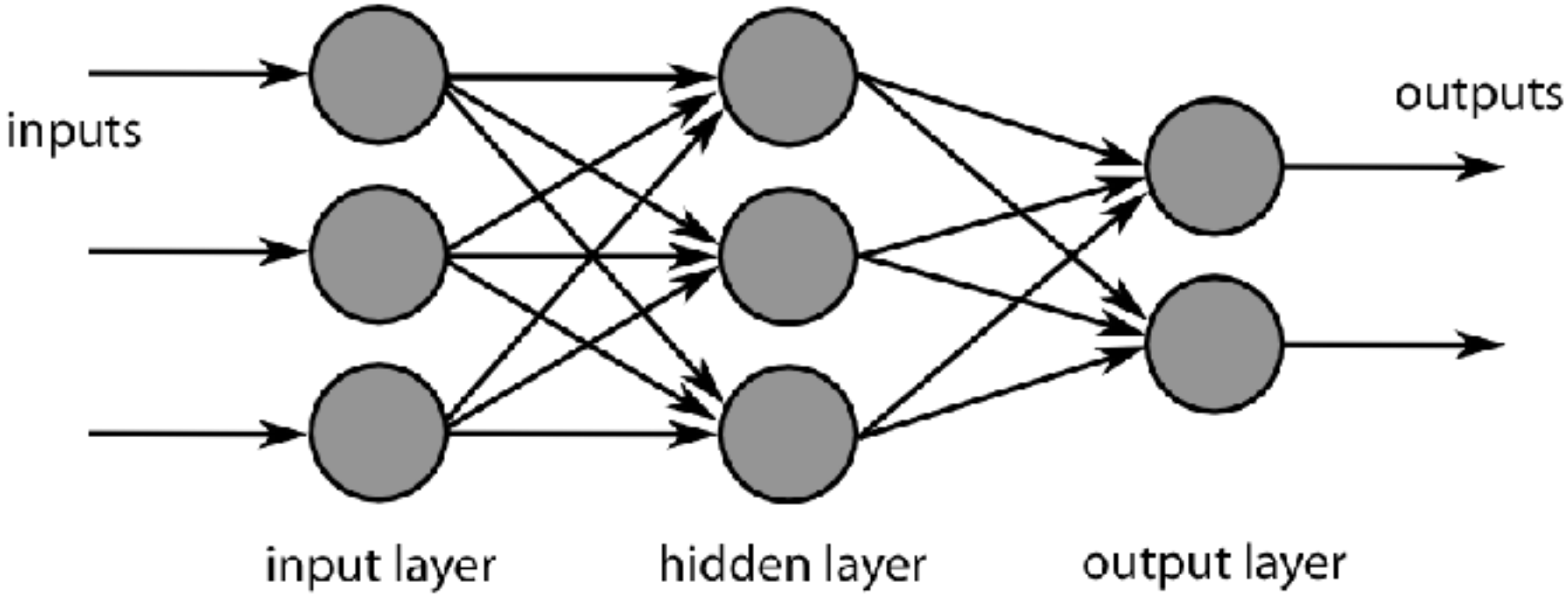
# Redes Neuronales Feed-Forward (RNA pre-alimentada)

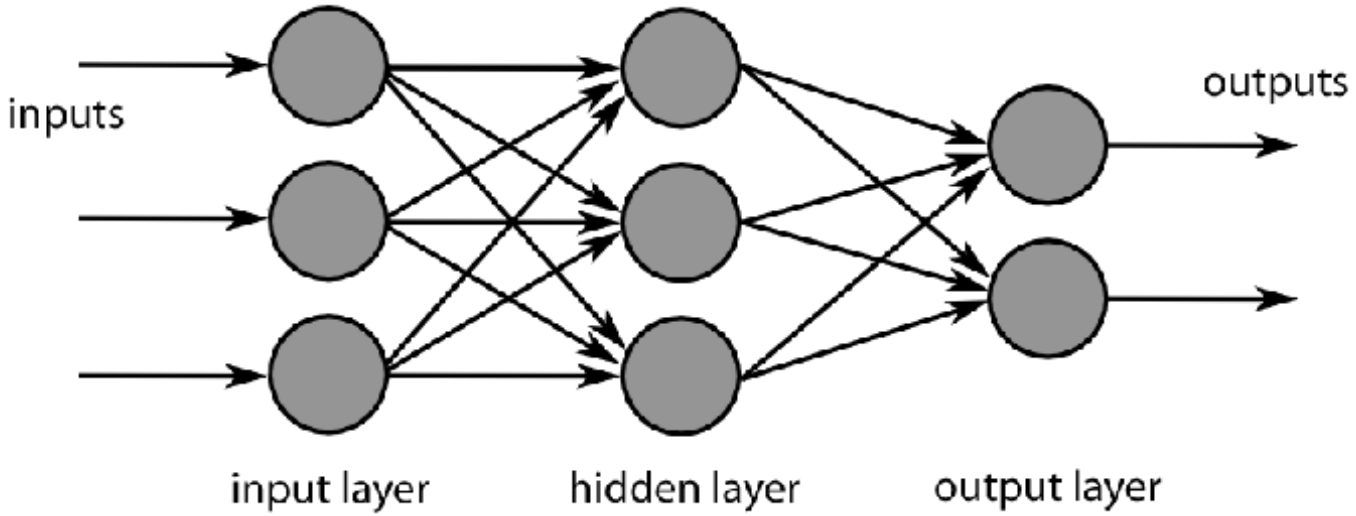
**Algoritmo Back Propagation**



# Redes Neuronales Feed-Forward

- Estructura de las redes feed-forward
- Funciones de activación
- El algoritmo de aprendizaje back-propagation
- Factores del aprendizaje
- Ejemplo

# Estructura feed-forward multicapa



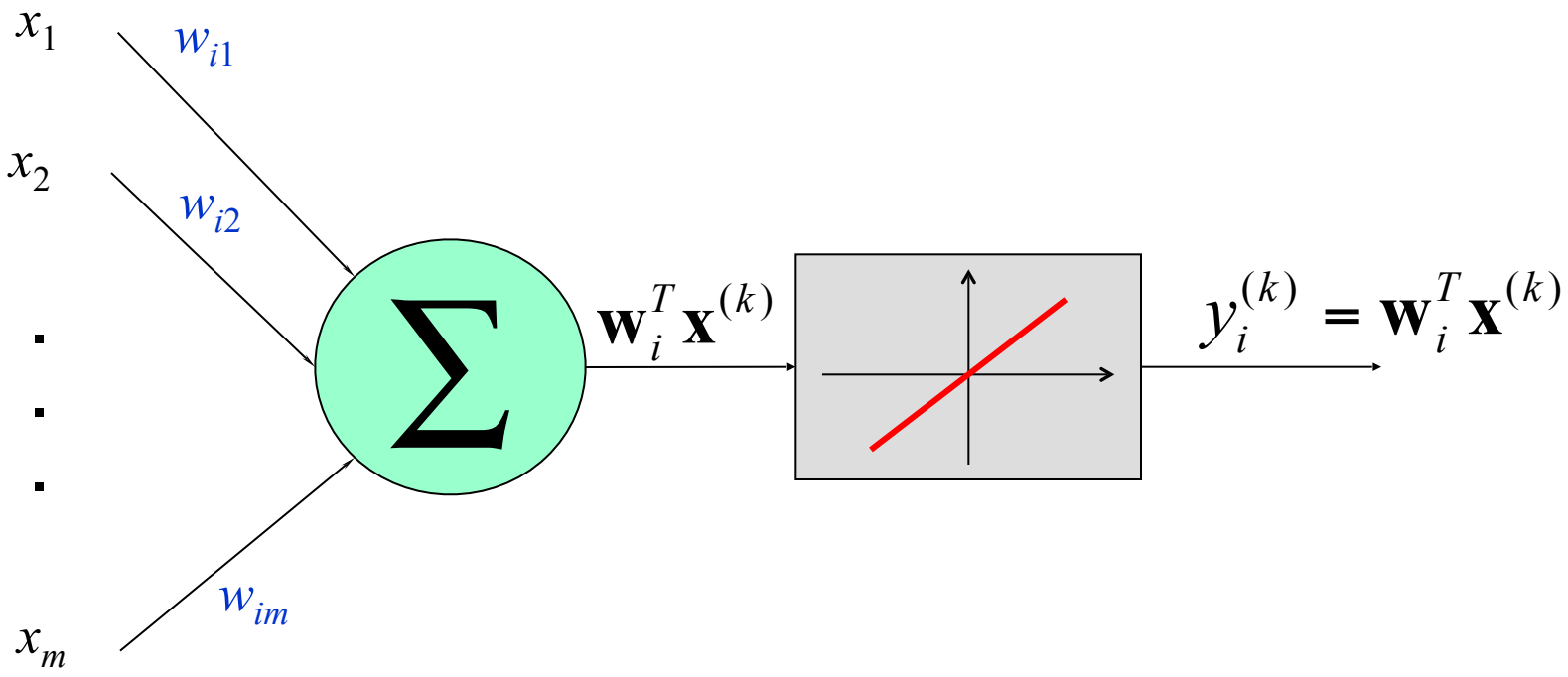

RNA's donde las conexiones entre las unidades NO forman un ciclo.

En esta red, la información se mueve en una única dirección: adelante.

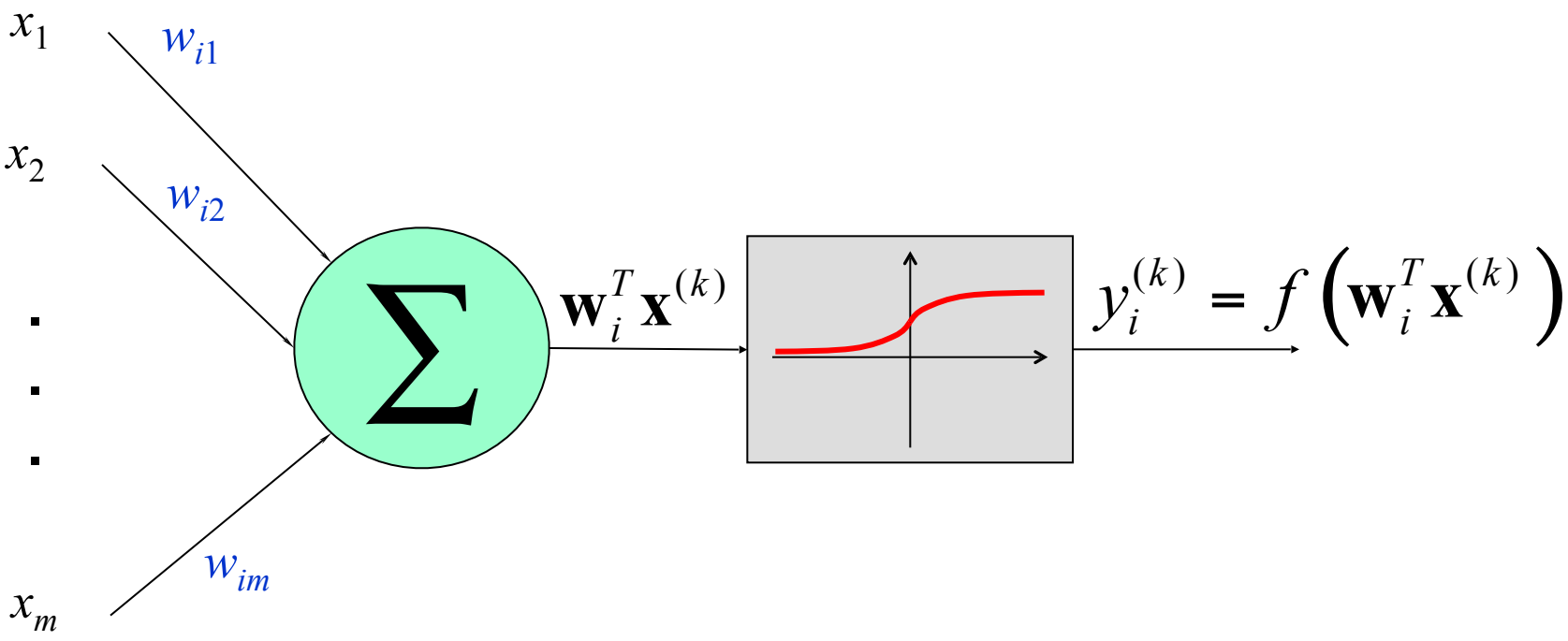
Desde los nodos de entrada, a través de los nodos ocultos hacia los nodos de salida.

# Funciones de Activación

# Lineal - purelin

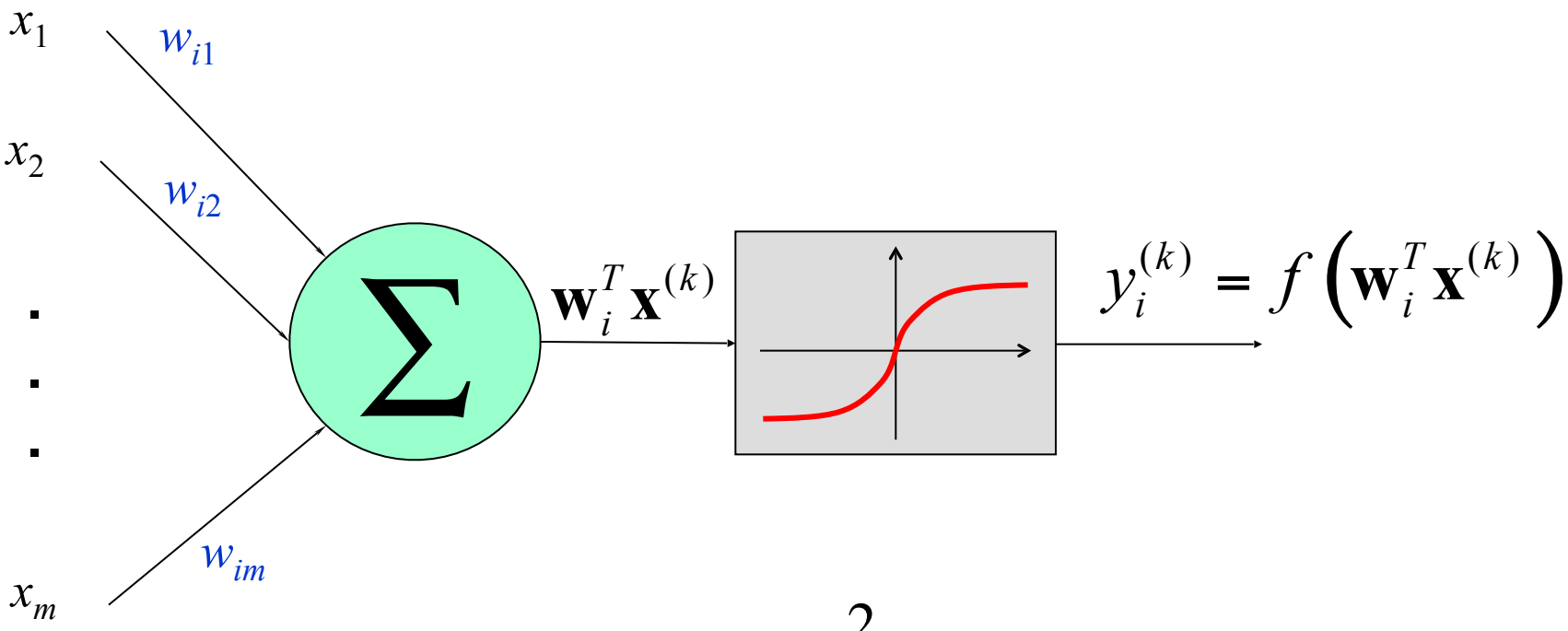


# Sigmoide unipolar - logsig




$$f(neta_i) = \frac{1}{1 + e^{-\lambda neta_i}}$$

# Sigmoide bipolar - tansig




$$f(neta_i) = \frac{2}{1 + e^{-\lambda neta_i}} - 1$$

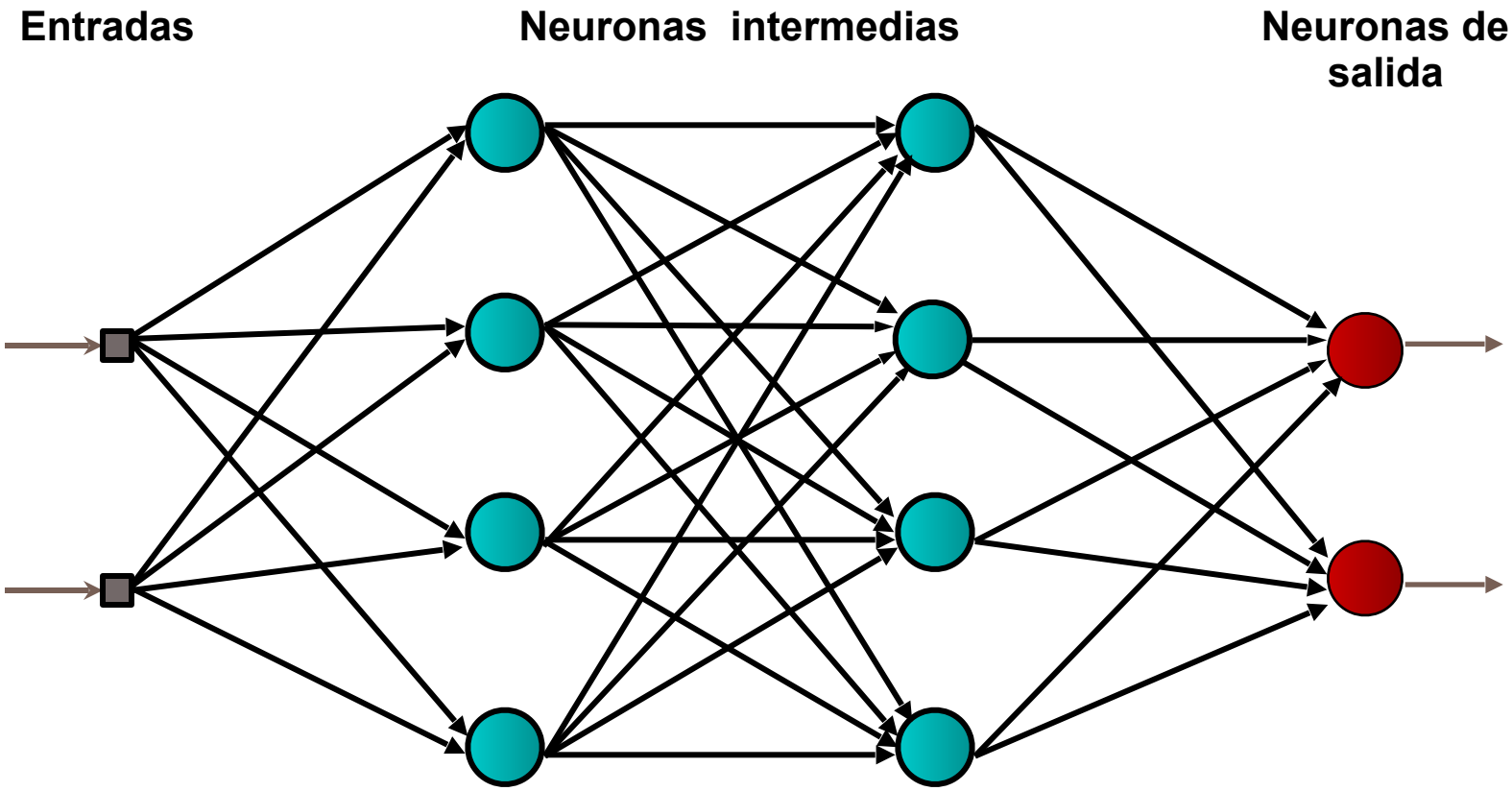
# El algoritmo back-propagation
## (Algoritmo de aprendizaje de propagación reversa)

# El algoritmo back-propagation

- Un procedimiento de aprendizaje que permite entrenar a las redes feedforward multicapa.

- En teoría se puede capturar "cualquier" mapeo de entrada-salida

# Red Neuronal Multicapa

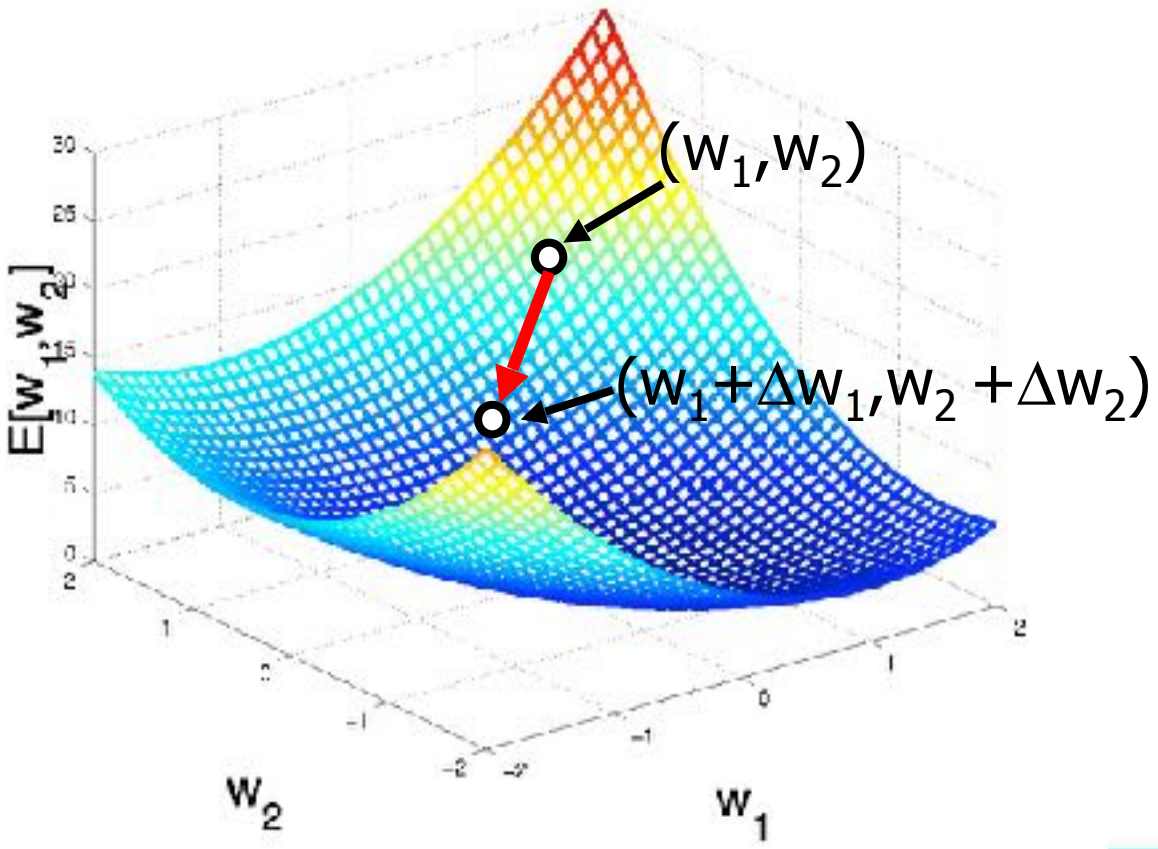


# Descenso por el gradiente

Back-propagation con uso de la técnica de descenso por el gradiente.

Para minimizar *E*,

$$\Delta \mathbf{w} = -\eta \nabla E$$



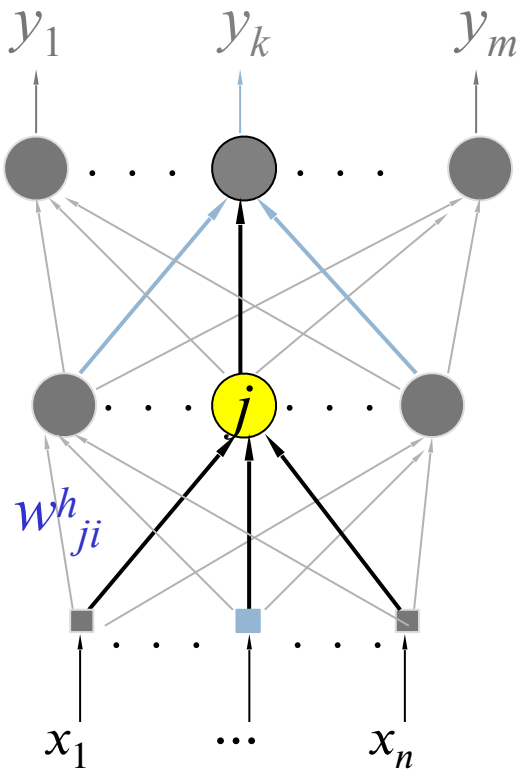
# Algoritmo back-propagation

1. Aplicar el p-ésimo vector de patrones de entrada

$$p = [x_1, x_2, \cdots, x_n]$$

2. Calcular los valores netos procedentes de las entradas para las unidades de la capa oculta. La entrada neta de la j-ésima unidad oculta es,

$$neta_{pj}^{\ h} = \sum_{i=1}^{n} w^{h}_{\ ji} x_{pi}$$



3. Calcular las salidas de las neuronas de la capa oculta.

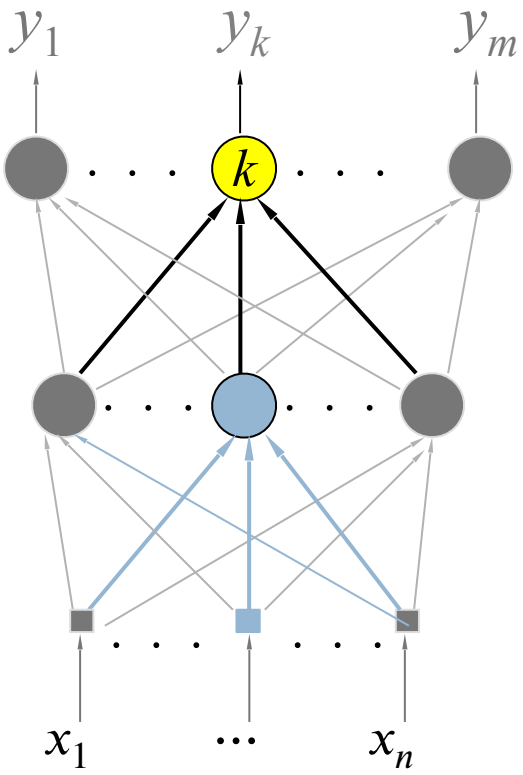
$$a_{pj} = f_j^h(neta_{pj}^h)$$

4. Calcular los valores netos de las entradas de la capa de salida

$$neta_{pk}^o = \sum_{j=1}^{m} w^o{}_{kj} a_{pj}$$

5. Calcular las salidas

$$y_{pk} = f_k^o(neta_{pk}^o)$$

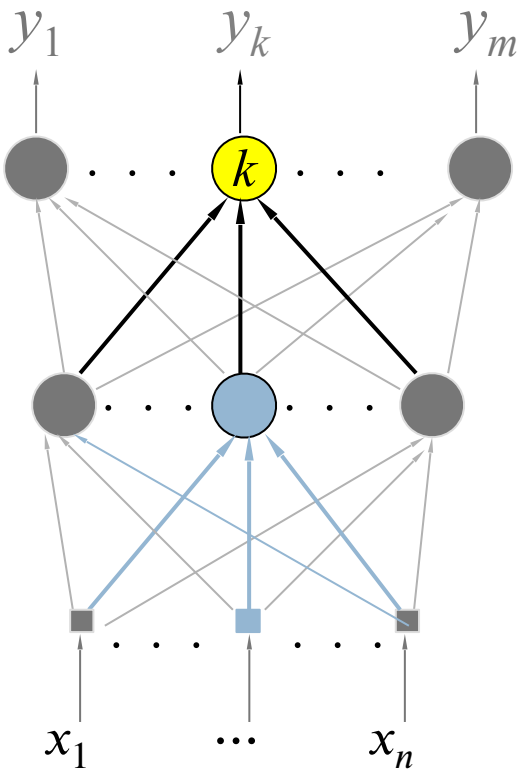


6. Calcular los términos de error para las unidades de salida.

$$\varepsilon_{pk} = (t_{pk} - y_{pk})$$

El error se minimiza por la regla delta generalizada aplicada a la suma de los cuadrados del error de todas las unidades de salida.

$$E_p = \frac{1}{2} \sum_{k=1}^{m} \varepsilon^2_{pk}$$



Calcular el gradiente con respecto a los pesos.

$$E_p = \frac{1}{2}\sum_{k=1}^{m} \varepsilon^2{}_{pk} = \frac{1}{2}\sum_{k=1}^{m} \left(t_{pk} - y_{pk}\right)^2$$

$$\frac{\partial E_p}{\partial w^o{}_{kj}} = \frac{\partial}{\partial w^o{}_{kj}} \frac{1}{2}\sum_{k=1}^{m} \varepsilon^2{}_{pk} = \frac{1}{2}\sum_{k=1}^{m} \frac{\partial \varepsilon^2{}_{pk}}{\partial w^o{}_{kj}}$$

Por lo tanto,

$$\frac{\partial \varepsilon^2{}_{pk}}{\partial w^o{}_{kj}} = \frac{\partial \varepsilon^2{}_{pk}}{\partial y_{pk}} \frac{\partial y_{pk}}{\partial w^o{}_{kj}}$$

Calculando el primer término

$$\frac{\partial \varepsilon^2{}_{pk}}{\partial y_{pk}} = \frac{\partial}{\partial y_{pk}}\left(t_{pk} - y_{pk}\right)^2 = -2\left(t_{pk} - y_{pk}\right) f_k{}^{o'}\left(neta_{pk}{}^{o}\right)$$

Calculando el segundo término

$$\frac{\partial y_{pk}}{\partial w^o{}_{kj}} = \frac{\partial\left(\sum_{j=1}^{m} w^o{}_{kj} a_{pj}\right)}{\partial w^o{}_{kj}} = a_{pj}$$

Reuniendo los resultados de los dos términos,

$$\frac{\partial \varepsilon^2{}_{pk}}{\partial w^o{}_{kj}} = -2\left(t_{pk} - y_{pk}\right) f_k{}^{o'}(neta_{pk}{}^o) a_{pj}$$

La actualización de pesos se considera proporcional a (1/2) del gradiente negativo,

$$\Delta w^o{}_{kj} = \eta \delta_{pk}{}^o a_{pj}$$

Con,

$$\delta_{pk}{}^o = \left(t_{pk} - y_{pk}\right) f_k{}^{o'}(neta_{pk}{}^o)$$

$\eta$: Velocidad de aprendizaje.

7. Calcular los términos de error para las unidades ocultas

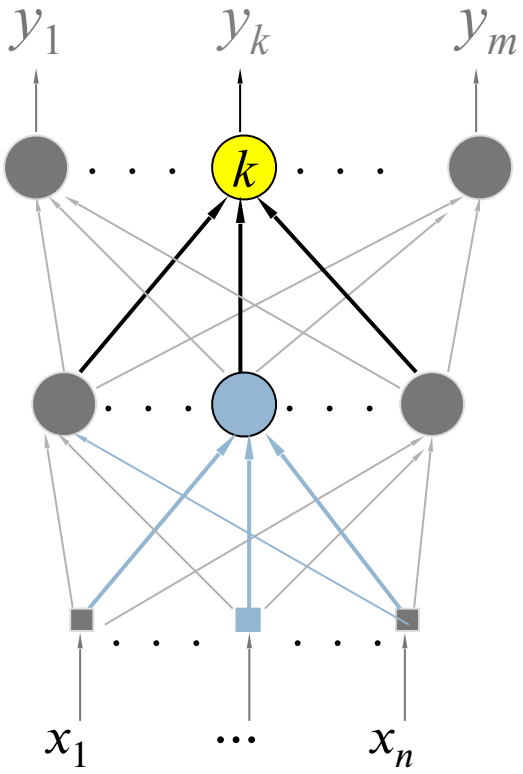
Problema: No conocemos la salida deseada.

Asume: El error debe estar relacionado con las salidas de la capa oculta.

$$E_p = \frac{1}{2}\sum_{k=1}^{m}\left(t_{pk} - y_{pk}\right)^2$$

$$E_p = \frac{1}{2}\sum_{k=1}^{m}\left(t_{pk} - f_k^{\,o}\left(neta_{pk}^{\,o}\right)\right)^2$$

$$E_p = \frac{1}{2}\sum_{k=1}^{m}\left(t_{pk} - f_k^{\,o}\left(\sum_{j=1}^{m} w^o{}_{kj} a_{pj}\right)\right)^2$$



Notar que $a_{pj}$ depende de los pesos de las capas ocultas, por lo tanto podemos calcular el gradiente del error respecto a los pesos de las capas ocultas.

$$\frac{\partial E_p}{\partial w^h_{ji}} = \frac{1}{2} \sum_k \frac{\partial}{\partial w^h_{ji}} \left( t_{pk} - y_{pk} \right)^2$$

$$\frac{\partial E_p}{\partial w^h_{ji}} = - \sum_k \left( t_{pk} - y_{pk} \right) \frac{\partial y_{pk}}{\partial (neta_{pk}{}^o)} \frac{\partial (neta_{pk}{}^o)}{\partial a_{pj}} \frac{\partial a_{pj}}{\partial (neta_{pj}{}^h)} \frac{\partial (neta_{pk}{}^h)}{\partial w^h_{ji}}$$

Calculando cada término se obtiene,

$$\frac{\partial E_p}{\partial w^h_{ji}} = -\sum_k \left(t_{pk} - y_{pk}\right) f_k^{o'}(neta_{pk}^{o}) w^o_{kj} f_j^{h'}(neta_{pj}^{h}) x_{pi}$$

La actualización de pesos es proporcional al negativo de la anterior ecuación.

$$\Delta w^h_{ji} = \eta \delta_{pj}^{h} x_{pi}$$

Con,

$$\delta_{pj}^{h} = f_j^{h'}(neta_{pj}^{h}) \sum_k \delta_{pk}^{o} w^o_{kj}$$

8. Actualizar los pesos de la capa de salida

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta\delta_{pk}{}^{o}a_{pj}$$

9. Actualizar los pesos de la capa oculta

$$w^{h}_{ji}(t+1) = w^{h}_{ji}(t) + \eta\delta_{pj}{}^{h}x_{pi}$$

10. Calcular el error

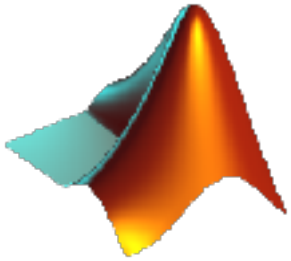
$$E_p = \frac{1}{2}\sum_{k=1}^{m} \varepsilon^{2}_{pk}$$

# Factores del Aprendizaje

# Factores del aprendizaje

- Pesos iniciales
- Velocidad de aprendizaje ($\eta$)
- Funciones de costo
- Datos de entrenamiento y generalización
- Numero de capas
- Numero de nodos ocultos

# Algoritmos para aumentar la velocidad de convergencia



- Velocidad de aprendizaje variable
- Gradiente conjugado
- Métodos de segundo orden
- Método Gauss – Newton
- Algoritmo de Levenberg-Marquardt

| Función de entrenamiento | Algoritmo |
|---|---|
| 'trainlm' | Levenberg-Marquardt |
| 'trainbr' | Regularización bayesiana |
| 'trainbfg' | BFGS Quasi-Newton |
| 'trainrp' | Retropropagación resiliente |
| 'trainscg' | Gradiente conjugado escalado |
| 'traincgb' | Gradiente conjugado con reinicios Powell / Beale |
| 'traincgf' | Gradiente conjugado de Fletcher-Powell |
| 'traincgp' | Gradiente conjugado de Polak-Ribiére |
| 'trainoss' | Secante de un paso |
| 'traingdx' | Descenso de gradiente de tasa de aprendizaje variable |
| 'traingdm' | Descenso de gradiente con impulso |
| 'traingd' | Descenso de gradiente |

# EJEMPLO

**Problema**

Estimar el PGC (Porcentaje de Grasa Corporal) de una persona en función de sus medidas anatómicas.

**Medidas anatómicas**

1. Años de edad
2. Peso en libras
3. Altura (pulgadas)
4. Circunferencia del cuello (cm)
5. Circunferencia del pecho (cm)
6. Circunferencia del abdomen (cm)
7. Circunferencia de la cadera (cm)
8. Circunferencia del muslo (cm)
9. Circunferencia de la rodilla (cm)
10. Circunferencia del tobillo (cm)
11. Circunferencia del bíceps (extendido) (cm)
12. Circunferencia del antebrazo (cm)
13. Circunferencia de muñeca (cm)

# EJEMPLO

**Datos de entrenamiento**

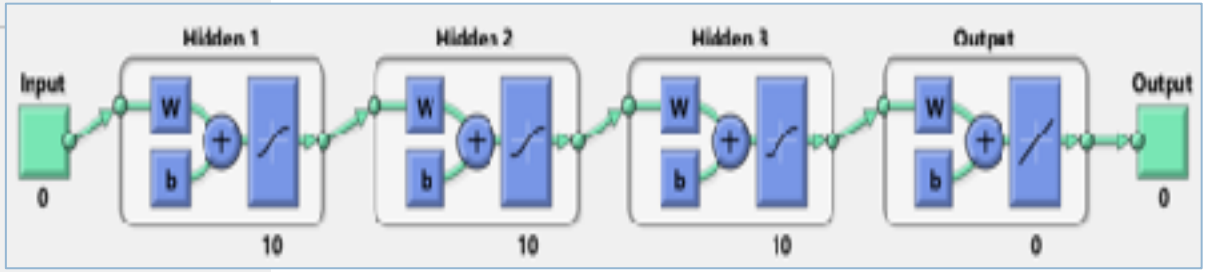
D_train: matriz de 13 características para 252 personas

D_target: vector con 252 medidas del PGC correctas

# EJEMPLO

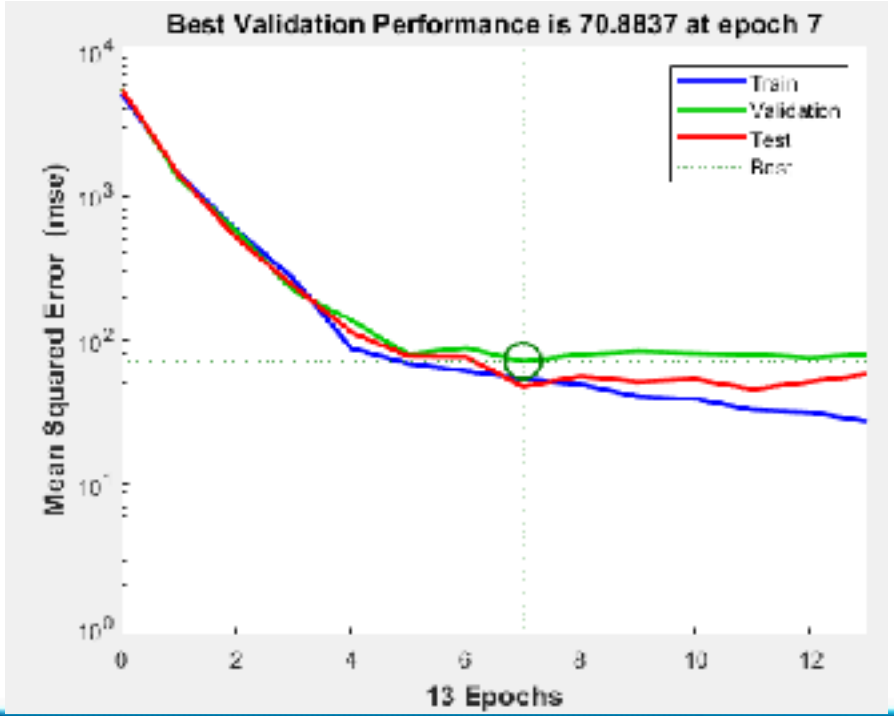

Herramienta de entrenamiento de RNA's en Matlab

El grafico indica la cantidad de capas y neuronas por capa, proyectadas en nuestra solución.



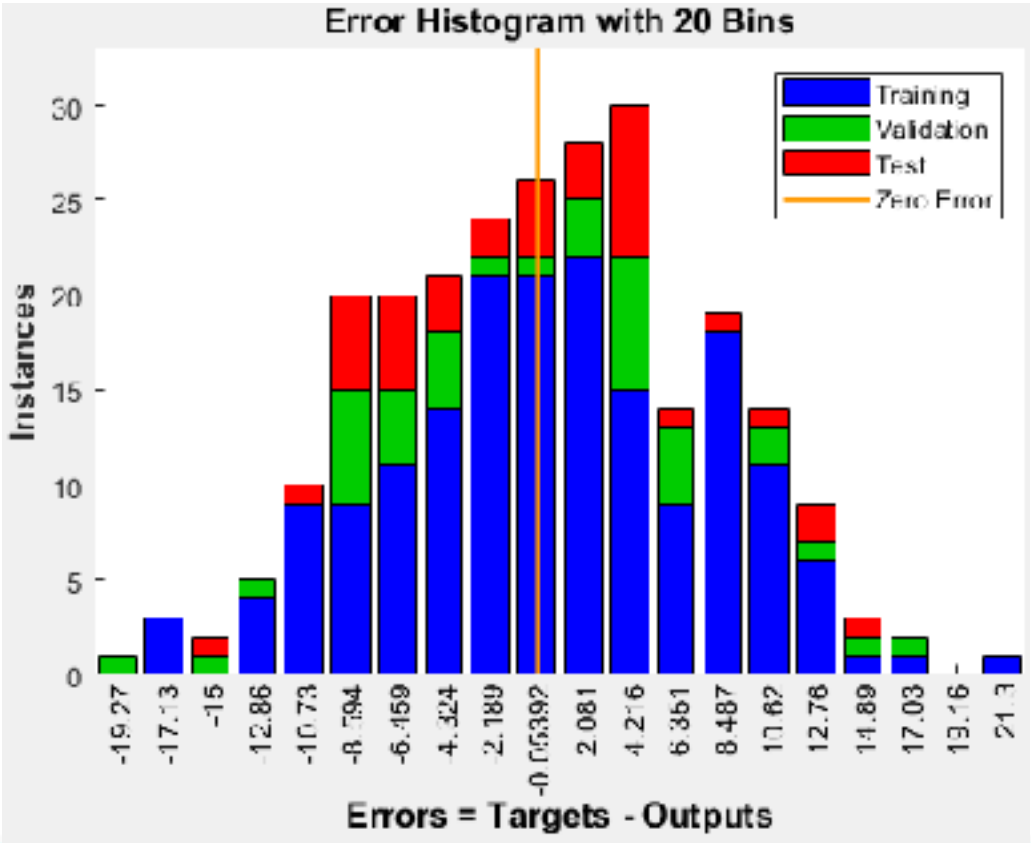
# EJEMPLO

El grafico de desempeño muestra el mejor valor de MSE (error cuadrático medio) de los datos de validación. En este caso ocurrido en la época 7 dentro de las 13 épocas del ejemplo. Recordar que una época significa cada vez que se utiliza el vector de entrenamiento para re-ajuste de los pesos sinápticos.



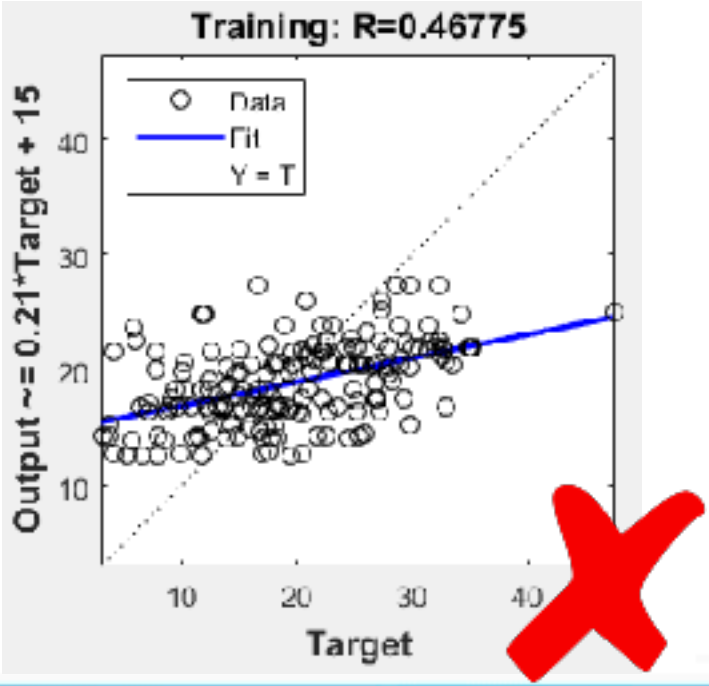
# EJEMPLO

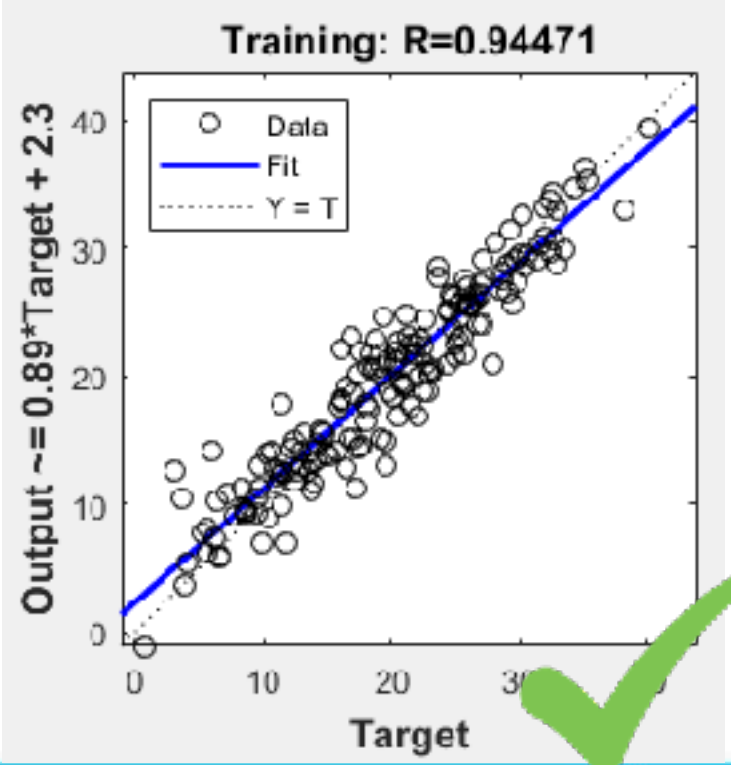
El histograma del error muestra la distribución de los tamaños del error, el propósito es que la mayor concentración este cerca de cero. Este grafico discrimina datos de entrenamiento, validación y de prueba.



# EJEMPLO

El grafico de regresión muestra el grado de ajuste que ha tenido la red a los datos de entrenamiento, validación y prueba. El ajuste perfecto o ideal se encuentra en línea gris punteada, la cual intersecta estrechamente las esquinas inferior izquierda y superior derecha del gráfico, mientras que el ajuste real de la RNA en color azul.



# ¿ Preguntas ?