 

**Universidad Autónoma de Sinaloa**

**Ingeniería de Software**

**4-03**

**Redes Neuronales**

**Tarea 3.4: How a Machine Learns**

**Elaborado por: Valdez Villegas P. C. (Evie C.)**

**Profesor: M.C. Edgar Omar Pérez Contreras**

**07 de Mayo del 2024**

**Índice**

[Introducción: 3](#_Toc165932755)

[Vocabulario (140): 4](#_Toc165932756)

[Cuestionario (141) 5](#_Toc165932757)

# Introducción:

En el mundo de la inteligencia artificial y el aprendizaje automático, las redes neuronales son una herramienta fundamental. En esta segunda edición de "Introduction to Neural Networks with Java", nos sumergimos en los conceptos esenciales que sustentan estas poderosas redes, explorando cómo pueden aprender y adaptarse para resolver una variedad de problemas.

El capítulo resumen destaca la importancia crucial de la tasa de error en el proceso de entrenamiento de una red neuronal. Desde calcular el error de salida para un solo elemento del conjunto de entrenamiento hasta determinar el error RMS para todo el conjunto, cada paso nos acerca más a una red más precisa y efectiva.

El entrenamiento es la clave para el éxito de una red neuronal. Ya sea mediante la modificación de los pesos de la sinapsis para mejorar la salida en el entrenamiento supervisado, o dejando que la red determine las respuestas correctas en el entrenamiento no supervisado, cada método tiene su lugar en el proceso de aprendizaje. Hebb's Rule se destaca como una herramienta invaluable para el entrenamiento no supervisado, mientras que el Delta Rule brilla en el entrenamiento supervisado.

En este capítulo, hemos destapado el velo sobre cómo las máquinas aprenden, revelando los principios básicos que subyacen a este proceso fascinante. El concepto de retropropagación (backpropagation) emerge como una técnica avanzada derivada del Delta Rule, prometiendo un aprendizaje aún más sofisticado. En el próximo capítulo, exploraremos en profundidad la retropropagación y veremos cómo se implementa en la clase de redes neuronales.

Con esta base sólida y un sentido de anticipación, nos preparamos para sumergirnos en el corazón mismo de la inteligencia artificial, listos para desentrañar sus misterios y desafíos con la ayuda de Java y las redes neuronales. ¡Bienvenidos a un viaje de descubrimiento y aprendizaje sin igual!

# Vocabulario (140):

**Clasificación - Classification:** Proceso de asignar una etiqueta o categoría a un conjunto de datos según ciertas características o atributos.

**Regla Delta - Delta Rule:** Algoritmo utilizado en el aprendizaje supervisado para ajustar los pesos de una red neuronal artificial con el fin de minimizar el error entre las salidas deseadas y las salidas reales.

**Época - Epoch:** En el contexto del entrenamiento de una red neuronal, una época se refiere a una pasada completa a través de todo el conjunto de datos de entrenamiento durante el proceso de aprendizaje.

**Regla de Hebb -** Hebb's Rule: Postulado en neurociencia que establece que la fuerza de una conexión sináptica entre dos neuronas aumenta si ambas neuronas son activadas simultáneamente de manera repetida.

**Iteración - Iteration:** Un paso o ciclo repetitivo en un algoritmo o proceso, especialmente en el contexto del entrenamiento de redes neuronales donde una iteración puede referirse a un paso a través de un solo ejemplo de entrenamiento.

**Red Neuronal de Kohonen - Kohonen Neural Network:** Un tipo de red neuronal artificial utilizada para el aprendizaje no supervisado, especialmente en tareas de agrupamiento o clustering, donde las neuronas compiten por representar diferentes regiones de entrada.

**Tasa de Aprendizaje - Learning Rate:** Un parámetro en algoritmos de aprendizaje automático que controla la magnitud de los ajustes de los pesos de la red neuronal durante el entrenamiento.

**Raíz Cuadrada Media (RMS) - Root Mean Square (RMS):** Una medida estadística utilizada para calcular el promedio de los valores cuadrados de un conjunto de números.

**Mapa Auto-organizado (SOM) - Self-Organizing Map (SOM):** Un tipo de red neuronal artificial que organiza datos de alta dimensión en un espacio de baja dimensión de manera topológica preservada, útil para la visualización y la agrupación de datos.

**Entrenamiento Supervisado - Supervised Training:** Un tipo de entrenamiento de redes neuronales donde el modelo se ajusta utilizando ejemplos de entrada y salida esperada, lo que permite al modelo aprender a predecir salidas correctas para nuevas entradas.

**Entrenamiento No Supervisado - Unsupervised Training:** Un tipo de entrenamiento de redes neuronales donde el modelo se ajusta sin necesidad de ejemplos de salida esperada, permitiendo al modelo descubrir patrones o estructuras ocultas en los datos de entrada.

# Cuestionario (141)

**1. Explica la diferencia entre el entrenamiento supervisado y el no supervisado.**

En el entrenamiento supervisado, el modelo se ajusta utilizando ejemplos de entrada y salida esperada, permitiendo al modelo aprender a predecir salidas correctas para nuevas entradas. En cambio, en el entrenamiento no supervisado, el modelo se ajusta sin necesidad de ejemplos de salida esperada, lo que le permite descubrir patrones o estructuras ocultas en los datos de entrada.

**2. ¿Cuál es la diferencia principal entre la regla delta y la regla de Hebb?**

La regla delta se utiliza en el aprendizaje supervisado para ajustar los pesos de una red neuronal artificial con el fin de minimizar el error entre las salidas deseadas y las salidas reales. En cambio, la regla de Hebb es un postulado en neurociencia que establece que la fuerza de una conexión sináptica entre dos neuronas aumenta si ambas neuronas son activadas simultáneamente de manera repetida.

**3. Considera los siguientes cuatro resultados; calcula el error RMS.**

Conjunto de Entrenamiento #1, Esperado = 5, Actual = 5

Conjunto de Entrenamiento #2, Esperado = 2, Actual = 3

Conjunto de Entrenamiento #3, Esperado = 6, Actual = 5

Conjunto de Entrenamiento #4, Esperado = 8, Actual = 4

Conjunto de Entrenamiento #5, Esperado = 1, Actual = 2

Calcula el error RMS.

**Paso 1: crear tabla de errores cuadraticos**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Esperado | Actual | Calculos |
| 1 | 5 | 5 | (5-5)^2=0^2=0 |
| 2 | 2 | 3 | (2-3)^2=-1^2=1 |
| 3 | 6 | 5 | (6-5)^2=1^2=1 |
| 4 | 8 | 4 | (8 - 4)^2 = 4^2 =16 |
| 5 | 1 | 2 | (1-2)^2=1^2=1 |

**Paso 2: promediar los valores recibidos**

(0 + 1 + 1 + 16 + 1) / 5 = 19 / 5 = **3.8**

**Paso 3: sacarle la raíz cuadrada al valor del paso 2**

RMS Error = √3.8 ≈ **1.949**

**4. Usa la regla de Hebb para calcular el ajuste de peso.**

Dos Neuronas: N1 y N2

Peso de N1 a N2: 3

Activación de N1: 2

Activación de N2: 6

**Convertir la formula**

ΔW = (N1 Activation) \* (N2 Activation)

ΔW = 2 \* 6 = **12**

Calcula el delta de peso.

**5. Usa la regla delta para calcular el ajuste de peso.**

Dos neuronas: N1 y N2

Peso de N1 a N2: 3

Activación de N1: 2

Activación de N2: 6, Esperado: 5

Calcula el delta de peso.

**Paso 1: Convertir la formula: *ΔW=α×(Expected−Actual)×N1 Activation***

ΔW=0.1×(5−6)×2= 0.1×(−1)×2= **-0.2**

*“El peso entre N1 y N2, se reducira en 0.2 unidades”*