

## PERCEPTRÓN MULTICAPA

### Regiones de Decisión

La capacidad para resolver distintos problemas de una red multicapa, viene dada por la no linealidad que existe en cada uno de los nodos.

Una red con una capa intermedia, genera una Región de Decisión que está limitada por la intersección de los hiperplanos generados por cada una de las neuronas de la capa intermedia.

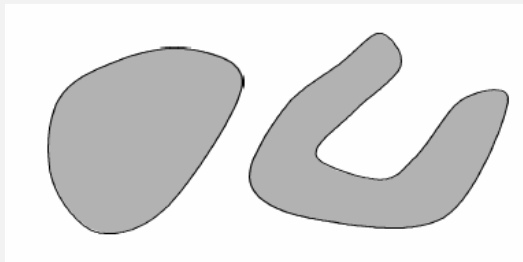
Para redes con una capa intermedia estas regiones son **convexas**, es decir que el camino recorrido entre dos puntos cualesquiera de su borde está compuesto solamente por puntos pertenecientes a la misma región.

Para generar regiones de decisión **no convexas**, se deben utilizar redes con dos capas intermedias.

Cuando la función no lineal es del tipo sigmoidea, los límites de las Regiones de Decisión toman formas de curvas suaves.

Región convexa

Región no convexa



## MODO DE OPERACIÓN AUTOASOCIATIVO

El método BP es del tipo supervisado, es decir que para cada patrón de entrenamiento  $X_j$  es necesario contar con el correspondiente valor deseado de salida  $D_j$ .

Cuando el valor deseado es el mismo patrón de entrada, este modo de operación de la red se denomina **autoasociativo**.

En consecuencia la señal de entrada pasa por la capa intermedia y se reproduce en la capa de salida.

Si bajo estas condiciones la red tiene una capa intermedia con pocas neuronas, se puede concluir que estas pocas neuronas intermedias mantienen prácticamente toda la información de la señal de entrada.

Se produce un proceso de compresión de datos.

Las neuronas intermedias extraen y almacenan las principales componentes de la señal de entrada, con las cuales es posible reproducir la señal a la salida.

La extracción de las características principales de la señal es una propiedad de las neuronas intermedias.

Con este tipo de operación de la red, se obtiene una representación compacta del dominio de trabajo.

## MODOS DE ENTRENAMIENTO

**Modo patrón:** los patrones se presentan de a uno por vez y se implementan los procesos de cálculo "hacia adelante" y "hacia atrás". Durante el proceso "hacia atrás" se realiza la adaptación de los pesos en cada conexión y para cada patrón presentado, de acuerdo a la ecuación:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)$$

**Modo Batch:** se presentan los patrones de uno por vez, se implementa el cálculo del proceso "hacia adelante" y en el proceso "hacia atrás" solamente se calculan los gradientes locales para cada patrón presentado, pero sin realizar el ajuste de los pesos.

Los gradientes calculados se acumulan en cada iteración y al final del epoch (presentación de todo el conjunto de patrones), se realiza un único cálculo de adaptación de los pesos, utilizando un gradiente promedio mediante la ecuación:

$$\Delta w_{ji}(n) = -\frac{\eta}{N} \sum_{n=1}^N e_j(n) \frac{\partial e_j(n)}{\partial w_{ji}(n)}$$

Este modo tiene mayor cantidad de cálculos, por lo tanto es más lento y en consecuencia poco utilizado. Sin embargo existen aplicaciones donde este modo demostró ser más eficiente.

## GENERALIZACIÓN

Concluido el proceso de entrenamiento de una red, es necesario verificar su comportamiento frente a patrones no incluidos en el conjunto de patrones de entrenamiento. Esta etapa se denomina proceso de test de la red.

Mediante el proceso de test se determina la capacidad de generalización de la red.

## Definición:

Una red se dice que generaliza en forma correcta cuando la relación entrada/salida es satisfactoria para un grupo de patrones diferente al utilizado durante el proceso de entrenamiento.

Durante el proceso de test los pesos no se modifican.

El comportamiento de la red se mide a través del cálculo del error promedio para todo el conjunto de test (Error de Generalización).

La generalización es influenciada por tres factores:

- ✓ Tamaño y calidad del conjunto de patrones de entrenamiento.
- ✓ Arquitectura de la red.
- ✓ Complejidad del problema a resolver.

Una regla práctica para obtener una buena generalización es utilizar la menor arquitectura de la red que resuelva eficientemente el problema a considerar.

No es fácil determinar cual es, para cada caso, esta menor arquitectura.

Una primer aproximación sería comenzar con una arquitectura mínima e ir aumentando la misma a medida que los procesos de entrenamiento no verifiquen las condiciones de convergencia.

Este proceso tiene dos inconvenientes:

- ✓ Lentitud.
- ✓ Una arquitectura pequeña es más sensible a los valores de inicialización y en consecuencia es más probable que el sistema quede atrapado en un mínimo local.

Otra aproximación a una arquitectura de óptima generalización es comenzar con una red sobredimensionada, para reducirla, eliminando los pesos con menor incidencia en la solución.

Si se comparan los procesos de entrenamiento y de test, referidos al error que ocurre en cada proceso en función del tiempo de entrenamiento, se detecta que para tiempos de entrenamientos grandes el error del test puede volver a aumentar, aun cuando el error de entrenamiento sigue disminuyendo.

Este problema, denominado sobreentrenamiento, ocurre debido a que la red comienza a reconocer regularidades espúreas al conjunto de patrones.

El problema de sobreentrenamiento es más probable que ocurra cuando la red está sobredimensionada, es decir que tiene un mayor número de neuronas y pesos que los necesarios.

Cuando el sistema tiene menos grados de libertad (pesos), estos se utilizan para adaptarse a las principales regularidades de los datos, ignorando las regularidades menores generalmente no representativas del problema.

En consecuencia partiendo de una red sobredimensionada es necesario su reducción, mediante metodologías que permiten reducir el número de pesos eliminando aquellos que tengan una menor influencia sobre la solución definitiva, estas metodologías se denominan *Procesos de Pruning*.

## Procesos de "Pruning"

Existen dos grupos de metodologías para implementar el proceso de Pruning en una red sobredimensionada.

Un primer grupo utiliza el cálculo de un factor de sensibilidad.

Este factor de sensibilidad para cada peso tiene en cuenta la diferencia entre los errores de la red, durante el entrenamiento, con o sin el peso, según la siguiente expresión:

$$S_{ji} = - \frac{E(w^f) - E(w^r)}{w^f - w^r} w^f$$

Donde  $w^f$  es el valor final del peso después del entrenamiento y  $w^r$  es el valor del peso cuando se lo retira.

En la práctica no se implementa este proceso de eliminación de pesos a través del cálculo de errores, por ser poco eficiente respecto al tiempo de cómputo.

En su lugar se utiliza una aproximación que tiene en cuenta la suma de todos los cambios experimentados por los pesos durante el proceso de entrenamiento.

Si un peso ha cambiado mucho durante el entrenamiento es probable que haya acumulado mucha información.

La sensibilidad estimada para uno de los métodos viene dada por:

$$S_{ji}^* = - \sum_{n=0}^{N-1} \frac{\partial E}{\partial w_{ji}} \Delta w_{ji}(n) \frac{w_{ji}^f}{w_{ji}^f - w_{ji}^i}$$

Donde **N** es el número de épocas de entrenamiento y  $w_{ji}^i$  es el valor inicial del peso.

Cada uno de estos términos son accesibles durante el entrenamiento y por lo tanto es factible el cálculo del factor de sensibilidad.

Después del entrenamiento cada peso tiene asignada una estimación del factor de sensibilidad, siendo eliminados aquellos pesos con un menor valor del factor.

En el caso de que todas las conexiones de un nodo son eliminadas, ese nodo también puede ser eliminado.

Una vez reducida la estructura se vuelve a entrenar la red.

El otro grupo de métodos implementa una modificación de la función error que utiliza el método de Backpropagation, mediante el cual se cuantifican los cambios de ese peso durante su entrenamiento.

Luego es el propio método de Backpropagation el que se encarga de realizar la eliminación de los pesos haciendo cero su valor durante el entrenamiento.

## SELECCIÓN DE LOS DATOS

El conjunto de datos utilizados para el procedimiento de Backpropagation debe ser depurado, eliminando aquellos datos dudosos o incompletos.

Para obtener buenos resultados es conveniente elegir el conjunto de patrones de entrenamiento de manera que:

- ✓ los datos deben estar igualmente distribuidos entre los distintos grupos de salida.
- ✓ los datos deben ser representativos de todo el problema.

### Consideraciones para la selección de los datos:

#### ➤ Consideraciones relacionadas con la aplicación:

- ✓ Es factible identificar cuales son las características de la aplicación conocidas en forma directa o bajo control?
- ✓ Es factible detectar los aspectos relevantes?

➤ Consideraciones sobre los datos:

- ✓ Complejidad de su obtención.
- ✓ Es posible medir las características que deben ser utilizadas?
- ✓ Tamaño del conjunto de patrones de entrenamiento.

**Objetivo:**

Establecida una configuración inicial de la red, determinar el número de patrones de entrenamiento apropiado para lograr una eficiente generalización.

Teniendo en cuenta que se cuenta con una configuración dada, el número de pesos de la red es conocido, por lo tanto el tamaño del conjunto de patrones de entrenamiento se relaciona con el número de pesos de la red.

Una regla práctica para definir el número de patrones de entrenamiento ( **$Np$** ) necesarios para lograr una buena generalización, nos dice que  **$Np$**  debe ser aproximadamente diez veces el número de pesos de la red (para una red multicapa perceptrón).

Esta regla tiene implicancias prácticas para aquellos problemas donde el número de pesos de la red es grande, puesto que genera dificultad para obtener el elevado número de patrones de entrenamiento.