

## Bootcamp Data Science - Ejercicio conceptual 12 - Gerardo Rodríguez

### a) Redes neuronales recurrentes (RNR)

Estamos diseñando una RNR para el análisis de sentimientos en reseñas de películas. Las reseñas son cadenas de texto que contienen palabras como "bueno", "malo", "genial", "terrible", etc. Nuestra RNR deberá procesar estas reseñas palabra por palabra y, al final, emitir una calificación que indique si la reseña es positiva o negativa.

1. Entrada: ¿Qué tipo de dato sería la entrada para nuestra red? ¿Cómo podríamos representar cada palabra para que la red neuronal la entienda?

De entrada, suponemos que la data está en formato de texto, por lo que deberemos procesar las palabras identificadas en los comentarios. Actualmente ya existen librerías que permiten procesar enunciados, lo cual permitirá separar el enunciado por palabras indexadas (tokenizadas), con el fin de que las palabras puedan ser segmentadas y simplificar la estructura de enunciados.

2. Procesamiento: Dado que estamos usando una RNR, ¿cómo manejaría la red la secuencia de palabras? ¿Qué importancia tendría la "memoria" en este caso?

El sistema recibe las palabras y maneja los índices en formato de matriz, para esto también tenemos librerías que permitirían identificar el estado de humor de los autores de los comentarios. Podríamos incluso ponderar como Positivo, un poco positivo, un poco negativo y Negativo.

Por lo que podríamos separar los datos ya clasificados en buenos y malos para entrenar el modelo.

La construcción de textos es compleja, ya que mientras una redacción avanza el significado puede cambiar, por lo que la posibilidad de que las neuronas de la red sean recurrentes, dará la posibilidad de que el resultado sea reprocesado incluso al momento de haberse liberado un resultado en la misma neurona. Ejemplo de esto son los signos de interrogación, admiración o el sarcasmo. La recurrencia permite guardar en memoria las palabras previas y hacer interpretaciones conjuntas de los datos acumulados al momento. Para esto deberá activarse la bidireccionalidad de las neuronas.

3. Salida: ¿Qué tipo de salida deberíamos esperar de nuestra red? ¿Cómo interpretaríamos esa salida para decidir si una reseña es positiva o negativa?

Al usar el entrenamiento para clasificar los comentarios, el sistema estaría identificando para cada comentario si este fue: Positivo, un poco positivo, un poco negativo y Negativo respectivamente.

Esta interpretación se podría dar al recibir porcentajes de estimación del comentario.

Finalmente, el sistema en la fase de training, estaría identificando si la data de training evaluó correctamente y en qué porcentaje es confiable.

Otra forma sería estimarlo de manera binaria mediante unos y ceros, para definir si fue negativo o positivo el comentario, en este caso redondeando el porcentaje de la salida a uno o cero.

## b) Redes neuronales convolucionales (RNC)

Supongamos que queremos construir una RNC para identificar si una imagen contiene un gato o un perro. Las imágenes se presentan en diferentes tamaños, colores y poses de los animales.

1. Entrada: ¿Qué tipo de datos serviría como entrada para nuestra RNC? ¿Cómo deberíamos preprocesar estas imágenes para que sean aptas para la red?

Para que el sistema pueda procesar las imágenes de manera homogénea, necesitamos que todas las imágenes tengan la misma proporción en forma y calidad (resolución en píxeles).

Por tanto, si las imágenes no son homogéneas, debemos reprocesarlas para que tengan las mismas proporciones.

De igual modo puede considerarse una entrada la configuración del modelo: filtros, padding, strides, tipo de función, la cantidad de capas, entre otras.

2. Convolución y Pooling: Explica cómo estas dos operaciones contribuyen al proceso de identificación de características en las imágenes. ¿Qué representa cada una de ellas en el proceso de comprensión de las imágenes?

Al haber definido el modelo, se procesarán 'n' veces las imágenes, iniciando con el proceso de "convolución".

En la Convolución se estarán extrayendo fragmentos de cada imagen, dependiendo como haya sido configurado el modelo, y convirtiendo la información de estas en matrices de datos. Estas matrices (kernel) se utilizarán para identificar las características de las imágenes (bordes, texturas o patrones).

Por cada proceso de Convolución, posteriormente se realizará un proceso de "Pooling", este consiste en identificar la información más importante de la imagen y eliminar aquella que no es vital para interpretarla. De esta manera aceleramos el proceso lectura de y procesamiento de imágenes.

3. Clasificación: Una vez que hemos obtenido una serie de mapas de características a través de las capas convolucionales y de pooling, ¿cómo continúa el proceso para clasificar la imagen como un gato o un perro?

Finalmente, mediante entrenamiento le diremos al algoritmo cuales fotos son de perros y de gatos.

Así podremos definir un porcentaje de las imágenes para entrenamiento y otro para pruebas, por lo que el mismo sistema definirá la mejor forma de interpretar que representa la imagen.

Hacer esto permitirá mediante datos controlados validar la efectividad del modelo para aplicarlo con imágenes diferentes a las del modelo.