**Redes Neuronales Básicas** Parte 1 – Python y Numpy

**Oscar Fernando López Barrios - Carné 20679**

**Alejandro José Gómez Hernández - Carné 20347**

En clase vimos un modelo simple, utilizando solo Python y Numpy, para resolver Regresiones Lineales mediante redes neuronales. Utilizando el código desarrollado (o si lo desea uno propio), responda a las siguientes preguntas:

1. Cambie el número de observaciones a 100,000. Explique qué es lo que ocurre en términos de:
   1. El tiempo de ejecución para resolver el problema

**R//** En este caso el tiempo de ejecución aumenta de mayor manera debido a que al momento de realizar tantas observaciones más, los cálculos que se deben de realizar también aumentan. Por lo tanto, el tiempo que se ejecuta para resolver problemas es mayor.

* 1. El resultado final vrs lo encontrado en clase: es igual, o diferente. ¿Por qué?

**R//** En general el resultado final se diferencia al obtenido durante la clase debido a que se cuenta con una mayor cantidad de tiempo de ejecución, además que se puede notar que cuando se realiza esta nueva prueba con la cantidad definida la pérdida aumenta con respecto a la cantidad de las 1000 observaciones que se tenía antes en clase.

* 1. Las gráficas para representar los datos/resultados

**R//** Se puede observar a comparación de las gráficas anteriores que existen diferencias con respecto a como se mira la cantidad de datos que existen dispersos por toda la gráfica, en general se puede ver una cantidad de datos mayormente juntos y la gráfica se puede ver totalmente lineal.

1. Cambie el número de observaciones a 1,000,000. Explique qué es lo que ocurre en términos de:
   1. El tiempo de ejecución para resolver el problema

**R//** Así como pasó con el ajuste anterior el tiempo de ejecución fue mucho mayor al que brindaba la ejecución en la que se realizaron 1000 observaciones. En este caso realmente hay una gran diferencia debido a la gran cantidad de cálculos que se deben de realizar.

* 1. El resultado final vs lo encontrado en clase: es igual, o diferente. ¿Por qué?

**R//** En general el resultado es distinto a lo que se logró encontrar durante la clase, debido a que 100000 de iteraciones brindan resultados más precisos, esto debido a que la cantidad de iteraciones brinda resultados más acertados, principalmente porque en varios casos existe la forma de que a más cantidad de iteraciones se logra un resultado más preciso.

Además, se puede ver la diferencia que existe entre los pesos y los sesgos que se tienen:

1000 Observaciones

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

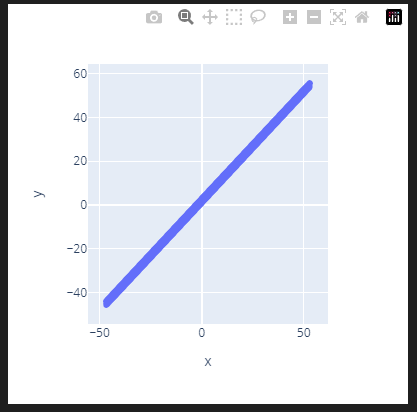
1000000 Observaciones

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

* 1. Las gráficas para representar los datos/resultados

**R//** En este caso la gráfica que brinda al momento de representar los resultados se puede ver de manera más clara la forma lineal de esta, además se puede ver como existe mayor densidad en los datos que contiene esta gráfica.



1. “Juegue” un poco con el valor de la tasa de aprendizaje, por ejemplo 0.0001, 0.001, 0.1, 1.

Para cada uno de estos indique:

* 1. ¿Qué ocurre con el tiempo de ejecución?

**R//** Al momento de realizar estos cambios se puede visualizar por medio de la ejecución de tasas de aprendizaje más pequeñas el tiempo aumenta, esto debido a que al momento de tener una tasa de aprendizaje de menor tamaño las iteraciones que se deben de realizar para que esta logre el objetivo son mayores.

* 1. ¿Qué ocurre con la minimización de la pérdida?

**R//** La minimización de la pérdida realmente se ve afectada por los cambios en la tasa de aprendizaje, debido a que cuando se utilizan cantidades menores la tasa de pérdida es menor y cuando se utilizan tasa de aprendizaje mayor la tasa de pérdida es mayor.

* 1. ¿Qué ocurre con los pesos y los sesgos?

**R//** En el caso de los pesos y los sesgos se puede conocer que dependiendo de la cantidad de la tasa de aprendizaje que se tenga, puede ser mayor la divergencia entre los valores de estos. En el caso de una tasa de aprendizaje mayor puede llevar a valores fuera del rango que se espera.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* 1. ¿Qué ocurre con las iteraciones?

**R//** En el caso de las iteraciones, un número de tasa de aprendizaje menor hace que la cantidad de iteraciones deba de ser mayor para lograr un resultado óptimo con respecto a la forma en la que funciona el modelo que se está analizando.

* 1. ¿El problema queda resuelto o no?

**R//** La resolución del problema depende del número de la tasa de aprendizaje que se elija, esto debido a que dependiendo de la tasa de aprendizaje se puede obtener resultados, pero en algunos casos dependiendo de esta, los resultados son más precisos y mejores.

* 1. ¿Cuál es la apariencia de la última gráfica? ¿Se cumple con la condición de que sea de 45 grados?

**R//** Por medio de la ejecución se puede visualizar que la gráfica realmente si cumple con la condición de que sea de 45 grados.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

1. Cambie la función de pérdida “L2-norm” a la misma pero sin dividir por 2. Explique lo que ocurre en términos de:
   1. El tiempo que se tarda el algoritmo en terminar, comparado a lo que vimos en clase

**R//** El tiempo en el que se ejecuta realmente cambia a un tiempo un poco menor, debido a que existen menos cálculos que se deben de realizar la momento de lograr encontrar los valores de la pérdida.

* 1. Si la pérdida se minimiza igual que lo que vimos en clase

**R//** En este caso sucede lo contrario, la pérdida que se tiene realmente aumenta y se puede notar por medio de los valores que brinda el modelo.

Texto

Descripción generada automáticamente

* 1. Si los pesos y sesgos son parecidos a los vistos en clase

**R//** En este caso los pesos son un poco distintos a los que se vieron en la clase, pero en este caso se puede visualizar mediante los resultados que los pesos y sesgos brindan mejores resultados a los que anteriormente se deberían de haber obtenido.

Una captura de pantalla de un celular con texto e imagen

Descripción generada automáticamente con confianza media

* 1. Si el problema se resuelve como ocurrió en clase

**R//** Realmente el problema se resuelve y brinda un resultado correcto, pero en este caso se puede observar como la cantidad de pérdida es mayor a la que se tenia al momento de realizar las pruebas durante la clase.

* 1. Si se obtiene un mejor resultado al hacer más iteraciones

**R//** Al momento de realizar un mayor número de iteraciones, estas logran que el modelo se ajuste de una manera mejor, pero esto no brinda la certeza de que el modelo esté hecho de manera correcta debido a que puede existir overfitting.

1. Cambie la función de pérdida de la “L2-norm” a la “L1-norm”. Explique lo que ocurre en términos de:
   1. El tiempo que se tarda el algoritmo en terminar, comparado a lo que vimos en clase

**R//**

* 1. Si la pérdida se minimiza igual que lo que vimos en clase
  2. Si los pesos y sesgos son parecidos a los vistos en clase
  3. Si el problema se resuelve como ocurrió en clase 5. Si se obtiene un mejor resultado al hacer más iteraciones

6. ¿Tendrá una de estas más limitaciones que la otra?

1. Cree una función f(x1,x2) = 13 \* x1 + 7 \* x2 - 12.
   1. ¿Funciona el algoritmo de la misma forma?

# Parte 2 – Tensorflow2

En clase vimos un modelo simple de una red neuronal utilizando TensorFlow 2. Utilizando el código desarrollado (o si lo desea uno propio pero que funcione correctamente), responda a las siguientes preguntas:

1. Cambie el número de observaciones a 100,000. ¿Qué ocurre?
2. “Juegue” un poco con la tasa de aprendizaje. Los valores como 0.0001, 0.001, 0.1, 1 son interesantes para observar ¿Qué diferencias se observan? ¿Se comporta bien el algoritmo?
3. Cambie la función de pérdida. Una función altenativa es la “Huber Loss”.

La función de pérdida Huber es más adecuada que la L2.norm cuando tenemos valores atípicos, ya que es menos sensitiva a los mismos (en nuestro ejemplo no tenemos valores atípicos, pero seguramente se topará con ellos en el futuro). La L2-norm eleva todas las diferencias al **cuadrado**, por lo que los valores atípicos tienen mucha influencia sobre los resultados. La sintáxis correcta de la función de pérdida Huber es “huber\_loss”.

¿Cómo se comparan los resultados al cambiar la función de pérdida?

Referencia:

https://www.tensorflow.org/versions/r1.15/api\_docs/python/tf/keras