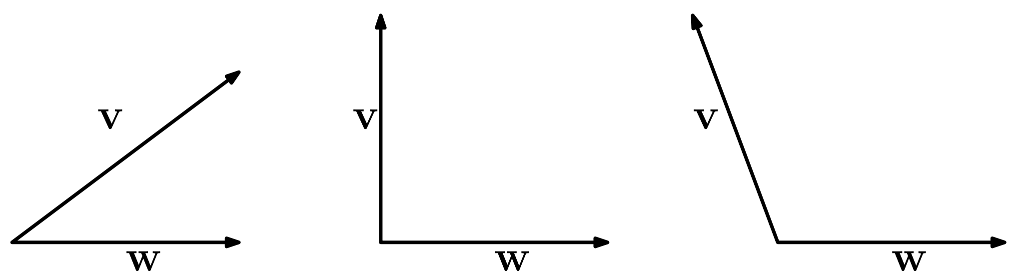
Seminar Week 2: Perceptron

Question 1

Consider the three scenarios drawn below. For each one say whether $\mathbf{v} \cdot \mathbf{w}$ is greater than, less than or equal to zero.



Question 2 (pen and paper)

The gradient descent algorithm requires calculating the gradient, either numerically or analytically. In the case of the Perceptron and MLP, the gradient can be calculated analytically. Let's make sure it is all clear to you.

Consider function $y(x)$ defined as $y(x) = w \times x$ where w is a constant and x is a real number. What is $\frac{dy}{dx}$? Let's now assume that w is not a constant but a variable, just like x, i.e., $y(x, w) = w \times x$. Now, we can ask how y changes as a result of a change in either w or x. We refer to these as partial derivatives. What is $\frac{\partial y}{\partial x}$? What is $\frac{\partial y}{\partial w}$? The gradient of y is defined as the **vector** whose components are $\frac{\partial y}{\partial x}$ and $\frac{\partial y}{\partial w}$.

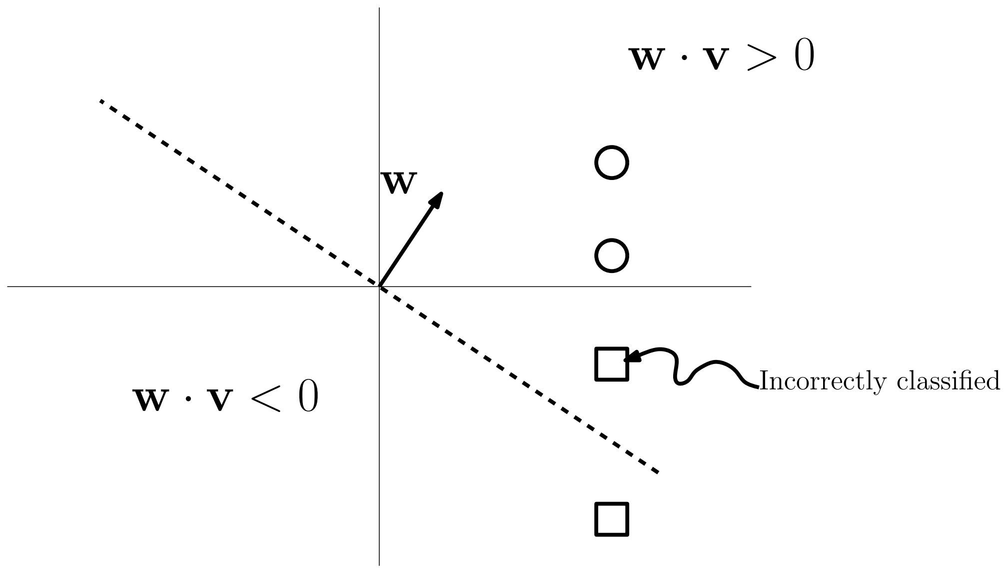
We now consider y defined as $y(x_1, x_2, x_3, w_1, w_2, w_3) = w_1 \times x_1 + w_2 \times x_2 + w_3 \times x_3$. What are the partial derivatives of y with respect to x_1, x_2 and x_3? What are the partial derivatives of y with respect to w_1, w_2 and w_3? What is the gradient of y with respect to $\mathbf{w}$ where $\mathbf{w}$ is the vector whose components are w_1,w_2, and w_3?

You should realise that if $\mathbf{w}$ and $\mathbf{x}$ are high-dimensional vectors (e.g., hundreds of components) it will be very convenient to rewrite y as a function of those vectors (rather than their components). What is that function? What is its gradient with respect to $\mathbf{w}$? What is its gradient with respect to $\mathbf{w}$? Please note the consistency with your answer to the first sub-question. You are still differentiating a product but now it is a product of two vectors!

The Perceptron error criterion introduced in lecture is given by: $E(\mathbf{w}) = -(\mathbf{w} \cdot \mathbf{x}) \times t$ where t is the target (± 1). Write the gradient of this error.

Question 3 (pen and paper)

For the example given in the diagram below (see next page), work through a couple of iterations of the gradient descent algorithm using the Perceptron error criterion: $E(\mathbf{w}) = -(\mathbf{w} \cdot \mathbf{v})t$ where t is the target (± 1) and v is the input (it was denoted x previously – you should not get too attached to any given notation). To complete this question, you will need to use the gradient you calculated in the previous question. Once this is done, draw the weight vector at each stage. For this example we threshold at zero with the decision boundary being given by the dashed line. Use a learning rate, η, equal to 1. Initially only one square is incorrectly classified.



Question 4 (programming exercise)

Download the Matlab data file `sem2a_q4_linsep_data.mat`. The data are given in the form (x-coordinate, y-coordinate, class taking values +1, -1). Please confirm visually that the data are linearly separable. Then, using the algorithm above, implement and train a Perceptron to correctly classify the data. At each update of the weight vector plot the decision boundary. Knowing that the data are linearly separable, what sort of stopping criterion can you use? How do you initialise the weights?

Next, use the data contained in Matlab data file `sem2a_q4_nonlinsep_data.mat`. After checking (visually) that the data are not linearly separable, consider what needs to be adapted in your implementation for a satisfactory outcome. What is a satisfactory outcome?

NB: If you'd rather not use Matlab, that's fine. Some languages will allow you to import Matlab files (e.g., Python). If your language does not make it possible, please ask the tutor.