Algoritmo de retropropagación

Matias F. Gerard

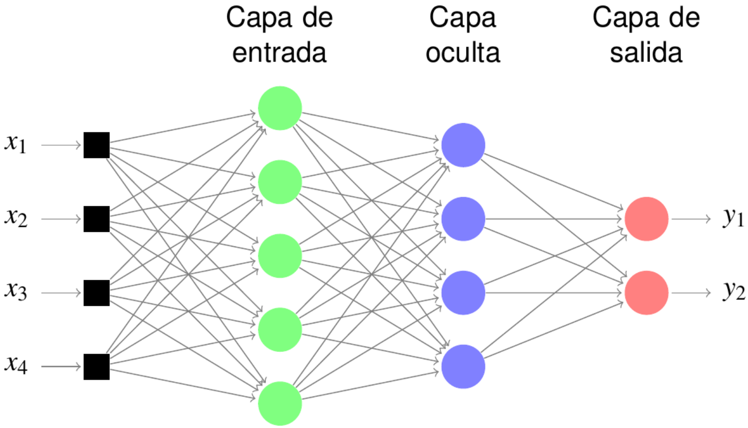
s ı n c (*i*)

Research Institute for Signals,
Systems and Computational Intelligence

Departament of Informatics
Faculty of Engineering and Water Sciences (FICH)
Universidad Nacional del Litoral (UNL)
National Scientific and Technical Research Council (CONICET)

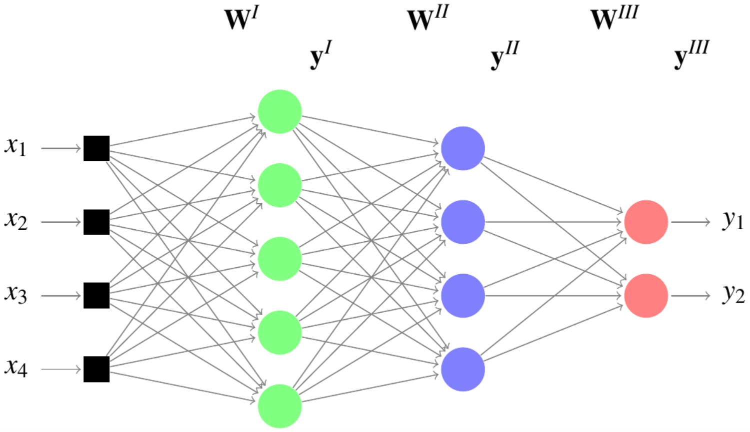
Perceptrón multicapa (MLP)

Arquitectura



Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Perceptrón multicapa (MLP)
Arquitectura



Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Perceptrón multicapa (MLP)
Algoritmo Backpropagation

1. Inicialización aleatoria.
2. Propagación hacia adelante.
3. Propagación hacia atrás.
4. Adaptación de los pesos.
5. Volver a 2 hasta convergencia o finalización.

Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

1. Inicialización aleatoria.
2. Propagación hacia adelante.
3. Propagación hacia atrás.
4. Adaptación de los pesos.
5. Volver a 2 hasta convergencia o finalización.

Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

- Capa *I*

$$v_j^I = \left\langle \mathbf{w}_j^I, \mathbf{x} \right\rangle = \sum_{i=0}^{N} w_{ji}^I x_i \rightarrow \mathbf{v}^I = \mathbf{W}^I \mathbf{x}$$

$$\Rightarrow y_j^I = \phi(v_j^I) = \frac{2}{1 + e^{-b v_j^I}} - 1$$

- Capa *II*

$$v_j^{II} = \left\langle \mathbf{w}_j^{II}, \mathbf{y}^I \right\rangle \rightarrow y_j^{II} = \phi(v_j^{II})$$

- Capa *III*

$$v_j^{III} = \left\langle \mathbf{w}_j^{III}, \mathbf{y}^{II} \right\rangle \rightarrow y_j^{III} = \phi(v_j^{III}) = y_j$$

Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

sinc(*i*)

Criterio del error: *Suma del error cuadrático instantáneo*

$$\xi(n) = \frac{1}{2}\sum_{j=1}^{M} e^2(n)$$

Actualización de los pesos

$$\Delta w_{ji} = -\mu \frac{\partial \xi(n)}{\partial w_{ji}(n)}$$

$$\frac{\partial \xi(n)}{\partial w_{ji}(n)} = \frac{\partial \xi(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

$$\boxed{\Delta w_{ji} = -\mu \frac{\partial \xi(n)}{\partial w_{ji}(n)}}$$

$$\frac{\partial \xi(n)}{\partial w_{ji}(n)} = \frac{\partial \xi(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = \frac{\partial \sum_{i=0}^{N} w_{ji}(n) y_i(n)}{\partial w_{ji}(n)} = y_i(n)$$

Gradiente de error local instantáneo: $\delta_j = \frac{\partial \xi(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)}$

$$\boxed{\Delta w_{ji} = \mu \; \delta_j \; y_i(n)}$$

Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

sinc(i)

Derivada de la función de activación

$$\delta_j = \frac{\partial \xi(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \rightarrow \frac{\partial y_j(n)}{\partial v_j(n)} = \frac{\partial \left\{ \frac{2}{1 + e^{-v_j(n)}} - 1 \right\}}{\partial v_j(n)}$$

$$= 2 \frac{e^{-v_j(n)}}{(1 + e^{-v_j(n)})^2}$$

$$= 2 \frac{1}{1 + e^{-v_j(n)}} \frac{e^{-v_j(n)}}{1 + e^{-v_j(n)}}$$

$$= 2 \frac{1}{1 + e^{-v_j(n)}} \frac{\overbrace{-1 + 1}^{=0} + e^{-v_j(n)}}{1 + e^{-v_j(n)}}$$

$$= 2 \frac{1}{1 + e^{-v_j(n)}} \left(\frac{-1}{1 + e^{-v_j(n)}} \frac{1 + e^{-v_j(n)}}{1 + e^{-v_j(n)}} \right)$$

S1

Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Derivada de la función de activación

$$\boxed{y_j(n) = \frac{2}{1+e^{-bv_j^I}} - 1 \Rightarrow \frac{y_j(n)+1}{2} = \frac{1}{1+e^{-bv_j^I}}}$$

$$\begin{aligned}
\frac{\partial y_j(n)}{\partial v_j(n)} &= 2\frac{1}{1+e^{-v_j(n)}}\left(1-\frac{1}{1+e^{-v_j(n)}}\right)\\
&= 2\frac{y_j(n)+1}{2}\left(1-\frac{y_j(n)+1}{2}\right)\\
&= \left(y_j(n)+1\right)\left(1-\frac{y_j(n)+1}{2}\right)\\
&= \left(y_j(n)+1\right)\left(\frac{2-y_j(n)-1}{2}\right)\\
&= \frac{1}{2}\left(1+y_j(n)\right)\left(1-y_j(n)\right)
\end{aligned}$$

Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Gradiente de error local instantáneo

$$\delta_j = \frac{\partial \xi(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)}$$

$$\frac{\partial y_j(n)}{\partial v_j(n)} = \frac{1}{2}\Big(1 + y_j(n)\Big)\Big(1 - y_j(n)\Big)$$

$$\Rightarrow \delta_j = \frac{\partial \xi(n)}{\partial y_j(n)} \frac{1}{2}\Big(1 + y_j(n)\Big)\Big(1 - y_j(n)\Big)$$

Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Retropropagación en la capa III (salida)

$$\boxed{\Delta w_{ji}^{III} = \mu\, \delta_j^{III}\; y_i^{II}(n)}$$

$$\delta_j^{III}(n) = -\frac{\partial \xi(n)}{\partial y_j^{III}(n)} \cdot \frac{1}{2}\left(1 + y_j^{III}(n)\right)\left(1 - y_j^{III}(n)\right)$$

$$\delta_j^{III}(n) = -\frac{\partial \xi(n)}{\partial e_j(n)}\, \frac{\partial e_j(n)}{\partial y_j^{III}(n)} \cdot \frac{1}{2}\left(1 + y_j^{III}(n)\right)\left(1 - y_j^{III}(n)\right)$$

$$\delta_j^{III}(n) = -\frac{\partial\left\{\frac{1}{2}\sum_j e_j^2(n)\right\}}{\partial e_j(n)} \cdot \frac{\partial\left\{d_j^{III}(n) - y_j^{III}(n)\right\}}{\partial y_j^{III}(n)} \cdot \frac{1}{2}\left(1 + y_j^{III}(n)\right)\left(1 - y_j^{III}(n)\right)$$

$$\delta_j^{III}(n) = \frac{1}{2} e_j(n)\left(1 + y_j^{III}(n)\right)\left(1 - y_j^{III}(n)\right)\star$$

$$\boxed{\Delta w_{ji}^{III}(n) = \mu\; e_j(n)\left(1 + y_j^{III}(n)\right)\left(1 - y_j^{III}(n)\right)\, y_j^{II}(n)}$$

Retropropagación en la capa II (oculta)

$$\boxed{\Delta w_{ji}^{II} = \mu\, \delta_j^{II}\, y_i^{I}(n)}$$

$$\delta_j^{II}(n) = -\frac{\partial \xi(n)}{\partial y_j^{II}(n)} \cdot \frac{1}{2}\left(1 + y_j^{II}(n)\right)\left(1 - y_j^{II}(n)\right)$$

$$\delta_j^{II}(n) = -\frac{\partial \left\{\frac{1}{2}\sum_k e_k^2(n)\right\}}{\partial y_j^{II}(n)} \cdot \frac{1}{2}\left(1 + y_j^{II}(n)\right)\left(1 - y_j^{II}(n)\right)$$

$$\delta_j^{II}(n) = -\frac{1}{2}\sum_k \frac{\partial e_k^2(n)}{\partial y_j^{II}(n)} \cdot \frac{1}{2}\left(1 + y_j^{II}(n)\right)\left(1 - y_j^{II}(n)\right)$$

$$\delta_j^{II}(n) = -\sum_k e_k(n)\frac{\partial e_k(n)}{\partial y_j^{II}(n)} \cdot \frac{1}{2}\left(1 + y_j^{II}(n)\right)\left(1 - y_j^{II}(n)\right)$$

Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Retropropagación en la capa II (oculta)

$$\boxed{\Delta w_{ji}^{II} = \mu \, \delta_j^{II} \, y_i^{I}(n)}$$

$$\delta_j^{II}(n) = -\sum_k e_k(n) \frac{\partial e_k(n)}{\partial y_j^{II}(n)} \cdot \frac{1}{2}\left(1 + y_j^{II}(n)\right)\left(1 - y_j^{II}(n)\right)$$

$$\delta_j^{II}(n) = -\sum_k e_k(n) \frac{\partial e_k(n)}{\partial y_j^{III}(n)} \frac{\partial y_k^{III}(n)}{\partial v_k^{III}(n)} \frac{\partial v_k^{III}(n)}{\partial y_j^{II}(n)} \cdot \frac{1}{2}\left(1 + y_j^{II}(n)\right)\left(1 - y_j^{II}(n)\right)$$

$$\delta_j^{II}(n) = -\sum_k e_k(n) \frac{\partial \left\{d_k^{III}(n) - y_k^{III}(n)\right\}}{\partial y_k^{III}(n)} \cdot \frac{1}{2}\left(1 + y_k^{III}(n)\right)\left(1 - y_k^{III}(n)\right) \cdot \frac{\partial \left\{\sum_j w_{kj}^{III}(n) y_j^{II}(n)\right\}}{\partial y_j^{II}(n)} \cdot \frac{1}{2}\left(1 + y_j^{II}(n)\right)\left(1 - y_j^{II}(n)\right)$$

$$\delta_j^{II}(n) = -\sum_k e_k(n)\,(-1) \cdot \frac{1}{2}\left(1 + y_k^{III}(n)\right)\left(1 - y_k^{III}(n)\right) \cdot w_{kj}^{III}(n) \cdot \frac{1}{2}\left(1 + y_j^{II}(n)\right)\left(1 - y_j^{II}(n)\right)$$

S1

Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Retropropagación en la capa II (oculta)

$$\boxed{\Delta w_{ji}^{II} = \mu\, \delta_j^{II}\, y_i^{I}(n)}$$

De la capa III★, sabemos que:

$$\delta_j^{III}(n) = \frac{1}{2} e_j(n) \left(1 + y_j^{III}(n)\right)\left(1 - y_j^{III}(n)\right)$$

También sabemos que:

$$\delta_j^{II}(n) = \sum_k e_k(n) \cdot \frac{1}{2}\left(1 + y_k^{III}(n)\right)\left(1 - y_k^{III}(n)\right) \cdot w_{kj}^{III}(n) \cdot \frac{1}{2}\left(1 + y_j^{II}(n)\right)\left(1 - y_j^{II}(n)\right)$$

Reemplazando:

$$\delta_j^{II}(n) = \sum_k \delta_k^{III}(n) \cdot w_{kj}^{III}(n) \cdot \frac{1}{2}\left(1 + y_j^{II}(n)\right)\left(1 - y_j^{II}(n)\right)$$

Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Retropropagación en la capa II (oculta)

Dado que:

$$\boxed{\Delta w_{ji}^{II} = \mu\ \delta_j^{II}\ y_i^{I}(n)}$$

$$\delta_j^{II}(n) = \sum_k \delta_k^{III}(n) \cdot w_{kj}^{III}(n) \cdot \frac{1}{2}\left(1 + y_j^{II}(n)\right)\left(1 - y_j^{II}(n)\right)$$

$$\delta_j^{III}(n) = \frac{1}{2} e_j(n)\left(1 + y_j^{III}(n)\right)\left(1 - y_j^{III}(n)\right)$$

Se deduce:

$$\boxed{\Delta w_{ji}^{II} = \mu \left[\sum_k \delta_k^{III}(n) w_{kj}^{III}(n)\right]\left(1 + y_j^{II}(n)\right)\left(1 - y_j^{II}(n)\right) y_i^{I}(n)}$$

Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

sinc(i)

Generalización para la capa p

$$\Delta w_{ji}^{II} = \mu \left[\sum_k \delta_k^{III}(n) w_{kj}^{III}(n) \right] \left(1 + y_j^{III}(n)\right)\left(1 - y_j^{III}(n)\right) y_i^{I}(n)$$

$$\Downarrow$$

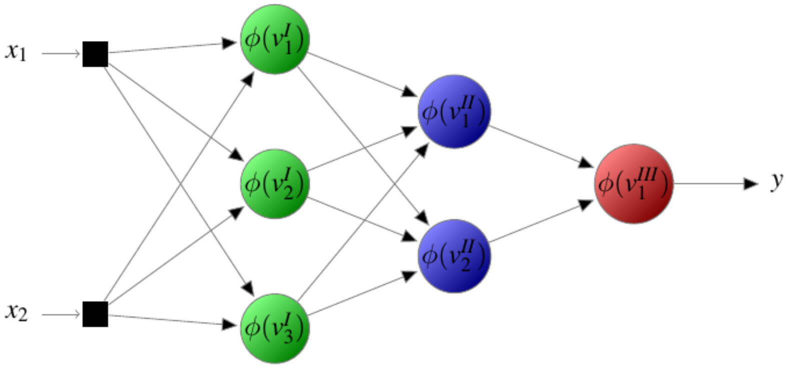
$$\boxed{\Delta w_{ji}^{p} = \mu \left\langle \delta^{p+1}, \mathbf{w}_j^{p+1} \right\rangle \left(1 + y_j^{p}(n)\right)\left(1 - y_j^{p}(n)\right) y_i^{p-1}(n)}$$

Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

1. Inicialización aleatoria.
2. Propagación hacia adelante.
3. Propagación hacia atrás.
4. Adaptación de los pesos.
5. Iteración: vuelve a 2 hasta convergencia o finalización.

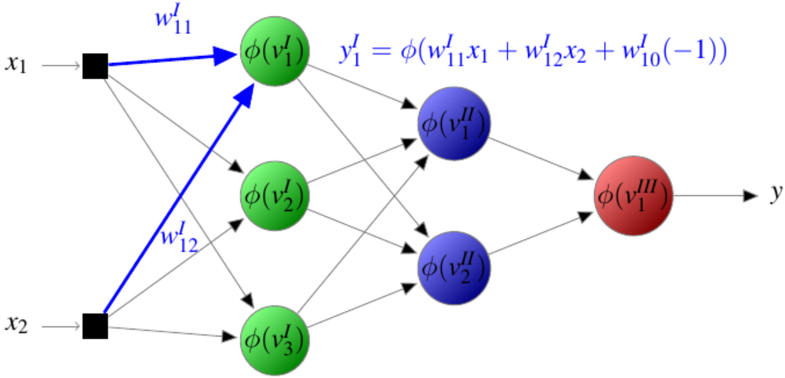
Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Propagación hacia adelante



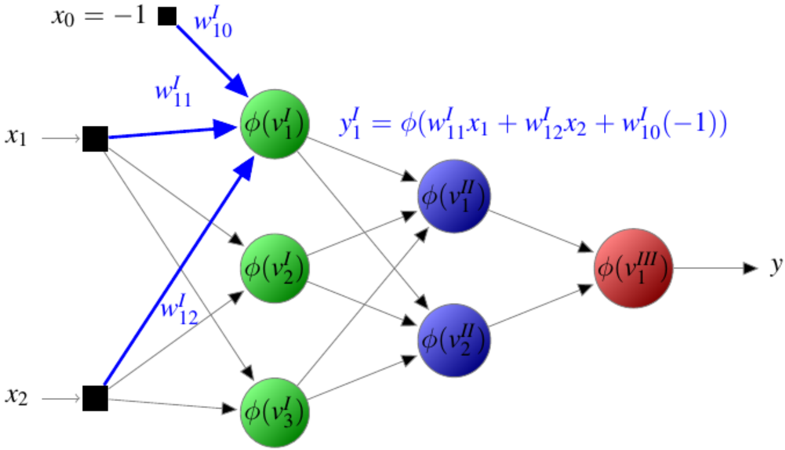
Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Propagación hacia adelante

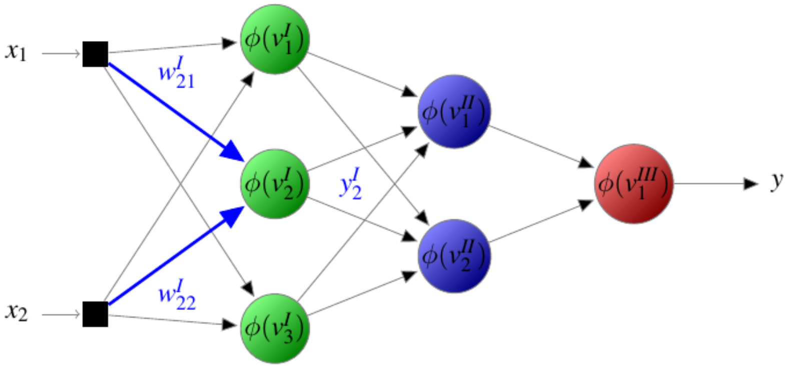


Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Propagación hacia adelante

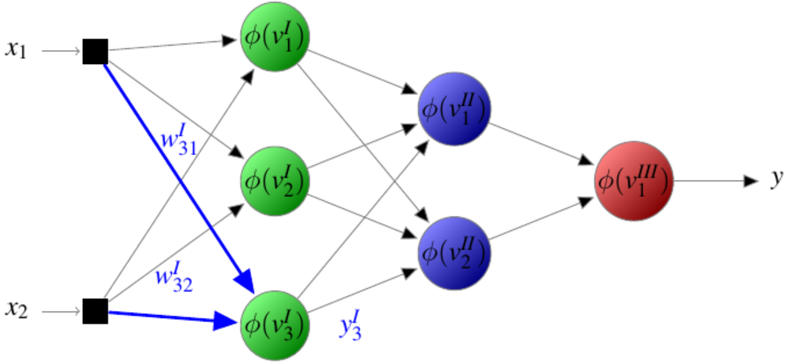


Propagación hacia adelante



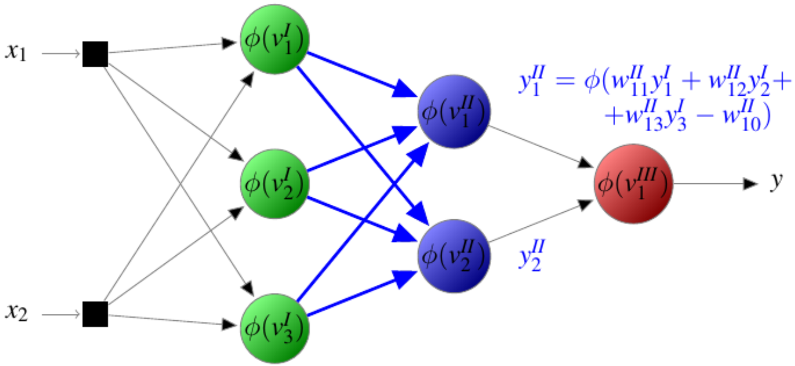
Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Propagación hacia adelante



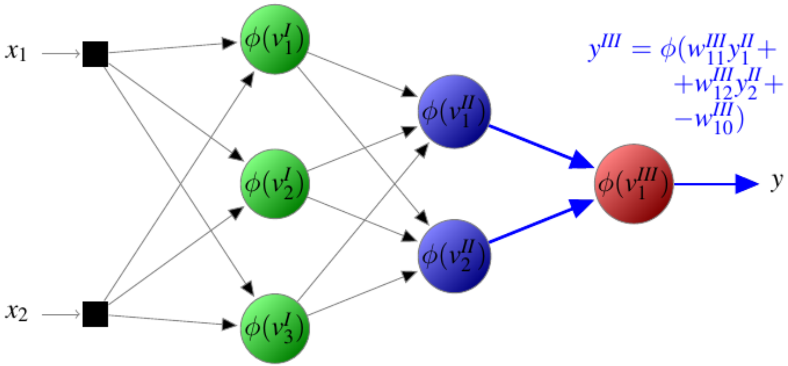
Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Propagación hacia adelante



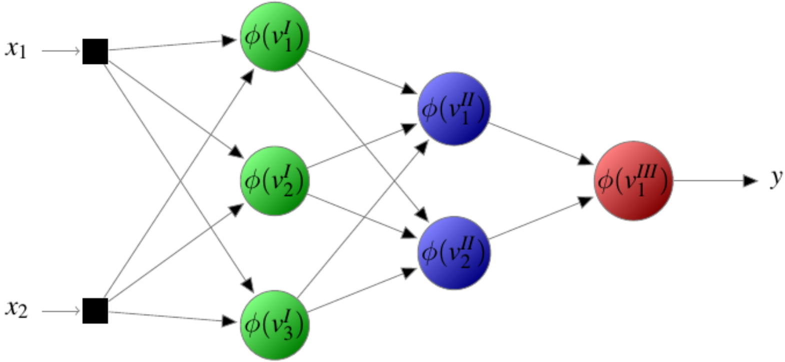
Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Propagación hacia adelante



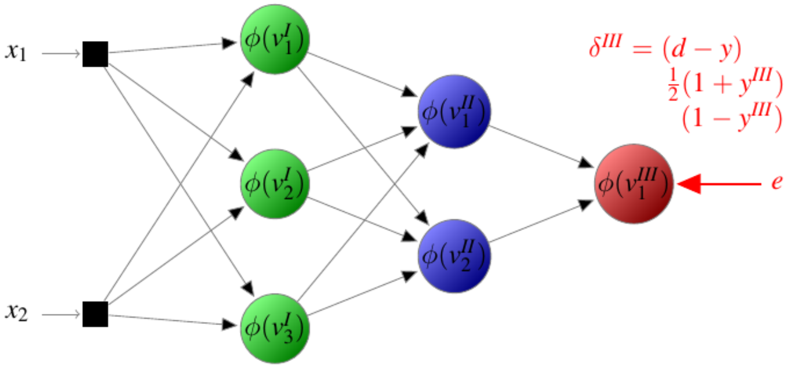
Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Propagación hacia atrás



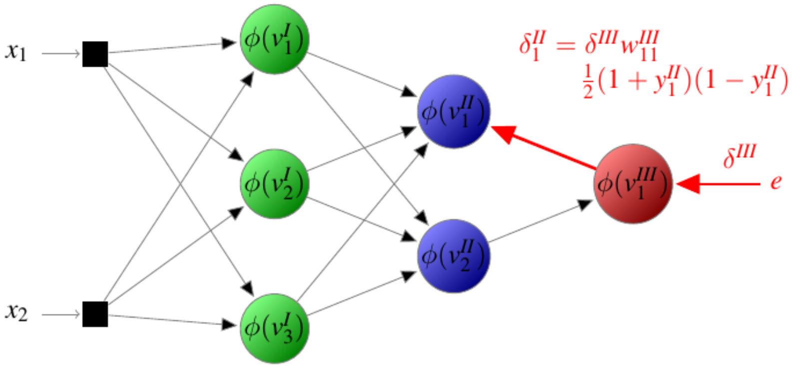
Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Propagación hacia atrás



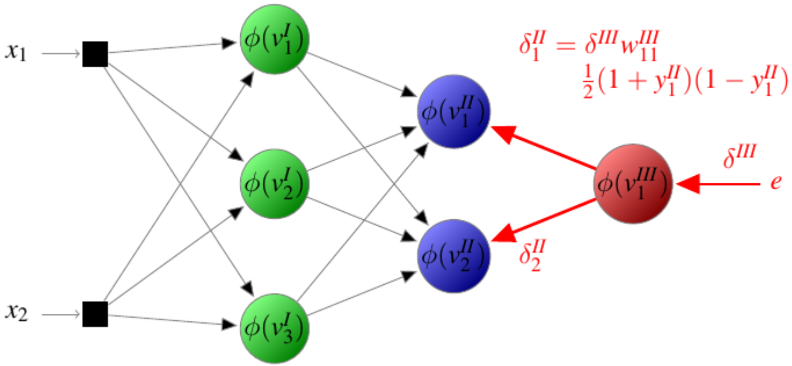
Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Propagación hacia atrás



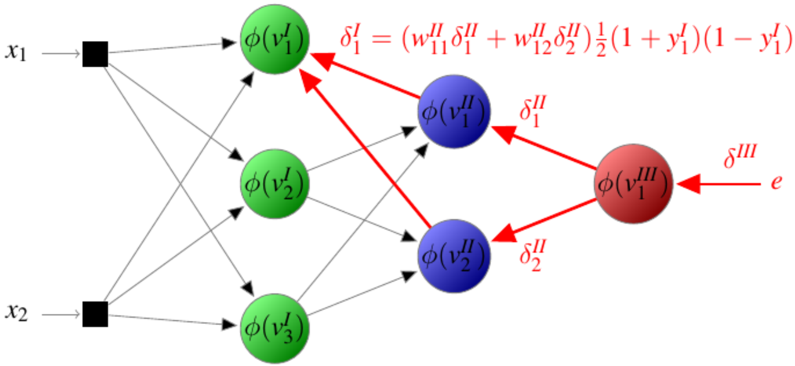
Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Propagación hacia atrás



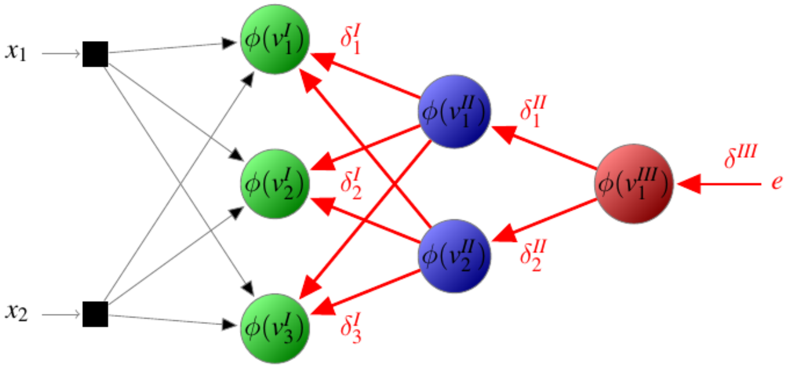
Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Propagación hacia atrás



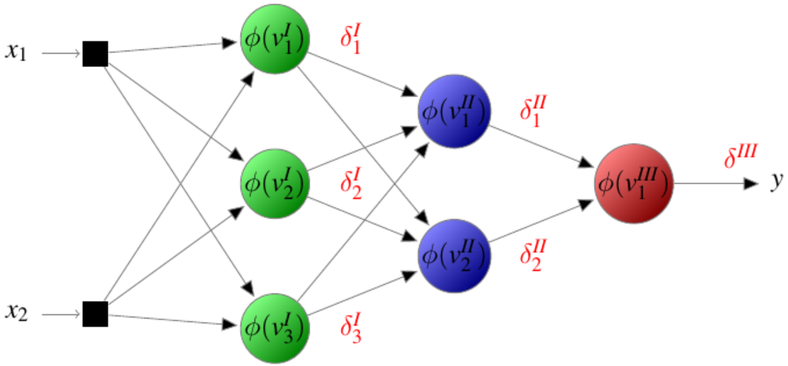
Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Propagación hacia atrás



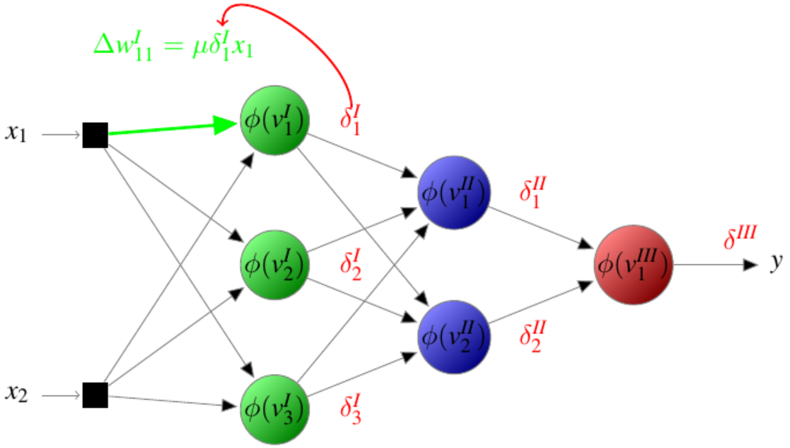
Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Ajuste de pesos



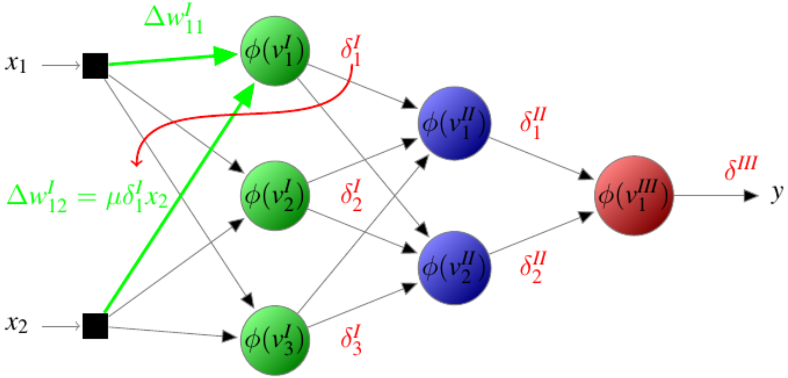
Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Ajuste de pesos



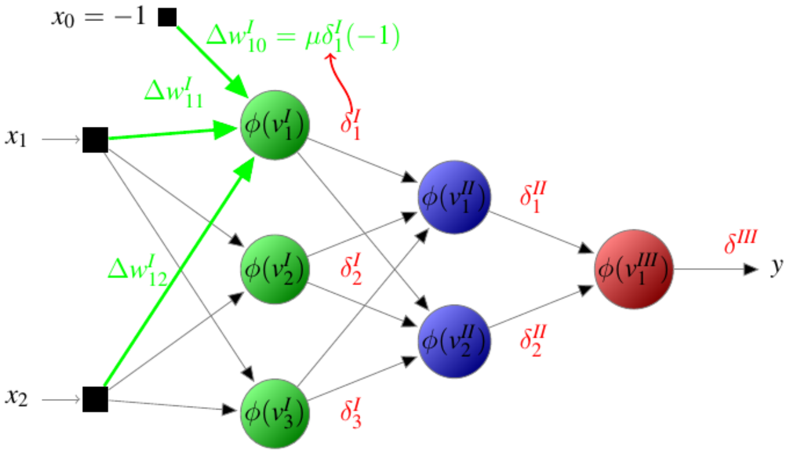
Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Ajuste de pesos



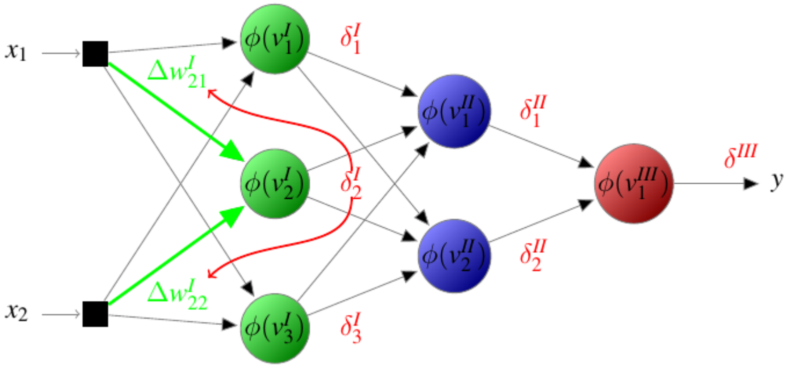
Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Ajuste de pesos



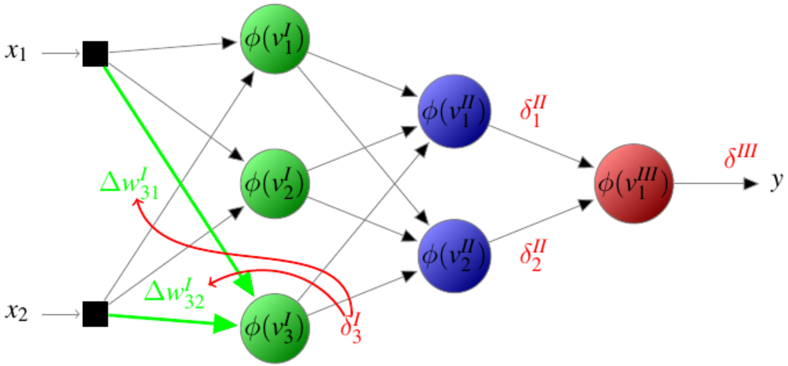
Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Ajuste de pesos



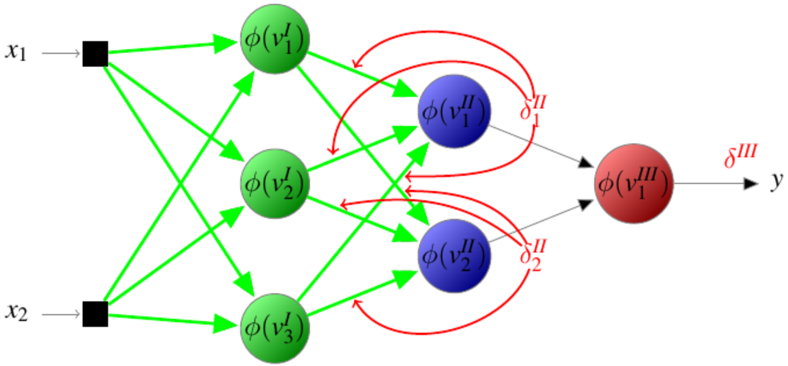
Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Ajuste de pesos



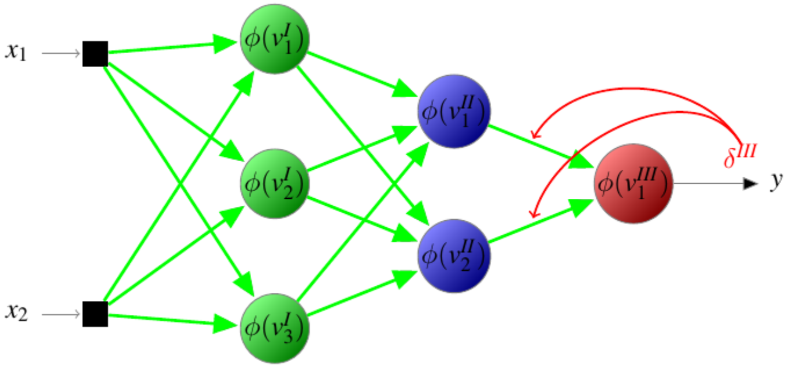
Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Ajuste de pesos



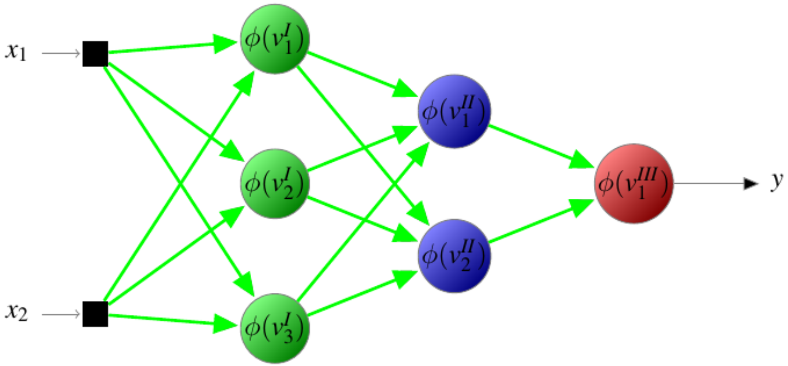
Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.

Ajuste de pesos



Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Ajuste de pesos



Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

Perceptrón multicapa (MLP)
Término de momento

$$\Delta w_{ji}(n) = -\eta \frac{\partial \xi(n)}{\partial w_{ji}(n)} + \alpha \Delta w_{ji}(n-1)$$

- $0.5 \leq \alpha \leq 1.0$.
- Permite utilizar tazas de aprendizaje más altas.
- Reduce las oscilaciones que pueden producirse durante el entrenamiento.
- Acelera la convergencia en zonas con poca pendiente.
- Favorece el suavizado de la curva de convergencia.
- Introduce un aumento el costo de almacenamiento, ya que deben almacenarse las actualizaciones de peso $\Delta w_i(n-1)$.

Extraído del material del curso "Inteligencia Computacional", 4to año FICH-UNL.

$$\boxed{\Delta w_{ji}(n) = -\eta \frac{\partial \xi(n)}{\partial w_{ji}(n)} + \alpha \Delta w_{ji}(n-1)}$$

- $0.5 \leq \alpha \leq 1.0$.
- Permite utilizar tazas de aprendizaje más altas.
- Reduce las oscilaciones que pueden producirse durante el entrenamiento.
- Acelera la convergencia en zonas con poca pendiente.
- Favorece el suavizado de la curva de convergencia.
- Introduce un aumento el costo de almacenamiento, ya que deben almacenarse las actualizaciones de peso $\Delta w_i(n-1)$.

Extraído del material del curso “Inteligencia Computacional”, 4to año FICH-UNL.