

# Chapter 2: Error Calculation Methods

## Resumen del capitulo

El cálculo de errores es indispensable en una red neuronal, dado que nos proporciona un valor indicando el error producido por la red neuronal. Existen dos niveles en los que se producen errores: local y global. Para calcular el error global, necesitamos calcular el error local. Aclarando que hasta este punto nos estamos refiriendo a una red neuronal que es sometida a entrenamiento supervisado, y le proporcionamos una salida esperada por cada valor de entrada. Aclarado esto el error local se calcula con la resta del valor esperado menos el valor obtenido. Para el cálculo de los errores globales tenemos distintos métodos, los abordados en esta lección son:

* **Error cuadrado Medio (MSE):** Expresado como porcentaje, consiste en sumar los cuadrados del error local, divido entre el número total de casos. Muestra la tasa de error porcentualmente.
* **Error de suma de cuadrados (ESS):** Este valor no se expresa como porcentaje, si no, como un valor numérico, entra más grande el valor más grave el error. Y se calcula con la suma de los cuadros del valor del error local divido entre 2.
* **Error cuadrado medio raíz (RMS):** Se calcula sacando raíz cuadrada al MSE, el valor obtenido es expresado de manera porcentual y siempre es mayor al valor de MSE.

## Preguntas de revisión

Considere la siguiente tabla, en ella se ilustra las entradas, las salidas producidas y las salidas esperadas de una red neuronal.

|  |  |  |  |
| --- | --- | --- | --- |
| **l1** | **l2** | **Salida Esperada** | **Out O1** |
| **1** | **1** | **0** | **0.1044108073** |
| **1** | **0** | **1** | **0.9058454223** |
| **0** | **1** | **1** | **0.9058565170** |
| **0** | **0** | **0** | **0.0972743271** |

1. Calcule el error local para cada salida.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **l1** | **l2** | **Salida Esperada** | **Out O1** | **Error** |
| **1** | **1** | **0** | **0.1044108073** | **0 - 0.1044108073 = -0.1044108073** |
| **1** | **0** | **1** | **0.9058454223** | **1 - 0.9058454223 = 0.0941545777** |
| **0** | **1** | **1** | **0.9058565170** | **1 - 0.9058565170 = 0.0941434830** |
| **0** | **0** | **0** | **0.0972743271** | **0 - 0.0972743271 = -0.0972743271** |

1. Calcule el error global mediante la fórmula MSE.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **l1** | **l2** | **Salida Esperada** | **Out O1** | **Error** |  |
| **1** | **1** | **0** | **0.1044108073** | **-0.1044108073** | **0.0109016166803189** |
| **1** | **0** | **1** | **0.9058454223** | **0.0941545777** | **0.0088650845047712** |
| **0** | **1** | **1** | **0.9058565170** | **0.0941434830** | **0.0088629953822324** |
| **0** | **0** | **0** | **0.0972743271** | **-0.0972743271** | **0.0094622947199112** |

**MSE** = =

**MSE** = **0.952299782%**

1. Calcule el error global mediante la fórmula ESS.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **l1** | **l2** | **Salida Esperada** | **Out O1** | **Error** |  |
| **1** | **1** | **0** | **0.1044108073** | **-0.1044108073** | **0.0109016166803189** |
| **1** | **0** | **1** | **0.9058454223** | **0.0941545777** | **0.0088650845047712** |
| **0** | **1** | **1** | **0.9058565170** | **0.0941434830** | **0.0088629953822324** |
| **0** | **0** | **0** | **0.0972743271** | **-0.0972743271** | **0.0094622947199112** |

**ESS** = =

**ESS** = **0.01904599564362**

1. Calcule el error global mediante la fórmula RMS.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **l1** | **l2** | **Salida Esperada** | **Out O1** | **Error** |  |
| **1** | **1** | **0** | **0.1044108073** | **-0.1044108073** | **0.0109016166803189** |
| **1** | **0** | **1** | **0.9058454223** | **0.0941545777** | **0.0088650845047712** |
| **0** | **1** | **1** | **0.9058565170** | **0.0941434830** | **0.0088629953822324** |
| **0** | **0** | **0** | **0.0972743271** | **-0.0972743271** | **0.0094622947199112** |

**RMS** = =

**RMS** = **0.952299782%**

## Vocabulario

**Error local:**

Diferencia producida entre la salida ideal menos la salida producida de una red neuronal artificial. Este error es necesario para el cálculo de error global e indica en forma numérica el error producido en las salidas de una red neuronal artificial.

**Salida ideal:**

Salida proporcionada a la red para indicarle cual es la salida esperada para una entrada especifica. Se utiliza para calcular el error producido por la red, y en base a ese valor generar un ajuste en los valores de los pesos de conexión entre las neuronas.

**Error global:**

Dependiendo del método a utilizar para el cálculo global podemos tener que el error global se puede presentar en forma porcentual o numérica. En ambos casos este valor indica la tasa de error producido en las salidas de la red neuronal.

**Salida real:**

Valor producido por la red neuronal artificial.

# Chapter 4: How a Machine Learns

## Resumen del capitulo

¿Cómo aprenden las maquinas?, es la pregunta que se aborda en este capítulo. Dentro del contexto de las redes neuronales el significado de aprender comprende del ajuste de los valores de la matriz de pesos de la red neuronal a medida que se le presentan entradas nuevas. Estos ajustes provocan que la red vaya aumentando su tase de éxito. Nos encontramos con métodos de entrenamiento, supervisado y no supervisado, cada uno de ellos tiene sus cualidades específicas y la forma de realizar los cálculos para los ajustes son un tanto distintos.

Para el entrenamiento supervisado tenemos el algoritmo Delta, que se base en el calculo de error para realizar el ajuste de los pesos, su funcionamiento consiste en lo siguiente, para esto consideremos tener la matriz de pesos en 0 y el cálculo de las salidas ya preparadas:

1. Se debe proporcionar una salida esperada por cada entrada, de tal manera que cada patrón presentado a la red será acompañado del valor de salida esperado, además del valor de la tasa de aprendizaje esperado.
2. La red neuronal calcula la salida, a través de la obtención de este valor calcula el error local restando a la salida esperada la salida obtenida.
3. Por cada neurona obtenemos el valor de delta, multiplicando la entrada correspondiente con el error local por el valor de la tasa de aprendizaje.
4. Una vez terminado los valores de delta se acumulan en la matriz de pesos en sus respectivos lugares.
5. Procedemos con el siguiente patrón.

Para un entrenamiento no supervisado no proporcionamos salida esperadas, por lo tanto, no tenemos forma de realizar el calculo de error local, por lo tanto, para el cálculo de delta tomamos el valor de la entrada, el valor de la salida y la tasa de aprendizaje. Lo demás es similar a como se trabaja en el algoritmo delta. Par este tipo de redes se utiliza el algoritmo Hebb.

## Preguntas de revisión

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

Para empezar, en el entrenamiento supervisado se proporcionan las salidas esperadas por la red neuronal lo que permite el cálculo del error local producido por la red, y en base a este valor realizar los ajustes de pesos de la red neuronal. Mientras que en el no supervisado la red neuronal no posee salidas anticipadas, y el ajuste de pesos se realiza en base a las entradas y las salidas producidas.

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

La regla delta realiza el cálculo de error local producido por la red, y en base a ese valor se realiza el ajuste de la red, mientras que en la regla de hebb se toman en cuenta las entradas y las salidas producidas.

1. **Considere los siguientes cuatro resultados; calcular el error RMS.**

**Conjunto de entrenamiento #1, Esperado = 5, Real = 5**

**Conjunto de entrenamiento #2, Esperado = 2, Real = 3**

**Conjunto de entrenamiento #3, Esperado = 6, Real = 5**

**Conjunto de entrenamiento #4, Esperado = 8, Real = 4**

**Conjunto de entrenamiento #5, Esperado = 1, Real = 2**

195%

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

**Dos neuronas: N1 y N2 N1 a N2 Peso: 3**

**Activación N1: 2**

**Activación N2: 6**

**Calcule el delta del peso.**

**Output = 3 \* 2 = 6**

**Delta w1 = 1 \* 2\* 6 = 12**

**Delta w2 = 1 \* 6 \* 6 = 36**

1. **Utilice la regla delta para calcular el ajuste de peso.**

**Dos neuronas: N1 y N2 N1 a N2 Peso: 3**

**Activación N1: 2**

**Activación N2: 6, Se espera: 5**

**Calcule el delta del peso.**

**Output = 3 \* 2 = 6**

**Error = 5 – 6 = -1**

**Delta w1 = 1 \* 2 \* -1 = -2**

**Delta w2 = 1 \* 6 \* -1 = -6**

## Vocabulario

**Classification**

Dentro del tema de redes neuronales, se refiere a un tipo de red neuronal que se encarga de agrupar patrones dada ciertas características, en donde se utiliza entrenamiento no supervisado, de esta manera es la propia red quien se encarga de aprender el cómo agrupar los patrones que se son proveídos.

**Delta Rule**

Algoritmo de entrenamiento utilizado en el entrenamiento supervisado, dado la característica de que se le proporcionan las entradas ideales para cada entrada. Por entrada realiza un ajuste en los pesos de conexión de las neuronas correspondientes, de tal manera que la red neuronal aprende. El único inconveniente de esta red neuronal es que los valores de ajuste nunca serán iguales a los esperados, solo serán una aproximación.

**Epoch**

Entrenamiento realizado con un conjunto de datos. E la época la red neuronal itera sobre este conjunto de datos, uno a uno, procesa los valores y realiza un ajuste a la matriz de pesos.

**Hebb's Rule**

Algoritmo de entrenamiento, que es utilizado en el entrenamiento no supervisado. Tiene como cualidad el hecho de que no recibe salidas anticipadas, por lo tanto, el reajuste de pesos lo realiza en base a las entradas y las salidas producidas. Este tipo de algoritmos

**Iteration**

Procesamiento de un valor dentro de un conjunto de datos dados para la red neuronal. De manera más precisa, las redes neuronales se entrenan por épocas, en cada época se proporcionan un conjunto de valores que serán entradas para la red, y el procesamiento individual de cada valor representa una iteración.

**Kohonen Neural Network ó Self-Organizing Map (SOM):**

Tipo de red neuronal que tiene un entrenamiento no supervisado que funcionan para clasificación. A este tipo de redes se les proporciona un conjunto de datos y ella misma se encarga de elegir el criterio de clasificación, esto lo desarrolla en base se le vayan presentando patrones, permitiéndole agrupar los datos con características similares.

**Learning Rate**

Valor que se le proporciona a la red indicando la velocidad con la que esta debe aprender.

**Root Mean Square (RMS)**

Método para el cálculo de erro de un conjunto de datos, basado en las salidas anticipadas proporcionadas a la red neuronal. Proporciona un valor indicando la tasa de e error producido por la red, sin tomar en cuenta si los valores están por debajo o por encima.

**Supervised Training**

Tipo de entrenamiento para una red neuronal en que implica que por cada patrón presentado a la red se le proporcionara una salida esperada, de tal manera, que se pueda obtener un valor de error y así realizar un ajuste los pesos en base a este valor.

**Unsupervised Training**

Tipo de entrenamiento para una red neuronal en el cual no se le provee una salida anticipada a la red por cada patrón, de esta manera, el entrenamiento no supervisado no realiza cálculo de error y el ajuste de pesos lo realiza en base a las entras y las salidas producidas.