

# TC1003B – Modelación de la Ingeniería con Matemática Computacional

## Examen de Módulo

Nombre: \_\_\_\_\_

Matrícula: \_\_\_\_\_

Nombre: \_\_\_\_\_

Matrícula: \_\_\_\_\_

Nombre: \_\_\_\_\_

Matrícula: \_\_\_\_\_

Nombre: \_\_\_\_\_

Matrícula: \_\_\_\_\_

*Lee cuidadosamente y contesta lo que se te pide. Este examen está pensado para resolverse **en equipo**, y en poco menos de 90 minutos. Se sugiere que administres bien tu tiempo.*

*Al momento de contestar, intenten ser lo más explícito posible: se calificará con base en lo que esté escrito, y se considerará el proceso aún cuando la respuesta final esté errada. Recuerda que puedes revisar material de la clase, el libro de texto o tus notas. Buena suerte.*

### 1. Redes Neuronales (100 %)

Una red neuronal es un mecanismo de aprendizaje automático muy utilizado en inteligencia artificial que sirve para aproximar funciones usando reconocimiento de patrones. La función que la red neuronal aproxima puede servir para separar en grupos definidos (clasificación) o bien para predecir comportamientos (regresión).

Para entender cómo funciona una red neuronal, primero es necesario entender cómo funciona cada neurona.

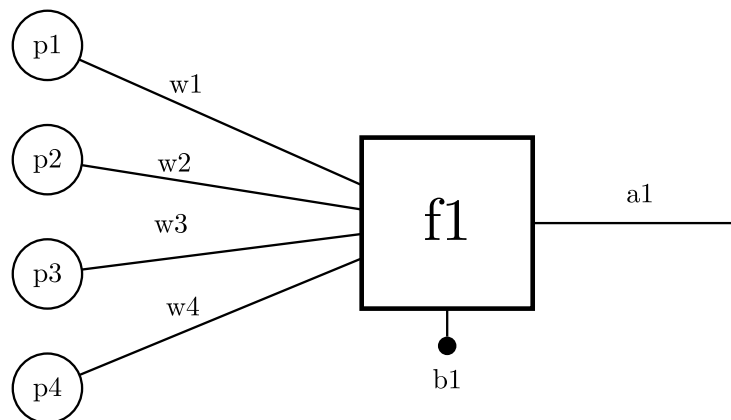


Figura 1: Perceptrón: una red neuronal de una sola neurona.

Un **perceptrón** es una red neuronal que consta de una sola neurona, como se muestra en la Figura 1. Una neurona recibe señales por medio de una serie de entradas. Cada entrada tiene conexiones hacia la

neurona, las cuales suelen tener distintos grados de influencia sobre la misma. Si en conjunto, las señales son lo suficientemente *estimulantes*—a pesar de la predisposición sensorial de la neurona—ésta se activa y *genera* una reacción. Podemos representar cada uno de sus componentes con símbolos matemáticos:

- Un conjunto de **entradas** o receptores sensoriales  $p_j$ , donde cada uno representa una característica o variable a tomar en cuenta
- Un conjunto de **pesos**  $w_j$ , donde cada uno determina la importancia que cada entrada ejerce sobre la neurona
- Una predisposición sensorial o **sesgo**  $b_i$ , que es una cantidad que afecta a la capacidad de la neurona para reaccionar a los estímulos
- Una **función de activación**  $f_i$ , que genera una reacción a los estímulos que percibe
- Una **salida** o reacción  $a_i$ , que es generada por la neurona dados los estímulos de sus entradas

La suma de cada una de las señales multiplicadas por la influencia que tienen sobre la neurona, más la predisposición sensorial de la misma, es evaluada usando la función de activación de la  $i$ -ésima neurona para generar una reacción  $a_i$ :

$$a_i = f_i \left( \sum_{j=1}^R w_j p_j + b_i \right) \quad (1.1)$$

donde  $R$  es el número de entradas en la red.

♦ Contesta para la Figura 1:

- ¿Cuántas entradas tiene la red? (2 %) \_\_\_\_\_
- ¿Cuántos pesos tiene la red? (2 %) \_\_\_\_\_
- ¿Cuántas neuronas tiene la red? (2 %) \_\_\_\_\_
- ¿Cuántas funciones de activación tiene la red? (2 %) \_\_\_\_\_
- ¿Cuántas salidas tiene la red? (2 %) \_\_\_\_\_

Para hacer el entrenamiento del modelo, el perceptrón toma ejemplos etiquetados—una serie de valores para cada entrada y un resultado esperado. Tras hacer la evaluación del ejemplo, el resultado obtenido se compara con el resultado esperado, y la diferencia entre ambos se toma como el *error* en la predicción. Los pesos de la red se ajustan, siempre buscando reducir el error, y se continúa el proceso con otro ejemplo. Después de unos cuantos (usualmente miles o incluso millones) de ejemplos, el perceptrón aprende un modelo general que trata de aproximar la mayoría de los ejemplos. Sin embargo, existen limitaciones de la arquitectura, y por ello se ha trabajado con arquitecturas más complejas. Por ejemplo, podemos organizar las neuronas en **capas**, de tal manera que podemos obtener una serie de reacciones de cada capa en lugar de una sola.

La Figura 2 ilustra una red neuronal de una sola capa, la cual tiene múltiples entradas y múltiples salidas. Bajo esta arquitectura, cada entrada  $j$  tiene una conexión hacia cada neurona  $i$ .

♦ Contesta para la Figura 2:

- ¿Cuántas entradas tiene la red? (2 %) \_\_\_\_\_
- ¿Cuántos pesos tiene la red? (2 %) \_\_\_\_\_

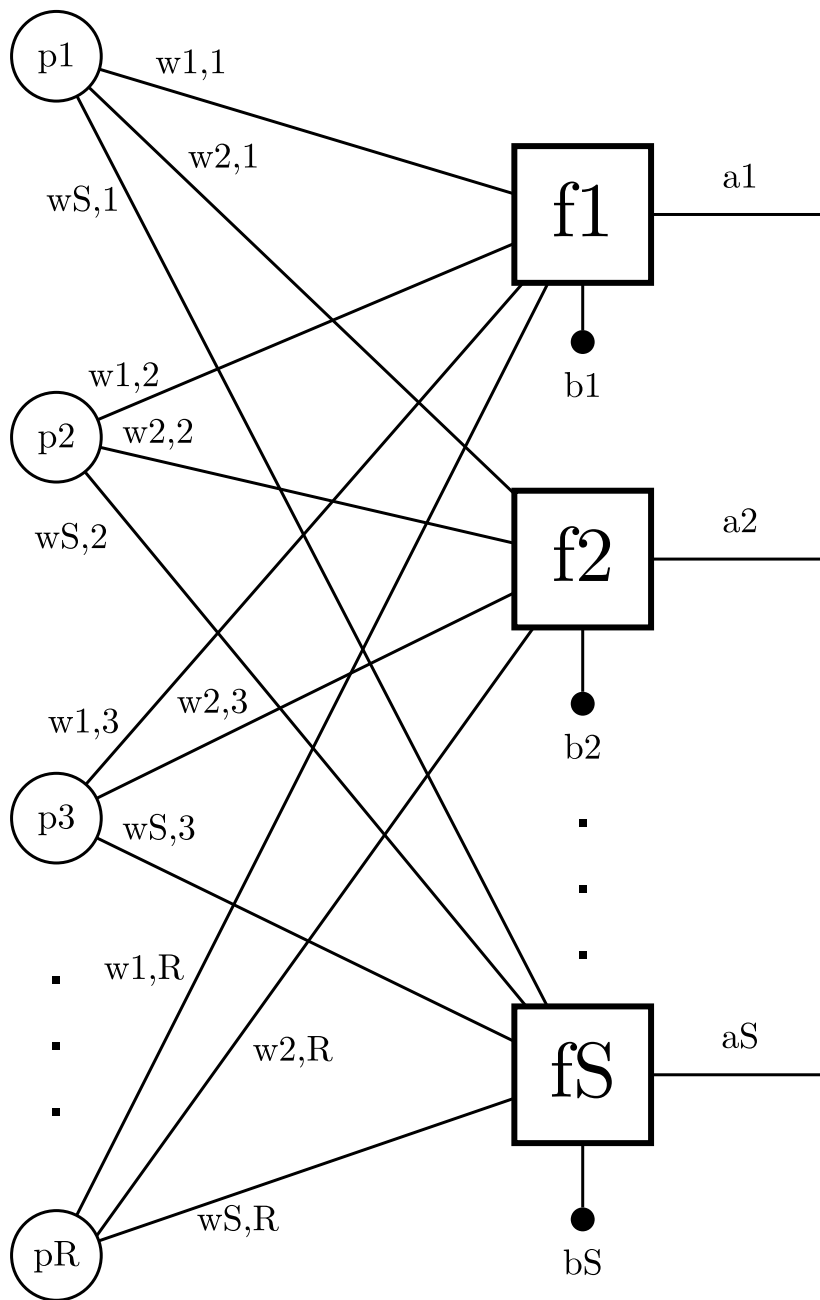


Figura 2: Red neuronal de una sola capa.

- ¿Cuántas neuronas tiene la red? (2%) \_\_\_\_\_
- ¿Cuántas funciones de activación tiene la red? (2%) \_\_\_\_\_
- ¿Cuántas salidas tiene la red? (2%) \_\_\_\_\_

Si te das cuenta, es posible *agrupar* muchas de las variables en juego usando estructuras de datos.

◆ ¿Qué estructura de datos usarías para representar los componentes de la nueva arquitectura? Describe la estructura de cada uno (5%)

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_

Al trabajar todo de manera vectorial, podemos *encadenar* capas para conseguir resultados más finos.

La Figura 3 ilustra cómo es el proceso de encadenamiento de capas, donde el vector de salidas de una capa es el vector de entradas de la siguiente capa.

En este caso, el proceso de entrenamiento es el mismo, sin embargo las operaciones se pueden simplificar si trabajamos con vectores:

$$\mathbf{a} = \mathbf{f}(W\mathbf{p} + \mathbf{b}) = \mathbf{q} \quad (1.2)$$

Vale la pena considerar que una serie de ejemplos  $P = [\mathbf{p}_1 \dots \mathbf{p}_S]$  suele venir en forma matricial, con dimensiones  $S \times R$ , por lo que para poder multiplicar hay que transponer los ejemplos.

A continuación se presentan algunos ejemplos  $\mathbf{p}$  etiquetados con sus resultados  $t$  en la forma  $P = \mathbf{t}$ :

$$3p_1 + 6p_2 - 9p_3 = 3 \quad (1.3)$$

$$2p_1 + 4p_2 - 8p_3 = 0 \quad (1.4)$$

$$-2p_1 - 3p_2 + 4p_3 = -1 \quad (1.5)$$

$$4p_1 + 3p_2 - 7p_3 = -6 \quad (1.6)$$

$$5p_1 + 4p_2 - 9p_3 = -7 \quad (1.7)$$

$$7p_1 + 7p_2 - 13p_3 = -6 \quad (1.8)$$

$$-2p_1 - 5p_2 + 9p_3 = -2 \quad (1.9)$$

$$-p_1 + 6p_2 + 2p_3 = 22 \quad (1.10)$$

◆ Genera un sistema de ecuaciones lineales usando tres ecuaciones. Para seleccionarlas, utiliza el **último dígito de tu matrícula** mód 8, enumerando las ecuaciones del 0 al 7 (35%)

Considera ahora que tienes una de las siguientes matrices de pesos para la primera capa de una red neuronal con los siguientes datos:

$$W^0 = \begin{bmatrix} 1 & 2 & 2 \\ 2 & 3 & 1 \\ 4 & 2 & 2 \\ 3 & 0 & 2 \end{bmatrix} \quad W^1 = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 1 & 1 \\ 3 & 2 & 2 \\ 5 & 2 & 4 \\ 1 & 2 & 0 \end{bmatrix}$$

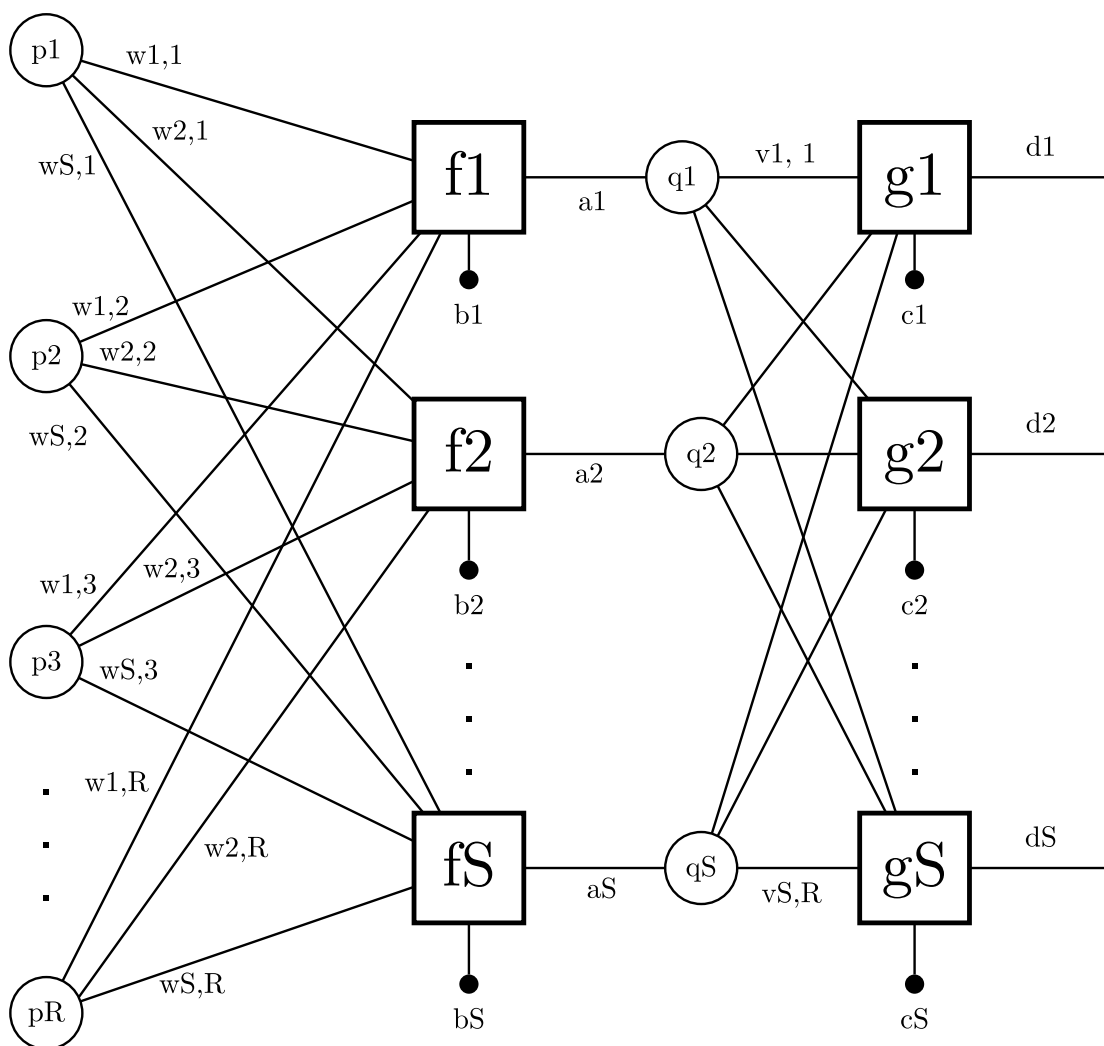


Figura 3: Red neuronal de múltiples capas.

♦ Selecciona una de las dos matrices usando tu **número de equipo** mód 2 y contesta:

- ¿Cuál es la longitud del vector de entradas  $\mathbf{p}$  de tu matriz? (5 %)
- 

- ¿Cuántas neuronas tiene la primer capa de la red neuronal que te tocó? (5 %)
- 

♦ A continuación, toma la matriz de ejemplos  $P$  (Ecuaciones 1.3–1.10) y calcula  $P^T$  (5 %)

♦ Después, encuentra el vector de salidas de la primera capa ( $\mathbf{a}$ ) considerando que la función de activación de cada neurona en dicha capa de tu red es la función identidad (es decir que  $y = x$ ), y el vector de sesgos  $\mathbf{b}$  es o bien  $\mathbf{b}^0 = [1 \ 2 \ -1 \ 1]^T$  o  $\mathbf{b}^1 = [-1 \ 0 \ 0 \ 2 \ -1]^T$ . *Hint: la ecuación 1.2 es de vital importancia* (25 %)

## Rescate (+5 %)

¿Cuál es la principal diferencia entre la descomposición de la matriz de coeficientes entre el método Jacobi y el Gauss-Seidel?

De acuerdo con el Código de Ética del Tecnológico de Monterrey, mi desempeño en esta actividad estará guiado por la integridad académica.