Análisis de sentimientos con Deep Learning

Jorge Luis Ramos Zavaleta
Centro de Investigación en Matemáticas. Unidad Monterrey.

***Resumen*—En los ultimos años las redes neuronales profundas han marcado una fuerte tendencia al mostrar un gran poder en el terreno del machine learning , principalmente cuando se aplica en algunos tipos de datos no estructurados como imágenes y texto, donde las aplicaciones que hacen uso de este tipo de métodos ha crecido de una manera bastante amplía. Particularmente en el caso de textos ha logrado encontrar un campo de aplicaciones muy amplio desde traduccion de textos y clasificacion de ellos hasta normalización lexicológica. Sin embargo, este progreso no ha sido parejo para todos los idiomas dado que cada uno tiene su propia estructura grámatical en algunos casos se torna mucho mas complicado su aplicación. En este trabajo se presenta una aplicación de Deep Learning para detectar el sentimiento positivo o negativo en varios textos en inglés y español con el fin de verificar su potencial.**

I. Introducción

El aprendizaje maquina se ha convertido en uno de los campos mas deseados por las empresas en esta era de información. Los métodos que ha desarrollado esta área de las ciencias computacionales ha nutrido sistemas en los cuales se ha comenzado a replica el comportamiento humano de aprendizaje, razonamiento, resolución de problemas y la comprensión de patrones en general inlcuyendo aquellos que contienen los lenguajes humanos. Con el surgimiento del Procesamiento de Lenguaje Natural se crearon distintos modelos de tipo probabilistico que dieron resultados parciales a algunos de los problemas, sin embargo muchos de los problemas se antojaban imposibles siguiendo este enfoque.

El Deep Learning es una técnica de vanguardia en el campo de Machine Learning hoy en dia, aunque sus origines son mucho mas antiguos, pero debido a que el poder computacional que existía en los tiempos en que se propuso no eran suficientes para permitir su uso por lo que se optó por no hacer uso de dicha técnica. Sin embargo, en nuestros dias con la aparación de procesadores mucho mas potentes y de tarjetas de video que permiten ejecutar operaciones en ellas el uso de dicha técnica ha comenzado a destacar por las aplicaciones que ha encontrado dentro del area.

Cuando se compara el monto de los datos generados en una base diaria, los textos ciertamentente muestran sobrepasar los generados por imágenes, video y adios. La razón de este inmenso crecimiento es obvio, ya que hemos comenzado a comunicarnos a través de medios electrónicos y redes sociales en los últimos años que hasta es casi imposible imaginar una vida sin ellos. Por lo que el desarrollo de aplicaciones para este tipo de datos se ha convertido en uno de los grandes potenciales para las empresas.

II. Deep Learning

El término Deep Learning fue introducido en el area de Machine Learning en 1986, y después comenzo a utilizarse para denotar de esta manera al uso de las redes neuronales artificiales en los 2000's. Los métodos de Deep Learning permiten hacer el uso de multiples capas que permiten el aprendizaje de características a diferentes niveles de abstracción.

Deep Learning permite a las computadoras aprender conceptos complicados a partir de partirlos en conceptos mas simples y homologando la información obtenida en cada proceso para obtener una comprensión mucho mas amplia. En el caso de redes neuronales artificiales, Deep Learning o aprendizaje jerarquico se trata de asignar creditos en muchos pasos computacionales de manera precisa, para transformar la activación agregada de la red. Es decir, se hace uso de operaciones no lineales bajo diferentes arquitecturas con el fin de permitir a nuestra red neuronal aprender funciones muy complejas que representen de mejor manera las características del objeto de estudio.

Las redes neuronales artificiales han tenido un trayecto muy largo. La primera generación de estas redes se conoce como Perceptron el cual solo estaba compuesto por una sola capa de neuronas por lo que se limitaba su uso dado que solo permitia cómputos muy simples. La segunda generación integro el uso de algoritmos de Backpropagation para permitir la actualización de los pesos de las neuronas de acuerdo a las tasas de error obtenidas. Sin embargo, el Support Vector Machine emergió como un nuevo método que termino superando los resultados obtenidos por las ANN de esta segunda generación. Para superar las limitaciones que se tenian con los algoritmos de Backpropagation se propuso utilizar Maquinas de Boltzmann Restringidas que permitian que el aprendizaje se realizara de una manera mas fácil.

A partir de estos desarrollo comenzaron a gestarse nuevas técnicas y arquitecturas para las redes neuronales artificales que han permitido ampliar el numero de aplicaciones a las que se puede acceder bajo este enfoque. Por ejemplo se crearon arquitecturas como las Redes Neuronales con alimentación hacia adelante (FNN), Redes Neuronales Convolucionales (CNN), Redes Neuronales Recurrentes (RNN) entre otras.

II-A. Recurrent Neural Networks

Las Redes Neuronales Recurrentes (RNN) son una familia de redes neuronales utilizadas para el procesamiento

de datos secuenciales. De la misma manera que las Redes Convolucionales son redes neuronales que estan especializadas para procesar una gradilla de valores X tales como una imagen, una RNN es una red neuronal que esta especializada para procesar una sucesión de valores $x^1, ..., x^(n)$. Asi como las redes convolucionales pueden escalarse para procesar imágenes con altura y ancho muy grandes, las RNN pueden ser escaladas a sucesiones mucho mas largas de las que serían practicas para redes sin dicha especialización. Inclusive muchas de las redes recurrentes pueden tambien procesar sucesiones de longitud variable.

Sin emabrgo, hay un reto matemático para establecer las dependencias de largo plazo en las redes recurrentes. El problema basico es que los gradientes se propagan a lo largo de muchas fases y tiende a desvanecerse o a explotar. El primer caso es el que suele darse mas seguido incluso si asumimos que los parámetros son tomados de forma que la red recurrente sea estable (puede almacenar memorias, con gradientes que no explotan), la dificultad con dichas dependencias de largo plazo surge debido a los pesos exponencialmente pequeños dados por dichas interacciones de largo plazo, las cuales involucran la multiplicación de muchos jacobianos comparaados con los de corto plazo. Para subsanar un poco este tema se han establecido arquitecturas nuevas a partir la idea inicial de la red recurrente.

II-B. Long-Short Term Memory Network

Hasta el dia de hoy, los modelos que funcionan con sucesiones mas efectivamente en aplicaciones practicas son llamados gated RNNs. Los cuales incluyen al LSTM y redes basadas en gated recurrent unit (GRU). Estas redes con puertas de acceso (gated) estan basadas en la idea de crear caminos a través del tiempo que tengan derivadas que no se desvanezcan ni exploten.

Las redes LSTM son un tipo especial de RNN, capaces de aprender dependencias de plazo largo. Fueron introducidas por Hochreiter Schmidhuber en 1997, y fueron refinadas y popularizadas por mucha gente en trabajos subsecuentes. Estas redes trabajan tremendamente bien en una gran variedad de problemas, y son ampliamente usadas.

LSTM's estan explicitamente diseñadas para evitar el problema de dependencia de largo plazo. Esto lo logra, recordando información por largo periodos de tiempo siendo este su comportamiento basico.

Todas las redes neuronales recurrentes tienen la forma de una cadena de modulos repetidos de redes neuronales. En las RNNs estandar, este modulo repetido tiene una estructura muy simple, que puede ser una simple capa de activación con la función tangente hiperbólico como puede observarse en la figura 1. Por su parte las redes LSTM también tienen esta estructura de cadena, pero el modulo de repetición tiene una estructura diferente. En lugar de tener una sola capa de de una sola red neuronal, tiene cuatro, interactuando en una forma muy especial como puede verse en la figura 2.

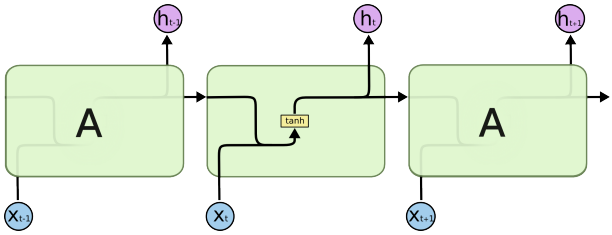


Figura 1. El módulo repetido en una RNN estándar contiene una sola capa.

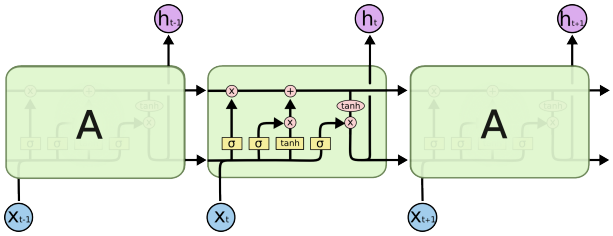


Figura 2. El módulo repetido en una red LSTM tiene cuatro capas interactuando.

La idea clave en las redes LSTM esta en el estado celda que es la linea que corre de manera horizontal a través de la parte superior de la figura 2 y que puede observarse mejor en la figura 3 donde puede verse que funciona como una cinta de transporte que avanza a través de toda la cadena con solo algunas interacciones lineales menores, por lo que la información fluye a lo largo del estado sin cambiar demasiado. Mientras que las otras flechas casi imperceptibles en esta última figura funcionan como puerta que interactuan con el estado celda observando que información es realmente relevante para permitir su paso.

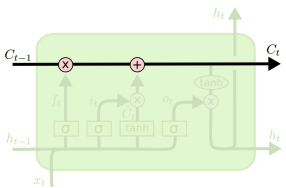
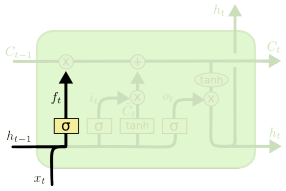


Figura 3. Estado celda de una red LSTM.

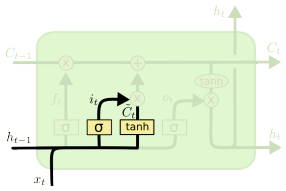
El primer paso es nuestra red LSTM es decidir que información vamos a conservar para que continue su camino a lo largo del estado celda. Esta decisión se logra usando una capa sigmoide llamada la capa de la puerta del olvido. Esta capa puede verse en la figura 4 donde la información es discriminada al hacerse uso de una función sigmoide. Después de esto, la información que sobrevivio a este proceso entra a una nueva capa donde nuevamente es procesada por

una función sigmoide y reconfigurada después en otra capa con una función tangente hiperbólico con el fin de crear un vector de nuevos valores candidatos como se observa en la figura 5.



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

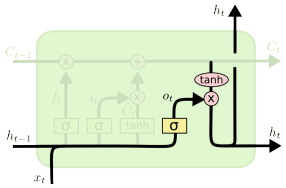
Figura 4. Capa sigmoide



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

Figura 5. Capas sigmoide y Tanh

Por último este nuevo vector se filtra de nuevo usando una combinación de función de activación sigmoide y tangente hiperbólico como puede observarse en la figura 6.



$$o_t = \sigma\left(W_o \; [\, h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figura 6. Capas finales sigmoide y Tanh

III. Análisis de Sentimientos

El análisis de sentimientos o minería de opinión es el estudio computacional de opiniones, sentimientos, emociones y actitudes de las personas hacia entidades como productos, servicios, organizaciones, individuos, tópicos, eventos y los atributos que ellos conllevan. El rapido crecimiento del campo coincide con el de las redes sociales en el internet, por ejemplo, reseñas, foros de discusión, blogs, Twitter y otras redes sociales, porque por primera vez en la historia de la humanidad, tenemos un enorme volumen de datos de opinión y se encuentran en un formato digital. Desde inicios de la decada de los 2000's, el análisis de sentimientos ha crecido hasta ser una de las áreas mas activas de investigación en el area de procesamiento de lenguaje natural.

De hecho, debido al rango de sus aplicaciones su estudio se ha expandido a otras areas como las ciencias administrativas y las ciencias sociales. Esta proliferación es debida al hecho de que las opiniones son centrales para todas las actividades humanas y son capaces de inlfuencias nuestro comportamiento. Nuestra creencia y percepción de la realidad, y las elecciones que hacemos, son en un grado considerable, condicionadas a como los los demás evaluan el mundo. Por esta razón, generalmente cuando necesitamos tomar una decisión buscamos la opinión de otros, y esto es verdad tanto para personas como para organizaciones.

Desde hace varios años se han producido un gran numero de tecnicas para varias tareas de análisis de sentimientos, que incluyen tanto métodos supervisados como no supervisados. En el lado supervisado, en los primeros trabajos se hizo uso de toda clase de tecnicas de aprendizaje maquina supervisado como SVM, Naive Bayes, Entropia máxima y también se hizo uso de diferentes características. Por el lado no supervisado se buscaron formas de explotar lexico de sentimientos, análisis gramatical y patrones sintácticos. Por lo que se genero un numero muy grande de publicaciones en esta subarea.

Desde hace poco mas de una década, el uso de técnicas de Deep Learning ha emergido como una practica muy poderosa en el área de machine learning, y ha permitido producir resultados que ahora son el estado del arte de muchos dominios de aplicación, desde visión por computadora hasta reconocimiento de discurso. La aplicación de Deep Learning al análisis de sentimientos se ha vuelto muy popular debido a los resultados que ha mostrado. Sin embargo, los resultados principales que se han logrado para NLP se han dado solo en algunos idiomas, principalmente inglés, por lo que su aplicación no es directa cuando se trata de otro tipo de lenguaje.

IV. Datos

Para este trabajo se hizo uso de 5 conjuntos de datos, 4 de ellos son reseñas de productos o servicios en inglés y el otro consiste de Tweets en español. Cada uno de los datos presenta su propia forma específica por lo que no debería considerarse creer que un solo modelo entrenado con uno de dichos conjuntos de datos podría funcionar con los demás, pero aun asi se realizo este experimento para corroborar esta hipótesis. Para ello las pruebas se realizaron tomando un conjunto de 80 % de los datos para entrenar el modelo y del resto se uso para probar la precisión.

Los conjuntos de datos en inglés estan son:

1. UMICH SI650 que contiene 7086 reseñas de peliculas, se encuentra bien balanceado entre las reseñas positivas y negativas, y contiene reseñas como las siguientes:
   - The Da Vinci Code book is just awesome.
   - Oh, and Brokeback Mountain is a TERRIBLE movie...
2. Reseñas de celulares en Amazon que contiene 1,000 reseñas bien balanceadas.
3. Reseñas peliculas en IMDB que contiene 1,000 reseñas bien balanceadas.

4. Reseñas de restaurantes en Yelp! que contiene 1,000 reseñas bien balanceadas.

Dado la poca disponibilidad de conjuntos en español solo se pudo contar con un conjunto de Twitts y se probó el algoritmo con los datos originales y con una versión en donde se eliminaron algunas URLS que no aportaban contenido real al Tweet y tambien se eliminaron los emoticons en los Tweets positivos para verificar su efecto. Los datos se ven de la siguiente manera:

1. 3 campañas en 1 año @eliasbendodo @margadcm @Francissalado ;-)) gracias ;-)) @ppmalaga http://t.co/R7FXTMYy
2. 3 campañas en 1 año @eliasbendodo @margadcm @Francissalado gracias @ppmalaga

Cabe aclarar que estos datos solo se encuentran etiquetados de manera binaria, por lo que se le asigna un cero si el sentimiento es negativo y un uno si el sentimiento es positivo.

V. Modelos

Inicialmente se generó un modelo para cada uno de los conjuntos de datos haciendo uso de la arquitectura de la red LSTM con lo que se consiguieron buenos resultados en general, y después para los conjuntos en inglés se genero un modelo y se intento predecir los sentimientos en los otros conjuntos con este modelo para comprobar nuestra hipotesis de que no se obtendrían buenas clasificaciones, al menos no mejores que con los obtenidos por los modelos individuales.

La arquitectura establecida para todos los conjuntos de datos fue la siguiente
Embeddings $\rightarrow$ LSTM con Hidden layer $\rightarrow$ Capa densa con una neurona $\rightarrow$ Función de activación sigmoide

Sin embargo, los parámetros variaron. En el caso de los datos de UMICH SI650 y los tweets se utilizo un conjunto de valores, mientras que para los demás datos se utilizo otro conjunto. En el primer caso se usaron vectores de longitud 128 para la capa de embeddings, 64 neuronas en cada capa escondida, lotes de tamaño 32, 10 epocas y una taza de Dropout de 20 %. Por su parte para los otros datos se usaron vectores de longitud 512 para la capa de embeddings, 128 neuronas en cada capa escondida, lotes de tamaño 32, 15 epocas y una taza de Dropout de 30 %.

Para establecer la convergencia de la red se hizo uso de un función de error de tipo Binary Cross Entropy debido a que los datos estan clasificados de manera binaria, y además se hizo uso del optimizador RmsProp dado que el que mostró mejor desempeño aun cuando en la literatura teórica se recomienda usar el optimizador ADAM. El optimizador se utilizó con los parámetros por defecto que tiene el ambiente Keras.

VI. Resultados

Los resultados para el caso cuando se consideraron modelos individuales para cada conjunto de datos se detalla en la tabla I.

Datos	Test score	Precision	Recall positivo	Recall Negativo
UMICH SI650	0.071	98.7 %	98 %	99 %
Amazon	0.78	83.5 %	84 %	83 %
IMDB	1.303	74.5 %	73 %	78 %
Yelp!	1.083	77.5 %	75 %	79 %
Twitter	0.184	78 %	73 %	82 %
Twitter limpio	0.194	77.3 %	69 %	84 %

Cuadro I
TABLA DE RESULTADOS DE LOS RESULTADOS PARA LOS MODELOS INDIVIDUALES

Puede observarse que en todos los casos se logra una clasificación no tan mala, incluyendo sus tazas de recall positivo y negativo, siendo que en los casos de los datos de UMICH SI650 y las reseñas de Amazon se lograron las mejores clasificaciones. Por otra parte, debe observarse que al realizarse la limpieza de los datos para el caso de los tweets se obtuvieron resultados similares, pero el recall positivo es mas bajo en el caso de los tweets limpios por lo que podemos pensar que al eliminar los emoticons de los tweets positivos permite que se generen mas falsos positivos.

Para el caso en que considero un modelo para probar todos los datos en inglés se encontró que aquel que generaba los mejores resultados es el generado con los datos UMICH SI650, los resultados se resumen en la tabla II.

Datos	Test score	Precision
IMDB	0.426	52.8 %
Amazon	0.408	49.6 %
Yelp!	0.423	49.4 %

Cuadro II
RESULTADOS OBTENIDOS USANDO SOLO UN MODELO PARA CLASIFICAR LOS DEMAS CONJUNTOS DE DATOS

Como puede observarse las clasificaciones empeoraron mucho con respecto a los obtenidos con los modelos individuales, principalmente en el caso de los datos de Amazon donde su precisión bajo casi 30 %, probando con esto nuestra hipótesis de que un solo modelo no podría ser suficiente para clasificar bien los otros datos, pues el vocabulario dentro de cada una de las reseñas de los productos o servicios es específico para cada uno de ellos.

VII. Conclusiones

El uso de tecnicas de Deep Learning obedece a la creciente de necesidad de buscar mejores maneras de establecer predicciones principalmente en datos con estructuras complejas como imágenes y texto, en donde ha mostrado una capacidad muy por encima de algoritmos tradicionales debido a que es capaz de usar la estructura inherente de los datos en lugar de establecer transformaciones que terminan perdiendo una gran parte de la información de la estructura.

Sin embargo, hay varios problemas al tratar con este tipo de modelos, de entrada debe considerarse el tipo de arquitectura a utilizar para obtener buenos resultados, de no considerarse una buena arquitectura no se puede aprovechar la estructura de los datos, y por tanto pueden terminar obteniendose resultados peores que con métodos probabilisticos mas tradicionales. Aparte de esto, los modelos de Deep Learning tienen un gran numero de parámetros por lo que se vuelve complicado encontrar los mejores para obtener buenos resultados sin establecer un sobreajuste.

En particular, en el caso de procesamiento de lenguaje natural los métodos de Deep Learning han demostrado ser la punta de lanza para un gran numero de aplicaciones, y puede observarse que las publicaciones que se han comenzado a realizar en esta area generalmente hacen uso de estas tecnicas, puesto que han demostrado una capacidad muy buena de clasificación y predicción para este tipo de datos.

VIII. Referencias

Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y. (2016). Deep learning (Vol. 1). Cambridge: MIT press.

Liu, B. (2015). Sentiment analysis: Mining opinions, sentiments, and emotions. Cambridge University Press.

Medsker, L., Jain, L. C. (1999). Recurrent neural networks: design and applications. CRC press.