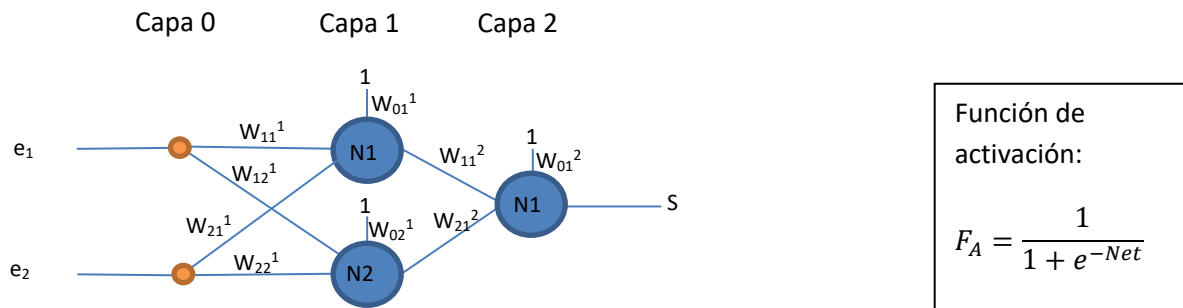


Práctica Inteligencia Artificial Curso 2022/2023. Convocatoria enero

Calcular una propagación y una retropropagación de la siguiente red de neuronas del tipo perceptrón multicapa, utilizando en algoritmo de retropropagación del gradiente.



Para esta práctica, todos los parámetros de entrenamiento son proporcionados en un Excel adjunto al cuestionario de la práctica, los cuales incluyen:

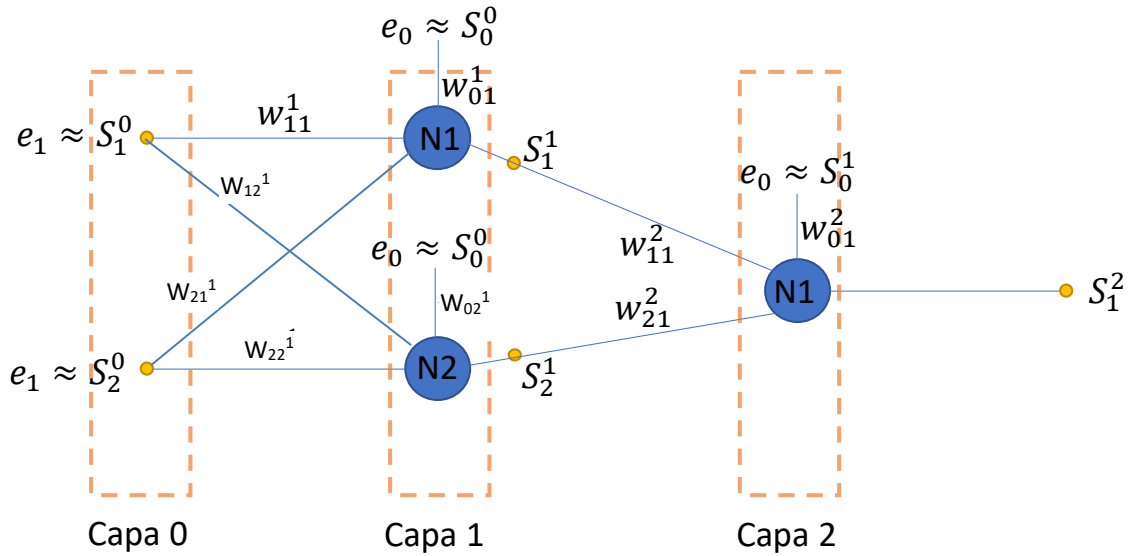
Entradas	PesosCapa_1	PesosCapa_2	SalidaDeseada	Alpha ()
E0	W_{01}^1	W_{01}^2	sd	α
E1	W_{11}^1	W_{11}^2		
E2	W_{21}^1	W_{21}^2		
	W_{02}^1			
	W_{12}^1			
	W_{22}^1			

Entregable: Contestar el cuestionario de Moodle sobre la práctica, donde se debe poner el valor de algunos de los parámetros resultantes del entrenamiento (10 puntos).

Fecha final de entrega: 30 de noviembre de 2022

Algoritmo de entrenamiento del perceptrón multicapa

Este es un ejemplo de la arquitectura de un perceptrón multicapa, con 1 capa de salida (con 1 neurona) y 1 capa oculta (con 2 neuronas), con la notación de todos sus componentes.



Para este ejemplo se ha utilizado la siguiente notación:

1. Propagación

La salida S_j^k de cada neurona j de la capa k es igual a:

$$S_j^k = F_A \left(NETA_j^k \right) = \frac{1}{1 + e^{-NETA_j^k}}$$

$$NETA_j^k = \sum_i \left(S_i^{k-1} \cdot w_{ij}^k \right)$$

Donde la i representa cada una de las neuronas de la capa $k-1$ cuya salida S_i^{k-1} es la entrada de la neurona j de la capa k , w_{ij}^k representa el peso de la conexión que va de la neurona i de la capa $k-1$ a la neurona j de la capa k , y F_A es la función de activación de la neurona, siendo el $NETA_j^k$ de la neurona j de la capa k el sumatorio de todos los pesos w_{ij}^k multiplicado por el valor de sus entradas S_i^{k-1} .

2. Retropropagación

El cálculo del incremento de los pesos en el algoritmo de retropropagación del gradiente w para una conexión entre la neurona i de la capa $k-1$ y la neurona j de la capa k se expresa mediante la siguiente función:

$$\Delta w_{ij}^k = \alpha \cdot S_i^{k-1} \cdot \delta_j^k$$

Donde α es el coeficiente de aprendizaje, S es la salida de la neurona i de la capa $k-1$ y δ es la función de retropropagación, que se define como sigue:

2.1 Para la capa de salida

δ de la neurona j de la capa k es igual a

$$\delta_j^k = (d_j - S_j^k) \cdot S_j^k \cdot (1 - S_j^k)$$

Donde d_j es la salida deseada del conjunto de entrenamiento para la entrada de la red y S_j^k es la salida de la propagación de la neurona j de la capa k .

2.2 Para las capas ocultas

δ de la neurona j de la capa k para las capas ocultas es igual a:

$$\delta_j^k = V_j^k \cdot S_j^k \cdot (1 - S_j^k)$$

Donde S_j^k es la salida de la propagación de la neurona j de la capa k y V_j^k , para cada neurona j de la capa k , es igual a:

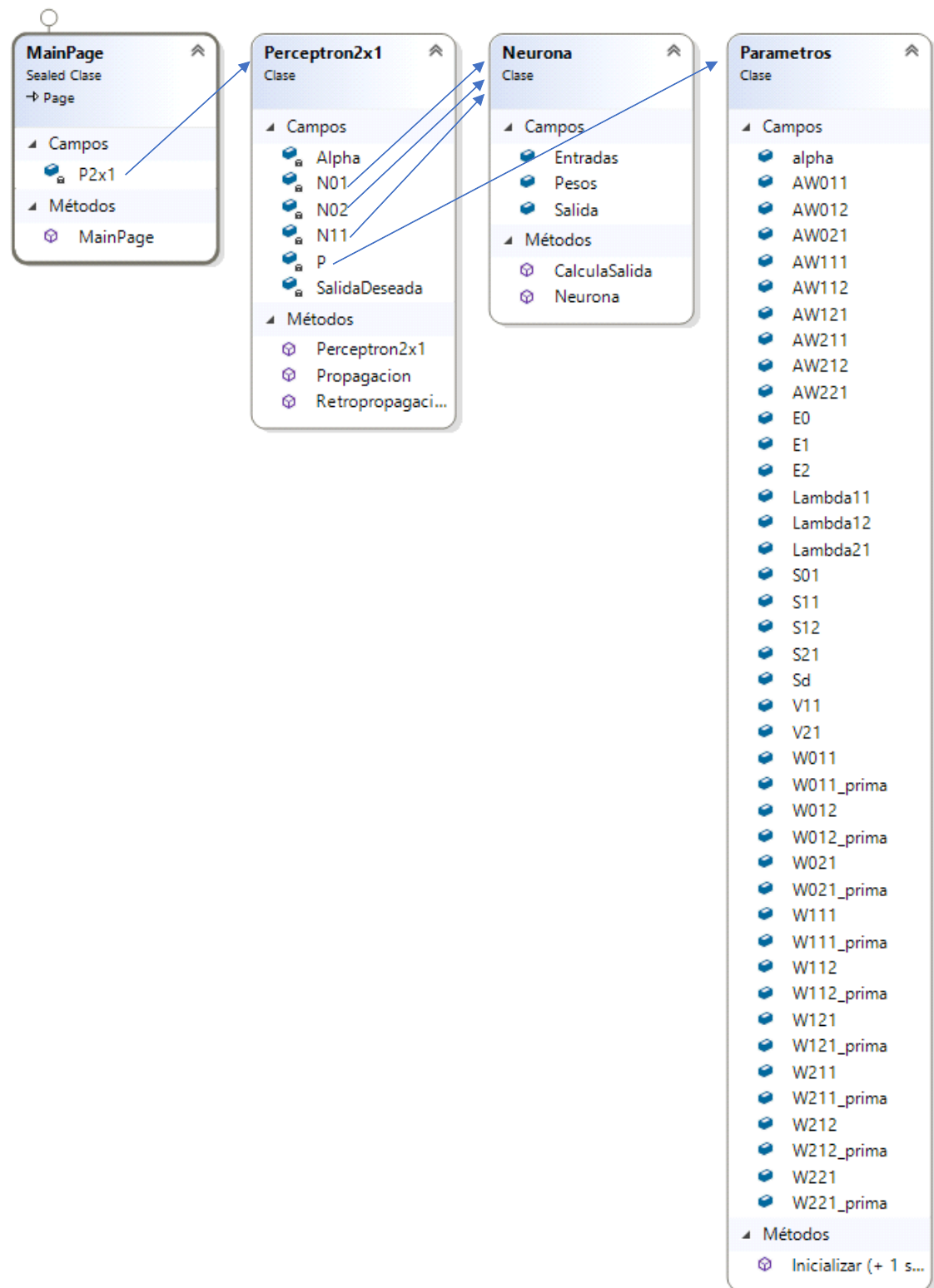
$$V_j^k = \sum_q (w_{jq}^{k+1} \cdot \delta_q^{k+1})$$

Donde q representa cada una de las neuronas de la capa $k+1$ cuya entrada es la salida de la neurona j de la capa k .

Finalmente, una vez calculados los ΔW_{ij}^k para todas las capas, se actualizan los pesos W_{ij}^k para la siguiente iteración ($t+1$) del entrenamiento:

$$W_{ij}^k(t+1) = W_{ij}^k(t) + \Delta W_{ij}^k(t)$$

DIAGRAMA DE CLASES (sugerido)



```

public class Neurona →
    public double[] Entradas=new double[3];
    public double[] Pesos = new double[3];
  
```

El resto de atributos son de tipo double