Universidad de los Andes
Colombia | Facultad de Ingeniería

Universidad de los Andes
Maestría en Inteligencia Analítica para la Toma de Decisiones.
Modelos Avanzados para el Análisis De Datos 2.
Prof. Martín Andrade Restrepo.

Taller 4. Redes densamente conectadas para problemas de clasificación y de regresión

Observaciones:

En este taller usaremos redes densas (redes densamente conexas o redes vainilla) para nuevos problemas de clasificación y de regresión.

Parte I (50 ptos)

En este ejercicio utilizaremos la base de datos de IMDB (Internet Movie Database). Este dataset contiene 50 mil reseñas de películas divididas en un conjunto de entrenamiento y un conjunto de prueba (ambos con 25 mil reseñas). Cada conjunto está dividido respectivamente en reseñas positivas (50%) y reseñas negativas (50%).

Para cargar el dataset use:

```
from tensorflow.keras.datasets import imdb
(train_data, train_labels), (test_data, test_labels) = imdb.load_data(num_words
    =10000)
```

El argumento `num_words` determina el número de palabras del conjunto de entrenamiento (organizadas por frecuencia) que se van a utilizar para su modelo. En otras palabras, palabras usadas poco en las reseñas (más allá del puesto diez mil) no se utilizarán en su entrenamiento.

1. **(10 ptos)** Investigue qué representan las matrices/vectores cargados y sus posibles valores. Analice de forma preliminar sus datos (ocurrencias y distribuciones). Para esto puede ser útil extraer las palabras de los datos codificados de entrenamiento y prueba. Para esto puede usar el diccionario *word_index*:

```
word_index = imdb.get_word_index()
reverse_word_index = dict([(value, key) for (key, value) in word_index.items
    ()])
decoded_review = " ".join([reverse_word_index.get(i - 3, "?") for i in
    train_data[0]])
```

   Como podrá ver, este código devuelve las palabras de la primera reseña del conjunto de entrenamiento (`train_data[0]`) que estén dentro de las diez mil más frecuentes.

2. **(15 ptos)** Prepare sus datos para que puedan ser recibidos por una red neuronal (como usted habrá visto, los vectores del conjunto de entrenamiento y del conjunto de prueba tienen longitudes variadas). Para esto puede utilizar el multi-hot-encoding. Investigue y describa qué es el multi-hot-encoding. Luego, implemente una función que codifique sus datos utilizando este método (en este caso debe crear un vector de longitud 10000 que tenga 1s en las posiciones de las palabras que aparecen en la respectiva reseña).

3. **(15 ptos)** Utilizando `Keras`, implemente diferentes tipos de redes con diferentes activaciones y arquitecturas pero sin ningún método de regularización. Experimente también con los optimizadores más frecuentes. Tenga en cuenta que la activación de la última capa debe ser sigmoide o softmax, al ser un problema de clasificación binaria (recuerde que cada uno tiene su respectiva función de costo apropiada). Describa sus resultados.

   Sugerencia: Como está implícitamente probando varios hiperparámetros puede dividir su conjunto de prueba en dos (validación y prueba). Guarde sin utilizar una fracción de estos datos para realizar predicciones (Punto 5).

4. **(10 ptos)** Implemente métodos de regularización vistos en clase para mejorar su aprendizaje. Describa sus resultados.

5. **(Bono: 5 ptos)** Realice predicciones con datos que no hayan sido utilizados antes. ¿Qué opina de su modelo?

Parte II (50 ptos)

En este ejercicio utilizaremos la base de datos de precios de hogares en algunos suburbios de Boston en los años 70 para realizar un ejercicio de regresión (predecir un valor continuo en lugar de una etiqueta discreta). Este dataset contiene 506 datos divididos en un conjunto de entrenamiento (de tamaño 404) y un conjunto de prueba (de tamaño 102).

Para cargar el dataset use:

```
from tensorflow.keras.datasets import boston_housing
(train_data, train_targets), (test_data, test_targets) = (boston_housing.load_data
    ())
```

1. **(10 ptos)** Investigue qué representan las matrices/vectores cargados y sus posibles valores. Analice de forma preliminar sus datos (ocurrencias y distribuciones).

2. **(10 ptos)** Prepare sus datos para que puedan ser recibidos por una red neuronal (como usted habrá visto, los valores de cada una de las entradas de sus vectores parecen tener rangos diversos). En este tipo de situaciones se suele normalizar los datos (restar la media y dividir por la desviación estándar de cada una de los "features" del conjunto de entrenamiento). Los datos de prueba también se normalizan utilizando las medias y desviaciones estándar del conjunto de entrenamiento.

3. **(20 ptos)** Utilizando Keras, implemente diferentes tipos de redes con diferentes activaciones y arquitecturas pero sin ningún método de regularización. Experimente también con los optimizadores más frecuentes. Tenga en cuenta que la activación de la última capa debe ser lineal, o a lo sumo ser ReLU o alguna variante de ReLU que no tenga cotas superiores. Si utiliza estas activaciones puede utilizar la función de costo o pérdida de error cuadrático medio (MSE). Describa sus resultados.

   Sugerencia: Como está implícitamente probando varios hiperparámetros y su conjunto es pequeño, puede utilizar validación cruzada (cross-validation).

4. **(10 ptos)** Implemente métodos de regularización vistos en clase para mejorar su aprendizaje. Describa sus resultados.

5. **(Bono: 5 ptos)** Realice predicciones con datos artificiales que usted haya creado teniendo en cuenta las descripciones de los features de sus inputs. ¿Qué opina de su modelo?