Tecnológico de Monterrey

Aprendizaje Profundo para Datos Secuenciales

Daniel Otero Fadul

Departamento de Ciencias
Escuela de Ingeniería y Ciencias

Datos secuenciales

Los **datos secuenciales** se refieren a información organizada en una secuencia o serie donde el orden de los elementos es importante y puede tener significado en el análisis de los datos. Los datos secuenciales son comunes en diversas áreas como la biología (por ejemplo, secuencias de ADN), la industria financiera (series temporales de precios de acciones), el procesamiento de lenguaje natural (secuencias de palabras en textos) y otras disciplinas donde se estudian patrones y relaciones en secuencias de eventos o elementos.

Datos secuenciales

Estas son algunas aplicaciones en las que se trabaja con datos secuenciales:

- Clasificación de documentos y clasificacón de series temporales, por ejemplo, para identificar el tema de un artículo o el autor de un libro.
- Comparaciones de secuencias de texto para estimar qué tan relacionados están dos documentos.
- Aprendizaje de secuencia a secuencia: traducir una frase en inglés al francés.
- Análisis de sentimiento: clasificar el sentimiento de tweets o reseñas de películas como positivo o negativo.
- Pronóstico de series temporales: predecir el clima futuro en una ubicación determinada dados los datos meteorológicos recientes.

Datos secuenciales

Hay varios modelos de aprendizaje profundo que han sido diseñados para trabajar con datos secuenciales, entre estos, las **redes neuronales recurrentes** (RNN) y las **redes convolucionales de 1D** (1D CNN). Comenzaremos este módulo hablando de las RNNs.

RNN

Una **Red Neuronal Recurrente** (RNN) es un tipo de red neuronal artificial diseñada para procesar datos secuenciales o de series temporales. A diferencia de las redes neuronales tradicionales, que solo consideran la entrada actual para predecir la salida, las RNN tienen conexiones que les permiten mantener y usar información anterior, lo que las hace ideales para trabajar con datos secuenciales donde la historia es importante para la predicción.

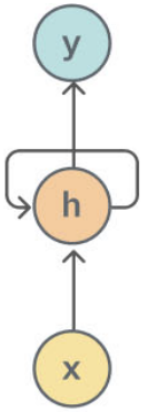

Figura: Una RNN de una sola capa. Imagen tomada de [3].

RNN

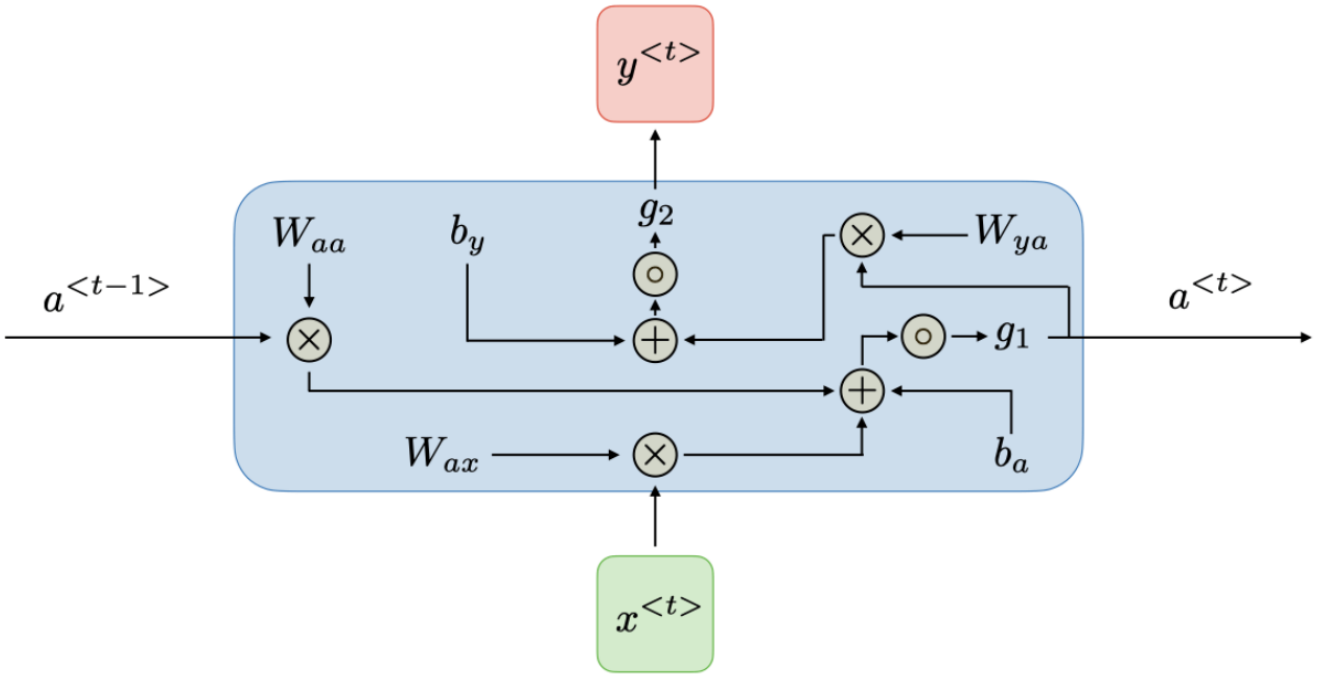

Figura: Estructura interna de una RNN. La entrada, salida y activación de la RNN en el instante de tiempo t son $x^{\langle t \rangle}$, $y^{\langle t \rangle}$ y $a^{\langle t \rangle}$, respectivamente. Los parámetros que se modifican en el entrenamiento se almacenan en W_{aa}, W_{ax}, W_{ya}, b_a y b_y. Las funciones de activación son g_1 y g_2. Imagen tomada de [3].

Dado lo anterior, tenemos que

$$\begin{aligned} a^{\langle t \rangle} &= g_1\left(W_{aa}a^{\langle t-1 \rangle} + W_{ax}x^{\langle t \rangle} + b_a\right), \\ y^{\langle t \rangle} &= g_2\left(W_{ya}a^{\langle t \rangle} + b_y\right). \end{aligned}$$

Tipos de RNN

Sean T_x y T_y el número de entradas y salidas de la RNN, respectivamente. Dado esto, diferentes valores de T_x y T_y nos generan las distintas arquitecturas que una RNN puede tener. La más simple se da cuando $T_x = T_y = 1$, la cual es simplemente una red neuronal convencional.

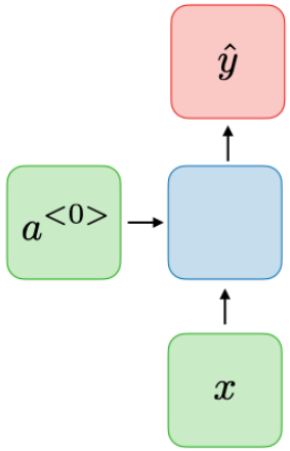

Figura: Una RNN uno a uno, o "one-to-one" en inglés, es una red neuronal tradicional. Esta arquitectura es útil para resolver problemas de clasificación. Imagen tomada de [3].

Tipos de RNN

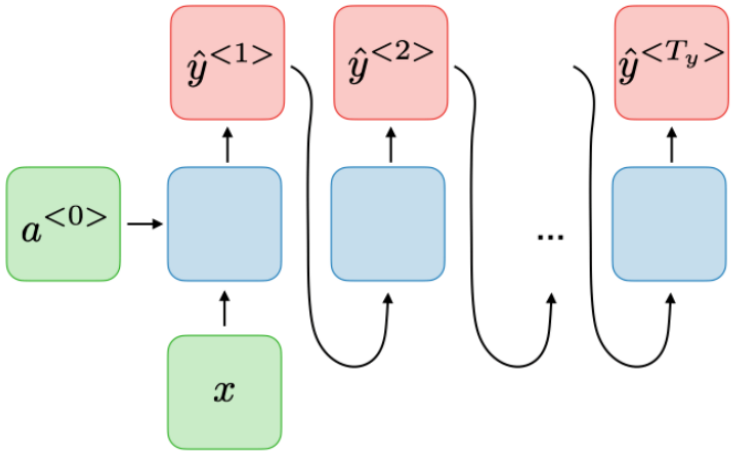

Figura: Si $T_x = 1$ y $T_y > 1$, tenemos un tipo de RNN conocida como "one-to-many" en inglés. Esta arquitectura es idónea para generar una secuencia larga de datos a partir de una sola entrada, por ejemplo, en generación de música. Imagen tomada de [3].

Tipos de RNN

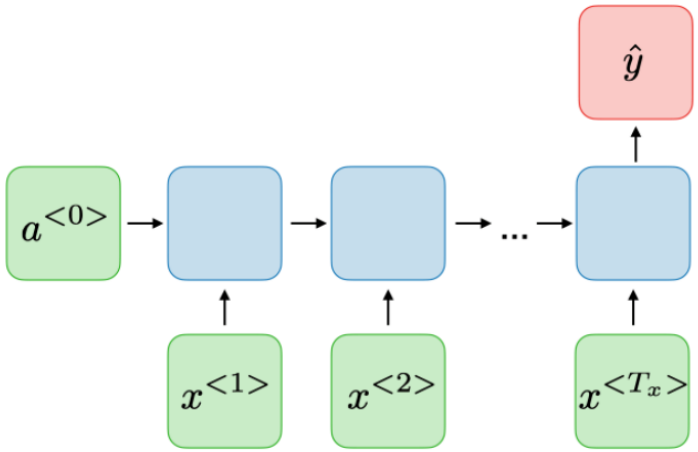

Figura: Si $T_x > 1$ y $T_y = 1$, tenemos un tipo de RNN conocida como "many-to-one" en inglés. Esta arquitectura es apropiada en aplicaciones como análisis de sentimiento. Imagen tomada de [3].

Tipos de RNN

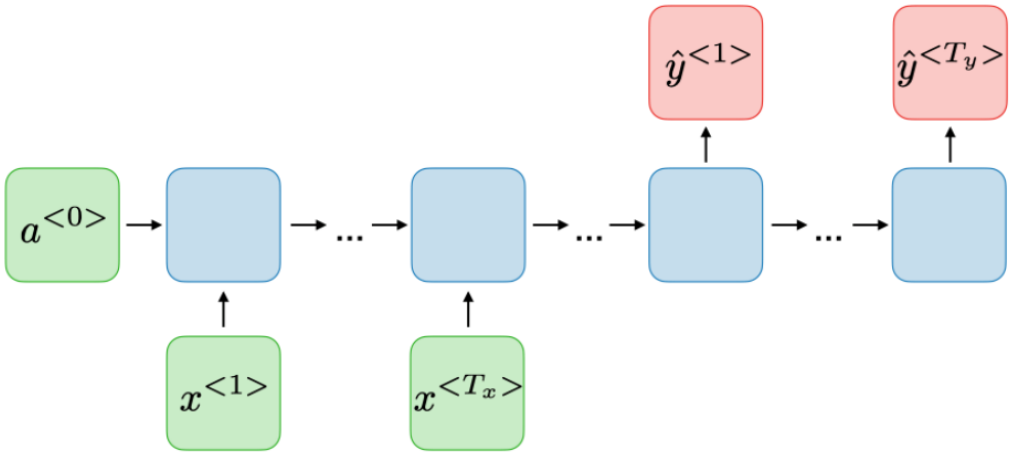

Figura: Otro tipo de RNN se le conoce como "many-to-many" en inglés. En este caso, podemos tener que $T_x = T_y$ o que $T_x \neq T_y$. En la imagen podemos ver el caso en el que $T_x \neq T_y$, el cual es útil para traducción de texto. Imagen tomada de [3].

Ventajas y desventajas de las RNN

Algunas de las ventajas de las RNN son las siguientes:

- La arquitectura de las RNN está diseñada de tal forma que puede procesar entradas de cualquier longitud. Aunque el tamaño de la entrada aumente, el tamaño del modelo no aumenta.
- Un modelo RNN se modela para recordar cada información a lo largo del tiempo, lo que es muy útil en cualquier predictor de series temporales.
- La memoria interna de las redes neuronales recurrentes es una propiedad inherente que se utiliza para procesar series arbitrarias de entradas, lo que no ocurre con las redes neuronales 'feedforward".

Ventajas y desventajas de las RNN

En cuanto a los inconvenientes, tenemos estos problemas con las RNN:

- El entrenamiento de los modelos RNN puede ser muy difícil y llevar mucho tiempo en comparación con otras redes neuronales artificiales.
- Son propensas a problemas como la explosión y la desvanecimiento del gradiente.
- Las RNN no pueden apilarse en modelos muy profundos.
- Las RNN no son capaces de realizar un seguimiento de las dependencias a largo plazo.

Ventajas y desventajas de las RNN

En conclusión, las RNN se utilizan en una amplia variedad de aplicaciones, como reconocimiento de voz, procesamiento de lenguaje natural, traducción automática, generación de texto, análisis de series temporales, entre otros. Sin embargo, las RNN también tienen limitaciones, como el problema de desvanecimiento/explosión del gradiente. Para abordar estas limitaciones han surgido variaciones de las RNN, como las LSTM (Long Short-Term Memory) y las GRU (Gated Recurrent Unit), que han demostrado ser más efectivas en el modelado de dependencias a largo plazo y en el entrenamiento de redes profundas.

LSTM

Una LSTM, “Long Short-Term Memory” en inglés, es un tipo de red neuronal recurrente (RNN) que se utiliza en el aprendizaje profundo y el procesamiento del lenguaje natural (NLP). Fue propuesta por Hochreiter y Schmidhuber en 1997 como una solución al problema de desvanecimiento del gradiente que afectaba a las RNN tradicionales.

La LSTM está diseñada para aprender dependencias a largo plazo en secuencias de datos, lo que la hace especialmenteútil en tareas donde es importante recordar información relevante de entradas anteriores a medida que se procesa una secuencia.

LSTM

Las principales características de una LSTM incluyen:

- **Celdas de memoria:** Mantienen y actualizan la información a lo largo del tiempo.
- **Puertas:** Son mecanismos que controlan el flujo de información en la red:
  - *Puerta de entrada (Input Gate):* Determina qué nueva información se debe agregar a la celda de memoria.
  - *Puerta de olvido (Forget Gate):* Decide qué información de la celda de memoria anterior debe olvidarse.
  - *Puerta de estado candidata (State Candidate Gate):* Ayuda a determinar qué tan relevante es la información de un estado previo.
  - *Puerta de salida (Output Gate):* Calcula la salida de la celda de LSTM basada en la información almacenada en la memoria.

Estas puertas permiten que la LSTM controle cuándo olvidar, agregar o leer información de la celda de memoria, lo que ayuda a mitigar el problema de desvanecimiento del gradiente y a capturar dependencias a largo plazo en los datos de secuencia.

LSTM

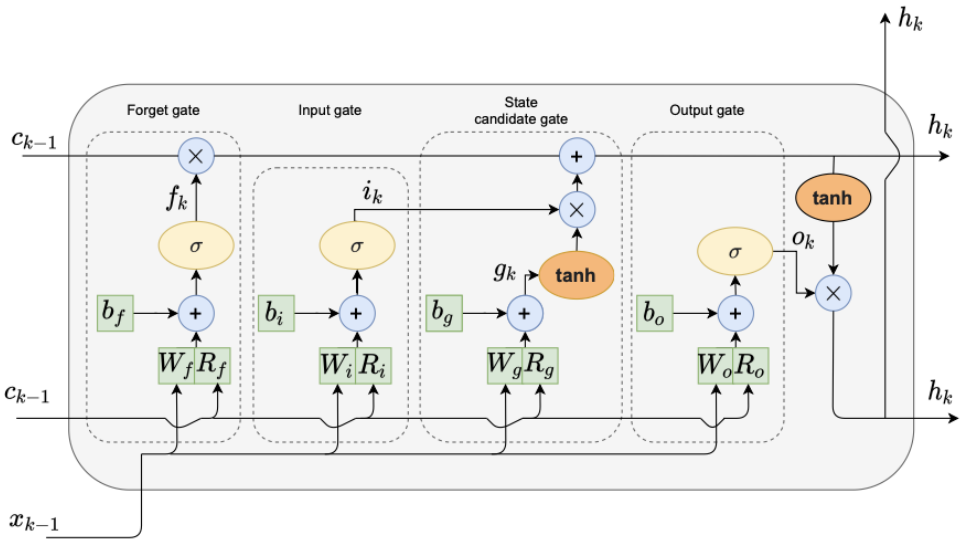

Figura: Arquitectura de una celda LSTM. Los parámetros que se modifican durante el entrenamiento son las matrices W_i, W_f, W_g, W_o, R_i, R_f, R_g y R_o, y los vectores b_i, b_f, b_g y b_o. Imagen tomada de [4].

LSTM

Dado lo anterior, tenemos que

$$\begin{aligned} i_k &= \sigma\left(W_i x_k + R_i h_{k-1} + b_i\right), \\ f_k &= \sigma\left(W_f x_k + R_f h_{k-1} + b_f\right), \\ g_k &= \tanh\left(W_g x_k + R_g h_{k-1} + b_g\right), \\ o_k &= \sigma\left(W_o x_k + R_o h_{k-1} + b_o\right). \end{aligned}$$

El nuevo estado de la celda está dado por

$$c_k = f_k \circ c_{k-1} + i_k \circ g_k,$$

donde $\circ$ denota el producto Hadamard. Finalmente, el estado oculto es igual a

$$h_k = o_k \circ \tanh(c_k).$$

GRU

Una GRU, o Red Neuronal Recurrente con Puertas (Gated Recurrent Unit en inglés), es un tipo de arquitectura de red neuronal recurrente (RNN) que se utiliza en el campo del aprendizaje profundo y el procesamiento del lenguaje natural (NLP). Fue propuesta por Kyunghyun Cho en 2014 como una variante de las redes LSTM.

La GRU está diseñada para abordar algunos de los problemas de las RNN tradicionales, como el problema de desvanecimiento del gradiente, que dificulta el aprendizaje de dependencias a largo plazo en secuencias de datos. La GRU utiliza mecanismos de puertas para controlar el flujo de información en la red, lo que ayuda a mantener y actualizar la información relevante a lo largo del tiempo en las secuencias.

GRU

Las principales características de una GRU incluyen:

- *Puerta de reinicio (Reset Gate):* Decide cuánta información de la memoria anterior debe olvidarse.
- *Puerta de actualización (Update Gate):* Controla cuánta información de la memoria anterior debe mantenerse para la siguiente iteración.
- *Puerta de estado candidata (State Candidate Gate):* Ayuda a determinar qué tan relevante es la información de un estado previo.

GRU

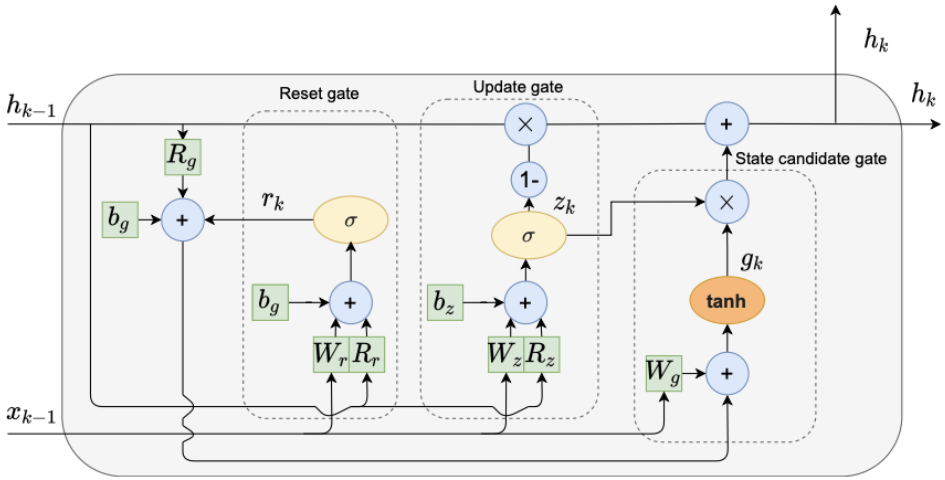

Figura: Arquitectura de una celda GRU. Los parámetros que se modifican durante el entrenamiento son las matrices W_r, W_z, W_g, R_r, R_z y R_g, y los vectores b_r, b_z y b_g.Imagen tomada de [4].

GRU

Dicho lo anterior, tenemos que

$$\begin{aligned} r_k &= \sigma\left(W_r x_k + R_r h_{k-1} + b_r\right), \\ f_k &= \sigma\left(W_f x_k + R_z h_{k-1} + b_z\right), \\ g_k &= \tanh\left(W_g x_k + r_k \circ R_g h_{k-1} + b_g\right), \\ h_k &= (\mathbf{1} - z_k) \circ g_k + z_k \circ h_{k-1}, \end{aligned}$$

donde $\mathbf{1}$ es un vector que contiene solo unos.

Redes Neuronales Convolucionales (CNN)

Una de las redes neuronales más usadas son las **redes neuronales convolucionales**, conocidas como CNN por sus siglas en inglés. Este tipo de arquitectura es bastante útil para datos de alta dimensionalidad y que están relacionados espacialmente (o temporalmente). Si utilizaramos una red neuronal totalmente conectada la cantidad de parámetros de la red sería innecesariamente grande.

Redes Neuronales Convolucionales (CNN)

Por ejemplo, en los problemas de clasificación de imágenes, los píxeles vecinos suelen compartir información por su proximidad espacial, lo cual debería reflejarse en la arquitectura de la red. Basándose en esta observación, parece razonable tener redes neuronales que tengan campos receptivos locales en el sentido de que recojan información conjuntamente de entradas espacialmente cercanas.

Redes Neuronales Convolucionales (CNN)

Además, en el procesamiento de imágenes, es relevante tener un clasificador que sea invariante bajo distintas operaciones como la traslación o la rotación de imágenes. Dado esto, es razonable codificar tales invariancias en la arquitectura.

Las redes convolucionales combinan los tres aspectos ya mencionados: manejo de entradas de alta dimensionalidad, procesamiento de datos relacionados espacialmente de manera local, y ser "invariantes" a traslaciones.

Redes Neuronales Convolucionales (CNN)

Vale la pena decir que las CNN realmente no son invariantes a traslaciones, sino **equivariantes**. Por otro lado, las convoluciones no son realmente la operación de **convolución** que se define para funciones continuas y discretas, estas son realmente **correlaciones cruzadas**.

Redes Neuronales Convolucionales (CNN)

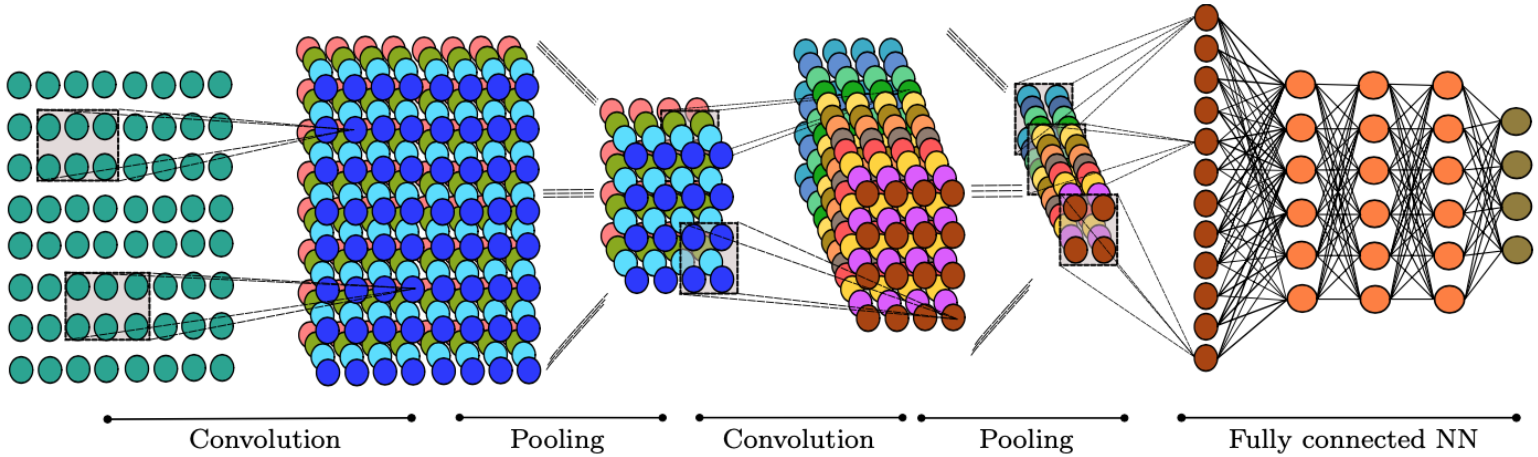

Figura: Arquitectura de una red neuronal convolucional con bloques convolucionales bidimensionales y submuestreo 2×2 como operación de agrupación. Imagen tomada de [5].

Redes Neuronales Convolucionales (CNN)

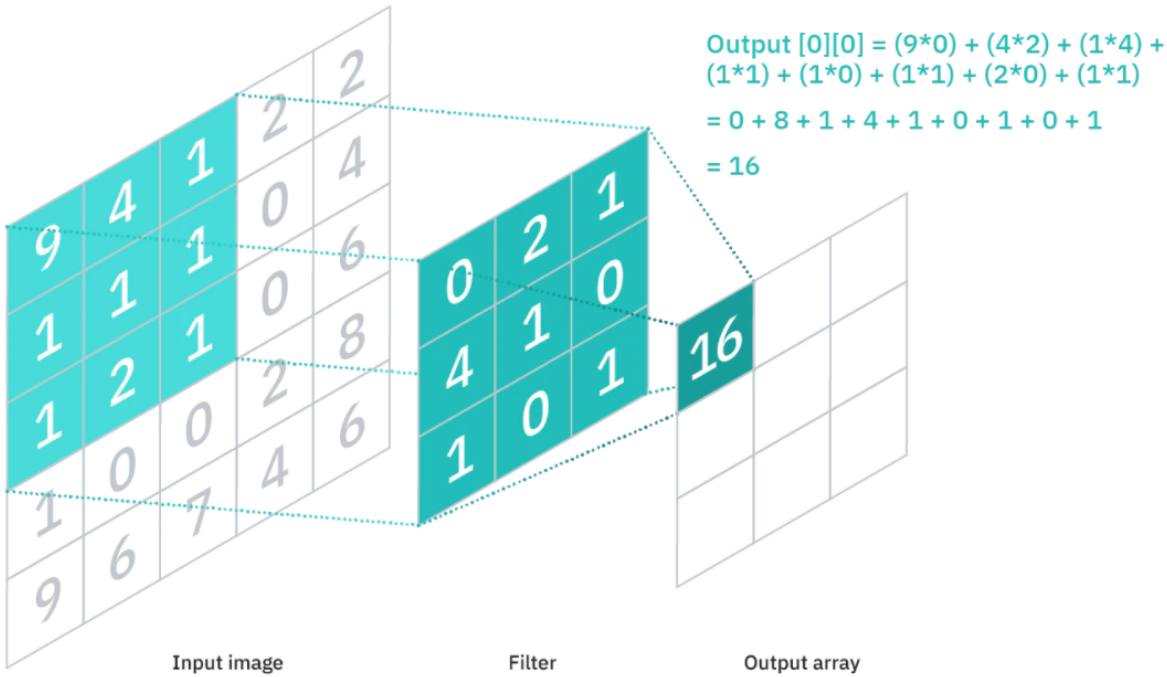

Figura: En la figura se puede apreciar como se realiza la "convolución" con un filtro de tamaño 3×3. El filtro recorre toda la imagen y cada valor calculado se almacena en un arreglo. Estos filtros, los cuales se encuentran en cada capa convolucional, le ayudan a la red a aprender características relevantes de las imágenes. Imagen tomada de `https://www.ibm.com/cloud/learn/convolutional-neural-networks`.

Redes Neuronales Convolucionales (CNN)

Las capas de **"pooling"** realizan una reducción de la dimensionalidad por medio de un submuestreo. Hay dos tipos principales de "pooling:

- **"Max pooling":** A medida que el filtro se desplaza por la entrada, selecciona el píxel con el valor máximo para enviarlo a la matriz de salida.
- **"Pooling" promedio:** En este caso se calcula el valor medio dentro del campo receptivo para enviarlo a la matriz de salida.

Aunque se pierde mucha información en la capa de pooling, esta técnica ayuda a reducir la complejidad de la red, lo cual limita el riesgo de sobreajuste.

Redes Neuronales Convolucionales (CNN)

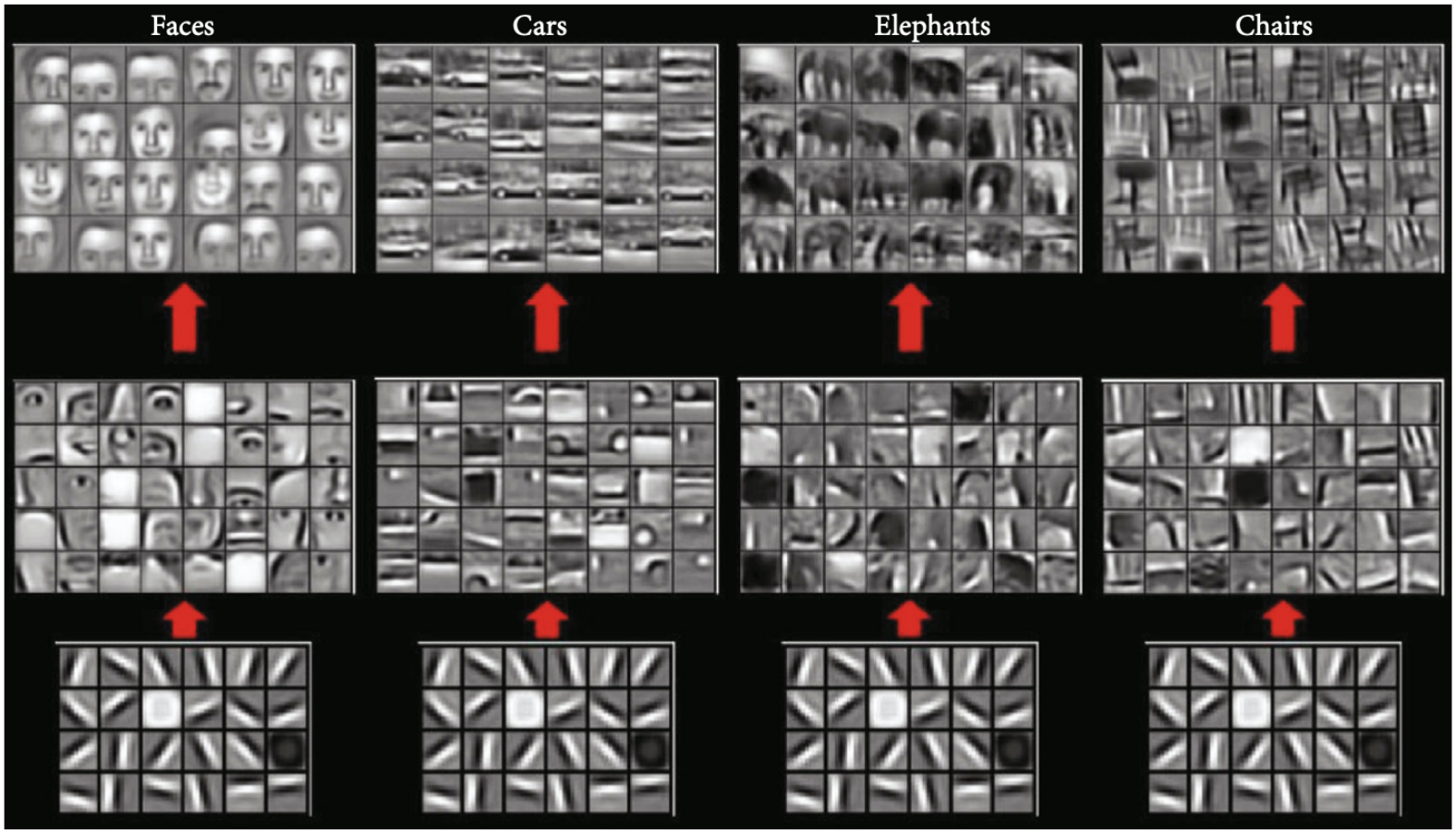

Figura: Para las diferentes capas convolucionales de una CNN, las características de la imagen aprendidas por cada capa son diferentes. Los patrones aprendidos por las capas más superficiales tienden a ser más generales, mientras que para capas más profundas, las características aprendidas son más específicas dependiendo del conjunto de datos con el que se esté trabajando. Imagen tomada de [6].

BIBLIOGRAFÍA

1 Chollet, Francois. *"Deep learning with Python,"* Simon and Schuster, 2021.

2 `https://code.google.com/archive/p/word2vec`

3 `https://nlp.stanford.edu/projects/glove`

4 Zarzycki, K., Ławryńczuk, M., *"LSTM and GRU neural networks as models of dynamical processes used in predictive control: A comparison of models developed for two chemical reactors"*, Sensors, 21(16), 5625, (2021).

5 J. Berner et al., *"The modern mathematics of deep learning,"* arXiv preprint arXiv:2105.04026, 2021.

6 Y. Wang et al., *"Multiclassification of endoscopic colonoscopy images based on deep transfer learning"*, Computational and Mathematical Methods in Medicine, 2021.