

Redes neuronales

Una red neuronal es un conjunto de algoritmos y objetos que se puede usar para identificar relaciones entre variables de un set de datos. Una vez que esta red tiene identificadas de manera adecuada estas relaciones se puede usar para estudiar o predecir el resultado de otros sets de datos, de un mismo tipo.

La estructura del tipo de red con el que se trabajó para este programa es la siguiente. Las neuronas de una dada capa estará conectadas con todas las de la capa siguiente y anterior. Se adjunta una ilustración de esto.

Una “neurona” aquí es una variable que guardará información, en nuestro caso, un número de punto flotante. Cada neurona tiene “inputs” que pueden ser datos externos o el output de otras neuronas. El output de la última capa de neuronas es el resultado final de la red. En nuestro caso un único número entre 0 y 1 que representará la fracción de votos obtenida por un candidato.

Para conocer la activación de una neurona de una dada capa se hace la suma pesada de los outputs de todas las de la capa anterior y se le agrega el bias de la propia neurona. Luego esta activación alimenta una función de activación, qué es lo que da el output de la neurona. En nuestro caso, se trabajó con neuronas cuya función de activación era la sigmoide.

Algoritmo Genético

El algoritmo genético es el método utilizado para entrenar a la red neuronal sobre el set de datos.

Este método consiste en crear “N” (la población) redes neuronales con sus parámetros (pesos y biases) inicializados al azar. Se calcula la función costo de todas las redes de la población sobre el set de datos de entrenamiento, Una vez hecho esto se elige a las dos redes neuronales con el menor costo como “padres”.

Se crea una nueva generación de N redes de la siguiente manera:

- para settear un dado peso o bias se elige el padre del que se lo hereda con una probabilidad de 50%.
- con una probabilidad baja (en nuestro caso 0.1%) dicho peso o bias se cambia por un valor al azar

El proceso enunciado en los dos párrafos anteriores se repite hasta que se llegue a un número máximo de iteraciones o se llegue a un costo que se haya juzgado suficientemente bajo. Estos valores de corte se definen en tiempo de

compilación y en nuestro caso son 20.000 iteraciones y 0.012 para la función costo (que por como es la base de datos equivale a un error medio de 3 puntos en la estimación de los porcentajes de cada candidato).

Como resultado se obtiene una red neuronal que puede incorporar los datos de la economía y otras variables del candidato además de su promedio en encuestas. Esto permite obtener una mejor estimación de cuál sería el resultado de una elección que si solo se considerara su promedio en la encuestas.

Uso del programa

El programa tiene como objetivo predecir los porcentajes obtenidos por un grupo de candidatos que compitan en una misma elección en base a su promedio en las encuestas y el estado de la economía del país.

Al ejecutarse el programa el usuario tiene dos opciones:

- cargar los pesos y bias de una red neuronal ya entrenada: Si se elige esta opción el programa prosigue a la etapa siguiente,
- cargar una base de datos: Aquí el programa le pedirá al usuario ingresar una base de datos mediante dos archivos. Uno contendrá los datos de la economía del país y otro datos sobre los candidatos. La base de datos tiene que tener un formato definido, el cual se explicará más adelante.

Si se elige la segunda opción el programa cargará la base de datos sobre la que se entrenará la red neuronal. Para ello pedirá al usuario los path de los archivos de dicha base.

A continuación se entrenará a la red neuronal según el algoritmo genético definido en la sección anterior. Luego le preguntará al usuario si desea guardar los pesos y bias de la red. Esto se hace para no tener que entrenar una nueva red cada vez que se quiera ejecutar el programa.

En cualquier caso, a continuación, el programa ofrece al usuario la posibilidad de simular una elección. Si el usuario acepta hacer esto, el programa le pedirá que ingrese por pantalla los datos de la economía del país del año electoral y el anterior:

- desempleo: fracción de la población que no tiene trabajo (un número de 0 a 1),
- pobreza: fracción de la población que es pobre (un número de 0 a 1),
- crecimiento del PBI: un número mayor a -1,
- inflación: un número mayor a -1,

- número de candidatos.

A continuación le pedirá que ingrese por pantalla los datos de los candidatos:

- puntaje: un número entre 0 y 1 que representa su ideología (0 extrema izquierda, 1 extrema derecha),
- poder: un booleano que indica si el partido del candidato está en el gobierno o no,
- imagen: la suma de la imagen positiva y regular del candidato (un número entre 0 y 1),
- encuestas: el promedio en las encuestas del candidato (un número entre 0 y 1).

A continuación el programa arrojará los porcentajes que se espera que obtenga cada candidato en la elección haciendo la proyección de indecisos (normalizando los porcentajes a 100). Luego, le ofrecerá al usuario la posibilidad de guardar el resultado de la simulación. Si se elige guardarlo, el resultado se escribe a un archivo .txt.

Por último le preguntará al usuario si quiere simular otra elección. El programa continuará hasta que el usuario ya no desee realizar más simulaciones.

Base de datos

La base de datos con la que se trabaja para entrenar la red tiene el siguiente formato.

En un primer archivo figuran los datos de la economía del país del siguiente modo:

- año de la elección,
- desempleo el año electoral,
- desempleo el año anterior,
- pobreza el año electoral,
- pobreza el año anterior,
- variación del PBI el año electoral,

- variación del PBI el año anterior,
- inflación el año electoral,
- inflación el año anterior,
- número de candidatos que compiten en la elección.

El tipo de dato que es cada uno de estos parámetros es el mismo que cuando se solicita su ingreso por pantalla.

En otro archivo figuran los datos de los candidatos se la siguiente manera:

- año de la elección (se utiliza como comprobación de que ambos archivos se estén leyendo del modo correcto, si los años no coinciden hay un problema),
- puntaje ideológico el candidato,
- booleano que indica si su partido se encuentra o no en el poder,
- imagen positiva y regular del candidato,
- promedio en las encuestas,
- resultado que haya obtenido en la elección.

Figuran 2 elecciones para el año 2015 dado que en dicho año la elección se definió en una segunda vuelta.

Comentarios finales

Se adjuntan los siguientes archivos:

- Eleccion_Genetica.cpp : main del programa.
- Red_Genetica.h : declaración de la clase Neural_Network y sus funciones.
- Datos.h : estructuras de datos utilizadas y funciones para leer y escribir archivos .txt
- Data.txt : archivo con los datos de los candidatos,
- Data_econom.txt : archivo con los datos de la economía del país,
- Pesos_y_bias_con_2019.txt : archivo con los pesos y biases de una red neuronal ya entrenada.
- Pesos_y_bias_sin_2019 : red neuronal entrenada sin utilizar los datos correspondientes a 2019 (se uso para observar la predicción de una elección

con la que la red no se entrenó y de la cual se conocía el resultado).

- Resultados_2021.txt : resultados para las elecciones de 2021 de acuerdo a las proyecciones del Ministerio de Economía y los promedios en encuestas a octubre del 2020. Obtenidos utilizando la red neuronal entrenada sobre la base de datos completa.
- Resultados_2019_21.txt : resultados para las elecciones de 2019 y 2021. Obtenidos utilizando la red entrenada sin los datos correspondientes a 2019.
- Error.txt : archivo que contiene el valor de la función costo para cada iteración. Para la red entrenada sobre la base de datos completa.
- Costo_vs_iteracion.png : gráfico en base al archivo Error.txt