Modelar la Probabilidad de Palabra Normalizada con una Red Convolucional Sobre Caracteres

1ro John Edward Miller
Ingeniería/Ciencia de la Computación
Pontificia Universidad Católica del Perú
Lima, Perú
jemiller@pucp.edu.pe

2do Ricardo Jose Linares Juarez
Ingeniería/Ciencia de Computación
Pontificia Universidad Católica del Perú
Lima, Perú
ricardo.linares@pucp.edu.pe

***Abstract*—Presentamos un modelo de palabra «word model» como simplificación de un modelo de lenguaje «language model». Utilizamos vectores de segmentos de caracteres en lugar de oraciones compuestas por palabras, las salidas del modelo son las probabilidades de las palabras normalizadas por el largo del vector de segmentos de caracteres. El modelo generado es una red neuronal de arquitectura convolucional con puertas (gated-CNN). La función de pérdida empleada fue la entropía cruzada sobre los segmentos de caracteres y la medida primaria de evaluación es la perplejidad. Experimentamos con parámetros de arquitectura: número de capas, número de filtros, ancho de convoluciones, la fuente de «embedding» de segmentos de caracteres. El corpus consiste en tablas de palabras etiquetadas como secuencias de segmentos de caracteres del alfabeto fonético internacional (IPA). El proyecto todavía esta en proceso.**

***Index Terms*—convolucional, red neuronal, modelo de lengua, modelo de palabra**

I. Introducción

Construimos un modelo generativo de palabra como una generalización de un «language model», modelo generativo de lenguaje, con el fin de estimar la probabilidad de palabra normalizada por el largo de la palabra en segmentos de caracteres. En vez de oraciones compuestas de palabras, tenemos palabras compuestas de segmentos de caracteres, las cuales denominamos el «word model». Aplicaciones de este modelo son: 1) Juzgar si la ortografía o pronunciación es correcta según el lenguaje, por ejemplo: corrector ortográfico. 2) determinar el idioma del texto u oración verbal, por ejemplo: identificación de lenguaje. 3) discriminar entre palabras nativas o prestadas de otro idioma.

La línea base de un «word model» es el modelo y proceso de Markov de orden dos o tres, donde las secuencias de caracteres están modeladas con probabilidad de palabra dependiente en uno o dos caracteres anteriores. Nosotros desarrollamos un modelo de red neuronal con convoluciones de secuencias de caracteres. Nuestro modelo neuronal es del tipo generativo con «auto-encoder». Las entradas del modelo (segmentos de caracteres) son las mismas salidas a predecir. Por tanto, entrenamos el modelo sin etiquetar los datos, sin supervisión, como en un «language model».

Las características del modelo son las siguientes: 1) El vocabulario de segmentos de caracteres está compuesto típicamente por < 100 segmentos, 2) El modelo usa un «embedding» de segmentos con dimensión entre $[8..32]$, como la dimensión del embedding varía con la raíz cuadrada del vocabulario. 3) Algunas capas de convolución para recibir y entender los patrones de los «embeddings». 4) La salida «fully connected» con «SoftMax» para calcular las probabilidades y coincidir con la entrada durante el entrenamiento.

El objetivo de nuestro «word model» es discriminar entre palabras nativas o prestadas basado en secuencias de sonidos representados por caracteres del alfabeto internacional fonético (IPA). Usamos la base de datos de «World Loan Database» [6] que tiene 41 lenguas con tablas de $\approx$ 1300 palabras anotados con secuencias de segmentos de caracteres del IPA y con la respectiva probabilidad de ser un préstamo.

Por tanto, es un problema de pocos datos («small data»). Se debe tener cuidado especialmente con el problema de sobreajuste («overfitting»). El desafío es obtener un modelo con perplejidad baja al predecir las secuencias de segmentos y que a su vez sea un modelo suficientemente minimalista para no tener el problema de sobreajuste.

§II revisa el origen y desarrollo de modelos de lenguaje y el concepto de «embedding». §III explica mas sobre el modelo de red convolucional de palabras, algunos experimentos con la arquitectura y parámetros; y el método de evaluación del modelo.

II. Estado de Arte

Revisamos algunos modelos de lenguaje y adaptación al modelo de palabra «word model», y el problema de «embedding» con caracteres.

A. Modelos de lenguaje

Andrey Markov en su trabajo de cadenas de probabilidades discretas y condicionados en los últimos estados, «Markov chains», calculaba secuencias de vocales de la novela "Eugene Onegin". Claude Shannon [9] desarrollaba su teoría de comunicaciones usando modelos de Markov con «finite state machines» generando secuencias de caracteres. Este es el inicio en la modernidad de modelos generativos de secuencias de caracteres.

El modelo de Markov se volvió una herramienta importante para modelar lenguaje en general, por ejemplo: lenguaje, categorías, morfemas, sonidos y caracteres. Chen y Goodman [3]

desarrollaron y demostraron el estado de arte al calcular probabilidades de Markov con alisamiento, para tomar en cuenta la variabilidad de frecuencias del corpus. Usamos el modelo de Markov de "Natural Language Toolkit" [2] como línea base en este trabajo.

El modelo de lenguaje que marca la pauta hacia los modelos de redes neuronales es de Bengio et al. [1]. Es un modelo generativo de oraciones de lenguaje. Tiene por entrada «embedding», «fully connected» a la capa escondida con activación de $tanh$, y «fully connected» a la capa de salida con activación de SoftMax. Nos atrae la simplicidad de este modelo, pero intentamos algo mas robusto sin agregar demasiada complejidad.

En la actualidad, existen modelos más robustos, como los modelos recurrentes, modelos recurrentes con atención, y modelos únicamente con atención [13]. Estos modelos tan complejos están sobre utilizados en modelos de palabras.

Recientemente, existen modelos convolucionales que tienen poder computacional similar que los modelos mas robustos, pero con menos parámetros [5]. El modelo convolucional con puerta de Dauphin et al. [4] tiene buenos resultados en el «Google Billion Word» como datos de prueba, los cuales son competitivos con los mejores modelos pero con mucho menos recursos. El uso de la puerta brinda algo de poder como en LSTM o GRU, y la arquitectura convolucional mantiene la eficiencia. Este es nuestro punto de partida para la construcción de un modelo de palabras, «word model».

B. «Embedding»

Aunque había la idea de usar «embedding» para palabras y categorías simbólicos en general. El concepto de «embedding» es que un vector de números reales representa palabras o categorías mejor que códigos «1-hot». La dimensión del vector de números reales es el orden de la raíz cuadrada de la dimensión de «1-hot» y tiene propiedades semánticas de razonamiento muy útiles. Bengio et al. [1] usa «embedding» de las entradas calculadas por la propia red neuronal.

Mikolov et al. [8] desarrolló el método «word2vec» para aprender utilizando los vectores de «embedding» eficientemente aparte de la red neuronal que usa. Ahora es común usar tales vectores prefabricados para redes neuronales con o sin actualizarlos durante el entrenamiento.

Silferberg et al. [10] realizó una revisión de «embedding» con caracteres, similar a nuestro problema, y encontró que «singular value decomposition» (SVD) con truncamiento de un matriz de valores «positive pointwise mutual information» (PPMI) funcionaba mejor para construir los vectores que las soluciones como «word2vec». En nuestro trabajo, probamos con «embedding» como parte directo del modelo de red neuronal, y con «embedding» con aprendizaje por PPMI-SVD con truncamiento.

III. METODOLOGÍA

Implementar la red neuronal para modelar probabilidades de palabras. Las entradas son palabras segmentadas por sus caracteres del alfabeto fonética internacional y representadas con «embeddings». Las capas escondidas son segmentos de bloques de unidades de convoluciones causales con puertas. La capa de salida es un SoftMax «fully connected» con la salida de las convoluciones y prediciendo la probabilidad de cada carácter. Ver figura 1 y figura 2 para mas detalle.

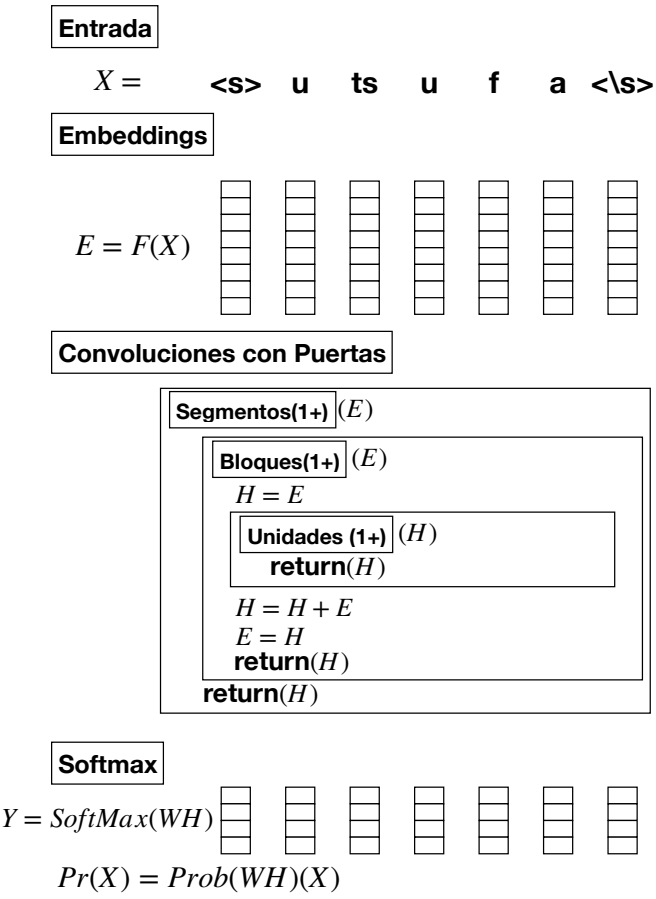

Figure 1. Arquitectura de red neuronal convolucional con puertas

El modelo base [11] implementa el algoritmo de [4] en «Keras Tensor Flow» para un modelo de lenguaje usando el corpus de «Penn Tree Bank» (PTB). La implementación modela secuencias de palabras de largo fijo (=35), marcando el inicio o fin de una oración con el símbolo <eos> y el inicio y fin de secuencias, con <SS> y <EE>. No hay respeto por la integridad de las oraciones cruzando secuencias porque las secuencias están formadas aleatoriamente. Hemos hecho una conversión ligera del modelo base para tratar con palabras y segmentos de caracteres IPA. Pero este modelo no respeta la integridad de las palabras cruzando secuencias. El primer experimento es convertir el modelo para respetar la integridad de la palabras – tratar con palabras individualmente o solo no truncar palabras entre secuencias.

A. Experimentos

Experimentamos con la alimentación de palabras a la red neuronal, con la arquitectura de la red neuronal convolucional y con los parámetros de las capas de la red neuronal.

1) Alimentación de la Red Neuronal: Además del modelo base adaptado para utilizar palabras, construimos una version que respetan la integridad de las palabras.

- Conversión simple de oraciones a palabras. No respeta la integridad de palabras. Es la linea base red neuronal del estudio.
- Se respeta la integridad de palabras con palabras múltiples. Apropiado para aplicación de un corrector ortográfico o identificación de lenguaje, para el estudio de palabras prestadas cuando el orden de palabras es aleatorio.

2) «Embedding»: Calculamos «embeddings» como parte del la red neuronal o de antemano usando «singular value decomposition» con el tamaño de ventana como otro parámetro. En ambos casos probamos con dimensiones alternativas de vectores numéricos.

- Método para calcular los «embeddings»
  - Como parte del modelo de red neuronal.
  - «Singular value decomposition» truncado de «positive pointwise mutual information» (PPMI-SVD) – con tamaño de ventana ±2 para calcular los PPMI.
- Dimensión de «embedding» (8, 16, 32).

3) Modelo convolucional: Queremos determinar cuál es el modelo convolucional más efectivo para modelar probabilidades de las palabras. Ver figura 1 y figura 2 para más detalle. Los parámetros a variar son.

- Parámetros de la unidad:
  - Ancho de convolución = 1, 2, 3
  - Cantidad de filtros= 8, 16, 32
- Configuraciones de la arquitectura:
  - Segmentos = 1, 2, 3
  - Bloques = 1, 2, 3
  - Unidades = 1, 2, 3

4) Fuente de palabras: Las tabla de palabras de WOLD [6] indica cuales son prestadas. Entonces, por cada idioma probamos un modelo solo con palabras no prestadas y un modelo con todas las palabras de la tabla.

B. Entrenamiento y Evaluación

Por cada idioma dividimos la tabla entre entrenamiento, validación, prueba en 70%, 15%, 15% respectivamente.

La función de pérdida es el promedio de entropía cruzada, $J = -\frac{1}{N}(\sum_{i=1}^{N} y_i \log(\hat{y}_i))$. La optimización es mediante el método «RMSprop». La medida de evaluación es la perplejidad, $PPL = e^{-J}$, directamente relacionada con la función de pérdida.

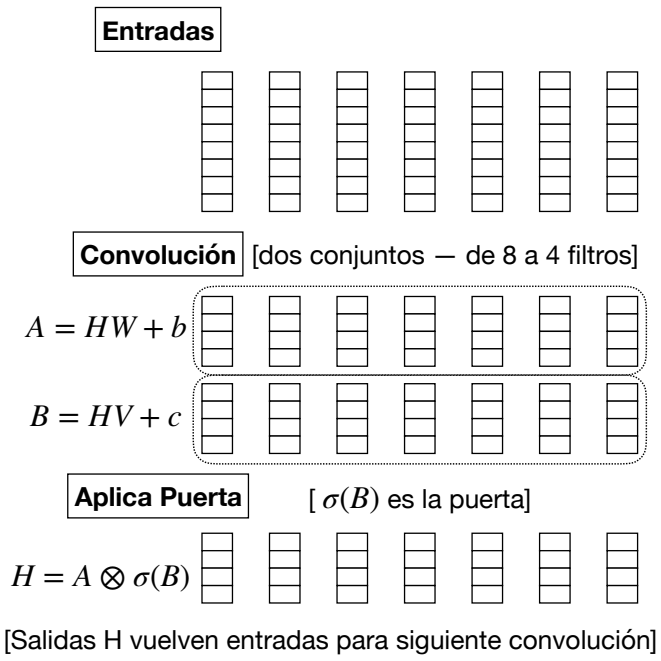

Figure 2. detalle de convolucional con puertas

REFERENCES

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. J. Mach. Learn. Res., 3:1137–1155, Mar. 2003

[2] E. K. Steven Bird and E. Loper. Natural Language Processing with Python. 2019.

[3] S. F. Chen and J. Goodman. An empirical study of smoothing techniques for language modeling. Computer Speech & Language, 13(4):359–394, 1999.

[4] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. In Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, pages 933–941, Sydney, NSW, Australia, 09 2017. JMLR, JMLR.org.

[5] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al. Recent advances in convolutional neural networks. Pattern Recognition, 77:354–377, 2018.

[6] M. Haspelmath and U. Tadmor, editors. World Loanword Database (WOLD). Max Planck Institute for Evolutionary Anthropology, Leipzig, 2009.

[7] M. R. Key and B. Comrie, editors. Intercontinental Dictionary Series (IDS). Max Planck Institute for Evolutionary Anthropology, Leipzig, 2015.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013. A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox. On the information bottleneck theory of deep learning. In International Conference on Learning Representations, 2018.

[9] C. E. Shannon. A mathematical theory of communication. SIGMOBILE Mob. Comput. Commun. Rev., 5(1):3–55, Jan. 2001. Original from: Bell Systems Journal. 1948.

[10] M. Silfverberg, L. J. Mao, and M. Hulden. Sound analogies with phoneme embeddings. In Proceedings of the Society for Computation in Linguistics (SCiL) 2018, pages 136–144, Salt Lake City, Utah, 01 2018. Society for Computation in Linguistics.

[11] DYK stikbuf. Language modeling with different models. GitHub Project: https://github.com/stikbuf/Language Modeling, June 2018.

[12] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. CoRR, abs/1503.02406, 2015.

[13] T. Young, D. Hazarika, S. Poria, and E. Cambria. Recent trends in deep learning based natural language processing. IEEE Computational intelligence magazine, 13(3):55–75, 2018.