UNIVERSIDAD CARLOS III DE MADRID
OPENCOURSEWARE
REDES DE NEURONAS ARTIFICIALES
INES M. GALVAN - JOSE Mª VALLS

EVANNAI



# PREPROCESAMIENTO DE DATOS

## Características de los datos

Para que un problema pueda ser abordado con una red de neuronas es necesario que se disponga de un conjunto de datos, muestras o instancias representativo del problema a resolver. Los datos o también patrones pueden estar de compuestos de:

- Variables, datos o patrones de **entrada** y variables, datos o patrones de **salida deseada** para aquellas redes que utilicen aprendizaje supervisado. Las variables de salida deseada servirán para corregir la respuesta de la red.
- Variables, datos o patrones de **entrada** para aquellas redes que utilicen aprendizaje no supervisado. Las variables de salida deseadas no se conocen.

Los patrones de entrada y salida generalmente están formados por un conjunto de valores que reciben el nombre de atributos. A diferencia de otros sistemas de aprendizaje automático, las redes de neuronas sólo trabajan con atributos numéricos y no con atributos nominales. Los atributos numéricos son atributos que toman valores reales o enteros, como por ejemplo temperatura, humedad, edad de una persona, número de hijos. Los atributos nominales son atributos discretos o categóricos, como por ejemplo estado del cielo, viento, DNI, tiene/no tiene coche.

Cuando el problema a abordar posee todos o algunos atributos nominales, es necesario discretizarlos (asignar a cada valor un número entero o real) para poder ser procesados por una red de neuronas. Esta discretización puede influir en los resultados de la red. Hay otros algoritmos de aprendizaje automático que permiten el uso tanto de atributos numéricos como nominales, y están diseñados con esta finalidad.

## Transformación de datos

La calidad de los modelos de redes de neuronas puede depender de la calidad de los datos. Por ello, después de la recopilación de datos es necesario preparar el conjunto de datos disponibles. En esta preparación, hay dos fases, que aunque no son obligatorias, siempre es recomendable realizar. Ellas son:

- **Normalización de los datos.** Es aconsejable trabajar con datos normalizados en un cierto intervalo, generalmente el intervalo [0,1]. Si bien no es obligatorio, siempre es recomendable pues evita problemas durante el aprendizaje, como la saturación de las neuronas.
- **Aleatorización de los datos**. Para evitar sesgos en el aprendizaje, siempre es conveniente aleatorizar los patrones o datos disponibles. Posteriormente a esta aleatorización, el conjunto de patrones disponibles se separa en conjunto de entrenamiento y test (ver evaluación de la red de neuronas).

Además de estas dos fases, los datos pueden sufrir más transformaciones antes de ser procesados por una red de neuronas: eliminación de atributos irrelevantes y reducción de dimensionalidad, aunque estas fases no siempre es necesario llevarlas a cabo y no son motivo de estudio para el presente curso. En cualquier caso, se describen brevemente a continuación.

- **Atributos irrelevantes**: En ocasiones los patrones disponibles poseen una serie de atributos que a priori, y según el conocimiento del problema, se consideren irrelevantes para la resolución de dicho problema. Si esto es conocido, es conveniente eliminarlos.
- **Reducción dimensionalidad**: Para algunos problemas, los datos de entrada poseen alta dimensión (entendiendo como tal un número de atributos alrededor o mayor que una centena), lo cual puede impedir o dificultar el aprendizaje de la red. En estos casos es conveniente aplicar técnicas de reducción de dimensionalidad, que bien seleccionan un subconjunto de atributos, bien transforman los datos de entrada en otro conjunto de menor dimensión. No se entra en detalle en los diferentes métodos de reducción de dimensionalidad, pues no es objetivo del curso.

## Evaluación de la red de neuronas

Para la obtención y construcción de una red de neuronas se utiliza un conjunto de datos, llamado **conjunto de entrenamiento**. Para medir la calidad del modelo o capacidad de generalización de la red es necesario observar el comportamiento de la red con datos no utilizados para su entrenamiento; estos son los llamados **datos de test**. Esto permitirá medir la capacidad de generalización del modelo, es decir, la capacidad de responder correctamente antes situaciones (o datos) diferentes, pero representados en el conjunto de entrenamiento. Por ejemplo, si a un alumno se le evalúa (examen) con los mismos problemas con los que aprendió, no se demuestra su capacidad de generalización.

Para ello, habitualmente lo que se hace es dividir el conjunto de datos disponible en un subconjunto para entrenamiento y otro para test. Esta división debe realizarse aleatoriamente, o al menos utilizando un mecanismo que garantice que los datos de test están representados en el conjunto de entrenamiento. En el caso de las redes de neuronas artificiales, en ocasiones, del conjunto de entrenamiento se extrae una porción de datos, llamado **conjunto de validación** que se utiliza para parar el aprendizaje de la red.

Es posible, incluso por azar, que los datos de entrenamiento y test aparezcan sesgados, lo cual no es deseable. También puede ocurrir que si el número de datos que representan el problema es reducido, hacer una separación de un cierto porcentaje para entrenamiento y test, puede conducir a una evaluación engañosa de la red, además de suponer un riesgo para el aprendizaje el no poder utilizar los patrones que han caído en el conjunto de test. Para evitar estos problemas y hacer una evaluación realista es conveniente dividir el conjunto de datos disponibles utilizando el método conocido como **validación cruzada**. Este método consiste básicamente en dividir varias veces el mismo conjunto de datos en entrenamiento y test y calcular la media de los resultados de evaluación en los diferentes conjuntos de test. Así es más complicado que todas las veces se produzcan sesgos. El procedimiento general consiste en:

Se divide el conjunto de datos original en k partes. Con k=3 tenemos los subconjuntos A, B, y C. Se realizan entonces k=3 iteraciones:

— Aprender con A, B y test con C (T1 = medida de evaluación con C)

— Aprender con A, C y test con B (T2 = medida de evaluación con B)

— Aprender con B, C y test con A (T3 = medida de evaluación con A)

— medida de evaluación final T = (T1+T2+T3)/3

El modelo final se construye con los tres conjuntos (A, B y C) y se supone que T es una estimación de la evaluación del modelo. Generalmente, se suele utilizar k=10

## Medidas de Evaluación

Una vez separados los datos de entrenamiento y test y construida la red, es necesario definir y fijar una medida de evaluación, la cual va a depender de la tarea o problema a resolver. A continuación, se hace un breve repaso sobre las medidas más utilizadas en cada uno de los diferentes problemas.

**Clasificación:**

Lo más habitual es evaluar la calidad de la red con base en su precisión predictiva, la cual se calcula como el número de patrones del conjunto de prueba (entrenamiento o test) clasificadas correctamente, dividido por el número total de instancias en dicho conjunto.

Las salidas de una red de neuronas como el Perceptron Multicapa o las Redes de Base Radial toman valores continuos (reales). Cuando se aborda un problema de clasificación, las salidas deben indicar la clase a la que pertenece su correspondiente patrón de entrada, por lo que los valores continuos deben transforme a valores discretos. Es decir, es necesario interpretar la clase que representa esa salida. Esta interpretación depende de cómo se haya formulado el problema de clasificación desde el punto de vista de las redes de neuronas.

Cuando la salida deseada para la red se define como $S(n)=(0…1…0)$ si el patrón de entrada n pertenece a la clase i, entonces la interpretación de la salida de la red $S^{red}(n)=(a_1,a_2,…,a_m)$ será:

> El patrón n pertenece a la clase correspondiente a la neurona con la máxima activación. Es decir, si $a_j=max(a_i)$, entonces el patrón n pertenece a la clase j.

A lo hora de decidir si la red resuelve con éxito un problema de clasificación, es necesario tener presente una serie de criterios para decidir si una red es o no aceptable resolviendo el problema. Estos son:

- En problemas de clasificación con M clases, el porcentaje de aciertos debe superar el 100*1/M. De otra manera, sería mejor tirar una moneda (azar) que utilizar el clasificador para predecir
- Si por la naturaleza del problema, se dispone de una clase con muchos más datos que otra, el porcentaje de aciertos a superar es el porcentaje de datos de la clase mayoritaria. Por ejemplo, si tenemos dos clases (+ y -) y en los datos disponibles hay 90 datos + y 10 -; un clasificador que prediga siempre + (independientemente de los atributos), ya acertará en un 90%. Hay que hacerlo mejor que eso.
- En ocasiones, en problemas de clasificación, el costo de fallar en una clase no es el mismo que fallar en otra. Por ejemplo, para un clasificador de cáncer si/no, es preferible predecir que una persona tiene cáncer (sin tenerlo) que predecir que no lo tiene (teniéndolo). Para analizar esos casos es conveniente utilizar la matriz de confusión, que contiene información sobre las falsos positivos y falsos negativos. Dado un problema de clasificación con 2 clases, la matriz de confusión es una matriz cuadrada

(de 2 x 2 elementos), que contiene la siguiente información:

| | Clasificado como + | Clasificado como - |
|---|---|---|
| Dato realmente + | TP (true positive) | FN (false negative) |
| Dato realmente - | FP (false positive) | TN (true negative) |

Los datos correctamente clasificados están en la diagonal, los incorrectos fuera de ella: TP=Acierto, FN=Omisión, FP=Falsa alarma, TN=Rechazo correcto.

- El porcentaje de aciertos es

  Accuracy = (TP+TN)/(TP+TN+FP+FN)

- El porcentaje de aciertos de + es:

  Recall (sensitivity o true positive rate) = TP/(TP+FN)

- El porcentaje de aciertos – es:

  Specificity (true negative rate) = TN/(TN+FP)

**Regresión:**

En problemas de regresión la salida de la red es un valor numérico, y la manera más habitual (aunque existen otras) de evaluar la red es mediante el error cuadrático medio cometido por ésta. Este error evalúa la suma mediada por el número de patrones de las diferencias al cuadrado de la salida de la red y la salida deseada, es decir:

$$E = \sum_{i=1}^{N} (sm_i - sd_i)^2$$

siendo $sm_i$ y $sd_i$ la salida de la red y salida deseada para el patrón i, respectivamente.

**Agrupación:**

Para los problemas de agrupación o clustering, los mapas de Kohonen miden la cohesión de cada grupo o cluster y la separación entre los grupos. Estas características, cohesión y separación, se suele formalizar utilizando la distancia media de los miembros de un grupo a su centro y la distancia media entre los grupos, respectivamente. En forma general, esta distancia es la Euclidiana.