Assignment 2

Due November 03, 2025 at 11:59pm

This assignment is to be done individually.

Important Note: The university policy on academic dishonesty (cheating) will be taken very seriously in this course. You may not provide or use any solution, in whole or in part, to or by another student.

You are encouraged to discuss the concepts involved in the questions with other students. If you are in doubt as to what constitutes acceptable discussion, please ask! Further, please take advantage of office hours offered by the instructor and the TA if you are having difficulties with this assignment.

DO NOT:

- Give/receive code or proofs to/from other students
- Use Google to find solutions for assignment

DO:

- Meet with other students to discuss assignment (it is best not to take any notes during such meetings, and to re-work assignment on your own)
- Use online resources (e.g. Wikipedia) to understand the concepts needed to solve the assignment.

Submitting Your Assignment

The assignment must be submitted online on Canvas. You must submit a report in **PDF format**. You may typeset your assignment in LaTeX or Word, or submit neatly handwritten and scanned solutions. We will not be able to give credit to solutions that are not legible.

---

1 Linear Regression

a) ***Gaussian Noise Regression Model***

Consider a simple regression model with a noise variable ϵ as follows: $\hat{y} = \boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}) + \epsilon$, where $\boldsymbol{x} \in \mathcal{R}^M, y_i \in \mathcal{R}$, $\boldsymbol{\phi}$ is the basis function and $\boldsymbol{w}$ is the coefficients vector. Assume that ϵ follows a Gaussian distribution.

Now consider a data set of inputs $\mathbf{X} = \boldsymbol{x}_1, ..., \boldsymbol{x}_N$ with corresponding target values $y_1, ..., y_N$. Unlike Gaussian noise with the same value of variance at every training data point, consider that there is a different value of noise variance at each training data point.

The probability density function with different variances is given below:

$$p(y|\mathbf{X}, \boldsymbol{w}, \sigma) = \prod_{i=1}^{N} \mathcal{N}(y_i|\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}), \sigma_i^{-1}) \tag{1}$$

where σ_i is the inverse variance.

In this scenario, solve the following:

(i) Derive the log likelihood of $p(y|\mathbf{X}, \boldsymbol{w}, \sigma)$. Please show the detailed expression of $\mathcal{N}(y_i|\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}), \sigma_i^{-1})$, $\ln \mathcal{N}(y_i|\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x}), \sigma_i^{-1})$, and the $\ln p(y|\mathbf{X}, \boldsymbol{w}, \sigma)$ clearly in your solutions. (**9 marks**)

(ii) Comment on the relationship between the sum of squared error function and the log likelihood of $p(y|\mathbf{X}, \boldsymbol{w}, \sigma)$. (**3 marks**)

b) ***Weighted Linear Regression***

Given training data of the form: $\mathcal{D} = (\mathbf{X}, \boldsymbol{y}) = \{(\boldsymbol{x}_i, y_i)\}, i = 1, 2, \ldots, N$, where $\boldsymbol{x}_i \in \mathcal{R}^M$, i.e. $\boldsymbol{x}_i = (x_{i,1}, \cdots, x_{i,M})^\top, y_i \in \mathcal{R}, \mathbf{X} \in \mathcal{R}^{N \times M}$, where row i of $\boldsymbol{X}$ is $\mathbf{x}_i^\top$, and $\boldsymbol{y} = (y_1, \cdots, y_N)^\top$. Linear regression assumes a parametric model of the form: $\hat{y}_i = \boldsymbol{w}^\top \boldsymbol{x}_i + \epsilon_i$ where ϵ_i are noise terms from a given distribution, and seeks to find the parameter vector $\boldsymbol{w}$ that provides the best fit of the above regression model. One criterion to measure fitness is to find $\boldsymbol{w}$ that minimizes a given loss function $\mathcal{J}(\boldsymbol{w})$. We have shown that if we take the loss function to be the square error, i.e.:

$$\mathcal{J}(\boldsymbol{w}) = \sum_i \left(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i\right)^2 = (\mathbf{X}\boldsymbol{w} - \boldsymbol{y})^\top (\mathbf{X}\boldsymbol{w} - \boldsymbol{y})$$

Then,

$$\boldsymbol{w}^* = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \boldsymbol{y} \tag{2}$$

(i) Now we assume that $\epsilon_1, \ldots, \epsilon_N$ are independent and each $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$. Write down the formula for the MLE of $\boldsymbol{w}$: $\boldsymbol{w}_{MLE} = \arg\min_{\boldsymbol{w}}$(You need to fill here). Please use $\sigma_i, \exp, y_i, x_i, \boldsymbol{w}$ instead of $\mathcal{N}()$. (**3 marks**)

(ii) Calculate the MLE of $\boldsymbol{w}$ based on last question. Please show the following items:

i. The detailed expression of the MLE of $\boldsymbol{w}$ in matrix notation after ignoring derivative-irrelevant items. You may define a new matrix (vector) if needed. (**3 marks**)

ii. The process of computing derivatives. (**3 marks**)

iii. The final result in matrix notation. (**3 marks**)

2 Bayesian Regression, Information Measures, and the Bias–Variance Trade-off

As discussed in the course, a viewpoint in statistical learning is the Bayesian perspective, where we place a prior distribution over the parameters we wish to estimate. In this question, you will revisit the same linear regression problem from this Bayesian standpoint to gain deeper insight into the underlying principles of probabilistic modeling.

In the Bayesian framework, we treat model parameters $\overrightarrow{w}$ as random variables and express our beliefs about them probabilistically. Bayes' theorem provides the foundation for updating these beliefs after observing data $\mathcal{D}$:

$$p(\overrightarrow{w} \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \overrightarrow{w})\, p(\overrightarrow{w})}{p(\mathcal{D})}.$$

Here,

- $p(\overrightarrow{w})$ is the **prior**, representing our belief about the parameters before seeing any data.
- $p(\mathcal{D} \mid \overrightarrow{w})$ is the **likelihood**, describing how probable the observed data are under given parameters.
- $p(\mathcal{D}) = \int p(\mathcal{D} \mid \overrightarrow{w}) p(\overrightarrow{w})\, d\overrightarrow{w}$ is the **evidence** or **marginal likelihood**, acting as a normalizing constant.
- $p(\overrightarrow{w} \mid \mathcal{D})$ is the **posterior**, capturing our updated belief about the parameters after observing data.

This formulation elegantly combines prior knowledge and observed evidence to yield a coherent probabilistic description of parameter uncertainty.

Consider a Bayesian linear regression model for a scalar output y and feature vector $\overrightarrow{x} \in \mathbb{R}^d$. Assume a Gaussian likelihood with fixed noise variance σ^2:

$$p(y \mid \overrightarrow{x}, \overrightarrow{w}) = \mathcal{N}(y \mid \overrightarrow{w}^\top \overrightarrow{x},\, \sigma^2),$$

and a zero-mean isotropic Gaussian prior on the weight vector:

$$p(\overrightarrow{w}) = \mathcal{N}(\overrightarrow{w} \mid 0,\, \tau^{-1} I),$$

where $\tau > 0$ is the prior precision (inverse variance). Let $D = \{(\overrightarrow{x_i}, y_i)\}_{i=1}^N$ be the training data, X the design matrix (a matrix that stacks all your input feature vectors for the training set), and $\overrightarrow{w}_{\mathrm{MAP}}$ the maximum a posteriori estimate of $\overrightarrow{w}$. In the following, y refers to the output corresponding

to a single input $\overrightarrow{x}$, while $\overrightarrow{y}$ denotes the concatenation of outputs for multiple inputs (i.e., for the dataset X). We explore probabilistic regression, information-theoretic measures, and the bias–variance trade-off in this Bayesian setting.

a) **Posterior Derivation:** Derive the posterior distribution $p(\overrightarrow{w} \mid D)$ for the Bayesian linear regression model. Show that the posterior is Gaussian $p(\overrightarrow{w} \mid D) = \mathcal{N}\big(\overrightarrow{w} \,|\, \overrightarrow{m}_N,\, S_N\big)$, and give expressions for the posterior mean $\overrightarrow{m}_N$ and covariance S_N in terms of X, $\overrightarrow{y}$, σ^2, and τ **(6 marks)**.

Hint: Complete the square in the exponent combining the Gaussian prior and likelihood. You should find that:

$$S_N^{-1} = \tau I + \frac{1}{\sigma^2} X^\top X, \quad \overrightarrow{m_N} = S_N \frac{1}{\sigma^2} X^\top \overrightarrow{y}.$$

b) **Predictive Distribution:** Derive the predictive distribution $p(y_* \mid \overrightarrow{x_*}, D)$ for a new input $\overrightarrow{x_*}$. Show that by marginalizing out w, the predictive is Gaussian:

$$p(y_* \mid \overrightarrow{x_*}, D) = \mathcal{N}\Big(y_* \,|\, \mu_*,\, \sigma_*^2\Big),$$

with mean $\mu_* = \overrightarrow{m}_N^\top \overrightarrow{x_*}$ and variance $\sigma_*^2 = \sigma^2 + \overrightarrow{x_*}^\top S_N \overrightarrow{x_*}$. Explain how the Bayesian predictive is connected to Maximum a Posterior (MAP) predictor? Also interpret $\overrightarrow{x_*}^\top S_N \overrightarrow{x_*}$ term in variance of predictive distribution? What does it measure?

Hint: First show this general fact: if $p(w) = \mathcal{N}(\mu, \Lambda^{-1})$ and $p(y \mid w) = \mathcal{N}(Aw + b, L^{-1})$, then the marginal over w is Gaussian with

$$\mathbb{E}[y] = A\mu + b, \qquad \mathrm{Var}(y) = L^{-1} + A\Lambda^{-1}A^\top.$$

then use it to map means and covariances. **(6 marks)**

c) **Entropy of Predictions:** Using your result from previous sections, compute the differential entropy of the predictive distribution at a given $\overrightarrow{x}_*$. Simplify your answer to show that:

$$H[p(y_* \mid \overrightarrow{x_*}, D)] = \frac{1}{2} \ln(2\pi e\, \sigma_*^2),$$

where σ_*^2 is the predictive variance. Briefly interpret this result: how does the entropy (a measure of uncertainty) depend on the location $\overrightarrow{x_*}$? What happens to the entropy if $\overrightarrow{x_*}$ is in a region far from the training data versus a region well-supported by data? **(3 marks)**

d) **Variational View and the MAP–KL Connection:** Variational inference (VI) provides a general framework for approximating the posterior by minimizing a divergence between a tractable distribution $q(\overrightarrow{w})$ and the true posterior:

$$q^\star = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\big(q(\overrightarrow{w}) \,\|\, p(\overrightarrow{w} \mid D)\big), \quad \text{where} \quad p(\overrightarrow{w} \mid D) \propto p(\overrightarrow{w}) \prod_{i=1}^{N} p(y_i \mid \overrightarrow{x_i}, \overrightarrow{w}).$$

Let $\mathcal{Q} = \{\delta(\vec{w} - \vec{\mu}) : \vec{\mu} \in \mathbb{R}^d\}$ be the family of point-mass (Dirac) distributions—formally, the limit of $q_{\vec{\mu}}(\vec{w}) = \mathcal{N}(\vec{w}|\vec{\mu}, \varepsilon I)$ as $\varepsilon \to 0$. Show the following **(4 marks)**.

$$\arg\min_{\vec{\mu}} \ \mathrm{KL}\big(\delta(\vec{w} - \vec{\mu}) \,\|\, p(\vec{w} \mid D)\big) \ = \ \arg\max_{\vec{\mu}} \ \log p(\vec{\mu} \mid D) \ = \ \vec{w}_{\mathrm{MAP}}.$$

Hint: Use the identity $\mathrm{KL}(q\|p) = \mathbb{E}_q[\log q - \log p]$, and note that the $\mathbb{E}_q[\log q]$ term does not depend on $\vec{\mu}$ when the variance is fixed; in the Dirac limit, $\mathbb{E}_q[\log p(\vec{w} \mid D)] = \log p(\vec{\mu} \mid D)$.

e) **Bias–Variance Trade-off:** Let the true regression function be $f(\vec{x}) = \mathbb{E}[y \mid \vec{x}]$, and assume the observation noise has variance $\mathrm{Var}(y \mid \vec{x})$. Given a learning algorithm L a dataset $\mathcal{D}$, the learned parameters are $\vec{w^*}$ and the predictor is

$$f(\vec{x}; \vec{w^*}(\mathcal{D}, L)).$$

Then, the expected prediction error at point $\vec{x}$ is:

$$\epsilon(\vec{x}, f, L) = \mathbb{E}_{y,\mathcal{D}}\Big[\big(f(\vec{x}; \vec{w^*}(\mathcal{D}, L)) - y\big)^2 \mid \vec{x}\Big].$$

This can be decomposed as:

$$\epsilon(\vec{x}, f, L) = \underbrace{\Big(\mathbb{E}_{\mathcal{D}}\Big[f(\vec{x}; \vec{w^*}(\mathcal{D}, L)) \mid \vec{x}\Big] - \mathbb{E}_y[y \mid \vec{x}]\Big)^2}_{\text{Bias}^2} + \underbrace{\mathrm{Var}\Big(f(\vec{x}; \vec{w^*}(\mathcal{D}, L)) \mid \vec{x}\Big)}_{\text{Variance}} + \underbrace{\mathrm{Var}(y \mid \vec{x})}_{\text{Irreducible Error}}.$$

i. Provide a derivation of this bias–variance decomposition for mean-squared error **(3 marks)**.

ii. Discuss how the Bayesian regression model balances bias and variance. What role does the prior precision τ play in this trade-off? What happens in the limits of an extremely informative prior ($\tau \to \infty$) versus an uninformative prior ($\tau \to 0$)? Describe the qualitative effect on bias and variance in predictions **(2 marks)**.

3 Neural Network and Non-Linear Optimizations

a) ***Neural Networks***

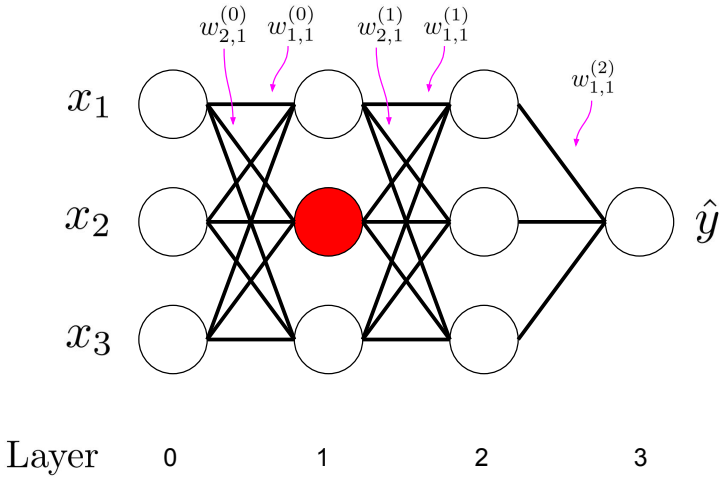
We will perform the forward and backward passes on the neural network below.

Notation: Please use the notation following the examples of names for weights given in the figure. For instance, the red node would have pre-activation of $z_2^{(1)} = w_{2,1}^{(0)} x_1 + w_{2,2}^{(0)} x_2 + w_{2,3}^{(0)} x_3$ and post-activation of $h_2^{(1)} = g(z_2^{(1)})$. In this question, the superscript will refer to the network layer the variable belongs to. In vector notation, we would have $\vec{z}^{(1)} = W^{(0)} \vec{x}$, where:

$$\vec{z}^{(1)} = \begin{pmatrix} z_1^{(1)} \\ z_2^{(1)} \\ z_3^{(1)} \end{pmatrix}, \quad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad W^{(0)} = \begin{bmatrix} w_{1,1}^{(0)} & w_{1,2}^{(0)} & w_{1,3}^{(0)} \\ w_{2,1}^{(0)} & w_{2,2}^{(0)} & w_{2,3}^{(0)} \\ w_{3,1}^{(0)} & w_{3,2}^{(0)} & w_{3,3}^{(0)} \end{bmatrix} \tag{3}$$

Activation functions: Assume the activation functions $g(\cdot)$ for the hidden layers are Sigmoid. For the final output node assume the activation function is an identity function $g(a) = a$.

Loss function: Assume this network is doing regression, trained using the standard squared error for a given data point $(\vec{x}, y)$ so that $L(\vec{W}) = \frac{1}{2}(\hat{y} - y)^2$.



(a) Compute the outputs of the middle layers and the final output of the network $\hat{y}$ for the given input vector $\vec{x} = (1, 1, 1)^\top$ and the following weights. There are no biases. (**3 marks**)

$$W^{(0)} = \begin{bmatrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{bmatrix}, \quad W^{(1)} = \begin{bmatrix} 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \\ 0.3 & 0.3 & 0.3 \end{bmatrix}, \quad W^{(2)} = [0.5, 0.5, 0.5]$$

(b) Compute

$$\frac{\partial L}{\partial \hat{y}}, \frac{\partial L}{\partial \vec{z}^{(2)}}, \frac{\partial L}{\partial \vec{z}^{(1)}}$$

(**6 marks**)

(c) Compute

$$\frac{\partial L}{\partial W^{(0)}}, \frac{\partial L}{\partial W^{(1)}}, \frac{\partial L}{\partial W^{(2)}}$$

Use this result to update all the weights in the network using gradient descent after one step if the ground truth label is $y = 2$, given the above-mentioned initial weights, and learning rate of 1 (**6 marks**).

b) ***Non-linear Optimization***

In this question, you need to implement iterative algorithms to solve the following non-linear optimization problem. Finish the codes in this Jupyter Notebook (there are total 7

"#<<TODO#x>>" in this question). Please make a copy of the notebook in Google Drive and run your codes in there.

Consider the following objective function.

$$f(x, y) = x^2 + \frac{(y-2)^2}{2} \tag{4}$$

For the following questions, you can describe the results from the perspective of whether function reaches the minimum value, and the convergence speed, etc.

(a) Implement a basic Gradient Descent algorithm for the objective function (4). (**1 points**)

- Use step size of 0.2, total iterations of 100, and initial point of $(-10, 10)$ for the objective function.
- Change the step size to 0.01. Provide plots for both step size and report your observation.

(b) Implement the Adaptive Gradient (AdaGrad) algorithm for the objective function (4). (**3 points**)

- Use step size of 0.2, total iterations of 100, and initial point of $(-10, 10)$ for the objective function.
- Change the step size to 0.3. Provide plots for both step size and report your observation.

(c) Implement the Adam algorithm for the objective function (4). (**3 points**)

- Use step size of 0.2, total iterations of 150, and initial point of $(-10, 10)$ for the objective function. Use default values for other parameters. Does the solution converge to the minimal objective value? Provide the plot and also report your observation.

(d) Implement the Newton's Method algorithm for the objective function (4). (**3 points**)

- Use step size of 0.01, total iterations of 100, and initial point of $(-10, 10)$ for the objective function.
- Change the step size to 0.1. Provide plots for both step size and report your observation.