

Módulo 9

Procesamiento de Lenguaje Natural o Minería de textos

Mtro. Luis Enrique Argota Vega



Tema 7: PLN con Redes Neuronales

Objetivo

El participante identificará el método de redes neuronales para la clasificación de textos, así como los modelos para el enfrentamiento al problema de generación de resúmenes de textos, a partir de las bibliotecas y servicios implementados.

Contenido

1. Redes neuronales para clasificación de textos
2. Modelos Secuencia a Secuencia
3. Redes neuronales recurrentes para modelos del lenguaje
4. Modelos basados en atención y *transformers*

Referencias

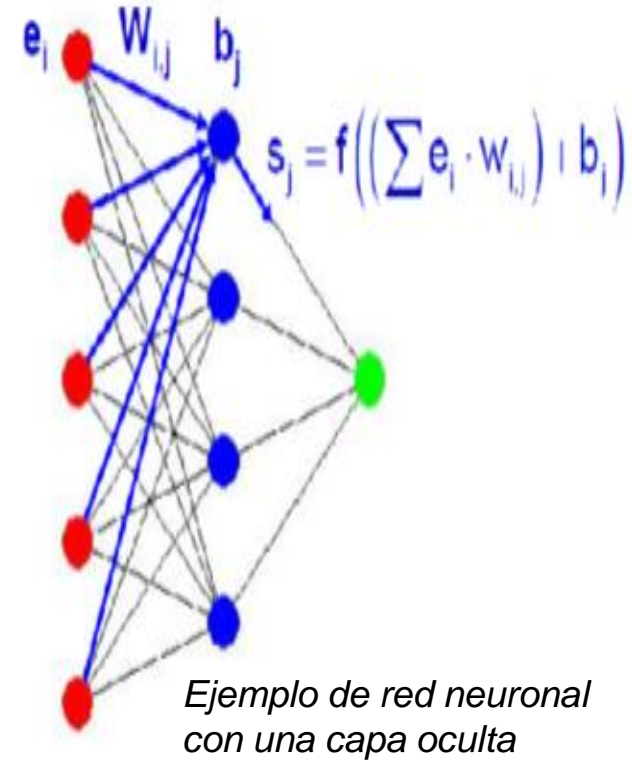
- Deep learning in natural language processing / Li Deng, Yang Liu, editors -- Singapore: Springer, [2018] 1 recurso en línea (xvii, 329 páginas): ilustraciones <https://link.springer.com/book/10.1007%2F978-981-10-5209-5>
- Deep learning for natural language processing: creating neural networks with Python / by Palash Goyal, Sumit Pandey, Karan Jain -- Berkeley, California: Apress, [2018] 1 recurso en línea (xvii, 277 páginas): ilustraciones <https://link.springer.com/book/10.1007%2F978-1-4842-3685-7>
- Guridi Mateos, G. (2017). Modelos de redes neuronales recurrentes en clasificación de patentes (Bachelor's thesis). Obtenido de: <http://hdl.handle.net/10486/679893>
- León Pacheco, P. (2017). Extracción de características de textos y clasificación según género literario mediante redes neuronales (Bachelor's thesis). Obtenido de: <http://hdl.handle.net/10016/27299>
- Peirano, M. D. (2020). Resumen de Textos con Modelos Secuencia-a-Secuencia: Varias Aproximaciones (Doctoral dissertation). Obtenido de: <https://riunet.upv.es/handle/10251/150240>

Introducción



Características red neuronal

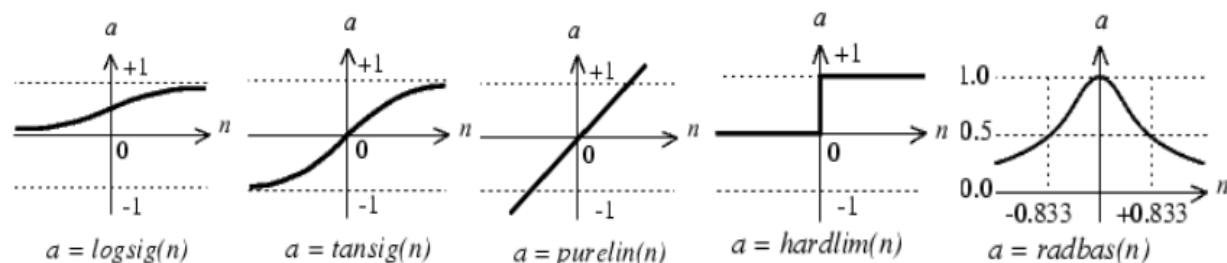
- ✓ Capaz de detectar **relaciones complejas y no lineales** entre variables, a partir de unidades sencillas como las neuronas, al disponer muchas de estas unidades en paralelo.
- ✓ Las variables se dividen en **variables de entrada y de salida**, relacionadas por algún tipo de correlación o dependencia (no necesariamente causa - efecto).
- ✓ También es posible que la salida sea la clasificación de las variables de entrada en diferentes grupos.



Funcionamiento red neuronal

El funcionamiento de una neurona consiste en la transformación de los valores de las entradas a través de las conexiones, en una salida. La salida se obtiene a partir de:

- ✓ La **función de propagación** más común consiste en el sumatorio de todas las entradas multiplicadas por los pesos de las conexiones, más un valor de sesgo o “bias”.
- ✓ La **función de activación**, en caso de que exista, activa o desactiva la salida de esta neurona.
- ✓ La **función de transferencia** se aplica al resultado de la función de propagación y normalmente consiste en una función de salida acotada como la sigmoidea (*logsig*) $[0,1]$, o la tangente hiperbólica (*tansig*) $[-1,1]$. Otras funciones de transferencia pueden ser una función lineal (*purelin*) $[-\infty, +\infty]$, base radial (*radbas*) $[0,1]$ o una función de discriminación (*hardlim*) $[0,1]$.



Tipos de redes neuronales

Según tipo de conexiones

- Redes de propagación hacia delante (***feed - forward***)
- Redes ***recurrentes***

Según tipo de aprendizaje

- Aprendizaje ***supervisado***.
- Aprendizaje ***no supervisado o autoorganizado***.

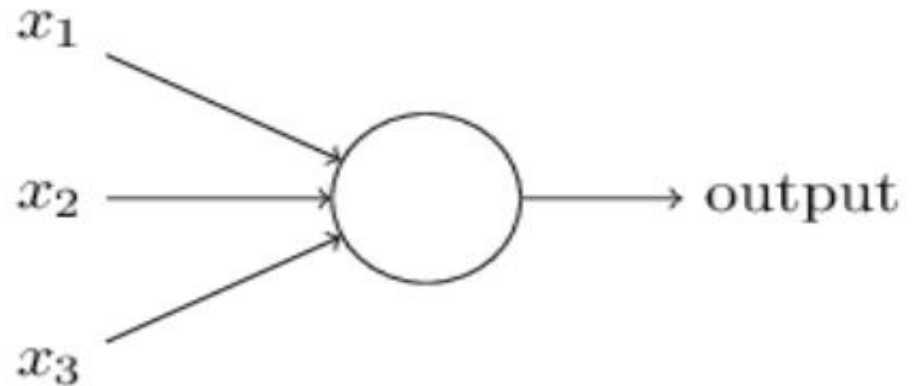
Ventajas de las RNA

- ✓ **Aprendizaje:** Tienen la habilidad de aprender mediante la etapa de aprendizaje.
- ✓ **Auto organización:** Crea su propia representación de la información en su interior, descartando al usuario de esto.
- ✓ **Tolerancia a fallos:** Debido a que almacena la información de forma redundante, ésta puede seguir respondiendo de manera aceptable aun si se daña parcialmente.
- ✓ **Flexibilidad:** Puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada.
- ✓ **Tiempo real:** La estructura de una RNA es paralela, por lo cual, si esto es implementado con computadoras o en dispositivos electrónicos especiales, se pueden obtener respuestas en tiempo real.

Desventajas de las RNA

- ✓ Complejidad de aprendizaje para grandes tareas, cuantas más cosas se necesiten que aprenda una red, más complicado será enseñarle.
- ✓ Tiempo de aprendizaje elevado.
- ✓ No permite interpretar lo que se ha aprendido.
- ✓ Elevada cantidad de datos para el entrenamiento.
- ✓ Otros problemas con las redes neuronales son la falta de reglas definitorias que ayuden a realizar una red para un problema dado.

Perceptrón



$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

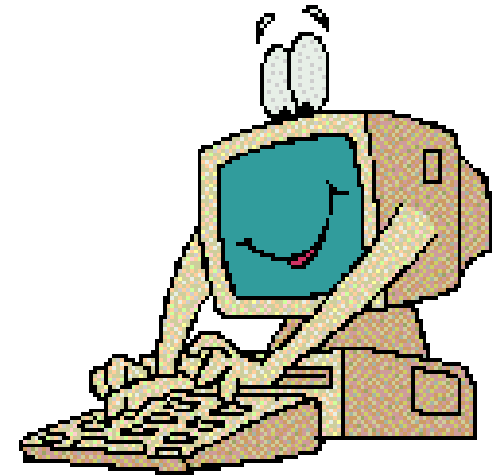
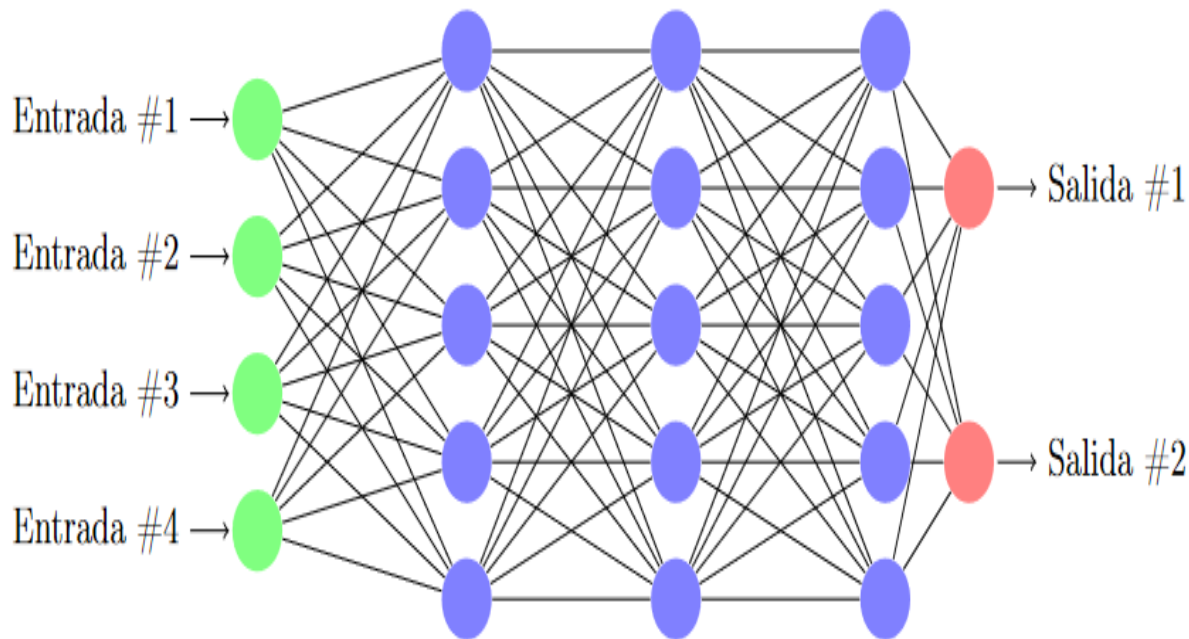
Un Perceptrón Simple es una red neuronal que posee una única neurona de entrada, que recibe tantas entradas binarias como atributos tengan los datos de entrada y produce una salida binaria.

Este modelo, en base a características del texto como el vector de la bolsa de palabras puede clasificar textos para una categoría.

Perceptrón Multicapa (Multi-Layer Perceptron)

- Implementa una versión ligeramente más compleja para acomodar diversidad de entradas y salidas.
- La salida de cada neurona afecta a las neuronas de la siguiente capa, dando pie a que la función generada para la salida sea más compleja.
- Este modelo de red se puede adaptar a casi cualquier tipo de problema.
- En el caso de un problema de clasificación, basta con ajustar la entrada para que la red se corresponda con el tamaño necesario y que la salida de la red se componga de cuantas neuronas sea necesario para representar las clases en las que se distribuyen los ejemplos.

Práctica



Ejercicio7(es)-PLN con Redes Neuronales.ipynb

Modelos de secuencias

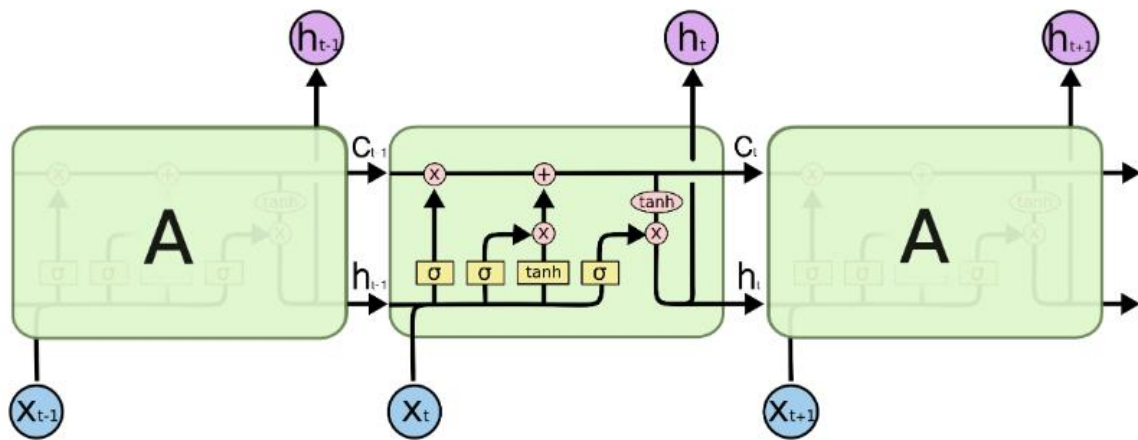
Los modelos de secuencias (*sequence models*) son las técnicas utilizadas cuando el orden y la secuencia de los datos aportan mucho valor predictivo.

Redes neuronales recurrentes

- Son una evolución de los perceptrones en las que puede haber ciclos (bucles de retroalimentación) en las conexiones entre las neuronas de cada capa; es decir, la salida de la red en la iteración n se introduce como entrada adicional en la iteración $n+1$.
- Sufren del Problema de desvanecimiento de gradiente (*Vanishing gradient problem*).
- Las neuronas de capas finales aprenden más rápido que las de las primeras capas.
- Existen al menos 20 tipos distintos de RNR, pero la predilecta hasta la fecha es la LSTM (*Long Short Term Memory*)

LSTM

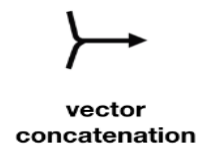
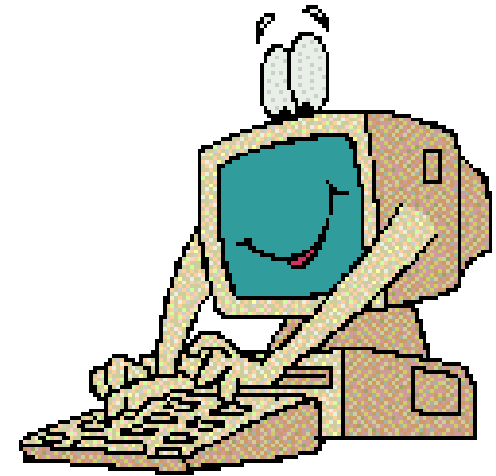
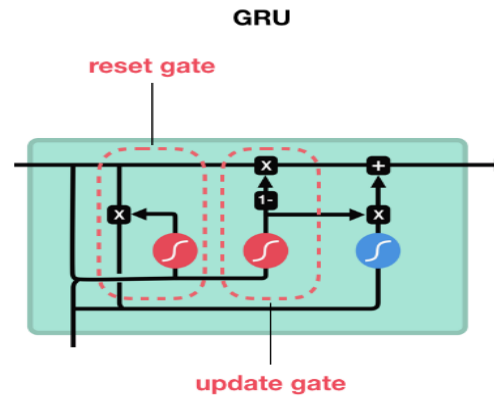
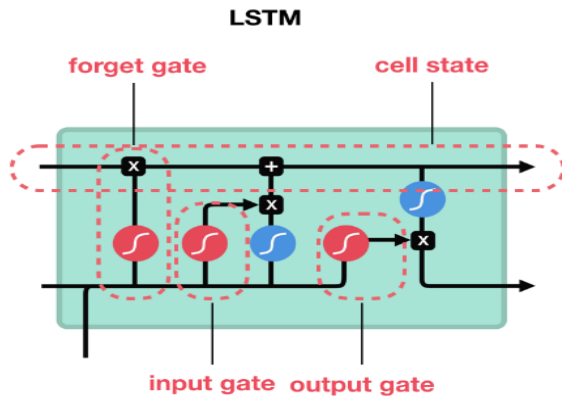
- La gran mayoría de modelos con RNR implementan esta variante.
- Permite conservar la información de iteraciones arbitrariamente lejanas a la actual.
- Consta de una célula, una puerta de entrada (x_t), una puerta de salida (h_t) y una puerta de olvido (C_t).
- La célula almacena la información entre iteraciones mientras que las puertas regulan el flujo de dicha información.



GRU (Gated recurrent units)

- Aumenta su eficiencia manteniendo su eficacia y solventar el problema de desaparición de gradiente.
- Agrega dos puertas (puertas de actualización y restablecimiento, *Update* y *Reset*) que mantienen un registro de la información más relevante para la predicción.
- La puerta de *Update* determina cuánta información previa se debe transmitir al siguiente paso de tiempo.
- La puerta de *Reset* se utiliza para determinar cuánta información es irrelevante y debe olvidarse.

Práctica



Ejercicio7(es)-PLN con Redes Neuronales.ipynb

Modelos basados en atención y *transformers*

A pesar de la introducción de las LSTM y las GRU, el problema de interdependencia de palabras alejadas entre sí persistía. Los modelos no eran capaces de captar dichas relaciones entre posiciones distanciadas. Esto se consiguió solucionar mediante un mecanismo llamado ***atención***. Esta técnica revolucionaria consiste en almacenar los contextos que se producen en cada iteración del *encoder* para después pasarlos conjuntamente al *decoder* de tal manera que el modelo aprenda a “prestar atención” a las partes de la secuencia de entrada que más le convenga.

Transformers

Los *transformers* se tratan modelos de Secuencia a secuencia (*Sequence to Sequence*, o comúnmente *Seq2Seq*). Son modelos que teniendo una secuencia como *input* devuelven otra secuencia.

Un *Transformer* es una arquitectura que convierte una secuencia en otra con la ayuda de dos partes (codificador y decodificador), pero es diferente de los modelos secuenciales anteriores ya que no utiliza RNN.

Transformers

En la imagen siguiente se pueden ver el codificador (izquierda) y decodificador (derecha) básicos de un Transformer.

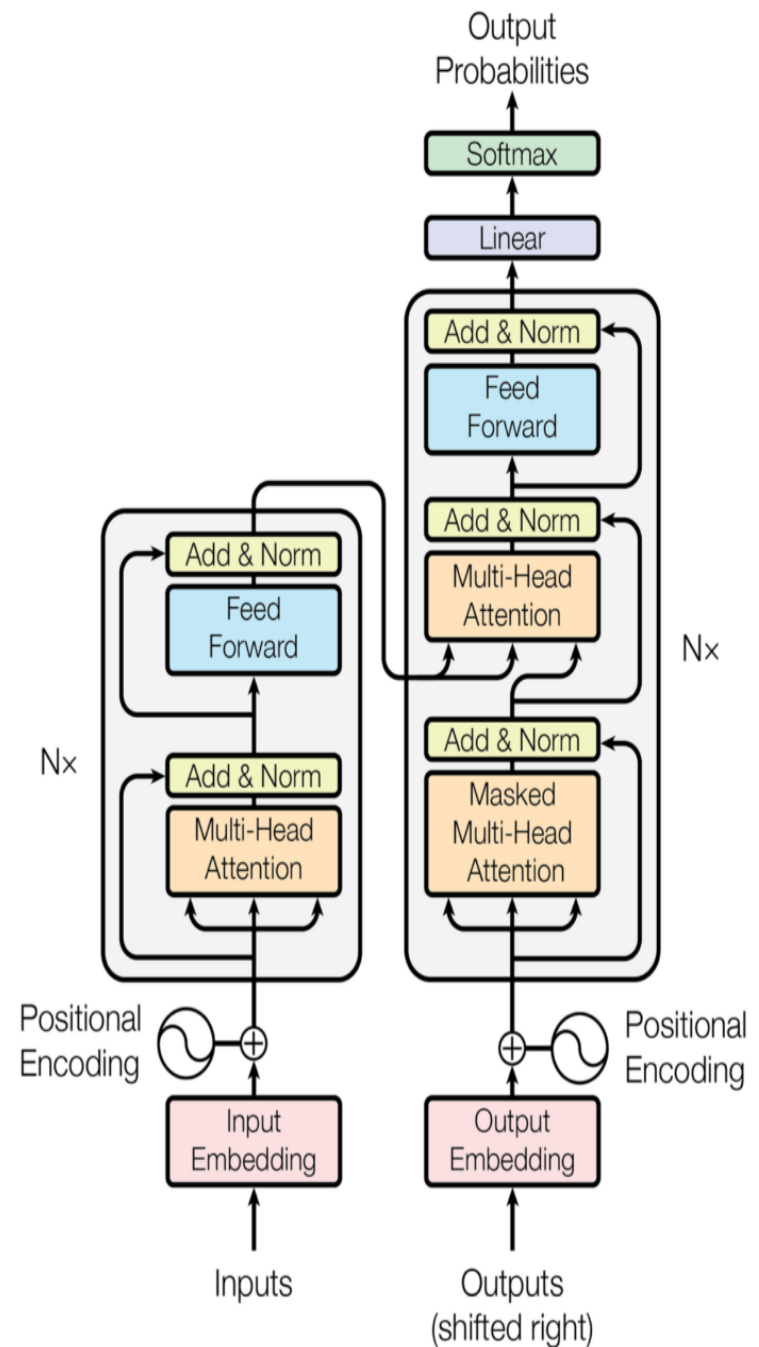


Figure 1: The Transformer - model architecture.

BERT

*Bidirectional
Representations*

*Encoder
from*

Transformers (BERT) es un modelo basado en *transformers* diseñado para crear representaciones de las secuencias condicionadas por los contextos de la capa anterior y la siguiente, esto permite conseguir resultados del estado del arte simplemente añadiendo una capa de salida para la tarea en cuestión.

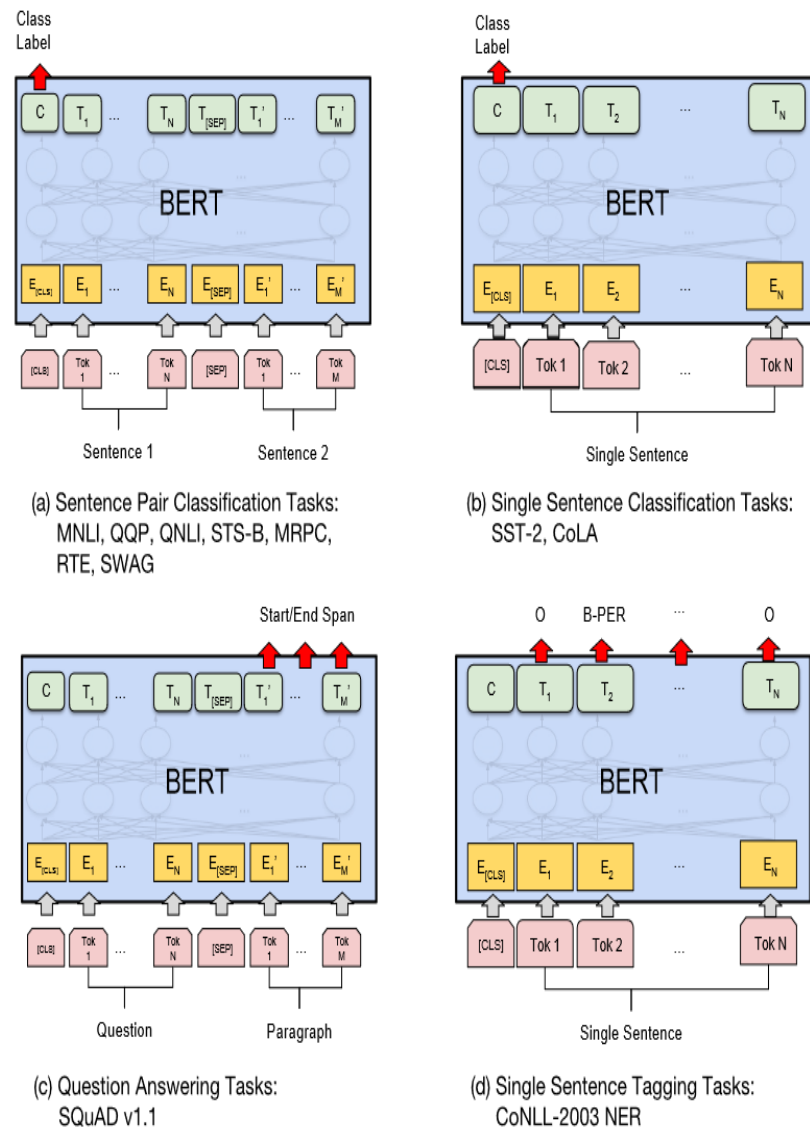


Figure 4: Illustrations of Fine-tuning BERT on Different Tasks.

BART

Esta aproximación consta de dos partes, en primer lugar, se corrompe el texto de entrada y después se entrena un modelo para que aprenda a reconstruirlo. Este usa una arquitectura basada en *transformers* lo que provoca que sea altamente paralelizable. Una de las novedades que presenta BART es el uso de un *encoder* bidireccional y un *decoder* con flujo de izquierda a derecha. Esto puede verse como una generalización de BERT, GPT y otras arquitecturas recientes.

Transformers

- Get To The Point
- PEGASUS
- ProphetNet
- ERNIE-GEN

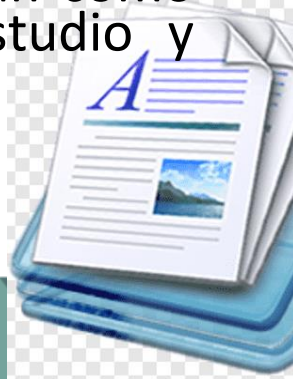
Conclusiones

- La teoría de Redes Neuronales Artificiales presenta grandes ventajas con respecto a otros modelos típicos de solución de problemas de Ingeniería, una de ellas es su inspiración en modelos biológicos del funcionamiento del cerebro, lo que facilita su estudio debido a las analogías que pueden introducirse para su análisis.
- Las redes neuronales son una teoría relativamente nueva que junto a otras técnicas de inteligencia artificial ha generado soluciones muy confiables a problemas de Ingeniería, los cuales, a pesar de poder ser solucionados por métodos tradicionales, encuentran en las redes neuronales una alternativa fácil de implementar y altamente segura.



Conclusiones

- La red tipo Perceptrón es una red que puede implementarse exitosamente para resolver problemas de clasificación de patrones que sean linealmente separables, la red responderá mejor entre más sencillos sean los patrones que debe clasificar. A pesar de que cuenta con serias limitaciones, esta red conserva su importancia ya que sirvió como inspiración para otros tipos de redes, como por ejemplo las redes multicapa.
- La principal ventaja de las redes neuronales es su capacidad para aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante; en términos generales las redes neuronales son una teoría relativamente nueva y como tal presentan aún algunas limitaciones, pero su facilidad de implementación y la calidad en la información que entregan como respuesta, son la motivación suficiente para que su estudio y desarrollo continúe.



Contacto

Luis Enrique Argota Vega

Máster en Ciencia e Ingeniería de la Computación

luiso91@gmx.com

Tels: 5578050838

Redes sociales:



<https://cutt.ly/ifPyTEH>



<https://cutt.ly/WfPtYZz>