Exercise 1: linear classifier (10 points). Consider a binary linear classifier $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$ with $\mathbf{w} = \binom{-3}{4}$ and $w_0 = -12$, where $\mathbf{x} \in \mathbb{R}^2$. Let class 1 be its positive side ($g(\mathbf{x}) > 0$) and class 2 its negative side ($g(\mathbf{x}) < 0$).

1. (4 points) Sketch the decision boundary in $\mathbb{R}^2$. Compute the points at which it intersects the coordinate axes. Indicate which is the positive side of the boundary (class 1).
2. (4 points) Compute the signed distance of the following points to the decision boundary: the origin; $\binom{-1}{3}$; $\binom{4}{6}$. Classify those points.
3. (2 points) Give a vector $\mathbf{u} \in \mathbb{R}^2$ that is parallel to the decision boundary and has norm 1.

Exercise 2: linear classifier (20 points). We have a classification problem with $K = 3$ classes in $\mathbb{R}^2$ with the following discriminant functions:

$$\begin{aligned} g_1(\mathbf{x}) &= 2x_1 + 3x_2 + 3 \\ g_2(\mathbf{x}) &= x_1 + 4x_2 + 3 \\ g_3(\mathbf{x}) &= 2x_1 + 6x_2 + 2. \end{aligned}$$

1. (2 points) Give a rule to decide which class a point $\mathbf{x} \in \mathbb{R}^2$ should be assigned to.
2. (4 points) Classify the following points: $\binom{-1}{0}$, $\binom{1}{1}$, $\binom{-1}{-1}$, $\binom{1}{0}$.
3. (6 points) Give the equation that a point $\mathbf{x} \in \mathbb{R}^2$ must satisfy for it to be on the boundary between classes 1 and 2. Repeat for the boundary of class 1 and 3, and for class 2 and 3.
4. (2 points) Give the equation that a point $\mathbf{x} \in \mathbb{R}^2$ must satisfy for it to be on the boundary between all 3 classes.
5. (6 points) Based on the above, sketch the boundaries that delimit the 3 classes, indicating numerically where they cross the coordinate axes and which region corresponds to which class.

Exercise 3: logistic regression (14 points). Consider a binary classification problem in dimension D with a training set $\{(\mathbf{x}_n, y_n)\}_{n=1}^N$, where $\mathbf{x}_n \in \mathbb{R}^D$ and $y_n \in \{0, 1\}$ for $n = 1, \dots, N$.

1. (4 points) Write the cross-entropy objective function $E(\mathbf{w}, w_0)$ for logistic regression.
2. (8 points) Compute and simplify the gradient of E with respect to the parameters $\mathbf{w} \in \mathbb{R}^D$ and $w_0 \in \mathbb{R}$. Show your work.
3. (2 point) Write the update formulas for the parameters using gradient descent with a step size $\eta > 0$.

Exercise 4: multilayer perceptrons (8 points). Construct manually a perceptron that calculates the NAND of its two inputs. That is, given a training set

$$\{(\mathbf{x}_n, y_n)\}_{n=1}^N = \left\{ \left(\binom{0}{0}, 1\right),\ \left(\binom{0}{1}, 1\right),\ \left(\binom{1}{0}, 1\right),\ \left(\binom{1}{1}, 0\right) \right\}$$

of 2D points in two classes $\{0, 1\}$, give numerical values of the perceptron's parameters that solve this classification problem.

Exercise 5: properties of the logistic and $\tanh$ **functions (10 points).** Consider the logistic function $\sigma(x) = \frac{1}{1+e^{-x}} \in (0,1)$ for $x \in \mathbb{R}$. Prove the following properties:

1. (2 points) Inverse of logistic: $\sigma^{-1}(y) = \mathrm{logit}(y) = \log\left(\frac{y}{1-y}\right) \in (-\infty, \infty)$ for $y \in (0,1)$.
2. (2 points) Derivative of logistic: $\frac{d\sigma(x)}{dx} = \sigma'(x) = \sigma(x)(1-\sigma(x))$.
3. (1 points) $\sigma(x) + \sigma(-x) = 1$.

Consider now the hyperbolic tangent $\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \in (-1,1)$ for $x \in \mathbb{R}$. Work out the expression for:

1. (2 points) The inverse of tanh.
2. (2 points) The derivative of tanh, using the value of tanh itself.
3. (1 points) $\tanh(x) + \tanh(-x)$.

Exercise 6: multilayer perceptrons (9 points). Consider an MLP with a single hidden layer in which the hidden unit activation functions are the logistic function $\sigma(x) = \frac{1}{1+e^{-x}}$. Show that there exists an equivalent MLP which computes exactly the same function as the original MLP, but where the hidden unit activation functions are $\tanh x$. Hint: find a relation between $\sigma(x)$ and $\tanh x$.

Exercise 7: RBF networks (20 points). Consider a Gaussian radial basis function (RBF) network $\mathbf{f}\colon \mathbb{R}^D \rightarrow \mathbb{R}^{D'}$ that maps input vectors $\mathbf{x} \in \mathbb{R}^D$ to output vectors $\mathbf{y} \in \mathbb{R}^{D'}$:

$$\mathbf{f}(\mathbf{x}) = \sum_{h=1}^{H} \mathbf{w}_h e^{-\frac{1}{2}\left\|\frac{\mathbf{x}-\boldsymbol{\mu}_h}{\sigma}\right\|^2} \qquad \text{or, elementwise:} \quad f_e(\mathbf{x}) = \sum_{h=1}^{H} w_{he} e^{-\frac{1}{2\sigma^2}\sum_{d=1}^{D}(x_d-\mu_{hd})^2} \quad e = 1,\dots,D'$$

where the RBF network parameters are the weight vectors $\{\mathbf{w}_h\}_{h=1}^H \subset \mathbb{R}^{D'}$, the centroids $\{\boldsymbol{\mu}_h\}_{h=1}^H \subset \mathbb{R}^D$ and the bandwidth $\sigma > 0$. We want to train $\mathbf{f}$ in a regression setting by minimizing the least-squares error with a fixed regularization parameter $\lambda \geq 0$, given a training set $\{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$:

$$E\left(\{\mathbf{w}_h, \boldsymbol{\mu}_h\}_{h=1}^H, \sigma\right) = \sum_{n=1}^{N} \|\mathbf{y}_n - \mathbf{f}(\mathbf{x}_n)\|^2 + \lambda \sum_{h=1}^{H} \|\mathbf{w}_h\|^2 = \sum_{n=1}^{N}\sum_{e=1}^{D'} (y_{ne} - f_e(\mathbf{x}_n))^2 + \lambda \sum_{h,e=1}^{H,D'} w_{he}^2. \tag{1}$$

A simple but approximate way to train the RBF network is by fixing the value of its bandwidth $\sigma > 0$ (this value is eventually cross-validated) and its centroids $\{\boldsymbol{\mu}_h\}_{h=1}^H$ (e.g. to a random subset of training points, or to the result of running k-means on the training set), and then optimizing eq. (1) over the weights (which results in a linear system).

Instead, we wish to train the RBF network parameters by gradient descent, as with multilayer perceptrons.

1. (15 points) Using the chain rule, compute the gradients of E in eq. (1) wrt the parameters:
   (a) The weights $\{\mathbf{w}_h\}_{h=1}^H$: $\frac{\partial E}{\partial w_{he}} = \dots$ for $h = 1,\dots,H$ and $e = 1,\dots,D'$.
   (b) The centroids $\{\boldsymbol{\mu}_h\}_{h=1}^H$: $\frac{\partial E}{\partial \mu_{hd}} = \dots$ for $h = 1,\dots,H$ and $d = 1,\dots,D$.
   (c) The bandwidth σ: $\frac{\partial E}{\partial \sigma} = \dots$
2. (5 points) What would be a good initialization for these parameters (to start gradient descent)?

Exercise 8: ensemble learning (9 points). Consider the setting of regression from input vectors $\mathbf{x} \in \mathbb{R}^D$ to a single real output $y \in \mathbb{R}$. Imagine we have trained L learners $f_1,\dots,f_L\colon \mathbb{R}^D \rightarrow \mathbb{R}$ in some way (e.g. each on a bootstrapped sample from a training set). We combine them using their average: $f(\mathbf{x}) = \frac{1}{L}\sum_{l=1}^{L} f_l(\mathbf{x})$. What kind of model is the resulting f in each of the following cases? Be as specific as possible. *Hint*: we give the answer to the first case below.

1. (0 points) If $f_1,\dots,f_L$ are polynomials of degree q.
   Answer: f is another polynomial of degree q, whose coefficients are equal to the average of the corresponding coefficients in $f_1,\dots,f_L$.
2. (3 points) If $f_1,\dots,f_L$ are Gaussian RBF networks each with H centroids.
3. (3 points) If $f_1,\dots,f_L$ are linear regressors.
4. (3 points) If $f_1,\dots,f_L$ are MLPs each with a single hidden layer of H sigmoidal units and an output linear unit.