Traducción estadística basada en frases: RNN

Jaime Ferrando Huertas

Enero 2021

Contents

1 Introducción

El objetivo de esta práctica es evaluar el toolkit NMT-KERAS, una herramienta basada en Deep Learning para entrenamiento de modelos de traducción automática construido sobre Keras y Tensorflow. Similar a la primera práctica se ha entrenado un modelo básico de traducción español-ingles según las instrucciones del boletín de prácticas y despues se han realizado distintos experimentos para los ejercicios. Para evaluar nuestros modelos hemos hecho uso de la métrica BLEU [3] y hemos usado el corpus de EuTrans.

2 Ejercicios

2.1 Experimento inicial

El experimento inicial consiste en una RNN con arquitectura encoder-decoder, el encoder se compone de una capa LSTM [2] de 64 neuronas y un vector de embedding de tamaño 64. Por otro lado el decoder es también de una capa LSTM con 64 neuronas y tamaño de embeddings de 64. Se ha entrenado con learning rate 0.001 durante 5 epochs y Adam como optimizador. Hemos obtenido 93.70 de BLEU.

Experimento	Emb. Size	Enc. Size	Dec. Size	Optimizer	BLEU
Experimento inicial	64	1x64	1x64	Adam	93.70

Table 1: Experimento inicial

Este experimento inicial supone ya un incremento de BLEU de un punto sobre nuestro mejor modelo SMT creado con MOSES, ahora vamos a realizar más experimentos por si es posible incrementar este valor aún más.

2.2 Ejercicio 1

Para este ejercicio hemos evaluado el tamaño de los embedding tanto para la capa de entrada del encoder como los del decoder. Hemos usado tamaños similares para estos dos en cada experimento. Hemos hecho las modificaciones sobre la configuración del experimento inicial, obtenemos los siguientes resultados:

Experimento	Emb. Size	Enc. Size	Dec. Size	Optimizer	BLEU
Embeddings-32	32	1x64	1x64	Adam	89.79
Embeddings-64	64	1x64	1x64	Adam	93.70
Embeddings-128	128	1x64	1x64	Adam	**98.21**
Embeddings-256	256	1x64	1x64	Adam	97.71
Embeddings-512	512	1x64	1x64	Adam	96.90

Table 2: Evolución del BLEU dado tamaño de embeddings de entrada y salida

Experimento	Emb. Size	Enc. Size	Dec. Size	Optimizer	BLEU
1l-enc 1l-dec.	128	1x64	1x64	Adam	**98.21**
2l-enc 2l-dec.	128	2x64	2x64	Adam	97.07
3l-enc 3l-dec.	128	3x64	3x64	Adam	94.47

Table 3: Evolución del BLEU dado número de capas en encoder/decoder

Podemos observar como 128 embbedings nos da un aumento considerable de BLEU, posicionándose muy encima de otros experimentos vistos en el laboratorio de MOSES (91.97).

2.3 Ejercicio 2

Una vez encontrado el número óptimo de embeddings (128) queremos estudiar diferentes tamaños de nuestro modelo de red neuronal. Hay dos formas de aumentar el tamaño de nuestra red, ya sea más capas o aumentando el tamaño de las capas existentes. Primero hemos realizado experimentos con distintos números de capas en el encoder-decoder y hemos obtenido los siguientes resultados:

Observamos que aumentar el número de capas no ha reportado ningún beneficio en el BLEU y procedemos a experimentar con el número de neuronas en ellas. El modelo base contaba con 64 neuronas por capa y ahora vamos a probar con valores menores y mayores.

Experimento	Emb. Size	Enc. Size	Dec. Size	Optimizer	BLEU
32 neurons	128	1x32	1x32	Adam	95.50
64 neurons	128	1x64	1x64	Adam	**98.21**
128 neurons	128	1x128	1x128	Adam	98.08
256 neurons	128	1x256	1x256	Adam	96.28

Table 4: Evolución del BLEU dado número de neuronas en capas LSTM

Vemos que la configuración inicial de 64 neuronas por capa es la que mejores resultados nos reporta y decidimos mantenerla. También hay que tener en cuenta que esta configuración por defecto de nmt-keras y está configurada para entrenar solo hasta 5 epochs, normalmente los modelos con más capas necesitan más epochs para poder sacar beneficio de ellas y conseguir el mejor rendimiento debido al mayor coste computacional. Creemos que este análisis de distintos tamaños debería ser probado con distintas epochs para poder ser concluyente.

2.4 Ejercicio 3

En este ejercicio queremos estudiar optimizadores como Adadelta[5] y Adagrad[1], variaciones del optimizador adam que hemos estado usando hasta ahora. Se han probado sobre nuestro mejor modelo hasta el momento obtenido en el experimento $64 neurons$.

Experimento	Emb. Size	Enc. Size	Dec. Size	Optimizer	BLEU
Adam	128	1x64	1x64	Adam	**98.21**
Adadelta	128	1x64	1x64	Adadelta	91.73
Adagrad	128	1x64	1x64	Adagrad	4.55

Table 5: Evolución del BLEU con distintos optimizadores

Observamos que los dos optimizadores tienen un peor BLEU, con los resultados de Adagrad siendo realmente alarmantes. Una explicación para estos resultados podría ser la misma que dimos a los experimentos donde modificábamos el tamaño de la red, puede que estos optimizadores estén diseñados para periodos de entrenamientos más largos (mayor número de epochs).

2.5 Ejercicio 4

Por último hemos probado a ejecutar la arquitectura de transformer [4]. Esta arquitectura itera sobre los modelos encoder-decoder y sustituye las capas de LSTM por capas de atención para guardar el contexto temporal. En nuestros experimentos hemos probado distintos tamaños de esta arquitectura y ha sido necesario tener un tamaño de embeddings igual al tamaño de modelo de transformer. Resultados:

Experimento	Emb. Size	Enc. Size	Dec. Size	Optimizer	BLEU
Transformer-64	64	1x64	1x64	Adam	87.24
Transformer-128	128	1x64	1x64	Adam	**91.70**
Transformer-256	256	1x64	1x64	Adam	88.34

Table 6: Evolución del BLEU para distintos tamaños de transformer

Podemos ver que 128 vuelve a ser el tamaño óptimo como en el modelo encoder-decoder con attention del ejercicio 2. Sin embargo el modelo de transformer no consigue ganar al encoder decoder con attention con la configuración por defecto. De nuevo pensamos que con un numero de epochs mayor y aumentando el número de capas el transformer debería ganar dada previas experiencias con modelos de esta arquitectura.

3 Conclusión

Gracias a esta práctica hemos hecho uso de NMT-Keras como primer contacto con modelos de traducción automática basados en Deep Learning. Estos experimentos que nos han permitido entender mejor distintos aspectos de los modelos RNN a la vez de como interactúan con sus parámetros. Si comparamos sobre nuestros pasados experimentos con MOSES hemos conseguido notables mejoras en BLEU (92 vs 98). Como futuro trabajo proponemos el estudio de distintos valores de learning rate y número de epochs de entrenamiento, creemos que un estudio exhaustivo de estas llevaría a más mejores de BLEU

Se puede encontrar ahora una tabla con todos los experimentos realizados.

Experimento	Emb. Size	Enc. Size	Dec. Size	Optimizer	BLEU
Experimento inicial	64	1x64	1x64	Adam	93.70
Embeddings-32	32	1x64	1x64	Adam	89.79
Embeddings-64	64	1x64	1x64	Adam	93.70
Embeddings-128	128	1x64	1x64	Adam	**98.21**
Embeddings-256	256	1x64	1x64	Adam	97.71
Embeddings-512	512	1x64	1x64	Adam	96.90
1l-enc 1l-dec.	128	1x64	1x64	Adam	98.21
2l-enc 2l-dec.	128	2x64	2x64	Adam	97.07
3l-enc 3l-dec.	128	3x64	3x64	Adam	94.47
32 neurons	128	1x32	1x32	Adam	95.50
64 neurons	128	1x64	1x64	Adam	98.21
128 neurons	128	1x128	1x128	Adam	98.08
256 neurons	128	1x256	1x256	Adam	96.28
Adam	128	1x64	1x64	Adam	98.21
Adadelta	128	1x64	1x64	Adadelta	91.73
Adagrad	128	1x64	1x64	Adagrad	4.55
Transformer-64	64	1x64	1x64	Adam	87.24
Transformer-128	128	1x64	1x64	Adam	91.70
Transformer-256	256	1x64	1x64	Adam	88.34

Table 7: Experimentos realizados

References

[1] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 07 2011.

[2] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[3] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[4] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[5] Matthew D. Zeiler. Adadelta: An adaptive learning rate method, 2012.