Skip-gram: Estudio de Hiperparámetros.

Matías Pizarro
Dept. de Ciencias de la Computación
Universidad de Chile
Santiago, Chile
matias.pizarro.o@ug.uchile.cl

Alexandre Poupeau
Dept. de Ciencias de la Computación
INP Grenoble
Grenoble, Francia
alexandre.poupeau@grenoble-inp.org

Mitcheel Lanas
Dept. de Informática Médica.
Universidad de Chile
Santiago, Chile
mitcheel@gmail.com

***Resumen*—El modelo de *Skip-gram* para representaciones de palabras es una de las últimas técnicas para trabajar con texto. Hemos hecho algunos experimentos con el fin de encontrar una relación entre los hiperparámetros de este modelo y la precisión que se puede alcanzar en un problema típico como lo es el de clasificación de *reviews* de películas. Los hiperparámetros que se han ido modificando son el tamaño de ventana y el tamaño de los vectores de palabra que genera esta técnica de *Word2Vec*. Para poder calcular las métricas obtenidas, se ha entrenado una red neuronal recurrente (*rnn*) para usarla como clasificador.**

I. Introducción

Dos similares pero a la vez distintas técnicas de representar las palabras como vectores (*words embeddings*) fueron introducidas con los modelos *Continuous Bag-of-Words (CBOW)* y *Continuous Skip-gram Model* creados por Mikolov *et al.* [1]. Este trabajo está inspirado en un posterior avance sobre *Skip-gram* [2] realizado por el mismo Mikolov, quien propone el uso de *hierarchical softmax* en lugar de *softmax* y *Negative Sampling* en vez de *Noice Constrastive Estimation*. En este estudio hemos usado distintos valores para el ancho de las ventanas y también distintos valores de tamaño vector de palabra que se genera. Para llevar esta tarea a cabo, se ha usado el conjunto de datos "*Large Movie Review Dataset*", del cuál se hablará más adelante, para generar los vectores que posteriormente fueron procesados por una *rnn* que clasifica una *review* como buena o mala. La estructura de este informe es la que se describe a continuación:

- Manejo del dataset.
- Presentación del modelo Skip-gram
- Modo de evaluación
- Resultados
- Conclusiones.

También hemos dejado el código que se ha creado para su libre uso y para futuros nuevos experimentos.

II. Construcción del Dataset

Como se ha mencionado en la introducción de este informe, el dataset que se ha utilizado para el entrenamiento y evaluación en este problema es el "*Large Movie Review Dataset*"[1]. Este dataset se distribuye en 25000 datos de entrenamiento y 25000 datos de prueba con sus respectivas etiquetas incrustadas en el nombre de los archivos. Para manejar estos datos, lo primero que se ha hecho es *parsear* su nombre para obtener la etiqueta y posteriormente se ha añadido cada *review* a un conjunto de *reviews* para ser usadas como *corpus*, esto considera ambos conjuntos, de entrenamiento y de prueba. Una vez obtenidos los vectores se han separado en conjunto de entrenamiento y prueba de la red neuronal.

[1]http://ai.stanford.edu/~amaas/data/sentiment/

III. El modelo Skip-gram

En primer lugar, vamos a explicar lo básico del modelo Skip-gram. En general, el modelo se representa de la forma en que lo indica la Figura 1.

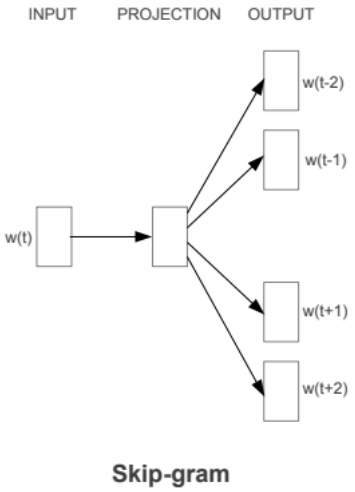

Figura 1. El modelo Skip-gram

La idea principal del modelo Skip-gram es predecir, para cada palabra, el contexto que se podría inferir a partir de esa palabra. Buscar ese contexto se traduce en buscar palabras que podrían aparecer antes y después de la palabra de entrada. En la Figura 1 se utiliza una ventana (*window*) de dos, esto significa que en ese caso el modelo intenta buscar dos palabras antes y dos palabras después que podrían aparecer para cada palabra del texto de entrenamiento.

Hay que entender que lo importante no es realmente el modelo, sino lo que genera, es decir vectores que son representaciones de palabras llamados también *words embeddings*. El entrenamiento es no supervisado es decir que no existe clases o *labels* que uno espera para cada palabra. El modelo genera vectores representaciones de palabras y busca las que

tienen más sentido. Palabras que tienen sentidos similares van a tener representaciones en vectores similares y van a estar más cerca en distancia.

IV. Modo de Evaluación

A. Contexto

1) Hiperparámetros evaluados: El propósito del estudio es de evaluar diversos hiperparámetros del modelo Skipgram y su impacto en la calidad de las representaciones de las palabras como vectores. Estudiamos dos hiperparámetros :

- *size* : Es la dimensión del vector que representa cada palabra. Tiene que ser única para cada palabra.
- *window* : Es el número de palabras antes y después que están tomados en cuenta para formar los *words embeddings*.

Se deduce que, *words embeddings* creados con hiperparámetros *size* y *window* más grandes van a ser de mejor calidad. ¿Cómo uno puede evaluar la calidad de *words embeddings*?

Lo que hacemos es evaluar esas representaciones utilizando un problema básico y clásico de *natural language processing*. Elegimos trabajar con los datos de la IMDb. El objetivo de la clasificación es de predecir, a partir del texto de una reseña de alguna película, si la reseña es buena o mala. Ese problema es muy básico y se sabe que funciona bien. El mejor resultado con esos datos sobre el conjunto de test (la mitad de las 50000 reseñas) hoy en día es de 96%[2].

2) Modelo de clasificación: El modelo que usamos para hacer la clasificación de las reseñas es un modelo de *Recurrent Neural Network (RNN)*. Usamos una célula llamada *Long-Short Term Memory* con una salida de tamaño 128. El modelo es de tipo *many-to-one*.

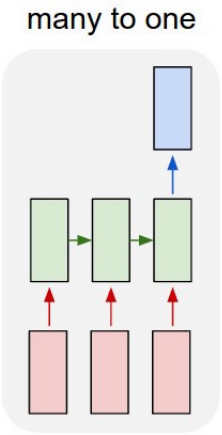

Figura 2. Many-to-one RNN

La entrada son vectores que representan las palabras de una reseña. Utilizamos un modelo tipo *Dynamic RNN*, es decir, que el modelo no tiene un número fijo de células *RNN* repetidas. Dicho número depende del número de palabras en cada reseña. El modelo se adapta a cada reseña. De esta forma el modelo es más eficiente y flexible, o sea, el modelo entrega una salida que toma en cuenta cada una de las palabras de una reseña.

Entrenar un modelo de *Dynamic RNN* toma mucho más tiempo porque no se puede hacer por *batch*. Al final, el vector tiene un tamaño de 128 y pasa por un *Multi-Layer Perceptron* de una capa para obtener un solo numero s en $[0, 1]$. Si $s \geq 0.5$ entonces la reseña es clasificada como "buena", de lo contrario se considera como "mala".

Para realizar todas las evaluaciones, hemos fijado el modelo de clasificación y todos sus hiperparámetros para solamente evaluar la calidad de los *words embeddings.* al variar los dos hiperparámetros mencionados anteriormente.

3) Modelos Entrenados: A continuación introducimos los modelos que se crearon y cómo nos referiremos a ellos. Digamos que S y W son los valores de *size* y de *window* respectivamente. Entonces un modelos de Skip-gram tendrá el nombre de MS_W. De esta forma, los modelos que se entrenaron fueron los siguientes:

- $M10\_2$, $M10\_3$, $M10\_5$, $M10\_10$
- $M25\_2$, $M25\_3$, $M25\_5$, $M25\_10$
- $M50\_2$, $M50\_3$, $M50\_5$, $M50\_10$
- $M100\_2$, $M100\_3$, $M100\_5$, $M100\_10$
- $M200\_2$, $M200\_3$, $M200\_5$, $M200\_10$
- $M500\_2$, $M500\_3$, $M500\_5$, $M500\_10$

A continuación veremos qué tal resultaron los modelos listados.

V. Resultados

Para cada uno de los modelos creados, se aplicó el mismo experimento de clasificación dando paso a los resultados que se ven en la Figura 3.

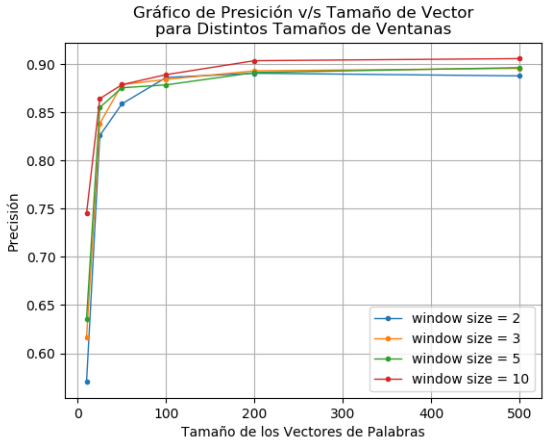

Figura 3. Curva de Precisión para todos los experimentos

Como era de esperarse, a mayor tamaño de ventana, mayor precisión se obtiene. También se puede apreciar que a mayor tamaño de vector de palabra, mayor es la precisión y esto se debe a que el vocabulario es enorme, por lo que, se necesitan muchas dimensiones para lograr discernir bien el uso de las palabras, aunque debiese llegar un punto en que

[2]http://nlpprogress.com/english/sentiment_analysis.html

mayor dimensión no brinde mejoras. De la misma forma un bajo número de dimensiones empobrece las relaciones que se puedan encontrar, pues usando *size* de 10 y *window* de 2, se obtuvo el peor resultado. Todos los resultados de las precisiones se pueden ver en la Tabla 1.

Tabla I
PRECISIÓN PARA LOS DISTINTOS MODELOS DE SKIP-GRAM

s\w	**2**	**3**	**5**	**10**
10	0.570	0.617	0.636	0.745
25	0.826	0.839	0.855	0.864
50	0.859	0.879	0.876	0.879
100	0.886	0.884	0.879	0.889
200	0.891	0.893	0.892	0.904
500	0.888	0.896	0.897	0.906

La Tabla 1 nos da a conocer con exactitud los valores de precisión que no se ven con claridad en el gráfico. Lo claro es que el mejor modelo obtenido es fue el $M500\_10$ obteniendo una precisión de $\sim 91\%$ lo cuál es muy bueno considerando el estado del arte mencionado en la sección IV.A.1

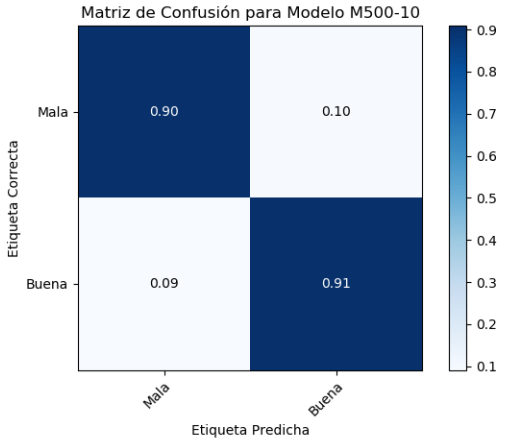

Figura 4. Matriz de confusión M500_10

La Figura 4 muestra la matriz de confusión obtenida con los resultados de nuestros mejores *words embeddings*. Esa matriz de confusión fue obtenida utilizando 25000 *test reviews* entonces el resultado es relevante y substancial. Se puede ver que las *reviews* fueron bien clasificadas por las *reviews* malas y buenas.

VI. CONCLUSIÓN

En este trabajo se pudo observar cómo el desempeño de los modelos varía según fue modificado el tamaño de la ventana y el tamaño del vector, y ha finalizado en general con buenos resultados, es decir, que a pesar de las métricas y el valor final de precisión alcanzado, que no fue tan cercano al alcanzado por el estado del arte, el hecho de que se haya obtenido una relación positiva entre el tamaño de ventana y la precisión, y el tamaño de vector y la precisión, es satisfactorio y era lo que se esperaba. Pues, hay casos que son más sencillos de analizar si miramos más contexto, a modo de ejemplo, si miramos la palabra ”bueno” dentro de una frase que califica una película, podríamos encontrarnos con dos semánticas distintas, una como una calificación positiva, usada como adjectivo calificativo, “el actor es bueno, trabaja bien...” o negativa, usada como interjección, “bueno, la película partía bien y terminaba mal.”. Además, si agregamos que se obtiene un mayor valor semántico al usar más contexto, nos resulta esperable que a mayor tamaño de ventana, mayor sea la precisión.

Además, es destacable notar que solamente con 25 de dimensión para describir las palabras (*size*), se puede obtener una precisión mayor a 85% con una *window* mayor a 5. Se puede inferir entonces, que no sirve de mucho utilizar una dimensión tan alta porque, según lo que se pudo observar en nuestro trabajo, la precisión no tiene un aumento significativo a partir del *size* 25. Este resultado puede depender del problema, es decir, usar menor dimensión en caso de textos con un vocabulario especifico o mayor dimensión si el vocabulario es mucho más grande (todo el diccionario). No obstante, debe generalizarse dado al carácter neutral de nuestra experimentación.

El código de este trabajo se encuentra libre para poder seguir ocupándolo y agregándole mejoras.[3]

REFERENCES

[1] T. Mikolov, I. Sutskever, Kai Chen *et al.*, “Efficient Estimation of Word Representations in Vector Space”.
[2] T. Mikolov, I. Sutskever, Kai Chen *et al.*, “Distributed Representations of Words and Phrases and their Compositionality”.

[3]https://github.com/poupeaua/skipgram-rnn