DIFERENTES APROXIMACIONES A LA CODIFICACIÓN DEL VOCABULARIO EN MODELADO DE LENGUAJE CONEXIONISTA

Salvador Tortajada Velert, María José Castro Bleda

stortajada@dsic.upv.es, mcastro@dsic.upv.es

Departamento de Sistemas Informáticos y Computación,
Universidad Politécnica de Valencia, Valencia, España

RESUMEN

En los últimos años ha crecido el interés en la aplicación de las redes neuronales artificiales aplicadas a problemas de procesamiento del lenguaje natural. Este trabajo emplea modelos conexionistas para modelado del lenguaje haciendo hincapié en diversas técnicas de codificación de los datos con el fin de superar las limitaciones impuestas por las grandes dimensiones de los vocabularios. Los resultados obtenidos muestran algunas de las ventajas y desventajas de estas distintas aproximaciones.

1. INTRODUCCIÓN

En este trabajo se propone el uso de redes neuronales artificiales (RNA) para estimar modelos del lenguaje. Un modelo del lenguaje es un conjunto de mecanismos que definen la estructura del lenguaje restringiendo adecuadamente las secuencias de unidades lingüísticas probables que definen el conjunto de frases permitidas. Generalmente, el modelado del lenguaje no es un fin en sí mismo, sino que se aplica en otros sistemas como, por ejemplo, reconocimiento automático del habla o traducción automática. Las aproximaciones clásicas al modelado del lenguaje están basadas mayoritariamente en N-gramas [1] y, en menor medida, en gramáticas estocásticas [2]. A pesar de ello, el uso de RNA para el modelado del lenguaje ha sido empleado con éxito anteriormente [3–8]. Sin embargo, las características propias del lenguaje y las grandes dimensiones del vocabulario con el que se suele trabajar impiden obtener resultados más satisfactorios. Para solventar este problema se propone el uso de un tipo de codificación distinto, más compacto y que reduce las dimensiones de los parámetros a estimar sin necesidad de reducir la talla del vocabulario.

En este trabajo se han empleado redes conectadas hacia adelante, Perceptrón Multicapa (PM), entrenadas con el algoritmo básico de retropropagación del error para estimar modelos de lenguaje. Los resultados obtenidos muestran que, efectivamente, el uso de este tipo de redes ofrece ciertas ventajas y, además, invitan a reflexionar acerca de la importancia de este paradigma dentro de la disciplina del tratamiento del lenguaje natural.

Este trabajo se ha desarrollado en el marco del proyecto EDECAN (TIN2005-08660-c04-02) subvencionado por el MEC.

2. CODIFICACIÓN LOCAL Y DISTRIBUIDA

Uno de los problemas más graves a la hora de enfrentarse con un problema de lingüística computacional es la dimensionalidad del vocabulario. La cantidad de términos que se pueden encontrar en los distintos ámbitos del lenguaje y las distintas combinaciones posibles son enormes. Esta dificultad afecta al tratamiento del lenguaje natural en general y al modelado del lenguaje en particular. La solución empleada en este trabajo para los modelos conexionistas está directamente relacionada con la manera en la que las unidades lingüísticas son representadas en el modelo.

2.1. Codificación local

Habitualmente la representación del vocabulario en los modelos conexionistas se realiza siguiendo una *codificación local* o "1-de-C", donde cada patrón tiene una dimensión igual a la talla del vocabulario Ω, siendo $|\Omega| = C$. Dicha codificación emplea una única unidad de la capa de entrada para representar una palabra. Esto es, aquella unidad i que representa a la palabra i-ésima del vocabulario se activa y el resto quedan desactivadas. La codificación local es una forma sencilla, explícita y fácil de interpretar y manipular. Sin embargo, para vocabularios de grandes dimensiones, una codificación "1-de-C" provoca una gran dispersión de los datos, así como una gran ineficiencia puesto que se genera un número de conexiones y parámetros excesivo.

A continuación podemos observar un ejemplo de esta representación con dos palabras de un vocabulario hipotético de talla 16:

Azul	0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
Rojas	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0

2.2. Codificación distribuida

La solución empleada en otros trabajos -y que se sigue en éste-, consiste en utilizar un tipo de codificación

distinta [5,9], conocida como *codificación distribuida*. En la codificación distribuida cada unidad de entrada puede participar en la representación de una o más palabras. Se entiende que al intervenir varias unidades en la representación de las palabras, se necesitarán menos unidades de entrada y, por lo tanto, se tendrán menos parámetros a estimar. Dentro de la codificación distribuida existen dos tipos principales que presentamos a continuación.

2.2.1. Distribuida simbólica

Siguiendo con el ejemplo del vocabulario hipotético, podemos ver cómo se codificarían las palabras *azul* y *rojas* con una representación distribuida simbólica:

Azul	1 0 1 0 1 0 0
Rojas	1 0 0 1 1 0 0

En este tipo de representación cada unidad corresponde a una característica de la palabra, sea semántica o sintáctica. Así, por ejemplo, la primera unidad podría indicar que las palabras son un adjetivo; la siguiente unidad señalaría que es un verbo; la tercera podría indicar si el género es masculino o femenino; etcétera. Este tipo de codificación es rápidamente interpretable por un observador humano. Sin embargo, la reducción de las dimensiones depende de la cantidad de características de los términos que se desee representar y, además, este tipo de representaciones son, generalmente, *ad-hoc*. Por último, la automatización de este tipo de codificación distribuida podría plantear un nuevo problema, ya que el conocimiento previo necesario para saber algo tan sencillo como si una palabra del vocabulario es masculina o femenina requiere un esfuerzo computacional añadido.

2.2.2. Distribuida subsimbólica

En este tipo de codificación, las unidades, al contrario que en el caso anterior, no tienen una interpretación directa preestablecida de ningún tipo. Así, las palabras de nuestro ejemplo, *azul* y *rojas*, podrían ser representadas del siguiente modo:

Azul	0.22 0.04 0.89 0.45
Rojas	0.12 0.91 0.09 0.77

Este tipo de representación puede ser automatizada y puede aprenderse a partir de un corpus de entrenamiento [9]. Las dimensiones suelen ser del orden de $\lg |\Omega|$, que es un factor de disminución considerable. En cambio, la codificación no puede ser interpretada por un observador humano.

2.3. Codificación del vocabulario

En este trabajo se ha probado una codificación distribuida subsimbólica en varios de los experimentos. El proceso de codificación seguido ha variado en función del uso, o no, del contexto que rodeaba a las palabras. Así, el codificador empleado -también basado en RNA- puede utilizar contexto o cifrar únicamente la palabra aislada. Este artículo se centra en la segunda opción, dejando la primera para futuras investigaciones.

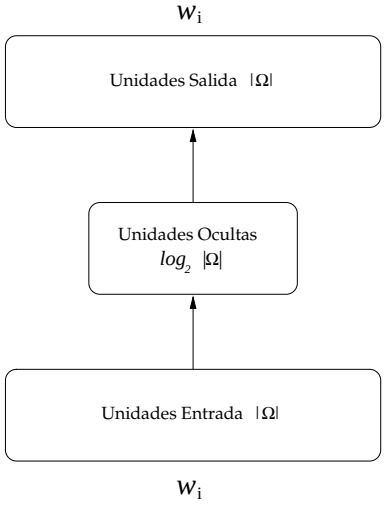

Figura 1. Perceptrón Multicapa para obtener una codificación distribuida subsimbólica a partir de la representación local de los datos. La entrada y salida son codificaciones locales de la palabra. Mediante el entrenamiento de la red podemos conseguir una codificación distribuida en la capa oculta.

La codificación sin emplear contexto se ha realizado a partir de un PM cuyas capas de entrada y salida consistían en una representación del vocabulario, Ω, codificado de forma local; y cuya capa oculta se compone de un número de unidades inferior a $|\Omega|$. Se puede ver un esquema en la Figura 1. Normalmente, con la codificación distribuida se reduce el número de unidades empleadas en un factor logarítmico. Es decir, $H = \lg |\Omega|$, donde H es el número de unidades ocultas empleadas en el PM codificador y, por tanto, el número de unidades empleadas para representar cada palabra del vocabulario.

Al entrenar no es necesario establecer un criterio de parada puesto que estamos ante un problema cerrado: los patrones son iguales en la entrada y en la salida y no existe peligro de sobreentrenamiento. Por lo tanto, la complejidad se limita a entrenar la red hasta obtener un error cuadrático medio que converja a cero.

3. MODELOS DEL LENGUAJE CONEXIONISTAS

Los modelos de lenguaje se estiman habitualmente haciendo uso de una aproximación estadística. Estos modelos se basan en la predicción de cada unidad lingüística de la frase dadas las anteriores. Los modelos basados en N-gramas son los más extendidos en la actualidad y consisten en aproximar la probabilidad como si únicamente influyeran las últimas $N-1$ palabras previas. Esto es,

$$Pr(W) \approx \prod_{i=1}^{|W|} Pr(w_i | w_{i-N+1} \dots w_{i-1}).$$

Partición	*No. de palabras*
Entrenamiento	1 004 073
Validación	80 156
Test	89 537

Tabla 1. Corpus Wall Street Journal.

Estos modelos se pueden aprender de forma automática a partir de un conjunto de entrenamiento. Los valores de cada N-grama pueden ser estimados por máxima verosimilitud.

Los modelos conexionistas introducen una aproximación diferente a la basada en N-gramas. En [3] se presentó este nuevo modelo. Posteriormente, en otros trabajos se han investigado y aplicado los modelos conexionistas para este fin [4–8].

La idea consiste en simular la ecuación de los modelos de lenguaje estadísticos. Para ello, en la capa de entrada se introduce la historia de la palabra a predecir, esto es, $h(w_i) = w_{i-N+1} \dots w_{i-1}$ y en la salida se espera que se prediga la palabra w_i. De este modo, utilizando la función de activación *softmax* en las unidades de salida se puede aproximar la probabilidad $Pr(w_i|h(w_i))$.

Los modelos conexionistas aportan dos ventajas respecto a los modelos convencionales. Primero, el número de parámetros a calcular crece linealmente con N en lugar de exponencialmente. Segundo, el suavizado se realiza automáticamente y se puede esperar una mejor generalización de los modelos. A pesar de esto, los modelos conexionistas presentan un inconveniente respecto a la aproximación basada en N-gramas: el coste temporal para entrenar el modelo.

4. LA TAREA PENN TREEBANK

El corpus usado en los experimentos es la parte del Wall Street Journal que había sido procesada en el Penn Treebank Project [10]. Este corpus consiste en un conjunto de textos en inglés extraídos del periódico Wall Street Journal. El número total de palabras es de 1 millón aproximadamente, con un vocabulario de más de $49\,000$. Todo el corpus fue etiquetado automáticamente con dos tipos de etiquetas: etiquetas POS y etiquetas sintácticas. El etiquetado POS consiste en un conjunto de 45 etiquetas diferentes, mientras que el sintáctico lo forman 14 categorías.

El corpus se dividió en tres conjuntos: entrenamiento, validación y prueba. Las características de las tres particiones se describen en la Tabla 1. Hemos utilizado en la experimentación las etiquetas POS y un vocabulario compuesto por las $10\,000$ palabras más frecuentes que aparecen en la partición de entrenamiento.

5. N-GRAMAS CONEXIONISTAS UTILIZANDO CATEGORÍAS

La primera aproximación al modelado de lenguaje con N-gramas ha sido utilizando las categorías del Penn Treebank, ya que nos ha permitido trabajar con un vocabulario muy reducido (45 categorías originales más el símbolo de contexto y el de token desconocido) sobre un corpus muy extenso. Así pues, hemos modelizado bigramas y trigramas con dicho corpus empleando un PM. Con el propósito de comparar los resultados obtenidos con los modelos de N-gramas estadísticos clásicos se ha empleado el toolkit SRILM para modelado de lenguaje con suavizado *Good-Turing* [11].

5.1. Codificación local

Para modelizar un bigrama con un PM se empleó una red con topología 47-H-47, donde H es el número de unidades ocultas, tanto en una como en dos capas. Las primeras 47 unidades representan la palabra de entrada, mientras que las 47 unidades de salida corresponden a las probabilidades que asocia el PM a cada una de las 47 posibles sucesoras. Para determinar el valor de H se ha realizado un barrido exhaustivo, probando diferentes combinaciones de una y dos capas, con diferente número de unidades en cada una. El mejor resultado se ha obtenido con una capa oculta de 16 unidades.

Para el caso del trigrama el proceso seguido fue prácticamente el mismo. Tras establecer una topología para el PM, que en este caso sería de 94-H-47, se pasó directamente a la búsqueda del mejor valor de H. En este caso, el mejor valor obtenido para H ha sido de 128 unidades con una única capa oculta.

5.1.1. Análisis de los resultados

Tras modelizar los N-gramas con $N = 2, 3$ podemos extraer una serie de conclusiones muy interesantes (todos los resultados se muestran en la Tabla 2). En ambos casos se han obtenido mejores resultados con el modelo conexionista que el correspondiente N-grama estadístico, siendo el número de parámetros mucho menor. El suavizado implícito de la red neuronal es capaz de afrontar muy bien la falta de muestras, mucho mejor que los suavizados de los N-gramas estadísticos. Por su parte, el entrenamiento de un N-grama conexionista es mucho más costoso en tiempo y además requiere un proceso de búsqueda de la mejor topología y de los mejores parámetros.

5.2. Codificación distribuida

La realización de un modelo conexionista basado en N-gramas empleando una codificación distribuida es una primera aproximación para llegar a realizar un modelo de lenguaje conexionista enfocado a superar las limitaciones que presentan los vocabularios de grandes dimensiones.

En esta aproximación tratamos de comprobar que una codificación distribuida de los datos -categorías- no sólo reduce el número de unidades en la capa de entrada, y por lo tanto el número de parámetros a estimar, sino que también proporciona un comportamiento similar al modelo con codificación local.

ML (POS)	*#*	*C*	*PPE*
Bigrama estadístico	2 209	-	9.52
Bigrama-PM, Local	1 567	257	9.47
Bigrama-PM, Distr.	807	146	9.61
Trigrama estadístico	103 823	-	8.36
Trigrama-PM, Local	12 207	115	8.27
Trigrama-PM, Distr.	7 692	30	8.49

Tabla 2. Perplejidad del conjunto de test (PPE) para los modelos de lenguaje del corpus de etiquetas POS del Penn Treebank con codificación local y distribuida, número de parámetros (#) y de ciclos (C) que ha costado entrenar la red neuronal, comparados con los modelos estadísticos.

En primer lugar se ha obtenido la codificación distribuida subsimbólica de las categorías POS del Penn Treebank con un PM tal como se explica en la Figura 1. El mejor resultado se obtuvo con una capa oculta de $\lceil \lg |\mathcal{C}| \rceil = 6$ unidades. Una vez obtenida la codificación distribuida, se entrena un PM con la mejor topología encontrada en los experimentos con codificación local, tanto para bigramas como para trigramas (una capa oculta de 16 unidades).

Dado que contamos con una codificación distribuida en la entrada y una codificación local en la salida, la topología final del PM es 6-16-47. Al emplear codificación distribuida en la entrada en lugar de local, el número de unidades se reduce de 47 a 6. Dicha reducción implica, a su vez, una disminución del número de parámetros de $1\,567$ a 807, notablemente inferior a los $2\,209$ parámetros necesarios para estimar los bigramas estadísticos.

Para el modelo con trigramas, la red entrenada y probada tiene una topología 12-128-47. La razón es que la capa de entrada cuenta ahora con las w_{i-2} y w_{i-1} palabras anteriores. Puesto que las palabras están representadas de forma distribuida tenemos $2\lceil \lg |\mathcal{C}| \rceil$ unidades en la capa de entrada; el número de unidades ocultas viene impuesto por el mejor resultado obtenido en las pruebas anteriores; la palabra w_i en la capa de salida se representa mediante codificación local. El número de parámetros total es $7\,692$, inferior los $12\,207$ parámetros a estimar con la codificación local o a los $103\,823$ parámetros del trigrama estadístico.

5.2.1. Análisis de resultados

Se puede observar en la Tabla 2 que la reducción de parámetros a estimar no es gratuita. Aunque la reducción del modelo y su estimación sean computacionalmente más eficientes, los resultados presentan una perplejidad ligeramente peor.

Debemos tener en cuenta que los resultados obtenidos para los modelos con codificación distribuida no han sido obtenidos a partir de un barrido exhaustivo, como se hizo para los modelos conexionistas con codificación local, sino que se han empleado aquellas topologías que mejores resultados ofrecieron en las pruebas mencionadas. Por lo tanto estos resultados son susceptibles de ser mejorados, aunque nada lo garantiza.

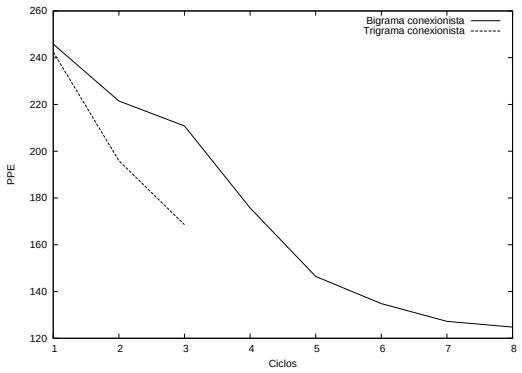

Figura 2. Evolución de la perplejidad de los modelos de lenguaje conexionistas (bigramas y trigramas) con codificación local para el corpus de palabras del Penn Treebnak. Los trigramas conexionistas todavía no han llegado a la zona de convergencia, por lo que, con un mayor número de ciclos, se podría obtener un modelo de mayor precisión.

ML (palabras)	*#*	*C*	*PPE*
Bigrama estadístico	10^8	-	137.06
Bigrama-PM, Local	$10^{7.16}$	8	124.79
Trigrama estadístico	10^{12}	-	119.81
Trigrama-PM, Local	$10^{7.39}$	3	168.60

Tabla 3. Perplejidad del conjunto de validación (PPE) para los modelos de lenguaje del corpus Penn Treebank con codificación local, número de parámetros (#) y de ciclos (C) que ha costado entrenar la red neuronal, comparados con los modelos estadísticos.

6. N-GRAMAS CONEXIONISTAS UTILIZANDO PALABRAS

Tras utilizar como vocabulario de la tarea las etiquetas POS y ver que los resultados obtenidos eran esperanzadores, se decidió pasar al conjunto de $10\,000$ palabras como vocabulario de la tarea.

6.1. Codificación local

Esta aproximación ha supuesto trabajar con una topología de $10\,000$ unidades en la capa de entrada para el caso del bigrama ($20\,000$ para el trigrama), y de $10\,000$ en la capa de salida.

En el caso de los bigramas, a falta de un barrido exhaustivo para seleccionar la mejor topología, se pueden ver los resultados obtenidos para el corpus de validación tras dos semanas de cálculo ininterrumpido en la Tabla 3. Como se puede observar, la PPE obtenida es muy buena, en comparación con la PPE del bigrama estadístico.

En el caso de trigramas, sólo hemos podido completar 3 ciclos de entrenamiento, que han tardado 30 días en terminar. Sin embargo, como muestra la Figura 2, la curva de perplejidad todavía puede disminuir con más ciclos de entrenamiento de modo que se podría reducir considerablemente.

6.2. Otras codificaciones

Una alternativa a la codificación local es la codificación distribuida, explicada en la sección 2. Sin embargo, la codificación del vocabulario no ha sido satisfactoria al no poder encontrar un mínimo local aceptable del error cuadrático medio. Otra alternativa estudiada es la codificación binaria, que consiste en obtener un diccionario con las palabras que forman el vocabulario y su correspondiente traducción en dígitos binarios. Esto consigue comprimir el tamaño de los vectores a longitud $\lg |\Omega|$. Sin embargo, los resultados obtenidos ofrecían un rendimiento del modelo bastante inferior a la codificación local.

7. CONCLUSIONES Y TRABAJO FUTURO

En el presente documento se han presentado los modelos de lenguaje conexionistas como modelos de lenguaje basados en un PM y planteados como alternativa a los modelos de N-gramas estadísticos tradicionales, prestando especial atención al problema de la dimensionalidad del vocabulario a través de la codificación distribuida y binaria.

En los modelos basados en clases (con vocabularios pequeños) se ha demostrado la capacidad de las redes neuronales para mejorar la eficiencia de los modelos estadísticos, así como el menor número de parámetros empleados. Del mismo modo, se demuestra que podemos disminuir la dimensionalidad para representar el vocabulario mediante la codificación distribuida, aún a costa de perder algo de eficiencia a cambio de emplear menos parámetros.

El hándicap de la dimensionalidad se presenta como el problema más grave en los modelos de lenguaje conexionistas. Como se ha podido comprobar, la codificación distribuida, empleada para reducir el número de unidades necesarias para representar el vocabulario, no funciona correctamente para tallas muy grandes. En este sentido giran los trabajos de Bengio y otros autores [5, 9]. Las posibles soluciones al problema están abiertas y siguen siendo un camino a investigar para la mejora de las prestaciones de los modelos conexionistas en general, y su aplicación al modelado del lenguaje, en particular. En esta línea se han presentado resultados de modelos de lenguaje que combinan RNA con otras técnicas [8, 12].

8. BIBLIOGRAFÍA

[1] F. Jelinek, *Statistical methods for speech recognition*, MIT Press, Cambridge, MA, USA, 1997.

[2] C. Chelba y F. Jelinek, "Structured language modeling," *Computer Speech and Language*, vol. 14, pp. 283–332, 2000.

[3] M. Nakamura y K. Shikano, "A study of English word category prediction based on neural networks," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP'89)*, Glasgow (Scotland), May 1989, pp. 731–734.

[4] W. Xu y A. Rudnicky, "Can Artificial Neural Networks learn Language Models?," in *Proceedings of the 6th International Conference in Spoken Language Processing (ICSLP'00)*, Beijing (China), 2000.

[5] Y. Bengio, R. Ducharme, P. Vincent, y C. Jauvin, "A Neural Probabilistic Language Model," *Journal of Machine Learning Research*, 2003.

[6] M.J. Castro y F. Prat, "New Directions in Connectionist Language Modeling," *LNCS*, vol. 2686, pp. 598–605, 2003.

[7] A Emami y F. Jelinek, "Exact Training of a Neural Syntactic Language Model," in *ICASSP*, 2004, vol. 1, pp. 245–248.

[8] H. Schwenk y J.-L. Gauvain, "Training neural network language models on very large corpora," in *Joint Conference HLT/EMNLP*, 2005, pp. 201–208.

[9] G.A. Casañ y M.A. Castaño, "Improvements on Automatic Word Codification for Connectionist Machine Translation," in *ECAI*, 2004, pp. 576–580.

[10] M.P. Marcus, B. Santorini, y M.A. Marcinkiewicz, "Building a Large Annotated Corpus of English: The Penn Treebank," *Computational Linguistics*, , no. 19, pp. 313–330, 1994.

[11] A. Stolcke, "SRILM - An extensible language modeling toolkit," in *International Conference on Spoken Language Processing*, 2002, vol. 2, pp. 901–904.

[12] F. Blat, M.J. Castro, S. Tortajada, y J.A. Sánchez, "A hybrid approach to statistical language modeling with multilayer perceptrons and unigrams," in *Proceedings of the 8th International Conference on Text, Speech and Dialogue*, 2005.