



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

Pronóstico de la productividad forrajera y balance hídrico en la región de los pastizales del Río de la Plata

BARTOL, TOMÁS AUGUSTO

PEIRETTI, PABLO

SÁNCHEZ, MILAGROS VERÓNICA

Ing. Jorge Néstor Zanabria

Trabajo Práctico Profesional

Índice

1	Introducción	2
2	Descripción del proyecto	3
3	Contexto de Desarrollo	5
4	Marco Teórico	7
4.1	Productividad Primaria Neta Aérea	7
4.2	Balance Hídrico Climático	9
4.3	Redes Neuronales	13
4.3.1	Funcionamiento General de las neuronas	14
4.3.2	Organización neuronal	18
4.3.3	Métodos de aprendizaje	19
4.3.4	Validación de los resultados	20
4.3.5	Backpropagation	22
4.3.6	Redes Neuronales Recurrentes	24
4.3.7	Redes Neuronales Convolucionales	27
4.3.8	Data Clustering	29
5	Desarrollo tecnológico	37
6	Conclusión	38
7	Referencias	39

1. Introducción

2. Descripción del proyecto

En la región que comprende Argentina, Brasil y Uruguay, el sector de la carne representa un componente importante del producto bruto interno del sector agropecuario (Bruinsma 2017, Rossini et al. 2017). Los productores de ganado en esta zona han acumulado valiosa experiencia en la gestión del ganado y la mejora genética (Cantet et al. 2017, Goszczynski et al. 2018). Sin embargo, a pesar de estos avances, persiste una brecha significativa entre los rendimientos máximos potenciales y los rendimientos promedio en los sistemas de producción, la cual no ha mostrado signos de disminución en las últimas tres décadas (Viglizzo et al. 2001, 2011). Esta brecha se atribuye en gran parte a la escasa información sobre la variación espacial y temporal de la principal fuente de energía para el ganado, el forraje.

La ganadería en estos tres países es principalmente de base pastoril, el ganado consume in situ el forraje producido por pastizales naturales y pasturas implantadas (Herrero y Thornton 2013, Irisarri et al. 2014, Thornton et al. 2009, Viglizzo et al. 2011). Como la oferta forrajera es mucho más variable que la demanda animal, es necesario manejar adecuadamente la variación en la cantidad y calidad del forraje (Bradford et al. 2006, Craine et al. 2017, 2010, Paruelo et al. 1997). En este sentido, contar con pronósticos de la producción forrajera y de las condiciones ambientales que la determinan será muy útil e impactará positivamente en los niveles de producción.

En las últimas tres décadas se ha avanzado significativamente en entender los controles de la variación espacial y temporal de la productividad primaria neta aérea (PPNA). La productividad primaria neta aérea (PPNA) es el principal determinante de la producción forrajera, y su variabilidad está estrechamente relacionada con la disponibilidad de agua en el suelo. En este contexto, es fundamental tener en cuenta el Balance Hídrico Climático (BHC), que evalúa la relación entre la entrada de agua en forma de precipitación y su salida debido a la evaporación y el uso por parte de los cultivos o la vegetación. El BHC es una herramienta valiosa para comprender cómo la disponibilidad de agua influye en la producción de forraje y, en última instancia, en la producción ganadera.

Recientemente, se ha desarrollado un sistema de seguimiento de la PPNA en tiempo casi real a partir de datos provistos por sensores remotos y estaciones meteorológicas. Pero no se dispone aún de un sistema de pronóstico de la PPNA ni de su control más importante.

El objetivo general de este proyecto es desarrollar un sistema de pronósticos de tanto el BHC como la PPNA. Los sistemas de pronóstico operarán para las áreas ganaderas y los recursos forrajeros más importantes de la región de los pastizales del Río de la Plata.

Los pronósticos se basarán en la dinámica de las condiciones endógenas (PPNA y BHC de períodos previos) y exógenas (precipitación, temperatura y evapotranspiración, de períodos previos, y de proyecciones de modelos climáticos). Las condiciones endógenas serán estimadas ad hoc, y las exógenas serán tomadas de bases de datos de dominio público. Los pronósticos serán generados mediante redes neuronales recurrentes, una familia de técnicas de aprendizaje automático especialmente apropiadas para el análisis y pronóstico de series temporales no lineales. Los pronósticos tendrán una anticipación de al menos 1 mes, cobertura temporal de entre 1 y 6 meses para toda la región, y resolución espacial de aproximadamente 6 ha.

Todos los actores de la cadena ganadera, desde los productores hasta los decisores políticos, se beneficiarán al contar anticipadamente con pronósticos del BHC y la PPNA. Esta información es clave para el uso y una planificación más educada de los recursos forrajeros en sistemas pastoriles. No existen en la región sistemas análogos al propuesto, ni tampoco abundan en el mundo.

3. Contexto de Desarrollo

Como se explicó previamente, el sistema de pronóstico propuesto funcionará a partir de una red neuronal a desarrollar. Para que esto sea posible, se requiere de una copiosa cantidad de datos con los que se pueda entrenar dicha inteligencia artificial. La captura de estos valores incurriría en un enorme costo, junto con una maniobra logística apreciable y, sumado a todo eso, una extensión temporal considerable. Es por ello que se decidió trabajar en conjunto con un grupo de investigadores del CONICET pertenecientes al Laboratorio de Fotointerpretación que a su vez forman parte del cuerpo docente de la Facultad de Agronomía de la Universidad de Buenos Aires (FAUBA).

Más en detalle, se encuentra en desarrollo un proyecto titulado *"Sistemas de pronóstico de la productividad forrajera y el balance hídrico para la región de los pastizales del Río de la Plata"*. El mismo tiene como objetivo crear un sistema de pronóstico del BHC y la PPNA. Se propuso entonces una relación simbiótica entre ambas partes: se nos proveerá información espectral recolectada por los integrantes del proyecto de investigación. Como contraprestación, los integrantes de este Trabajo Profesional se comprometen a prestar el *software* resultante para que ellos puedan arribar a las conclusiones pertinentes. Se aclara que quedará en manos de los autores del Trabajo Práctico Profesional la propiedad intelectual de los productos desarrollados.

Se aclara que en todo momento se mantendrá contacto entre las partes. La idea es lograr un trabajo interdisciplinario en el que se pueda favorecer el estado del arte de la tecnificación de la agroindustria en la región. Se cree que esto favorecerá el desarrollo del Trabajo Profesional ya que permite agregar un grupo de consultores técnicos de gran experiencia en el rubro agropecuario y, además, en el procesamiento de señales espectrales en ese ámbito.

El investigador titular es el Doctor Marcos Texeira, quien posee amplia experiencia en el estudio de procesos ecosistémicos, en particular la dinámica de la PPNA y sus controles mediante la aplicación de técnicas estadísticas modernas y datos provistos por sensores remotos. Además, se suma a él el doctor Carlos Marcelo Di Bella, eminencia en el área de procesamiento de imágenes en cultivos con más de dos décadas en el ámbito de la agricultura y ganadería, como co-tutor del respectivo Trabajo Profesional. Es importante destacar que se han mantenido fructíferas charlas con los integrantes del proyecto de investigación. Se acordó la colaboración y se mostraron dispuestos a instruir en los temas que se presenten adversos para lo aprendido en la carrera de Ingeniería Electrónica.

Ambos grupos de trabajo mantienen una comunicación constante, pudiendo así refinar los resultados del trabajo a medida que se desarrolle. Se evaluarán las ventajas de extender el alcance del proyecto propuesto por los investigadores del CONICET. Entre las propuestas barajadas se encuentran la inclusión de una mayor cantidad de datos del histórico de la zona así como también modelos de predicción de otros factores que se puedan considerar pertinentes (radiación, condiciones climáticas, etc.). La adición de los datos aquí propuestos pasará por dos etapas de evaluación. En primer lugar, se consultará con los especialistas en el tema si esta tarea conllevará una mejora del modelo. A su vez, se analizará la bibliografía pertinente así como también experiencias previas en el área que impliquen la posibilidad teórica de éxito de las medidas. Habiendo constatado la validez de la propuesta, se intentará hacer una prueba de concepto computacional de la idea. Por ejemplo, se propondrá entrenar al modelo con los

nuevos campos de análisis planteados y los resultados de estas acciones serán contrastados con sus instancias previas para corroborar su eficiencia. Los resultados de esos análisis se incluirán en las siguientes secciones de este informe.

Por otro lado, se menciona que los autores del Trabajo Práctico Profesional tendrán libertad de modificar las herramientas de trabajo delineadas en el anteproyecto presentado por los investigadores del CONICET. En el mismo se detallan metodologías de desarrollo de Redes Neuronales, plataformas web, entre otros. Si bien se evaluarán esas sugerencias para el desarrollo del trabajo, se menciona que se podrá optar por otras alternativas según se crea pertinente. Estas modificaciones pueden estar impulsadas por imprevistos en el proceso, alternativas que mejoren el desempeño del producto final o bien una forma de encausar el trabajo realizado de mejor manera dentro del entorno de un Trabajo Práctico Profesional de la carrera de Ingeniería Electrónica.

4. Marco Teórico

4.1. Productividad Primaria Neta Aérea

La Productividad Primaria Neta Aérea (PPNA) es un parámetro clave en de los ecosistemas en general. Sin embargo, toma un rol fundamental en los sistemas ganaderos pastoriles. La PPNA representa la tasa de creación de nueva biomasa vegetal aérea por unidad de área y tiempo. Como tal, se puede correlacionar directamente con la generación de energía disponible en una zona para los animales herbívoros que la habiten. La productividad de un suelo productor de material vegetal consumible (pastizales, pasturas, etc.) por el ganado es esencial para establecer la carga animal adecuada para el terreno (Oosterheld *et al.*, 1998) así como también planificar y manejar los excesos y déficit de forraje. Además, se puede considerar a la PPNA como una variable que funcione para integrar el funcionamiento ecosistémico y determina el nivel de muchos servicios de este tipo (Costanza *et al.*, 1998).

Sin embargo, hay poca información al respecto en las aplicaciones de agrotecnología actuales. Esto se debe a que su estimación y posterior extrapolación a situaciones similares resulta una tarea extremadamente difícil. Esto se debe, en parte, a que la Productividad Primaria Neta Aérea posee un carácter particularmente variable tanto temporal como espacialmente. En cuanto al espacio, se menciona que tiene una variación notable a lo largo de gradientes regionales, principalmente a causa de la heterogeneidad de precipitaciones en las zonas (Lauenroth, 1979; Sala *et al.*, 1988). A su vez, a menor escala, la PPNA varía en relación a gradientes topográficos, edáficos o de disturbio (Oosterheld *et al.*, 1999; Aragon y Oosterheld, 2008). En lo que al tiempo respecta, se menciona que la variación interanual de la PPNA de pastizales es considerable y acaba siendo una de las mayores incertidumbres del manejo de estos recursos para mejorar el marco de desarrollo de la explotación ganadera. Más aún, las variaciones correspondientes a las estaciones anuales tampoco pueden despreciarse y atentan contra el aprovechamiento de las virtudes de este parámetro.

La estimación de la PPNA debería incluir una gran variedad de factores. Entre ellos se mencionan los tiempos de crecimiento del forraje (hierbas, pastos verdes o secos y, por extensión, diversas plantas u órganos vegetales que se emplean para alimentar los animales domésticos, especialmente, el ganado) sin consumo del mismo por el ganado, el tiempo de rebrote sin consumo, etc. Sin embargo, esto implicaría afectar la zona de muestreo considerablemente y sería inviable para hacerlo en el largo plazo, ya que implicaría la defoliación al ras del suelo, exclusión del pastoreo, etc. Además, la misma captación de los factores de interés incurriría en un esfuerzo de muestreo alto y, comúnmente, incompleta.

Las problemáticas arriba mencionadas acabaron por dictar formas alternativas de estimar el parámetro a partir de otros factores que resulten menos invasivos para el área bajo estudio y, a su vez, de mayor facilidad de obtención. Así surge la aproximación de la PPNA basada en la integración temporal de la radiación absorbida. Ésta se basa en el modelo de Monteith para proceso ecosistémicos. El mismo propone que la Productividad Primaria Neta (PPN) (y por extensión, la PPNA) de una cubierta vegetal puede expresarse como

$$PPNA = RFA \cdot fRFAA \cdot EUR \quad (4.1)$$

En donde

$$[PPNA] = \frac{g}{m^2 \text{año}}$$

RFA = Radiación Fotosintéticamente Activa incidente, expresada en $\frac{MJ}{m^2 \text{año}}$

$fRFAA$ = fracción de la RFA Absorbida por la vegetación, la cual es adimensional

EUR = Eficiencia en el Uso de la Radiación, expresada en $\frac{g}{MJ}$

Analizando la ecuación planteada por Monteith, se ve que la vegetación fija nueva biomasa (según expresa la PPNA) en la medida que absorbe radiación solar fotosintéticamente activa y la convierte en biomasa. En este caso, al estar estimando la producción neta aérea, la EUR es la conversión de energía absorbida en masa aérea, que incluye aspectos de asignación de recursos de las partes tanto aéreas como subterráneas de la planta. Este modo de estimación facilita la obtención de los parámetros requeridos, ya que por un lado, la radiación incidente suele obtenerse con relativa sencillez de imágenes satelitales, además de que existen modelos de estimación de la misma.

Por otro lado, la fracción de RFA absorbida por la vegetación verde es posible estimarla con precisión a partir del uso de sensores remotos que midan la reflectancia en las bandas de frecuencia del rojo y el infrarrojo cercano (NIR). En cuanto al tercer parámetro, la eficiencia en el uso de la radiación, se menciona que el mismo varía dependiendo de la especie vegetal sobre la cual se esté tratando según distintas cuestiones relacionadas a la alimentación y respiración de la planta junto con diversas cuestiones ambientales. La estimación de este parámetro es sujeto de estudio y como tal se tienen fuentes confiables de aproximación del valor, ya que se ha probado que éste es el que menos influye en el dictamen final de la PPNA de pasturas y pastizales (Chapin *et al.*, 2002).

La estimación aquí propuesta se puede observar en la siguiente figura. A partir del Índice de Vegetación Normalizado (IVN) y de un modelo empírico se calcula la $fRFAA$. El producto entre la $fRFAA$ y la RFA estima la RFAA. Por otro lado, a partir de cosechas de biomasa aérea se estima la PPNA. Finalmente, la estimación de la EUR puede realizarse mediante un modelo de ajuste, por ejemplo lineal, entre la productividad y la radiación absorbida, o mediante el cociente. En círculos se muestran las variables medidas necesarias para la estimación de EUR, y en rectángulos las estimadas.

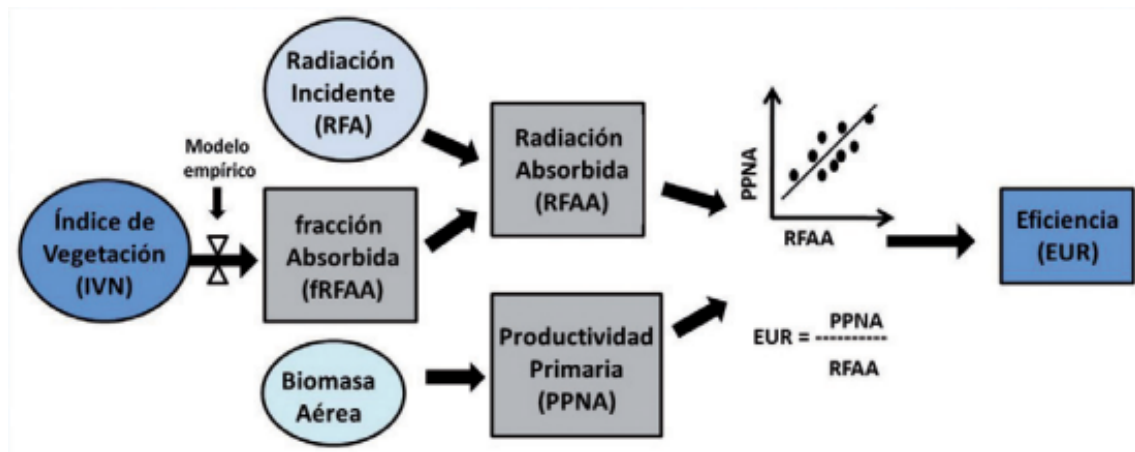


Figura 4.1.1: Esquema de estimación de la EUR

4.2. Balance Hídrico Climático

El Balance Hídrico Climático, también conocido como BHC por sus siglas, es la contabilización del agua del suelo. Ésta resulta del principio de conservación de la materia en un determinado volumen de suelo con vegetación. En esencia, el BHC representa el balance entre el agua que entró y la que salió de una determinada región o ecosistema. Para que el ecosistema se mantenga sano, se va a desear que el ingreso sea equivalente al egreso sumado a un saldo extra. Uno de los principales factores que afectan a la fuga de agua en el sistema de estudio es la evapotranspiración de las plantas en la zona. Por el contrario, la mayor fuente de aumento de volumen hídrico refiere a las precipitaciones ocurridas.

La ecuación que describe el Balance Hídrico expresa lo siguiente

$$P = E + R + I \quad (4.2)$$

En donde

P = Precipitaciones atmosféricas

E = Evapotranspiración

R = Escorrentía

I = Infiltración

La precipitación consiste en el agua meteórica que cae sobre la corteza terrestre. Si bien en la región de los pastizales del Río de la Plata ésta consiste mayoritariamente en lluvias, también se contemplan las cantidades provenientes de granizo, nieve, nieblas, rocío, humedad en el aire y demás formas posibles de depositar agua en los suelos. Se poseen varios equipos

de medición para estos fenómenos como pluviómetros, nivómetros, etc. por lo que la medición de las precipitaciones no suele presentar un desafío para el cálculo de BHC. No obstante, para obtener la precipitación total caída sobre la zona de estudio, se necesita conocer exactamente la zona de alimentación verdadera o hidrogeológica. En caso de no contar con este dato, se deberá estimar a partir de observaciones meteorológicas de la zona que estén a la misma altitud que aquella de interés. En la cotidianeidad, prevalecen dos tipos de mediciones de precipitaciones: el método de *Thiessen* (que requiere de algunos pluviómetros y algunas operaciones geométricas básicas) y el método de las Isohyetas¹ (que toma el estudio de estas curvas para obtener una precipitación media en la zona).

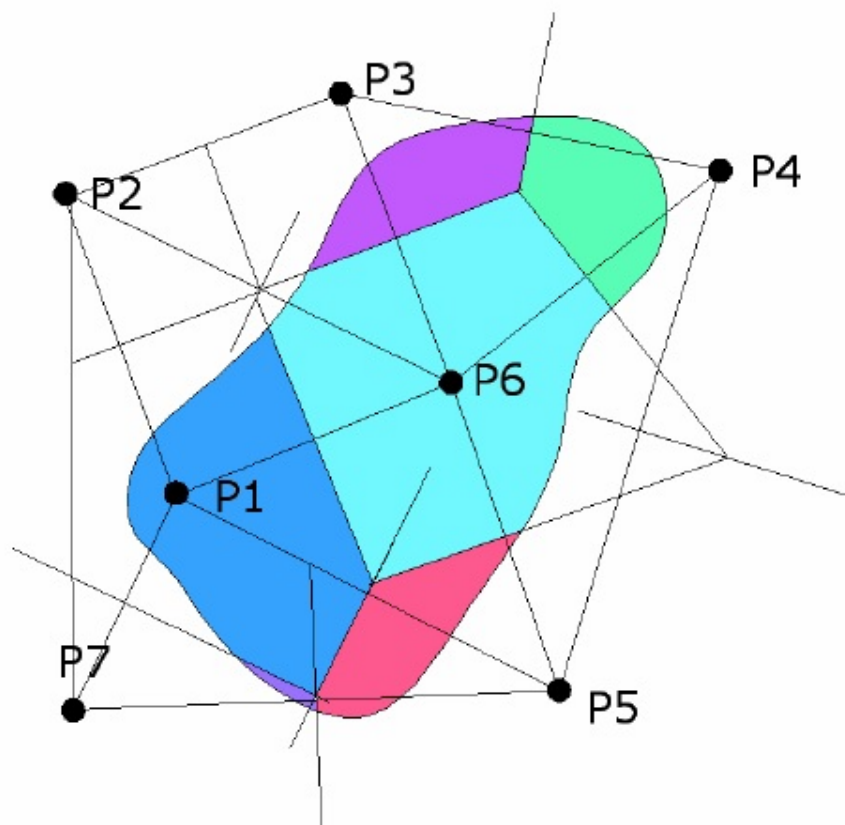


Figura 4.2.1: Demarcaciones sobre una región arbitraria para la aplicación del método de Thiessen

¹Una isohyeta es el lugar geométrico de los puntos sobre los que cae la misma cantidad de lluvia en un período de tiempo determinado.

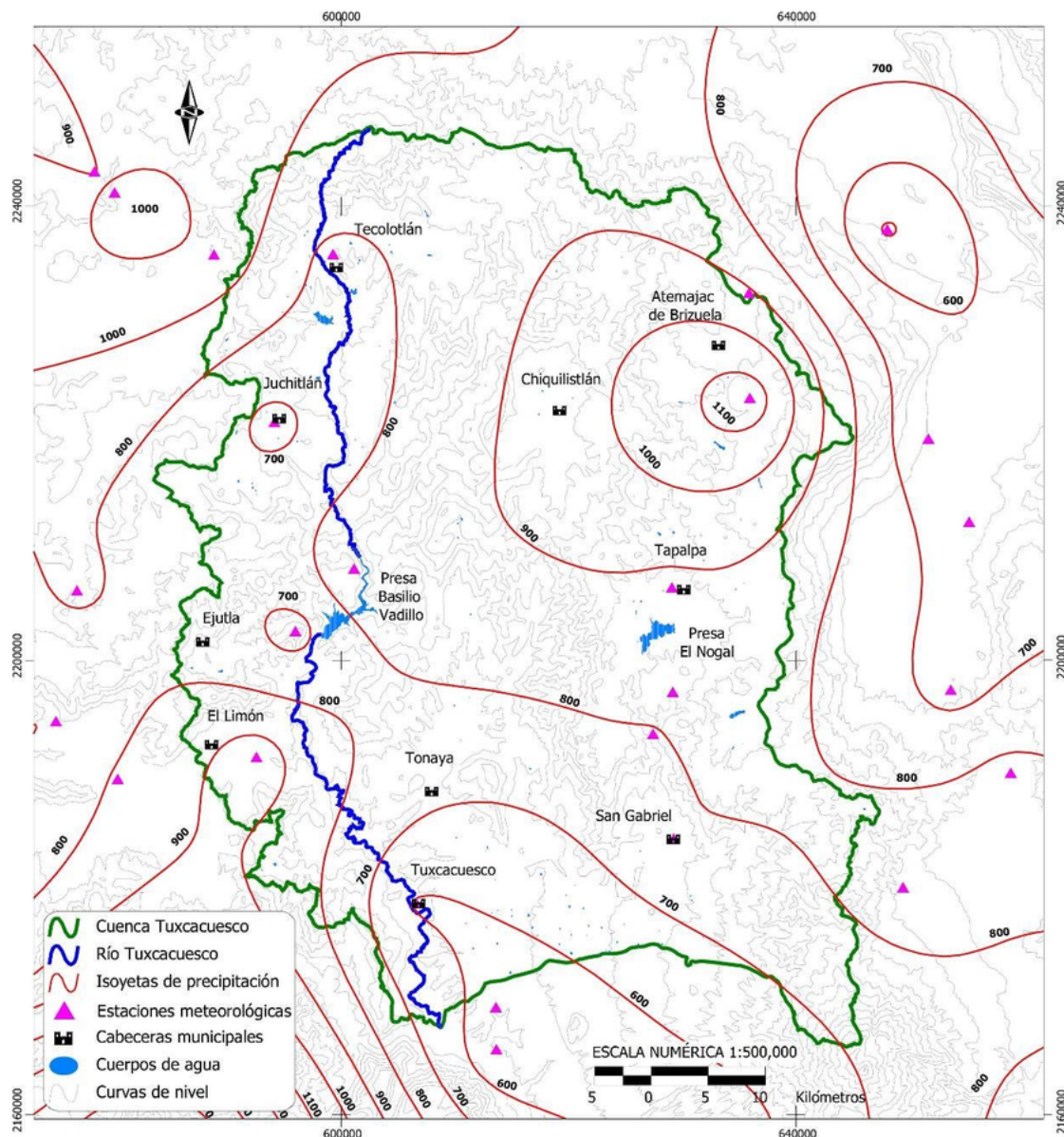


Figura 4.2.2: Isohietas de precipitación en una cuenca hídrica

La evapotranspiración se comprende de dos factores: un fenómeno físico del agua como lo es la evaporación, que se puede realizar en el aire, suelo, etc. y un fenómeno biológico, la transpiración, dada por la absorción de agua por parte de las plantas. Esto se puede observar gráficamente en la *Figura 4.2.3*. Dicha recolección presenta un alto porcentaje del destino del agua de precipitaciones y es por esto que se trata a la evapotranspiración como uno de los principales agentes del BHC. Este fenómeno depende principalmente de la temperatura del aire y agua, los rayos solares, la presión atmosférica, el viento, entre otros. La evapotranspiración se divide en dos procesos:

- Evapotranspiración real = E_r : aparece en las condiciones naturales de humedad en el

suelo en un instante cualquiera. Está ligada al porcentaje de humedad del suelo, el cual a su vez depende mayoritariamente de las precipitaciones.

- Evapotranspiración Potencial = E_p : es aquella que aparecería si existiese una continua reserva del agua para reponer sus pérdidas por los distintos medios. Si las precipitaciones fueran abundantes y perdurables en el tiempo, se tendría que $E_r = E_p$ mientras que en épocas de sequía $E_r < E_p$.

Al estar la evapotranspiración real tan estrechamente ligada con la humedad del suelo, su valor es variable en un mismo terreno en función del tiempo. Aquí es donde la E_p entra en juego, proponiendo un valor constante para un terreno cuyas pérdidas de agua han sido compensadas.

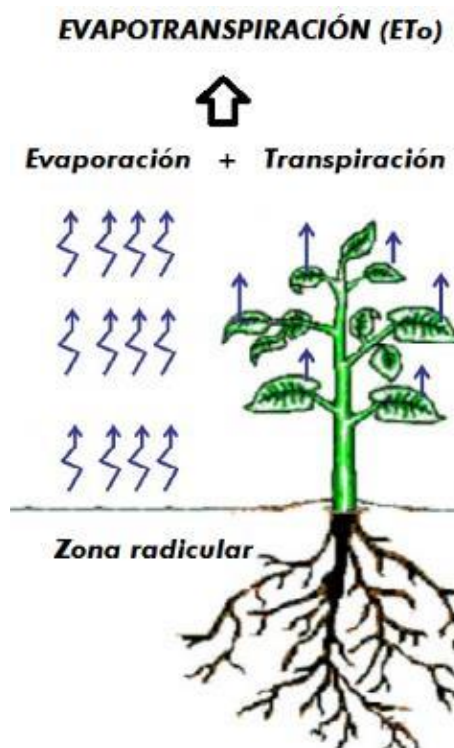


Figura 4.2.3: Representación gráfica de la composición de la evapotranspiración

Por otro lado, la esorrentía consiste en aguas que se escapan a la evapotranspiración y a la infiltración, ergo pueden ser de circulación superficial o subterránea (conocidas como aguas de circulación hipodérmica). Las aguas de esorrentía dependerán de la permeabilidad del suelo y la pendiente del terreno, entre otros, por lo que su existencia estará íntimamente ligada a la composición y estructura del terreno. Para medir el caudal de estas aguas existen numerosos métodos y fórmulas que pueden requerir leves modificaciones al terreno o diversos elementos de medición.

En cuanto a la infiltración, se menciona que ésta ocurre cuando el agua de lluvia ha completado el déficit de capacidad de agua de la zona de evapotranspiración. Ahí es cuando se infiltra a los suelos y se llega a una etapa de circulación hipodérmica hasta que parte de ella pase a darse como infiltración profunda (la que se evaluará en esta ecuación y por tanto se debe considerar la parte que se pierde circulando y rellenando el acuífero). Se podrá medir la infiltración a partir del uso de lisímetros en la zona de interés.

El uso del BHC como indicador funciona mejor que algunas de sus alternativas (como por ejemplo el índice de aridez de la región) para hacer un uso eficiente de los recursos de agua dulce así como también en su estudio hidrológico. Esto se debe a que las condiciones hidro-climáticas suelen describirse a partir de los factores que componen al Balance Hídrico Climático. El índice tomará valores positivos o negativos según si predomina la potencial evapotranspiración o las precipitaciones. Así, permite indicar si se está en situación de tener un excedente o un déficit en el recurso hídrico a partir de la incidencia del clima en la región y provee un método de evaluación de las ganancias y pérdidas de agua sufridas por el suelo en períodos de tiempo definidos.

El BHC presenta una alta variabilidad temporal y espacial, principalmente dado que las precipitaciones de una zona varían considerablemente según su ubicación y la época del año. Es importante entonces no sólo que todas las aguas medidas pertenezcan a la misma unidad hidrogeológica (es decir, que provengan del mismo acuífero) sino también que las mediciones realizadas deben tener al menos un año de seguimiento. Además, la medición de la evapotranspiración no siempre se encuentra para las regiones de interés, por lo que se dispone de varios modelos de aproximación matemática para obtener su valor, los cuales contemplan distintos criterios. Entre ellos se mencionan por ejemplo el de *Thornthwaite-Wilm* que sólo considera a la temperatura como parámetro climático o el de *Blaney-Criddle* el cual además de la temperatura incluye a la velocidad del viento y el porcentaje de humedad relativa en su estimación.

4.3. Redes Neuronales

La creación de las redes neuronales, contrario a la creencia popular, es un hecho que data de mediados del siglo pasado. Sin embargo, hoy en día éstas han retomado un papel protagónico y su conocimiento se vuelve menester. Las redes neuronales consisten un método de aplicación de la inteligencia artificial. Dicho de una forma sencilla, la inteligencia artificial un intento de los desarrolladores de descubrir y describir aspectos de la inteligencia humana que pueden ser simulados mediante máquinas. Así, se propone un método de procesamiento de la información por parte de los ordenadores que emule el funcionamiento de la parte fundamental del procesamiento humano: la neurona.

Al continuar con esta analogía de la inteligencia humana como pilar de la inteligencia artificial, se puede comprender mejor qué tipo de problemas ataca el uso de esta herramienta. Ésta toma importancia en cuestiones que no se puedan resolver necesariamente con un algoritmo, sino que en todas ellas prima una característica resolutive por sobre el resto: la experiencia previa. Como el ser humano, quien se apoya en situaciones transitadas previamente para evaluar una problemática, la inteligencia artificial buscará copiar ese *modus operandi* para arrojar los resultados deseados.

El principio de funcionamiento de las redes neuronales es, a grandes rasgos, siempre el

mismo. Se toman estímulos de entrada (*inputs*) que se procesan para arrojar resultados impulsados por dichos estímulos (*outputs*). Las redes neuronales tendrán la particularidad de que, al igual que el ser humano, tienen la capacidad de “aprender” para ir mejorando su funcionamiento. Ese aprendizaje se dará por el agregado de nuevos *inputs*, modificaciones en el procesamiento mencionado u otras cuestiones propias de la programación de estas herramientas. Se menciona además que, en esencia, las redes neuronales son unidades de procesamiento que permiten el intercambio de datos o información.

Ya despegándose de la analogía biológica utilizada para los lineamientos generales descritos, el desarrollo de las redes neuronales con el tiempo se a moldeado a las características particulares de la utilización de las mismas. Así, los modelos tecnológicos que fueron surgiendo no tienen por qué guardar una estrecha similitud con las cuestiones neuronales humanas. Por el contrario, los pormenores estarán dados por la necesidad de la aplicación que desea se cumplan. Se expondrán a continuación cuestiones más propias del funcionamiento de las redes neuronales como herramientas computacionales, haciendo hincapié en aquellas ramas de la tecnología aplicadas en el proyecto en cuestión.

4.3.1. Funcionamiento General de las neuronas

Las redes neuronales están compuestas, valga la redundancia, de neuronas (en este caso, artificiales). Es por ello que para comprender cómo funcionan estas redes primero se debe comprender cómo funciona su partícula fundamental. Para explicar qué son estas neuronas artificiales, también conocidas como perceptrones, se utilizará el primer modelo elemental que se planteó de ellas, en 1943. En este año el psiquiatra Warren McCulloch y el matemático Walter Pitts postularon el modelo de nombre homónimo, el cual se muestra en la *Figura 4.3.1.1*.

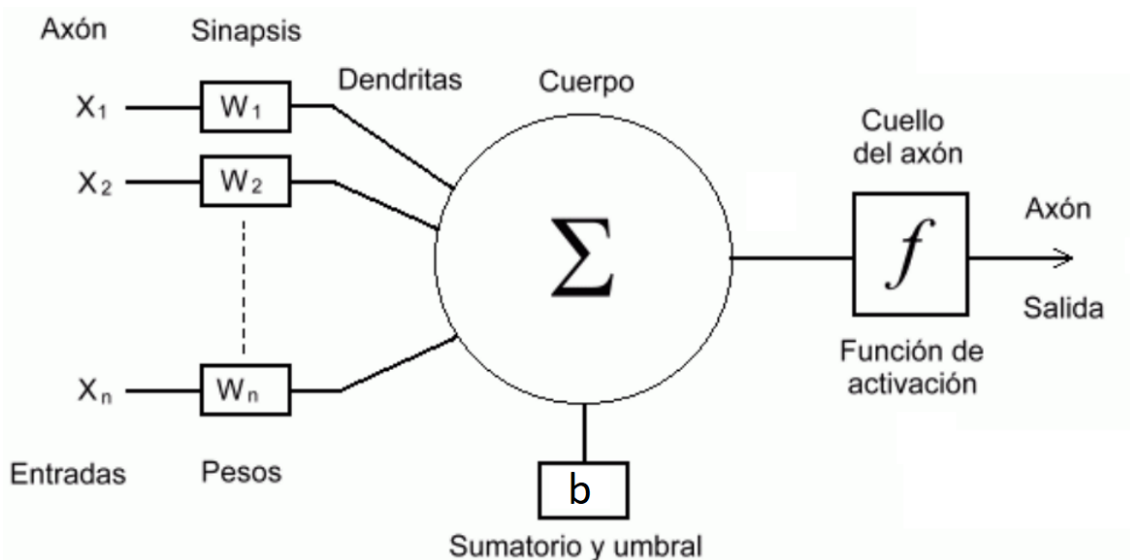


Figura 4.3.1.1: Modelo de una neurona artificial de McCulloch-Pitts

Para comenzar, se menciona que se han marcado en la figura algunas relaciones entre secciones del perceptrón y las partes de una neurona animal (las entradas como axón, los pesos como la sinapsis, etc.) que McCulloch y Pitts plantearon en su planteo, tal como se comentó al principio de esta explicación. Siguiendo el diagrama, se ve como los diferentes *inputs* X_i son modificados por los pesos sinápticos W_i para luego ser sumados entre sí. En esta sumatoria, se agrega un término aditivo b que actúa de sesgo o *bias*. Es a través del sesgo y los pesos sinápticos que el perceptrón se adapta al contexto de operación y aprende de él, lo cual se plasma en la modificación de estos parámetros, ya que las entradas representan dicho entorno de trabajo.

Luego, el resultado de la suma de todos estos parámetros se toma como la variable independiente de una función f conocida como Función de Activación. Esta expresión matemática sirve para calcular lo que se conoce como el estado de actividad de una neurona, transformando sus parámetros de entrada en un valor de activación. En el mundo de las redes neuronales, hay algunas funciones de activación típicas y éstas son las que, a priori, se evaluarán para la realización del Trabajo Práctico Profesional. Éstas se presentan a continuación. Se aclara que en todos los casos se tomará el valor de entrada a la función como x , en donde $x = \sum W_i X_i + b$

- Función lineal: acotada entre -1 (estado inactivo) y 1 (estado activo), propone que la salida de la función guarde una relación lineal con la entrada. Si se quiere que no haya cambio alguno, se planteará que la pendiente sea unitaria. La gráfica de la función resultante se puede observar en la *Figura 4.3.1.2*. La expresión matemática resulta:

$$f(x) = \begin{cases} -1 & x \leq -\frac{1}{a} \\ ax & -\frac{1}{a} < x < \frac{1}{a} \\ 1 & x \geq \frac{1}{a} \end{cases} \quad (4.3)$$

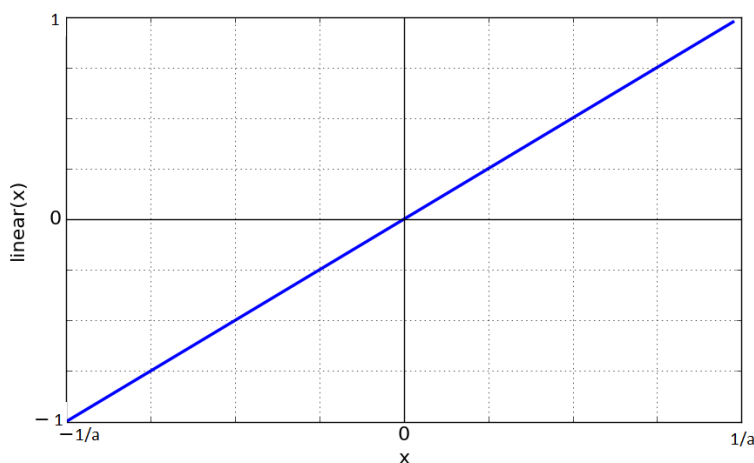


Figura 4.3.1.2: Gráfica de la Función de Activación lineal

- **Función Sigmoidea:** ahora la función toma valores entre 0 y 1, por lo que el estado inactivo se considerará como el valor nulo. A medida que se ajuste el parámetro denominado k , se irá variando la tangente de la curva de crecimiento de la función. A mayor valor de k , mayor será dicha pendiente. La función se puede apreciar en la *Figura 4.3.1.3* y la ecuación que la describe es

$$f(x) = \frac{1}{1 + e^{-kx}} \quad (4.4)$$

Se puede ver como los números de entrada grandes positivos pondrán a la neurona en estado activo (1) mientras que los negativos la llevarán a la inactividad (0). Si bien puede funcionar como una fiel representación de la velocidad de activación de la neurona, presenta como problema que la saturación cerca de los extremos del conjunto imagen genera un gradiente igual a cero, lo cual no es deseable.

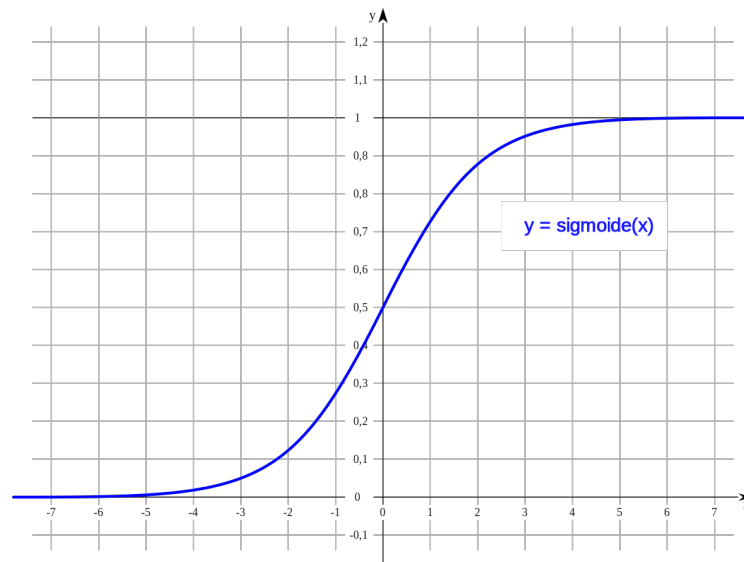


Figura 4.3.1.3: Gráfica de la Función de Activación Sigmoidea

- **Función Tangente Hiperbólica:** de comportamiento similar a la sigmoidea, corrige uno de los principales problemas que aquella presentaba: las salidas están centradas a cero. Esto es porque los valores que toma la función Tangente Hiperbólica se reparten entre -1 y 1. Este factor se puede apreciar en su forma gráfica en la *Figura 4.3.1.4*. Sin embargo, sigue presentando el mismo problema de saturación en los extremos que entorpecerá el correcto funcionamiento de la Red Neuronal. La pendiente de esta función de activación será adaptada a la utilidad modificando el parámetro k de su expresión matemática. La misma se presenta a continuación.

$$f(x) = \frac{e^{kx} - e^{-kx}}{e^{kx} + e^{-kx}} \quad (4.5)$$

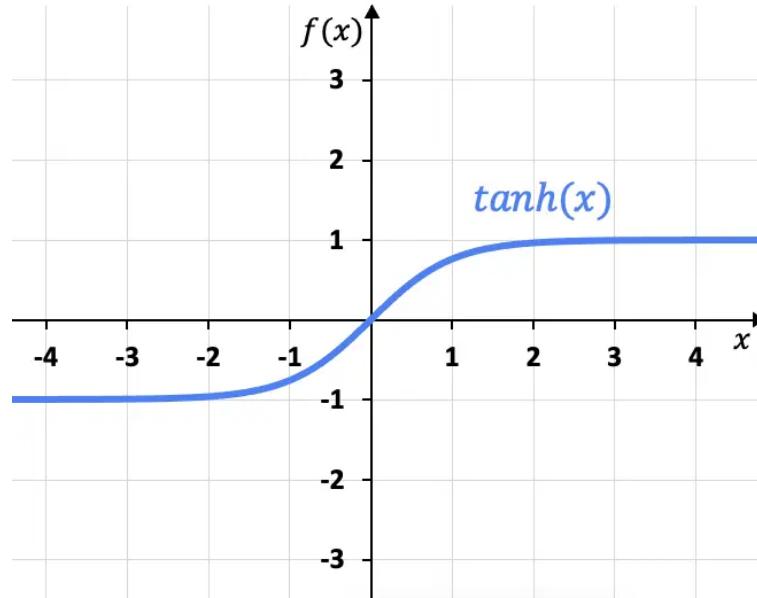


Figura 4.3.1.4: Gráfica de la Función de Activación Tangente Hiperbólica

- Unidad Lineal Rectificada: normalmente conocida como ReLU, esta función define la activación de la neurona cuando la entrada supera al 0. Así, la función anulará los valores negativos a la entrada y se quedará solo con los positivos sin modificarlos en módulo. Sobre ésta función se han ideado distintas variaciones como la *Leaky ReLU* o la *Randomized ReLU* de las cuales no se hablará en este trabajo puesto que no se considerarán para el proyecto. A continuación se presenta una expresión matemática de la ReLU junto con una imagen representativa del comportamiento de esta función.

$$f(x) = \max(0, x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases} \quad (4.6)$$

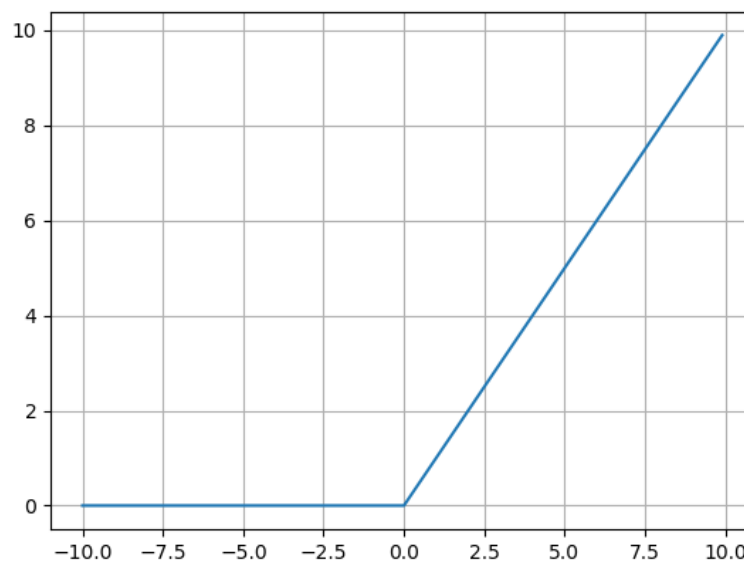


Figura 4.3.1.5: Gráfica de la Función de Activación ReLU

4.3.2. Organización neuronal

En la sección previa se trabajó el concepto de la neurona artificial y su funcionamiento. Pero, como el nombre de la tecnología lo indica, las Redes neuronales comprenden muchos nodos interconectados entre sí. De este modo, se debe establecer la arquitectura de las neuronas para generar el funcionamiento deseado. La distribución de éstas se hace, generalmente, en forma de capas, en donde cada una de ellas tendrá un número determinado de neuronas. Según su función dentro de la red, se distinguen a priori tres tipos de niveles.

- Capa de entrada: Recibe la información proveniente de las fuentes externas a la red.
- Capas ocultas: son aquellas intermedias entre la entrada y la salida, y como tales no tienen contacto directo con el entorno exterior. La cantidad de niveles ocultos que puede haber en una implementación dependerá de la misma. Las interconexiones entre estas capas, junto con la cantidad existentes, serán las que determinen la topología de la red empleada.
- Capa de salida: es aquella encargada de transmitir la información obtenida de la red neuronal al exterior.

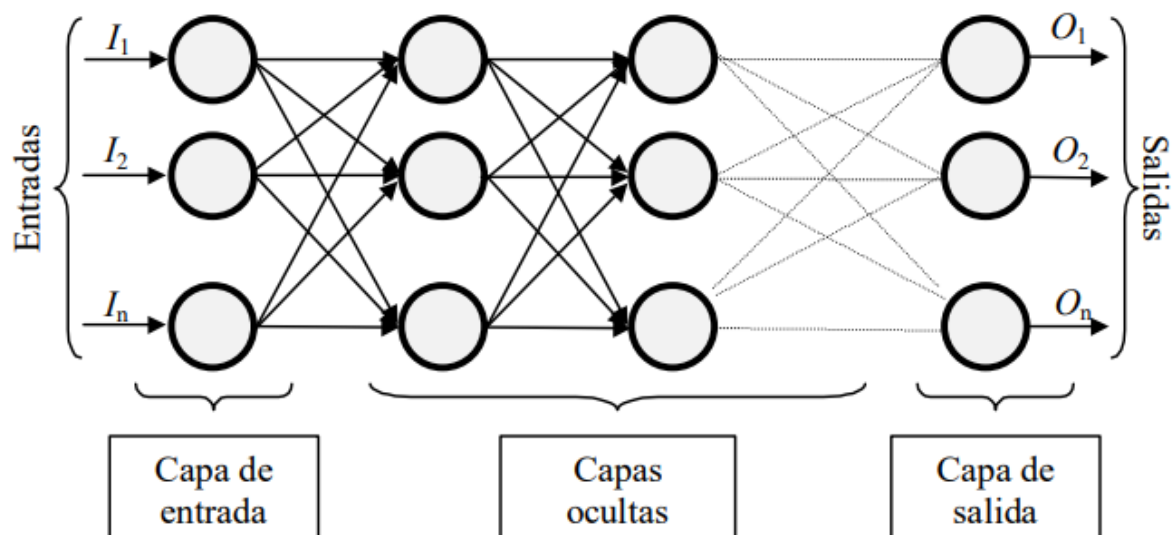


Figura 4.3.2.1: Representación gráfica de la arquitectura de capas de redes neuronales

Como se puede apreciar en la *Figura 4.3.2.1*, cada nodo únicamente estará conectado con neuronas de los niveles adyacentes. Además, se observa que normalmente habrá más interconexiones que neuronas en sí, ya que éstas estarán conectadas a uno o más nodos de los otros niveles. Estos vínculos entre neuronas serán una parte fundamental del proceso de decisión de la Red Neuronal

4.3.3. Métodos de aprendizaje

Recordando una parte de la analogía biológica previamente planteada, las redes neuronales tienen la capacidad, como el humano, de aprender. Para ello, se han definido distintos métodos de aprendizaje convencionales para lograr este acometido. Así, mediante este procedimiento, se irán variando las conexiones del sistema para proporcionar la salida deseada. A partir del aprendizaje de la red, se van a poder extraer generalizaciones a partir de un conjunto de ejemplos anteriores y otra información relevante que aporte al parámetro a evaluar.

Este proceso de aprendizaje se denomina entrenamiento, y para él se requiere un conjunto de datos dedicados exclusivamente a entrenar al modelo. En general, se analizará este *pool* de datos de manera aleatoria a lo largo del desarrollo del aprendizaje. A partir de este proceso, se irán realizando modificaciones en el comportamiento de la red para obtener los resultados deseados. Recordando el diagrama de la neurona artificial de la *Figura 4.3.1.1* y el esquema de capas de la *Figura 4.3.2.1*, se pueden ver todas las variables que hay en una Red Neuronal. En esta tecnología, se determina que la topología de la red y las funciones de la neurona no cambiarán durante este proceso de aprendizaje. No obstante, los pesos de cada nodo sí podrán variar. Es entonces que al entrenamiento de la red se lo determina como un proceso de adaptación de pesos.

Los pesos tendrán un valor inicial para todas las conexiones de las neuronas. Normalmen-

te, estos se determinan de manera aleatoria entre un intervalo arbitrario. El aprendizaje los modificará por valores que no necesariamente estén dentro de dicho intervalo. A partir de esto, se pueden generar cambios que van desde la creación de conexiones entre neuronas (cuando el peso pasa de cero a un valor distinto) hasta la destrucción de las mismas (en caso de que pase de un peso determinado a cero), incluyendo todas las modificaciones posibles del conexionado entre medio de esos dos términos.

En este punto, podría resultar pertinente preguntarse cómo se realiza esta modificación de pesos. Es aquí donde surge típicamente una división del aprendizaje en dos categorías principales: el aprendizaje supervisado y el aprendizaje no supervisado. En ambos, la estructura general del cambio de los valores será $Peso\ Nuevo = Peso\ Viejo + \Delta Peso$. De forma matemática, esto se expresa:

$$w_{ij(t+1)} = w_{ij(t)} + \Delta w_{ij(t)} \quad (4.7)$$

A continuación, se describen brevemente los dos tipos de entrenamientos arriba mencionados.

- **Aprendizaje Supervisado:** presenta a la red las salidas que debe proporcionar ante las señales que se le presentan. Luego se observa la salida de la red y se determina la diferencia entre ella y la señal deseada. Según el error cometido, se corrigen los pesos. Se requiere de un agente externo, supervisor, que determina lo que se considerará como respuesta correcta”.
- **Aprendizaje No Supervisado:** prescinden de la influencia externa para ajustar los pesos. Por el contrario, la red se organizará a ella misma agrupando las señales de entrada según algún concepto de similitud entre ellas. Así, la red neuronal no tiene información del entorno que le permita determinar el grado de bondad de su respuesta. A su vez, no se conocerá la señal que se desea como resultado a la salida de la red neuronal. Para evaluar la salida, se puede tomar a ésta como la representación del grado de similitud entre la información que se le ha presentado a la entrada con datos del pasado.

El proceso total de entrenamiento terminará cuando se detenga la variación de los pesos. Es decir, cuando $\frac{dw_{ij}}{dt} = 0$ ². Para determinar el momento de corte, se pone una condición de finalización. Esta pauta puede estar basada en la cantidad de ciclos de entrenamiento completados o en la superación de un umbral o error máximo deseado entre las salidas deseadas y las obtenidas.

4.3.4. Validación de los resultados

Como ya se explicó en la sección anterior, de la -idealmente- gran base de datos que se tiene para el desarrollo de la red neuronal, se debe declarar a un conjunto de valores como aquellos dedicados a entrenar al modelo. No obstante, no todos los datos pueden pertenecer a ese grupo, ya que una aparte se deberá destinar parte de la información poseída a la validación de la

²Nótese que ahora se utiliza un doble subíndice para los pesos, siendo i la numeración de la neurona que emite el estímulo (o la entrada) y j aquella de la neurona de la capa siguiente que recibe el estímulo en su entrada.

red resultante del proceso de entrenamiento. Para ello, se determina un conjunto de muestras llamado conjunto de validación o conjunto de testeo. Los datos pertenecientes a este subgrupo deberán tener información de los valores de las variables de entrada a la red neuronal y el valor de salida correspondiente medido, no estimado por el producto. A diferencia del conjunto de entrenamiento, no se “alimenta” a la red con la solución medida, sino que se comparará la solución calculada por el programa con la tabulada para cada ejemplo del conjunto.

Para poder reducir el sobreajuste, logrando que la red sea capaz de predecir datos nuevos y exponer poco el conjunto de testeo, se particionará el conjunto de datos en tres subconjuntos. El primero es el de entrenamiento, cuya función ya se explicó en la sección previa. Luego estará el conjunto de validación y finalmente el de testeo. El tamaño de cada subconjunto será distinto según la función que cumpla.



Figura 4.3.4.1: Subconjuntos de datos

Se utilizará el conjunto de validación para evaluar los resultados del conjunto del entrenamiento y continuar ajustándolos. Una vez que se obtenga el modelo que se desempeñe mejor mediante la aplicación del conjunto de validación se podrá evaluar el comportamiento a partir del conjunto de testeo. Se muestra a continuación un esquema gráfico del este procedimiento.

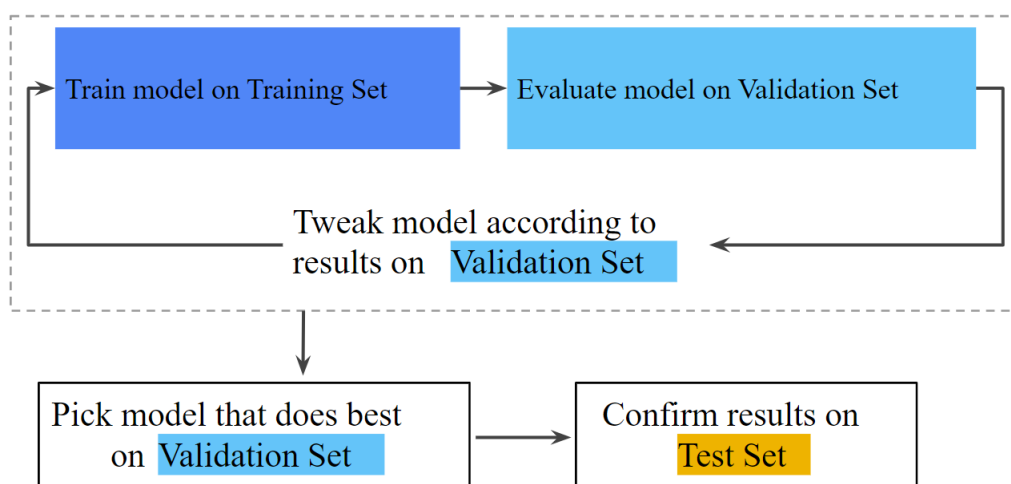


Figura 4.3.4.2: Flujo de entrenamiento y validación

La capacidad de un modelo para generalizar a nuevos datos esta dado por límites de la complejidad del modelo y el rendimiento del modelo en los datos de entrenamiento. Para evitar la complejidad del modelo que provoca el sobreajuste de los datos, existe la regularización, que se basa en la penalización de modelos complejos. El objetivo es minimizar la pérdida del modelo (minimización del riesgo empírico), la complejidad del modelo (minimización del riesgo estructural) y lograr un equilibrio adecuado entre simplicidad y ajuste de datos de entrenamiento.

$$\text{Minimizar}(P\acute{e}rdida_{(Datos/Modelo)} + \lambda \cdot Complejidad_{(Modelo)}) \quad (4.8)$$

Siendo λ la tasa de regularización, si su valor es demasiado alto el modelo será simple pero corre el riesgo de subajustar los datos, por lo tanto no aprenderá lo suficiente para poder hacer predicciones útiles. Si el valor es demasiado bajo, el modelo será más complejo pero se corre el riesgo de sobreajustar los datos dado que aprenderá demasiado sobre las particularidades de los datos de entrenamiento y no podrá generalizar sobre datos nuevos.

Para evaluar la eficiencia de predicción del algoritmo, se pueden utilizar distintos métodos complementarios entre sí. Por un lado, se puede utilizar expresiones matemáticas y probabilísticas que representen distintos factores de interés. Estos pueden ser errores entre la salida teórica y la calculada, probabilidades de fallo en la predicción, etc. Además, se puede configurar el programa para que aporte resultados en forma gráfica u otra manera organizada para que se constaten con los obtenidos de manera teórica en ese mismo formato. Este segundo método de validación depende de la capacidad del desarrollador de evaluar la información provista por el programa. Es decir, no se apunta a que la red neuronal reconozca las diferencias de lo que arroja sino que sólo se le pedirá que presente los datos de una manera más amena para su estudio. Los parámetros y formas a evaluar serán particulares de la aplicación desarrollada.

4.3.5. Backpropagation

El algoritmo de *Backpropagation* provee un método de entrenamiento de redes multicapa. Una de sus principales virtudes es que provee la posibilidad de entrenar varias capas ocultas, lo cual permitió en su momento superar las limitaciones de las redes monocapa. Este tipo de redes tiene un método de entrenamiento supervisado: a la red se le presentan pares de patrones, uno de entrada y otro de salida, y con cada una de estas adiciones se ajustan los pesos para disminuir el error entre la salida deseada y la respuesta de la red. El algoritmo de *Bacpropagation* consta de dos fases, las cuales se realizan para cada conjunto de datos presentados. Estas son

- Propagación hacia adelante: se da al ingresar un patrón en la capa de entrada de la red, en donde cada neurona de la entrada se corresponde con un elemento de dicho patrón. Los nodos de la capa toman este valor y lo procesan según se explicó en la **Sección 4.3.1**. Este valor de salida se propaga hacia todas las conexiones de salida de la neurona en cuestión. Este proceso se repetirá en cada capa hasta generar una salida.
- Propagación hacia atrás: al terminar la etapa previa, se inicia esta fase que apunta a la corrección. Para ello, las señales de salida se comparan con sus correspondientes salidas deseadas y se genera, para cada una de ellas, una señal de error. Este error es lo que se

propaga hacia atrás, partiendo de la capa de salida hacia todas las neuronas de la capa oculta que estén conectadas a ella. Lo que se recibe no es el error completo, sino que una fracción de éste según el peso que la neurona tenga en el resultado final a la salida. Este proceso se repetirá en todas las capas y será a partir del valor recibido en cada una de ellas que se realizará la modificación de los pesos de conexión de cada neurona. Se buscará entonces que la red converja a un estado en el que se puedan clasificar correctamente los patrones de entrenamiento.

A medida que se entrena la red, las neuronas irán reconociendo patrones a la entrada y podrán regenerar particularidades de un *input* de datos que pueda estar incompleto si en éste reconocen cualidades de los ya estudiados. Para ello, la red hará uso de dos conceptos matemáticos fundamentales en el algoritmo de *Backpropagation*: la derivación en cadena y el descenso del gradiente.

La regla de la cadena se utiliza para propagar el error hacia las distintas capas en la etapa de propagación hacia atrás. Con la regla de la cadena se calcularán los gradientes de la función de pérdida (distancia del valor obtenido al deseado) desde la salida hacia atrás. Esto se debe a que utilizando esta herramienta derivativa se podrá descomponer el gradiente de la función de pérdida en función de una serie de gradientes parciales a lo largo de cada capa de la red. Con los gradientes calculados, se ajustarán los pesos del sistema. Es importante destacar que estos gradientes indicarán la dirección y magnitud del cambio necesario para minimizar el error. A continuación se expresa la formulación general matemática de la regla de la cadena:

$$(g \circ f)' = g'_{(f(x))} f'_{(x)} \quad (4.9)$$

Por otro lado la aplicación del descenso del gradiente implica una técnica de optimización para minimizar la función de pérdida. Luego de calcular los gradientes de la forma ya explicada, la actualización de los pesos se hará con ayuda del descenso del gradiente. Se propone que $Peso_{Nuevo} = Peso_{Actual} - Tasa_{Aprendizaje} \cdot Gradiente$, en donde la tasa es un parámetro arbitrario elegido antes de comenzar el algoritmo que controlará la rapidez de convergencia del mismo. Este proceso de actualización se hará para todos los pesos de la red y se irá iterando para los distintos sets de datos de la base de entrenamiento. Este proceso se detendrá cuando la función de pérdida alcance un valor mínimo y se estabilice. Como se puede analizar de la *Figura 4.3.5.1*, se comienza con un valor inicial de peso (*Start*) el cuál se irá modificando en cada iteración según el gradiente calculado (representado por las flechas rojas). El largo del paso de modificación en cada iteración estará dado por la tasa de aprendizaje previamente definida. La dirección de los ajustes de los pesos es tal que se dirijan hacia la abscisa que minimice la función de pérdidas y, en consecuencia, el error de la salida de la Red Neuronal respecto del valor correcto. En cada paso se mueve según la derivada de la función hasta que esta se haga nula, marcando un mínimo local o absoluto y finalizando el proceso de iteración.

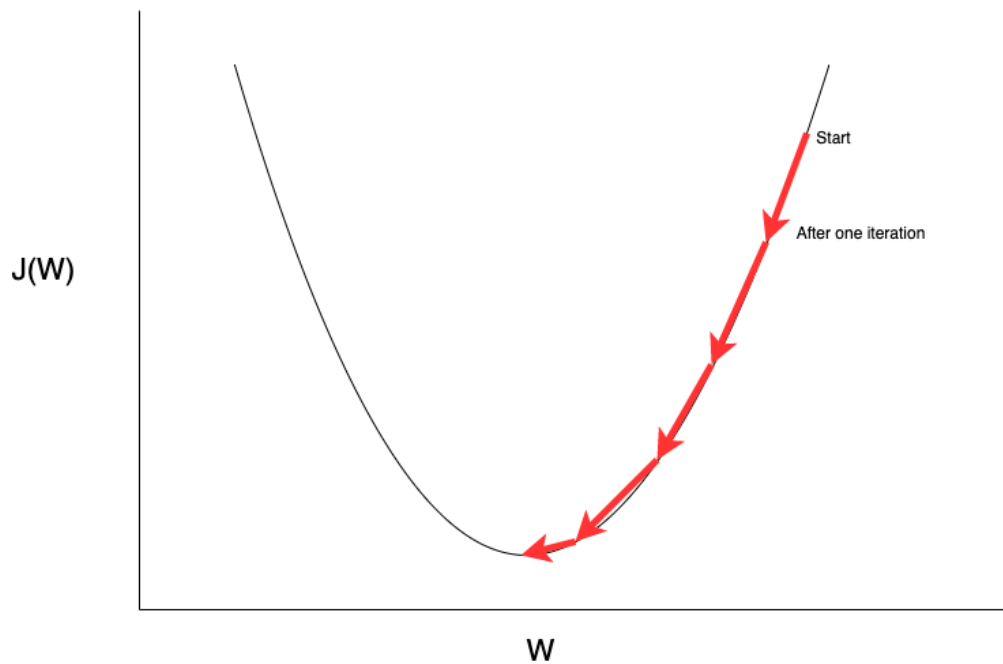


Figura 4.3.5.1: Representación del proceso de descenso del gradiente para una función de pérdidas $J(W)$ arbitraria según el valor de los pesos (W)

El algoritmo terminará si la red presenta una convergencia. Se suele utilizar como parámetro cualitativo el Error Cuadrático Medio (Error RMS) de la red. En el ideal, el error RMS entre la salida obtenida con los valores reales debería ser nulo para todas las entradas utilizadas en el algoritmo. No obstante, se acostumbra un valor umbral de corte pequeño, de modo que a media que la red aprenda, el valor RMS decrezca hasta superar dicho umbral. La convergencia puede complicarse en términos computacionales y temporales o en la aplicación particular del algoritmo, en donde puede que la red deje de aprender en un cierto punto.

4.3.6. Redes Neuronales Recurrentes

Dentro de la categoría de Redes Neuronales y *Deep Learning* surge esta categoría creada en el siglo pasado llamada Redes Neuronales Recurrentes, o RNN por sus siglas en inglés. Una de las principales ventajas de que provee este tipo de tecnología es que permite el planteo de relaciones de dependencia entre sus entradas, tanto secuenciales como temporales. Otra cuestión importante a remarcar de las Redes Neuronales Recurrentes es que brindan la posibilidad de compartir parámetros. Así, el modelo puede correlacionar las secuencias de entrada, algo imposible en un modelo en el que se asignan parámetros únicos a cada dato.

Otro aspecto fundamental de las RNN es que se adicionan ciclos a su arquitectura de red. Estos ciclos estarán conectados a nodos y constituyen una “Memoria interna” de la red

que permite constatar el dato actual recibido con aquellos del pasado. Un boceto de este funcionamiento se puede ver en la *Figura 4.3.6.1*, en donde se agregan retardos temporales para representar el desfase de estos valores como entradas dados por el proceso de iteración de la capa neuronal previo a su realimentación.

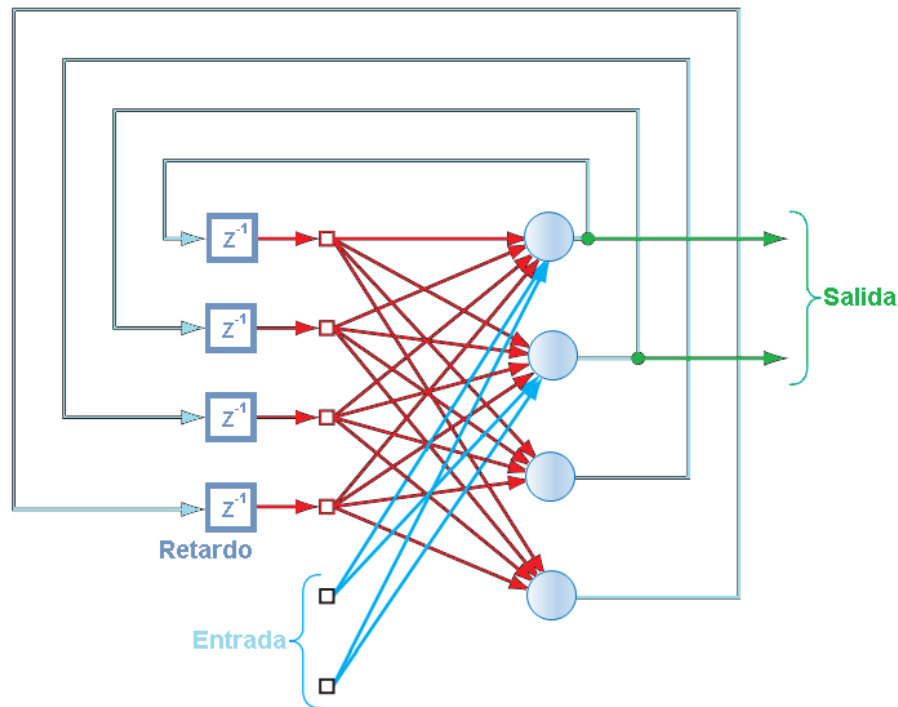


Figura 4.3.6.1: Esquema de la arquitectura de una Red Neuronal Recurrente

Si bien las Redes Neuronales Recurrentes presentan las ventajas mencionadas en los párrafos anteriores, algunos problemas surgen en sus diseños más básicos. Los principales surgen de la convergencia de la función y se dividen en dos inconvenientes muy conocidos en el mundo del *Deep Learning*: el *Vanishing Gradient Problem* y el *Exploding Gradient Problem* o Problema de Desvanecimiento del Gradiente y Problema de Explosión del Gradiente, respectivamente. Para explicar ambas situaciones, se recuerda que las RNN funcionan a partir de procesar secuencias de información, agregando cada elemento nuevo a lo que resulte del análisis anterior. En este contexto, puede darse que a medida que se procesen más y más elementos del grupo de datos de entrada, haciendo que la red “se olvide” de la información a medida que esta se aleja más en el tiempo de que se esté procesando. A medida que la RNN propaga el error del gradiente hacia las capas más viejas, éste varía exponencialmente. En caso de un crecimiento exponencial, se dice que el gradiente “explota” mientras que si tiende a cero, se dice que “se desvanece”. Esto genera que las Redes Neuronales Recurrentes más básicas tengan una precisión apreciable solo en rangos cortos de tiempo, lo cual podría significar problemas en la implementación deseada.

Para solucionar estos problemas comunes, se pueden plantear otro tipo de RNN que fue diseñado específicamente para ello. Éstas son las redes neuronales recurrentes de memoria de corto y largo plazo, o LSTM (*Long Short Term Memory*). La arquitectura de una celda LSTM se puede observar en la *Figura 4.3.6.2*.

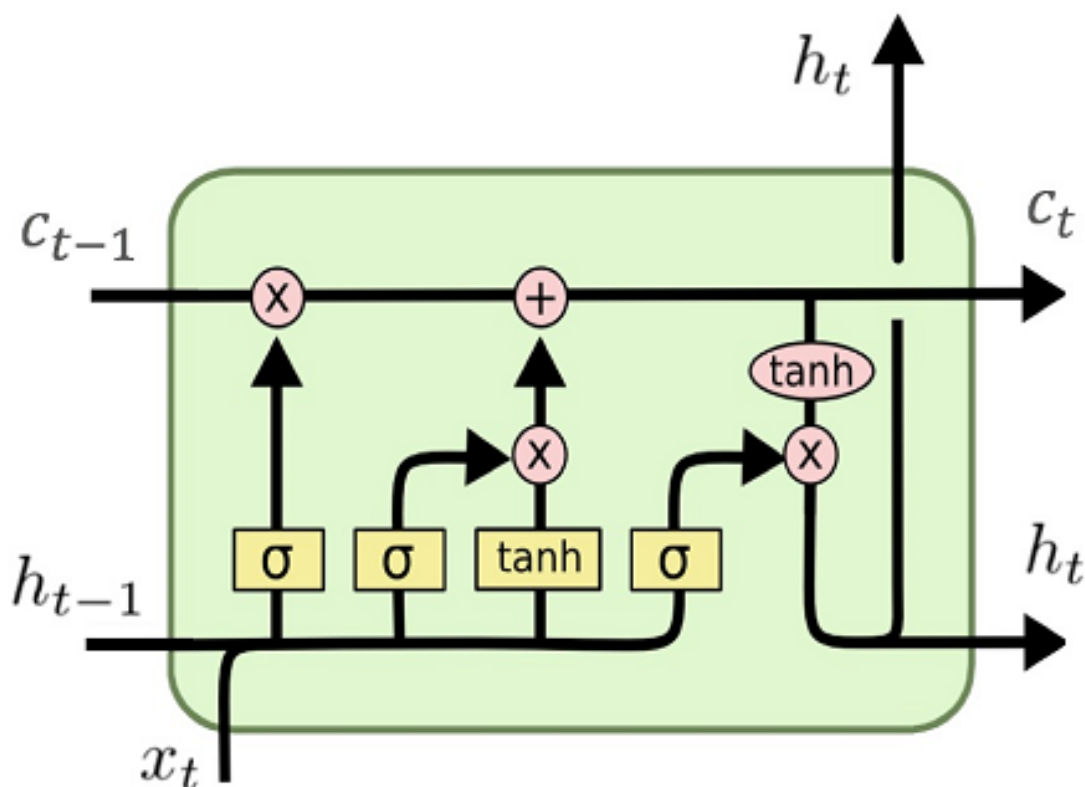


Figura 4.3.6.2: Diagrama de la arquitectura de las redes LSTM

Lo que aquí se puede ver es sólo una componente de la red, por lo que la información que llega a ésta en los terminales de la izquierda proviene del previo procesamiento del dato asociado a un período de tiempo. La entrada del terminal x_t refiere a la información de la unidad anterior de la LSTM. Las salidas de la derecha servirán de información para la siguiente componente que actuará igual que el mostrado en esta imagen. El procesamiento interno buscará alterar los valores de ponderación de la información que arriba al nodo para que la Red Neuronal recuerde lo importante y “se olvide” del resto. La celda LSTM permitirá agregar datos nuevos y eliminar aquellos que ya no aporten a la deducción actual que esté realizando la red. La aplicación de la función sigmoidea en algunas de las entradas (los bloques amarillos σ) permite un comportamiento valvular: permitirá el paso de los datos ingresantes a la celda o no (recordando la **Sección 4.3.1**, la sigmoide toma valores entre 0 y 1). Las funciones

tanh, ya explicadas previamente, actuarán para incluir la información nueva que llega a la celda al resultado que se pase a las celdas subsiguientes. En líneas generales, esto permitirá ir ponderando la información a través de la red para evitar su crecimiento exponencial o su decaimiento hasta la intrascendencia.

4.3.7. Redes Neuronales Convolucionales

Otra alternativa posible de desarrollo son las Redes Neuronales Convolucionales (más conocidas como CNN por sus siglas en inglés o *Convnets* como contracción de *Convolutional Networks*). Este tipo de redes toma su inspiración de la naturaleza, siguiendo los lineamientos de desarrollo explicados en la **Sección 4.3**, dado que la conexión entre neuronas se asemeja a la organización de la corteza visual animal. Como tal, uno de sus principales usos es el de reconocimiento de imágenes. Es así que la información de entrada a la red se suele representar en un formato bidimensional o tridimensional (piénsese una foto con un valor de color, brillo, etc.). Por otro lado, son muy utilizadas en el procesamiento de cuestiones secuenciales como por ejemplo el procesamiento del lenguaje (categorizado como una secuencia unidimensional de caracteres) o la predicción de series temporales.

Como su nombre lo indica, la operación que predomina en las *Convnets* es la convolución. Más precisamente, se está realizando una convolución entre matrices. Esta operación, pilar fundamental de ciertas áreas de la Ingeniería Electrónica, en esencia arroja como resultado una tercera matriz o función expresando como la forma de uno de los operandos modifica a la del otro. La convolución en las capas de la CNN se da con matrices llamadas *kernels de convolución*. El contenido de cada coordenada del kernel varía según la función que se le va a dar a la capa. Se pueden ver algunos ejemplos y sus efectos en la *Figura 4.3.7.1*. Esta acción de convolución se irá realizando secuencialmente entre el kernel y la información de entrada, recorriendo (en caso de ser un arreglo 2D) de izquierda a derecha y de arriba abajo en un formato de ventana deslizante.

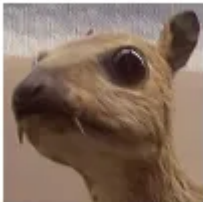
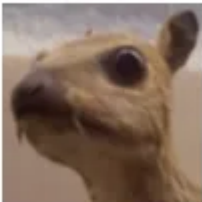
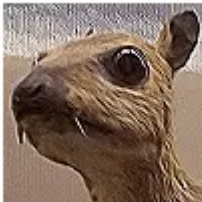

<i>Original</i>	<i>Gaussian Blur</i>	<i>Sharpen</i>	<i>Edge Detection</i>
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$
			

Figura 4.3.7.1: Ejemplos de kernel de convolución y sus efectos en el procesamiento de una imagen

Al analizar la arquitectura que las CNN presentan, se puede observar que siempre poseen como característica irrevocable que su primera capa en el modelo es una convolucional. Dicha capa será la que tome la carga computacional más alta. En ella se determina el tamaño y contenido de los kernels, la inclusión de *padding* para ajustar el tamaño de estos al de la entrada y se agrega una alinealidad con la aplicación de una ReLU. A esta capa se le pueden agregar subsecuentes capas convolucionales, lo que dará por resultado una estructura jerárquica entre ellas en la que se analice la entrada como secciones separadas en la primer capa y acabará por tomarse como un total en la última. Una representación gráfica de éste funcionamiento por jerarquías se puede ver en la *Figura 4.3.7.2*.



Figura 4.3.7.2: Representación gráfica del procesamiento de tres capas convolucionales sobre la imagen de una bicicleta

Luego del procesamiento convolucional, se agrega una capa de *pooling*. La acción que se realiza es similar a un *downsampling*, en donde se reduce la cantidad de parámetros de entrada. Con esto se obtiene una reducción del recurso computacional requerido para procesar la data. Se puede hacer un *Max Pooling* en donde se reduce una porción de datos únicamente al mayor de ellos o *Average Pooling*, que consiste en reducir la zona a un número que represente el promedio de todos los valores contenidos en ella. Normalmente, se utiliza el primero de estos dos métodos.

Finalmente, se suele utilizar una capa totalmente conectada (*Fully Connected*) para realizar

las interacciones alineales entre los datos de entrada. Para eso primero se suele “planar” los datos que salen de la capa de *pooling* hacia un vector unidimensional. Esto se hace para mejorar el procesamiento de la última capa, sea que se requiera categorización o regresión de los datos. En la capa totalmente conectada, valga la redundancia, todas las neuronas de ésta se encuentran conectadas entre sí para lograr predicción de salida requerida. En la siguiente figura se puede observar la arquitectura general aquí descrita para las Convnets; particularmente el diagrama de una red de clasificación de imágenes.

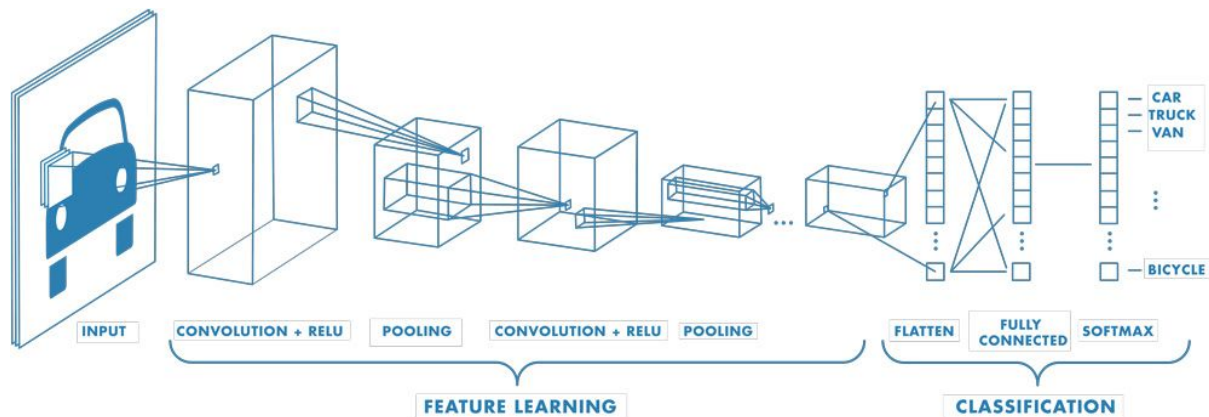


Figura 4.3.7.3: Representación gráfica de la arquitectura de capas de una Red Neuronal Convolutiva para la clasificación de imágenes

4.3.8. Data Clustering

El *clustering* es una técnica de *machine learning* no supervisada que consiste en dividir los distintos datos obtenidos en categorías o *clusters*. Se busca entonces que aquellos datos similares entre sí (según el criterio tomado para la división) queden agrupados en un mismo subconjunto, el cuál no estará dentro de los preexistentes. Así, se va a obtener una segregación cuantitativa de los datos a partir de uno o más factores arbitrariamente elegidos. Se va a intentar a partir de esto revelar al modelo estructuras escondidas dentro de la misma información, así como también simplificar sets de datos complejos. Un ejemplo gráfico del resultado del proceso se puede observar en la *Figura 4.3.8.1*.

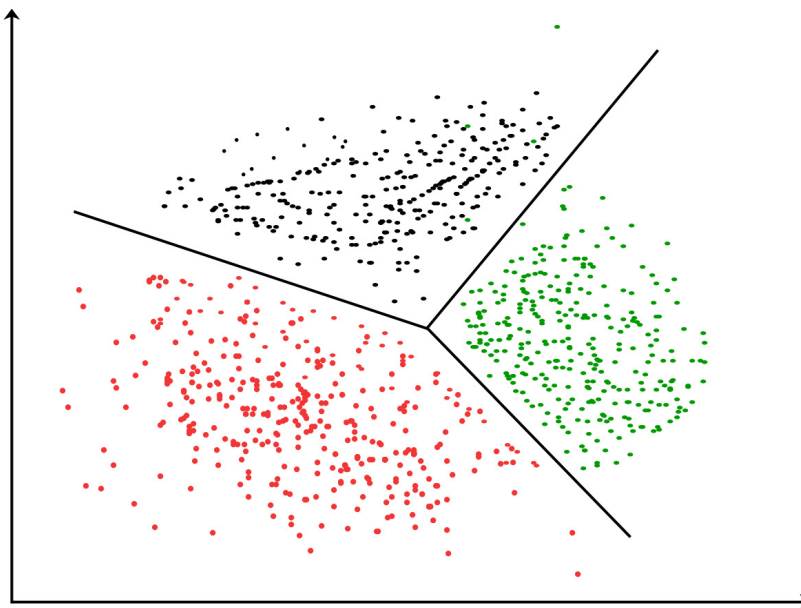


Figura 4.3.8.1: *datapoints* divididos en tres *clusters*, categorizados gráficamente por color

Como es común en los desarrollos computacionales, este método de modelado de la información de entrada tiene varias ramas que cargan con distintos pormenores para la separación y denominación de los *datapoints*. Mencionar y detallar todos ellos sería una tarea compleja y superflua. Por lo tanto se hará hincapié en lo utilizado en el Trabajo Profesional.

En primer lugar, se menciona que se hará un *hard clustering* o *clustering* duro. Es este el caso en el que cada dato que ingresa al modelo o bien pertenece a un *cluster* o no lo hace. Es decir, no se habla de una probabilidad de que pertenezca o no. En este método también conocido como *clustering* de partición se debe especificar la cantidad de *clusters* que se espera tener como resultado de manera previa a la división.

En cuanto al algoritmo de categorización utilizado, se decantó por un modelo de centroides. En este tipo de *clustering*, el programa elige un centroide y luego se itera para averiguar la proximidad del *datapoint* al mismo. Es importante destacar que el centro de cada fragmento no tendrá por qué ser un punto de los datos de entrada, sino que puede ser un valor que mejor comparta la información. Cada *cluster* resultante contendrá a los puntos que tengan al centro de éste como centroide más cercano. Resulta evidente entonces que se requiera conocer de antemano la cantidad de particiones a realizar en el proceso, por lo que este algoritmo está íntegramente ligado al *hard clustering* mencionado previamente. Dentro de esta categoría de algoritmos, se destaca por su popularidad el *K-means clustering*. Este método encuentra en cada iteración el máximo local. Tiene un funcionamiento en 5 pasos:

1. Se determina la cantidad de *clusters* entre los cuales se deberá clasificar la información de entrada.
2. Se asigna aleatoriamente cada punto a un *cluster*.

3. Se calculan los centroides según la distribución de los puntos en cada categoría.
4. Se reasignan los puntos al centroide más cercano.
5. Se recalculan los centroides, ahora teniendo en cuenta la nueva categorización obtenida en el paso 4.

Luego se repetirán los pasos 4 y 5 iterativamente hasta que no se pueda mejorar el *clustering* o no haya cambios en el cuarto paso en dos iteraciones seguidas. El algoritmo acaba teniendo una eficiencia de O_n . Como el comienzo es aleatorio, al correr el algoritmo de K-Means repetidamente se podrán encontrar resultados ligeramente distintos. Esta "cercanía" de la que se habla consiste en la menor distancia Euclídea. Es decir, la distancia en línea recta en un espacio cartesiano.

Para analizar los resultados del *clustering*, se va a hablar de distancias intracluster e intercluster, las cuáles se pueden ver representadas en la *Figura 4.3.8.2*. La primera refiere a la distancia entre puntos dentro del *cluster*, la cual se busca que sea pequeña para tener un fuerte efecto de agrupamiento. Por el contrario, la distancia intercluster refiere a cuánto distan dos puntos de distintos *clusters* y se va a buscar que sea lo más grande posible.

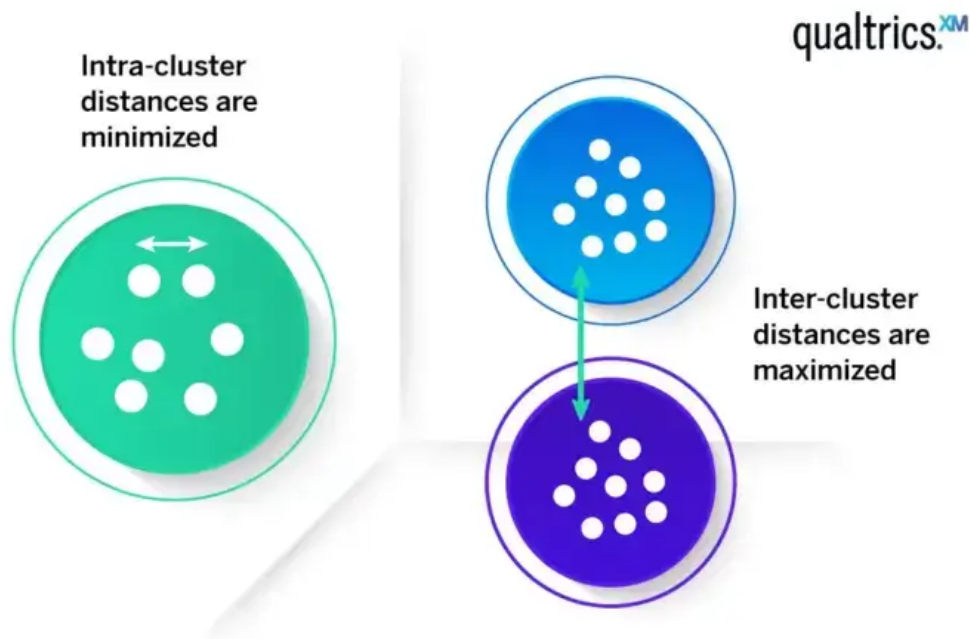


Figura 4.3.8.2: Representación gráfica de la distancia intra e inter cluster

Como ya se explicó, el algoritmo de K-Means requiere indefectiblemente determinar la

cantidad de *clusters* de la red previo a su inicialización. Sería natural entonces preguntarse qué criterios se deben inspeccionar para hallar el valor de K , el número óptimo de divisiones. Para ello, existe un amplio abanico de posibilidades. Éstas tienen fundamentos desde cálculos probabilísticos hasta la iteración del algoritmo de *clustering* y su posterior evaluación de resultados. Nuevamente, se hablará sólo de aquellos métodos efectivamente utilizados en el Trabajo Práctico Profesional.

Elbow Method

El primero de estos métodos es el más popular de todos, denominado método del codo. En éste se plasma en un gráfico cómo varía la suma total de errores cuadráticos (tomados como el cuadrado de la distancia intracluster entre el centroide y el *datapoint*) a medida que se aumenta el valor de K . Para ello, se debe correr el algoritmo de K-Means iterativamente dentro de un rango de valores determinado por el programador. En este método gráfico, se tomará como el valor óptimo aquel punto de las abscisas en el que la función discreta (recuérdese que K solo puede tomar valores pertenecientes al conjunto de los Naturales) parece “aplanarse” la adición de nuevos *clusters* no tiene un impacto tan significativo en la disminución del error cuadrático ya mencionado, también conocido como inercia. El nombre tan llamativo se debe a que la curva resultante suele tener una pendiente decreciente muy elevada para valores bajos de K que luego se torna asintótica con el eje de horizontal. Así, se forma una curva similar a un brazo en donde lo que se debe encontrar es su punto de inflexión, el codo de ese “brazo”. Un ejemplo de lo aquí explicado se muestra en la *Figura 4.3.8.3*, en donde se observa que el valor óptimo es $K = 5$.

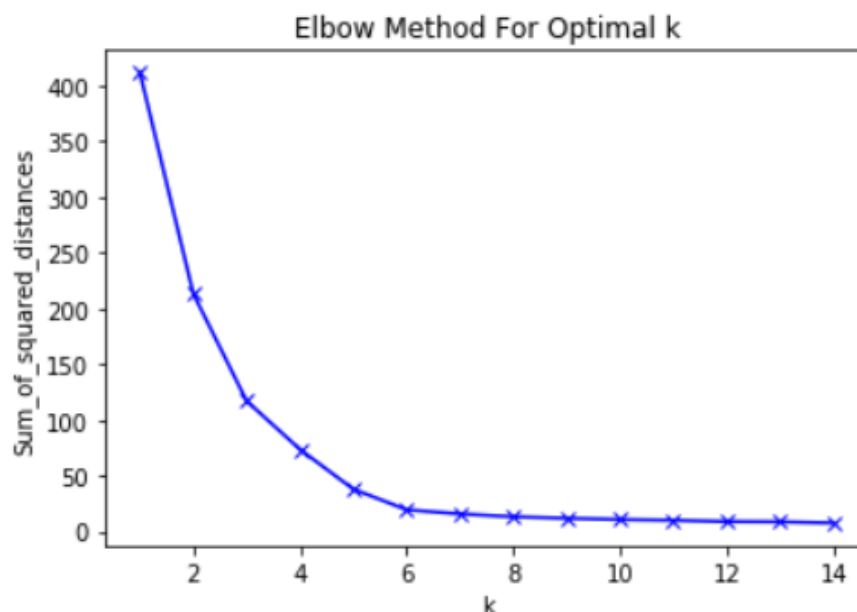


Figura 4.3.8.3: Ejemplo de la gráfica bajo análisis en el método del codo

Es importante mencionar que el método del codo puede presentar algunos problemas. Por un lado, la elección del punto de inflexión puede tornarse un tanto arbitraria y en algunos casos donde el cambio de pendiente no sea tan abrupto se puede estar tomando un valor de K óptimo pero no ideal. Por otro lado, no se debe olvidar que éste es un método iterativo que requiere la recurrente ejecución de K-Means. Esto puede presentar dos problemáticas: en primer lugar, si se tiene un set de datos muy grande, se puede incurrir en un costo computacional considerable; por otro lado, se recuerda que K-Means tiene un componente aleatorio en su paso a paso, por lo que los resultados podrían variar entre la prueba para obtener el valor de K y la aplicación per se del *clustering* en la Red Neuronal.

Shilouette Method

El método de la silueta consiste en obtener el coeficiente de silueta para el *dataset*. Para ello, se debe obtener la silueta de cada punto y luego obtener el promedio de estos valores. Nuevamente, se deberá aplicar el algoritmo de K-Means para distintos valores de K y, a partir de ello se define la silueta de un punto como

$$S_{(i)} = \frac{b_{(i)} - a_{(i)}}{\max(a_{(i)}, b_{(i)})} \quad (4.10)$$

En donde

$b_{(i)}$ = Distancia promedio entre el punto i y el *cluster* más cercano

$a_{(i)}$ = Distancia promedio entre el punto i y todos los otros puntos pertenecientes al mismo *cluster*

En consecuencia, se puede apreciar que $S_{(i)}$ podrá tomar valores comprendidos entre $[-1; 1]$ y va a dar una idea de si los puntos están correctamente asignados a sus *clusters*. En líneas generales, se plantea que

- Si $S_{(i)} \approx -1$, ese punto debería estar asignado a otro *cluster*.
- Si $S_{(i)} \approx 0$, el punto se encuentra aproximadamente a igual distancia de dos *clusters*.
- Si $S_{(i)} \approx 1$, dicho punto esta encasillado en el *cluster* apropiado.

En consecuencia, se buscará aquel valor de K que maximice el coeficiente de silueta para el *dataset*. Se puede lograr esto graficando discretamente los valores de los coeficientes para cada K a probar y buscando el máximo de la función. Un ejemplo de ello se puede ver en la *Figura 4.3.8.4*, en donde se determina que el valor óptimo es $K = 2$.

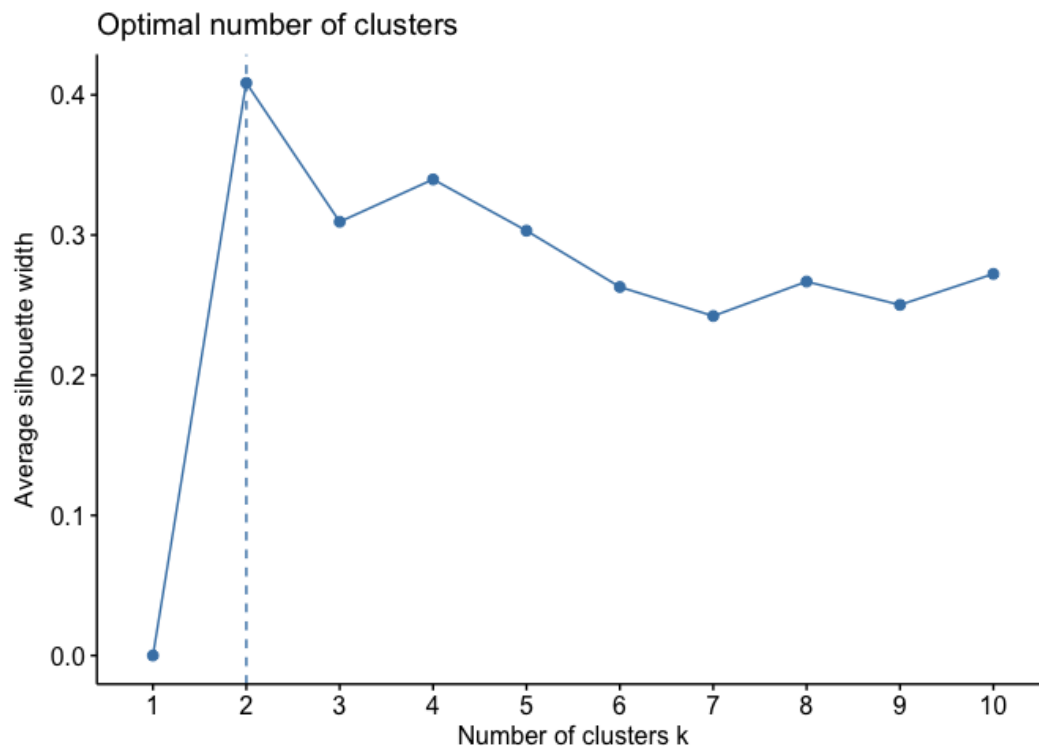


Figura 4.3.8.4: Ejemplo de gráfico de los coeficientes de silueta para cada valor de K a analizar.

Entre las problemáticas que se encuentran para este método, se menciona que el cálculo del coeficiente de silueta demanda una gran exigencia de procesamiento para *datasets* muy grandes. Además, al no haber necesariamente una tendencia decreciente del coeficiente a partir de cierto valor de K, se podría tomar un intervalo de valores que no contengan al K óptimo sino que se encuentre un máximo local dentro de este intervalo. Esto, sumado a que el agrandamiento del intervalo bajo análisis profundizaría esta demanda de cómputo, pueden acabar por dar resultados erróneos que no mejoren el rendimiento de la Red Neuronal.

Calinski-Harabasz Method

El último método que se va a mencionar es el método del índice de Calinski-Harabasz. El mismo se basa en la idea de que los *clusters* idealmente deberían ser muy compactos en sí y muy espaciados respecto de los otros *clusters*. Para obtener el valor del índice, se calcula la varianza de las sumas de los cuadrados de las distancias entre cada punto y su centroide y ésta se la divide por la suma del cuadrado de las distancias entre los centroides. La fórmula resultante para el índice CH se muestra a continuación.

$$CH = \frac{\sum_{i=1}^K \|c_i - c\|^2}{K - 1} \cdot \frac{N - K}{\sum_{i=1}^K \sum_{j=1}^{n_i} \|d_j - c_i\|^2} \quad (4.11)$$

En donde

K = Número de *clusters* con el que se está probando.

c_i = posición del centroide i

n_i = Cantidad de puntos en el centroide i

c = Centroide global

d_j = Punto j del *dataset*

Como se puede observar, nuevamente se está tratando con un método iterativo que requiere de la recurrente ejecución de K-means para distintos valores de K . En consecuencia, podemos tomar un enfoque similar a lo anteriormente planteado y obtener el valor gráficamente. Aquí lo que va a importar es que el índice tome el mayor valor posible, de modo que los *clusters* sean densos y estén bien separados. En caso de haber zonas en donde la curva dada por la unión de los valores discretos obtenidos se torna suave, sin “saltos. erráticos entre cada valor, la elección de un valor u otro de k dentro de ese tramo de la curva no generará cambios significativos. El método, aunque con algunas falencias, permite una forma rápida de estimar el valor óptimo de K y asegurando un criterio muy deseado en el *clustering* como lo es que cada división sea compacta y espaciada del resto.

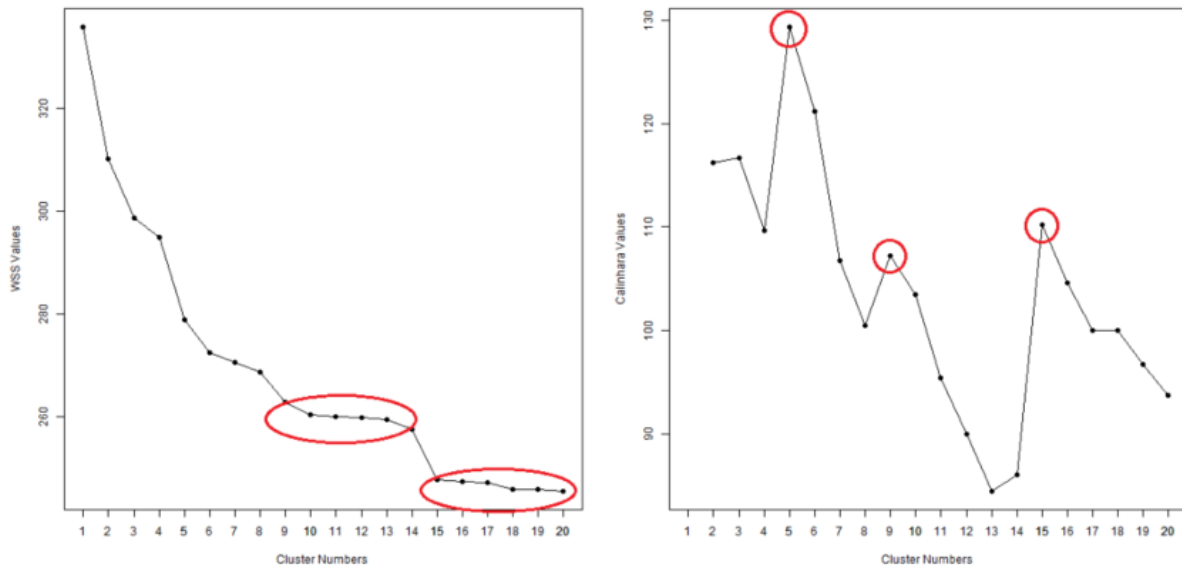


Figura 4.3.8.5: Ejemplos de curvas de índice de Calinski-Harabasz vs Número de *clusters*

Para finalizar, se desea destacar que estos métodos de estimación de K simplemente dan una sugerencia de valores que deberían mejorar las predicciones del modelo. Esto no quiere decir que tomar otro valor de K sea incorrecto, sino que se podría caer en una pérdida de recursos computacionales o, efectivamente el parcial empeoramiento del funcionamiento de la Red Neuronal. En consecuencia, no se debe quedar sólo en el análisis cuantitativo de estos métodos sino que también se debe inspeccionar y probar las distintas alternativas en caso de ser posible. Como ocurre comúnmente en el mundo de la inteligencia artificial, no hay una “receta” fija y repetible sino que es menester conocer los datos con los que se está trabajando y sus particularidades.

5. Desarrollo tecnológico

6. Conclusión

7. Referencias

- Bruinsma, J. (2017). *The resource outlook to 2050: By how much do land, water and crop yields need to increase by 2050?* In FAO Expert Meeting on How to Feed the World in 2050.
- OECD. (2018). *Meat consumption (indicator)*. OECD Data. Retrieved from <https://stats.oecd.org/>
- Gamon, J. A., et al. (1992). *Relationships between NDVI, canopy structure, and photosynthesis in three Californian vegetation types*. Ecological Applications, 2(4), 367-373.
- Paruelo, J. M., et al. (1997). *Estimation of primary production of subantarctic grasslands from AVHRR-NDVI data*. Journal of Vegetation Science, 8(5), 703-710.
- Reeves, M. C., et al. (2014). *Characterizing spatial and temporal patterns of vegetation dynamics with satellite-based NDVI time series in a sub-tropical savanna*. Remote Sensing, 6(7), 6653-6680.
- Popkin, B. M. (2003). *The nutrition transition and obesity in the developing world*. Journal of Nutrition, 133(3), 871S-873S.
- USDA. (2018). *Livestock and Poultry: World Markets and Trade*. United States Department of Agriculture.
- Viglizzo, E. F., et al. (2001). *Agricultura y ganadería*. In El ambiente natural y las actividades humanas en la región pampeana argentina (pp. 299-354). Ediciones INTA.
- Herrero, M., & Thornton, P. K. (2013). *Livestock and global change: Emerging issues for sustainable food systems*. Proceedings of the National Academy of Sciences, 110(52), 20878-20881.
- Irisarri, J. G., et al. (2014). *Impact of changes in the livestock industry on the use of natural resources for livestock production in Uruguay*. Agricultural Systems, 129, 43-53.
- Thornton, P. K., et al. (2009). *Livestock and climate change*. Livestock and Climate Change, 228, 1-12.
- Sala, O. E., & Austin, A. T. (2000). *Methods of estimating aboveground net primary productivity*. In Terrestrial Ecosystems in a Changing World (pp. 25-58). Springer.
- Scurlock, J. M. O., et al. (2002). *Techniques for using plant production and respiration models to estimate regional carbon balance*. In Global Change Biology (Vol. 8, pp. 989-1004). Wiley Online Library.
- Gamon, J. A., et al. (1992). *Remote sensing of the distribution and abundance of shrub and grassland communities in the Jornada Basin*. Journal of Range Management, 45(6),
- Oesterheld, M., et al. (2001). *Grazing management and the diversity of the grass-clove interaction*. Oikos, 92(2), 225-238.

- Wiegand, C. L., et al. (2004). *Predicting forage production of southern African grasslands with a remote sensing model*. Journal of Range Management, 57(4), 369-377.
- Liu, Y., et al. (2019). *Mapping grassland aboveground biomass using active and passive remote sensing at different scales*. Ecological Indicators, 99, 218-228.
- Smith, G. K. (1994). *Neural Networks for Statistical Modeling*. Van Nostrand Reinhold.
- Connor, J. N., et al. (1994). *Recurrent neural networks and robust time series prediction*. IEEE Transactions on Neural Networks, 5(2), 240-254.
- Greff, K., et al. (2016). *LSTM: A search space odyssey*. IEEE Transactions on Neural Networks and Learning Systems, 28(10), 2222-2232.
- Chollet, F., & Allaire, J. J. (2017). *Deep Learning with R*. Manning Publications.
- Hochreiter, S., & Schmidhuber, J. (1997). *Long short-term memory*. Neural Computation, 9(8), 1735-1780.
- Oesterheld, M., Loreti, J., Semmartin, M., & Sala, O. (2001). *Inter-annual variation in primary production of a semi-arid grassland related to previous-year production*. Journal of Vegetation Science, 12(1), 103-110.
- Wiegand, K., & Moloney, K. A. (2004). *Rings, circles, and null-models for point pattern analysis in ecology*. Oikos, 104(2), 209-229.
- Liu, J., Zhu, D., Wang, C., Zhou, Z., & Liu, S. (2019). *Modeling and simulation of primary productivity in ecological systems*. Ecological Modelling, 409, 108750.
- Nuñez-Torres, O. P. (2017). *Los costos de la alimentación en la producción pecuaria*. J.Selva Andina Anim. Sci. [online]. Vol.4, n.2, pp.93-94.
- Texeira, M. & Di Bella, C. & Paruelo, J. (2021). *Sistemas de pronóstico de la productividad forrajera y el balance hídrico para la región de los pastizales del Río de la Plata*.
- Matich, D. J. & Grupo de Investigación Aplicada a la Ingeniería Química (GIAIQ). (2001). *Redes Neuronales: Conceptos Básicos y Aplicaciones*. https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograis/matich-redesneuronales.pdf
- Haykin S. . *Neural Networks. A Comprehensive Foundation*, second edition. Pearson Prentice Hall, 1999.
- Kecman V. . *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*. The MIT Press, 2001.
- Kröse B. and Van der Smagt P. 1996. *An introduction to Neural Networks, eighth edition*. University of Amsterdam.
- Veelenturf L. . *Analysis and Applications of Artificial Neural Networks*. Prentice Hall, 1995.

- Universidad Politécnica de Madrid (Director). (s. f.). *Introducción al aprendizaje automático*. Recuperado 19 de octubre de 2023, de <https://dcain.etsin.upm.es/carlos/bookAA/05.1.RedesNeuronalesIntroduccion.html>
- *IBM documentation*. (s. f.). <https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model>
- Serrano, A. J., Soria, E., & Martín, J. D. (2009). *Redes neuronales artificiales*. Universitat ID Valencia. http://ocw.uv.es/ingenieria-y-arquitectura/1-2/libro_ocw_libro_de_redes.pdf
- Lackes, R., Mack, D., Ziola, J., & Ahern, K. (1998). *Neuronal Networks: Basics and Applications* (CBT Springer, Ed.).
- Sepp Hochreiter and Jürgen Schmidhuber. *Long short-term memory*. *Neural computation*, 9(8):1735–1780, 1997 [Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. Diploma, Technische Universität München, 91(1), 1991]
- Charniak, E. (2018). *Introduction to deep learning*.
- Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press. <http://www.deeplearningbook.org> (2016)
- Arana, C. (2021). *Redes Neuronales Recurrentes: Análisis de los modelos especializados en datos secuenciales* (J. M. Streb, Ed.). <https://ucema.edu.ar/publicaciones/download/documentos/797.pdf>
- Anderson, J.A. & Rosenfeld, E., eds. *Neurocomputing*. MIT Press, Cambridge, MA, 1989
- Basogain Olave, X. (s. f.). *Redes Neuronales artificiales y sus aplicaciones* (Escuela Superior de Ingeniería de Bilbao, Ed.). https://ocw.ehu.eus/pluginfile.php/40137/mod_resource/content/1/redes_neuro/contenidos/pdf/libro-del-curso.pdf
- StatQuest with Josh Starmer. (2020, 19 octubre). *Neural Networks Pt. 2: Backpropagation Main ideas* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=IN2XmBhILt4>
- Kalirane, M. (2023, 5 abril). *Gradient descent vs. backpropagation: What's the difference?* Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2023/01/gradient-descent-vs-backpropagation-whats-the-difference/#:~:text=Gradient%20descent%20is%20used%20to%20find%20a%20weight%20combination%20that,point%20of%20the%20cost%20function.>
- Choudhari, P. (2021, 14 diciembre). *Understanding “convolution” operations in CNN - Analytics Vidhya - Medium*. Medium. <https://medium.com/analytics-vidhya/understanding-convolution-operations-in-cnn-1914045816d4>
- Mandal, M. (2024, 23 febrero). *Introduction to Convolutional Neural Networks (CNN)*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>
- *What are Convolutional Neural Networks?* — IBM. (s. f.). <https://www.ibm.com/topics/convolutional-neural-networks>

- Saha, S. (2023, 8 abril). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. Medium. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- Melanie. (2023, 30 octubre). *Convolutional Neural Network: Everything You Need to Know*. Data Science Courses — DataScientest. <https://datascientest.com/en/convolutional-neural-network-everything-you-need-to-know>
- GfG. (2023, 20 diciembre). *Introduction to Convolution Neural Network*. GeeksforGeeks. <https://www.geeksforgeeks.org/introduction-convolution-neural-network/>
- *What Is a Convolutional Neural Network? — 3 things you need to know*. (s. f.). MATLAB & Simulink. <https://mathworks.com/discovery/convolutional-neural-network.html>
- Kaushik, S. (2024, 6 febrero). *Clustering — Different Methods, and Applications (Updated 2024)*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/#:~:text=Clustering%20is%20the%20task%20of,and%20assign%20them%20into%20clusters.>
- Unir, V. (2021, 8 octubre). *Clustering: qué es y cuál es su uso en Big Data*. UNIR. <https://www.unir.net/ingenieria/revista/clustering/>
- Byrd, J. (2023, 8 noviembre). *Data Clustering: intro, methods, applications*. <https://encord.com/blog/data-clustering-intro-methods-applications/>
- *What Is Cluster Analysis? Overview and examples*. (2023, 18 octubre). Qualtrics. <https://www.qualtrics.com/experience-management/research/cluster-analysis/>
- Saji, B. (2024, 7 enero). *Elbow Method for Finding the Optimal Number of Clusters in K-Means*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/01/in-depth-intuition-of-k-means-clustering-algorithm-in-machine-learning/>
- Baruah, I. D. (2021, 16 diciembre). *Cheat sheet for implementing 7 methods for selecting the optimal number of clusters in Python*. Medium. <https://towardsdatascience.com/cheat-sheet-to-implementing-7-methods-for-selecting-optimal-number-of-clusters-in-python-898241e1d6ad>
- Koli, S. (2023, 2 marzo). *How to Evaluate the Performance of Clustering Algorithms Using Silhouette Coefficient*. Medium. medium.com/@MrBam44/how-to-evaluate-the-performance-of-clustering-algorithms-3ba29cad8c03
- GfG. (2022, 25 abril). *Calinski-Harabasz Index - Cluster Validity indices - Set 3*. GeeksforGeeks. www.geeksforgeeks.org/calinski-harabasz-index-cluster-validity-indices-set-3/
- Wei, H. (2021, 13 diciembre). *How to measure clustering performances when there are no ground truth?* Medium. <https://medium.com/@haataa/how-to-measure-clustering-performances-when-there-are-no-ground-truth-db027e9a871c>
- Alcaraz-Segura, D., Beget, M. E., Blanco, L., Cabello, J., Campos, A., Caride, C., Conde, M. C., de Abelleira, D., Di Bella, C. M., Ferlijiwskyj, I., Garbullsky, M., García,

- A., Milkovic, M., Oesterheld, M., Oyarzabal, M., Oyonarte, C., Paruelo, J., Posse, G., Sznaider, G., & Verón, S. (s. f.). *Percepción remota y sistemas de información geográfica: Sus aplicaciones en Agronomía y Ciencias Ambientales* (J. M. Paruelo, C. M. Di Bella, & M. Milkovic, Eds.).
- Aragon, R. and Oesterheld, M. 2008. *Linking vegetation heterogeneity and functional attributes of temperate grasslands through remote sensing*. Applied Vegetation Science 11:115-128.
 - Oesterheld, M., Di Bella, C., and Herdiles, K. 1998. *Relation between NOAA-AVHRR satellite data and stocking rate of rangelands*. Ecological applications 8:207-212.
 - Oesterheld, M., Loreti, J., Semmartin, M. and Paruelo, J. 1999. *Grazing, fire, and climate effects on primary productivity of grasslands and savannas. Ecosystems of Disturbed Groud*. Edited by L. R. Walker. In: Ecosystems of the World 16:287–306.
 - Costanza, R., d'Arge, R., de Groot, R., Farber, S., Grasso, M., Hannon, B., Limburg, K., Naeem, S., O'Neill, R.V., Paruelo, J., Raskin, R.G., Sutton, P., Van den Belt, M. 1997. *The value of the world's ecosystem services and natural capital*. Nature 357:253-260.
 - Lauenroth, W. 1979. *Grassland primary production: North American grasslands in perspective*. N. French, editor. Perspectives in grassland ecology. Springer-Verlag, New York.
 - Sala, O.E., Parton, W.J., Joyce, L.A. and Lauenroth, W.K. 1988. *Primary production of the central grassland region of the United States*. Ecology 69:40-45.
 - Chapin, F.S., Matson, P.A. and Mooney, H.A. 2002. *Principles of terrestrial ecosystem ecology*. Springer, New York.
 - Lozada, Beatriz I, & Sentelhas, Paulo C. (2003). *Relaciones entre deficiencias y excedentes hídricos estimados a partir de los balances hídricos normal y secuencial*. Bioagro, 15(3), 209-216. Recuperado en 26 de octubre de 2023, de http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316-33612003000300008&lng=es&tlng=es.
 - Carmona Alzate, J. H. & Universidad de Manizales. (2018). Modelo de un balance hídrico y zonificación climática utilizando herramientas SIG.
 - De Goicoechea, N. (1969). *Nociones generales del balance hídrico*.