Procesamiento de Lenguaje Natural o Minería de textos
Tema 7: PLN con Redes Neuronales.

Objetivo: El participante identificará el método de redes neuronales para la clasificación de textos, a partir de las bibliotecas y servicios implementados

Temario:

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*

Lecturas:

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

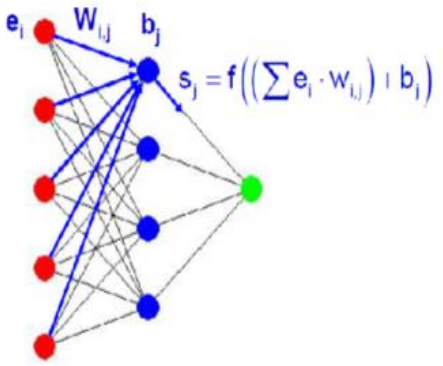
Peirano, M. D. (2020). Resumen de Textos con Modelos Secuencia-a-Secuencia: Varias Aproximaciones (Doctoral dissertation). Obtenido de: https://riunet.upv.es/handle/10251/150240

Introducción

Las redes de neuronas artificiales (RNA) son un tipo de modelo de aprendizaje automático, caracterizado por su inspiración en las estructuras biológicas a las que hacen referencia. Su estructura consiste en una red formada por nodos (o ***neuronas***) y ***conexiones***, razón por la cual se asemejan al cerebro de los seres humanos, del cual procede su nombre. Las redes neuronales se aplican en diversidad de problemas de reconocimiento de patrones y de aproximación de funciones, debido a su flexibilidad y facilidad de uso.

Características red neuronal

Una red neuronal es 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.



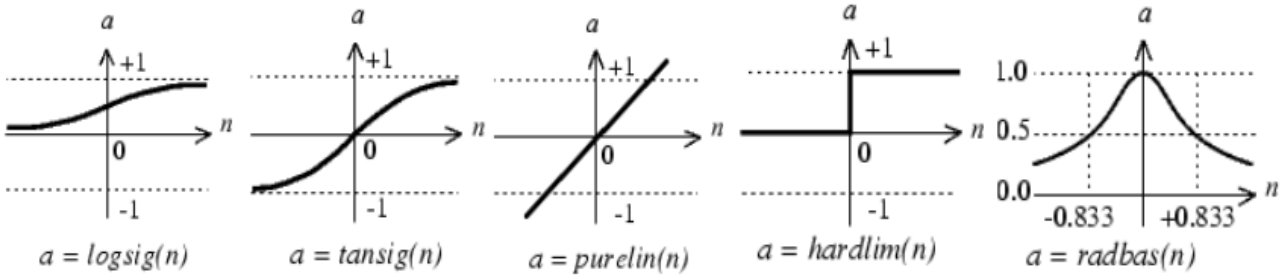
Las neuronas se pueden disponer en diferentes ***capas***. Las redes neuronales más sencillas constan de una capa de entrada, una capa de neuronas o capa oculta, y una capa de salida (Figura 1).

Ejemplo de red neuronal con una capa oculta

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 una función de propagación, una función de activación, y una función de transferencia.

- 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 ($hardlim$) $[0,1]$.



Funciones de transferencia

Tipos de redes neuronales

Los criterios más importantes para clasificar las redes neuronales son:

- Según el tipo de conexiones:

  - Redes de propagación hacia delante (***feed - forward***), donde las conexiones van en un solo sentido desde la capa de entrada hacia la capa de salida.
  - Redes ***recurrentes***, donde las conexiones pueden realizar ciclos.
- Según el tipo de aprendizaje
  - Aprendizaje ***supervisado***. Los datos (o entradas) tienen una respuesta conocida (o salida), con la cual se ajusta o entrena la red neuronal.
  - Aprendizaje ***no supervisado o autoorganizado***. Los datos son solamente entradas. Son redes empleadas fundamentalmente para clasificación y reconocimiento de patrones.

Ventajas y desventajas de las RNA

Ventajas

Las RNA tienen muchas ventajas debido a que están basadas en la estructura del sistema nervioso, principalmente el cerebro.

- ***Aprendizaje:*** Las RNA tienen la habilidad de aprender mediante una etapa que se llama etapa de aprendizaje. Esta consiste en proporcionar a la RNA datos como entrada a su vez que se le indica cuál es la salida (respuesta) esperada.
- ***Auto organización:*** Una RNA crea su propia representación de la información en su interior, descartando al usuario de esto.
- ***Tolerancia a fallos:*** Debido a que una RNA almacena la información de forma redundante, ésta puede seguir respondiendo de manera aceptable aun si se daña parcialmente.
- ***Flexibilidad:*** Una RNA puede manejar cambios no importantes en la información de entrada, como señales con ruido u otros cambios en la entrada (ej. si la información de entrada es la imagen de un objeto, la respuesta correspondiente no sufre cambios si la imagen cambia un poco su brillo o el objeto cambia ligeramente)
- ***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.

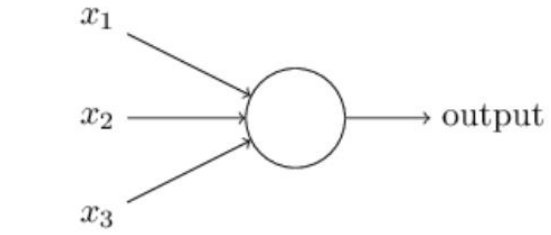
Hay muchas buenas razones para el uso de RN y los avances en este campo incrementarán su popularidad. Son excelentes como clasificadores/reconocedores de patrones – y pueden ser usadas donde las técnicas tradicionales no funcionan. Las RN pueden manejar excepciones y entradas de datos anormales, muy importante para sistemas que manejan un amplio rango de datos (sistemas de radar y sonar, por ejemplo). Muchas RN son biológicamente plausibles, lo que significa que pueden proveer pistas de cómo trabaja el cerebro según progresen. Avances en la neurociencia también ayudarán al avance en las RN y hasta el punto en que sean capaces de clasificar objetos con la precisión de un humano y la velocidad de una computadora.

Desventajas

- Complejidad de aprendizaje para grandes tareas, cuantas más cosas se necesiten que aprenda una red, más complicado será enseñarle.
- Tiempo de aprendizaje elevado. Esto depende de dos factores: primero si se incrementa la cantidad de patrones a identificar o clasificar y segundo si se requiere mayor flexibilidad o capacidad de adaptación de la RN para reconocer patrones que sean sumamente parecidos, se deberá invertir más tiempo en lograr que la red converja a valores de pesos que representen lo que se quiera enseñar.
- No permite interpretar lo que se ha aprendido, la red por si sola proporciona una salida, un número, que no puede ser interpretado por ella misma, sino que se requiere de la intervención del programador y de la aplicación en si para encontrarle un significado a la salida proporcionada.
- Elevada cantidad de datos para el entrenamiento, cuanto más flexible se requiera que sea la RN, más información tendrá que enseñarle para que realice de forma adecuada la identificación.
- Otros problemas con las redes neuronales son la falta de reglas definitorias que ayuden a realizar una red para un problema dado.

Redes neuronales para clasificación de textos

La red más básica en el conjunto de redes de neuronas se denomina ***Perceptrón***. 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 (Nielsen, 2017[1]). La salida tomará un valor u otro en función de los pesos de cada entrada y del valor de esta, además de un valor conocido como umbral, que define el punto en el que un valor obtenido a partir de las entradas producirá una salida u otra.



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

Diagrama de Perceptrón Simple y función de salida

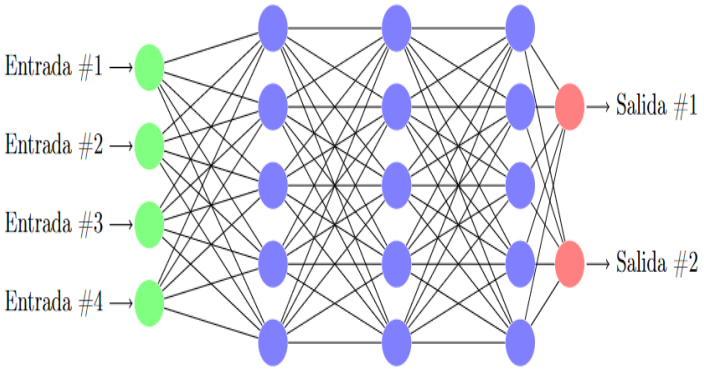
Esta función matemática indica el valor de la salida dependiente de si el sumatorio de 3 entradas binarias con su peso asignado supera el valor del umbral (*threshold*) definido. Este modelo, en base a características del texto como el vector de la bolsa de palabras puede clasificar textos para una categoría. Ampliando el modelo con perceptrón es independientes para cada categoría se tendría un modelo de clasificación válido. Pero estos perceptrones tienen una limitación, solo pueden separar linealmente, así que solo serán capaces de clasificar correctamente nuestros textos si en el espacio vectorial de nuestros vectores caracterizadores, las categorías son separables del resto por hiperplanos. La mayoría de las veces, especialmente con texto, esto no se cumple, así que debemos buscar modelos que puedan capturar una mayor complejidad. Siguiendo este modelo sencillo, el ***Perceptrón Multicapa*** implementa una versión ligeramente más compleja para acomodar diversidad de entradas y salidas.

[1] Nielsen, M. (May de 2017). *Chapter 1 Using neural nets to recognize handwritten digits*. Obtenido de *Neural Networks and Deep learning*: http://neuralnetworksanddeeplearning.com/chap1.html

Similar a como se ha definido al Perceptrón Simple, el ***Perceptrón Multicapa*** (***Multi-Layer Perceptron, MLP***) añade profundidad a este modelo de red neuronal, incluyendo varias "capas" de neuronas entre las neuronas de entrada y la neurona o neuronas de salida. La diferencia fundamental entre este Perceptrón y su análogo más sencillo es que 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 que un simple suma producto.

Este modelo de red se puede adaptar a casi cualquier tipo de problema, dado que el MLP es considerado un aproximador universal, independientemente del número de capas ocultas que posea. 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.

Las capas intermedias, llamadas capas ocultas, son las capas entre la capa de entrada y la de salida. Se les denomina capas ocultas porque son las encargadas de traducir los valores de las entradas mediante funciones de transformación no lineales, buscando "simplificar" el problema para la siguiente o siguientes capas, permitiendo que un problema inicialmente complejo se reduzca a problemas resolubles con funciones más sencillas. Básicamente, una red MLP puede tratar problemas no lineales.



Ejercicio7(es)-PLN con Redes Neuronales.ipynb

Modelos Secuencia a Secuencia

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

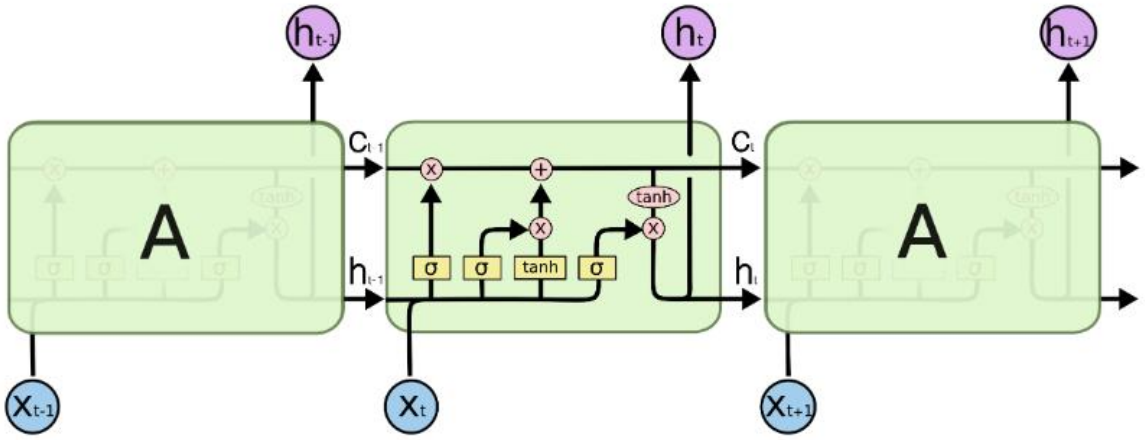
Redes neuronales recurrentes para modelos del lenguaje

Las redes neuronales recurrentes (RNR) 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 $n + 1$. Esto permite propagar la información durante varios pasos dentro de la secuencia procesada. Esto las convierte en buenas candidatas para trabajar con datos con dependencias estructurales en una dimensión, como el texto, o el audio.

Una manera bastante clara de entenderlas es desenrollarlas en el tiempo y observar su similitud con las redes neuronales tradicionales. Este desenrollado también nos ayuda con su entrenamiento, que se puede realizar con BPTT (*BackPropagation Through Time*). Pero este tipo de entrenamientos sufre un problema conocido como *Vanishing gradient* (Problema de desvanecimiento de gradiente). Cuando desenrollamos el

bucle temporal de la neurona obtenemos una cadena muy larga desde la entrada a la salida. Si intentamos actualizar los pesos como se contaba antes, usando la regla de la cadena, nos daremos cuenta de que el gradiente aplicado a las neuronas más próximas a la salida es mucho mayor que el que se aplica a las neuronas más próximas a la entrada. En otras palabras, las neuronas de capas finales aprenden más rápido que las de las primeras capas. Esto es un problema ya que sin valores adecuados en las primeras capas no se pueden capturar los problemas complejos para los que estas redes fueron concebidas.

Existen al menos 20 tipos distintos de RNR, pero la predilecta hasta la fecha es la LSTM (*Long Short Term Memory*)[2], esta fue propuesta en 1997 y desde entonces la gran mayoría de modelos con RNR implementan esta variante. Esto se debe a que esta red permite conservar la información de iteraciones arbitrariamente lejanas a la actual. La LSTM 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.



Neurona lstm

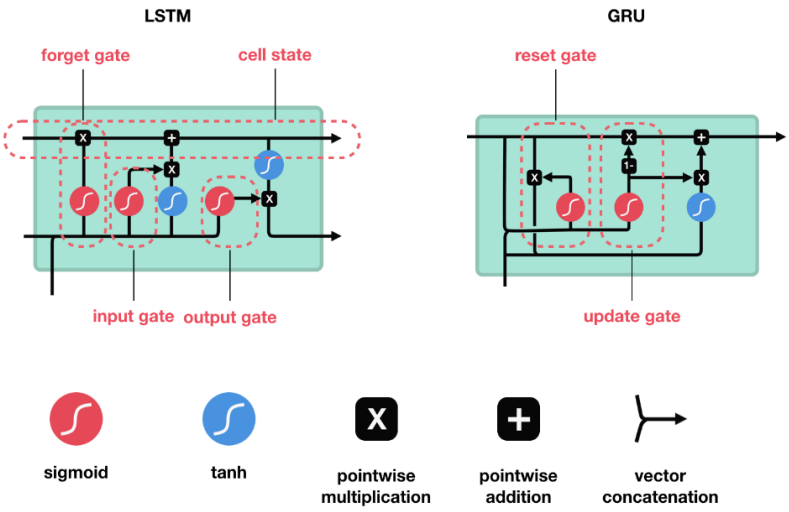
Fuente: Esquema realizado por Christopher Olah. http://colah.github.io

En 2014 *Kyunghyun Cho et al.*[3] propusieron una variante simplificada de la LSTM, la GRU, que aumentaba su eficiencia manteniendo su eficacia y solventar el problema de desaparición de gradiente. La *Gated recurrent units* (GRU) agrega dos puertas (puertas de actualización y restablecimiento, o en inglés *Update* y *Reset*) que mantienen un registro de la información más relevante para la predicción. Básicamente, son dos vectores que deciden qué información se pasa a la salida. 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.

[2] Sepp Hochreiter, Jurgen Schmidhuber LONG SHORT-TERM MEMORY

[3] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches

Ambas arquitecturas de red se utilizan para resolver los problemas de las RNN de gradiente y de memoria a corto plazo. En teoría, las celdas LSTM tienen una puerta adicional y, por lo tanto, son más complejas y tardan más en entrenarse. Esta complejidad añadida debería facilitar recordar secuencias más largas. Sin embargo, no hay evidencia empírica clara de que una de un tipo de redes supere a al otro en todos los casos. Expertos recomiendan comenzar con GRU, ya que son más simples y escalables que las celdas LSTM. Aunque se ha demostrado que la LSTM es estrictamente superior en la tarea de modelado[4], la GRU está ganando terreno al ser más eficiente y obtener los mismos resultados en cierto tipo de problemas.



Fuente: https://themachinelearners.com/modelos-secuencia/

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

Aunque se había solucionado en parte el problema de dependencias entre palabras alejadas entre sí dentro del texto, la naturaleza recurrente de estas redes hacía imposible su escalabilidad al no ser fácilmente paralelizables. Es aquí donde Vaswani et al.[5] proponen un modelo basado únicamente en mecanismos de atención, evitando cualquier tipo de recurrencia o convolución. Este modelo, además de ser menos exigente computacionalmente, mejoró el estado del arte obtenido hasta la fecha.

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

[4] Gail Weiss et al. On the Practical Computational Power of Finite Precision RNNs for Language Recognition
[5] Vaswani et al. Attention Is All You Need. URL: https://arxiv.org/abs/1706.03762

decodificador), pero es diferente de los modelos secuenciales anteriores ya que no utiliza RNN. En la imagen siguiente se pueden ver el codificador (izquierda) y decodificador (derecha) básicos de un Transformer.

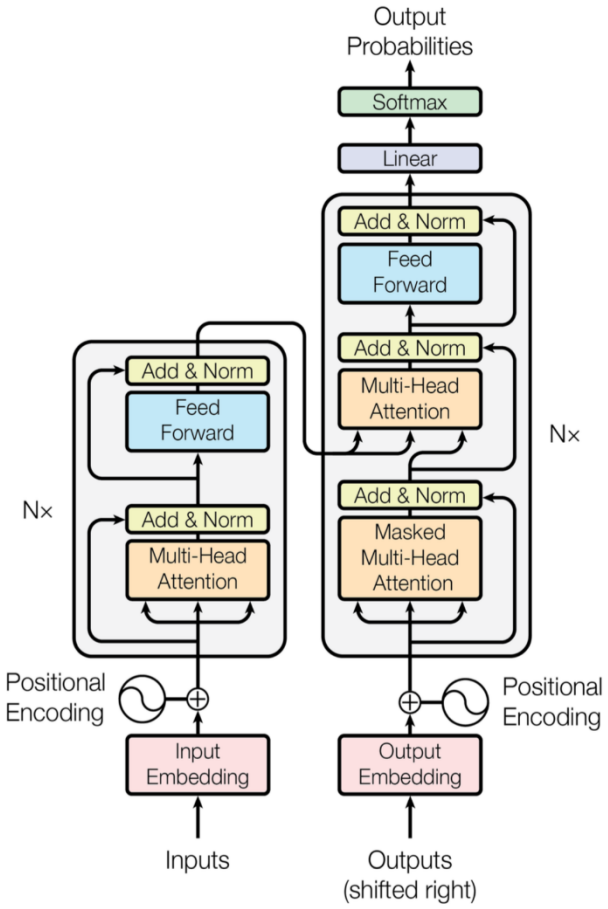


Figure 1: The Transformer - model architecture.

Fuente: https://themachinelearners.com/modelos-secuencia/

Estos modelos han avanzado mucho en los últimos años gracias a las grandes tecnológicas. Uno de los Transformers que más impacto han tenido en el segmento se trata de BERT, el algoritmo de Google que presentaron en este paper: *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*[6]. Sin embargo, el modelo más avanzado hoy en día se trata de GPT-3[7], introducido por *OpenAI*, una mejora del famoso GPT-2.

[6] Jacob Devlin et al. BERT: Pre-training of Deep Bidirectional Transformersfor Language Understanding. URL: https://arxiv.org/abs/1810.04805

[7] BROWN, Tom B., et al. Language models are few-shot learners. arXiv preprint arXiv:2005.14165, 2020. URL: https://arxiv.org/abs/2005.14165

BERT

Bidirectional Encoder Representations 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.

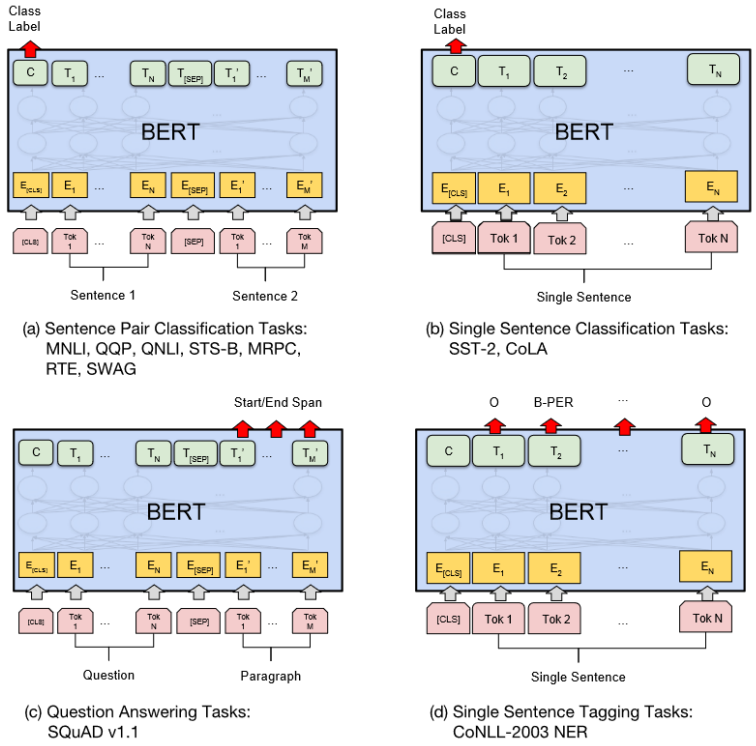


Figure 4: Illustrations of Fine-tuning BERT on Different Tasks.

Fuente: https://arxiv.org/pdf/1810.04805.pdf

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[8] 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.

Get To The Point

[8] Mike Lewis et al. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension.

Esta propuesta[9] presenta una arquitectura *pointing-generator* la cual permite copiar datos directamente del texto de entrada gracias al *pointing* a la vez que producir nuevos mediante el *generator*.

PEGASUS

La novedad que introduce *PEGASUS*[10] es enmascarar las frases importantes del documento de entrada para que el modelo aprenda a generarlas a partir del resto de frases.

ProphetNet

La principal diferencia que presenta *ProphetNet*[11] es la predicción de *n-gramas* en lugar de un único término por iteración, esto incentiva al modelo a pensar a futuro, evitando el *overfiting* en el caso de encontrar fuertes correlaciones locales.

ERNIE-GEN

ERNIE-GEN[12] propone 3 métodos para mejorar la habilidad generativa del modelo:

- *Span-by-span generation pre-training task* para que el modelo genere unidades sintácticamente completas en vez de una única palabra.
- *Infilling generation mechanism* para evitar el problema de *Exposure Bias*.
- *Multi-Granularity Target Fragments* para alentar la dependencia del *decoder* en el *encoder* en la fase de preentrenamiento.

Conclusiones[13]

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.

[9] Abigail See Get To The Point: Summarization with Pointer-Generator Net-works
[10] Jingqing Zhang et al. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization
[11] Yu Yan et al.ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training
[12] Dongling Xia et al. ERNIE-GEN: An Enhanced Multi-Flow Pre-training and Fine-tuning Framework for Natural Language Generation
[13] http://medicinaycomplejidad.org/pdf/redes/Conclusiones.pdf

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 dela 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.