

「放射狀基底函數網路」，其網路架構如圖1所示，為兩層的網路；假設輸入維度是I，以及隱藏層類神經元的數目是J，那麼網路的輸入可以表示成：

$$F(\underline{x}) = \sum_{j=1}^J w_j \varphi_j(\underline{x}) + \theta$$

$$= \sum_{j=0}^J w_j \varphi_j(\underline{x})$$

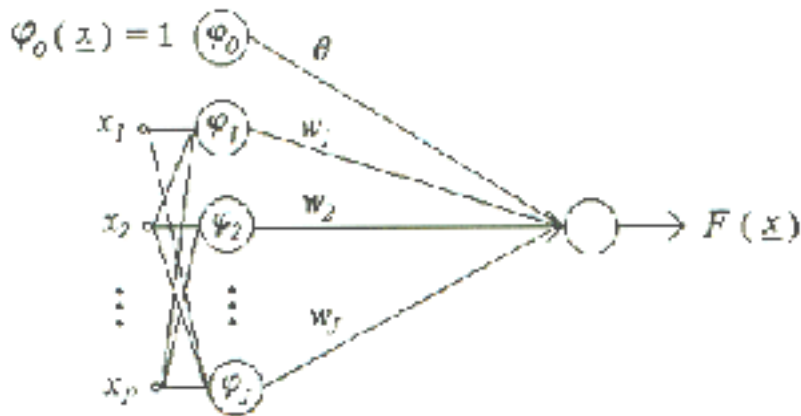


圖1：放射性基底函數網路的架構。

而 $\varphi_j(\underline{x})$ 我們選用高斯型基底函數：

$$\varphi_j(\underline{x}) = \exp\left(-\frac{\|\underline{x} - \underline{m}_j\|^2}{2\sigma_j^2}\right)$$

其中 $\underline{x} = (x_1, x_2, \dots, x_I)$ 、 $\underline{m}_j = (m_{j1}, m_{j2}, \dots, m_{ji})$ 、 $\|\cdot\|$ 代表向量絕對值

適應函數為：

$$E(n) = \frac{1}{2} \sum_{n=1}^N (y_n - F(\underline{x}_n))^2 \quad (1)$$

N：作業1產生的N筆成功到達目標的訓練資料

y_n ：表示訓練資料的方向盤期望輸出值

<為了計算方便，我們在此把訓練資料 x_i 正規化到 -1~1之間>

請用基因演算法，找出 w_j 、 m_j 、 σ_j ，在不同的數字J下，最好的基因向量。其中基因向

量維度公式為:

$$\text{Dim}(j) = 1 + j + i*j + j$$

即 $(\theta_{w_1, w_2, \dots, w_j, m_{11}, m_{12}, \dots, m_{1i}, m_{21}, m_{22}, \dots, m_{2i}, \dots, m_{j1}, m_{j2}, \dots, m_{ji}, \sigma_1, \sigma_2, \dots, \sigma_j})$

例如J為9、輸入x為3維向量，則表示基因向量是 $1+9+3*9+9=46$ 維度的向量；又例如J為7、輸入x為3維向量，則表示基因向量是 $1+7+3*7+7=36$ 維度的向量。演算的目標是使得評估函數 $E(n)$ (式1) **越小越好**。

基因向量中

θ 範圍為 -1~1之間

w_j 範圍為 -1~1之間

m_{ji} 範圍跟 x_i 範圍一樣

根據高斯函數的原理， σ_j 範圍則為0~1之間。但同學們也可以嘗試以大一點的值來初始化，觀察訓練過程會有什麼改變

由於 $\varphi_j(x)$ 的值介於0~1之間，**若我們把期望輸出值 y_n 由 -40 ~ 40度正規化到 -1~1之間**，即可保證落於網路的輸出值域之中，使得網路有能力逼近我們的期望值。

舉例而言，假設目前有一組4D輸入資料[22, 8.4, 8.4, 0]，

若我們以[0,80]作為距離的值域區間，[-40,40]作為角度的值域區間對資料正規化，

可得到[-0.45, -0.79,-0.79, 0]

若J取1，則基因向量為6維，假設為[1,1,1,1,1,1]

$$\text{則 } \varphi_1(x) = 0.0142$$

$$\text{則 } F(x) = 1.0142, E(n) = 0.5143$$

將 $F(x)$ 反正規化，即得到網路對於目前3個距離的響應角度。由上述的例子可以看出，網路的輸出不見得會落在-40~40之間，同學們可以想想看該如何處理這個問題。

基因演算法中需要包含**[複製，輪盤式選擇，競爭式選擇，交配，突變]**這些步驟
程式要能設定迭代次數，族群大小，突變機率，交配機率以及網路J值。

訓練出來的網路優劣可由平均誤差值 $Error(t) = \frac{\sum_1^N \|(y_n - F(\underline{x}_n))\|}{N}$ 來判斷

請找出一組設定值使得程式在數百次迭代後能夠通過軌道，並請跑出車子軌跡。