Statistical Learning II

Diego Raúl Alvarez Mendoza

September 2021

1 Redes Neuronales Recurrentes - RNN

Las redes neuronales recurrentes, RNN por sus siglas en inglés, (Recurrent Neural Network) son una variación de una red neuronal básica. Los RNNs son utilizados para procesar datos secuenciales, como el procesamiento del lenguaje natural y el reconocimiento de audio.

Dichas redes neuronales utilizan memoria secuencial para aprender patrones y predecir el N Ésimo elemento de una serie basado en los componentes anteriores.

Las RNNs, cómo su nombre lo indica, toman ventaja de un proceso recursivo mediante el cuál la salida de una subparte de la red es alimentada como entrada a la siguiente parte. De esta manera la red es capaz de predecir un término con un contexto previo de los N elementos precedentes.

A pesar de ser altamente eficientes para data secuencial, las RNNs pierden la capacidad de recordar términos anteriores conforme crece el número de elementos a ser aprendido. A este problema se le conoce como Short term memory problem, o problema de memoria a corto plazo. Esto es ocasionado por el problema de Vanishing Gradients ocasionado durante el proceso de aprendizaje, en el que el valor de los gradientes disminuye tanto que las capas más antiguas en la red son actualizadas por valores muy pequeños o nulos.

Para solucionar este problema se recomienda el uso de los algoritmos LSTM (Long Short Term Memory) o GRU (Gated Recurrent Units) que proveen mecanismos internos llamados compuertas que ayudan a regular el flujo de información.

Para la solución del presente problema se utilizó únicamente LSTM para resolver el problema de memoria a corto plazo. En LSTM, ocurren cuatro pasos para determinar la salida de una neurona. Primero, la puerta de olvido o forget gate decide qué es importante olvidar de los pasos anteriores. Segundo, la puerta de entrada o input gate, decide qué información es relevante agregar del paso actual. Tercero, la información es olvidada. Por último, la puerta de salida determina cuál debería ser el siguiente estado oculto.

1.1 Poesía utilizando RNN

La poesía es un género literario escrito en verso o prosa que se caracteriza por expresar ideas, sentimientos e historias de un modo estético. Las grandes

culturas antiguas desarrollaron estilos poéticos distintos y particulares, como los haikus en Japón o la oda en Grecia, que solían estar escritos en verso y daban importancia al uso de la métrica o la rima.

El poema, es la composición poética básica. En general, se compone de múltiples versos y estrofas. Cada estrofa se compone de múltiples versos y un poema se compone de una o varias estrofas.

El objetivo principal de la siguiente implementación era generar un poema compuesto por 5 estrofas generado por un modelo de RNN. Para esto se buscó generar versos libres de un máximo de 10 palabras utilizando redes neuronales recurrentes. Para ello, se entrenaron varias RNN utilizando embeddings, LSTM y LSTM bidireccional.

1.2 Datos de entrada

Como datos de entrada, se utilizaron los textos de dos libros de poesía en español. El primero, Eres Arte, escrito por Diego Alvarez, es un libro escrito en Guatemala y publicado en Junio del 2020. El segundo, un clásico de la poesía, 20 poemas de amor y una canción desesperada, escrito por Pablo Neruda en 1924.

Para la prueba inicial, se alimentó la RNN únicamente con el texto del libro eres arte. Con estos datos de entrada, se generó un primer modelo al cúal se le denominó eres arte.h5. Para este modelo, se obtuvo un accuracy final de 0.76 después de 100 epochs.

1.3 Arquitectura

De la Imagen 04 se puede observar como el valor del accuracy oscilaba entre los epochs 20 y 100. Por lo que para la segunda y tercera prueba, se implementaron las funcionalidades de Early Stopping y Model Checkpoint.

El Early Stopping se utilizó para terminar la ejecución del entrenamiento después de 5 epochs en los que el modelo no presentaba mejoras. Esto hizo que el tiempo de entrenamiento para los modelos fuera menor.

Para la segunda y tercer prueba se utilizó un texto de entrada combinando los dos libros Eres Arte y 20 poemas de amor y una canción desesperada. Para la segunda prueba, se utilizó una red neuronal recurrente con LSTM simple. Este modelo se denominó eres arte 20 poemas. Dicho modelo tuvo un accuracy final de 0.8162 después de 16 epoch.

Finalmente, se utilizó un LSTM bidireccional para entrenar al modelo con los mismos datos de entrada que en la prueba dos. Como se puede observar en la imagen 03 el uso de LSTM bidireccional tuvo un incremento en los parámetros entrenables de 877,245 del modelo original a 1,462,095. Esto tuvo como resultado un incremento significativo en el tiempo de entrenamiento y un incremento de 0.0128 en el accuracy final del modelo para un total de: 0.8290 contra 0.8162 del modelo anterior.

Debido al incremento marginal del accuracy y la duplicación del tiempo de

entrenamiento para este caso de uso no se recomienda el uso de LSTM bidireccional.

1.4 Resultados

Con los modelos generados, se buscó generar 5 estrofas para componer un poema final. Se buscó que el poema resultante fuera lo más parecido al texto producido por los modelos por lo que se trató que la modificación del texto fuera la mínima posible para que el poema tuviera sentido. Además, se buscó que el texto resultante cumpliera con dos funciones:

- Expresar una idea
- Evocar un sentimiento.

Cabe resaltar, que el poema final es una mezcla de textos producidos por dos diferentes modelos eres arte.h5 y eres arte 20 poemas final. Además dichos textos producidos por los modelos fueron editados por el autor de Eres Arte, Diego Alvarez para ayudar al modelo a cumplir sus objetivos.

1.5 Conclusiones y recomendaciones

En conclusión, la combinación de dos RNN entrenadas con dos textos de poesía produjeron un poema que cumple con el objetivo de comunicar una idea y expresar un sentimiento. El texto final fue un 63.04 porciento producido por el autor y un 36.96 porciento producido por el editor del texto.

Del entrenamiento se puede observar que los modelos pueden seguir mejorando su accuracy por lo que se recomienda que se entrene el modelo con más textos de poesía para evitar un overfitting de los datos. De esta manera el modelo podrá producir textos elocuentes de mayor longitud.

Además cabe mencionar que entre las limitaciones de los modelos realizados se encuentra la carencia de conocimiento de métricas y rimas. Por esto dichos modelos no podrían producir poemas con restricciones de métricas como los Haikus. Para resolver esto, se pueden utilizar datos de entradas específicos para poemas de este tipo. Con dicha información se espera que los modelos sean capaces de detectar dichas características y puedan producir poemas similares a un Haiku.

2 Feed Forward Network - Predicción de características del café

El café es considerado el segundo producto en volumen físico más comercializado en el mundo. Su consumo, a nivel global, se ha duplicado en las últimas décadas, pasando de 92 millones de sacos en 1990 a 162 millones de sacos estimados en 2019. Además, se estima que se bebe alrededor de 4.000 millones de tazas y la demanda sigue en aumento.

Los dos tipos de granos de café son arábica y robusta. El primero es más apreciado en el mercado, mientras el segundo tiende a ser más amargo y menos apetecible, pero tiene una concentración del 50

El objetivo de esta implementación fue predecir 10 características del café basado en factores cómo país de origen, altitud de cultivo, color, variedad y método de procesamiento.

- Acidez
- Aroma
- Sabor
- Dulzura
- Uniformidad
- Post Gusto
- Cuerpo
- Balance
- Limpieza
- Cupper Points

Uno de los principales retos para los catadores inexpertos, al ser la catación un proceso muchas veces subjetivo, es entender qué tan bien se están percibiendo las características del café sin un experto cómo guía. Este modelo puede ser utilizado para comparar las predicciones de catadores inexpertos con las predicciones de catadores expertos recopiladas por Coffee Quality Institue. Esto puede ayudar a estudiantes de la catación del café a saber qué tan acertadas son sus predicciones en comparación con las del modelo.

2.1 Dataset

El set de datos utilizados fue una recopilación de catas de café realizadas por profesionales alrededor del mundo. El set de datos fue recopilado a lo largo de 5 años por el Coffee Quality Institute. Posteriormente, el dataset fue extraido de la base de datos pública de Coffee Quality Institute por medio de web Scraping y ofrecida en línea en la plataforma Kaggle. El data set está compuesto por cataciones de 32 países, 28 variedades diferentes de granos de café y 5 métodos de procesamiento.

Cada muestras en el dataset representa una catación realizada por los profesionales que pertenecen al Coffee Quality Institute. Cada muestra posee 44 columnas que brindan diferente información sobre el café de la catación que se realizó. Esto incluye información como país de origen, nombre de la finca, puntuación por característica del café entre otros.

Para efectos de este proyecto, durante el proceso de análisis inicial de los datos se removieron columnas que no aportaban información a la predicción que se quería realizar. Por ejemplo, una de las columnas que se removió fue el nombre de la finca donde se cultivó el café. Esto se debe a que se consideró que no existían muestras suficientes por finca para poder predecir la calidad del café según la finca específica. Para el caso de Guatemala se contaba con información de 52 fincas diferentes. Sin embargo, la información no era suficiente para ser relevante al set de datos.

2.2 Data Engineering

Para el proceso de limpieza de datos, se inició con la definición de las columnas a utilizar para realizar la predicción. Se descartaron columnas que no proveían información relevante para el modelo. Luego se procedió a realizar el proceso de NA treatment para contrarrestar el efecto de los datos faltantes.

Posteriormente se realizó un tratamiento de Outliers, ya que se detectó que existían 4 observaciones para las cuales los valores de altitud mínima y máxima no tenían sentido. Esto puede deberse a un error al momento de introducir los datos durante el proceso de catación.

Por último, se procedió a hacer el proceso de Feature Encoding a las columnas con variables categóricas.

2.3 Arquitectura

Para la arquitectura de la FFN se utilizó una capa de entrada de tamaño 11, dos capas intermedias de tamaño con activación Relu y por último una capa de salida de tamaño 10 que representa las características que puede predecir el modelo.

2.4 Resultados

Después de entrenar el modelo, se obtuvo un MSE final de 0.2711 para la data de entrenamiento y 0.2426 para la data de pruebas. En la siguiente gráfica se muestra cómo decrece el MSE con el tiempo.

Al comparar los resultados con la set de datos de validación se obtuvo un MSE de 0.11. Por esto se concluye que el modelo es capaz de predecir las 10 cualidades del café de una manera precisa.

3 Red Neuronal Convolucional CNN

El reconocimiento óptico de caracteres es el uso de tecnología para distinguir caracteres de texto impresos o escritos a mano dentro de imágenes digitales de documentos físicos, como un documento en papel escaneado. El proceso básico de OCR implica examinar el texto de un documento y traducir los caracteres a un código que se puede utilizar para el procesamiento de datos. A veces, el OCR también se denomina reconocimiento de texto.

El objetivo de esta implementación fue crear una red Convolucional capaz de clasificar un caracter en una de 62 categorías. Para ello se entrenó un modelo capaz de clasificar una imagen en los caracteres A-Z, a-z y 0-9.

Data Set Una de las dificultades con este dataset fue la cantidad limitada de imágenes disponibles para entrenar el modelo. Se utilizaron 55 muestras para cada caracter.

3.1 Arquitectura

Para mejorar el accuracy del modelo se decidió utilizar la funcionalidad de image augmentation para generar diferentes versiones de una misma imagen. Esto ayudó a que el modelo tuviera más data disponible para su entrenamiento.

Para el modelo final, se utilizaron como capas intermedias capas convolucionales, max pooling, batch normalization y flatten. Para la salida se utilizó una capa densa con activación Softmax para obtener la mejor predicción del modelo.

Cabe mencionar que el modelo tiempo significativamente alto en comparación con los otros modelos (1hr). Esto se debe a que el modelo tenía que leer los archivos de las imágenes mientras estaba entrenando.

Para validar si los datos también se entreno una red FFN que recibía como parámetro de entrada un archivo .csv con las imágenes procesadas en blanco y negro.

3.2 Resultados

Después de entrenar el modelo, se llegó a un resultado final de Accuracy = 0.6718. Al realizar pruebas con el modelo se puede observar que algunas predicciones aún no incorrectas, como se muestra a continuación.

En conclusión, a pesar de que el proceso de data augmentation ayudó a mejorar el modelo, este no se encuentra aún en un nivel de accuracy aceptable para ser utilizado en un sistema. Se recomienda entrenar el modelo con más imágenes de caracteres.