universidad de león

Escuela de Ingenierías Industrial, Informática y Aeroespacial

GRADO EN INGENIERÍA INFORMÁTICA

Prácticas Externas

Conceptos básicos

Autor: María García Girón

30 de septiembre, 2021

Aprendizaje automático

El aprendizaje automático o machine learning es una rama de la inteligencia artificial que permite que las máquinas aprendan sin ser expresamente programadas para ello.

De forma más concreta, se trata de crear algoritmos capaces de generalizar comportamientos y reconocer patrones a partir de una información suministrada en forma de ejemplos para hacer predicciones.

Dataset

Un dataset no es más que un conjunto o colección de datos.

El dataset corresponde a los contenidos de una tabla, donde cada columna representa una variable en particular, y cada fila representa a un miembro determinado del conjunto de datos que estamos tratando.

Este conjunto de datos es el necesario para entrenar el modelo. El conjunto de datos se compone de instancias, y las instancias de características.

Instancia

Una instancia es cada uno de los datos de los que se disponen para crear el modelo. Si se quiere predecir si un correo es spam o no es spam, cada instancia correspondería a un email. Cada instancia, a su vez, está compuesta de características.

Característica

Una característica, o atributo, describen cada una de las instancias del conjunto de datos. En el caso de predicciones de correo podrían ser características el número de errores ortográficos o la hora de entrada del correo electrónico.

Etiqueta

Los datos etiquetados se refieren a los datos del conjunto de entrenamiento de los que ya se conoce su salida final. Este tipo de datos etiquetados se utiliza solamente en el entrenamiento supervisado.

Entrenamiento

Es el proceso en el que se proporcionan los datos de entrenamiento de los cuales aprender el algoritmo es cuestión. Una vez el modelo haya aprendido e identificados los patrones, se pueden hacer predicciones con nuevos datos que se incorporen al sistema.

Hiperparámetro

Los hiperparámetros de un modelo son los valores de la configuración utilizadas durante el proceso de entrenamiento.

Son valores que generalmente se no se obtienen de los datos, por lo que suelen ser indicados manualmente. El valor óptimo de un hiperparámetro no se puede conocer a priori. Por lo que se tiene que utilizar valores genéricos, reglas o los valores que han funcionado anteriormente en problemas similares.

Algunos ejemplos de hiperparámetros utilizados para entrenar los modelos son:

- El learning rate en el algoritmo del descenso del gradiente. La estrategia de selección consiste el comenzar con un valor muy pequeño como 10^{-5} e ir incrementando hasta llegar a 10.
- El número de capas del modelo.
- El número de neuronas.
- El batch size. Utilizar batches de tamaño 2 a 32 suele ser la selección de preferencia.

- La función de activación. Se suele utilizar la función relu en las hidden layers, la función sigmoide en la output layer de los problemas de clasificación y la función softmax en problemas de regresión.

- La función de optimización. Encontramos funciones de optimización muy variadas como Adam, Mini-Batch Gradient Descent o Stochastic Gradient Descent

En este contexto aparece la herramienta Keras Tuner, que se encarga de optimizar los hiperparámetros de un modelo.

Predicción

Anticiparse a las situaciones resulta de gran ayuda para la correcta toma de decisiones. La inteligencia artificial consigue a través del **análisis predictivo** obtener mejores resultados, optimizando tiempos y esfuerzos. He ahí la importancia de los datos para generar patrones predictivos.

Métricas

Las métricas en la inteligencia artificial se utilizan para comprobar como de bien ha aprendido nuestro algoritmo. La matriz de confusión es un claro ejemplo de comprobar *qué tipos de aciertos y errores está teniendo nuestro modelo a la hora de pasar por el proceso de aprendizaje con los datos*.

Dado que hay dos posibles valores reales y dos posibles valores de predicción. A partir de estas opciones podemos crear lo que se conoce como la matriz de confusión con 4 resultados posibles:

- Verdadero positivo: El valor real es positivo y la prueba predice un positivo.
- Verdadero negativo: El valor real es negativo y la prueba predice un negativo.

- Falso negativo: El valor real es positivo, y la prueba predice un negativo.
- Falso positivo: El valor real es negativo, y la prueba predice un positivo.

A partir de estas 4 variables surgen las métricas de la matriz de confusión:

- La Exactitud o Accuracy

Se refiere a lo cerca que está el resultado de una medición del valor verdadero. Se representa como la proporción de resultados verdaderos (tanto verdaderos positivos como verdaderos negativos) dividido entre el número total de casos. Se refiere entonces a la cantidad *de predicciones positivas que fueron correctas*.

El problema con la exactitud es que nos puede llevar al engaño, es decir, puede hacer que un modelo malo parezca que es mucho mejor de lo que es.

- La Precisión

Con la métrica de precisión podemos medir la calidad del modelo de machine learning en tareas de clasificación.

Se refiere a la dispersión del conjunto de valores obtenidos a partir de mediciones repetidas de una misma magnitud. Cuanto menor es la dispersión mayor la precisión.

Se representa por la proporción de verdaderos positivos dividido entre todos los resultados positivos. En forma práctica es el *porcentaje de casos positivos detectados*.

- Recall o Sensibilidad

Es la proporción de casos positivos que fueron correctamente identificadas por el algoritmo.

- Especificidad

También conocida como la Tasa de Verdaderos Negativos. Se trata de los casos negativos que el algoritmo ha clasificado correctamente.

- F1 SCORE

Esta es otra métrica muy empleada porque nos resume la precisión y sensibilidad en una sola métrica. Por ello es de gran utilidad cuando la distribución de las clases es desigual, por ejemplo, cuando el número de pacientes con una condición es del 15% y el otro es 85%

Redes neuronales

Las redes neuronales artificiales son un modelo inspirado en el funcionamiento del cerebro humano. Esta formado por un conjunto de nodos conocidos como neuronas que están conectadas y transmiten información entre sí. Estas señales se transmiten desde la entrada hasta generar una salida, que es el proceso conocido como forward propagation.

Las redes reciben una serie de valores de entrada y cada una de estas entradas llega a una neurona. Las neuronas de la red están a su vez agrupadas en capas que forman la red. Cada una de las neuronas de la red posee un peso que es un valor numérico. Los nuevos valores obtenidos salen de las neuronas y continúan su camino por la red.
Una vez que se ha alcanzado el final de la red se obtiene una salida que será la predicción.

Para conseguir que una red neuronal realice las funciones deseadas, **es necesario entrenarla.** El entrenamiento de una red neuronal se realiza modificando los pesos de sus neuronas. Para ello lo que se hace es introducir datos de entrenamiento en la red, en función del resultado que se obtenga, se modifican los pesos según el error obtenido. Este método se conoce como Backpropagation y se consigue que la red neuronal aprenda.

Funciones de pérdida

Una función de pérdida J(x) mide que tan insatisfechos estamos con las predicciones de nuestro modelo con respecto a una respuesta correcta y utilizando ciertos valores de θ.
Las máquinas aprenden mediante una función de pérdida. Es un método para evaluar qué tan bien un algoritmo específico modela los datos otorgados.

Si las predicciones se desvían demasiado de los resultados reales, la función de pérdida en Machine Learning arrojaría un número muy grande. Poco a poco, con la ayuda de alguna función de optimización, la función de pérdida en Machine Learning aprende a reducir el error en la predicción.

No existe una función de pérdida para todos los algoritmos en Machine Learning. La elección de una función de pérdida para un problema específico, como el tipo de algoritmo de Machine Learning elegido

En general, las funciones de pérdida pueden clasificarse en dos categorías principales dependiendo del tipo de tarea de aprendizaje con la que estamos tratando:

Pérdidas por regresión.

- Error cuadrático medio. Se mide como el promedio de la diferencia al cuadrado entre las predicciones y las observaciones reales.
- Error Absoluto Medio. Se mide como el promedio de la suma de las diferencias absolutas entre las predicciones y las observaciones reales.

Pérdidas por clasificación.

- Cross Entropy Loss. Se utiliza al ajustar los pesos del modelo durante el entrenamiento. El objetivo es minimizar la pérdida, es decir, cuanto menor sea la pérdida, mejor será el modelo. Un modelo perfecto tiene una pérdida de entropía cruzada de 0.

*** Diferencia entre función de optimización y de pérdida.** Cuando se trabaja en un problema de aprendizaje automático o de aprendizaje profundo, las funciones de pérdida se utilizan para optimizar el modelo durante el entrenamiento. El objetivo casi siempre es minimizar la función de pérdida. Cuanto menor sea la pérdida, mejor será el modelo.

Aprendizaje supervisado

Los algoritmos de aprendizaje supervisado basan su aprendizaje en un juego de datos de entrenamiento previamente etiquetados. Por etiquetado entendemos que para cada instancia conocemos el valor de su atributo que intentamos conocer con nuestro modelo. Esto le permitirá al algoritmo poder "aprender" una función capaz de predecir para datos nuevos. Las dos grandes familias de algoritmos supervisados son:

- Los algoritmos de regresión
- Los algoritmos de clasificación

Aprendizaje no supervisado

Los métodos no supervisados son algoritmos que basan su proceso de entrenamiento en un juego de datos sin etiquetas previamente definidas.. El aprendizaje no supervisado está dedicado a las tareas de agrupamiento, también llamadas clustering o segmentación, donde su objetivo es encontrar grupos similares en el conjunto de datos.

Regresión

Cuando usamos regresión, el resultado es un número. Es decir, el resultado de la técnica de machine learning que estemos usando será un valor numérico continuo, dentro de un conjunto de posibles resultados. Por ejemplo, predecir por cuánto se va a vender una casa.

Clasificación

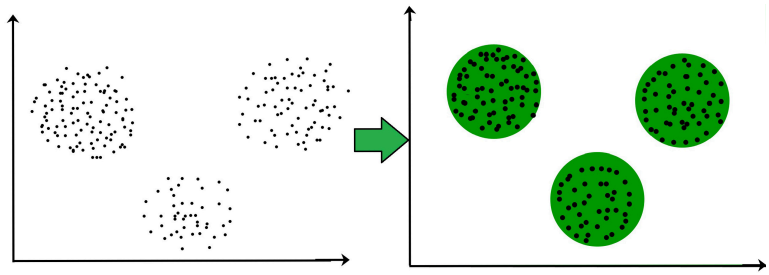
Cuando usamos clasificación, el resultado es una clase, entre un número limitado de clases. Por ejemplo, si queremos detectar si un correo es spam o no, sólo hay 2 clases.

Clustering

El clustering es una tarea que tiene como finalidad principal lograr el agrupamiento de conjuntos de objetos no etiquetados, para lograr construir subconjuntos de datos conocidos como clusters.

Cada cluster dentro de un grafo está formado por una colección de objetos o datos que a términos de análisis resultan similares entre si, pero que poseen elementos diferenciales con respecto a otros objetos pertenecientes al conjunto de datos y que pueden conformar un cluster independiente.

El algoritmo de agrupamiento más popular: K-Means.



Anomalía

La detección de anomalías es una tecnología que utiliza Inteligencia Artificial para identificar patrones anormales dentro de un conjunto de datos. Los sistemas definen la detección de anomalías como "un método utilizado para identificar patrones irregulares o inusuales en un entorno complejo".

Elegir si algo es normal o anormal es un problema de clasificación, que generalmente se resuelve mediante el aprendizaje supervisado con una combinación de datos etiquetados en las dos clases. Así, si la clase normal es perros, puede identificar cuatro patas y pelaje como características comunes. En ese caso, cuando el modelo de identificación de perros vea un auto, sabrá que es una anomalía.

La detección de anomalías se aplica en campos de seguridad para conocer violaciones de seguridad de la red. En el sector bancario sirve para identificar información que sea susceptible a fraude como transacciones sospechosas.

Overfitting

Cuando desarrollamos un modelo de machine learning, intentamos enseñarle cómo lograr un objetivo. Para ello partimos de una base de datos que será utilizada para entrenar el modelo. Sin embargo, si no hacemos las cosas correctamente, es posible que el modelo considere como validado, solo los datos que se han usado para entrenar el modelo, sin reconocer ningún otro dato que sea un poco diferente a la base de datos inicial.

Se dice que un modelo estadístico está sobreajustado cuando lo entrenamos con muchos datos. Cuando un modelo se entrena con tantos datos, comienza a aprender del ruido Entonces, el modelo no categoriza los datos correctamente, a este proceso se le conoce como overfitting.

Algunas de las formas de evitar el overfitting son:

- El entrenamiento con más datos
- Detención anticipada. La detención anticipada significa detener el proceso de entrenamiento antes de que el modelo pase el punto donde el modelo comienza a sobreajustarse a los datos de entrenamiento.
- Data augmentation. *Es* la generación artificial de datos por medio de perturbaciones en los datos originales. Esto nos permite aumentar tanto en tamaño como en diversidad nuestro set de datos de entrenamiento.
- Dropout en redes neuronales. La técnica Dropout consiste en remover aleatoria y temporalmente unidades (neuronas) de las capas internas y ocultas de la red neuronal.

Underfitting

Se refiere al escenario en el que un modelo de aprendizaje automático no puede generalizarse o encajar bien en un conjunto de datos.

Una clara señal de sobreajuste del aprendizaje automático es si tu error en el conjunto de datos de prueba o validación es mucho mayor que el error en el conjunto de datos de entrenamiento.

Conjunto de entrenamiento - Conjunto de validación - Conjunto de test

Básicamente dividimos nuestros datos en los 3 conjuntos que mencionas:

- Entrenamiento. Estos son los datos con los que se construye el modelo.
- Validación. Es una porción de datos que se usa para validar el modelo (prevenir sobre o infra- ajuste).
- Prueba. Es una última porción que se mantiene aparte y sobre la cual se evalúa el modelo. Usualmente se reporta la eficacia del modelo según los resultados.

Los porcentajes suelen variar. Los datos de entrenamiento suelen ser la mayoría (50% hasta un 80%). Los datos de validación y evaluación se suelen mantener en igual proporción (25–25%, 15%-15% o 10–10%).

TECNOLOGÍAS

- Jupyterlab
- Pandas
- Numpy
- Scikit-Learn
- Tensorflow
- Matplotlib
- Visión artificial
- NLP

TÉCNICAS

MLP

El perceptrón multicapa se trata de redes de tipo feed-forward, con neuronas de tipo perceptrón. La función de agregación es una suma ponderada, y la función de activación suele ser una función sigmoide o relu.

El principal problema por el que casi acaba desapareciendo el perceptrón es por el problema de linealidad de las entradas. El percetrón multicapa evoluciona y para ello incorpora capas de neuronas ocultas, con esto consigue representar funciones no lineales.

El perceptrón multicapa está compuesto por una capa de entrada, una capa de salida y n capas ocultas entremedias (hidden layers).

En el perceptrón multicapa se pueden diferenciar 2 fases:

1. Forward propagation en la que se calcula el resultado de salida de la red desde los valores de entrada hacia delante.
2. Aprendizaje en la que los errores obtenidos se van propagando hacia atrás (backpropagation) con el objetivo de modificar los pesos para que el valor estimado de la red se asemeje cada vez más al real, esta aproximación se realiza mediante la función gradiente del error.

Redes neuronales recurrentes

Son una clase de redes para analizar datos de series temporales permitiendo tratar la dimensión de "tiempo", que hasta ese momento no se había considerado con las redes neuronales.

Una red RNN incluye conexiones que apuntan "hacia atrás", una especie de retroalimentaciones entre las neuronas dentro de las capas.

Dado que la salida de una neurona recurrente en un instante de tiempo determinado es una función de entradas de los instantes de tiempo anteriores, se podría decir que una neurona recurrente tiene en cierta forma memoria.

Redes neuronales convolucionales

Las Redes neuronales convolucionales son un tipo de redes neuronales artificiales donde las "neuronas" corresponden a campos receptivos de una manera muy similar a las neuronas del cerebro humano. Este tipo de red es una variación de el perceptrón multicapa. Son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes.

Estas redes procesan sus capas imitando al cortex visual del ojo humano para identificar distintas características en las entradas Para ello, contiene varias capas ocultas especializadas y con una jerarquía: esto quiere decir que las primeras capas pueden detectar líneas, curvas y se van especializando hasta llegar a capas más profundas que reconocen formas complejas como un rostro o la silueta de un animal.

Para comenzar, la red toma como entrada los pixeles de una imagen. Luego se harán las llamadas "convoluciones". Estas consisten en tomar "grupos de pixeles cercanos" de la imagen de entrada e ir operando matemáticamente contra una pequeña matriz que se llama kernel.

Ese kernel genera una nueva matriz de salida, que en definitiva será nuestra nueva capa de neuronas ocultas. A medida que vamos desplazando el kernel, vamos obteniendo una "nueva imagen" filtrada. Estas imágenes nuevas lo que están "dibujando" son ciertas características de la imagen original. Esto ayudará en el futuro a poder distinguir un objeto de otro.

Autoencoders

Es una red neuronal que toma un cierto tipo de información, la comprime primero y la descomprime después. Utiliza para ello tres componentes: codificación, descodificación y función de distancia entre la información comprimida y descomprimida.

El objetivo es generar nuevos datos primero comprimiendo la entrada en un espacio de variables latentes y luego reconstruyendo la salida en base a la información adquirida.

Redes GAN

Las Redes Generativas Antagónicas, son una poderosa clase de redes neuronales. Las GAN consisten esencialmente en un algoritmo basado en un sistema de dos redes neuronales, el generador y el discriminador que compiten entre sí.

Dada una serie de muestras objetivo, el generador intenta producir muestras que puedan engañar al discriminador para que éste crea que son reales. El discriminador intenta resolver las muestras reales. Usando este enfoque de competición iterativa, eventualmente terminamos con un generador que es realmente bueno generando muestras similares a las muestras reales y por lo tanto es capaz de engañar al discriminador.

KNN (*k* vecinos más cercanos)

El KNN es un algoritmo de aprendizaje supervisado, es decir, que a partir de un juego de datos inicial su objetivo será el de clasificar correctamente todas las instancias nuevas.

El algoritmo clasifica cada dato nuevo en el grupo que corresponda, según tenga *k* vecinos más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al que pertenecer. Este grupo será, por tanto, el de mayor frecuencia con menores distancias.

Árboles de decisión

Los árboles de decisión se emplean en diversos ámbitos para analizar situaciones complejas con múltiples posibilidades de decisión y escoger la mejor. Se utilizan en modelos predictivos de aprendizaje supervisado.

Un árbol de decisión es una técnica que emplea algoritmos en forma de árbol para enseñar a las máquinas a tomar decisiones y, por tanto, a resolver problemas de regresión o de clasificación. Como resultado, obtenemos modelos predictivos precisos y fiables.

La máquina se somete a un entrenamiento previo. Ese entrenamiento le sirve como base para su futura toma de decisiones. Una vez entrenada, cuando se encuentra ante una disyuntiva, utiliza el conjunto de datos con el que ha sido alimentada para plantear correlaciones entre los datos de su entrenamiento y los que tiene ante sí y decantarse por la opción correcta.

Por tanto, en este contexto, el árbol de decisión se convierte en un modelo predictivo.

SVM

Las máquinas de vectores de soporte son una técnica que encuentra la mejor separación posible entre clases. Con dos dimensiones es fácil entender lo que está haciendo.
La línea que mejor distingue la zona de los puntos azules de la zona de los puntos rojos es la línea que maximiza el margen entre ambos.

Normalmente, los problemas de aprendizaje automático tienen muchísimas dimensiones. Así que, en vez de encontrar la línea óptima, el SVM encuentra el hiperplano que maximiza el margen de separación entre clases.

Hay veces en las que no hay forma de encontrar una hiperplano que permita separar dos clases. En estos casos decimos que las clases no son linealmente separables. Para resolver este problema podemos usar el truco del kernel queconsiste en inventar una dimensión nueva.

Regresión lineal

La regresión lineal es un algoritmo de aprendizaje supervisado. En su versión más sencilla, lo que haremos es "dibujar una recta" que *nos indicará la tendencia* de un conjunto de datos continuos (si fueran discretos, utilizaríamos Regresión Logística).

Los algoritmos aprenden por sí mismos y en este caso a obtener automáticamente esa "recta" que buscamos. Para hacerlo se mide el error con respecto a los puntos de entrada y el valor "Y" de salida real. El algoritmo deberá minimizar el coste de una función de error cuadrático y esos coeficientes corresponderán con la recta óptima.

Regresión logística

Hay muchos algoritmos de clasificación, pero la regresión logística es común y útil para resolver problemas de clasificación binaria. Es decir, un método estadístico para predecir clases binarias.

Describe y estima la relación entre una variable binaria dependiente y las variables independientes. Es también llamada función sigmoide ya que puede tomar cualquier número de valor real y asignar a un valor entre 0 y 1.

K-Means

K-means es un algoritmo de clasificación no supervisada que agrupa objetos en *k* grupos basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto y el centroide de su grupo o cluster. Se suele usar la distancia cuadrática.

El algoritmo consta de tres pasos:

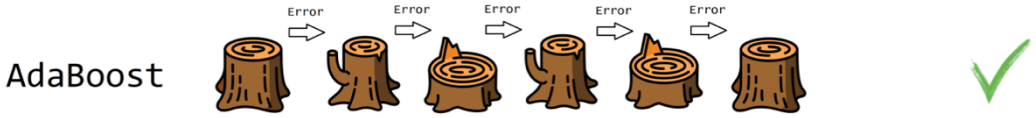
1. Inicialización: una vez escogido el número de grupos, *k*, se establecen *k* centroides aleatoriamente.
2. Asignación objetos a los centroides: cada objeto de los datos es asignado a su centroide más cercano.
3. Actualización centroides: se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo.

Se repiten los pasos 2 y 3 hasta que los centroides no se mueven, o se mueven por debajo de una distancia umbral en cada paso.

Adaboost

Es un tipo de algoritmo de boosting. Este es un enfoque basado en la idea de crear una regla de predicción altamente precisa combinando muchas reglas relativamente débiles e imprecisas.

Adaboost es un algoritmo que puede ser utilizado junto con otros *algoritmos de aprendizaje*. Funciona eligiendo un *algoritmo base* (por ejemplo, árboles de decisión) y mejorándolo iterativamente al tomar en cuenta los casos incorrectamente clasificados en el conjunto de entrenamiento.



XGBoost (eXtreme Gradient Boosting)

Se impulso con el objetivo principal de reducir el sesgo y la varianza. El objetivo es crear árboles débiles de forma secuencial para que cada árbol nuevo se centre en la debilidad (datos mal clasificados) del anterior. Después de que se agrega un árbol débil, las ponderaciones de los datos se reajustan, lo que se conoce como "reponderación". El conjunto forma un modelo sólido después de la convergencia debido a la autocorrección después de cada nuevo árbol agregado.

La fuerza de XGBoost es el paralelismo y la optimización del hardware. En el mundo de las competiciones de Kaggle el algoritmo XGBoost tiene el primer lugar.

Referencias bibliográficas

- Plantilla TFG Grado de Ingeniería Informática Universidad de León
- Apuntes asignatura Introducción a la Inteligencia Artificial
- https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/
- https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/
- https://aprendeia.com/aprendizaje-supervisado-logistic-regression/
- https://www.iartificial.net/maquinas-de-vectores-de-soporte-svm/
- https://www.aprendemachinelearning.com/tag/regresion-lineal/
- https://ichi.pro/es/que-es-xgboost-y-como-optimizarlo-232831486673332
- https://relopezbriega.github.io/blog/2017/06/10/boosting-en-machine-learning-con-python/
- https://agenciab12.com/noticia/que-son-arboles-de-decision-inteligencia-artificial
- https://www.merkleinc.com/es/es/blog/algoritmo-knn-modelado-datos
- https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/
- https://torres.ai/redes-neuronales-recurrentes/
- https://www.diegocalvo.es/red-neuronal-recurrente/
- https://interactivechaos.com/es/manual/tutorial-de-deep-learning/el-perceptron-multicapa
- https://protecciondatos-lopd.com/empresas/underfitting/
- https://www.iartificial.net/clasificacion-o-regresion/#Clasificacion