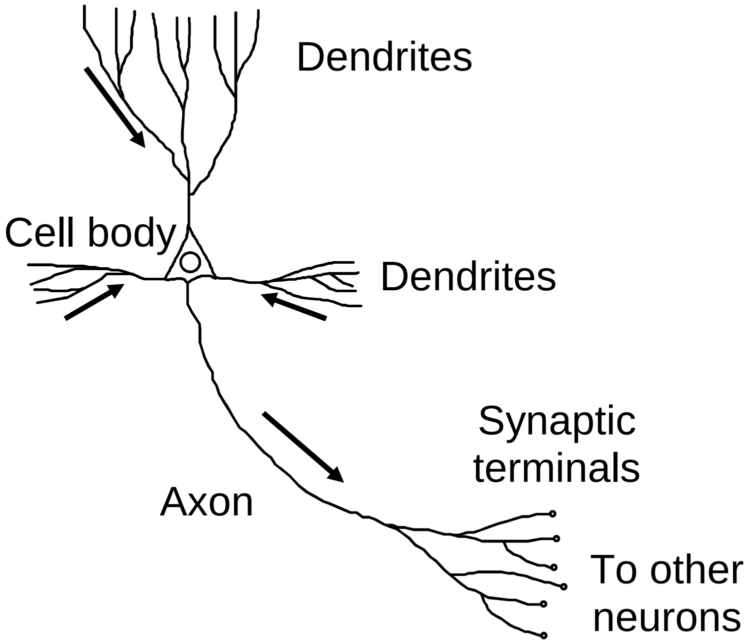
P01760 – Advanced Concepts in Signal Processing

Question Sheet 1

1. Sketch a diagram of a typical biological neuron. Explain briefly how it processes the information that it receives from its inputs. Compare this biological neuron to a simple artificial neuron, such as the McCulloch and Pitts neuron.

Ans:



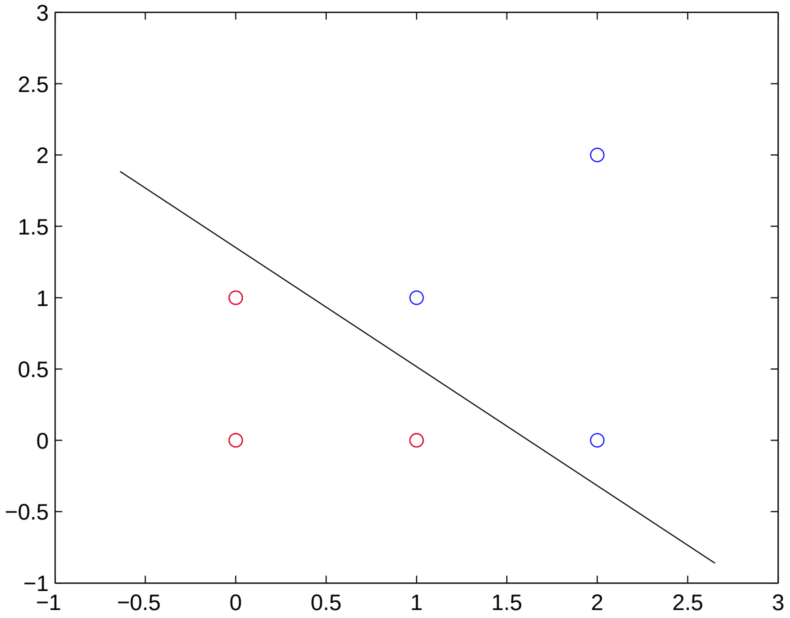
Biological neuron has inputs signals from dendrites, the synapses weight the output from connecting axons (of other neurons). At a given threshold the cell fires sending a fixed amplitude signal down the cell's axon. This is modelled by a set of linearly weighted inputs (c.f. synapses) passed into a thresholding function (c.f. firing) giving an output of either 0 or 1.

2. What is meant by *linearly separable*?

Ans: class labelled data is linearly separable if there exist linear disciminant functions, $g_k(\mathbf{x}) = w_0^{(k)} + \left[\mathbf{w}^{(k)}\right]^T \mathbf{x}$*, such that all data satisfies:*
$g_k(\mathbf{x}) > g_j(\mathbf{x})$*, j ≠ k if x belongs to class k.*

3. Consider the following data points from two categories:
ω_1: (1,1) (2,2) (2,0)
ω_2: (0,0) (1,0) (0,1)
Are they linearly separable? Explain your answer, giving an example or counterexample as appropriate.

Ans: Let's plot the data first:



Hence linearly separable (line shown separates the data).

4. Write down the function of a Linear Classifier. Explain how the data points can be augmented and normalized (i.e. standardized) to simplify the formulation of the linear classifier. Illustrate this by augmenting and normalizing the data in Q3.

Ans: A linear (2-class) classifier is defined by the function:

$$g(\mathbf{x}) = w_0 + \mathbf{w}^T \mathbf{x}$$

And we choose class 1 if $g(\mathbf{x}) > 0$ *and class 2 otherwise. Furthermore we can absorb* w_0 *into the weight vector by augmenting the input data with ones.*

e.g. for data above the augmented inputs are:

ω_1: (1,1,1) (1,2,2) (1,2,0)
ω_2: (1,0,0) (1,1,0) (1,0,1)
Then we only require:

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} > 0$$

To classify data as class 1.

Finally we can map all the data into the same class by multiplying the data vectors for class 2 by -1. E.g. for above data:

(1,1,1)	(1,2,2)	(1,2,0)
(-1,0,0)	(-1,-1,0)	(-1,0,-1)

5. Write down the fixed-increment single-sample perceptron algorithm. Suppose a 2-input perceptron had weights w_1=2.0, w_2=0.0, and bias w_0= –1.0. Calculate any

updates that occur as the data samples in Q3 are presented to a fixed-increment single-sample algorithm with an update factor of 1.0. The data is to be presented ω_1 first (left to right), then ω_2 (left to right).

Ans: try running the following matlab code:

```
% Perceptron algorithm applied to data in Q3
x = [1 1 1;1 2 2;1 2 0;-1 0 0;-1 -1 0;-1 0 -1]';

w = [-1 2 0]; % initialize weights;
test = 0;
while test ==0
    test=1;
    for i=1:6
        disp(w*x(:,i));
        if w*x(:,i)<=0
            w = w+x(:,i)'
            test = 0;
        end
    end
end
```

6. Use the Pseudoinverse method to find a discriminant function for the data samples in Q3. Use equal margins (b_i=1) for all data points.
Is this a separating hyperplane? Explain your answer.

Ans: try running the following matlab code:

```
b = ones(6,1);

w_mse = inv(x*x')*x*b

% Now test...

    for i=1:6
        disp(w*x(:,i));
        if w*x(:,i)<=0
           disp('Not separated!!!');
        end
    end
```

Yes it is separating (see code).

7. Consider the hyperplane used in discrimination. Show that the distance from the hyperplane $g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0 = 0$ to any point $\mathbf{x}_a$ is $|g(\mathbf{x}_a)| / \|\mathbf{w}\|$ by minimizing $\|\mathbf{x} - \mathbf{x}_a\|^2$ subject to the constraint $g(\mathbf{x}) = 0$.
Ans: given the constraint $g(\mathbf{x}) = 0$ *we can solve using a Lagrange multiplier:*

$$J(x, \lambda) = (\mathbf{x} - \mathbf{x}_a)^2 + \lambda(\mathbf{w}^T\mathbf{x} + w_0)$$

Differentiating and equating to zero, we get:

$$\frac{\partial J}{\partial \lambda}(x,\lambda) = 2(\mathbf{x}-\mathbf{x}_a)+\lambda\mathbf{w}=0$$

$$\frac{\partial J}{\partial \lambda}(x,\lambda) = (\mathbf{w}^T\mathbf{x}+w_0)=0$$

Which gives:

$$\mathbf{x}=\mathbf{x}_a-\frac{1}{2}\lambda\mathbf{w}$$

and

$$\mathbf{w}^T\left(\mathbf{x}_a-\frac{1}{2}\lambda\mathbf{w}\right)+w_0=0$$

$$\Rightarrow g(\mathbf{x}_a)-\frac{1}{2}\lambda\mathbf{w}^T\mathbf{w}=0$$

$$\Rightarrow \lambda = 2\frac{g(\mathbf{x}_a)}{\mathbf{w}^T\mathbf{w}}$$

Therefore the distance from $\mathbf{x}_a$ *to* $\mathbf{x}$ *is*

$$\|\mathbf{x}-\mathbf{x}_a\|^2=\left\|\left(\mathbf{x}_a-\frac{g(\mathbf{x}_a)}{\mathbf{w}^T\mathbf{w}}\mathbf{w}-\mathbf{x}_a\right)\right\|^2=\left\|\frac{g(\mathbf{x}_a)}{\mathbf{w}^T\mathbf{w}}\mathbf{w}\right\|^2=\left(\frac{\|g(\mathbf{x}_a)\|}{\|\mathbf{w}\|}\right)^2$$

Alternatively can be show using geometric arguments as in notes.

8. The *convex hull* of a set of vectors $\mathbf{x}_i,\ i=1,\ldots,n$ is the set of all vectors of the form $\mathbf{x}=\sum_{i=1}^{n}\alpha_i\mathbf{x}_i$ where the coefficients α_i are nonnegative and sum to one.
Given two sets of vectors, show that *either* they are linearly separable, *or* their convex hulls intersect.
[Hint: to answer this, suppose that both statements are true, and consider the classification of a point in the intersection of the convex hulls.]

Ans: we will consider only a 2-class system. Suppose $\mathbf{x}^*$ *is a point within the intersection of the two convex hulls and suppose the data is linearly separable.*

Therefore $w_0+\sum_{i=1}^{d}w_i\mathbf{x}_i^{(1)}>0$ *for class 1 and* $w_0+\sum_{i=1}^{d}w_i\mathbf{x}_i^{(2)}<0$ *for class 2.*

However since we can write $\mathbf{x}^*=\sum_{i=1}^{n}\alpha_i\mathbf{x}_i^{(1)}$ *we know that:*

$$w_0+\sum_{j=1}^{d}w_j\sum_{i=1}^{n}\alpha_i\left[\mathbf{x}_i^{(1)}\right]_j=\sum_{i=1}^{n}\alpha_i\left(w_0+\sum_{j=1}^{d}w_jx_i^{(1)}\right)>0$$

where $\left[\mathbf{x}_i^{(1)}\right]_j$ *denotes the jth element of the vector* $\mathbf{x}_i^{(1)}$. *Hence* $\mathbf{x}^*$ *is classified as class 1. An identical argument shows it is also classified as class 2: a contradiction.*

9. What is a *Linear Machine*? Write down a multicategory version of the Pseudoinverse rule, suitable for a Linear Machine. For a 10-category problem, how many pseudoinverses do you need to calculate?

Ans: A linear machine is a K-category linear classifier defined in terms of K linear discriminant functions: $g_k(\mathbf{x}) = w_0^{(k)} + \left[\mathbf{w}^{(k)}\right]^T \mathbf{x}$ *. Classification is performed by choosing class k if* $g_k(\mathbf{x}) > g_j(\mathbf{x}), j \neq k$ *.*

The multi-category version of the pseudoinverse rule:

$$\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_K]$$

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_K \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} B_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_K \end{bmatrix}$$

where

$\mathbf{Y}_i$ consists of feature rows attributed to the ith class

$$\mathbf{B}_i = \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \text{with 1s in the ith column}$$

Then the pseudoinverse rule is :

$$\mathbf{A} = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{B}$$

where we only have to calculate 1 pseudoinverse : $(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T$

10. Show that the decision regions of a multi-category linear machine are convex and explain the implication for the types of problem that be solved.

Ans: It is sufficient to show that given any $\boldsymbol{x_1}, \boldsymbol{x_2} \in \mathcal{R}_i$ *that* $\boldsymbol{x} = \lambda\boldsymbol{x_1} + (1-\lambda)\boldsymbol{x_2}$ *is also in region* $\mathcal{R}_i$ *for any*$0 \leq \lambda \leq 1$.

First note that

$$\boldsymbol{x_1} \in \mathcal{R}_i \Rightarrow \left(\mathbf{w_i} - \boldsymbol{w_j}\right)^{\boldsymbol{T}}\boldsymbol{x_1} + \left(w_{i0} - w_{j0}\right) > 0, \forall j \neq i$$

and

$$\boldsymbol{x_2} \in \mathcal{R}_i \Rightarrow \left(\mathbf{w_i} - \boldsymbol{w_j}\right)^{\boldsymbol{T}}\boldsymbol{x_2} + \left(w_{i0} - w_{j0}\right) > 0, \forall j \neq i$$

Therefore by linearity we have:

$$\begin{aligned}(\mathbf{w_i} - \boldsymbol{w_j})\boldsymbol{x_1} + (w_{i0} - w_{j0}) \\ &= \lambda\left[\left(\mathbf{w_i} - \boldsymbol{w_j}\right)^{\boldsymbol{T}}\boldsymbol{x_1} + \left(w_{i0} - w_{j0}\right)\right] \\ &+ (1-\lambda)\left[\left(\mathbf{w_i} - \boldsymbol{w_j}\right)^{\boldsymbol{T}}\boldsymbol{x_2} + \left(w_{i0} - w_{j0}\right)\right] > 0, \forall j \neq i\end{aligned}$$

because λ and $(1-\lambda)$are both non-negative and one must be positive. Hence $\boldsymbol{x} \in \mathcal{R}_i$ and the decision region must be convex.

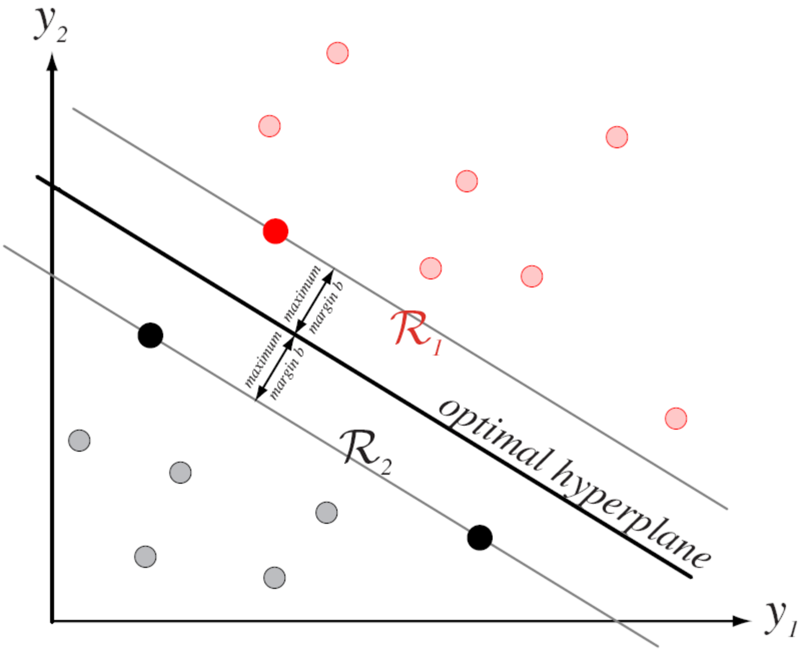
This implies that linear machines cannot classify data that cannot be grouped into into convex regions (e.g. when the data is not simply connected)

11. Show that the maximum margin classifier may only be a function of a subset of the data points

Ans: The maximum margin classifier is defined by the weight $\hat{a}$ given by

$$\hat{a} = min\, \|a\|^2, such\ that\colon \omega^{(i)} a^T y_i \geq 1$$

That the maximum margin classifier may only be a function of a a subset of the data is best seen from the following sketch:



If we write a support vector as $\boldsymbol{y} = \boldsymbol{y}_p + r$ where $\boldsymbol{y}_p$ is the projection of the support vector onto the decision boundary then any data point $\boldsymbol{y}' = \boldsymbol{y}_p + \alpha r$ for $\alpha > 1$ will not be effect the decision boundary in the sense that perturbing $\boldsymbol{y}'$will not change the maximum margin classifier.

12. In the multi-category case, a set of samples is said to be *linearly separable* if there exists a linear machine that can classify them all correctly. If any samples labelled ω_i can be separated from all others by a single hyperplane, we shall say the samples *totally linearly separable*. Show that totally linearly separable samples must be linear separable, but that the converse need not be true. (Hint for the converse simply find a counter example)

Ans:

If a set of samples is Totally Linearly Separable then there exist c hyperplanes that separate ω_i from not ω_i. Equivalently there exist c linear discriminant functions $g_i(x)$ such that:

$$g_i(x) > 0\ if\ x \in\ \omega_i$$

and

$$g_i(x) < 0\ if\ x \notin \omega_i$$

Note that such discriminant functions automatically define a linear machine since if $x \in \omega_i$ then:

$$g_i(x) > 0 > g_j(x) \; \forall \, j \neq i$$

Hence Totally Linearly Separable ⟹ Linearly Separable

To show the converse is not true we simply construct a counter example. The following data is Linearly Separable but is not Totally Linearly Separable:

$$\omega_1: (1,4), (1,2)$$

$$\omega_2: (2,4), (2,2)$$

$$\omega_3: (-1,0), (4,0)$$

(it is not possible to place a hyperplane that isolates ω_1 (or ω_2)

13. Consider a three-layer backpropagation network with d input units, n_H hidden units, c output units, and bias. The network has activation function $f(.)$, with input to hidden weights labelled w_{ji} and hidden to output weights labelled w_{kj}.
For a single input vector $\mathbf{x} = (x_1, \ldots, x_d)$ with associated target vector $\mathbf{t} = (t_1, \ldots, t_c)$, write down expressions for:
(a) the output activation y_j of a hidden unit
(b) the output activation z_k of an output unit (i.e. an output of the network)
(c) the squared error J.

Ans: Note this is basically bookwork:

a) $y_j = f\left(\sum_{i=1}^{d} w_{ji} x_i + w_{j0} \right)$

b) $z_k = f\left(\sum_{j=1}^{n_H} w_{kj} y_j + w_{k0} \right)$

c) $J(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^{c} (t_k - z_k)^2 = \frac{1}{2} \| \mathbf{t} - \mathbf{z} \|^2$

14. How many weights are in the network of Q11?

Ans: (d+1) n_H *+ (*n_H *+1)c*

15. For the network of Q11, using the chain rule, calculate expressions for the following derivatives:
(a) $\partial J / \partial z_k$ (b) $\partial J / \partial w_{kj}$ (c) $\partial J / \partial y_j$ (d) $\partial J / \partial w_{ji}$
Hence derive formulas for the weight updates required to perform steepest descent in the squared error J.
[This approach does not define the δ quantities used in the notes, but you should be able to confirm that the formulas are equivalent.]

Ans: Bookwork again. Everything can be derived from the general expression for the MLP:

$$g_k(\mathbf{x}) \equiv z_k = f\left(\sum_{j=1}^{n_H} w_{kj} f\left(\sum_{i=1}^{d} w_{ji}x_i + w_{j0}\right) + w_{k0}\right)$$

a) $\partial J / \partial z_k = (t_k - z_k)$ *note this is the squared error for a single observation (we could also have summed over n observations)*

b) $\dfrac{\partial J}{\partial w_{kj}} = \dfrac{\partial J}{\partial z_k}\dfrac{\partial z_k}{\partial net_k}\dfrac{\partial net_k}{\partial w_{kj}} = (t_k - z_k) f'(net_k) y_j$

where we are defining $net_k = \sum_j w_{kj} y_j$

c)

$$\frac{\partial J}{\partial y_j} = \frac{\partial}{\partial y_j}\left[\tfrac{1}{2}\sum_{k=1}^{c}(t_k - z_k)^2\right] = -\sum_{k=1}^{c}(t_k - z_k)\frac{\partial z_k}{\partial y_j}$$

$$= -\sum_{k=1}^{c}(t_k - z_k)\frac{\partial z_k}{\partial net_k}\frac{\partial net_k}{\partial y_j} = -\sum_{k=1}^{c}(t_k - z_k) f'(net_k) w_{kj}$$

d)

$$\frac{\partial J}{\partial w_{ji}} = \frac{\partial J}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}} = \frac{\partial J}{\partial net_j} x_i$$

$$= \frac{\partial J}{\partial y_j}\frac{\partial y_j}{\partial net_j} x_i = \frac{\partial J}{\partial y_j} f'(net_j) x_i$$

where we are defining $net_j = \sum_i w_{ji} x_i$