Arana, Carlos

Working Paper

Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales

Serie Documentos de Trabajo, No. 797

Provided in Cooperation with:
University of CEMA, Buenos Aires

Suggested Citation: Arana, Carlos (2021) : Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales, Serie Documentos de Trabajo, No. 797, Universidad del Centro de Estudios Macroeconómicos de Argentina (UCEMA), Buenos Aires

This Version is available at:
https://hdl.handle.net/10419/238422

UNIVERSIDAD DEL CEMA
Buenos Aires
Argentina

Serie
DOCUMENTOS DE TRABAJO

Área: Ingeniería

REDES NEURONALES RECURRENTES: ANÁLISIS DE LOS MODELOS ESPECIALIZADOS EN DATOS SECUENCIALES

Carlos Arana

Junio 2021
Nro. 797

www.cema.edu.ar/publicaciones/doc_trabajo.html
UCEMA: Av. Córdoba 374, C1054AAP Buenos Aires, Argentina
ISSN 1668-4575 (impreso), ISSN 1668-4583 (en línea)
Editor: Jorge M. Streb; asistente editorial: Valeria Dowding jae@cema.edu.ar

Redes Neuronales Recurrentes: Análisis de los Modelos Especializados en Datos Secuenciales

Carlos Arana *

Mayo 2021

Resumen

El aprendizaje automático es la rama de la inteligencia artificial (IA) que se especializa en inferir patrones a partir de un conjunto datos, utilizando para ello diferentes técnicas estadísticas y heurísticas. Los métodos tradicionales de aprendizaje automático están limitados en su capacidad para el procesamiento de datos en bruto debido a la necesidad de contar con un intenso trabajo previo de procesamiento de estos para poder realizar inferencias. El aprendizaje profundo es una clase de métodos de aprendizaje automático con múltiples niveles de representación que soluciona esta dificultad. Se compone de varios módulos sencillos no lineales que generan sucesivamente representaciones de nivel superior, ligeramente más abstractas a partir de las características desarrolladas en los niveles anteriores, empezando por la entrada en bruto. En este artículo analizaremos en profundidad las características y aplicaciones de las redes neuronales recurrentes (RNN). Esta clase de estructuras de aprendizaje profundo se especializan en el tratamiento de datos de tipo secuencial, que son los caracterizan a las series temporales y a estructuras de datos como textos, videos y audio. En el artículo se presentarán y analizarán en profundidad las características y la aplicación de 3 tipos de redes neuronales profundas especializadas en este tipo de datos : redes neuronales recurrentes, redes de memoria a corto y largo plazo (LSTM) y redes recurrentes con atención.

---

* Ingeniero Industrial (UBA) y Doctor en Ingeniería Industrial (cand). Es profesor de las materias 'Métodos Cuantitativos' y 'Ciencia de Datos en Negocios' en la Maestría en Dirección de Empresas (MADE) de UCEMA. Es director del posgrado en 'Ciencia de Datos y Big Data' de la Facultad de Ingeniería de la Universidad de Buenos Aires. A su vez se desempeña como consultor especializado en Inteligencia Artificial y sus aplicaciones en negocios e industria

1. Aprendizaje Automático y Aprendizaje Profundo

Los algoritmo de machine learning, o aprendizaje automático, se basan en el aprendizaje a partir de una muestra de datos de patrones y relaciones funcionales entre distintas variables. Mitchell [1] nos ofrece la definición "Se dice que un programa de ordenador aprende de la experiencia E con respecto a una clase de tareas T y una medida de rendimiento P , si su rendimiento en las tareas de T , medido por P , mejora con la experiencia E".

Las tareas de aprendizaje automático generalmente se describen en términos de cómo el sistema de aprendizaje automático debería procesar un ejemplo. Un ejemplo, o caso en la jerga del machine learning, es una observación de la cual hemos medidas ciertas valores de variables de tipo cuantitativas y relevado alguna característica de una variable tipo categórica o cualitativa. Por lo general, representamos un ejemplo como un vector $\boldsymbol{x} \in \boldsymbol{R^n}$ donde cada entrada $\boldsymbol{x}_i$ del vector es el valor de alguna de esta variable.

1.1 Conjuntos de entrenamiento y Validación. Medidas de Performance

Para tareas como clasificación o generación de casos la precisión es simplemente la proporción de ejemplos para los que el modelo produce la salida correcta. En el caso de la clasificación conocemos el valor de la variable objetivo de cada caso utilizado para entrenar el modelo, por lo que lo podremos comparar la predicción, o la salida, del modelo con este valor correcto. A su vez, si el modelo fuera de generación de nuevos datos, al momento de entrenarlos estaríamos utilizando los mismos datos de entrenamiento para corroborar qué tan parecido al dato de entrenamiento utilizado es el nuevo dato generado.

Por lo general, nos interesa saber cómo funciona el algoritmo de aprendizaje automático con datos que no ha visto antes, ya que esto determina cómo funcionará cuando se implemente en el mundo real. Por lo tanto, evaluamos estas medidas de rendimiento utilizando un conjunto de datos que está separado de los datos utilizados para el entrenamiento del sistema de aprendizaje automático. Este procedimiento se lleva a cabo separando del set original en dos subconjuntos, siendo que en uno de ellos correremos todos los procesos asociados al entrenamiento del modelo de clasificación o generación, y en el otro sólo corroboraremos su efectividad y mediremos su performance. Esta tarea de separación y las correspondiente tareas realizadas sobre cada uno de los subconjuntos generados (entrenamiento y validación de los procesos, respectivamente) es fundamental en los procesos de aprendizaje automático, y es la clave en la que radica la efectividad de sus procedimientos como la validez y la capacidad de generalización de sus modelos.

1.2 Tipos de Aprendizaje Automático

Los algoritmos de aprendizaje automático pueden clasificarse, a grandes rasgos, como supervisados y no supervisados, en función del tipo de procesos que se les permite realizar durante la fase de entrenamiento/aprendizaje.

Los algoritmos de aprendizaje no supervisado aprenden propiedades útiles de la estructura del conjunto de datos. En el contexto del aprendizaje profundo, normalmente queremos aprender la distribución de probabilidad que generó a este conjunto de datos, ya sea explícitamente como en la estimación de la densidad o implícitamente para tareas como la generación de datos sinterizados. Algunos otros algoritmos de aprendizaje no supervisado desempeñan otras funciones, como el "Clustering", que consiste en dividir el conjunto de datos en grupos que aglutinen a ejemplos de características similares, y a su vez separe a los grupos entre sí de la mejor manera posible. A su vez, los algoritmos de aprendizaje supervisado realizan diferentes procesos sobre un conjunto de datos a partir de sus características o atributos, pero en este caso contamos con un dato fundamental: conocemos el valor variable objetivo, que en la jerga de la disciplina se suele denominar la "clase".

A grandes rasgos, el aprendizaje no supervisado consiste en observar varios ejemplos de un vector aleatorio x, e intentar aprender implícita o explícitamente la distribución de probabilidad $p(x)$, o algunas propiedades interesantes de esa distribución, mientras que el aprendizaje supervisado consiste, a partir de la observación de varios ejemplos de un vector aleatorio$x$ y un valor o vector asociado y, aprender a predecir$y$ a partir de$x$, normalmente estimando ahora función de distribución de pobabildiad condicional $p(y|x)$. El término aprendizaje supervisado tiene su origen en la idea de que el objetivo o clase "y" es proporcionado por un "supervisor" que muestra al sistema de aprendizaje automático lo que debe hacer. En el aprendizaje no supervisado, no hay supervisor, y el algoritmo debe aprender a dar sentido a los datos sin esta guía.

Una forma común de describir a un conjunto de datos u observaciones (dataset) es con una matriz de diseño. Una matriz de diseño es una matriz que contiene un ejemplo diferente en cada fila. Cada columna de la matriz corresponde a una característica diferente. Los algoritmos de aprendizaje se diferencian en función de la forma de operar y procesa los datos contenidos en una matriz de diseño.

2. Redes neuronales

Las redes neuronales artificiales (RNA) se inspiran en las estructuras neuronales biológicas que se encuentran en el cerebro humano. Estas redes contienen capas organizadas de unidades interconectadas o nodos. Las RNA se utilizan principalmente para tareas en las que es difícil derivar restricciones lógicas de forma explícita, como el reconocimiento de patrones y el análisis predictivo [2]. El primer modelo computacional de una RNA fue desarrollado en 1943 por McCulloch y Pitts [3], un neurocientífico y un lógico, respectivamente. Propusieron una unidad de umbral binario como modelo para la neurona artificial. El modelo matemático de esta unidad es el siguiente

$$y = H\left(\sum_{j=1}^{n} w_j x_j - u\right) \tag{1}$$

donde H es la función de activación (en este caso la función escalón de Heaviside) con el umbral u, x_j es la señal de entrada y w_j es el peso asociado con $j = 1,2,\ldots,n$, donde n corresponde al número de entradas. La salida de esta unidad es 1 cuando la suma está por encima del umbral u, y es 0 en caso contrario. Este modelo llevó a Rosenblatt [4] a desarrollar una red neuronal pionera conocida como perceptrón.

El modelo actual de RNA consiste en una suma ponderada de las entradas y un sesgo (bias). Esta suma se somete entonces a una función no lineal para producir la salida del nodo o neurona, como puede verse en la figura Fig. 1.

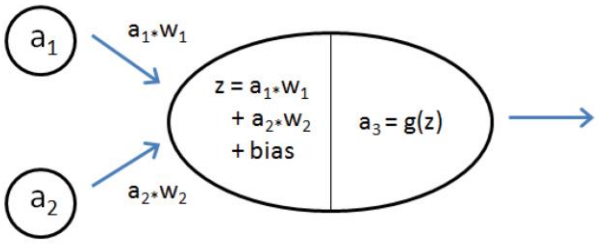


Fig. 1. Estructura de un nodo o neurona de una RNA

Como ya se ha mencionado, una RNA es un conjunto de neuronas organizadas en capas, que puede analizarse como un grafo dirigido ponderado. Las neuronas son los nodos del grafo, y sus aristas las conexiones entre la salida de una neurona y la entrada de otra. Las capas de una

RNA del tipo perceptrón son: una capa de entrada, una capa oculta y una capa de salida. Tenga en cuenta que cuando hay más de una capa oculta, la denominación de la red cambia y se convierte en Redes Neuronales Profundas. En la figura 2 se puede comprobar la organización de las capas de las RNA interconectadas.

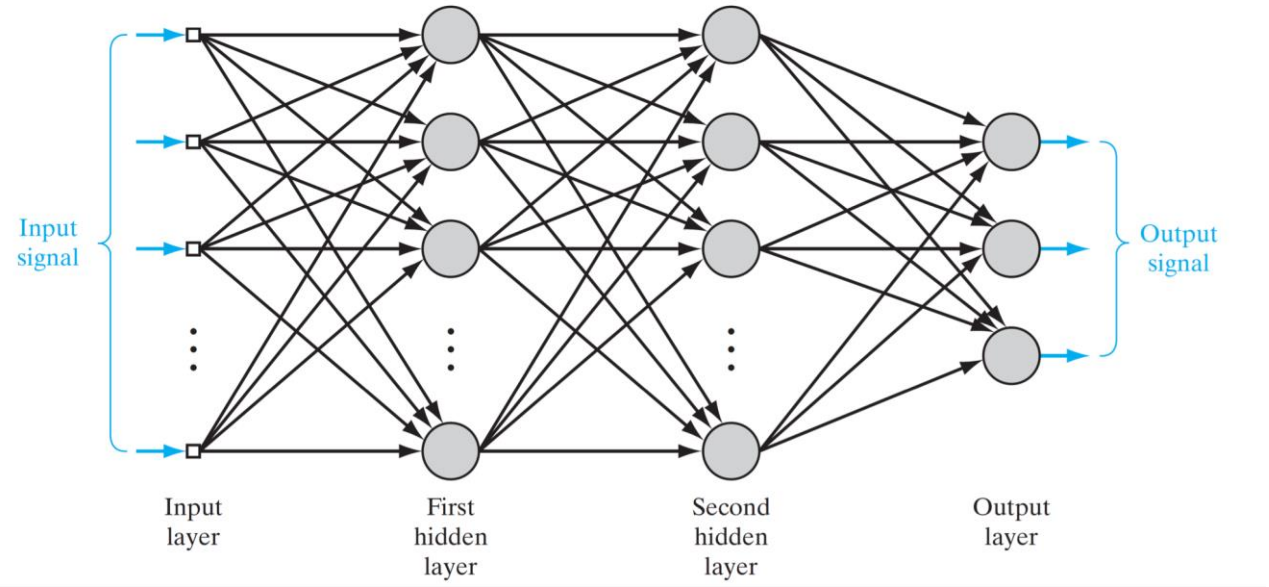


Figura 2 . Red Neuronal Artificial multicapa

3. Redes neuronales recurrentes (RNN)

Las redes neuronales recurrentes (RNN) son una clase de aprendizaje profundo basada en los trabajos de David Rumelhart en 1986. Las RNN son conocidas por su capacidad para procesar y obtener información de datos secuenciales. Por lo tanto, el análisis de vídeo, la subtitulación de imágenes, el procesamiento del lenguaje natural (PLN) y el análisis de la música dependen de las capacidades de las redes neuronales recurrentes. A diferencia de las redes neuronales artificiales ya vistas, que asumen la independencia entre los datos de entrada, las RNN capturan activamente sus dependencias secuenciales y temporales.

Uno de los atributos más definitorios de las RNN es la compartición de sus parámetros. Sin compartir parámetros, el modelo asignaría parámetros únicos para representar a cada dato en una secuencia y, por lo tanto, no podría realizar inferencias sobre secuencias de longitud variable. El

impacto de esta limitación puede observarse de forma notoria en el procesamiento del lenguaje natural. Una red multicapa tradicional en una red multicapa tradicional fallaría porque crearía una interpretación del lenguaje con respecto a los parámetros únicos establecidos para cada posición (palabra) en una frase. Las RNN, sin embargo, serían más adecuadas para la tarea, ya que comparten pesos entre los datos espaciados secuencialmente, en el ejempl0o del lenguaje, las palabras de nuestra frase

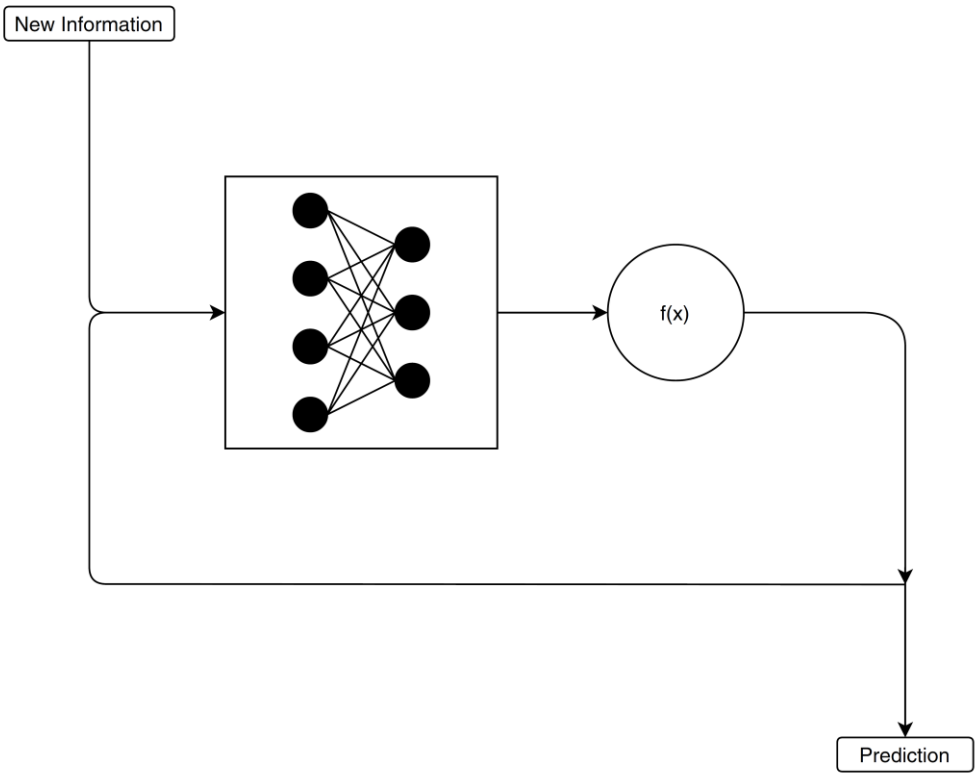


Fig. 3. Diagrama de grafo cíclico de una Red Neuronal Recurrente (Imagen extraída de [19])

Las RNN generalmente aumentan la arquitectura de red multicapa convencional con la adición de ciclos que conectan nodos adyacentes o pasos de tiempo. Estos ciclos constituyen la memoria interna de la red que se utiliza para evaluar las propiedades del dato actual con respecto a los datos del pasado inmediato. También es importante tener en cuenta que la mayoría de las redes neuronales convencionales del tipo perceptrón (también llamadas *feedforward*) están limitadas a un mapeo uno a uno de la entrada y la salida [5]. Las RNN, sin embargo, pueden realizar mapeos de uno a muchos, de muchos a muchos (por ejemplo, traducir el habla) y de muchos a uno (por ejemplo, identificar la voz). Se utiliza un grafo computacional para representar los mapeos entre las

entradas y las salidas y la pérdida. Al desplegar el gráfico en una cadena de eventos se obtiene una imagen clara del reparto de parámetros dentro de la red, como se ve en la Figura 4.

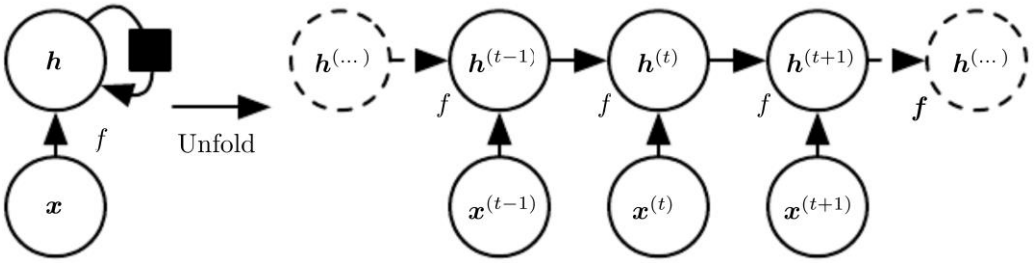


Figura 4 (Imagen extraída de [5])

La ecuación generalizada para las relaciones de recurrencia es $s^{(t)}$,

$$s^{(t)} = f(s^{t-1})) \tag{2}$$

que indica el estado del sistema que depende de un paso de tiempo anterior indicado por *t -1*.
Esta ecuación puede reescribirse como $h^{(t)}$

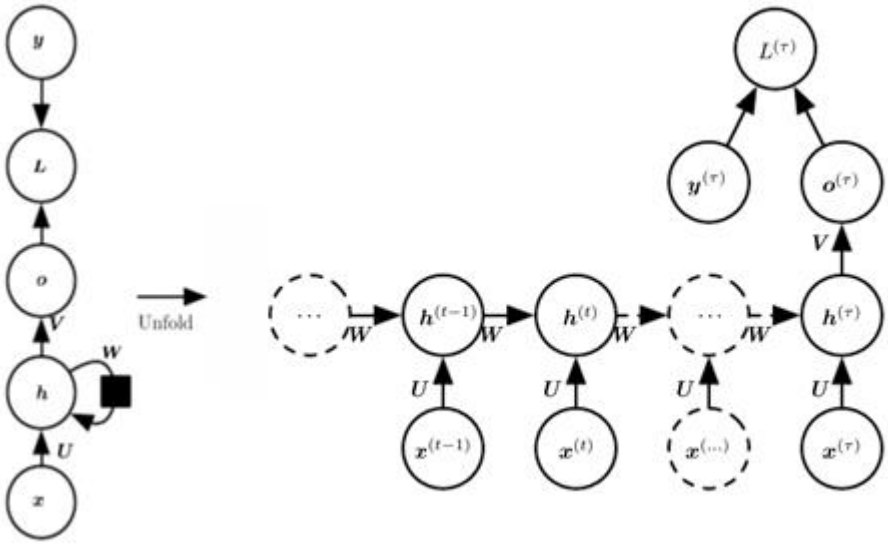
$$h^{(t)} = f(h^{t-1}, x^{(t)}) \tag{3}$$

y $x^{(t}$ representa la entrada de una instancia de tiempo en particular. La importancia de $h^{(t)}$es que es una representación de los aspectos relevantes de la secuencia pasada de entradas hasta t

3.1 Topologías de redes neuronales recurrentes

Dependiendo del problema que se trate de resolver, la red podrá trabajar con diferentes tipos de secuencias de entrada y de salida. Por ejemplo, analicemos el caso de un problema de análisis de series temporales en un mercado, en el que la red deberá tomar como entrada el valor de ciertas acciones en los últimos días y predecir cual será su valor en el día *n+1*. Otra variante seria el caso de predecir los valores para cada uno de los días, basándose en el valor del día anterior. Entre otros, estos dos ejemplos dan una perspectiva de lo que se puede conseguir con redes neuronales recurrentes si se elige la topología adecuada. Se pueden definir cuatro topologías principales: *sequence-vector, sequence-sequence, vector-sequence, Encoder-Decoder*

La Figura 5 muestra la topología sequence-vector. La imagen muestra una capa de neuronas recurrentes desarrollada a lo largo del tiempo en la que sólo la salida del último time step es tomada en cuenta como salida de la capa.



.Figura 5: Topología tipo sequence-vector. (Imagen extraída de [5])

En la Figura 6 se presenta la topología sequence-sequence, donde la capa devuelve el resultado procesado en cada período de tiempo. Usualmente, las redes neuronales recurrentes profundas que buscan un comportamiento tipo sequence-vector se diseñan acoplando varias capas tipo sequence-sequence y una última capa de tipo sequence-vector. Como consecuencia de esta estructura, la salida de la red será un único elemento que se calculará en función del resto de time steps del lote.

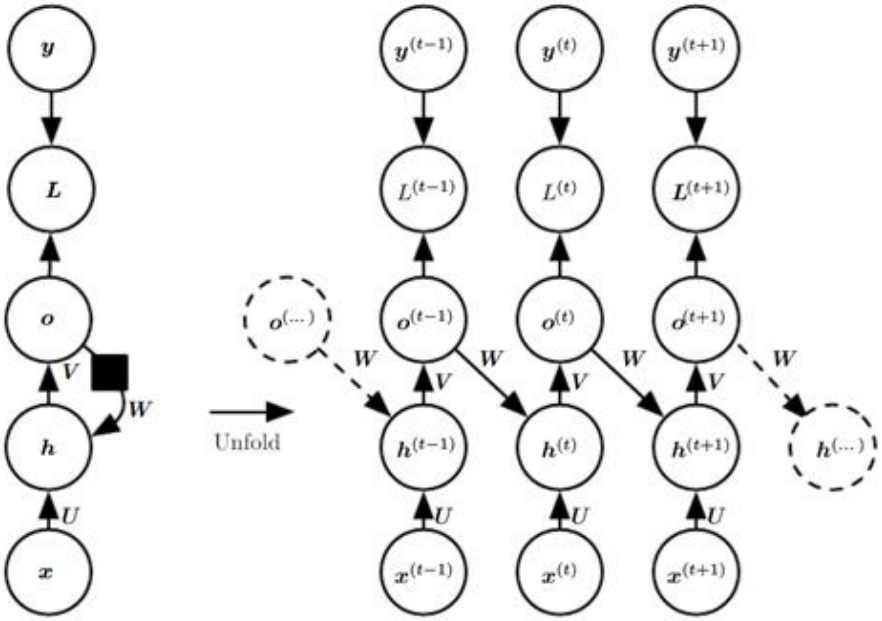


Figura 6. Topología de RNN sequence-sequence (Imagen extraída de [5])

La última de las arquitecturas, Encoder-Decoder [6],deriva de la unión de sequence-vector y vector-sequence. Como puede verse en la Figura 7, la primera parte del modelo, el Encoder, toma como entrada una secuencia generando como última instancia una representación vectorial de la misma (modelo sequence-vector). La segunda parte del modelo, el Decoder, toma dicha representación vectorial como entrada $h(t)$ y va devolviendo el resultado del procesamiento en cada período de tiempo. El modelo global puede verse como un sequence-sequence en el cual la longitud de la secuencia de entrada puede variar con respecto a la de la secuencia de salida. Esto es útil en problemas como la traducción automática, en el que una frase en un idioma determinado puede tener más símbolos o palabras que su homóloga en otro idioma diferente. Este tipo de modelos están teniendo una relevancia significativa en el campo del Procesamiento del Lenguaje Natural y en el de la composición automática de musica, sobre todo desde la aparición de los mecanismos de Attention, cuya inclusión en estos modelos resultó en las arquitecturas Transformer [7].

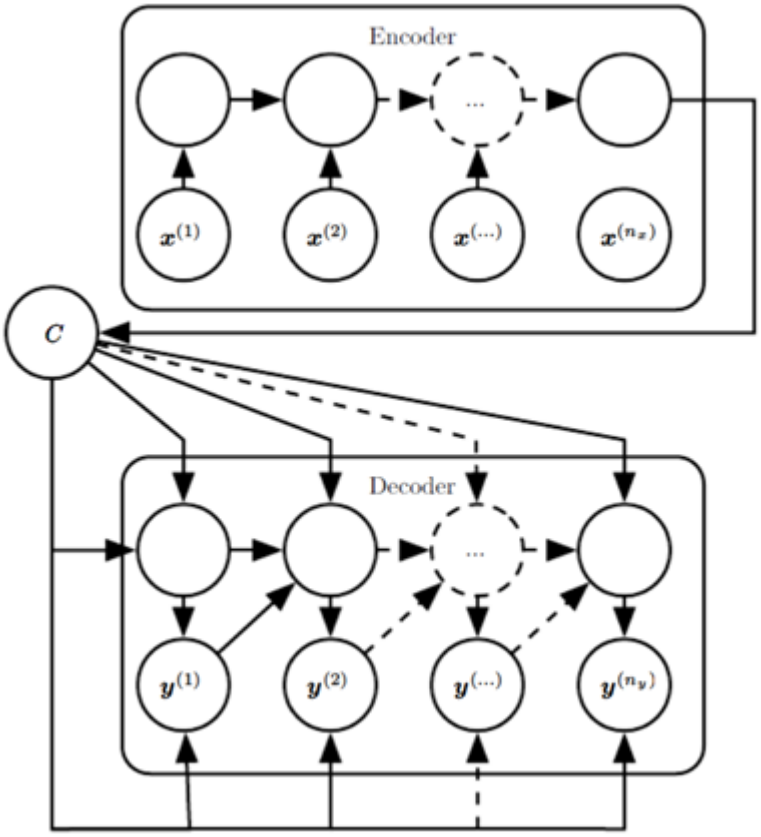


Figura 7: Topología tipo Encode-Decoder. (Imagen extraída de [5])

3.2 El problema de las redes neuronales recurrentes

Hasta este momento hemos visto cómo funcionan las celdas de memoria más básicas, las celdas RNN típicas. Este tipo de estructuras tienen, sin embargo, un problema llamados *short-term memory* [8]. Este problema deriva del bien conocido problema asociado al desvanecimiento del gradiente . Como se ha explicado hasta ahora, las redes neuronales recurrentes basan su funcionamiento en el procesado de secuencias de información, agregando al cómputo de cada elemento de la secuencia, el resultado del análisis del elemento anterior. Es decir, el resultado final de una RNN será una función que agregará el procesamiento de todos los elementos de la secuencia. Sin embargo, a medida que la red procesa más elementos de la cadena, tiene problemas en recordar información pasada.

Este problema surge como consecuencia del proceso matemático mediante el cual las redes neuronales recurrentes son capaces de entrenarse a partir de aros de un corpus: el algoritmo de

propagación hacia atrás en el tiempo o *back-propagation through time* [9]. A medida que la RNN recibe elementos de la secuencia, y por tanto se desarrolla a lo largo de los time steps, el gradiente del error se propaga hacia atrás del mismo modo que lo haría en un percepción multicapa clásico. Del mismo modo que sucede en ese tipo de redes, a medida que dicho gradiente alcanza las capas más cercanas al input (en el caso de las RNN, a los primeros períodos procesados), decae exponencialmente. Es por esto que las RNN más sencillas no son capaces de aprender patrones muy extendidos en el tiempo, si no que solo son eficaces en rangos cortos, como por ejemplo secuencias de 10 elementos.

Para mitigar el problema de *short-term memory,* aparece un nuevo tipo de celda de memoria que sí es capaz de extraer patrones de secuencias de mayor longitud. Esta celda más compleja se conoce como celdas de Memorias de Corto y Largo Plazo (LSTM, por sus siglas en inglés de *Long-Short Term Memory*)

4. Redes de Memoria Corta y Larga LSTM

Las redes neuronales de memoria de corto y largo plazo (LSTM) son un tipo particular de RNN que solucionan el problema de las RNN asociado a la memoria de corto plazo: el desvanecimiento o decaimiento del gradiente, y su explosión. Veremos como se supera esta dificultad mediante la nueva celda del tipo LSTM. En la figura 8 todo lo que hay en el recuadro punteado corresponde a una sola unidad de la red. En primer lugar, hay que tener en cuenta que hemos mostrado una sola componente de una LSTM , por lo que a la izquierda tenemos la información procedente del procesamiento del dato asociado al período, utilizando para ello dos estructuras de datos en este caso del tipo tensor, que son arreglos de datos de más de dos dimensiones. En la parte inferior tenemos la entrada de con información de la unidad anterior. A la derecha tenemos dos tensores que salen para informar a la siguiente unidad de tiempo y, como en las RNN simples, tenemos esta información “hacia arriba" en el diagrama para predecir la siguiente palabra y la pérdida (parte superior lado derecho).

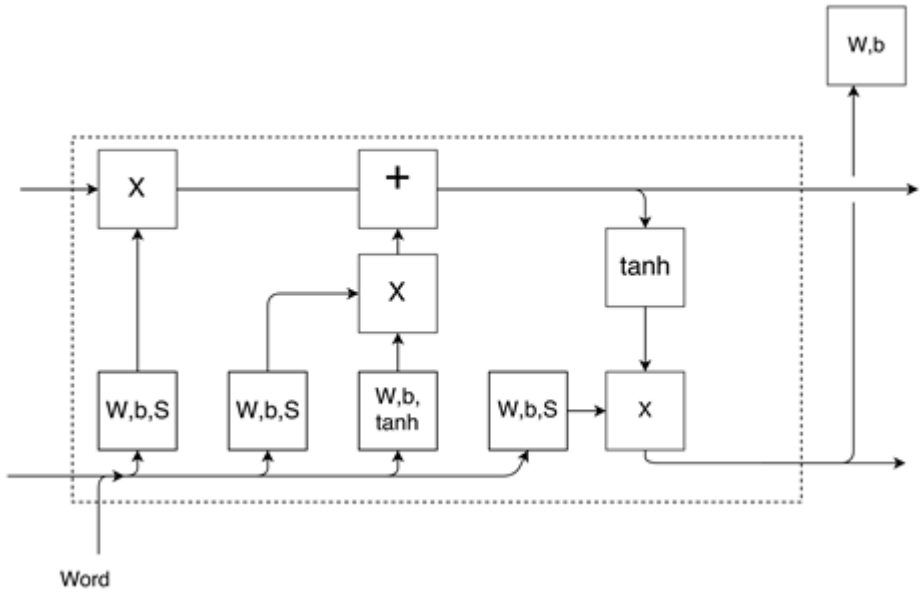


Figura 8 - Arquitectura de una celda LSTM (Imagen extraída de [19])

El objetivo es mejorar la memoria de la RNN de los eventos pasados entrenándola para que recuerde lo importante y olvide el resto. Para ello, las LSTM procesan dos versiones del pasado. La memoria selectiva "social" está en la parte superior y una versión más local en la parte inferior. La línea de tiempo de la memoria superior se llama el estado de la célula y se abrevia *c*. La línea inferior se llama *h*.

La figura 8 introduce varias conectivas y funciones de activación nuevas. En primer lugar, vemos que la línea de memoria se modifica en dos lugares antes de ser pasar a la siguiente unidad de tiempo. Están etiquetados como tiempos **X**, y $+$. La idea es que las memorias se eliminan en la unidad de tiempo **X** y se añaden en la unidad $+$. ¿Por qué decimos esto? Mira ahora la incrustación de la información actual que viene en la parte inferior izquierda. Pasa por una capa de unidades lineales seguida de una función de activación sigmoidea, como indica la anotación **W, b, S**. **W** y **b** forman la combinación lineal y **S** es la función sigmoidea de una neurona clásica. En notación matemática tenemos la operación:

$$\mathbf{h'} = \mathbf{h_t} \cdot \mathbf{e} \tag{4}$$

$$\mathbf{f} = S((\mathbf{h'}\mathbf{W_f} + \mathbf{b_f}) \tag{5}$$

Utilizamos un punto central para indicar la concatenación de vectores. Para repetir, en la parte inferior izquierda concatenamos la línea h anterior h_t y el dato actual e para obtener h_0, que a su vez se introduce en la unidad lineal de "olvido" (seguida de una sigmoidea) para producir f, la señal de olvido que se desplaza hacia arriba en el lado izquierdo de la figura. La salida de la sigmoidea se multiplica elemento a elemento memoria que viene de a parte superior izquierda. (Por "elemento a elemento" queremos decir que, por ejemplo, el $x_{[i,j]}$ a elemento de una matriz se multiplica por el $y_{[i,j]}$ elemento de la otra). La ecuación de esta operación se representa así:

$$\mathbf{c'_t} = \mathbf{c_t} \odot \mathbf{f} \tag{6}$$

Dado que los sigmoides están limitados por cero y uno, el resultado de la multiplicación debe ser una reducción en el valor absoluto en cada punto de la memoria principal. Esto corresponde al "olvido". En general, esta conjuración, sigmoidea que alimenta una multiplicación es un patrón común cuando se quiere una compuerta "suave".

Contrasta esto con lo que sucede en la unidad aditiva con que se encuentra la memoria. De nuevo, el siguiente dato, que ha llegado desde abajo izquierda, pasa ahora por separado a través de dos capas lineales, una con una con activación sigmoidea y otra con la función de activación Tanh, como se muestra en la figura 9. Tanh significa tangente hiperbólica.

$$\mathbf{a_1} = S(\mathbf{h'W_{a_1}} + \mathbf{b_{a_1}}) \tag{7}$$

$$\mathbf{a_2} = \tanh((\mathbf{h_t} \cdot \mathbf{e})\mathbf{W_{a_2}} + \mathbf{b_{a_2}}) \tag{8}$$

Es importante que, a diferencia de la función sigmoidea, Tanh puede tener como salidas tanto valores positivos o negativos, por lo que puede arrojar valores nuevos en lugar de sólo escalar el dato de entrada. El resultado de esto se añade al estado de la celda en la celda etiquetada "+":

$$\mathbf{c_{t+1}} = \mathbf{c'_t} \oplus (\mathbf{a_1} \odot \mathbf{a_2}) \tag{9}$$

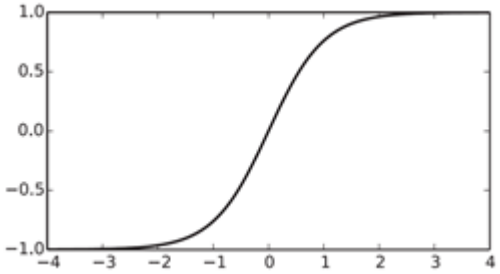


Figura 9. Función Tanh (tangente hiperbólica)

Después de esto, la línea de memoria se divide. Una copia sale por la derecha, y una copia pasa por un Tanh y luego se combina con una transformación lineal de la historia y el valor actual, para convertirse en la nueva línea h de la parte inferior:

$$\mathbf{h''} = \mathbf{h'}\mathbf{W_h} + b_h \tag{10}$$

$$\mathbf{h_{t+1}} = \mathbf{h''} \odot a_2 \tag{11}$$

Esta se concatenará con la siguiente entrada, y el proceso se repetirá. El punto a destacar aquí es que la línea de memoria de celdas nunca pasa directamente por las unidades lineales. La memoria se irá desvirtuando (se "olvidan") en la unidad "X" y se añadirá nueva información del dato actual en "\+", pero no habrá operaciones matemáticas ni de combinación lineal ni de escalamiento/transformación no lineal.

4.1 Mecanismos de *Attention*

El mecanismo de Attention fue presentado en 2014 por Dzmitry Bahdanauet al como extensión para mejorar el rendimiento del modelo Encoder Decoder en la tarea de la traducción automática [11]. En un encoder decoder convencional, el encoder representa mediante un vector de tamaño fijo (estado oculto final) toda la información de la frase de entrada. En palabras de los propios autores, "el rendimiento de un encoder decoder básico decae rápidamente a medida que la longitud de la cadena de entrada crece.". El mecanismo de *Attention* pretende solventar precisamente este problema. Mediante esta técnica, un conjunto de vectores es generado en lugar de un solo vector de dimensión fija. Entre dichos vectores se selecciona un subconjunto cercano al término que se pretende traducir en cada time step.

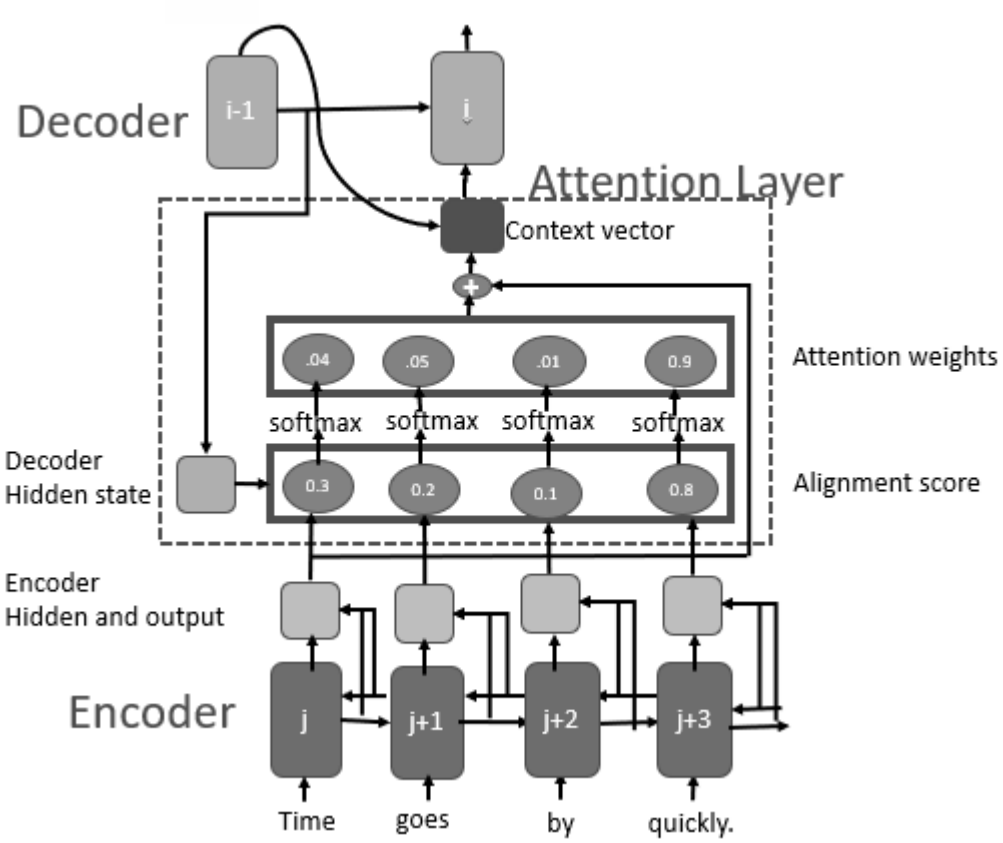


Figura 11: Arquitectura de attention (Imagen extraída de https://towardsdatascience.com/sequence-2-sequence-model-with-attention-mechanism-9e9ca2a613a)

En la Figura 11, puede verse un esquema de la arquitectura de un *encoder-decoder* intentando generar la t-ésima palabra objetivo y_t dada la frase $(x_1, x_2, \ldots, x_T)$. Como se ve, en lugar de mantener como vector final el estado oculto de la celda en el último step del encoder, se consideran todas las salidas intermedias hasta el momento y se computa la suma ponderada de dichos vectores intermedios. Este cálculo permite determinar qué palabras serán más relevantes para el decoder en ese step.

Existen varias propuestas basadas en este mecanismo:

- **Visual attention:** Fue propuesto en el año 2015 por Xuet et al. [12]. El objetivo de esta propuesta consistía en un sistema capaz de alinear la imagen de entrada y producir una palabra que la describiese como salida. La arquitectura consistía en una red convolucional que extraía características de la imagen y a continuación una red recurrente con mecanismos de atención para dar como salida la palabra. El mecanismo de attention consigue focalizarse

en ciertos elementos de la imagen quela definen de manera que la frase de salida se corresponda con dichos elementos.

- **Herarchical attention:** Esta propuesta llego en 2016 de la mano de Yanget al [13]. Se trata de un sistema de clasificación de documentos que aplica los mecanismos de attention en dos niveles. La arquitectura presenta dos módulos encoder decoder + attention. El primero de ellos aplicado a nivel de frase y el segundo a nivel de palabra.

- **Transformer**: La consecuencia más interesante de la aparición de los mecanismos de attention fue la aparición de los modelos Transformer, propuestos por Vaswaniet al.en 2017 [14]. Esta arquitectura supuso un nuevo nivel de mejora en el campo del Procesamiento del Lenguaje Natural. Este tipo de sistema permite alinear ciertas palabras de una secuencia con otras, calculando una representación de dicha secuencia mucho más precisa y eficiente que en anteriores modelos.

5. Aplicaciones de las LSTM

A continuación presentaré algunas aplicaciones de las redes recurrentes, en las que ser ha evidenciado de manea contundente su performance. La aplicación de estas redes es cada vez más notorio, presentándose nuevos papers científicos con novedosa aplicaciones de estas redes, en particular las Transformes. Veamos algunas de las aplicaciones más emblemáticas:

5.1 Generación automática de subtítulos de imágenes

El subtitulado automático de imágenes es la tarea en la que, dada una imagen, el sistema debe generar un pie de foto que describa el contenido de la imagen. En 2014 se produjo una explosión de algoritmos de deep learning logrando resultados muy impresionantes en este problema, aprovechando el trabajo de los mejores modelos de clasificación y detección de objetos en fotografías. [15]

Una vez que se pueden detectar objetos en fotografías y generar etiquetas para esos objetos, se puede ver que el siguiente paso es convertir esas etiquetas en una descripción de frases inteligibles. Los sistemas involucrados se entrenen, mediante redes neuronales convolucionales profundas para la

detección de objetos en las fotografías y luego se utiliza una red recurrente LSTM para convertir las etiquetas en una frase coherente.

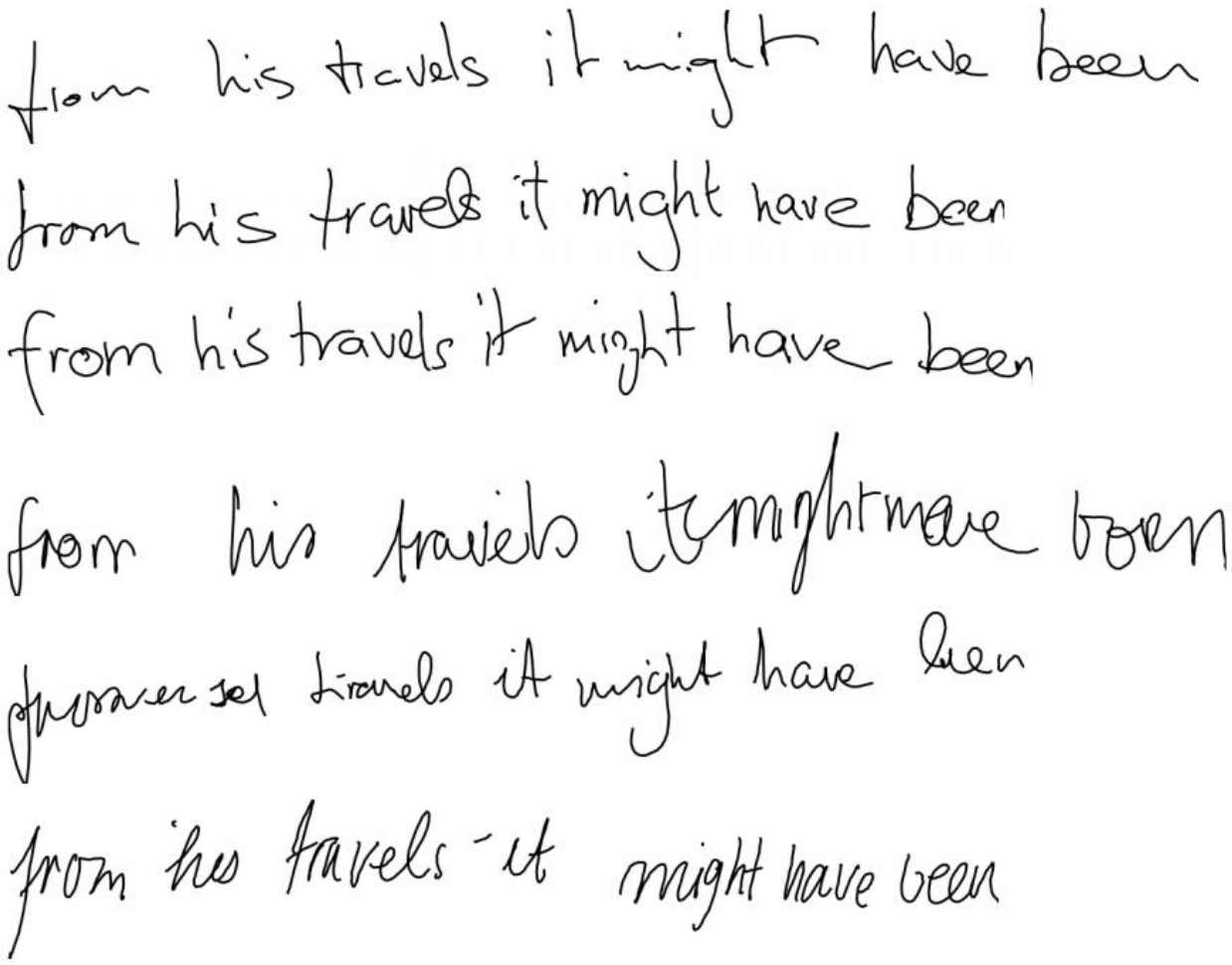
5.2 Traducción automática de textos

El modelo debe aprender la traducción de las palabras, el contexto en el que se modifica la traducción y soportar secuencias de entrada y salida que pueden variar en longitud tanto en general como entre sí. [16]

Type	Sentence
Our model	Ulrich UNK , membre du conseil d' administration du constructeur automobile Audi , affirme qu' il s' agit d' une pratique courante depuis des années pour que les téléphones portables puissent être collectés avant les réunions du conseil d' administration afin qu' ils ne soient pas utilisés comme appareils d' écoute à distance .
Truth	Ulrich Hackenberg , membre du conseil d' administration du constructeur automobile Audi , déclare que la collecte des téléphones portables avant les réunions du conseil , afin qu' ils ne puissent pas être utilisés comme appareils d' écoute à distance , est une pratique courante depuis des années .
Our model	" Les téléphones cellulaires , qui sont vraiment une question , non seulement parce qu' ils pourraient potentiellement causer des interférences avec les appareils de navigation , mais nous savons , selon la FCC , qu' ils pourraient interférer avec les tours de téléphone cellulaire lorsqu' ils sont dans l' air " , dit UNK .
Truth	" Les téléphones portables sont véritablement un problème , non seulement parce qu' ils pourraient éventuellement créer des interférences avec les instruments de navigation , mais parce que nous savons , d' après la FCC , qu' ils pourraient perturber les antennes-relais de téléphonie mobile s' ils sont utilisés à bord " , a déclaré Rosenker .
Our model	Avec la crémation , il y a un " sentiment de violence contre le corps d' un être cher " , qui sera " réduit à une pile de cendres " en très peu de temps au lieu d' un processus de décomposition " qui accompagnera les étapes du deuil " .
Truth	Il y a , avec la crémation , " une violence faite au corps aimé " , qui va être " réduit à un tas de cendres " en très peu de temps , et non après un processus de décomposition , qui " accompagnerait les phases du deuil " .

5.3 Generación automática de texto manuscrito

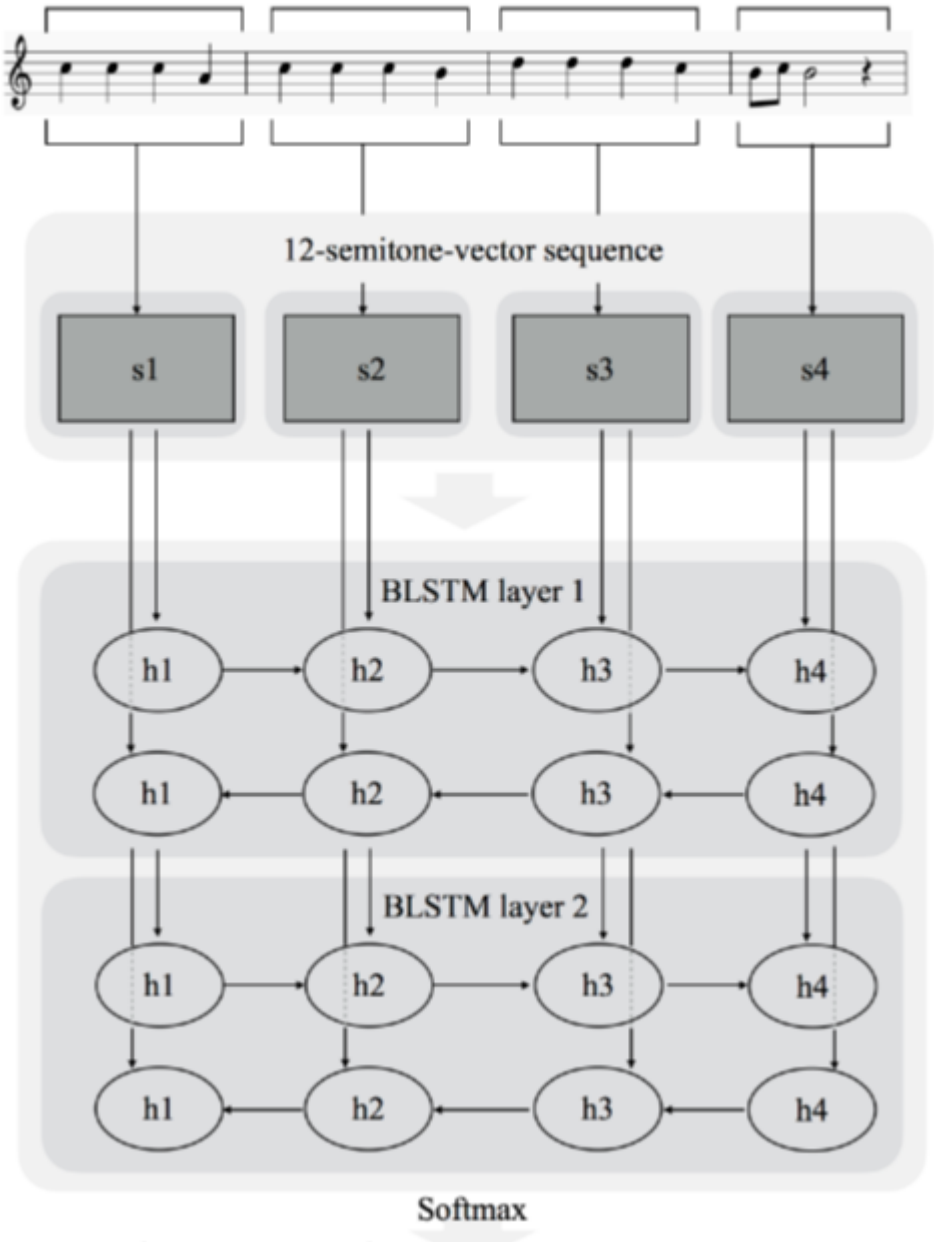
Se trata de una tarea en la que, dado un corpus de ejemplos de escritura, se genera una nueva palabra o frase determinada. La escritura se proporciona como una secuencia de coordenadas utilizadas por un bolígrafo cuando se crearon las muestras de escritura. A partir de este corpus, se aprende la relación entre el movimiento del bolígrafo y las letras, por lo que se pueden generar pueden generar nuevos ejemplos. Mediante la aplicación de estas técnicas se pueden aprender diferentes estilos para luego imitarlos. [17]



5.4 Generación automática de partituras musicales

En la literatura podemos encontrar ejemplos de este tipo de arquitectura aplicado al caso de la composición musical. Por ejemplo, en 2017, Limet al. propusieron un sistema basado en LSTM bidireccionales (BLSTM) para generar acompañamiento musical en forma de una progresión de acordes dada una melodía [18. La arquitectura consistía en una red bidireccional con dos capas LSTM de 128 celdas cada una. Esta red fue entrenada con un corpus de 2252 partituras de música moderna (rock, jazz, pop, etc.), resultando en un total de 1802 canciones. Para representar las notas de la melodía, se tuvieron encuentra 12 posibles valores (no se contó con ninguna octava extra) y para los acordes, únicamente se usaron triadas. El entrenamiento se llevó a cabo presentando a la red varios grupos de cuatro tiempos (negras) de melodía y comparando a la salida los acordes correspondientes. En la fase de generación, iterativamente se pasó a la red cadenas de cuatro tiempos, concatenando los acordes de salida hasta tener la progresión completa. Los autores compararon el sistema con otros modelos y lo evaluaron cuantitativamente (matrices de confusión)

y cualitativamente (con 25 oyentes). Los resultados mostraron mayor precisión que con los otros modelos, así como una tendencia por parte de los usuarios a elegir estas composiciones frente a las de los otros modelos.



6. Conclusiones

Los modelos de aprendizaje automático clásico, tanto los supervisados como los no supervisados, exigen algún tipo de control y conocimiento de las características o atributos medidos o relevados partir del cual intentaremos desarrollar un modelo predictivo o descriptivo. Conociendo estas representaciones de la realidad subyacente al entorno de análisis, expresadas en variables que asumimos independientes entre sí, podremos aplicar las más variadas técnicas de ajuste para poder estimar los parámetros de una función a partir de la cual arribar a los resultados buscados (sean estos de predicción, agrupamiento, etc.). No obstante ello hay muchos casos en los que esta representaciones y atributos se presentan de un formato no estructurada, como e el caso de fragmentos de texto, imágenes, audio o video. En estos casos las representaciones deben ser generadas algorítmicas, proceso en el cual generamos estas abstracciones de bajo nivel pero con la particularidad, una consecuencia no muy deseable, de que estas estén sumamente correlacionadas entre si (los píxeles en una imagen las palabras e un texto y las notas musicales y/o los sonidos en un archivos de audio).

Es por lo anterior que precisamos de un modelo cuya configuración e encuentre preparada para lidiar con estas representaciones de muy bajo nivel altamente correlacionadas entre si, solución que encontramos en los modelos redes neuronales artificiales, en especial las que utilizan las técnicas del aprendizaje profundo (Deep learning). Construidas para emular la estructura de las conexiones sinápticas del cerebro humano, las arquitecturas de aprendizaje profundo se utilizan de forma ubicua para la extracción de características, el análisis de patrones y la abstracción de datos mediante representación sucesiva y concatenada de complejidad creciente. . Se ha demostrado que estos modelos rinden mejor y más rápido que las actuales técnicas de análisis de última generación en tareas de aprendizaje supervisado y no supervisad. Cada una de las configuraciones y arquitecturas de las redes neuronales de aprendizaje profundo posee característica en que se destacan y que le confieren su particular y destacada utilidad.

Los modelos de aprendizaje profundo analizados en este artículo son las redes neuronales recurrentes (RNN por sus siglas en inglés), especializadas en el tratamiento e información presentada en forma secuencial (variables de corte longitudinal). Bajo esta definición encontramos a los lenguajes, tanto los naturales y los simbólicos, señales de audio (vos hablada, música), video y series temporales. Cada arquitectura de redes neuronales recurrentes se presenta de manera progresiva en función tanto de su aparición en la escena científica como en las mejoras que fueron aportando a las configuraciones que las precedieron. Es así que arribamos a los modelos más utilizados en la actualidad, las redes de memoria de corto y largo plazo LSTM, y los Transformers, configuraciones muy sofisticadas de redes recurrentes que han sorteado las dificultades de perdida

memoria y optimización de la función objetivo inherentes a las RNN clásicas, asegurando así una performance que las posiciona en el estándar actual de las arquitectitas de Deep learning especializados en el modelados de información de corte longitudinal. El artículo finaliza con la parentación de algunas aplicaciones emblemáticas de las redes recurrentes del tipo LSTM.

Bibliografía

[1] Mitchell, T. M. (1997). Machine Learning. McGraw-Hill, New York.

[2] B. Yegnanarayana, Artificial neural networks. PHI Learning Pvt. Ltd., 2009.

[3] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", The bulletin of mathematical biophysics, vol. 5, no. 4, pp. 115–133, 1943.

[4] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.", Psychological review, vol. 65, no. 6, p. 386, 1958.

[5] Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press. http://www. deeplearningbook.org (2016)

[6] Yunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, FethiBougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representationsusing rnn encoder-decoder for statistical machine translation.arXiv preprint ar-Xiv:1406.1078, 2014

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan NGomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. InAdvancesin neural information processing systems, pages 5998–6008, 2017.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory.Neuralcomputation, 9(8):1735–1780, 1997] [Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen netzen.Diploma,Technische Universität München, 91(1), 1991

[9] Ronald J Williams and David Zipser. Gradient-based learning algorithms for recu-rrent.Backpropagation: Theory, architectures, and applications, 433, 1995.

[10] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks.IEEEtransactions on Signal Processing, 45(11):2673–2681, 1997

[11] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate.arXiv preprint arXiv:1409.0473, 2014

[12]Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Sa-lakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural imagecaption generation with visual attention. InInternational conference on machinelearning, pages 2048–2057, 2015.

[13] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy.Hierarchical attention networks for document classification. InProceedings of the2016 conference of the North American chapter of the association for computationallinguistics: human language technologies, pages 1480–1489, 2016.

[14] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy.Hierarchical attention networks for document classification. InProceedings of the2016 conference of the North American chapter of the association for computationallinguistics: human language technologies, pages 1480–1489, 2016.

[15] Show and Tell: A Neural Image Caption Generator, 2014. https://arxiv.org/abs/1411.4555

[16] https://arxiv.org/abs/1409.3215

[17] Graves, A. (2013), 'Generating Sequences With Recurrent Neural Networks.', CoRR abs/1308.0850

[18] Hyungui Lim, Seungyeon Rhyu, and Kyogu Lee. Chord generation from symbolicmelody using blstm networks.arXiv preprint arXiv:1712.01011, 2017

[19] Charniak, E. (2018). Introduction to deep learning.