APRENDIZAJE ASOCIATIVO

Red Instar: Hasta ahora se han considerado solamente reglas de asociación entre entradas y salidas escalares. Si se examina la red de la figura 1, se nota como esta neurona esta enfrentada a un problema de reconocimiento de patrones cuya entrada es de tipo vectorial; esta neurona es el tipo de red más simple capaz de resolver esta clase de problemas y es llamada red Instar.

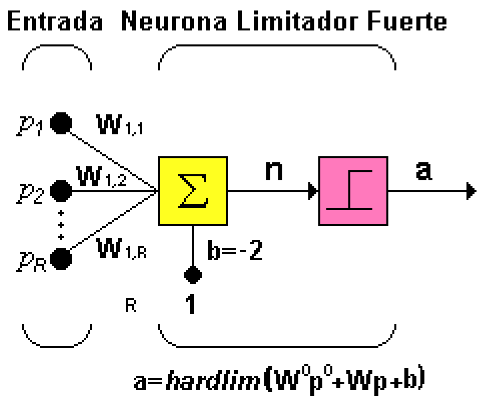

Figura 1. Red Instar.

Puede notarse la similitud entre la red Instar y la red Perceptrón o incluso a la red Adaline. Estas redes han tenido diferentes nombres, debido a razones históricas y a que su desempeño ha sido analizado en diferentes ambientes. Para la Instar no se considerará directamente su característica de decisión, concepto que fue bastante importante para el Perceptrón, en lugar de ello se analizará la capacidad de la Instar para reconocimiento de patrones a través de **asociaciones** y **aprendizaje no supervisado**.

La ecuación para determinar la entrada/salida de la Instar es:

$$a = hardlims\,(\mathbf{w}^T\mathbf{p}+b) \quad (1)$$

La red Instar se activará si el **producto punto** entre el vector de pesos (fila de la matriz de pesos) y la entrada sea mayor o igual a $-b$

$$\mathbf{w}^T\mathbf{p} \geq -b \quad (2)$$

Los vectores $\boldsymbol{w}$ y $\boldsymbol{p}$ son de longitud constante, por lo tanto el mayor producto punto se presentará cuando los dos vectores apunten en la misma dirección; dicho de otra forma cuando el ángulo entre $\boldsymbol{w}$ y $\boldsymbol{p}$ sea $\theta = 0$, esto permite observar que la red instar de la figura 1. se activará cuando $\boldsymbol{p}$ y $\boldsymbol{w}$ estén muy cercanos, escogiendo un apropiado valor para la ganancia b se puede determinar que tan cerca deben estar $\boldsymbol{p}$ y $\boldsymbol{w}$ para que la instar se active, si se fija

$$b = -\|\boldsymbol{w}\| \cdot \|\boldsymbol{p}\| \quad (3)$$

la instar se activará solamente cuando $\boldsymbol{p}$ apunte exactamente en la misma dirección de $\boldsymbol{w}$, de esta forma b se puede incrementar a valores ligeramente mayores a $-\|\boldsymbol{w}\| \cdot \|\boldsymbol{p}\|$, el mayor valor de b se presentará cuando la Instar esté activa. Es importante recalcar que este análisis asume que todos los vectores tienen la misma longitud.

Uno de los inconvenientes de la regla de Hebb con el **factor de olvido**, es que requiere que los estímulos se presenten de forma repetitiva o de lo contrario se perderá la asociación, se desea encontrar una regla alternativa que habilite el término con olvido solo cuando la Instar es activa $a \neq 0$, de esta forma los valores de los pesos seguirán siendo limitados, pero el porcentaje de olvido será minimizado. Para obtener los beneficios del término de peso con factor de olvido, se adiciona un nuevo término proporcional a $a_i(q)$.

$$w_{ij}(q) = w_{ij}(q-1) + \alpha\, a_i(q) p_j(q) - \gamma a_i(q) w_{ij}^{anterior} \quad (4)$$

El nuevo término de peso se hace proporcional a la salida escalar $a_i(q)$, ya que se desea controlar esta salida para que reproduzca el estímulo no condicionado; si se considera que el factor a la cual la red aprende nuevos pesos es igual al factor de olvido $\alpha = \gamma$, la ecuación (4) puede simplificarse como:

$$w_{ij}(q) = w_{ij}(q-1) + \alpha\, a_i(q)\left(p_j(q) - w_{ij}^{anterior}\right) \quad (5)$$

Esta ecuación es la llamada regla de Instar.

La regla de Instar en forma vectorial teniendo en cuenta el caso en que la instar esta activa (a_i=1), se convierte en:

$$\mathbf{w}(q) = \mathbf{w}(q-1) + \alpha\,(\mathbf{p}(q) - \mathbf{w}(q-1))$$

$$= (1-\alpha)\,\mathbf{w}(q-1) + \alpha\,\mathbf{p}(q) \quad (6)$$

Esta operación se muestra en la figura 2.

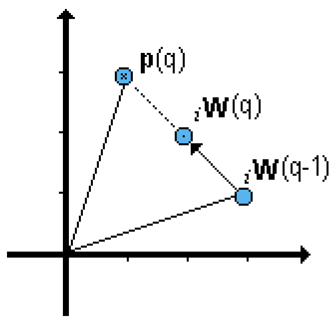

Figura 2. Representación gráfica de la regla de Instar

Cuando la instar se activa, el vector de pesos se mueve hacia el vector de entrada a lo largo de una línea entre el vector de pesos anterior y el vector de entrada. La distancia a la que se mueve el vector depende del valor del factor de aprendizaje α :

Cuando α =0, el nuevo vector de pesos es igual al vector de pesos anterior. **Cuando** α =1, el nuevo vector de pesos es igual al vector de entrada. **Si** α =0.5 el nuevo vector de pesos será la mitad entre el vector de pesos anterior y el vector de entrada.

Una característica útil de la regla Instar es que si los vectores de entrada son normalizados, entonces $\boldsymbol{w}$ será también normalizado una vez la red haya aprendido un vector particular $\boldsymbol{p}$, esta regla no solamente minimiza el factor de olvido, también normaliza los vectores de peso si el vector de entrada es normalizado.

Se aplicará la regla de Instar para solucionar el problema de la figura 3, similar al problema del asociador para una fruta; este nuevo caso cuenta con dos entradas, una indicando si la fruta ha sido visualizada o no (estímulo no condicionado) y otra consistente en un vector de tres medidas pertenecientes a la fruta (estímulo condicionado).

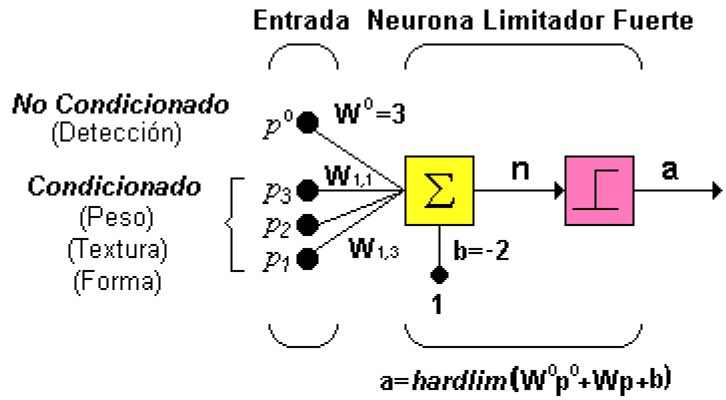

Figura 3. Reconocimiento de una fruta por medio de una Instar

La salida de la red, está determinada por

$$a = hardlim(w^0 p^0 + \boldsymbol{Wp} + b)$$

Los elementos de **entrada** a la red serán valores de 1 o –1, las tres propiedades que se medirán de la fruta son: forma, textura y peso, de esta manera la salida del sensor de forma será 1 si la fruta es aproximadamente redonda o –1 si la fruta es elíptica, la salida del sensor de textura será 1 si la superficie de la fruta es suave y será –1 si es rugosa y la salida del sensor de peso será 1 si la fruta pesa más de una libra o –1 si el peso de la fruta es menor de esta medida.

En este caso la elección del estímulo condicionado y el no condicionado ya no es aleatoria, pues como se dijo en análisis anteriores, el estímulo no condicionado se convierte la mayoría de las veces en la salida deseada de la red que es tipo de escalar para una red Instar, por lo tanto el sensor que representa la visualización de la red será el estímulo no condicionado y el vector de medidas de la fruta será el estímulo condicionado.

Con las dimensiones consideradas $\boldsymbol{p}$ es un vector normalizado con $\|p\| = \sqrt{3}$. La definición de p^0 y $\boldsymbol{p}$ es:

$$p^0 = \begin{Bmatrix} 1 & fruta\ \ detectada\ visualmente \\ 0 & fruta\ \ no\ \ detectada \end{Bmatrix} \qquad \boldsymbol{p} = \begin{bmatrix} forma \\ textura \\ peso \end{bmatrix}$$

El valor de la ganancia se asumirá como b = -2, un valor ligeramente más positivo que $-\|p\|_2$ = -3. Lo ideal es que la red tenga una asociación constante, entre la visualización de la fruta y su respuesta, para que w^0 sea mayor que – b.

Inicialmente la red no responderá a ninguna combinación de medidas de la fruta, puesto que la fruta no ha sido detectada visualmente, así que los pesos iniciales serán cero

$$w^0=3,\ \mathbf{W}(0) = {}_1\mathbf{w}^T(0) = [0\ 0\ 0] \quad (7)$$

Usando la regla Instar con un factor de aprendizaje $\alpha=1$, los pesos actualizados se encontrarán de la siguiente forma:

$$w(q) = w\,(q\,-1) + a(q)\,(p(q) - w(q\text{-}1)) \quad (8)$$

La secuencia de entrenamiento consistirá en repetidas presentaciones de la fruta, los sensores estarán actuando todo el tiempo sin embargo, en orden a observar la operación de la regla Instar se asumirá que el sensor que visualiza la fruta actuará intermitentemente, simulando así una falla en su construcción

$$\left\{p^0(1) = 0,\ \mathbf{p}(1) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}\right\},\ \left\{p^0(2) = 1,\ \mathbf{p}(2) = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}\right\} \quad (9)$$

Como la matriz $\boldsymbol{W}$ inicialmente contiene ceros, la Instar no responderá a los sensores de la fruta en la primera iteración

$$a(1)=hardlim(w^0\,p^0(1) + \boldsymbol{Wp}\,(1) - 2)$$

$$a(1) = hardlim\left(3*0 + \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - 2\right) = 0 \quad \textit{No hay respuesta} \quad (10)$$

Como la neurona no respondió, sus pesos no serán actualizados por la regla Instar

$$w(0) = w(0) + a\,(1)(p(1) - w\,(0))$$

$$= \begin{bmatrix}0\\0\\0\end{bmatrix} + 0\left(\begin{bmatrix}1\\-1\\1\end{bmatrix} - \begin{bmatrix}0\\0\\0\end{bmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix}\right) \quad (11)$$

En la **segunda iteración**, cuando la fruta haya sido detectada visualmente, la neurona responderá

$$a(2) = hardlim\,(w^0 p^0(2) + \boldsymbol{Wp}\,(2) - 2)$$

$$a(2) = hardlim\left(3 * 1 + \begin{bmatrix}0 & 0 & 0\end{bmatrix}\begin{bmatrix}1\\-1\\-1\end{bmatrix} - 2\right) = 1 \quad \textit{fruta detectada} \quad (12)$$

El resultado es que la red aprendió a asociar el vector de medidas de la fruta con su respuesta. El vector de pesos de la red, se convierte en una copia del vector de medidas de la fruta.

$$w(2) = w(1) + a\,(2)(p(2) - w\,(1))$$

$$= \begin{bmatrix}0\\0\\0\end{bmatrix} + 1\left(\begin{bmatrix}1\\-1\\1\end{bmatrix} - \begin{bmatrix}0\\0\\0\end{bmatrix} = \begin{bmatrix}1\\-1\\-1\end{bmatrix}\right) \quad (13)$$

La red puede ahora reconocer la fruta por sus medidas;

la neurona respondió en la tercera iteración, aún cuando el sistema de detección visual falló, por lo tanto la red realizará una asociación entre la presencia de la fruta y el vector de estímulos condicionados, sin importar si el sensor de visualización (estímulo no condiconado) opera adecuadamente.

$$a(3) = hardlim\,(w^0 p^0(3) + \boldsymbol{Wp}\,(3) - 2)$$

$$a(3) = hardlim\left(3*0 + \begin{bmatrix} 1 & -1 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - 2\right) = 1 \quad fruta\ detectada \qquad (14)$$

Cuando las medidas de la fruta han sido detectadas completamente, los pesos dejan de cambiar y se estabilizan.

$$w(3) = w(2) + a\,(3)(p(3) - w\,(2))$$

$$= \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + 1\left(\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}\right) \qquad (15)$$