability table for E.

Determine whether these conditional probability tables are consistent with the graphical model. Specifically:

(i) Does the factorization implied by the CPTs respect the independence assumptions of the graph?

(ii) If the tables do **not** match the graph, identify where the discrepancies occur and which independence assumptions are violated.

4. **Backpropagation with ℓ_2-Regularization**

Consider a neural network with one hidden layer. The network's output is:

$$\hat{y} = \sigma(w_2 \cdot h),$$

where:

- $h = \sigma(w_1 \cdot x)$,
- $\sigma(z)$ is the sigmoid activation function defined as $\sigma(z) = \frac{1}{1+e^{-z}}$,
- w_1 and w_2 are weight vectors,
- x is the input.

The network is trained using an ℓ_2-regularized loss function:

$$\mathcal{L}_{\text{reg}} = \mathcal{L} + \frac{\lambda}{2}\left(\|w_1\|_2^2 + \|w_2\|_2^2\right),$$

where $\mathcal{L}$ is the negative log-likelihood for binary classification:

$$\mathcal{L} = -y \log \hat{y} - (1-y)\log(1-\hat{y}),$$

and $\lambda > 0$ is the regularization strength.

(a) **Gradient of $\mathcal{L}_{\text{reg}}$ w.r.t.** w_2: Derive the total gradient of $\mathcal{L}_{\text{reg}}$ with respect to w_2. Select the correct expression:

A. $(\hat{y} - y) \cdot h + \lambda w_2$

B. $(\hat{y} - y) \cdot h - \lambda w_2$

C. $(\hat{y} - y) \cdot h$

D. $(\hat{y} - y) + \lambda w_2$

(b) **Gradient of $\mathcal{L}_{\text{reg}}$ w.r.t.** w_1: Derive the total gradient of $\mathcal{L}_{\text{reg}}$ with respect to w_1. Select the correct expression:

A. $(\hat{y} - y) \cdot w_2 \cdot \sigma'(w_1 \cdot x) \cdot x + \lambda w_1$

B. $(\hat{y} - y) \cdot \sigma'(w_1 \cdot x) \cdot x + \lambda w_1$

C. $(\hat{y} - y) \cdot w_2 \cdot \sigma'(w_1 \cdot x) \cdot x$

D. $(\hat{y} - y) \cdot \sigma'(x) \cdot w_2 + \lambda w_1$

(c) **Impact of Regularization**: Suppose the regularization term is removed ($\lambda = 0$). Which of the following best describes the impact on the optimization process?

A. The model will fit the training data more closely, potentially overfitting.

B. The model will have higher training error but better generalization.

C. The gradients with respect to w_1 and w_2 will increase in magnitude.

D. The optimization will converge more slowly.

(d) **Regularization Strength**: If λ is increased, what effect will this have on the learned weights w_1 and w_2?

A. The weights will shrink, reducing overfitting.

B. The weights will increase, fitting the training data better.

C. The weights will remain unchanged, as λ does not affect the optimization.

D. The weights will oscillate during training.

5. **Deriving the Structure and Properties of a Restricted Boltzmann Machine**

Consider a probabilistic model with two types of binary random variables:

- **Visible variables** $V = \{V_1, \ldots, V_n\}$, which represent observed data.
- **Hidden variables** $H = \{H_1, \ldots, H_m\}$, which encode dependencies between visible variables.

After analyzing the model, you determine the following structural properties:

1. Each visible variable V_i is connected to some hidden variables H_j.
2. Each hidden variable H_j is connected to some visible variables V_i.
3. **No direct connections exist between visible variables**.
4. **No direct connections exist between hidden variables**.
5. The joint probability distribution is **defined using an energy function**, rather than conditional probabilities.

(a) **Graphical Model Structure**: Given the properties above, which of the following best describes the structure of this model?

A. A directed graphical model (Bayesian network) where each hidden node is a parent of multiple visible nodes.

B. A fully connected undirected graphical model, where every variable (visible or hidden) is connected to every other variable.

C. A bipartite undirected graphical model, where edges exist **only** between visible and hidden nodes, and no edges exist within either group.

D. A Markov random field with local cliques, where visible variables are conditionally independent given their neighbors.

(b) **Marginal Probability of Visible Units**: The joint probability of visible and hidden units is given by the Boltzmann distribution:

$$P(V, H) = \frac{1}{Z} e^{-E(V,H)}$$

where the energy function takes the form:

$$E(V, H) = -\sum_i b_i V_i - \sum_j c_j H_j - \sum_{i,j} W_{ij} V_i H_j.$$

Which of the following correctly expresses the marginal probability $P(V)$ after summing over all hidden variables?

A. $P(V) = \sum_H P(V, H)$, summing out hidden variables explicitly.

B. $P(V) \propto e^{\sum_i b_i V_i}$, ignoring the hidden units.

C. $P(V) \propto \prod_j \left(1 + e^{c_j + \sum_i W_{ij} V_i}\right) e^{\sum_i b_i V_i}$.

D. $P(V) \propto e^{-\sum_{i,j} W_{ij} V_i H_j}$, treating the energy function as directly defining probabilities.

(c) **Role of Hidden Units**: Why do RBMs allow visible units to have statistical dependencies, even though there are no direct connections between them?

A. The hidden units introduce shared dependencies, making visible units **conditionally dependent** even though they are **conditionally independent given** H.

B. The visible units are always independent, since there are no direct edges between them.

C. The bipartite structure forces visible units to be independent **both marginally and conditionally**, meaning RBMs can only model very simple distributions.

D. The energy function forces every visible unit to depend only on itself, meaning hidden units have no real effect.