

Método de la ingeniería

Paso 1. Identificación del problema

1.1 El problema:

Se necesita un software capaz de identificar números de un dígito escritos a mano, para lo cual debe entrenarse utilizando el [dataset de MNIST](#). El programa debe ser capaz de leer, interpretar y mostrar los números del dataset.

1.2 Exploración:

Para permitir al usuario explorar el dataset, se debe disponer de una ventana que muestre un número del dataset y permita al usuario ir al anterior o siguiente utilizando botones. Adicionalmente, dicha ventana debe contener un recuadro de texto que permita al usuario digitar un número; una vez digitado el número, el programa debe mostrar una de las instancias de dicho número en el dataset.

1.3 Uso:

Como funcionalidad principal, se le debe permitir al usuario un recuadro para dibujar un número. Una vez dibujado, el programa debe determinar cuál de los números del 0 al 9 es. Además, luego de determinar el número, es pertinente preguntar al usuario si la clasificación fue correcta o no. El programa deberá guardar la imagen dibujada por el usuario junto con una etiqueta para identificar el número. Dicha etiqueta es también asignada por el usuario.

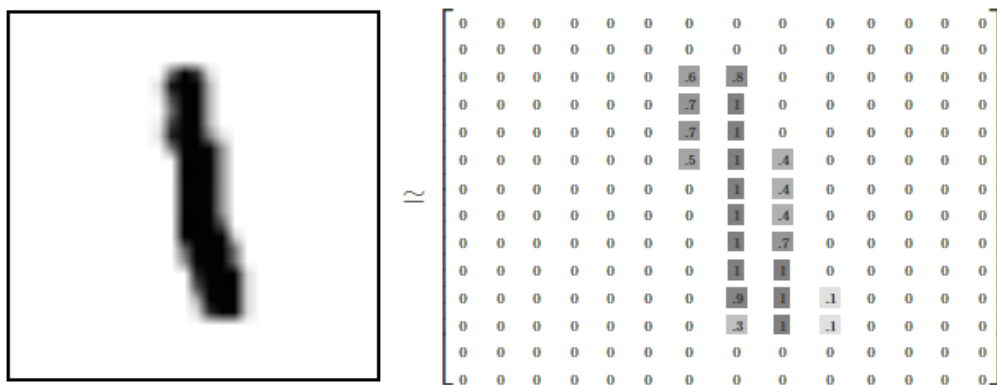
Paso 2. Recolección de información

2.1 Cómo interpretar el dataset de MNIST:

El conjunto de datos MNIST consiste de imágenes de 28x28 píxeles, cada una representando un entero en el intervalo $[0, 9]$. Por ejemplo:



Cada una de estas imágenes puede representarse como una matriz que describe que tan oscuro es un pixel. Una instancia del número uno podría ser:



Como cada imagen tiene 28x28 pixeles, tenemos una matriz con 784 elementos. Si *convertimos* dichas matriz en un vector, tendríamos un vector del mismo número de elementos. Entonces, podemos pensar en el dataset como un conjunto de 784 dimensiones, donde cada elemento es un número del cero al uno que representa que tan oscuro es el pixel correspondiente.

2.1.2 Representación en CSV:

El conjunto de datos original está compuesto por una serie de archivos “.ubyte”, que son legibles por las diferentes librerías de MNIST. Con el objetivo de usar la menor cantidad de librerías posibles, se utilizará una representación del dataset en formato CSV encontrada en [Kaggle](#).

Paso 3. Propuesta de soluciones

NOTA: Este apartado se concentrará únicamente en el problema de entrenar el software para reconocer números escritos a mano.

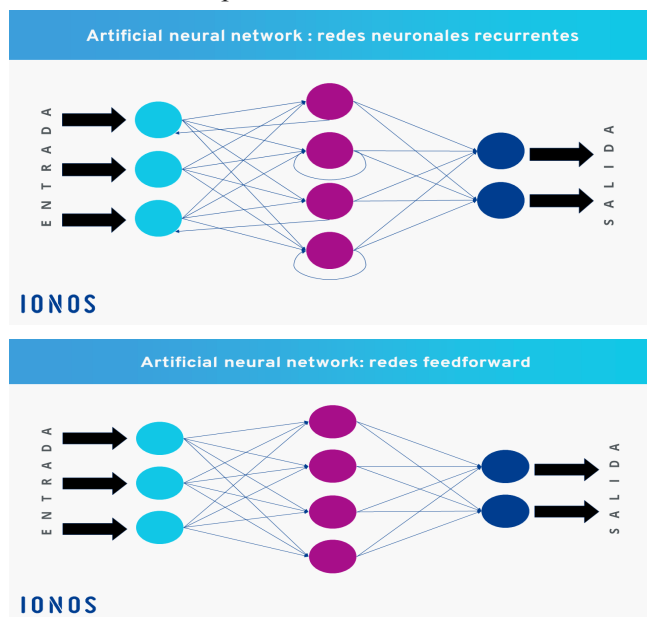
3.1 Red Neuronal Artificial:

Las redes neuronales artificiales (también conocidas como sistemas conexionistas) son un modelo computacional el que fue evolucionando a partir de diversas aportaciones científicas que están registradas en la historia. Consiste en un conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitir señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo valores de salida.¹

Cada neurona está conectada con otras a través de unos enlaces. En estos enlaces el valor de salida de la neurona anterior es multiplicado por un valor de peso. Estos pesos en los enlaces pueden incrementar o inhibir el estado de activación de las neuronas adyacentes. Del mismo modo, a la salida de la neurona, puede existir una función limitadora o umbral, que modifica el valor resultado o impone un límite que no se debe sobrepasar antes de propagarse a otra neurona. Esta función se conoce como función de activación.

Estos sistemas aprenden y se forman a sí mismos, en lugar de ser programados de forma explícita, y sobresalen en áreas donde la detección de soluciones o características es difícil de expresar con la programación convencional. Para realizar este aprendizaje automático, normalmente, se intenta minimizar una función de pérdida que evalúa la red en su totalidad. Los valores de los pesos de las neuronas se van actualizando buscando reducir el valor de la función de pérdida. Este proceso se realiza mediante la propagación hacia atrás.

El objetivo de la red neuronal es resolver los problemas de la misma manera que el cerebro humano, aunque las redes neuronales son más abstractas. Las redes neuronales actuales suelen contener desde unos miles a unos pocos millones de unidades neuronales.



¹ "Red neuronal artificial - Wikipedia, la enciclopedia libre."
https://es.wikipedia.org/wiki/Red_neuronal_artificial. Fecha de acceso 10 abr.. 2021.

Las tareas se aplican a las redes neuronales artificiales tienden a caer dentro de las siguientes categorías generales:

Aproximación de funciones, o el análisis de regresión, incluyendo la predicción de series temporales, funciones de aptitud y el modelado.

Clasificación, incluyendo el reconocimiento de patrones y la secuencia de reconocimiento, detección y de la toma de decisiones secuenciales.

Procesamiento de datos, incluyendo el filtrado, el agrupamiento, la separación ciega de las señales y compresión.

- Robótica, incluyendo la dirección de manipuladores y prótesis.
- Ingeniería de control, incluyendo control numérico por computadora.
- Las áreas de aplicación incluyen la identificación de sistemas y el control (control del vehículo, predicción de trayectorias, el control de procesos, manejo de recursos naturales), la química cuántica, juegos y la toma de decisiones (backgammon, ajedrez, póquer), el reconocimiento de patrones (sistemas radar, reconocimiento facial, clasificación de señales, reconocimiento de objetos y más), de reconocimiento de secuencia (gesto, voz, reconocimiento de texto escrito a mano), diagnóstico médico, aplicaciones financieras (por ejemplo, sistemas automatizados de comercio (trading algorítmico)), minería de datos (o descubrimiento de conocimiento en bases de datos, "KDD"), la visualización, traducción automática, diferenciando entre informes deseados y no deseados en redes sociales, prevención de spam (correo basura) de correo electrónico.

Las redes neuronales artificiales se han utilizado también para el diagnóstico de varios tipos de cáncer. Un sistema de detección de cáncer de pulmón híbrido basado ANN llamado HLND mejora la precisión del diagnóstico y la velocidad de la radiología cáncer de pulmón. Estas redes también se han utilizado para diagnosticar el cáncer de próstata. Los diagnósticos se pueden utilizar para hacer modelos específicos tomados de un gran grupo de pacientes en comparación con la información de un paciente dado. Los modelos no dependen de suposiciones acerca de las correlaciones de diferentes variables. El cáncer color rectal también se ha previsto el uso de las redes neuronales. Las redes neuronales podrían predecir el resultado de un paciente con cáncer color rectal con más precisión que los métodos clínicos actuales. Después del entrenamiento, las redes podrían predecir múltiples resultados de los pacientes de instituciones relacionadas.

Paso 4. Búsqueda de soluciones creativas y paso 5 selección de la mejor solución:

NOTA: Este problema ya tiene una solución definida por lo que no es necesario buscar más soluciones aunque aquí haremos unas menciones honoríficas a otras soluciones que podrían haber sido seleccionadas para solucionar este problema.

Clasificación Lineal:

En el campo del aprendizaje automático, el objetivo de la clasificación estadística es utilizar las características de un objeto para identificar a qué clase (o grupo) pertenece. Un clasificador lineal lo consigue tomando una decisión de clasificación basada en el valor de una combinación lineal de las características. Las características de un objeto también se conocen como valores de características y suelen presentarse a la máquina en un vector llamado vector de características. Estos clasificadores funcionan bien para problemas prácticos como la clasificación de documentos y, en general, para problemas con muchas variables (características), alcanzando niveles de precisión comparables a los clasificadores no lineales y tardando menos tiempo en entrenarse y utilizarse.

k-nearest neighbors algorithm

En estadística, el algoritmo de k-próximos (k-NN) es un método de clasificación no paramétrico desarrollado por primera vez por Evelyn Fix y Joseph Hodges en 1951, y posteriormente ampliado por Thomas Cover. Se utiliza para la clasificación y la regresión. En ambos casos, la entrada consiste en los ejemplos de entrenamiento más cercanos del conjunto de datos. La salida depende de si k-NN se utiliza para la clasificación o la regresión:

- En la clasificación k-NN, la salida es una pertenencia a una clase. Un objeto se clasifica mediante un voto plural de sus vecinos, asignando el objeto a la clase más común entre sus vecinos más cercanos (k es un número entero positivo, normalmente pequeño). Si $k = 1$, el objeto se asigna simplemente a la clase de ese único vecino más cercano.
- En la regresión k-NN, el resultado es el valor de la propiedad del objeto. Este valor es la media de los valores de los k vecinos más cercanos.

k-NN es un tipo de clasificación en el que la función sólo se aproxima localmente y todo el cálculo se aplaza hasta la evaluación de la función. Dado que este algoritmo se basa en la distancia para la clasificación, si las características representan diferentes unidades físicas o vienen en escalas muy diferentes, entonces la normalización de los datos de entrenamiento puede mejorar su precisión de forma espectacular.

Tanto para la clasificación como para la regresión, una técnica útil puede ser asignar pesos a las contribuciones de los vecinos, de forma que los vecinos más cercanos contribuyan más a la media que los más lejanos. Por ejemplo, un esquema de ponderación común consiste en dar a cada vecino un peso de $1/d$, donde d es la distancia al vecino.

Los vecinos se toman de un conjunto de objetos de los que se conoce la clase (para la clasificación k-NN) o el valor de la propiedad del objeto (para la regresión k-NN). Esto puede considerarse como el conjunto de entrenamiento para el algoritmo, aunque no se requiere un paso de entrenamiento explícito.

Gradient boosting

es una técnica de aprendizaje automático para problemas de regresión y clasificación, que produce un modelo de predicción en forma de un conjunto de modelos de predicción débiles, típicamente árboles de decisión. Cuando un árbol de decisión es el aprendiz débil, el algoritmo resultante se denomina árboles de refuerzo de gradiente, que suele superar a los bosques aleatorios. Construye el modelo por etapas, como hacen otros métodos de refuerzo, y los generaliza al permitir la optimización de una función de pérdida diferenciable arbitraria.