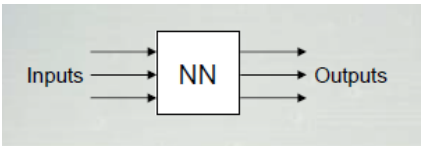
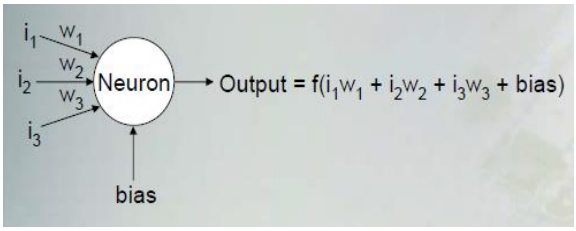
TEORIA

Las redes neuronales artificiales consisten en un conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitirse señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo unos valores de salida.



Cada neurona está conectada con otras a través de unos enlaces. En estos enlaces el valor de salida de la neurona anterior es multiplicado por un valor de **peso**. Estos pesos en los enlaces pueden incrementar o inhibir el estado de activación de las neuronas adyacentes.

Del mismo modo, a la salida de la neurona, puede existir una función limitadora o **umbral**, que modifica el valor resultado o impone un límite que se debe sobrepasar antes de propagarse a otra neurona. Esta función se conoce como **función de activación**: la salida de una neurona es la suma ponderada de las entradas más un bias.



El **bias** se considera un peso más (w0) con una entrada fija (i0 → 1 o -1). La función de toda la red neuronal es simplemente el cálculo de las salidas de todas las neuronas (determinista)

Estos sistemas aprenden y se forman a sí mismos, en lugar de ser programados de forma explícita, y pueden ser usadas en problemas de clasificación (reconocer patrones, imagenes…) o en predicción (extrapolación basada en datos anteriores). Tienen la habilidad para generalizar salidas razonables para entradas para las que no ha sido enseñada.

Para realizar este aprendizaje automático, normalmente, se intenta minimizar una función de pérdida que evalúa la red en su total. Los valores de los pesos de las neuronas se van actualizando buscando reducir el valor de la función de pérdida. Este proceso se realiza mediante la **propagación hacia atrás**.

Durante el **entrenamiento**, se presenta la red con algunos datos de muestra y se modifican los pesos para aproximarse a la función deseada. El objetivo de la red neuronal es resolver los problemas de la misma manera que el cerebro humano, aunque las redes neuronales son más abstractas.

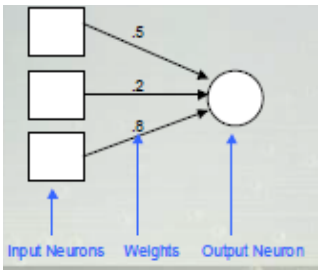
Tipos de entrenamiento

- **Supervisado:** se suministra a la red neuronal las entradas y las salidas deseadas y se mide la respuesta de la red a las entradas.

- **Sin supervisión:** Solo se suministran las entradas y la red neuronal ajusta sus propios pesos para que entradas similares causen salidas similares.

Perceptron

Es la forma más simple de una red neuronal con la capacidad de aprender. Se compone sólo de neuronas de entrada y de salida. Las neuronas de entrada suelen tener dos estados: ON y OFF. Se utiliza una **función de activación simple** y realiza un **entrenamiento supervisado**. Con un perceptron, solo se pueden resolver **problemas linealmente separables**.

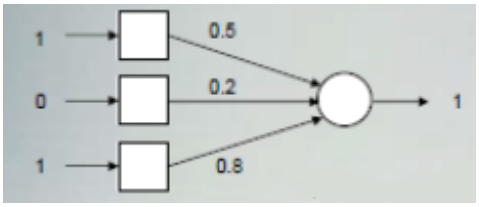


Se realiza el producto escalar entre las entradas y sus pesos ($w_1*i_1 + w_2*i_2$ …) y si este es superior al umbral, la salida es 1, sino es 0. Si la salida no es la esperada, se ajustan los pesos siguiendo esta fórmula, donde α es el ratio de aprendizaje:

$$w_{nuevo} = w_{antiguo} + \alpha * (outputDeseado - outputObtenido) * input$$

En este ejemplo, calcularíamos el siguiente producto escalar:

$$1 * 0.5 + 0 * 0.2 + 1 * 0.8 = 1.3$$



Suponiendo un umbral de 1.2, como 1.3 > 1.2, la salida obtenida sería 1. Imaginamos que la salida que se esperaba obtener es 0, se reajustarían los pesos. (asumimos $\alpha = 1$).

$$w_{1nuevo} = 0.5 + 1 * (0 - 1) * 1 = -0.5$$
$$w_{2nuevo} = 0.2 + 1 * (0 - 1) * 0 = 0.2$$
$$w_{3nuevo} = 0.8 + 1 * (0 - 1) * 1 = -0.2$$

Perceptron Multicapa (MLP)

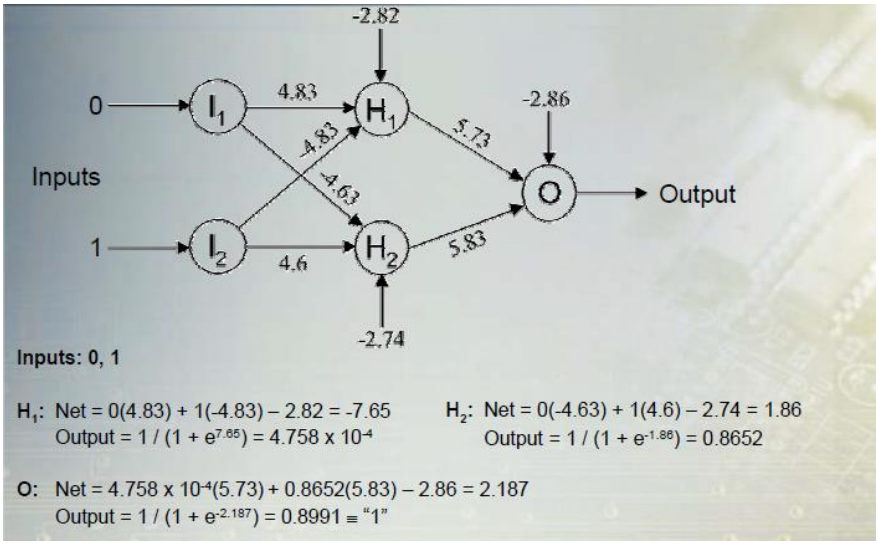
El perceptrón multicapa es una red neuronal formada por múltiples capas, de tal manera que tiene capacidad para resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón. Es la red neuronal más común, es un **entrenamiento supervisado** y se utiliza una **función de activación sigmoid**. El perceptrón multicapa puede estar totalmente o localmente conectado.:

- **Totalmente:** cada neurona es entrada de todas las neuronas de la capa siguiente.
- **Localmente:** cada neurona es entrada de algunas neuronas de la capa siguiente.

Tipos de capas

- **Capa de entrada:** Neuronas que introducen los valores de entrada en la red. No realizan ningún procesamiento.
- **Capas ocultas:** Neuronas cuyas entradas provienen de capas anteriores y cuyas salidas pasan a neuronas de capas posteriores.
- **Capa de salida:** Neuronas cuyos valores de salida se corresponden con las salidas de toda la red.

La **propagación hacia atrás** es un algoritmo utilizado en el entrenamiento de estas redes. De esta manera se obtienen los pesos de la red y se modifican para mejorarla. El algoritmo básico se basa en minimizar el error de la red usando derivadas de la función de error.



Como función sigmoid, se usa la función logística:

f (x) = 1 / (1 + e-x)

-La manera más común de medir el error es:

E = (target – output)^2

El cálculo de las derivadas se propaga por la red. Estas apuntan en la dirección de máximo incremento de la función error.

El ratio de aprendizaje es importante:

-Si es muy pequeño, la convergencia puede ser muy lenta.

-Si es muy grande, puede no convergir.

Para la mayoría de problemas, con una capa es suficiente. Se usan dos cuando la función es discontinua. El número de neuronas también es importante:

-Si hay pocas, no puede aprender los detalles

-Si hay muchas, aprende cosas insignificantes y se sobrecargan los datos (overfitting)

-Se empieza con pocas y se va subiendo poco a poco hasta tener resultados satisfactorios.

El Perceptrón Multicapa no extrapola bien, es decir, si la red se entrena mal o de manera insuficiente, las salidas pueden ser imprecisas. La existencia de mínimos locales en la función de error dificulta considerablemente el entrenamiento, pues una vez alcanzado un mínimo, el entrenamiento se detiene aunque no se haya alcanzado la tasa de convergencia fijada.

Cuando caemos en un mínimo local sin satisfacer el porcentaje de error permitido, se puede:

- Cambiar el número de capas y de neuronas
- Cambiar los pesos iniciales
- Cambiar los parámetros de aprendizaje
- Cambiar el conjunto de entrenamiento

El conjunto de todas las muestras se divide en 2 grupos independientes:

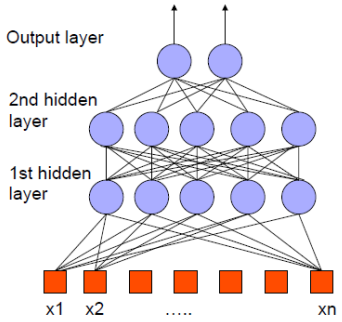
- Entrenamiento
- Validación

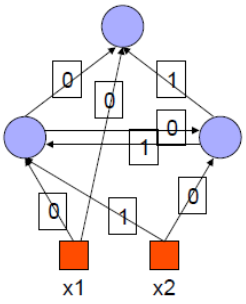
Un error común es evaluar la red con las muestras usadas para entrenarla.

Varias redes neuronales

Sirven para clasificar, discriminar y estimar. Existen 2 tipos:

- **Pre-alimentación:** La información va de las entradas a las salidas. No existen ciclos.
- **Recurrente:** Topologías arbitrarias. Contiene ciclos. Mayor dificultad de entrenamiento





La respuesta deseada de una red en una función con unas entradas particulares es conocida. El "profesor" proporciona ejemplos y entrena a la red a como realizar cierta tarea.

Aprendizaje no supervisado

Grupo típico de datos de entrada en función de unos criterios de semejanza desconocidos a priori.. No se necesita un "profesor" → La red encuentra ella sola la correlación entre los datos.

Para crear una red neuronal, se necesita:

- Elegir las entradas relevantes
- Colección de datos para las fases aprendizaje y la evaluación de la red
- Encontrar el número óptimo de capas y neuronas
- Realizar el entrenamiento y la evaluación de la red
- Si la actuación no es satisfactoria, repetir los pasos.

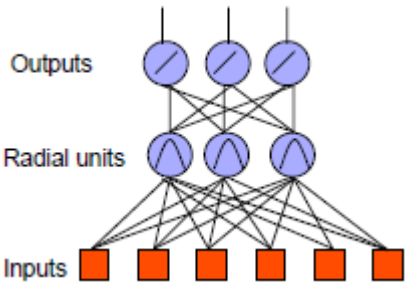
Además de los perceptrones y los MLP, existen otras arquitecturas neuronales:

Red de funciones de base radial (RBF)

Es una función de valor real cuyo valor depende sólo de la distancia desde algún otro punto c, llamado centro:

$$\phi(x) = f(||x-c||)$$

Cualquier función ϕ que satisface la propiedad ϕ(x) = f(||x-c||) es una función radial. La distancia es usualmente la distancia Euclídea. La salida común de una función de base radial es la función Gaussiana. Tiene una capa oculta y la activación de esta se determina por la función de base radial.



Entrenamiento

El entrenamiento se realiza decidiendo cuantos nodos ocultos deben haber y los centros de las Gaussianas. En el primer paso, el conjunto de datos de entrada se usan para determinar los

parámetros del RBF. En el segundo paso, los RBF se mantienen fijos mientras los pesos de la segunda capa son aprendidos (backpropagation).

Mapa autoorganizado (SOM)

Es un tipo de red neuronal artificial, que es entrenada usando **aprendizaje no supervisado** para producir una representación discreta del espacio de las muestras de entrada, llamado mapa. Usan una función de vecindad para preservar las propiedades topológicas del espacio de entrada.

Al igual que la mayoría de las redes neuronales artificiales, los SOMs operan en dos modos: entrenamiento y mapeo. En el entrenamiento construye el mapa usando ejemplos entrenantes, mientras que en el mapeo clasifica una nueva entrada.

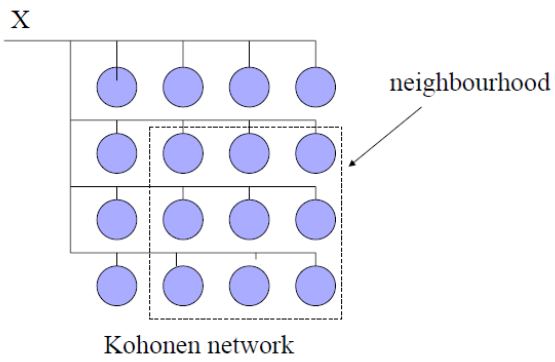
El modelo fue descrito por primera vez como una red neuronal artificial por el profesor finlandés Teuvo Kohonen (1980). Existen 4 requisitos:

- Las entradas se presentan a todas las neuronas y cada una produce una salida.
- La salida mide las diferencias entre la entrada y los valores almacenados en la neurona
- Se selecciona la neurona con mayor respuesta
- Método para reforzar la mayor respuesta.

Las neuronas no están conectadas y se colocan en cuadrículas de 2 dimensiones.

Se normalizan las entradas y los pesos y no existen umbrales ni bias. La salida de cada neurona es la suma de los pesos. Siguen la estrategia de **"winner takes all"**:

- Los pesos se inicializan aleatoriamente y todas las neuronas reciben las entradas
- Se comparan todas las salidas y el ganador es que tenga un valor de salida mayor
- Los pesos del ganador son ajustados, juntamente con las de las neuronas cercanas a esta.



Kohonen network

Conforme avanza el entrenamiento, el vecindario se hace más pequeño. Los pesos se ajustan siguiendo esta fórmula:

$$w_{nuevo} = w_{antiguo} + \alpha * (input - w_{antiguo})$$

En este ejemplo, tenemos unas entradas de [0.6, 0.6, 0.6] y 2 neuronas con peso [0.5, 0.3, 0.8] y [-0.6, -0.5, 0.6].

Calculamos la suma ponderada de los pesos:
Neurona 1 → 0.6 * 0.5 + 0.6 * 0.3 + 0.6 * 0.8 = 0.96
Neurona 2 → 0.6 * -0.6 + 0.6 * -0.5 + 0.6 * 0.6 = -0.3

Por tanto, ganaria la neurona 1. Ajustamos sus pesos con un α de 0.4:

$$w1 = 0.5 + 0.4 *(0.6 – 0.5) = 0.54$$
$$w2 = 0.3 + 0.4 *(0.6 – 0.3) = 0.42$$
$$w3 = 0.8 + 0.4 *(0.6 – 0.8) = 0.72$$

PREGUNTAS EXAMEN MASIP

1. Se puede afirmar que, en esencia, una red neuronal artificial es un aproximador de funciones.

2. Que una red neuronal sea capaz de generalizar quiere decir que produce salidas razonables para entradas para las cuales no ha sido entrenado.

3. Habitualmente en una red neuronal la función de activación NO se aplica a la suma de las entradas de una neurona, la cual se multiplica por un peso (weight) para producir la salida.

4. El mecanismo de entrenamiento en las redes tipo SOM (Self Organising Feature Maps) es el "winner takes all" que determina la neurona ganadora (winner) que ajusta sus pesos.

5. Las redes neuronales recurrentes pueden tener conexiones entre unidades que conformen un ciclo.

6. Entrenamiento / training: presentar a una red neuronal con varios datos de muestra de manera que ajuste sus pesos para aproximar mejor la función deseada.

7. Hidden layer: el número de neuronas en la capa oculta no debe ser demasiado grande para evitar el overfitting.

8. Usar un conjunto de muestras (samples) de test diferente del conjunto de entrenamiento (training set) es lo mejor para verificar el comportamiento de una red neuronal.

9. La diferencia entre el aprendizaje supervisado y el no supervisado es que el aprendizaje no supervisado no necesita un "teacher" pues la red encuentra por sí misma las correlaciones entre los datos de entrada.

10. Un tipo alternativo de red neuronal es la red RBF (Radial Basis Function) que se caracteriza porque la activación de las unidades de la capa oculta (hidden layer) viene determinada por una función RBF que es cualquier función real cuyo valor depende solo de la distancia a un punto c llamado un centro.

PRACTICAS

- Session 1 → Perceptron
- Session 2-4 → MLP