Question 1 – Bayes' rule with one sensor **[10 points]**

A robot is driving down a dark corridor that may contain an obstacle ($O = 1$ if an obstacle is present, $O = 0$ otherwise). Before using any sensors, the robot believes $P(O = 1) = 0.10$. It has two sensors:

	$P(\text{detect} \mid O = 1)$	$P(\text{detect} \mid O = 0)$
Visible camera (C)	0.90	0.20
Infrared sensor (I)	0.80	0.05

("detect" means the sensor reports an obstacle; $C = 1$ and $I = 1$ denote detections.)

(a) The camera detects an obstacle and the infrared sensor is not used. What is $P(O = 1 \mid C = 1)$? *(5 pts)*

(b) The infrared sensor detects an obstacle and the camera is not used. What is $P(O = 1 \mid I = 1)$? *(5 pts)*

Question 2 – Bayes' rule with two independent sensors **[10 points]**

Use the same prior and sensor error rates as Question 1. Now assume the camera and the infrared sensor make *conditionally independent* errors given the true state O (i.e., $P(C, I \mid O) = P(C \mid O)\, P(I \mid O)$).

(a) Both sensors detect an obstacle. What is $P(O = 1 \mid C = 1, I = 1)$? *(6 pts)*

(b) The camera detects an obstacle but the infrared sensor does *not*. What is $P(O = 1 \mid C = 1, I = 0)$? *(4 pts)*

Question 3 – Bayes' rule with dependent sensors **[10 points]**

In reality the two sensors from Question 1 are *not* conditionally independent: bright sunlight from a nearby window can cause *both* to produce false alarms at the same time. The individual detection rates from Question 1 still hold, but the joint behavior of the two sensors is measured to be

$$P(C = 1, I = 1 \mid O = 1) = 0.75, \qquad P(C = 1, I = 1 \mid O = 0) = 0.04.$$

The prior is still $P(O = 1) = 0.10$.

(a) Both sensors detect an obstacle. What is $P(O = 1 \mid C = 1, I = 1)$? *(5 pts)*

(b) Your answer to (a) is different from your answer to Question 2(a). Which is larger? Using the numbers above, explain in one or two sentences what the independence assumption got wrong. *(5 pts)*

Question 4 – Forward pass through a tiny neural network **[10 points]**

Consider a network with two inputs (x_1, x_2), two hidden ReLU neurons, and one linear output neuron, where $\mathrm{ReLU}(z) = \max(0, z)$:

$$h_1 = \mathrm{ReLU}\big(\,1 \cdot x_1 + (-1) \cdot x_2 + 0.5\,\big)$$
$$h_2 = \mathrm{ReLU}\big(\,(-1) \cdot x_1 + 1 \cdot x_2 + 0\,\big)$$
$$y = 2\,h_1 + 3\,h_2 - 1$$

The input is $(x_1, x_2) = (2,\ 1)$, the target is $t = 1$, and the loss is $L = \frac{1}{2}(y - t)^2$.

(a) Compute the pre-activations and the activations h_1 and h_2. *(5 pts)*

(b) Compute the network output y. *(3 pts)*

(c) Compute the loss L. *(2 pts)*

Question 5 – Backpropagation by hand **[12 points]**

Consider the scalar network (no biases)

$$z = w_1 x, \qquad h = \mathrm{ReLU}(z), \qquad y = w_2 h, \qquad L = \tfrac{1}{2}(y - t)^2,$$

with $x = 2$, $t = 2$, $w_1 = 0.5$, and $w_2 = 3$.

(a) Do the forward pass: compute z, h, y, and L. *(3 pts)*

(b) Use the chain rule to compute $\dfrac{\partial L}{\partial y}$, $\dfrac{\partial L}{\partial w_2}$, $\dfrac{\partial L}{\partial h}$, $\dfrac{\partial L}{\partial z}$, and $\dfrac{\partial L}{\partial w_1}$. *(5 pts)*

(c) Perform one gradient descent step with learning rate $\eta = 0.1$. Give the new values of w_1 and w_2. *(2 pts)*

(d) Now suppose the initial weight were $w_1 = -0.5$ instead (everything else unchanged). Compute $\frac{\partial L}{\partial w_1}$ and $\frac{\partial L}{\partial w_2}$ and say what this means for learning. *(2 pts)*

Question 6 – Linear models and regression **[10 points]**

Fit the linear regression model $\hat{y} = wx + b$ to the data

$$(x, y): \quad (0, 1),\ (1, 2),\ (2, 2),\ (3, 3),$$

by minimizing the mean squared error. For one-dimensional data the least-squares solution is

$$w = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}, \qquad b = \bar{y} - w\,\bar{x}.$$

(a) Compute w and b. *(4 pts)*

(b) What does the fitted model predict for $x = 4$? *(1 pts)*

(c) Compute the mean squared error of the fitted model on the four training points. *(3 pts)*

(d) True or False (justify in one sentence): the model $\hat{y} = w_1 x + w_2 x^2 + b$ is a *linear* model. *(2 pts)*

Question 7 – Gradient descent and training **[10 points]**

Let $f(w) = (w - 3)^2$ and start gradient descent at $w_0 = 0$.

(a) With learning rate $\eta = 0.25$, compute w_1 and w_2. *(4 pts)*

(b) With learning rate $\eta = 1.5$, compute w_1. Will gradient descent converge to the minimum? Explain briefly. *(3 pts)*

(c) While training a neural network, you observe that the training loss keeps decreasing every epoch, but after epoch 20 the validation loss begins to increase. What is this called, and name two things you could do about it. *(3 pts)*

Question 8 – Automatic differentiation **[8 points]**

An autograd system is asked to differentiate $f(x, y) = (x + y) \cdot y$ at $x = 2$, $y = 3$. It builds the computational graph $a = x + y$, $f = a \cdot y$.

(a) Run the forward pass: give the values of a and f. *(2 pts)*

(b) Run the backward (reverse-mode) pass: starting from $\partial f/\partial f = 1$, compute $\partial f/\partial a$, $\partial f/\partial x$, and $\partial f/\partial y$. Note that y is used twice in the graph. *(4 pts)*

(c) In one sentence, explain why reverse-mode autodiff is preferred over forward-mode for training neural networks. *(2 pts)*

Question 9 – Tokenization and embeddings **[10 points]**

(a) **Byte-pair encoding.** The initial vocabulary is $\{a, b, c\}$ and the training text is the character sequence `ababcab`. Perform *one* BPE merge step. Which pair is merged, what is the new token sequence, and what is the new vocabulary? *(3 pts)*

(b) **Cosine similarity.** The embeddings of three words are king $= (1, 2)$, queen $= (2, 1)$, and apple $= (2, -1)$. Using cosine similarity $\cos(u, v) = \dfrac{u \cdot v}{\|u\| \, \|v\|}$, compute $\cos(\text{king}, \text{queen})$ and $\cos(\text{king}, \text{apple})$. Which word is closer to "king"? *(3 pts)*

(c) Answer each in one sentence:

(i) Why do modern language models use subword tokens (like BPE) rather than whole words? *(2 pts)*

(ii) Why are learned dense embeddings preferred over one-hot vectors for words? *(2 pts)*

Question 10 – Scaled dot-product attention **[10 points]**

A single attention head uses key dimension $d_k = 4$. The scaled dot-product attention score between a query q and a key k is $s = \frac{q \cdot k}{\sqrt{d_k}}$, the attention weights are the softmax of the scores, and the output is the weighted sum of the values. Given

$$q = (2, 0, 0, 0), \quad k_1 = (2, 0, 0, 0),\ v_1 = (10, 0), \quad k_2 = (0, 2, 0, 0),\ v_2 = (0, 10),$$

(a) Compute the scaled scores s_1 and s_2. *(2 pts)*

(b) Compute the attention weights α_1, α_2 using softmax. (Use $e^2 \approx 7.39$.) *(3 pts)*

(c) Compute the attention output. *(3 pts)*

(d) Why do we divide the dot products by $\sqrt{d_k}$? *(2 pts)*