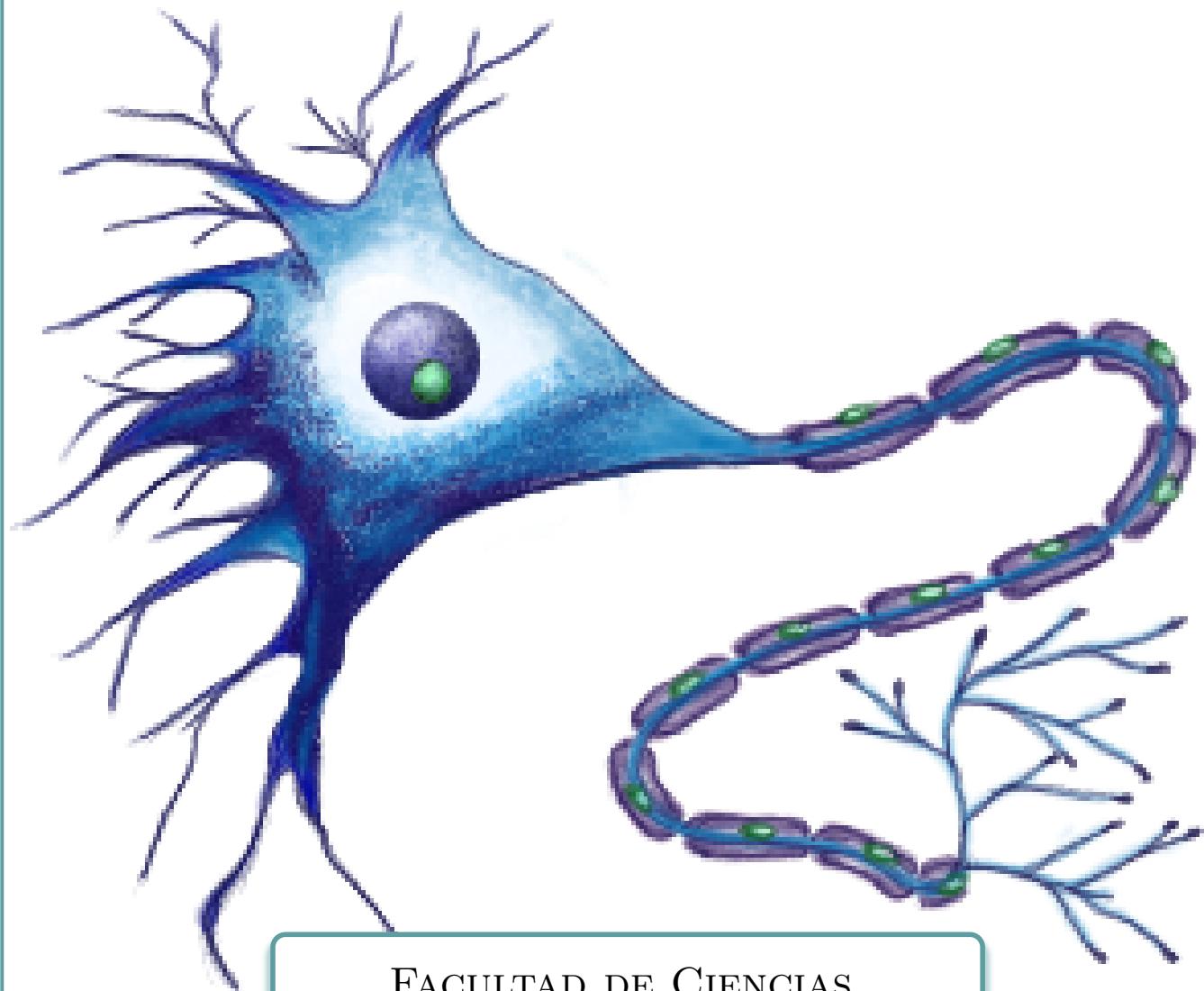


Redes Neuronales

Notas de clase

Karla Fernanda Jiménez Gutiérrez
Verónica Esther Arriola Ríos



FACULTAD DE CIENCIAS,
UNAM

Índice general

Índice general

| | |
|--|-----------|
| I Antecedentes | 2 |
| 1 Neurona biológica | 3 |
| 1.1 Neurociencias computacionales | 3 |
| 1.2 Sistema Nervioso | 5 |
| 1.2.1 Cerebro | 8 |
| 1.2.2 Zonas funcionales | 10 |
| 1.3 Neurona biológica | 11 |
| 1.3.1 La neurona | 11 |
| 1.3.2 Elementos de las neuronas y tipos | 13 |
| 1.3.3 Sinapsis | 15 |
| 1.3.4 Campos receptivos | 19 |
| 1.3.5 Señal eléctrica | 21 |
| 2 Modelo de Hodgkin-Huxley | 24 |
| 2.1 Introducción | 24 |
| 2.2 Membrana y canal | 26 |
| 2.3 Potenciales de Nerst o de reposo | 29 |
| 2.4 Modelo de la membrana como bicapa de lípidos | 29 |
| 2.4.1 Las conductancias iónicas | 33 |
| 2.5 Modelo de las compuertas iónicas controladas por voltaje | 35 |
| 2.6 Dinámica del voltaje durante un disparo | 38 |
| 2.7 Simulación usando el método de Euler | 42 |
| 2.8 Información condificada en las dendritas | 44 |
| 3 Aprendizaje de máquina | 48 |
| 3.1 Introducción | 48 |
| 3.2 Espacio de Hipótesis | 50 |
| 3.3 Clasificación de los conjuntos de datos | 51 |
| 3.3.1 Tipos de aprendizaje | 52 |

ÍNDICE GENERAL

| | |
|---|-----------|
| II Redes dirigidas acíclicas | 54 |
| 4 Perceptrón simple | 55 |
| 4.1 Perceptrón | 55 |
| 4.2 Compuertas lógicas con neuronas | 58 |
| 4.3 Funciones de activación | 60 |
| 4.4 Funciones de error | 60 |
| 4.5 Medidas de rendimiento | 63 |
| 5 Perceptrón multicapa | 69 |
| 5.1 Intro | 69 |
| 5.2 XOR | 69 |
| 5.3 Propagación hacia adelante manual | 71 |
| 5.4 Propagación hacia adelante vectorizada (con matrices) | 72 |
| 5.5 Interpretación matemática del mapeo no lineal | 74 |
| 5.6 Propagación hacia adelante multicapa | 75 |
| 6 Entrenamiento por retropropagación | 77 |
| 6.1 Función de error | 77 |
| 6.2 Gradiente de la función de error | 77 |
| 6.3 Descenso por el gradiente | 77 |
| 6.4 Otros algoritmos de optimización | 77 |
| 7 Optimización del entrenamiento | 78 |
| 7.1 Problemas en redes profundas | 78 |
| 7.2 Gradiente desvaneciente (o que explota) | 78 |
| 7.3 Entrenamiento en línea vs en lotes | 78 |
| 7.4 Normalización y normalización por lotes | 78 |
| 7.5 Regularización | 78 |
| 8 Caso de análisis e interpretación | 79 |
| 8.1 Red Hinton árbol familiar con numpy (entrenamiento) | 79 |
| 8.2 Red Hinton árbol familiar con pytorch | 79 |
| 9 Entrenamiento con genéticos | 80 |
| 9.1 Algoritmos genéticos | 80 |
| 9.2 Neuroevolución | 80 |
| 9.2.1 Antecedentes: Aprendizaje por refuerzo en videojuegos | 80 |
| 9.2.2 Arquitectura para estimar la función de recompensa | 80 |
| 9.2.3 Entrenamiento | 80 |
| 10 Mapeos autoorganizados | 81 |
| 10.1 Aprendizaje no supervisado | 81 |
| 10.2 Mapeos auto-organizados | 81 |
| 10.3 Kohonen | 81 |

| | |
|---|--------|
| 11 Redes Neuronales Convolucionales | 82 |
| 11.1 Convolución | 82 |
| 11.2 Redes Convolucionales | 82 |
| 11.3 Softmax | 82 |
| 11.4 MNIST | 82 |
| III Redes con ciclos | 83 |
| 12 Redes Neuronales Recurrentes | 84 |
| 12.1 Derivadas ordenadas | 84 |
| 12.2 Retropropagación en el tiempo | 84 |
| 12.3 Sistemas dinámicos y despliegue del grafo | 84 |
| 12.4 Arquitectura recurrente universal | 84 |
| 12.5 Función de error | 84 |
| 12.6 Forzamiento del profesor | 84 |
| 13 Atención | 85 |
| 14 LSTM | 86 |
| 15 GRU | 87 |
| 16 Casos de análisis: etiquetado de palabras y conjugación de verbos | 88 |
| IV Redes no dirigidas | 89 |
| 17 Redes de hopfield | 90 |
| 17.1 Entrenamiento | 90 |
| 18 Máquinas de Boltzman | 91 |
| 18.1 Entrenamiento | 91 |
| 18.1.1 Partículas y partículas de fantasía | 91 |
| 18.1.2 Máquinas de Boltzman Restringidas | 91 |
| 19 Redes adversarias | 92 |
| 19.1 GANs | 92 |
| A Ecuaciones diferenciales | 93 |

Etc

A lo largo del texto se utilizará la siguiente notación para diversos elementos:

| | |
|-----------|----|
| Conjuntos | C |
| Vectores | x |
| Matrices | M |
| Unidades | cm |

Parte I

Antecedentes

1 | Neurona biológica

Neurociencias computacionales

Las redes neuronales surgieron completamente inspiradas en los sistemas biológicos. Lo que estamos haciendo los computólogos es tomar una idea de la naturaleza, una idea que ha probado ser sumamente efectiva para procesar información y que logra resolver problemas que nosotros aún no sabemos solucionar con modelos diseñados explícitamente. Los más notorios son:

- Problemas de visión por computadora.
- Procesamiento del lenguaje natural.

A lo largo del texto obtendremos una somera idea de qué hace el sistema nervioso de un ser humano, tomaremos también ejemplos de animales como el calamar gigante y cangrejos; ejemplos que han permitido estudiar biológicamente cómo funcionan las neuronas y cómo funciona su sistema nervioso.

Entonces por un momento pensemos en el sistema nervioso como un todo, lo que realmente está pasando al computar no es el cálculo del proceso de una sola neurona sino de la colección de todas ellas. Lo que sucede con los sistemas biológicos es que son muchísimo más complicados que lo que vamos a ver nosotros como modelos computacionales, sin embargo muchísimas empresas están utilizando estas técnicas. El sistema nervioso como un todo es bastante más complejo, pero conforme han ido evolucionando las redes neuronales computacionales, ya con sus arquitecturas y organizaciones, se están volviendo también más complejas. Varias de las estructuras más exitosas tienen un análogo muy fuerte con un sistema nervioso natural.

Veamos un campo conocido como **neurociencias computacionales** el cual se dedica explícitamente al estudio/modelado de los sistemas biológicos pero ya conjuntando varios campos. Se interesan notablemente en: descripciones y modelos funcionales biológicamente realistas de neuronas y sistemas neuronales. En contraposición, los modelos que veremos en redes neuronales computacionales no necesariamente tienen que ser realistas, lo que nos interesa es que resuelvan los problemas, si se desvían un poco de cómo funcionan los sistemas naturales en un principio no es problema.

1. Neurona biológica

Ahora, ¿qué le interesa modelar a las neurociencias computacionales? Se fijan en la fisiología y en la dinámica de estos sistemas, combinando varias ciencias tales como:

- **Biofísica** por el estudio de las propiedades físicas detrás de los sistemas biológicos.
- **Neurociencias tradicionales** con modelos matemáticos.
- **Ciencias de la computación** tanto en la parte del modelado como en la parte de la implementación de estos modelos y la generación de simulaciones computacionales.
- **Ingeniería eléctrica** donde se está diseñando hardware especializado para ejecutar modelos de manera eficiente, algunos de los modelos matemáticos están basados en circuitos eléctricos.
- **Ciencias cognitivas** que tratan de ver qué se está codificando dentro de un sistema nervioso y cómo podemos interpretar esa información que está ahí guardada.

Vamos a ver cómo están influyendo todos estos antecedentes en lo que van a hacer las ciencias de la computación pero con su propio modelo de redes neuronales, pues existe una conexión muy fuerte entre estos dos campos.

Las neurociencias computacionales, como se mencionó anteriormente, estudian modelos del sistema nervioso y clasifican estos modelos en tres tipos:

1. **Modelos descriptivos**, nos limitamos a decir qué está haciendo un sistema; en particular aquí son muy famosos los experimentos con ratones, se está tratando de ver qué puede hacer, que no puede hacer, que puede aprender, que no, pero no se puede explicar “¿cómo?”, simplemente se dice qué es lo que está sucediendo.
2. **Modelos mecanistas**, donde ahora sí nos interesa saber, ¿cómo es que están haciendo las cosas? Aquí vamos a ver cómo los modelos matemáticos precisamente nos están tratando de describir cómo puede ser que se están conectando estas neuronas, cómo pueden estar funcionando las redes de neuronas, cómo podría estarse almacenando la información y transfiriendo de un lado a otro.
3. **Modelos interpretativos**, nos dan una idea del por qué o para qué lo hacen. Se tiene que buscar intencionalidad, razonamiento de más alto nivel.

Cuando trabajemos con redes neuronales computacionales vamos a notar que sí necesitamos adentrarnos un poco en los tipos 2 y 3. Para romper esa traba con nuestras redes neuronales, donde sabemos que aprendieron, pero no estamos ni siquiera seguros de qué aprendieron o porqué lo aprendieron así, vamos a tener que utilizar herramientas matemáticas para tratar de descubrir qué es lo que realmente está haciendo la red entrenada.

Ahora revisemos los **objetivos del modelado** en neurociencias:

(Empezando desde lo más granular que es cada una de las neuronas)

- Las **corrientes** que están pasando a través de las membranas de las neuronas, la influencia que tienen en el paso de la información.
- Las **proteínas** van a jugar un papel importante en la conducción de elementos iónicos no transmisores (acoplamientos químicos).

(El siguiente nivel con combinaciones de varias neuronas)

- Las **oscilaciones de las redes** completas, qué pasa con estas señales, pulsos eléctricos que se están transfiriendo de unas regiones a otras y que empiezan a producir oscilaciones con ciertos períodos, regiones de actividad o regiones que se apagan.
- **Arquitectura topográfica y de columnas** cómo están organizadas estas neuronas, quiénes están conectadas con quiénes, cómo reaccionan dentro de ciertas regiones identificadas, cómo interactúan con otras regiones. Se puede identificar una arquitectura tanto desde el punto de vista fisiológico como del punto de vista funcional. Un caso particular de estas estructuras es la formación de columnas de neuronas que están altamente conectadas y trabajan como una unidad.
- El **aprendizaje**, es decir estamos procesando información, guardando información, recuperándola y eso permite que los seres que cuentan con un sistema nervioso tengan características especiales cuyo comportamiento se puede modificar conforme aprenden.
- La **memoria**, necesitamos almacenar información y recuperarla para procesarla.

Sistema Nervioso

¿Qué es un nervio? Un nervio es una gran colección de axones que están viajando todos juntos en una especie de cable (fibra), pasan vasos sanguíneos por en medio de los nervios. Esto es de lo que está formando el sistema nervioso, se originan desde la médula espinal (31 pares de nervios raquídeos) o encéfalo (12 pares de nervios craneales).

Los nervios son estructuras conductoras de impulsos nerviosos situados fuera del sistema nervioso central, es decir, estamos hablando de todos estos axones que salen desde del cráneo, la médula espinal y están descubriendo el resto del cuerpo. Están formados por un conjunto de axones agrupados cada uno de los cuales procede de una neurona. Pueden ser clasificados como:

- **Motores** salidas, ejecución/acción
- **Sensitivos** entradas

1. Neurona biológica

- **Mixtos** son mayoría, tienen tanto fibras sensitivas como motoras

Tenemos dos grandes partes del sistema nervioso, el **sistema nervioso periférico** y el **sistema nervioso central**, como se puede ver en la imagen 1.1.

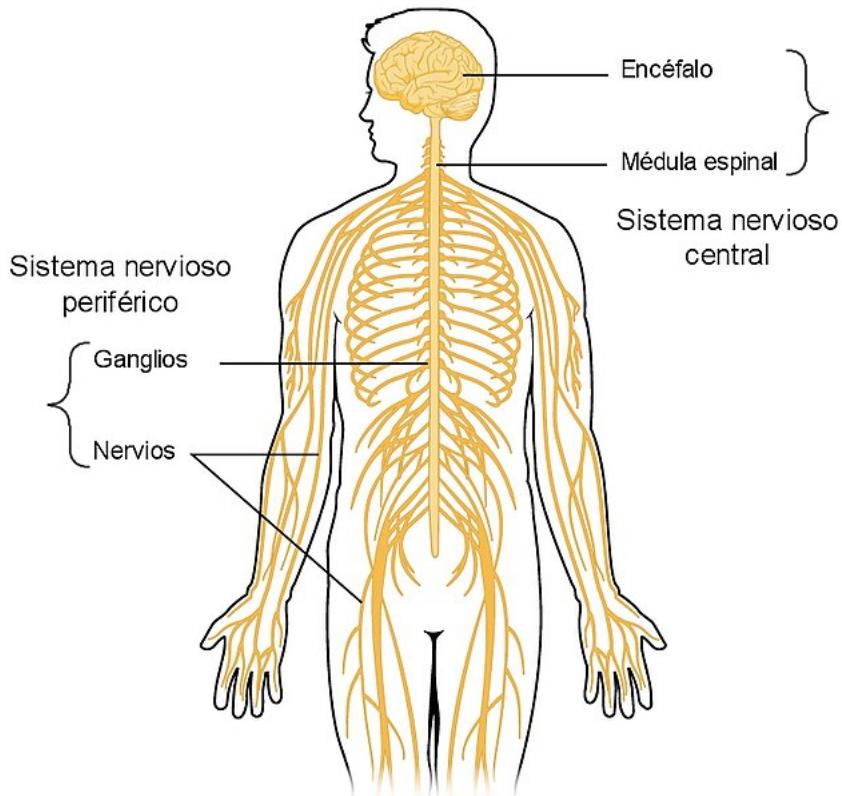


Figura 1.1 Overview of Nervous System esp, OpenStax, 20 diciembre 2018, WIKIMEDIA COMMONS, https://upload.wikimedia.org/wikipedia/commons/0/07/1201_Overview_of_Nervous_System_esp.jpg, CC BY-SA 4.0

En del sistema nervioso periférico tenemos al:

- **Sistema somático** se controla de forma voluntaria, se conforma de nervios conectados a músculos voluntarios esqueléticos y receptores sensoriales, de los cuales unos son:
 - * de entrada, **aferentes**
 - * de salida, **eferentes**
- **Sistema autónomo** funciona de forma involuntaria, se conforma de nervios que se conectan con el corazón, los vasos sanguíneos, los pulmones, el estómago, los intestinos, glándulas

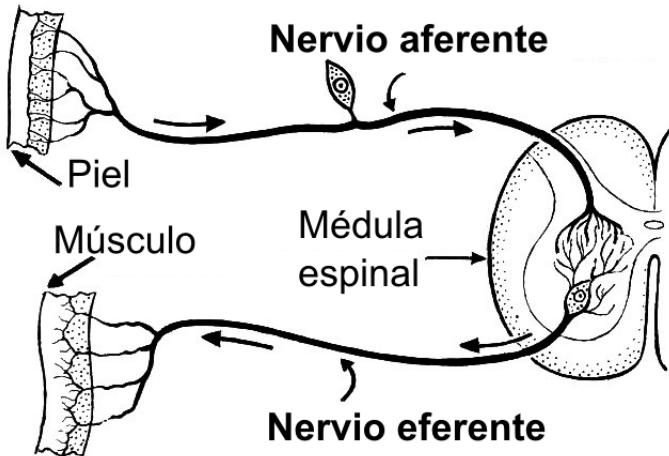


Figura 1.2 Diagrama explicativo del recorrido eferente y el aferente, Pearson Scott Foresman, 26 August 2010, WIKIMEDIA COMMONS, https://upload.wikimedia.org/wikipedia/commons/3/3e/Afferent_%28PSF%29.es.png, CC0

Ahora respecto al sistema nervioso central lo integra:

- La médula espinal
 - * Dentro de esta hay una organización, la presencia de **ciclos de retroalimentación local**, es decir, nuestro sistema va a estar en diferentes etapas son nervios que no necesitan pasar por todo el procesamiento cerebral, las señales simplemente entran llegan a una fase local e inmediatamente reaccionan, ver el ejemplo de la imagen 1.3. Ocurren en un ciclo local y esto también puede convertirse en algo muy importante a la hora de hacer cálculos, no siempre es necesario pasar todo por todas las capas de procesamiento.
 - * **Señales de control motor descendientes** del cerebro hacia las neuronas motoras, estas son señales que provienen de un campo en una capa mucho más alta de procesamiento y provocan movimientos.
 - * **Axones sensoriales ascendentes** donde el cuerpo de la neurona está afuera y la información va a viajar hacia arriba, desde los músculos, piel y estas señales viajan hasta el cerebro.
- El encéfalo

Cada colección de nervios que sale de la base del cerebro se asocian con funciones muy específicas (en su mayoría).

Notas:

- Este sistema está hecho en diferentes niveles locales, entradas y salidas

1. Neurona biológica

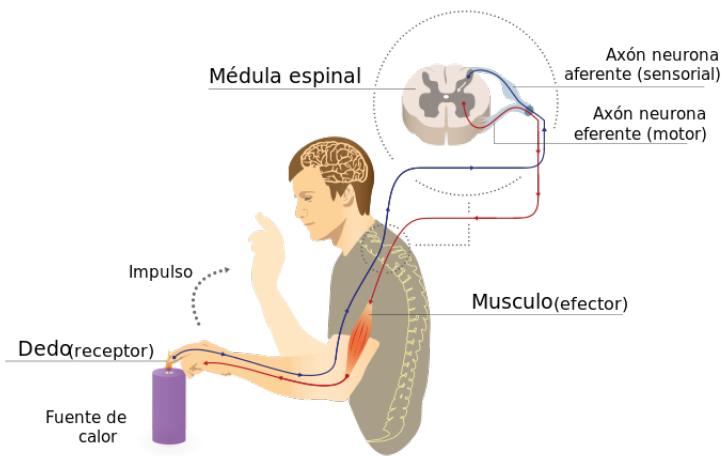


Figura 1.3 Esquema explicativo del arco reflejo, Marta Aguayo, 18 diciembre 2014, WIKIMEDIA COMMONS, https://upload.wikimedia.org/wikipedia/commons/c/cb/Imgnotraçat_arco_reflexo_esp.svg, CC BY-SA 3.0.

- El procesamiento que esté ocurriendo en el encéfalo puede tener diferentes capas y eso se verá reflejado cuando nosotros definamos arquitecturas para las redes neuronales.
- Las redes neuronales actuales, que han tenido más éxito, se componen de diferentes subunidades o diferentes redes que hacen cosas locales. Es decir esta estructura global que estamos viendo, se está empezando a reproducir/imitar ya con las neuronas computacionales.

Cerebro

En esta parte vamos a preocuparnos sobre todo por la parte funcional. Haciendo una breve analogía, vamos a hacer una visión general del “hardware”, para ver qué efectos va a tener en el “software”. En general la arquitectura de cada cerebro es completamente diferente al cerebro de otras personas. Se ha intentado averiguar qué está haciendo cada región con diferentes estudios por ejemplo, ver cuánta sangre se está bombeando en diferentes regiones del cerebro dependiendo de los estímulos que se le presentan a una persona, o si alguna persona tiene un padecimiento se tratan de tomar escaneos para ver qué regiones del cerebro están funcionando y cuáles presentan lesiones. A partir de las lesiones, lo que hacen es que una vez que está identificada la actividad que ya no se puede realizar de forma normal, averiguan qué región era responsable de esa actividad, que ahora está dañada.

Gracias a esos estudios, se ha logrado identificar más o menos en manera general, a qué se dedica cada una de las regiones del cerebro. En ocasiones no se puede decir exactamente qué tan vinculadas están (las regiones) o por qué se están activando otras regiones.

Hay partes funcionales que se comparten entre las diferentes regiones y no están ubicadas en un solo lugar. Otra parte importante a mencionar es, el cerebelo que se considera prácticamente vital, cumple con funciones tales como el equilibrio, la coordinación, el control fino de los músculos, de hecho tiene más neuronas que el cerebro y aun así hay niños que nacen y viven sin cerebelo.

A continuación se mencionan algunas de las diferentes funciones de las regiones, que se han identificado en la imagen 1.4:

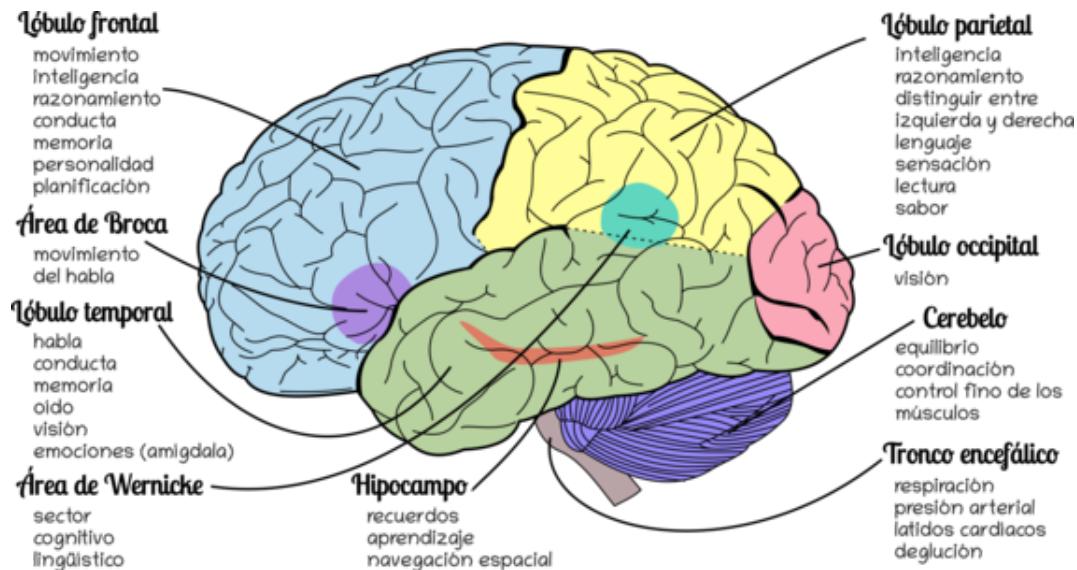


Figura 1.4 Diagrama básico de las regiones del cerebro.

Lóbulo frontal se le puede asociar con la parte del raciocinio, la parte de inteligencia, la conducta, la memoria, la personalidad, la capacidad para realizar planes complejos a largo plazo y también es responsable de algunas actividades de movimiento. Dentro de este destaca el área de broca, su principal función es el movimiento del habla, mover los labios, la boca.

Lóbulo temporal aquí está otra parte del habla, que tiene que ver más con el uso de símbolos para el lenguaje, la conducta, memoria, aquí se procesa el oído, un poco de visión y emociones. Dentro de este está (compartida entre el lóbulo parietal) el área de Wernicke, trabaja con la parte lingüística, y de cognición. También dentro de este está el **hipocampo** trabaja con recuerdos, aprendizaje y navegación espacial, cómo sabemos cómo llegar de un lado hacia otro.

Lóbulo parietal trabaja con la inteligencia, razonamiento, distinguir entre izquierda y derecha, lenguaje, sensación, lectura y sabor.

1. Neurona biológica

Lóbulo occipital se dedica prácticamente solamente a visión, es una región un tanto amplia. En particular en el área de robótica cuando están programando un robot o móvil, los robots tienen dos laptops y una de ellas se dedica prácticamente solo a procesar la visión.

Cerebelo se encarga del equilibrio, la coordinación fina de los músculos.

Tronco encefálico se encarga de la respiración, presión arterial, latidos cardíacos, de ilusión, conciencia.

Zonas funcionales

Para visualizar mejor la parte de la arquitectura, que tiene el cerebro para realizar todo lo que se le conoce como, la ruta desde la sensación hasta la cognición, veremos un diagrama de la parte funcional del cerebro.

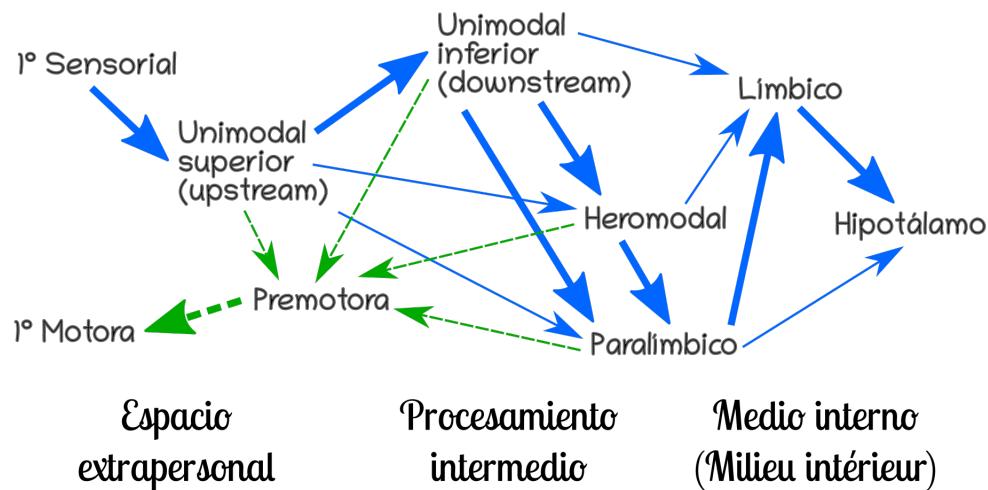


Figura 1.5 Diagrama de la arquitectura del cerebro a nivel funcional.

Explicando el diagrama 1.5, en la primera parte (espacio extrapersonal) vamos a pensar en la entrada sensorial, que se enfoca muchísimo en la parte de visión y audio (en general todos los sentidos), notamos qué de las neuronas que están en la parte sensorial, su primera conexión es hacia una capa que se le llama unimodal superior, aquí se procesa la información de cada sentido de manera individual, es decir, las neuronas o solamente están procesando visión o solamente audio, todavía no se mezclan, por ejemplo de visión, se separan colores e intensidad lumínica, se empieza a detectar algunas esquinas, alguna inclinación, la dirección de las luces y las sombras. Notemos que desde aquí hay una rápida conexión a la sección premotora y luego hacia la parte motora, recordando la mención de los circuitos locales y de reflejos, aquí prácticamente lo podemos ver (en este pequeño camino).

Pasando de este primer procesamiento básico entramos al siguiente que es el unimodal inferior, (aquí aún se está trabajando con procesamiento de una sola modalidad) visión sigue siendo visión, audio sigue siendo audio, pero ya son procesamientos un poco más complejos, por ejemplo, reconocimiento de rostros, de objetos. En esta parte tenemos un rápido ciclo de regreso a la parte premotora, por ejemplo la acción de ver a mi mamá y saludarla (aquí aún no se tiene que razonar demasiado).

En la siguiente fase (medio interno), se conecta hacia tres áreas, la **heromodal**, ya se integran diferentes modalidades (audio y visión) ejemplo, oigo que me hablan y volteo a ver, aquí se está juntando ambas cosas, el **límbico** y el **paralímbico** que trabajan con la parte de las emociones y conceptos abstractos.

Finalmente, llegamos al **hipotálamo** que es donde están todas las emociones, en las conexiones entre estas regiones, estarían los procesamientos de alto nivel.

Ahora estas diferentes regiones se replican de cierta manera cuando estamos haciendo los diseños de las arquitecturas modernas para redes neuronales. En algunas ocasiones se comienza con algunas capas de neuronas, haciendo procesamientos con una sola modalidad, extrayendo datos básicos, después se van componiendo en figuras más complejas y después hasta podemos combinar bloques de neuronas, para poder resolver problemas que tomen en cuenta diferentes modelos.

Neurona biológica

La neurona

La neurona es un tipo de célula perteneciente al sistema nervioso central, que se comunica tanto por señales eléctricas como por señales químicas. Cada neurona tiene:

- Un cuerpo celular (**soma**) que contiene un núcleo y otros componentes celulares.
- Una zona de recepción denominada **dendritas**.
- Una zona de emisión conocida como **axón**, compuesto de:
 - ★ Cono axonico.
 - ★ Membrana plasmática axonica y citoplasma.
 - ★ Recubrimientos de mielina, interrumpidos con intervalos regulares de nódulos (anillos) de Ranzier.
 - ★ Terminales del axón donde se encuentran los **botones sinápticos**.

Pensemos en la neurona como toda una compuerta, por un lado, está el cuerpo de una neurona típica, en las dendritas tenemos una mezcla de neurotransmisores y iones que

1. Neurona biológica

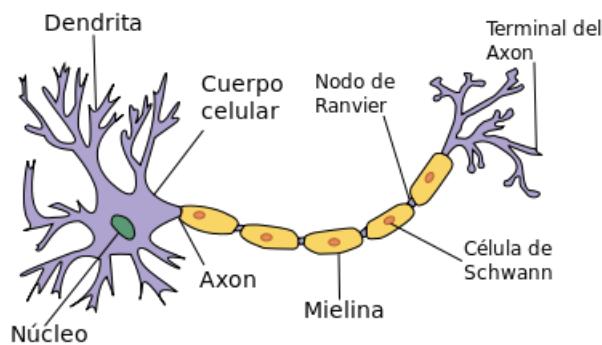


Figura 1.6 Neurona, Acracia, 14 January 2007, Wikimedia Commons, <https://commons.wikimedia.org/wiki/File:Neurona.svg>, Creative Commons Attribution-ShareAlike 2.5 Generic

pueden moverse a través de la membrana. La forma en que intercambia información es mediante sustancias químicas y iones que se están intercambiando, entre la parte de afuera y de adentro de la neurona. Particularmente en las dendritas, se tienen terminaciones que se pueden conectar con otras neuronas y de esta manera permitir el paso de información.

- Neurona presináptica, transmite una señal.
- Neurona posináptica, recibe una señal.

En el interior de la neurona hay una cierta carga eléctrica, en el exterior (el líquido de afuera) hay otra carga eléctrica, es decir, hay una **diferencia de potencial** entre el interior y el exterior de la neurona, por eso se dice que la membrana axónica en sí misma tiene una carga eléctrica. Dado que es porosa, esta membrana va a estar intercambiando partículas con el exterior, esto va a hacer que la polarización de esta membrana vaya cambiando, si en algún momento la diferencia de potencial neta rebasa un cierto umbral.

Transmisión de señales y almacenamiento de información:

1. La neurona desde sus dendritas recibe señales de otras neuronas vecinas.
2. Cada señal se va acumulando en su cuerpo hasta el cono axónico, donde se van a estar sumando la contribución de todos los efectos de cambios de potencial.
3. En el momento que se rebase un cierto valor umbral, la diferencia de potencial se propaga hasta los botones terminales.
4. La neurona entra en un período refractario, donde empieza a cambiar el potencial entre el cono axónico y el axón de la neurona.
5. Se va a transmitir un disparo eléctrico en seguida,

6. La neurona se va a quedar totalmente quieta, durante un breve momento para que la señal pueda viajar hacia el axón.
7. Se va a notar un cambio muy violento en el voltaje, que se va recorriendo a lo largo de todo el axón.

La neurona típica tiene unas células de mielina, que forman nodos que van cubriendo al axón para evitar que se pierda la señal, estos nodos recargan otra vez la señal y permite que avance, al siguiente nodo, donde se recarga nuevamente y avance, hasta que logre llegar al final de axón.

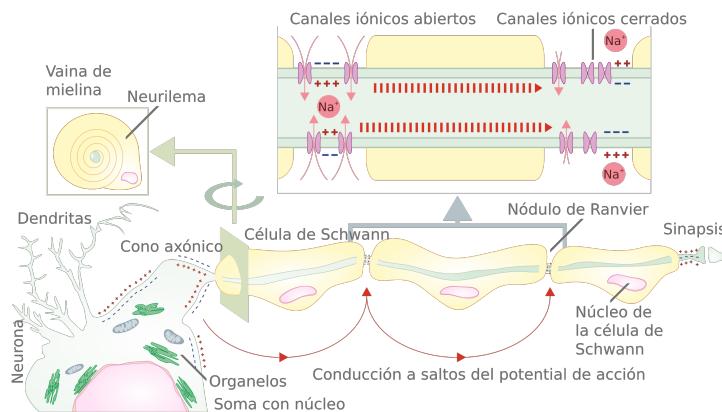


Figura 1.7 Corriente iónica por el axón (efecto de corto plazo), Helixitta, 1 octubre 2015, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Propagation_of_action_potential_along_myelinated_nerve_fiber_en.svg, Creative Commons Attribution-ShareAlike 4.0 International

Este trayecto puede ser de una neurona a unas pocas neuronas vecinas, hasta unos cuantos metros (ej. esta podría estar en la médula espinal y el axón llegar hasta el dedo). Cuando la señal llega a la terminal de el axón, hay varias terminales que van a reaccionar ante el cambio de electricidad, mediante la liberación de unas vesículas, que contienen **neurotransmisores**.

Elementos de las neuronas y tipos

Elementos durante la transmisión de señales:

- **Neurotransmisores:** son los mensajeros químicos que se comunican entre neuronas adyacentes; La liberación de neurotransmisores de una neurona ayudará a despolarizar o hiperpolarizar (aumentar la magnitud de la carga) la neurona adyacente, lo que hará que sea más o menos probable que ocurra un potencial de acción en la siguiente neurona.

1. Neurona biológica

- **Impulsos eléctricos:** potenciales de acción que son, cambios de voltaje que van a ir ocurriendo a lo largo del axón. Sucede una vez que se acumularon demasiadas señales a través de las dendritas, entonces la neurona puede disparar un impulso eléctrico, a través del axón, que va a provocar que su terminal libere más químicos, estos químicos son los que hacen los efectos pequeños en cada uno de los cuerpos de las neuronas posinápticas.
- **Plasticidad:** modificación a largo plazo de las conexiones entre neuronas. En el cerebro las neuronas pueden cambiar de manera permanente, perder canales (que permiten el intercambio de nuevos transmisores sin impulsos eléctricos), formar más canales o incluso pueden crear protuberancias. Por ejemplo, cuando un cerebro aprende está transformando su arquitectura, es decir, los aprendizajes de largo plazo, modifican el cerebro y en consecuencia va a pensar y reaccionar distinto, que antes del aprendizaje.

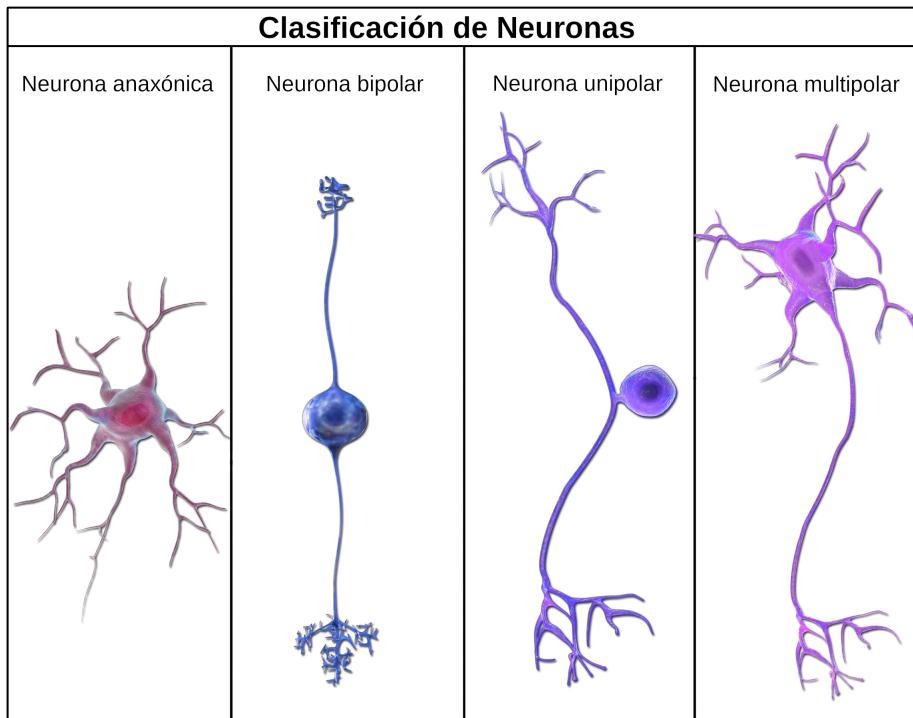
Clasificación de tipos de neuronas:

Figura 1.8 Representación de la clasificación de neuronas, BruceBlaus, 26 junio 2017, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Neuron_Classification.png, Creative Commons Attribution-ShareAlike 4.0 International

- **Neuronas sin axones**, nunca dispara, pero si tiene intercambios de neurotransmisores en las dendritas

- **Neuronas bipolares**, tienen dos axones.
- **Neurona unipolar**, solamente hay una conexión entre el cuerpo y el axón, pero el axón tiene dos ramas, cuando dispara va a disparar hacia los dos lados, haciendo llegar su señal a diferentes regiones.
- **Neurona multipolar**, la más conocida, empieza con un cuerpo con dendritas y luego un largo axón que va a terminar con varias terminaciones axónicas.

Cuando modelamos redes neuronales lo típico es modelar, una neurona con dendritas, su disparo y su axón, que se conecta con las siguientes dendritas, pero aquí ya estamos viendo que la naturaleza nos dice que hay que pensar más y plantear cómo hacer las representaciones de estas conexiones que nos presenta la naturaleza, un poco diferente pero tal vez con resultados más satisfactorios.

Sinapsis

Aquí veremos más a detalle cómo una neurona recibe o transmite información a otras neuronas, donde para un solo disparo están participando un montón de elementos que veremos más adelante.

El momento en que dos neuronas transmiten información se llama **sinapsis** y es mediante conexiones que se dan en las terminales del axón (vesículas sinápticas) de la neurona presináptica hacia la postsináptica. Es importante notar que estas neuronas no tienen contacto anatómico, sino que están separadas por un espacio muy pequeño, **la brecha sináptica**. Lo que sucede en estas conexiones es un intercambio electroquímico que produce cambios de polaridad a lo largo la membrana.

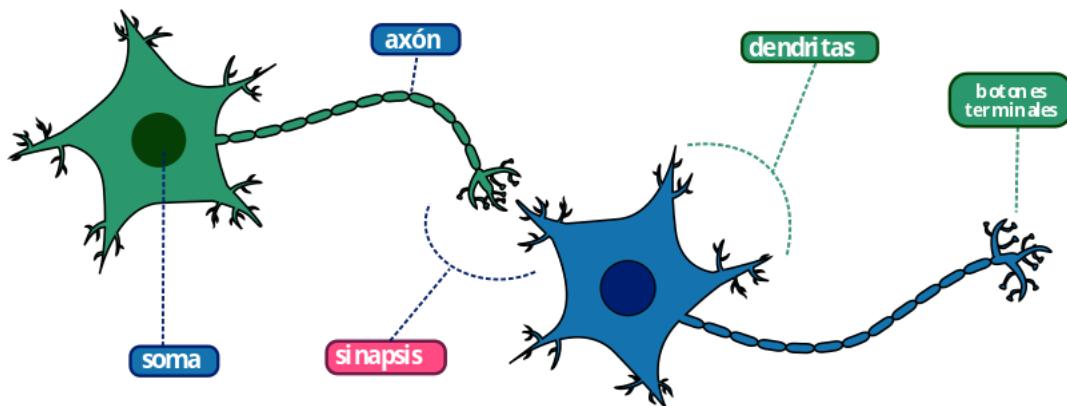


Figura 1.9 Part of neurons in Spanish, Dana Scarinci Zabaleta, 24 febrero 2019, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Part_of_neurons_in_Spanish.svg, OpenStax, CC0

Clasificación de sinapsis, las terminales del axón de la neurona presináptica puede hacer contacto con la neurona postsináptica en:

1. Neurona biológica

1. su dendritas, axodendrítica.
2. su cuerpo (soma), axosomática.
3. su axón, axoaxónica.

Distingamos entre dos tipos de sinapsis:

Sinapsis eléctrica: las membranas de las células pre y posinápticas se unen en la brecha sináptica por una unión tipo gap, o unión comunicante, que son pequeños canales que permiten el paso de iones.

1. Posee una transmisión bidireccional de los potenciales de acción.
2. Sincronización en la actividad neuronal, lo cual hace posible una acción coordinada.
3. Los potenciales de acción pasan a través del canal proteico directamente sin necesidad de la liberación de los neurotransmisores, por tanto, es más rápida.

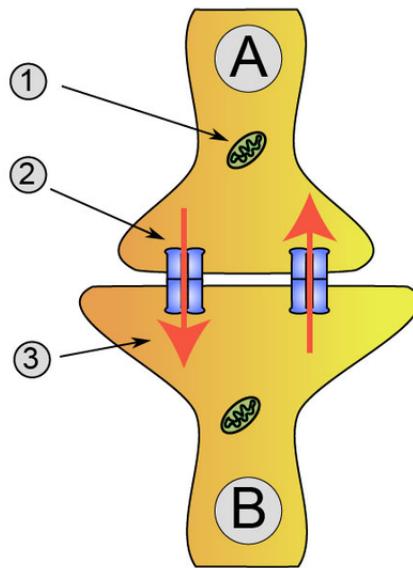


Figura 1.10 Synaptical transmission (electrical). Neurona A transmisora, Neurona B receptora, 1. Mitocondria, 2. Uniones gap formadas por conexinas, 3. Señal eléctrica, Nrets commonswiki, 23 September 2005, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Synapse_diag2.png, Inkscape 0.42, CC-BY-SA 3.0

Sinapsis química: la neurona libera moléculas neurotransmisoras a otra neurona adyacente en un pequeño espacio (la brecha sináptica) ver 1.11. Se puede resumir en cuatro etapas principales:

1. Un potencial de acción llega al botón terminal proveniente desde cono axónico.
2. Los neurotransmisores contenidos en las vesículas que están él los botones terminales, son liberados en la brecha sináptica y se dispersan.
3. Cada neurotransmisor se une a su receptor ubicado en la membrana de la neurona postsináptica.
4. El exceso de neurotransmisores que queda en el espacio sináptico es degradado o recaptado.

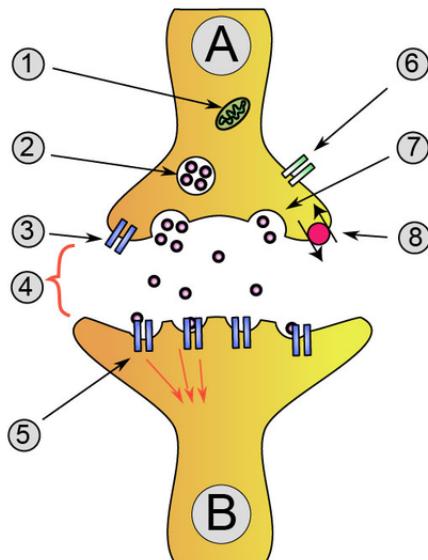


Figura 1.11 Synaptical transmission (chemical). Neurona A transmisora, Neurona B receptora, 1. Mitocondria, 2. Vesícula sináptica llena de neurotransmisor, 3. Autorreceptor, 4. Brecha sináptica, 5. Receptor de neurotransmisores, 6. Canal de calcio, 7. Neurotransmisor liberador de vesículas fusionadas, 8. Bomba de recaptación de neurotransmisores, Utilisateur:Dake, 23 September 2005, Wikimedia Commons, https://commons.wikimedia.org/wiki/File:Synapse_diag1.png, Inkscape 0.42, CC-BY-SA 3.0

Detallando el proceso de la sinapsis química, la neurona postsináptica está recibiendo un montón de señales por la liberación de neurotransmisores tanto de sus vecinos, como lo que ella misma va intercambiando, una vez que están generando el efecto completo de cambiar la polarización de la membrana, van a provocar que la neurona haga un disparo eléctrico. En el cuerpo están llegando estos intercambios de iones que se suman en el cono axónico, empiezan a viajar a través del axón, en las vainas de mielina (donde se refuerza la señal). Aquí hacemos mención por primera vez de los iones positivos: sodio y potasio, estos iones lo que hacen es, que *la membrana tenga una cierta carga la mayor parte del tiempo*. Cuando salen tres sodios entran dos potasios, entonces siempre hay

1. Neurona biológica

más positivos afuera que en el interior de la neurona, es decir, por lo general *tiene una carga más negativa que su entorno* (ver 1.17). Cuando ocurre un disparo de la neurona y se da el cambio de polarización en la membrana, se abren sus poros/ canales. El hecho que los canales abran o cierren depende de varios cambios que puedan estar ocurriendo alrededor de la neurona, en particular los que transmiten el disparo eléctrico, reaccionan ante el cambio de potencial que ocurrió en la membrana de la neurona.

La señal va pasando por los nodos de Rainvier, se refuerza y pasa por los canales iónicos ya abiertos, hasta finalmente llegar a la sinapsis a esto le llaman la **conducción a saltos** (ver 1.7). Ahora lo que ocurre al final del recorrido es que, el cambio de electricidad otra vez provoca que unas vesículas, que están en el interior de la neurona, que contienen neurotransmisores, se peguen a la membrana axónica y se liberen esos neurotransmisores nuevamente a otra neurona.

Entonces la información le va a llegar a la neurona vecina, en la forma de neurotransmisores que fueron liberados (en lo que sea que lo haya recibido, típicamente son dendritas, pero podría ser su cuerpo o su axón), eso es lo que va a percibir la otra neurona y otra vez esta otra neurona va a empezar a sumar los efectos de estos neurotransmisores, para que en algún momento decida a lo mejor disparar y otra vez provocar que se liberen neurotransmisores a su final e influir con otras neuronas.

Neurotransmisión

Cuando la neurona no está mandando señales eléctricas, tiene un potencial de reposo, su diferencia de potencial entre el interior y el exterior de la neurona, es más negativo en el interior y más positivo (o menos negativo) en el exterior. En el caso de las sinapsis químicas, llega un disparo y se altera el potencial de la membrana, entrarán las células de calcio y entra la participación de las vesículas para liberar neurotransmisores. Los neurotransmisores están flotando en la brecha sináptica, viajan hasta adherirse a los receptores de la neurona posináptica, en ese momento están alterando el intercambio normal que existe entre iones en el interior y en el exterior de la célula y van a cambiar las cargas netas que hay adentro y afuera. Este es un cambio local que está ocurriendo en las espinas dendríticas (pequeñas prolongaciones citoplasmáticas).

este cambio en sí es una especie de transferencia de información, pero muy local. Se puede distinguir entre dos efectos en la membrana:

- **El efecto excitatorio**, despolariza la membrana postsináptica es decir ahora va a ser más propensa a disparar porque ya le cambió la diferencia de potencial que tenía.
- **El efecto inhibitorio**, hiperpolariza la membrana postsináptica, es decir, va a incrementar la diferencia de potencial entre el exterior e interior, pero de tal manera que ahora ya no va a querer disparar esta neurona.

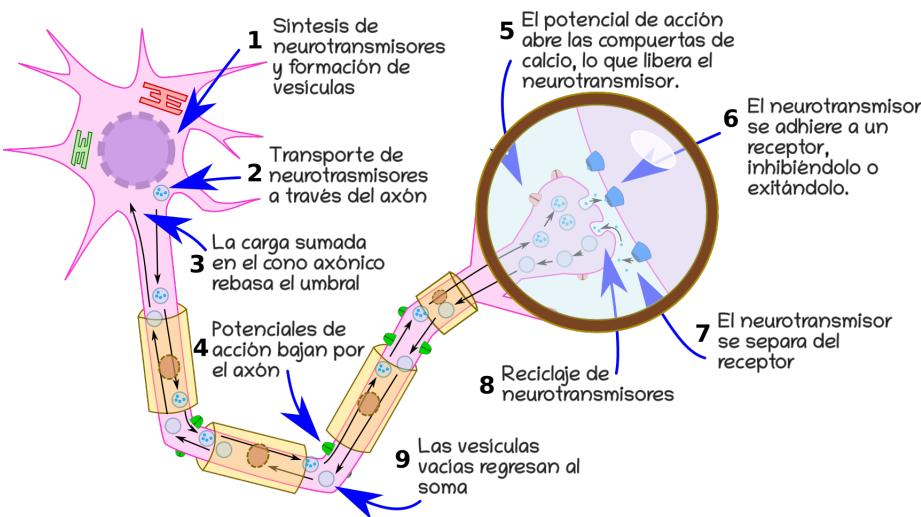


Figura 1.12 Esquema detallado de una neurotransmisión.

De estos efectos también va a darse el efecto de la **plasticidad**, que es cuando dos neuronas tienden a excitarse juntas, después de esta conexión se va a tender fortalecer, sí más bien tienden a inhibirse lo que va a suceder después es que estos canales empiezan a encoger, haciendo que se reduzcan y ya no dispare.

Ejemplos de neurotransmisores: serotonina, dopamina, oxitocina, endorfinas, adrenalina.

Campos receptivos

Aquí lo que nos interesa es, en qué región puede ser afectada una neurona. Se define un campo receptivo, como la región en la periferia sensorial dentro de la cual los estímulos pueden, influir la actividad de las células sensoriales (ver 1.13). Hay diferentes niveles donde pueden aparecer los campos receptivos tanto cerca de la piel, cerca del gusto, el olfato, donde las neuronas van a estar asociadas con otras células que les pueden ayudar, que son sensibles a los cambios correspondientes, a veces la misma neurona va a tener alguna protuberancia especializada. También podemos encontrarlos más hacia adentro del nivel de procesamiento, no necesariamente todos van a estar pegados a la parte sensorial física.

Comprenden a los receptores sensoriales que alimentan a las neuronas sensoriales, pueden ser:

- Receptores específicos en una neurona como protuberancias especializadas.
- Conjuntos de receptores capaces de activar una neurona mediante conexiones sinápticas.

1. Neurona biológica

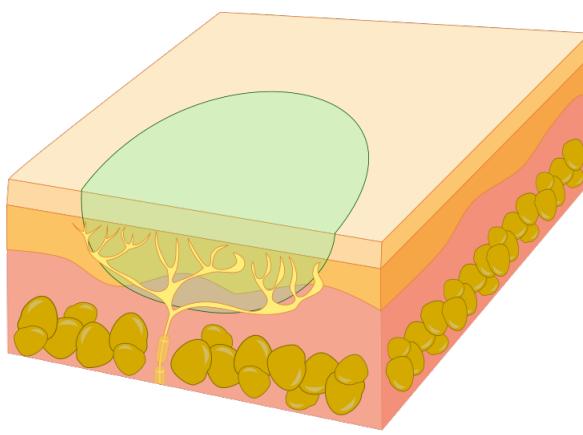


Figura 1.13 Campo receptivo. Se muestra una región que está bajo cierto estímulo, las terminales de la neurona están recibiendo información y transmitiéndola.

- Describen la ubicación donde debe estar presente un estímulo sensorial para licitar una respuesta desde una célula sensorial.

Ejemplos:

- En la piel tenemos células que nos están protegiendo en la epidermis, una de las células auxiliares **la célula de Merkel** que es sensible a la presión. Esta puede estar muy cerca a una neurona, sus terminales se activan de acuerdo a las acciones de la célula de Merkel y va a pasar la información (ver 1.14).

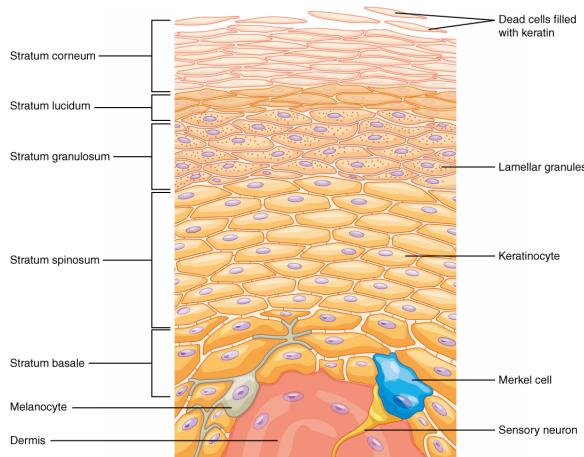


Figura 1.14 Capas de la epidermis y en azul la célula de Merkel.

- El ojo, para procesamiento visual, actualmente se utiliza una de las redes neuronales más famosas que son las redes convolucionales, que están inspiradas en el ojo,

nosotros tenemos campos receptores donde hay unos fotorreceptores en los conos y los bastones que son sensibles a luces de diferentes colores a cambios de intensidad de la luz y que pueden detectar, por ejemplo, en una cierta región física si está llegando luz o por ejemplo, si llega en la periferia entonces va a inhibir el disparo de estos elementos, por otro lado, tenemos también su complemento que permite ser estimulado por las señales que llegan, como que en la parte de afuera de un círculo y más bien se inhiben con un estímulo en la parte de afuera (ver 1.15). Esta especie de células que tienen una posición física y geométrica relevante van a determinar cuando disparan o no las neuronas. Los siguientes niveles del cerebro se van a encargar de interpretar mejor el cambio de sombras, como a una persona que pasó corriendo, un auto que se está moviendo cerca o reconocer algún tipo de alimento.

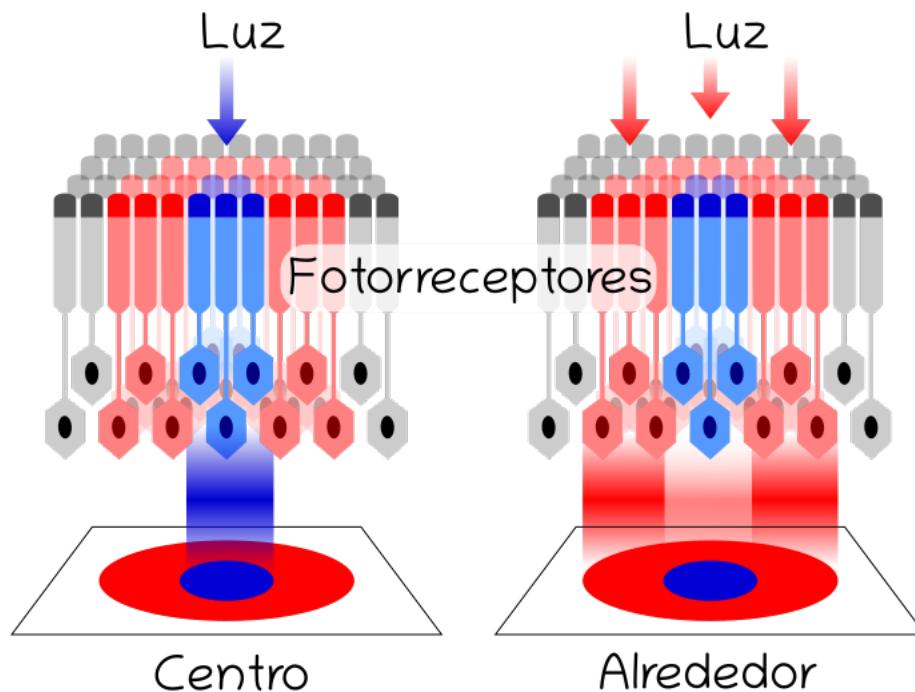


Figura 1.15 Capas de la epidermis y en azul la célula de Merkel.

Señal eléctrica

Veamos que pasa en los canales de iones y el paso de la señal eléctrica primero diferenciamos los tipos de compuertas iónicas:

- **Canal por fuga:** Estos se abren y cierran aleatoriamente, todo el tiempo están activos en la neurona, intercambiando por ejemplo: sodio y potasio.

1. Neurona biológica

- **Canal regulado por ligado:** Aquí se hace presente un neurotransmisor que es el que va a provocar que se abran o al revés impedir que se abran.
- **Canal por estímulo mecánico:** Permiten que pasen más iones o menos iones dependiendo, si se ejerció una presión, por ejemplo, con las neuronas cerca de la piel, las células de merkel.
- **Canales regulados por el voltaje:** Tienen el rol protagónico en la transmisión del pulso eléctrico (que se ha estado mencionando) describiendo uno de ellos, este canal tiene una pequeña compuerta abajo, que la puede cerrar independientemente del hecho de que el canal se abre o se cierre. Existen varias variantes de este tipo de canales regulados por el voltaje, la forma en que se están activando y desactivando sus compuertas, es lo que permite el paso del pulso.

Existen realmente una buena cantidad de iones presentes en el cerebro, pero los más protagonistas son precisamente el **potasio**, el **sodio**, el **cloro** y son los que vamos a utilizar para un modelo matemático de las neuronas.

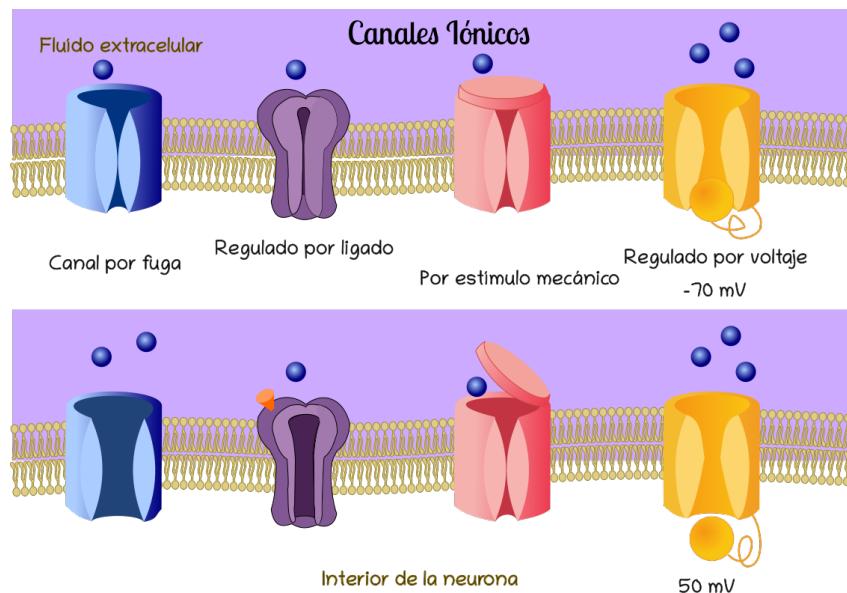


Figura 1.16 Representación de la clasificación de los canales iónicos (más representativos).

Por último veamos brevemente **la neuroplasticidad**, es lo que nos permite el aprendizaje a largo plazo en el cerebro, es un mecanismo de aprendizaje del cerebro en el cual:

Cuando las neuronas se activan simultáneamente con frecuencia la conexión entre ellas se fortalece.

Este mecanismo constituye la principal inspiración para el diseño de las redes neuronales artificiales, concretamente en esto se inspiran los algoritmos de entrenamiento. Lo que

se hace es calcular, qué conexiones debemos reforzar y cuáles debemos de debilitar para que nuestras redes neuronales calculen las funciones que a nosotros nos interesan.

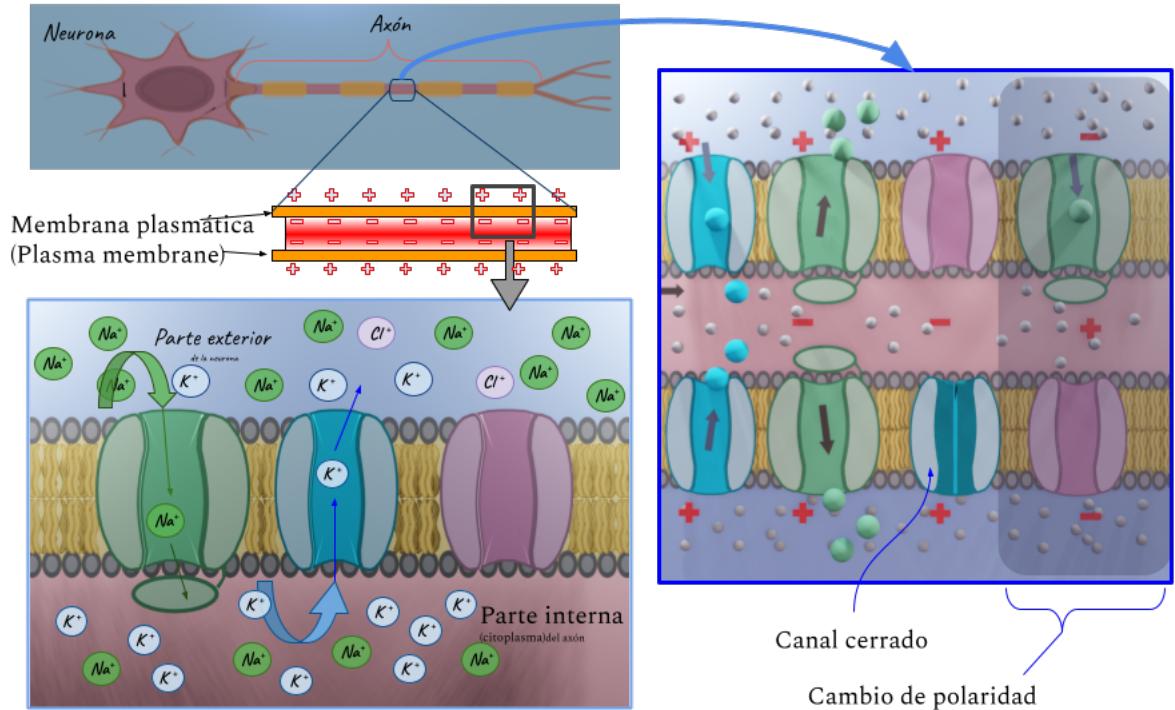


Figura 1.17 Representación de la membrana axónica en potencial de reposo en la parte inferior izquierda, y en la parte derecha con un estímulo que genera el cambio de polaridad en la misma, así como el cierre de canales y paso de iones.

2 | Modelo de Hodgkin-Huxley

Introducción

Esta sección se enfocará a la parte de transmisión de información y tipo de operaciones lógicas matemáticas que ocurren para que un cerebro pueda realizar cálculos. Específicamente se detallará la mecánica de los disparos de las neuronas, siendo estos una de las características más relevantes a la hora de modelar las redes neuronales artificiales. Si en algún momento de su vida han visto temas relacionados con: compuertas digitales, arquitectura de computadoras o diseño electrónico digital, les será más fácil abstraer el concepto, pues vamos a ver los procesos de paso de información a través de compuertas pero en un sistema biológico (de la naturaleza).

Notemos primeramente un impulso nervioso, recordemos que éste es una onda que avanza desde el cono axónico de la neurona hasta la neurona postsináptica. Esta onda electroquímica ocurre dada la diferencia de potencial entre la parte interna y externa de neurona, esta diferencia se da a consecuencia de las distintas concentraciones de iones en ambos lados de la membrana plasmática. Los estados en la membrana plasmática (del axón) se pueden diferenciar en, potenciales neuronales:

- **Potencial de reposo:** Es la diferencia de cargas en la membrana y está polarizada a -70mV. Es positiva por fuera (Na^+) y negativa por dentro por Cl^- y proteínas- y no transmite señal.
- **Potencial de acción o membrana:** Un estímulo umbral de 55 mV, despolariza la membrana y abre los canales del Na^+ y K^+ y avanza la señal nerviosa, es un cambio muy rápido en la polaridad de la membrana de negativo a positivo y vuelta a negativo.

Retomando la sinapsis eléctrica, donde participan los canales iónicos y las entradas de las neuronas (dendritas) están siendo alteradas poco a poco, hasta que ocurre la suficiente carga (diferencia de potencial) en sus dendritas y en el cuerpo de la neurona, para que desde el cono axónico se dé un disparo o potencial de acción (spike), transmitiendo la información gracias a la apertura y cierre de ciertos canales de iones cargados. Este

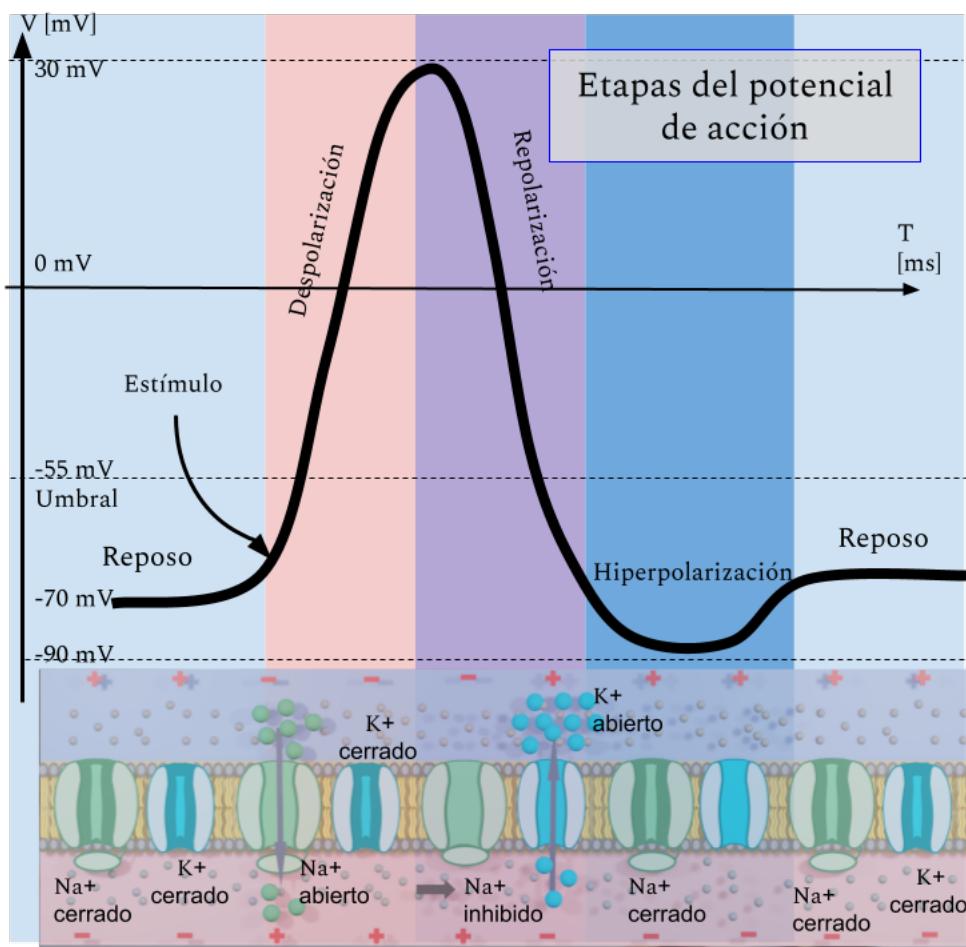


Figura 2.1 Representación gráfica de la respuesta de los canales iónicos de sodio (Na^+ en verde) y potasio (K^+ en azul) ante un estímulo de voltaje, dando como resultado un potencial de acción que viajará a lo largo de todo el axón.

cambio brusco de la diferencia de potencial, se nota en forma de un pulso eléctrico (ver 2.1), para saber más a detalle qué está ocurriendo en esta rápida elevación en la diferencia de potencial, se contará de dónde salió este modelo y por qué toma la forma que tiene.

Los primeros científicos que estudiaron el potencial de acción y dieron un modelo (de la unión sináptica eléctrica) fueron Alan Lloyd Hodgkin y Andrew Fielding Huxley alrededor de 1952, obteniendo un modelo matemático ¹, que intenta explicar qué es lo que estaba pasando en las neuronas. Ellos trabajaron con un calamar gigante (que puede medir hasta 4 metros de largo) que dado su gran tamaño, tiene un axón también bastante gigantesco, que recorre casi la mitad del cuerpo del calamar y su grosor es de medio milímetro, considerando el tamaño estándar de un axón de una neurona (1-20 μm). El axón del calamar gigante es tan grande que les permitió introducir dispositivos para medir

¹El texto original de este experimento se puede encontrar en la siguiente url: <https://physoc.onlinelibrary.wiley.com/doi/pdf/10.1113/jphysiol.1952.sp004764>

2. Modelo de Hodgkin-Huxley

el voltaje, es decir, la diferencia de potencial entre, el interior de la neurona y la parte de afuera (el ambiente externo de la neurona). Con estas mediciones experimentales que lograron obtener, se pudo determinar qué pasaba con las cargas eléctricas tanto en el interior como en el exterior y así estudiar cómo se lograba la transferencia de electricidad cuando disparaba este pulso. Se dan cuenta de que podían modelar este comportamiento como un circuito eléctrico donde están corriendo estas corrientes, si bien aún no sabían todavía cuál era exactamente el mecanismo biológico por detrás, si observaron que había dos elementos protagonicos que serían el sodio y el potasio. Notaron que estos existen en diferentes concentraciones, en la parte de afuera y en la parte de adentro de las neuronas. Con esto nosotros podemos aprender también el por qué es importante consumir algo de sal y nunca estar bajos de potasio, pues estos dos elementos son indispensables para que las neuronas puedan transmitir sus señales.

Membrana y canal

Hodgkin y Huxley se dedicaron a estudiar qué pasaba con las concentraciones de estos iones (sodio y potasio) en la parte de afuera o en la parte de adentro cuando empezaban a fluir las corrientes. El sistema parecía una especie de circuito eléctrico, se lo imaginaron como una especie de membrana porosa (lo cual es bastante cercano a lo que después se descubrió con la microscopía) y la forma en que lo vieron fue como un circuito eléctrico donde *la membrana está funcionando como un capacitor* que almacena ligeramente las cargas cuando están tratando de pasar de un lado hacia el otro y además con la cualidad que tenía de veces dejar pasar más iones y a veces no (semipermeable), modelan esto como una especie de *resistencias variables*. Bajo ciertas condiciones de voltaje de la diferencia de potencial entre la parte de afuera y la parte de adentro, estos canales permiten pasar más de estos iones (ya sean sodio, potasio o calcio) o, por el contrario, impiden su paso (ver 2.2).

Ahora se necesitan más detalles de la representación de los canales y toman en cuenta que el comportamiento de estas resistencias viene acompañado con un voltaje de reposo, en estos voltajes particulares cada tipo de ion (de la resistencia modelada) se estabiliza y ya no va a cambiar esta resistencia (ver 2.3).

Lo que observan es que el **ion de sodio (Na^+)** y su resistencia va a variar dependiendo del voltaje, a esto se le llama un **canal transitorio** porque en ciertos voltajes si puede pasar; si es muy bajo, no puede pasar y si rebasa un cierto umbral entonces se vuelve a tapar y ya no puede pasar. Lo que sucede con el **ion de potasio (K^+)** es que, puede salir si el voltaje está más allá de un cierto valor, si no, no pasan y va variando un poquito que tanto puede pasar, a esto se le llama **canal persistente**. Por estas características de que el potasio es un intervalo dentro de la recta y el sodio es a partir de cierto valor, por tanto, se les modelan de maneras ligeramente diferentes. Más adelante se descubrió porque tenían este comportamiento, básicamente el canal de potasio es una puerta hecha de cuatro subpuertas por donde los elementos pasan o no pasan, el canal de sodio es

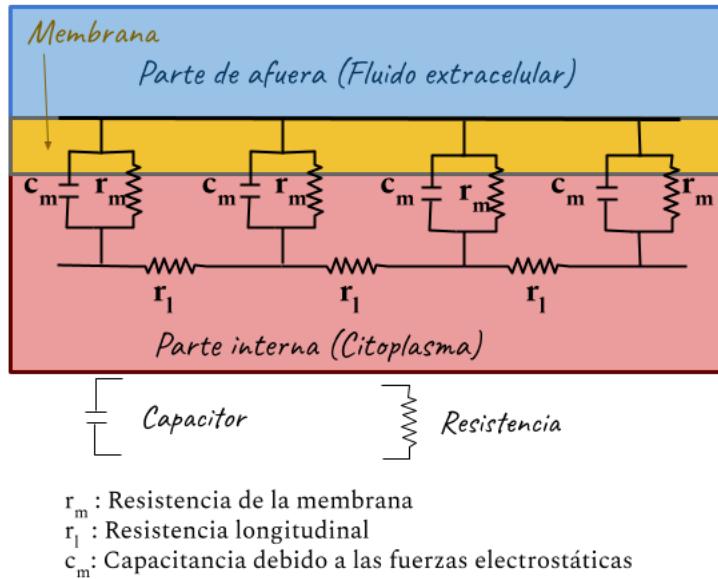


Figura 2.2 Un primer modelo de la membrana axónica modelada como circuito eléctrico. La parte amarilla es la membrana.

como una compuerta que está hecha de tres subpuertas que se pueden abrir y tiene aparte un tapón extra, que hace que aunque estas tres están abiertas bloquee toda la compuerta. Las neuronas están trabajando con muchos más iones aparte de estos dos, uno que destaca bastante es el caso del cloro (Cl^-) que tiene carga negativa. Se tienen canales para intercambio aleatorio de otros iones, **L** un canal aleatorio (leaky).

Entonces con lo que ellos midieron experimentalmente, notaron cómo se estaban comportando estas resistencias dependiendo del voltaje o la diferencia de potencial que había entre ambos lados de la membrana. A partir de estas pudieron describir matemáticamente y simular los disparos que se conocen como potenciales de acción. Vamos a ver cuáles fueron estos conceptos de electricidad que se están utilizando para el modelo tenemos este concepto de potenciales eléctricos.

- Potenciales eléctricos E ó V ; resultan de la separación de cargas opuestas. Se mide en mV .

* $E_{(\text{Na}, \text{K}, \text{L})}$ voltaje en reposo para los iones de Na, K y L, o también conocido como potencial de inversión iónico, es el potencial de membrana en el que no hay flujo neto (total) de ese ion en particular de un lado de la membrana al otro.

* V_m , el potencial eléctrico de la membrana.

- Corriente I ; Movimiento de cargas. Se mide en μA .

* $I_{(\text{Na}, \text{K}, \text{L})}$ corriente entrante a los canales de Na, K o L.

2. Modelo de Hodgkin-Huxley

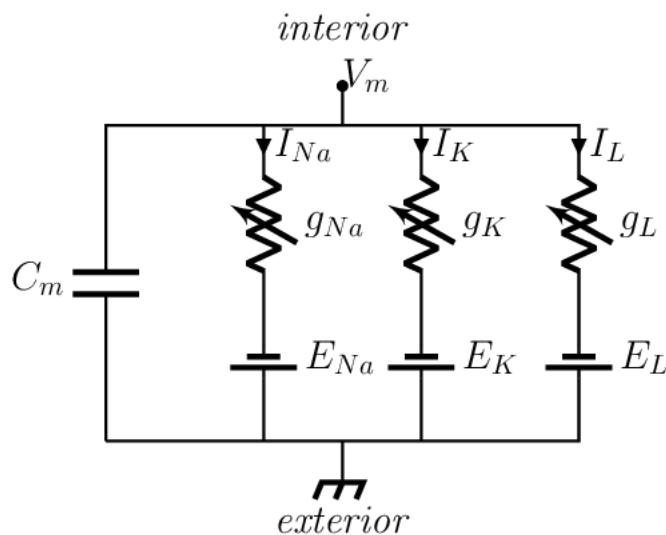


Figura 2.3 Modelo de la membrana axónica modelada como circuito eléctrico, con los distintos canales presentes y su voltaje de reposo.

- Resistencia R ; Medida de la oposición al movimiento de las partículas cargadas.
- Capacitancia o capacidad eléctrica C . Cantidad de energía eléctrica almacenada en un capacitor para una diferencia de potencial eléctrico dada.
 - * C_m la capacitancia de la membrana.
- Conductancia g ; Inverso de la resistencia $\frac{1}{R}$, es decir, facilidad de transmisión de las partículas cargadas.
 - * $g_{(Na,K,L)}$ la conductancia del canal de sodio, potasio, cloro y la L también refiriéndose a otros canales de iones.

Lo que está pasando en los **potenciales eléctricos** es que hay mucho sodio en la parte externa de la membrana por ej. tres iones de sodio que son cargas positivas por dos iones de potasio que hay en la parte interna, entonces hay muchas más cargas positivas en la parte de afuera que las que hay en la parte de adentro y eso es lo que provoca la diferencia de cargas que es lo que estamos viendo como un potencial eléctrico.

La capacidad eléctrica o **capacitancia** es la que estamos utilizando para modelar la membrana conformada por lípidos, que es una capa de grasa y esa es la cantidad de energía eléctrica almacenada en un capacitor para una diferencia de potencial eléctrico dada. Éste comportamiento bastante interesante porque las cargas quedan almacenadas un momento, pero se van liberando poco a poco y se va descargando ese capacitor.

Durante el experimento con el axón, se le dieron cargas eléctricas directamente al axón y gracias a eso lograban ir midiendo que era lo que estaba pasando con las concentraciones de cargas afuera y adentro en el caso de las neuronas reales, esto en un ambiente

no alterado ocurre cuando entran en juego los neurotransmisores y provocan que haya cambios, en estas corrientes. Entonces Hodgkin y Huxley jugaron el rol que tendrían que jugar usualmente los *neurotransmisores* para abrir otras compuertas. Nosotros en la manera en la que lo vamos a simular es precisamente con estas corrientes que son las que se están poniendo en el experimento y vamos a ver cómo reacciona el axón.

Potenciales de Nerst o de reposo

Los Potenciales de Nerst o de reposo son los potenciales a los cuales el flujo neto de iones a través de los canales abiertos es cero. Aquí vemos precisamente porque estamos utilizando la E generalmente la vamos a utilizar para referirnos a la diferencia de potencial entre la parte de afuera de la célula y la parte de adentro, las vamos a utilizar para representar a aquellos voltajes donde cada una de las compuertas encontrarán su equilibrio. Estos voltajes son distintos para cada una de las compuertas, esto va a provocar precisamente la dinámica de la de la neurona, por ejemplo:

- $E_{Na} \ 50mV$
- $E_{Ca} \ 150mV$
- $E_K \ 80mV$
- $E_{Cl} \ 60mV$

Aquí vemos que el sodio estaría su equilibrio en un valor positivo, el calcio que es el que va a jugar un rol de que se activen los neurotransmisores y se transmita el disparo, observamos que el voltaje tendría que ser bastante positivo. El potasio que es el que usualmente está trabajando intercambiándose casi todo el tiempo en la neurona, veremos que el punto de equilibrio usual de la neurona anda por los $-76mV$ y el del cloro. Cada uno de estos canales pues está tratando de jalar la dinámica hacia su potencial de equilibrio y no hay precisamente un acuerdo entre ellos y eso es precisamente lo que hace que las neuronas cobren "vida".

Modelo de la membrana como bicapa de lípidos

La membrana de una neurona es modelada como un elemento de un circuito con capacitancia C_m y potencial V , las corrientes que fluye a través de la bicapa lipídica están regidos por las siguientes ecuaciones:

$$I_m = C_m \frac{dV_m}{dt} \quad (2.1)$$

2. Modelo de Hodgkin-Huxley

Esta sería la ecuación principal (2.1) donde $\frac{dV_m}{dt}$ está representando el cambio voltaje en la membrana respecto al tiempo.

$$C_m \frac{dV_m}{dt} = -g_{Na}m^3h(V - E_{Na}) - g_Kn^4(V - E_K) - g_L(V - E_L) + I_{ext} \quad (2.2)$$

Cada una de las partes del lado izquierdo de la ecuación 2.2 corresponde a las compuertas de los canales y la corriente de un estímulo externo que pueda influir a la membrana (este estímulo siempre será desde el exterior hacia el interior).

Retomando lo escrito anteriormente el canal de sodio es una compuerta compuesta de **tres** subpuertas y una subpuerta que actúa como tapón y el canal de potasio es una compuerta compuesta de **cuatro** subpuertas iguales, con esto podemos notar claramente que las conductancias sean representadas como:

- $\frac{1}{R_{Na}} = g_{Na} * m^3 * h$ donde g_{Na} es una constante que representa el valor de la conductancia máxima, m es la proporción de los canales de sodio abiertos (representa la concentración de sodio) y nos indica la activación (subpuertas abiertas) del canal, h es el “tapón” de la compuerta que puede impedir el paso de iones independientemente de las otras tres subpuertas, es decir la inactivación (compuerta bloqueada). Los movimientos combinados de m y h son los que controlan la compuerta de sodio.
- $\frac{1}{R_K} = g_K * n^4$ donde g_K es una constante que representa el valor de la conductancia máxima, n es la proporción de los canales de potasio abiertos (representa la concentración de potasio) y nos indica la activación del canal de potasio.
- g_L es una constante, de los canales por fuga, que representa la concentración de los demás iones que pasan por la membrana.

Ahora m , n y h , son variables de activación que describen la probabilidad de que los canales iónicos estén abiertos, se puede describir mediante las siguientes ecuaciones diferenciales ordinarias:

$$\frac{1}{\gamma(T)} \frac{dn}{dt} = \alpha_n^\infty(V)(1 - n) - \beta_n(V)n = \frac{n(V) - n(t)}{\tau_n(V)} \quad (2.3)$$

$$\frac{1}{\gamma(T)} \frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m = \frac{m^\infty(V) - m(t)}{\tau_m(V)} \quad (2.4)$$

$$\frac{1}{\gamma(T)} \frac{dh}{dt} = \alpha_h(V)(1 - h) - \beta_h(V)h = \frac{h^\infty(V) - h(t)}{\tau_h(V)} \quad (2.5)$$

2.4 Modelo de la membrana como bicapa de lípidos

Donde la ecuación 2.3 representa al canal de potasio y las ecuaciones 2.4 y 2.5 representando al canal de sodio tomando en cuenta que tiene dos tipos de subpuertas.

Las expresiones de α y β están dadas por las siguientes ecuaciones:

$$\begin{aligned}\alpha_n &= \frac{0.01(10 - V)}{\exp\left(\frac{10 - V}{10}\right) - 1} & \beta_n &= 0.125 \exp\left(-\frac{V}{80}\right) \\ \alpha_m &= \frac{0.01(25 - V)}{\exp\left(\frac{25 - V}{10}\right) - 1} & \beta_m &= 4 \exp\left(-\frac{V}{18}\right) \\ \alpha_h &= \frac{0.07}{\exp\left(-\frac{V}{20}\right)} & \beta_h &= \frac{1}{1 + \exp\left(\frac{30 - V}{10}\right)}\end{aligned}$$

Los factores α y β se denominan como constantes de velocidad de transición. α es el número de veces por segundo que se abre una puerta que está en estado cerrado, mientras que β es el número de veces por segundo que se cierra una puerta que está en estado abierto. Si la membrana tiene una carga negativa, α debe aumentar y la β debe disminuir, cuando la membrana esté despolarizada.

Hasta ahora sabemos que en la bicapa de lípidos, una pequeña carga está pasando entre sus capas de grasa. También sabemos que la carga es almacenada por un breve periodo de tiempo, dando como resultado que la bicapa se comporte como un **capacitor**. Esta membrana también está con cierta resistencia al paso de corriente. Con esto tenemos el siguiente diagrama ² 2.4

Tenemos dadas las siguientes ecuaciones:

$$I_C + I_R - I_{ext} = 0 \quad (2.6)$$

$$C \frac{dV}{dt} + \frac{V}{R} - I_{ext} = 0 \quad (2.7)$$

$$C \frac{dV}{dt} = -\frac{V}{R} + I_{ext}$$

Ahora por la ley de corriente de Kirchhoff ³ tenemos que la suma de las corrientes del capacitor y la resistencia debe ser cero por la conservación de corriente y si consideramos

²Otra explicación más profunda de las ecuaciones dadas partir del diagrama 2.4la podemos encontrar en https://neurowiki.case.edu/wiki/Action_Potential_IV:_Hodgkin-Huxley_Equations_and_Other_Conductances

³La ley de la corriente de Kirchhoff dice que la suma de todas las corrientes que fluyen hacia un nodo es igual a la suma de las corrientes que salen del nodo

2. Modelo de Hodgkin-Huxley

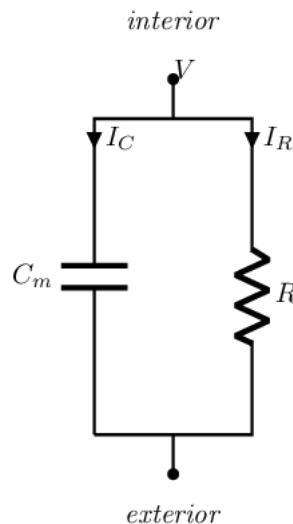


Figura 2.4 Modelo de la bicapa de lípidos donde, V son los cambios de voltaje en la membrana que es el potencial eléctrico, I_C es la corriente del capacitor, I_R es la corriente de la resistencia, C_m es la capacitancia de la membrana, R es la resistencia.

un factor adicional de una corriente externa aplicada o administrada a la neurona, tenemos la ecuación 2.6.

Después tenemos la relación entre la diferencia de potenciales, que almacena energía y la carga eléctrica que guarda, donde: C es la capacidad, medida en faradios, Q la carga eléctrica almacenada, medida en culombios, V la diferencia de potencial medida en voltios. Entonces $C = Q/V$, despejando a Q tenemos $Q = CV$ y derivando de ambos lados respecto al tiempo y considerando que C es una constante al ser una propiedad de la membrana $\frac{dQ}{dt} = C \frac{dV}{dt}$. Como la definición de corriente es el cambio de carga en el tiempo tenemos que $I_C = C \frac{dV}{dt}$. Notemos finalmente la corriente de la resistencia I_R , recordando la ley de Ohm ⁴ la podemos rescribir como $\frac{V - V_{rest}}{R}$. Sustituyendo de lo anterior en la ecuación 2.6 se obtiene la siguiente ecuación:

$$C \frac{dV}{dt} + \frac{V - V_{rest}}{R} - I_{ext} = 0 \quad (2.8)$$

$$C \frac{dV}{dt} = -\frac{V - V_{rest}}{R} + I_{ext}$$

⁴La ley de Ohm establece que la diferencia de potencial V que aplicamos entre los extremos de un conductor determinado es directamente proporcional a la intensidad de la corriente I que circula por el conductor, es decir $V = R * I$. Notemos también que $V = V_m - V_{rest}$

Ahora multiplicando todo por R:

$$RC \frac{dV}{dt} = -V + (V_{\text{rest}} + RI_{\text{ext}}) \quad (2.9)$$

Denotando RC como la constante de tiempo τ y tomando en cuenta que en cuanto se aplica la corriente va a empezar a cambiar el voltaje poco a poco hasta establecerse en un voltaje de equilibrio (ahí se va a quedar quieta). Entonces cuando el voltaje ya no está cambiando con el tiempo quiere decir que su derivada con respecto al tiempo es cero. Observemos que $dV/dt = 0$ significaría que V es igual a infinito y que el voltaje en el estado estacionario cuando, $dV/dt = 0$ depende del potencial de reposo y del producto entre la resistencia con la corriente externa suministrada, entonces tenemos que:

$$V_{\infty} = V_{\text{rest}} + RI_{\text{ext}} \quad (2.10)$$

Sustituyendo con 2.10 y la constante, en 2.9 tenemos que:

$$\tau \frac{dV}{dt} = -V + V_{\infty} \quad (2.11)$$

Las conductancias iónicas

Nuestro objetivo aquí es encontrar ecuaciones que describan las conductancias con precisión razonable y lo suficientemente simples para el cálculo teórico de *el potencial de acción* y *el período refractario*.

Si tomamos las ecuaciones diferenciales anteriores notamos que las soluciones tienen este tipo de forma 2.5:

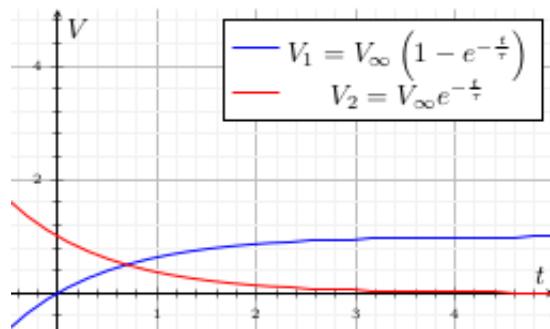
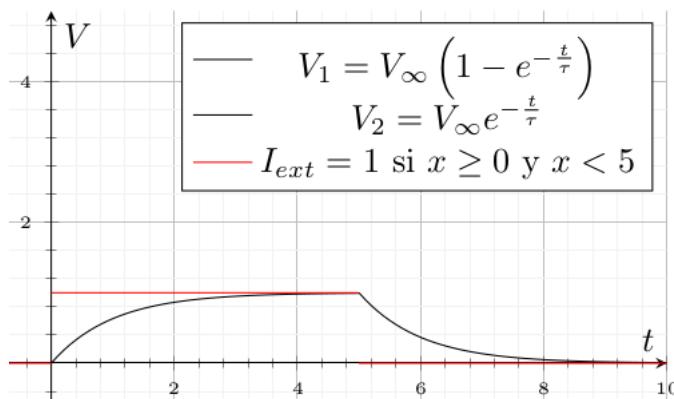


Figura 2.5 Soluciones para el pulso.

Donde si estamos aplicando una corriente externa lo que sucede es lo que estamos viendo en azul, un exponencial que va creciendo y que tiende hacia un cierto valor límite que sería de infinito. Si dejamos de aplicar la corriente externa entonces ahora tendremos

2. Modelo de Hodgkin-Huxley

**Figura 2.6** Soluciones para el pulso escalón.

un exponencial, pero que tiende hacia el cero y se va a estabilizar en cero, lo que vemos en rojo.

Simulando lo que hicieron Hodgkin y Huxley que fue al axón de repente darle un toque, siendo en el origen de la gráfica (que visualizamos en la figura 2.6) la parte en la que le están dando el toque al axón, momentos antes estaba quieta la neurona de repente le aplican una cierta cantidad de electricidad y va a empezar a cambiar el comportamiento de los canales la porosidad de la membrana, vamos a ver que empieza a incrementarse la diferencia de potencial hasta que llegan a un nuevo equilibrio (alrededor de $t = 3$) y si siguieran dándole el toque en esa cantidad se quedaría ahí la neurona, ya no veríamos más cambios lo que va a suceder entonces es que, retiramos las pinzas (se le deja de dar el toque) y los canales otra vez van a empezar a regresar a la normalidad y vamos a ver un descenso en adelante.

Entonces hasta aquí ya tenemos la idea de cómo va a reaccionar la neurona ante cierto estímulo; sin embargo, esto que acabamos de ver en las gráficas sería como si tuviéramos un solo tipo de canal, ahora qué pasa si consideramos que tenemos diferentes tipos de canales pasando iones, en condiciones distintas. Aquí es donde va a importarnos el hecho de que existen diferentes tipos de canales con voltajes de equilibrio diferente. Retomando a los potenciales de Nerst E_{Na} , E_K , E_L notemos que están dados por:

$$E = \frac{k_B T}{zq} \ln \frac{[\text{adentro}]}{[\text{afuera}]} \quad (2.12)$$

Estos potenciales están relacionados con las características termodinámicas, en la ecuación anterior k_B la constante de Boltzman, q es la carga del ion, y z es el número de iones. El logaritmo natural representando el promedio de cuántos elementos tenemos en la parte de adentro con respecto a cuántos elementos tenemos en la parte de afuera.

Considerando los diferentes puntos de equilibrio en los cuales se puede encontrar la diferencia de potencial en la membrana, vamos a distinguir entre tres estados de esta (también se puede ver en 2.1):

1. Polarizada en su estado de reposo con $V < 0$ ($V \approx -70\text{mV}$).

- Su estado de reposo, cuando la neurona no está haciendo nada simplemente están corriendo los sodios y entran los potasios.

2. Despolarizada cuando $V \geq 0$.

- Cuando en sus dendritas y en el cuerpo de la neurona se acumula una carga muy grande (un voltaje eléctrico, disparo), se abre la compuerta de sodio y van a empezar a entrar el sodio, esta diferencia de potencial que existía entre lo fuera y lo adentro se va a reducir de hecho se puede llegar a reducir bastante dependiendo de la carga que se le aplique.
- Iones positivos entran a la membrana.
- Valores positivos en la diferencia de potencial.

3. Hiperpolarizada cuando la diferencia de potencial incrementa su magnitud $V \ll 0$.

- En cuanto se despolarice van a empezar interacciones entre los diferentes tipos de canales que lo que van a intentar hacer es regresar a la neurona en su estado normal.
- Los canales de potasio abren sus compuertas, provocando que salga el potasio que está dentro del axón (en el citoplasma).
- Iones positivos salen de la membrana.
- Si antes estaba quieta a los -70mV , ahora va a quedar todavía más abajo alrededor de -90mV . Esto va a permitir un fenómeno que se le conoce como *el periodo de refracción* y ese periodo sirve para que simplemente se lance un disparo y que el comportamiento eléctrico no se rebote otra vez en dirección contraria en la neurona, va a quedar muy quieta la neurona durante un rato y después regresará otra vez a su estado de equilibrio.

Modelo de las compuertas iónicas controladas por voltaje

Retomando el modelo del circuito eléctrico modelando la membrana, junto con los canales y los iones, volvamos a verlo ahora en la 2.7

2. Modelo de Hodgkin-Huxley

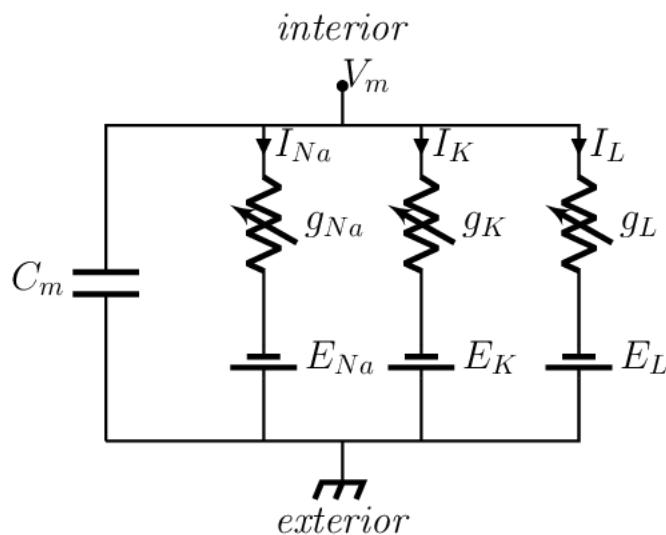


Figura 2.7 Modelo de la membrana axónica modelada como circuito eléctrico, con los distintos canales presentes y su voltaje de reposo.

Recordemos brevemente las definiciones de los dos tipos de canales protagonistas en el modelo:

Definición 2.1

Canal persistente Tiene un solo tipo de compuerta y dos estados posibles:

1. **Activado**
2. **Desactivado**

Definición 2.2

Canal transitorio Tiene compuertas de activación e inactivación, y tres estados:

1. **Activado** Ambas compuertas abiertas.
2. **Desactivado** Compuerta de activación cerrada, inactivación abierta.
3. **Inactivada** Compuerta de inactivación cerrada.

Y retomando la primera ecuación diferencial donde tenemos, por un lado, la corriente que está pasando a través del capacitor, por otro lado, vamos a tener las corrientes que están circulando a través de los diferentes canales,

$$C_m \frac{dV_m}{dt} = -g_{Na} m^3 h(V_m - E_{Na}) - g_K n 4(V_m - E_K) - g_L (V_m - E_L) + I_{ext} \quad (2.2)$$

Las capacitancias y variables del lado izquierdo están explicadas en la sección [Modelo de la membrana como bicapa de lípidos](#), aquí vamos a retomar las ecuaciones 2.3,2.4,2.5 de esa misma sección, (recordemos que estas ecuaciones describen la probabilidad de que los canales iónicos estén abiertos) que son las siguientes:

$$\frac{1}{\gamma(T)} \frac{dn}{dt} = \alpha_{n^\infty}(V)(1-n) - \beta_n(V)n = \frac{n(V) - n(t)}{\tau_n(V)} \quad (??)$$

$$\frac{1}{\gamma(T)} \frac{dm}{dt} = \alpha_m(V)(1-m) - \beta_m(V)m = \frac{m^\infty(V) - m(t)}{\tau_m(V)} \quad (??)$$

$$\frac{1}{\gamma(T)} \frac{dh}{dt} = \alpha_h(V)(1-h) - \beta_h(V)h = \frac{h^\infty(V) - h(t)}{\tau_h(V)} \quad (??)$$

Ahora notemos los elementos en estas ecuaciones anteriores con ion pudiendo denotar las compuertas del potasio n o del sodio, ya sea m o h :

- $\frac{1}{\gamma(T)}$ Este es el coeficiente de escala temporal, dependiente de la temperatura los por eso está apareciendo aquí una t . Para las simulaciones que nosotros vamos a hacer vamos a pensar que estamos en una temperatura fija.
- $\alpha_{\text{ion}}(V)$ probabilidad de que una compuerta transite de cerrada a abierta.
- $\beta_{\text{ion}}(V)$ probabilidad de que una compuerta transite de abierta a cerrada.
- $\text{ion}^\infty(V)$ probabilidad de compuerta abierta en el equilibrio cuando $t \rightarrow \infty$.
- (ion) Probabilidad de que cada compuerta (n, m, h) esté abierta.
- $(1 - \text{ion})$ Probabilidad de que cada compuerta (n, m, h) esté cerrada.
- $\tau_{\text{ion}}(V)$ Tiempo que toma llegar al equilibrio.

Lo que vamos a ver es que forma de escribir la ecuación depende precisamente del número de compuertas que tenían para poder abrirse y cerrarse. Reescribir la ecuación de esta manera lo que nos permite es medirlo en términos de estas probabilidades de que se abran y cierren las compuertas que serían

Esta probabilidad se empieza a alterar conforme cambiamos el voltaje, pero no va a llegar a su valor de equilibrio sino hasta después de pasado un cierto periodo.

2. Modelo de Hodgkin-Huxley

Dinámica del voltaje durante un disparo

Ahora qué pasa cuando tomamos en cuenta que todas las compuertas están reaccionando al mismo tiempo. Notemos primeramente como está reaccionando la membrana ante un impulso eléctrico o voltaje, en la siguiente imagen 2.8.



Figura 2.8 En la primera parte los canales están en un estado de reposo y la membrana está polarizada. En la segunda parte los canales han sido afectados por un impulso recibido desde el cono axónico, las compuertas de sodio se abren permitiendo el paso de iones de sodio al interior de la membrana y dejando a la membrana despolarizada. Momentos después la membrana llega a un estado de hiperpolarización donde intentará regresar al estado de equilibrio que tenía previamente, para esto la subpuerta de inactivación de sodio cerrará hasta no permitir el paso de sodio y el canal de potasio abrirá sus compuertas para dar salida a los potasios (iones positivos) que fluyan hacia el exterior de la membrana, así dejando el voltaje de la membrana incluso más negativo de lo que tenía durante su estado polarizado.

Ahora lo que vemos en la imagen 2.8 se grafica de manera un poco más apegada a lo que pasa en los experimentos de la siguiente forma en la imagen 2.9.

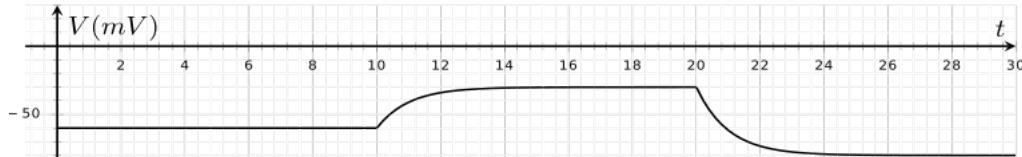


Figura 2.9 Gráfica del cambio de voltaje en la membrana dado un impulso a través del tiempo. Cuando se rebasa el voltaje umbral, los canales de Na^+ y K^+ interactúan para producir una rápida despolarización de la membrana provocando una elevación del voltaje, para luego hiperoxplorizarla. Durante el momento de despolarización la membrana puede llegar a valores positivos en este caso en particular el estímulo no es tan grande y se queda en valores negativos.

Ahora lo anterior está en términos de lo deseado, veamos entonces que pasó en las mediciones de Hodgkin y Huxley con el axón en las siguientes gráficas 2.10, donde nos está mostrando como las subpuertas n y los factores τ , α y β del canal de potasio se comportaron antes, durante y después del estímulo del voltaje. Recodemos que n nos indica la probabilidad que las compuertas de potasio estén abiertas, este es un factor

adimensional. τ es el tiempo que tarda en llegar a un estado de equilibrio. α y β las probabilidades que las compuertas del canal de potasio pasen de cerrados a abiertos y viceversa.

Entonces notamos que en el mismo periodo de tiempo, la membrana está en reposo y conforme va recibiendo el voltaje incrementa la probabilidad que las compuertas de potasio estén abiertas, es decir n va incrementando conforme al estímulo, mientras que el estado de reposo es claramente alterado provocando que el valor τ disminuya considerablemente. La probabilidad que las compuertas de potasio pasen de un estado cerrado a uno abierto durante el proceso (α) aumenta prácticamente de forma exponencial, mientras que la probabilidad de que pasen de abierto a cerrado disminuye poco a poco. Con esto cumpliendo lo esperado en la dinámica del voltaje, al notar como está reaccionando el canal de potasio durante la polarización y despolarización (pulso).

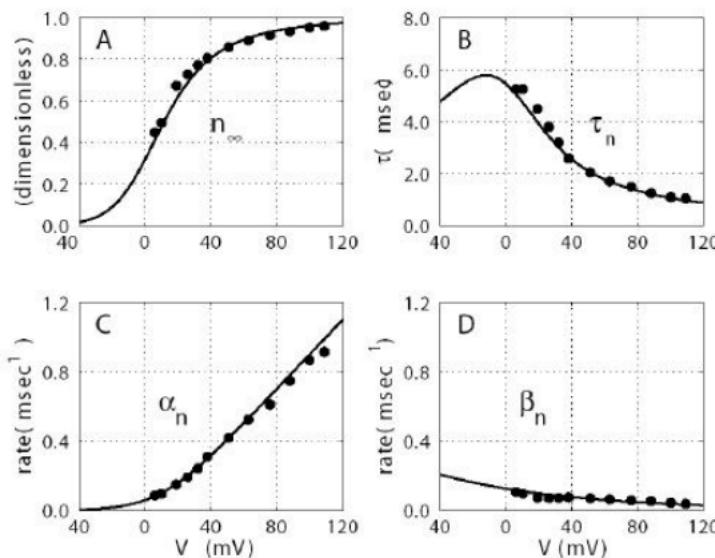


Figura 2.10 Medición experimental de los parámetros y ajuste manual de curvas. Imagen de Nelson 2004

Con estas medidas experimentales ellos dan con curvas paramétricas ajustadas a los factores α y β expresadas de la siguiente forma:

$$\begin{aligned} \alpha_n &= \frac{0.01(10-V)}{e^{\left(\frac{10-V}{10}\right)} - 1} & \beta_n &= 0.125e^{-\frac{V}{80}} \\ \alpha_m &= \frac{0.01(25-V)}{e^{\left(\frac{25-V}{10}\right)} - 1} & \beta_m &= 4e^{-\frac{V}{18}} \end{aligned}$$

2. Modelo de Hodgkin-Huxley

$$\alpha_h = 0.07e^{-\left(\frac{V}{20}\right)}$$

$$\beta_h = \frac{1}{e^{\left(\frac{30-V}{10}\right)} + 1}$$

Ahora veamos las gráficas de la dinámica del disparo pero ya con los las compuertas del potasio y sodio interactuando al mismo tiempo, en las figuras 2.11 y 2.12

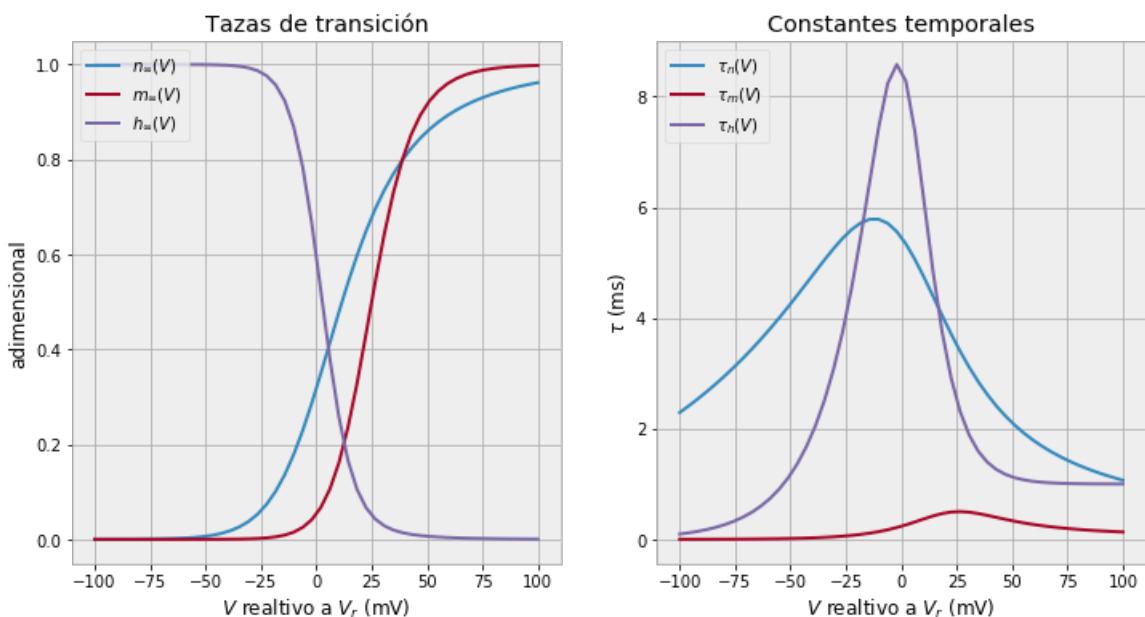


Figura 2.11 Activación e inactivación de los canales. Del lado izquierdo vemos que alrededor de los -50mV aumenta la probabilidad que las compuertas de potasio abran y los potasios salgan, el sodio tarda un poco más en reaccionar para dejar entrar a los sodios, y la probabilidad que quede no bloqueado disminuye, hasta bloquear por completo a los iones de sodio alrededor de los 50mV. Del lado derecho vemos como τ , va interactuando conforme al voltaje, el Na^+ reacciona rápidamente ante él impuso pues regresa rápidamente a su estado de equilibrio, esto indica porque aún que el sodio abre sus compuertas, la compuerta de bloqueo se cierra rápidamente, dejando inactivado el canal, luego el K^+ reacciona más lentamente para regresar a su estado de equilibrio dejando más tiempo activado el canal y salgan los potasios.

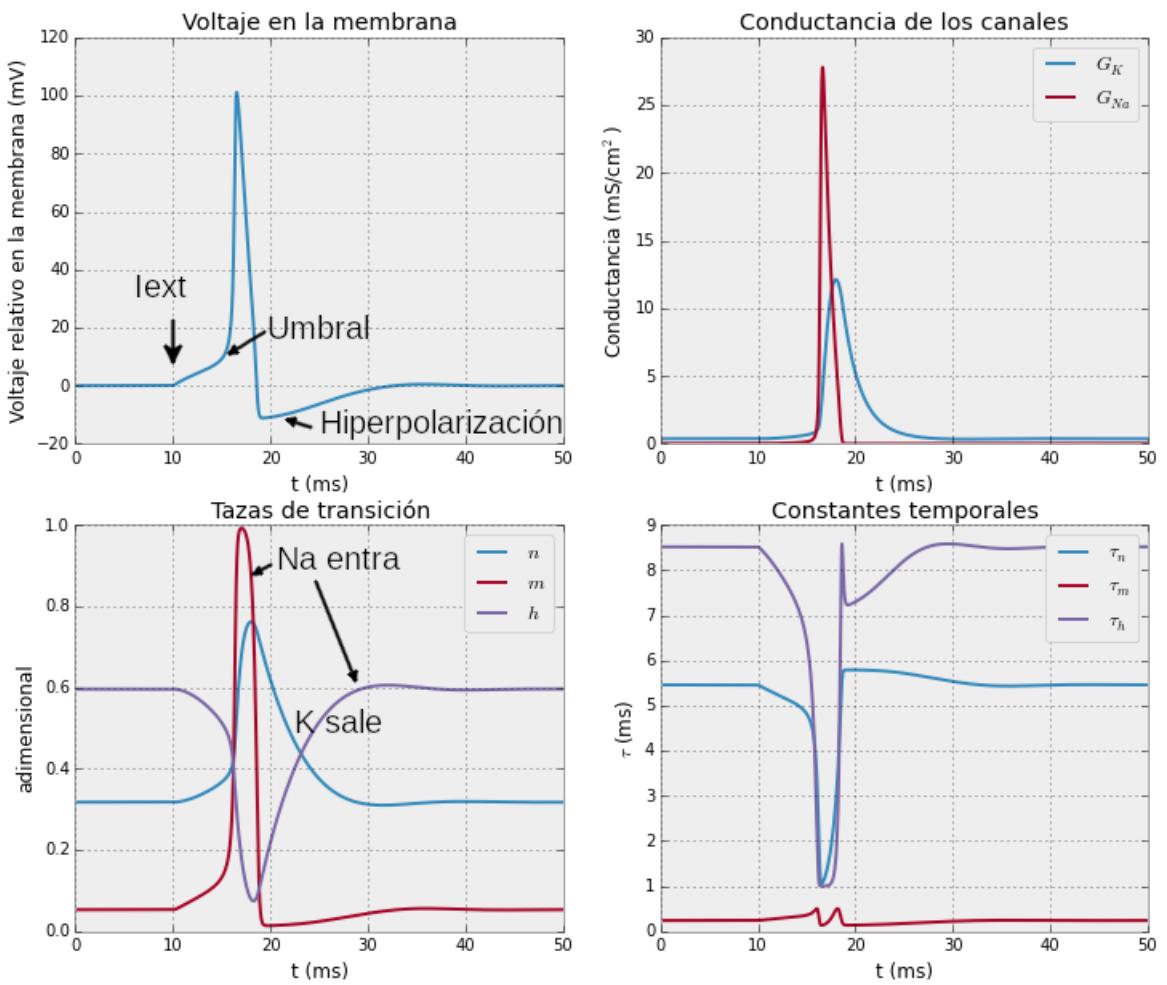


Figura 2.12 Medidas experimentales del estímulo y la reacción de todos los componentes de la membrana. El origen en estas gráficas realmente representa el punto de equilibrio que es alrededor de los -70mV. Del lado superior izquierdo mostrando como cambia el voltaje de la membrana respecto al tiempo, del lado superior derecho el comportamiento de la conductancia de los canales mostrando que aunque la conductancia del sodio reacciona más, es más breve, mientras que el potasio reacciona en menor medida pero durante más tiempo. En la parte inferior derecha la taza de transición de las compuertas n, m y h , mostrando que en cuando llega el impulso el primero en reaccionar es el sodio permitiendo que entre muy brevemente el sodio e inactivándose casi inmediatamente después, mientras que en ese momento los potasios se están liberando hacia la parte exterior de la neurona. Finalmente del lado inferior izquierdo mostrando como se comporta las constantes temporales, mostrando que las compuertas de activación de sodio no son tan afectadas comparado a la compuerta de bloqueo, notando algunos cambios bruscos en esta, mientras que el potasio si se nota más afectado y durante más tiempo pero recuperándose sin cambios bruscos.

Simulación usando el método de Euler

En esta sección se listará como estamos resolviendo las ecuaciones diferenciales, para tener una simulación numérica y se trata del algoritmo de integración⁵. Las gráficas de la sección anterior se obtuvieron con el método de Euler que se describe más adelante.

Algoritmo 1 Algoritmo de integración de Euler [Wells pp51].

```

1: function INTEGRADISPARO( $T, \Delta T, V_0, I_{ext}(t)$ )
2:   Inicializar arreglos de logitud  $T[]$ ,  $V[]$ ,  $n[]$ ,  $m[]$ ,  $h[]$ ,  $G_{Na}[]$ ,  $G_K[]$ ,  $\tau_n[]$ ,  $\tau_m[]$ ,  $\tau_h[] \leftarrow$ 
   arreglo[numeroDePasos]
3:    $V[0] \leftarrow V_0$ 
4:   for  $t = 0$  at  $= T$  cada  $\Delta t$  do
5:     Calcular  $\alpha_n, \beta_n, \alpha_m, \beta_m, \alpha_h, \beta_h$  utilizando  $V(t)$ .
6:     Calcular las tres  $\tau_x$  y  $x^\infty$  apartir de las anteriores.
7:     Calcular las probabilidades de las compuertas  $n, m, h$ , utilizando las ecuacio-
       nes en diferencias en su forma matricial  $\Pi(t + \Delta t) = A_\pi \Pi(t) + B_\pi$ .
8:     Calcular  $G_{Na} = g_{Na} m^3 h$  y  $G_K = g_K n^4$ .
9:     Almacenar los resultados de este paso en los arreglos
    $T[], V[], n[], m[], h[], G_{Na}[], G_K[], \tau_n[], \tau_m[], \tau_h[]$ 
10:     $I_{ext} \leftarrow I_{ext}(t)$ 
11:    Calcular  $V_m(t + \Delta t)$ 
12:   end for
13:   Devolver los arreglos  $T[], V[], n[], m[], h[], G_{Na}[], G_K[], \tau_n[], \tau_m[], \tau_h[]$  con los re-
       sultados para los tiempos  $[0, T]$ 
  
```

Comenzamos con un valor inicial y a partir de ahí empleamos las ecuaciones, para calcular las tangentes, aproximamos a la curva con su tangente.

Para la función INTEGRADISPARO necesitamos cuatro valores de entrada que van a provocar diferentes comportamientos:

1. T nos indica durante cuánto tiempo queremos correr la simulación.
2. ΔT nos indica que tan finos queremos que sean los pasos recordemos que vamos a aproximar la función con segmentos de recta siguiendo la tangente. Si hacemos pasos demasiado pequeños nos vamos a tardar demasiado en hacer el cálculo.
3. V_0 es el voltaje inicial en donde empieza nuestra simulación donde estaba en nuestra neurona cuando empezamos a trabajar.

⁵Hodgkin-Huxley Simulation Using Euler's Method lo puedes encontrar en la siguiente liga <https://webpages.uidaho.edu/rwells/techdocs/Biological%20Signal%20Processing/Chapter%2003%20The%20Hodgkin-Huxley%20Model.pdf>

4. $I_{ext}(t)$ es la corriente externa, de qué magnitud fue el toque que le estamos dando en este momento al axón

A partir de los elementos iniciales proporcionados ya podemos calcular lo demás. Vamos a querer guardar lo que está ocurriendo para todos los tiempos desde cero hasta t en cada delta, y lo vamos a hacer en forma de arreglos donde en la posición [0], está la primera medición en $t=0$ y la posición t está la medición en el tiempo t . Vamos a tener toda una serie de puntos donde estamos guardando estos pasos para inicializarlo.

Sabemos que necesitamos es un primer valor a partir del cual vamos a calcular la tangente y vamos a ir aproximando lo demás, entonces para eso queríamos *el voltaje inicial*, como nosotros sabemos donde estaba en reposo nuestra célula originalmente, vamos a poder guardar ese voltaje como el primer valor para nuestra simulación. Ahora todos los demás elementos como α_s y β_s y etc. se pueden calcular si ya conocímos ese voltaje inicial, entonces a partir de este momento podemos repetir el mismo ciclo tantas veces como sea necesario para cubrir, el intervalo. Desde el tiempo inicial hasta el tiempo t , brincando de delta T en delta T . Entonces dado un voltaje vamos a calcular las diferentes α_s que son las que se median experimentalmente originalmente utilizando las ecuaciones de las curvas paramétricas ajustadas ?? (paso 5).

Ya calculadas las alfas y betas ahora si se puede calcular las τ , n^∞ , m^∞ , h^∞ (paso 6). Teniendo estas entonces podemos calcular las probabilidades para las compuertas n , m , h usando las ecuaciones en diferencias en forma matricial, estas matrices se pueden expresar como 2.13.

$$\begin{bmatrix} n(t + \Delta t) \\ m(t + \Delta t) \\ h(t + \Delta t) \end{bmatrix} = \begin{bmatrix} (1 - \Delta t/\tau_n) & 0 & 0 \\ 0 & (1 - \Delta t/\tau_m) & 0 \\ 0 & 0 & (1 - \Delta t/\tau_h) \end{bmatrix} \begin{bmatrix} n(t) \\ m(t) \\ h(t) \end{bmatrix} + \begin{bmatrix} (\Delta t/\tau_n)n_\infty \\ (\Delta t/\tau_m)m_\infty \\ (\Delta t/\tau_h)h_\infty \end{bmatrix}.$$

Figura 2.13 Forma matricial para el cálculo de las probabilidades de las compuertas n , m , h .

Ahora esta parte fue importante sobre todo por el asunto de las pérdidas numéricas, recordemos que la computadora tiene una representación en punto flotante, lo cual quiere decir que un número real no se puede representar en la computadora, esto es importante por el problema del truncamiento, cada vez que se truncan dígitos al momento de calcular estamos perdiendo precisión, por esto es importante la forma en la que se procede a hacer los cálculos porque puede que se llegue a resultados un tanto distintos a los deseados aunque los cálculos sean correctos.

Ya teniendo estos datos podemos fácilmente calcular las conductancias, simplemente sustituyendo los valores (paso 8).

Es bueno una vez que ya terminamos de calcular estos términos que se van a necesitar

2. Modelo de Hodgkin-Huxley

en la ecuación principal 2.2 que es la del voltaje de la membrana, ir almacenando en los *resultados temporales* dentro de nuestros arreglos en la casilla que les corresponda, para ese momento (paso 8). En el paso 9, es donde ya vamos a utilizar *la corriente externa* para meterla en la ecuación diferencial para el voltaje. Una vez que tengamos esto tenemos que repetirlo para cada paso, se está calculando el tiempo t, entonces conforme avance el tiempo, va a dar el siguiente valor del voltaje, ya teniendo el siguiente valor del voltaje se va a poder usar para el siguiente paso ("como valor inicial nuevo") y así nos vamos a seguir todo el tiempo. Terminando el tiempo asignado a la simulación, tenemos ya los arreglos llenos de los datos que nos van a poder permitir graficar, qué fue lo que sucedió en cada tiempo, con cada canal y es precisamente así que se puede graficar las mediciones presentadas en la sección anterior.

Información codificada en las dendritas

En esta última sección de este capítulo, se va a tratar el cómputo en las neuronas, que vamos a simplificar para poder modelar las neuronas artificiales. Ya vimos detalladamente que pasa a lo largo del axón de la neurona, se ha mencionado que son quienes conforman la región que recibe la información (disparos / pulsos). Estos mandados desde los botones de las terminales de las neuronas presinápticas y son las dendritas las encargadas de enviar estas señales hasta el soma de la neurona y hacer que estas señales químicas se transformen o no en impulsos eléctricos. Ahora la siguiente cuestión es ¿Podemos hacer que se dispare más de una vez (con el mismo estímulo)?.

Los disparos que produzca una neurona depende fuertemente de la interacción con sus neuronas vecinas. Recordemos que durante el período refractario lo que sucede, es que cuando se acumula información en las periferias del cuerpo de la neurona a través de las dendritas y de su cuerpo es porque; está recibiendo información de neuronas vecinas y eso va a hacer que dispare.

Algunas neuronas tienen el período de refractario muy pequeño y disparen frecuentemente, por las conexiones que tienen con sus vecinas, son neuronas que mandan frecuentemente información y a veces pueden lanzar una serie de pulsos muy seguidos, en otras ocasiones puede haber interrupciones entre estas series de pulsos (ver imagen 2.14, en caso de los seres humanos podemos referirnos a las neuronas motoras que están en constante recepción y transmisión de información).

Por otro lado, hay otras neuronas que son muchísimo más pasivas y disparen muy rara vez (neuronas que están a niveles más altos, como reconocimiento de olores, colores, imágenes). Y también nos encontramos con casos donde aparentemente una neurona no reacciona ante ningún estímulo, hasta que pasa un estímulo muy específico⁶ y está

⁶Neuronas individuales que forman conceptos abstractos, responden por ejemplo, al nombre de un ser humano. Es así como se descubrió la neurona 'Jennifer Aniston', que disparaba cada vez que el retrato de la actriz se mostraba a los sujetos. Estas neuronas, que responden ante la presentación de

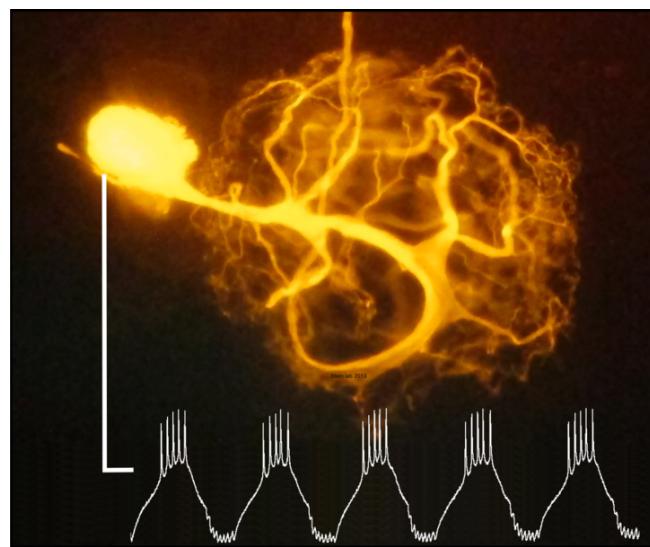


Figura 2.14 Disparos de una neurona, Wstein, 14 September 2013, Wikimedia Commons. Esta foto muestra la neurona de cangrejo que se tiñó mediante la inyección intracelular de un colorante fluorescente. El recuadro muestra una grabación de las oscilaciones rítmicas del potencial de membrana. Las grabaciones fueron realizadas por Christopher Goldsmith en el laboratorio de Wolfgang Stein en la Universidad Estatal de Illinois, https://upload.wikimedia.org/wikipedia/commons/c/ca/PD_neuron_staining_and_recording.png, CC BY-SA 3.0.

dispara (neuronas conceptuales o aisladas). La comunidad científica aún no se atreve a afirmar si este tipo de neurona solo dispara ante ese estímulo específico, o simplemente se trata de un evento que cumple con todas las condiciones para que esta neurona reaccione y mande un pulso. Cabe mencionar que estas neuronas se encuentran en las capas más altas de abstracción de procesamiento del cerebro⁷. Este tipo de diferencias entre neuronas te será más fácil recordar si regresas a la última sección del primer capítulo donde se encuentra un diagrama de los procesos de información en los que puede invertir una neurona, este diagrama es 1.5.

En este momento notamos la gran importancia de la codificación de la información en las redes neuronales de un cerebro. Donde la frecuencia de potenciales de acción nos muestran, ciertos patrones que nos indican el nivel de abstracción al que está respondiendo la neurona y el tipo de información que ayuda codificar o decodificar.

Teniendo ahora noción del papel que juegan los patrones de la frecuencia de disparos, se han hecho numerosas investigaciones acerca de estos. En un intento de entender mejor como es que un cerebro recibe la información desde el exterior, procesa y finalmente provoca ciertas reacciones ante el estímulo. En la siguiente imagen 2.15 que es el resultado

algunas imágenes recibieron la denominación de “células abuelas”.

⁷En la siguiente liga se puede encontrar información más detallada acerca de la investigación con estas neuronas: <https://www.nature.com/articles/s41598-020-64466-7>

2. Modelo de Hodgkin-Huxley

de un artículo donde se utilizaron diferentes métodos para simular disparos de neuronas y notamos la clasificación de estos patrones que se dan ante ciertos estímulos. Cada uno de estos patrones, están codificando cosas distintas, donde una misma neurona podría alternar entre diferentes patrones dependiendo de cuál es el estímulo que está recibiendo. Si bien ha habido algunas propuestas donde se tratan de hacer neuronas artificiales que tomen en cuenta la frecuencia de disparos aún hay un amplio campo de investigación en este tema.

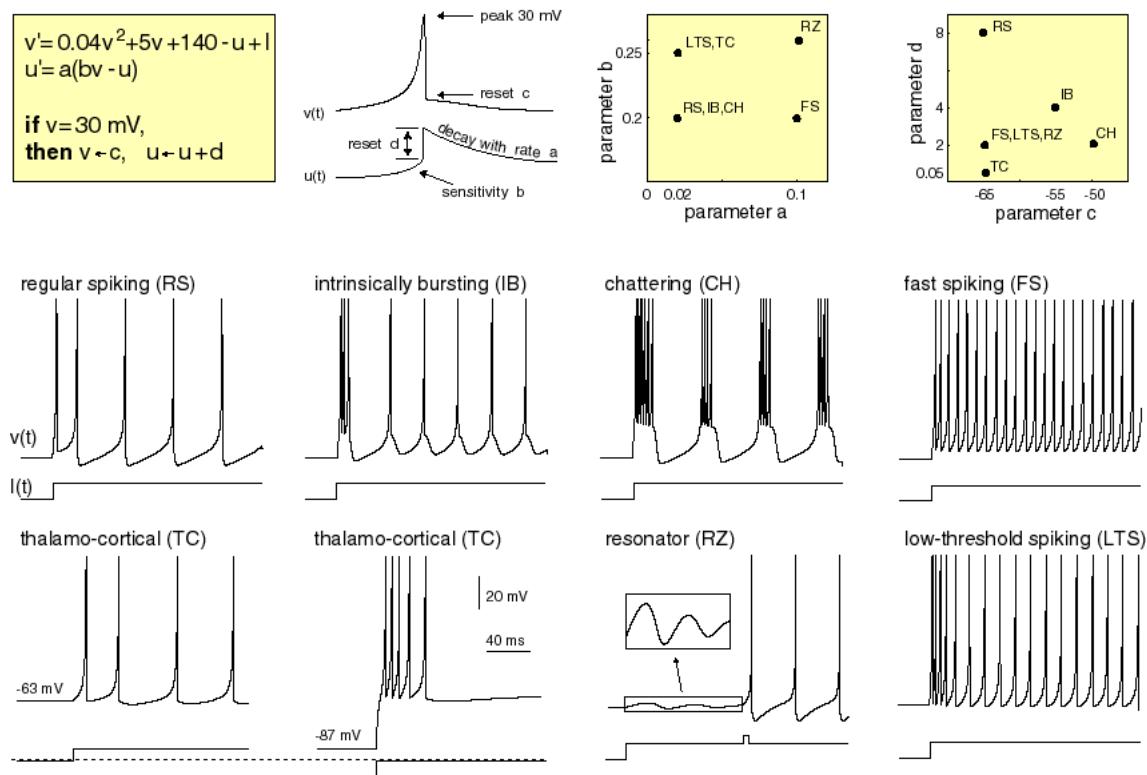


Figura 2.15 Diferentes patrones de disparo (simulados), Eugene M. Izhikevich, 2003, The Neurosciences Institute, <http://www.izhikevich.org/publications/spikes.html>

Por último mostremos la situación del sistema auditivo con la siguiente imagen 2.16:

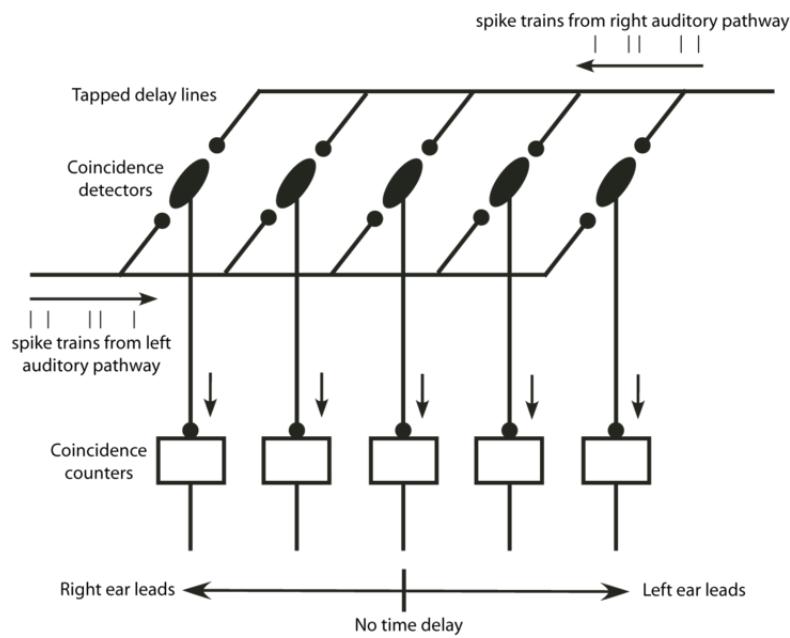


Figura 2.16 Localización del sonido por disparidad, Ian Stevenson, 7 enero 2008, El sistema auditivo registra y analiza pequeñas diferencias en el tiempo de llegada de los sonidos a los dos oídos para estimar la dirección desde la cual el sonido es emitido, <http://www.scholarpedia.org/w/images/4/4d/JeffressFig1.png>

La imagen muestra cómo computan las neuronas, este ejemplo se trata del sistema de audio humano. Cuando nosotros oímos podemos tratar de inferir aproximadamente desde donde está siendo emitido el sonido y eso se logra gracias a que tenemos neuronas conectadas al área auditiva (tanto con el oído derecho como con el oído izquierdo). La señal auditiva va a tardar más en llegar a un oído que al otro, dependiendo de su posición en el espacio, por lo que el cerebro siempre tratar de calcular ese desfase. En el esquema se representa como dentro del espacio de una persona se encuentra una fuente de sonido. Dependiendo de la dirección que provenga (izquierda o derecha), las neuronas receptoras de cada oído harán un cálculo rápido de la distancia a la que se percibió la señal, estás llegaran al cerebro y en algún punto enlazaran con neuronas donde estás cuentan las diferencias de distancias percibidas hasta llegar a dar con el resultado, dándonos a saber cuáles receptores (izquierda o derecha) están más cerca de la señal, y así ubiquemos desde donde proviene el sonido.

3 | Aprendizaje de máquina

Introducción

En este capítulo se desarrolla el procesos que pasa una maquina para que "aprenda", para esto notemos el concepto de aprender. En lo seres humanos se denota como el hecho de adquirir el conocimiento de algo mediante el estudio o la experiencia a partir de ejemplos específicos en nuestro medio, entonces aquellos problemas que inicialmente no pueden resolverse, puedan ser resueltos después de obtener más información acerca del problema. Desde pequeños empezamos por aprender por palabras (conceptos) que asociamos con algo específico, para después relacionar que varios objetos pertenecen a un tipo de conjunto y otros no. Como que un muñeco, una pelota y unos bloques de construcción de plastico, pertenecen a un conjunto denotado como "juguetes", y que plato, taza, y tazón no pertenecen a este conjunto sino al conjunto "vajilla". Entonces para organizar los conceptos que vamos aprendiendo hacemos uso de una función booleana, donde la entrada es el concepto, la pregunta es, ¿Este objeto pertenece a un cierto conjunto de objetos con características similares? y la salida es falso o verdadero. A este proceso, se le conoce como *aprendizaje de conceptos* y en este curso lo simplificamos como *función booleana de aproximación mediante ejemplos*.

Ahora lo que denotamos como el hecho que una maquina aprenda, lo vemos como cualquier programa que mejore su desempeño en alguna tarea mediante la experiencia. Con más formalidad se denota como:

Definición 3.1

Aprendizaje maquina: Se dice que un programa de computadora aprende, si su desempeño en T, medido por P , mejora con la experiencia E. Tal que:

- *T* es un tipo de tarea, como:
 - ★ Jugar un juego de mesa.
 - ★ Clasificar varios tipo de hojas.

- ★ Reconocer una voz en particular.
- *P* es una medida de desempeño, como:
 - ★ Porcentaje de juegos ganados en las partidas.
 - ★ Porcentaje de hojas correctamente clasificadas.
 - ★ Porcentaje de reconocimiento de timbre de la voz.
- *E* la experiencia con ejemplos (entrenamiento), como:
 - ★ Jugar juegos de práctica.
 - ★ Una secuencia de imágenes etiquetadas.
 - ★ Una secuencia de audios etiquetados.

, ^a.

^aMachine Learning, Mitchell 1997, pag. 14.

A partir de ahora nos dedicaremos a definir correctamente tareas que nos interese que un programa aprenda, para entender la forma más abstracta del problema y así proponer los algoritmos que nos ayuden a resolverlo.

Consideremos también que los sistemas de redes neuronales artificiales son un tipo de algoritmo para la representación del proceso de aprendizaje. Un problema de aprendizaje bien definido requiere una tarea bien especificada, medidas de desempeño y datos para obtener experiencia.

El aprendizaje maquina se apoya de disciplinas, como la inteligencia artificial, probabilidad, estadística, complejidad computacional, psicología, neurobiología y filosofía.

Para proponer un algoritmos de aprendizaje automático necesitamos, elegir el tipo de experiencia de entrenamiento, definir la función objetivo a aprender y un algoritmo para aprender la función objeto a partir de ejemplos de entrenamiento.

Los algoritmos de aprendizaje maquina han sido utilizados ampliamente por la industria bancaria, por gobiernos y por su puesto por el área de la salud. En la industria bancaria por ejemplo, donde es necesario aprender las reglas generales para determinar la solvencia crediticia, a partir de las bases de datos. Por los gobiernos para el reconocimiento de rostros humanos a partir de imágenes. En el área de salud para a partir de bases de datos de pacientes descubrir automáticamente regularidades implícitas en los resultados de tratamientos.

Espacio de Hipótesis

El aprendizaje automático, es utilizar datos disponibles para, aprender una tarea mediante una función que mejor mapee entradas a ciertas salidas. A esto se le llama aproximación de función, en el que aproximamos una función de destino desconocida (que suponemos que existe) que puede asignar mejor las entradas a las salidas en todas las observaciones posibles del dominio del problema.

Una función de un modelo que se aproxima a la función objetivo y realiza asignaciones de entradas a salidas se denomina hipótesis.

Ahora estas funciones pueden tener formas muy generales en el aprendizaje de máquina pueden tener forma, por ejemplo, de estructuras de datos, como los árboles de decisión, donde cada nodo pregunta si o no, pertenece una clasificación. pueden ser también funciones matemáticas como el caso de las redes neuronales, entonces la forma que tomen estas hipótesis en general puede abarcar muchos métodos y estructuras.

Entonces el aprendizaje consiste en, explorar un espacio de posibles hipótesis para encontrar la hipótesis (una función) que mejor encaje, de acuerdo a lo se obtuvo en los ejemplos de entrenamiento, y predecir alguna característica de salida deseada. Usualmente se denotan como sigue:

- h (hipótesis): una sola hipótesis, por ejemplo una instancia o modelo candidato específico que asigna entradas a salidas, se puede evaluar y se usa para hacer predicciones.
- H (conjunto de hipótesis): Un espacio de posibles hipótesis para mapear entradas.

Una breve ejemplo para denotar un espacio de hipótesis sería el problema es saber los días que nos conviene ir al cine, donde nuestra tarea T es aprender a predecir el conjunto de días que nos conviene ir al cine, basado en los atributos de los días, donde cada hipótesis la representaremos apartir de un conjunto de atributos de las instancias (días), entonces cada hipótesis es un vector con tres atributos, *tiene2x1, esEstrenoDePelícula, actoresConocidos*. Para cada atributo de la hipótesis tendría uno de los siguientes valores; Si, No, ?. Donde ? indica que cualquier valor es valido para ese atributo.

Cuando alguna instancia x cumpla con todos los atributos de una h , entonces $h(x) = 1$ y x es un ejemplo positivo. Entonces para representar la hipótesis, que nos conviene ir solo los días con 2x1, y que hay películas donde los actores son conocidos, la escribimos como $h(<\text{Si}, ?, \text{Si}>) = 1$, la hipótesis que cualquier día nos conviene ir al cine la denotamos como $h(<?, ?, ?>) = 1$, nuestra función objetivo la denotamos como una función booleana $c : X \rightarrow \{0, 1\} | X$, el conjunto de los 365 días del año, entonces $c(x) = 1$ cuando en los datos nos dicen que con la instancia x conviene ir al cine, $c(x) = 0$ en caso que no. Por tanto para aprender la tarea T , necesitamos *una hipótesis h en H tal que $h(x) = c(x)$*

para todas las x en X . La tarea de aprendizaje del concepto c requiere aprender el conjunto de instancias que lo satisface, describiendo este conjunto mediante una conjunción de restricciones sobre los atributos de la instancia.

Estas hipótesis (funciones) pueden llegar a ser sumamente complejas y tener que mapear datos de entrada con muchas formas ej. imágenes, trayectorias, etc. En el caso de las redes neuronales, el espacio de hipótesis está determinado por la arquitectura de la red. Vamos a definir el espacio de hipótesis, cuando decidimos qué neuronas vamos a poner en nuestro sistema, como las conectamos entre sí y cómo van a transferirse información de una a la otra y cuántas neuronas van a ser. Lo que veremos a lo largo del curso son diferentes arquitecturas y el impacto que tiene hacer diferentes modificaciones así como las matemáticas que existen detrás de estas.

Clasificación de los conjuntos de datos

La experiencia E para aprender la vamos a obtener mediante un conjunto datos, llamados datos de entrenamiento, estos se separan en tres bloques:

- **Entrenamiento:** Datos con los cuales se ajustan los parámetros de la hipótesis (del 50 % al 80 % de los datos). En este bloque se escoje que función del espacio fue mejor para el aprendizaje.
- **Validación:** Datos utilizados para ajustar los parámetros (hiperparametros) del algoritmo de entrenamiento, que puedan afectar qué hipótesis es seleccionada (del 25 % al 10 % de los datos y no deben ser usado durante el entrenamiento). Un ejemplo de un hiperparámetro para redes neuronales son el número de nodos ocultos en cada capa.
- **Prueba:** Datos utilizados para evaluar la posibilidad de que la hipótesis aprendida generalice¹ a datos no vistos anteriormente. Esta porción que se mantiene aparte. Con estos se evalúa el modelo, se reporta la eficacia del modelo según los resultados en este conjunto (del 25 % al 10 % de los datos).

El conjunto de datos de entrenamiento se usa para aprender una hipótesis y el conjunto de datos de prueba para evaluarla.

¹Se desea que nuestro modelo de aprendizaje, una vez entrenado con datos que ya hemos visto, se pueda usar con datos nuevos. Para ello debemos asegurarnos que el modelo no ha simplemente memorizado las muestras de entrenamiento, sino que ha aprendido propiedades del conjunto.

Tipos de aprendizaje

Aprendizaje Supervisado, el modelo usa datos etiquetados a una respuesta específica(labeled data), durante el entrenamiento se intenta encontrar una función que aprenda a asignar los datos de entrada (input data) con los datos en el etiquetado. Para después predecir una relación, dado un dato totalmente nuevo para el modelo. Los modelos pueden ser:

- Regresión: Un modelo de regresión busca predecir valores de salida continuos. Por ejemplo, en predicciones meteorológicas, de expectativa de vida, de crecimiento de población.
- Clasificación: En un problema de clasificación se desea predecir una salida discreta. Por ejemplo, identificación de dígitos, diagnósticos.

Aprendizaje no supervisado, es usado cuando no se tienen datos “etiquetados” para el entrenamiento. Solo sabemos los datos de entrada. Por tanto, únicamente podemos describir la estructura de los datos, para intentar encontrar algún tipo de organización que simplifique un análisis. Por ello, no se tienen valores correctos o incorrectos (es utilizado para aprender de una manera autoorganizada).

Aprendizaje por refuerzo, inspirado en la psicología conductista; donde el modelo aprende por sí solo el comportamiento a seguir basándose en *recompensas y penalizaciones*. Este tipo de aprendizaje se basa en mejorar la respuesta del modelo usando un proceso de retroalimentación (*feedback*). Su información de entrada es el feedback que obtiene del mundo exterior como respuesta a sus acciones. Aprende a base de ensayo-error.

Mientras que el aprendizaje supervisado y el no supervisado aprenden a partir de datos obtenidos en el pasado, el aprendizaje por refuerzo aprende desde cero, es decir, con un estado inicial y su ambiente, va aprendiendo a futuro, mediante posibles penalizaciones o recompensas. El *aprendizaje por refuerzo* es usado en videojuegos porque cada vez que se realizan las acciones correctas se ganan puntos y entonces se entrena a la gente para que pueda conseguir la mayor cantidad de puntos. En este siempre hay: un agente, un ambiente definido por estados, acciones que el agente lleva a cabo (que le llevan de un estado a otro), y recompensas o penalizaciones que el agente obtiene.

En cada acción, el agente solo conoce el estado en el cual se encuentra y las acciones posibles que puede elegir a partir de ese estado. No sabe si llegando al siguiente estado, obtendrá mejores o peores recompensas, irá aprendiendo en cada estado qué acciones lo llevarán a obtener una mayor recompensa a largo plazo, y qué el valor de las acciones en ese estado puedan subir. *Se enfoca en que el agente aprenda una política óptima para alcanzar el objetivo*. El agente siempre está en fases de *exploración* y *explotación*, en la fase de exploración el agente toma una acción de manera aleatoria, y en la de explotación va a tomar acciones basándose en cuán valiosa es realizar una acción a partir de un estado dado.

En plataforma de ventas en línea es donde podemos encontrar este tipo de modelo que están entrenados con este tipo de aprendizaje, donde al iniciar la sesión no conoce nada del usuario, solamente tiene un ambiente dado por los productos de la plataforma y su estado inicial es cero, para hacer individual la experiencia del usuario y que compre más. El algoritmo realiza la acción de mostrar ciertos productos (algún estado) si el usuario da clic a estos productos, el agente recibirá un punto de recompensa, por lo cual pasará a otro estado donde ofrecerá productos del mismo estilo donde pueda maximizar una venta, así se irá adaptando a cada usuario.

Parte II

Redes dirigidas acíclicas

4 | Perceptrón simple

Perceptrón

El perceptrón fue la primera red neuronal artificial (o ANS, Artificial Neural Systems) descrita algorítmicamente. En las décadas de los 60's y 70's, lo popularizó el psicólogo Franck Rosenblatt, en su libro llamado Principios de neurodinámica, donde presentó varios modelos de perceptrones, en el Laboratorio Aeronáutico de Cornell en Estados Unidos, originalmente estaba diseñado para ser una máquina, en vez de un algoritmo. Estaba diseñado específicamente para el reconocimiento de imágenes donde, cada peso era un cable físico por pixel de entrada, este era una matriz de 200 x 200, conectados aleatoriamente a las "neuronas", las actualizaciones de los pesos se realizaron mediante motores eléctricos.

El perceptrón es en sí, es la representación de una sola neurona, este se ocupa para la clasificación de patrones en un conjunto de datos multivariados, con ese se obtienen fronteras lineales en el plano, mediante un algoritmo de aprendizaje que veremos más adelante.

Recordando, una neurona es una célula elemental que a partir de un vector de entrada procedente del exterior o de otras neuronas (estímulo), proporciona una única respuesta (si activo el potencial de acción o no), ver figura 4.1. Los elementos que actúan en una neurona los podemos listar como:

- **Entradas:** $x_j(t)$. Las variables de entrada y salida pueden ser binarias (digitales) o continuas (analógicas) dependiendo del modelo de aplicación.
- **Pesos sinápticos:** w_{ij} . Representan la intensidad de interacción entre cada neurona presináptica j y la neurona postsináptica i .
- **Regla de propagación:** $h_i(t) = \sigma(w_{ij}x_j(t))$. Proporciona el valor del potencial postsináptico, de la neurona i en función de sus pesos y entradas.
 - * $h_i(t) = \sum_{j=0}^n w_{ij}x_j$, Es una suma ponderada de las entradas con los pesos sinápticos. Así, si la entrada es positiva, dependiendo de los pesos podemos

4. Perceptrón simple

saber si fue una sinapsis excitadora (pesos positivos) o inhibidora (pesos negativos).

- **Función de activación o de transferencia:** $a_i(t)$ Proporciona el estado de activación actual, de la neurona i en función de su estado anterior, $a_i(t - 1)$ y de su potencial postsináptico actual.

- $\star a_i(t) = f_i(a_i(t - 1), h_i(t))$, es la que usualmente se usa.
- $\star a_i(t) = f_i(h_i(t))$, en algunos modelos solo se considera que el estado actual no depende del tiempo anterior.

- **Función de salida:** $F_i(a_i(t))$ Da la salida actual, $y_i(t)$, de la neurona i en función de su estado de activación actual. El estado de activación de la neurona se considera como la propia salida.

- $\star y_i(t) = F_i(a_i(t))$
- $\star y_i(t) = F_i(f_i(a_i(t - 1), \sigma(w_{ij}, x_j(t))))$

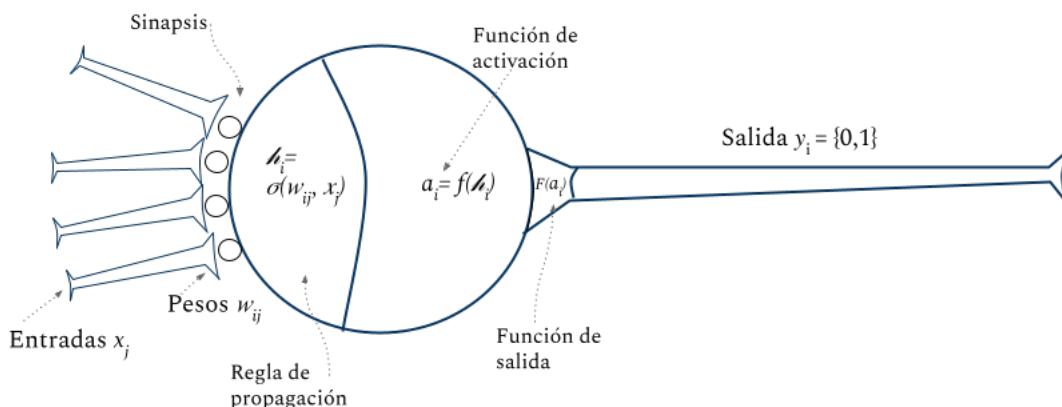


Figura 4.1 Neurona vista como un modelo artificial (perceptrón).

Un perceptrón toma un vector de entradas de números reales, calcula una combinación lineal de estas entradas, luego emite un 1 si el resultado es mayor que algún umbral y -1 de lo contrario. Es decir, dadas las entradas $x_1 \dots x_n$, la salida $o(x_1, \dots, x_n)$ calculada por el perceptrón es 1 si $w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n > 0$ y -1 de lo contrario, donde cada w es una constante \mathbb{R} , un peso, que determina la contribución de la entrada x a la salida del perceptrón. La constante w_0 es un *umbral* (bias) que la suma de las entradas con los pesos debe superar para que el perceptrón emita un 1. En otras palabras es un peso que va a actuar junto con una entrada de valor 1, que vamos a poder ajustar para que nuestra función de activación se mueva de derecha a izquierda en el plano para ayudarnos a ajustar nuestros resultados, provocando un gran impacto en el aprendizaje.

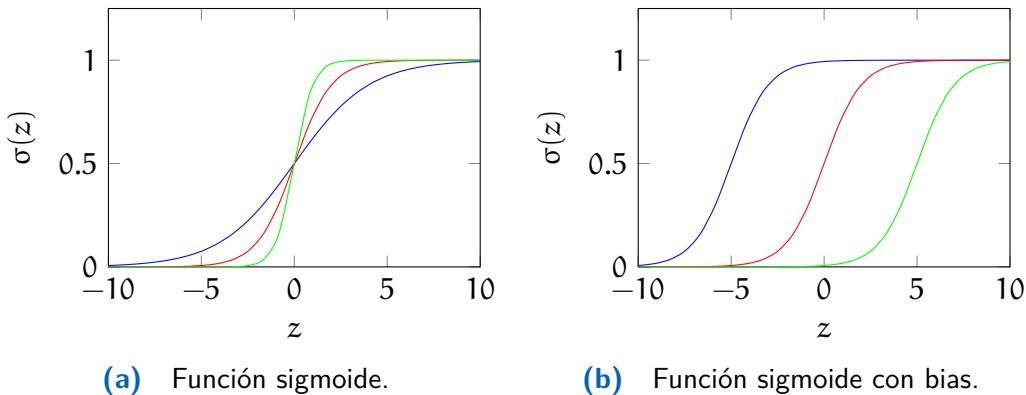


Figura 4.2 Comportamiento de la función de activación (sigmoide) de un perceptrón con una sola entrada, (b) el perceptrón sin el uso de bias, (b) con el uso del bias, donde apesar de estar representandos con la misma entrada , el uso del bias afecta en los resultados de salida. Así si quisieramos que este perceptrón nos diera $y = 0$ con una entrada $x = 2$ sin el uso, ni ajuste del bias sería imposible, pues en (a) apesar que la gráfica azul está la entrada está ajustada con el peso $w = 0.5$, la roja con el $w = 1$, y la verde con el $w = 2$ solo lo logramos alargarla un poco, haciendo que entradas que antes eran correctas ahora caigan 0 también. Entonces lo que necesitamos es más bien "mover" la gráfica, esto lo logramos con la gráfica (b) donde la entrada (única) está sumada con un bias (umbral) $x_0 = 1$, ajustado en azul con peso $w_0 = 5$, en rojo con $w_0 = 0$, en verde con $w_0 = -5$ y el peso $w_1 = 1$. Donde con $w_0 = -5$ logramos nuestro objetivo de tener una salida $y = 0$ con $x = 2$. El bias nos permite mover la función fuera del origen.

Esto se muestra en las siguientes graficas 4.2b. Más adelante hablaremos de su regla de entrenamiento (Training rule).

El hecho de que un perceptrón aprenda implica elegir valores para los pesos denotados también por θ . Ahora, el espacio H de las hipótesis candidatas consideradas en el aprendizaje del perceptrón, es el conjunto de todos los posibles vectores de pesos.

$$H = \{w | w \in \mathbb{R}^{n+1}\}$$

Si bien en el momento que se publicó los logros con el modelo del percetrón las expectativas eran bastante altas, a medida de los años los científicos Marvin Minsky and Seymour Papert desestiman en gran medida los alcances que realmente se puede tener con el perceptrón, al mostrar¹ que no puede predecir operaciones lógicas que no sean linealmente separables, tal es el caso de la función XOR, que no es separable linealmente, siendo imposible que pueda aprender esta función. Esto y el gran costo que representaba procesar todos los elementos que implicaba el entrenamiento, causa que por un buen

¹En 1969 publican Marvin Minsky y Seymour Papert que perceptrones de una sola capa (simples) solo son capaces de aprender a distinguir patrones linealmente separables, en el libro "Perceptrons".

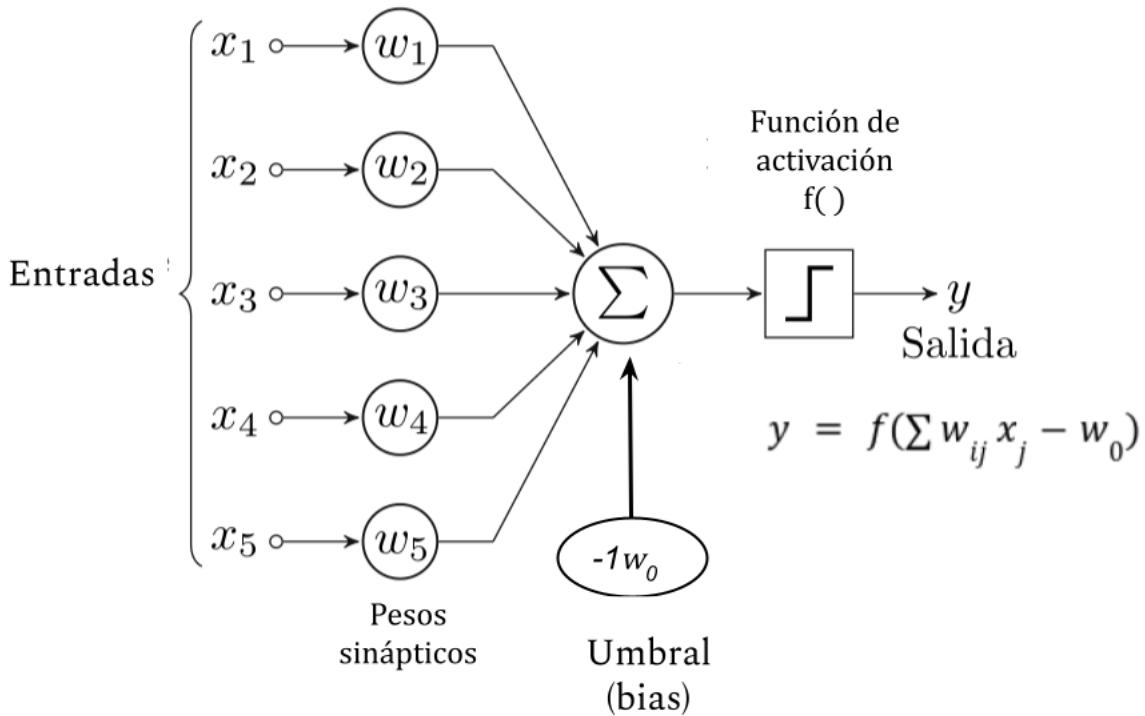


Figura 4.3 Modelo estandar de un perceptrón.

tiempo se desetime el uso del perceptrón. Es hasta después de unas decadas que vuelve a tener relevancia con la propuesta de un perceptrón multicapa usando retropagación (Feedforward), siendo estos capaces de resolver la función XOR

Compuertas lógicas con neuronas

Aquí se muestra como se puede utilizar un perceptrón para simular compuertas lógicas tales como el or, not, and.

Para simular la compuerta *not*, como está es una función booleana de $B \rightarrow B$, tal que $\text{not}(x) = -x$ entonces en un plano de dos dimensiones, la podemos representar con dos puntos, el $p_1 = (0, 0)$ y $p_2 = (1, 0)$ donde p_1 representa cuando $\text{not}(0) = 1$, p_1 representa cuando $\text{not}(0) = 1$. Teniendo el espacio de la función definido lo que nos toca es, separar el plano para clasificar las entradas, este claramente se puede separar con una linea vertical, o con líneas con pendiente 1 o -1 . Para esta función solo necesitamos de una entrada y un sesgo (bias), donde la entrada la combinaremos con un peso, este peso lo asignaremos a tanteo (por la sencillez de la operación). Así el peso $w_1 = -1$ y el peso asignado al bias sera $w_0 = 0.5$, ahora con esto datos podemos:

- Hacer la función de propagación donde $h(x) = (x * -1) + (0.5) * 1 = 0.5 - x$

- Hacer la función de activación escalón $a(x) = \text{sgn}(h) = \text{sgn}(0.5 - x)$
- Dar la salida donde $s(1) = \text{sgn}(0.5 - 1) = 0$ y $s(0) = \text{sgn}(0.5 - 0) = 1$, en este caso la salida es la identidad de la activación.

| x | h | s |
|---|------|---|
| 0 | 0.5 | 1 |
| 1 | -1.5 | 0 |

Algo similar va a pasar con la compuerta *and* y *or* donde al necesitar de dos entradas para la compuerta, asignaremos dos entradas para el perceptrón igualmente y las representaremos en el plano con cuatro puntos, donde cada punto representa una instancia y se le asigna valor positivo o negativo en el plano, así pues para el *and* tenemos los puntos $p_1 = (0, 0)$, $p_2 = (0, 1)$, $p_3 = (1, 0)$, negativos y $p_4 = (1, 1)$ el único positivo. Así nos damos cuenta que necesitamos una recta con pendiente negativa y fuera del origen, que nos separe estas clases de puntos. Por tanto para el bias le asignamos un peso de $w_0 = -1.5$, $w_1 = 1$ y $w_2 = 1$, con estos datos podemos:

- Hacer la función de propagación donde $h((x_1, x_2)) = (x_1 * 1) + (x_2 * 1) + (-1.5) * 1 = x_1 + x_2 - 1.5$
- Hacer la función de activación escalón $a((x_1, x_2)) = \text{sgn}(h) = \text{sgn}(x_1 + x_2 - 1.5)$
- Dar la salida donde $s = a(x)$, es la identidad de la activación.

| x_1 | x_2 | h | s |
|-------|-------|------|---|
| 0 | 0 | -1.5 | 0 |
| 0 | 1 | -0.5 | 0 |
| 1 | 0 | -0.5 | 0 |
| 1 | 1 | 0.5 | 1 |

Para la compuerta *or* es algo muy similar pues podemos igualmente representar la función con cuatro puntos en el espacio cada uno representando una instancia, solo que ahora tres de estos puntos serán positivos y solo uno negativo, los puentes positivos serían $p_2 = (0, 1)$, $p_3 = (1, 0)$ y $p_4 = (1, 1)$, mientras que $p_1 = (0, 0)$ negativo, el plano lo podemos dividir con una linea recta con pendiente negativa, así le asignamos $w_0 = -0.5$, $w_1 = 1$ y $w_2 = 1$, con esto hacemos los paso que ya sabemos:

- Hacer la función de propagación donde $h((x_1, x_2)) = (x_1 * 1) + (x_2 * 1) + (-0.5) * 1 = x_1 + x_2 + 0.5$
- Hacer la función de activación escalón $a((x_1, x_2)) = \text{sgn}(h) = \text{sgn}(x_1 + x_2 + -0.5)$

4. Perceptrón simple

- Dar la salida donde $s = a(x)$, es la identidad de la activación.

| x_1 | x_2 | h | s |
|-------|-------|------|-----|
| 0 | 0 | -0.5 | 0 |
| 0 | 1 | 0.5 | 1 |
| 1 | 0 | 0.5 | 1 |
| 1 | 1 | -1.5 | 1 |

Tomando el hecho que en la naturaleza las neuronas van pasando información en una estructura que forma niveles de abstracción, esto lo modelamos como capas de neuronas conectadas entre sí, cada capa haciendo su trabajo de abstracción. El perceptrón simple es un modelo neuronal unidireccional, una capa de entrada y otra de salida, que por si solo no puede separar todas las funciones lógicas pues tenemos el XOR, para resolver esto usaron perceptrones multicapa que se explicará más adelante en el curso.

Funciones de activación

Recordando que la forma de las funciones de activación es $y = f(x)$, donde x representa el potencial postsináptico e y el estado de activación de la neurona, es decir si va a lanzar un disparo o no. Las funciones de activación más empleadas son:

Funciones de error

Entonces tomando como base el perceptrón, una vez obtenidas las salidas con una primera iteración nos daremos cuenta de que tan lejos o que tan cerca estuvimos de la respuesta correcta, con esto darnos la oportunidad de que pesos ajustar respecto a sus entradas, en la segunda iteración. Ahora para facilitarnos esto y tomando en cuenta que el entrenamiento consiste en varias iteraciones hasta llegar a aprender la tarea T , hacemos uso de una función de error que nos ayude a minimizar la diferencia de error en las salidas.

Primero veamos el entrenamiento para una sola neurona, para esto haremos uso de la regla de aprendizaje del perceptrón (learning rule perceptron), donde para cada entrada, en la capa de salida se le calcula la desviación a la función objetivo. El cual utilizamos para ajustar los pesos del perceptrón (ver fig 4.5).

Usualmente al principio del entrenamiento se asignan pesos aleatorios, a medida que avance el entrenamiento, se van modificando con cada iteración, así $w_i \leftarrow w_i + \Delta w_i$. Esto con base a [la regla de aprendizaje](#) donde:

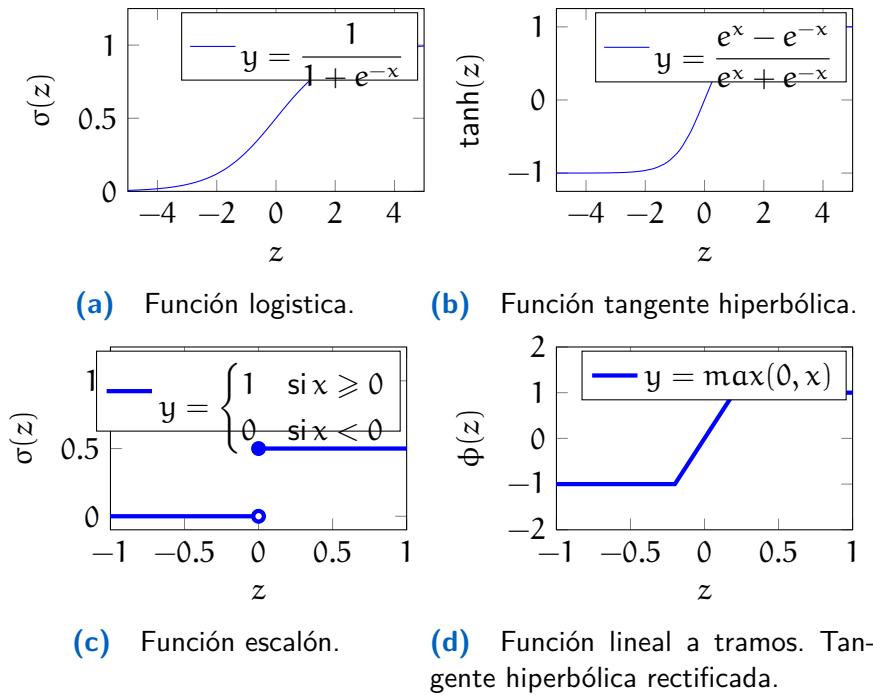


Figura 4.4 Las funciones de activación más usadas son la función sigmoide $\sigma(z)$ y la tangente hiperbólica $\tanh(z)$.

$$\Delta w_i = \alpha(y - y_{out})x_i \quad (4.1)$$

Con y es la salida deseada, y_{out} la salida generada, α la taza de aprendizaje (learning rate) y x_i la entrada i . Lo que hace la taza de aprendizaje es moderar el grado en que los pesos son cambiados con cada iteración, se le asigna un valor muy pequeño (0.1 o 0.2) y conforme se logran ajustar los pesos se minimiza aún más.

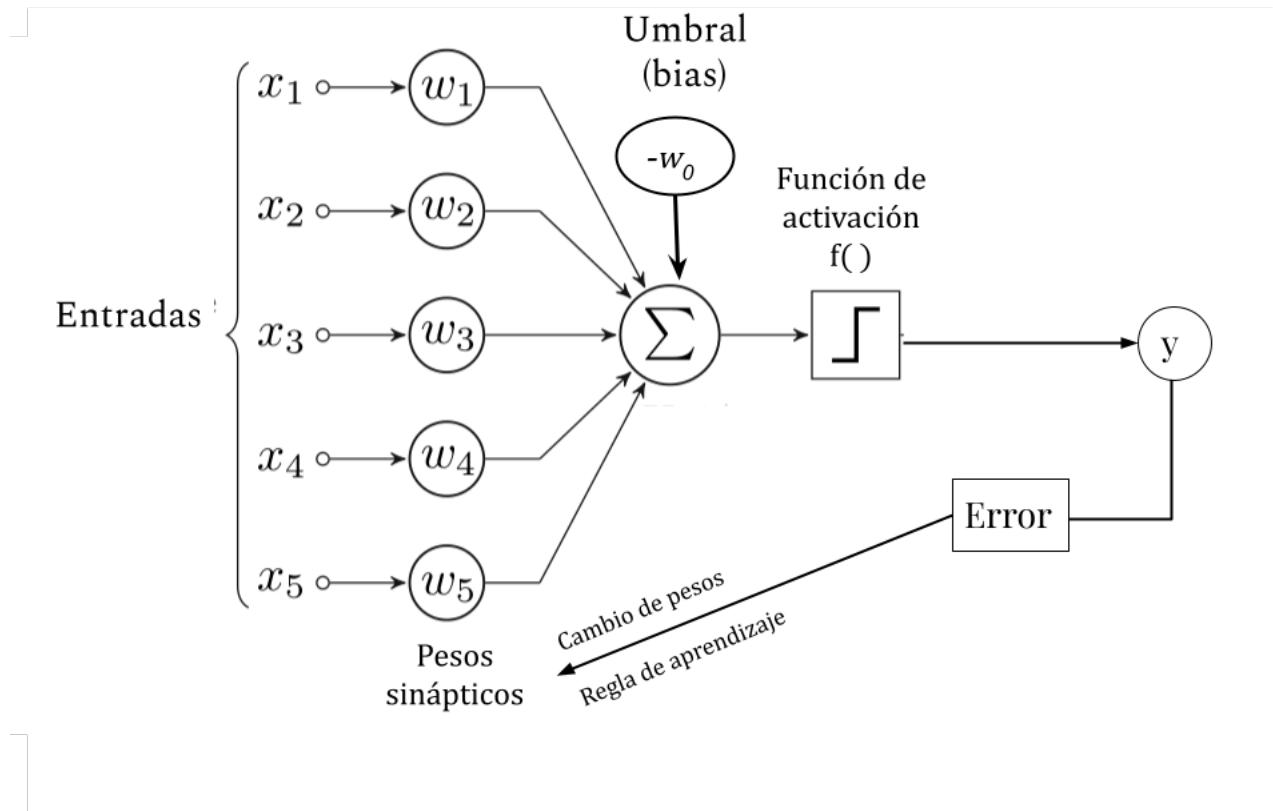
Para entrenar un perceptrón, utilizamos cualquier método de optimización de funciones para encontrar los parámetros w que minimizan el error con alguna de las siguientes funciones de error:

Diferencias al cuadrado, también conocida como regla aprendizaje delta, es la suma de cuadrados de errores que se tuvieron con cada ejemplar el el conjunto de entrenamiento. Los podemos describir como:

$$\frac{1}{2m} \sum_{m=0}^{M-1} (y_m - a_m)^2 \quad (4.2)$$

con y_m y a_m , la salida obtenida dado un ejemplar m y la salida correcta del ejemplar m respectivamente.

4. Perceptrón simple

**Figura 4.5** Modelo estandar de un perceptrón.

Entropía cruzada. Se usa para problemas de clasificación ya que su comportamiento es más suave (soft) y permite hacer clasificaciones más certeras. Se define como:

$$L(\Theta) = -\frac{1}{m} \sum_{m=1}^{m=0} (y^{(m)} \log(a_m) + (1 - y_m) \log(1 - a_m)) \quad (4.3)$$

Entonces juntando los conceptos que ya sabemos podemos describir el algoritmo de entrenamiento para el perceptrón de la siguiente forma:

1. Iteramos sobre todos los ejemplares.
2. Para cada ejemplar se calcula la función de propagación, es decir la suma ponderada.
3. Se calcula la función de activación con esta suma.

4. Se calcula la función de salida.
5. Actualización de pesos de acuerdo a la regla de aprendizaje.
6. Repetir de 1-6 hasta que los pesos nos satisfagan, un número de iteraciones establecidas.

Medidas de rendimiento

Las medidas de rendimiento de una red nos sirven para ver de manera concreta como se comportó nuestra red, durante el entrenamiento, que fue lo que pudo aprender, que sobre aprendio, es decir, memorizo, y que aún le cuesta trabajo aprender, por tanto, será incapaz de predecir. Se usan las siguientes:

Matriz de confusión : (Confusion matrix) Es una matriz, donde las celdas representan las predicciones que hizo nuestro modelo de clasificación de clases, respecto a las salidas esperadas. Así siendo las columnas las salidas y 's del modelo entrenado y las filas las salidas esperadas y_{true} . Nos facilita a ver cuando un clasificador está confundiendo clases, contabilizando a que clase etiqueto a los diferentes ejemplares. Ahora veamos, que puede estar representando cada celda en la matriz de confusión, estos pueden ser:

1. **VP** Verdaderos positivos (*TP, True Positive*): La clasificación de los ejemplares predichos, condicen con las etiquetas esperadas de los ejemplares.
2. **VN** Verdaderos negativos (*TN, True Negative*): El ejemplar que no es parte de clase, no son asignados a esa clase, son predichos correctamente.
3. **FP** Falso positivo (*FP, False Positivo*): El ejemplar que **no** es parte de una clase i fue clasificado como tal.
4. **FN** False negativo (*FN, False Negative*): El ejemplar que es parte de una clase i no fue clasificado como tal.

Ejemplo 4.1. Notemos un ejemplo sencillo, supongamos que tenemos una tarea binaria, donde queremos indicar que una persona está embarazada (de acuerdo a unos estudios). Ahora tenemos que:

- **VP**, sería predecir que una mujer está embarazada y que en efecto esté embarazada. (**Correcto**)
- **VN**, sería con un hombre que no está embarazado y pues en efecto no está embarazado. (**Correcto**)
- **FP**, sería predecir que un hombre está embarazado y **no** este embarazado. (**Error, tipo 1**)

4. Perceptrón simple

- ***FN***, sería predecir que una mujer ***no*** está embarazada y esta embarazada. (**Error, tipo 2**)

Entonces dados los resultados binarios que nos entregó el modelo, los médicos nos dan las respuestas correctas a 10 estudios, representadas en la siguiente tabla.

| Ejemplar | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------|----|----|----|----|----|----|----|----|----|----|
| Sujeto | M | F | M | F | M | M | F | F | F | M |
| Etiqueta | No | Si | No | Si | No | No | No | Si | No | No |
| Clase | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| Predicho | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| Valores | VN | FN | VN | VP | VN | FP | VN | VP | VN | VN |

Con estos datos, podemos construir nuestra matriz de confusión MC contabilizando las salidas obtenidas respecto a las deseadas. Quedando así de la siguiente manera:

| | | Salidas y | |
|-----------|----|-----------|--------|
| | | Si | No |
| Etiquetas | Si | VP = 2 | FN = 1 |
| | No | FP = 1 | VN = 6 |

Tabla 4.1 Matriz de confusión binaria.

Ahora para un modelo que sea multiclase, en el que tengamos que clasificar varias clases de ejemplares. Tendremos una matriz M de $n * n$ donde n es el número de clases, así los valores ***VP, VN, FP, FN***, son calculados para cada clase e identificados en la matriz M de la siguiente forma, también puedes ver la figura 4.6:

- ***VP***, para la clase i será la celda $M[i][j]$ con la $i = j$.
- ***VN***, para la clase i será la suma de los valores de toda la matriz menos los valores de la fila i ni los valores de la columna i.
- ***FN***, para la clase i será la suma de los valores en la fila i, excepto la celda $M[i][j]$, con $i = j$ que es el ***VP***.
- ***FP***, para la clase i será la suma de los valores en la columna i, excepto la celda $M[i][i]$ que es el ***VP***.

| Valores para la clase i | | Salidas y del clasificador | | | |
|-------------------------|---------|----------------------------|---------|---------|---------|
| | | clase 1 | clase 2 | clase i | clase n |
| Etiquetas ytrue | clase 1 | VN | VN | FP | VN |
| | clase 2 | VN | VN | FP | VN |
| | clase i | FN | FN | VP | FN |
| | clase n | VN | VN | FP | VN |

Figura 4.6 Matriz de confusión para clasificador multiclasses.

Ejemplo 4.2. Tenemos un clasificador encargado de identificar las cinco vocales del español, escritas a mano. Entonces tenemos un total de 5 clases representadas por a, e, i, o, u, cada letra representada por un número del 0 al 4. Así la clase 0 = a, la clase 1 = e y así respectivamente. Este fue entrenado con 15 ejemplares etiquetados y se obtuvieron los siguientes resultados:

| Ejemplar e | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Etiqueta | a | e | i | o | u | a | a | e | i | i | o | u | o | u | a |
| Clase y _{true} | 0 | 1 | 2 | 3 | 4 | 0 | 0 | 1 | 2 | 2 | 3 | 4 | 3 | 4 | 0 |
| Predicho y | 0 | 1 | 3 | 3 | 2 | 1 | 1 | 1 | 2 | 2 | 4 | 4 | 3 | 2 | 0 |

Ahora para hacer la matriz de confusión MC notamos que tenemos una matriz de 5x5, donde $MC[i] = \text{Predicho}$ y $MC[j] = \text{EtiquetasReales}$. Entonces necesitamos contabilizar las predicciones y asignarlas a sus respectivas celdas, donde con la tabla anterior notamos que en el ejemplar e3 con $y_{true}(e3) = 2$, se predijo que $y(e3) = 3$, entonces $MC[2][3] += 1$, pues la etiqueta nos posiciona en la fila 2 y lo predicho en la columna 3. Ahora en el ejemplar e4 con $y_{true}(e4) = 3$ se predijo que $y(e4) = 3$, entonces $MC[2][2] += 1$. Así con cada ejemplar vamos a ir sumando su clasificación. Para construirla tendríamos un pseudocódigo siguiente:

```
def getMC (Ejemplares, y):
    """
    :param y: salidas obtenidas (int)
    """
    MC = [len(y)] [len(Ejemplares)] # y == Ejemplares
    full_zeros(MC)
    for e in range (0, len(Ejemplares)):
        predicho = y[e]
        y_true = Ejemplares[e].etiqueta
```

4. Perceptrón simple

```
MC[predicho][y_true] += 1
return MC
```

Dados los datos anteriores tenemos, la siguiente matriz de confusión:

| | | Salidas y | | | | |
|-----------|---|-----------|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| Etiquetas | 1 | 2 | 2 | 0 | 0 | 0 |
| | 2 | 0 | 2 | 0 | 0 | 0 |
| | 3 | 0 | 0 | 2 | 1 | 0 |
| | 4 | 0 | 0 | 0 | 2 | 1 |
| | 5 | 0 | 0 | 2 | 0 | 1 |

Tabla 4.2 MC = Matriz de confusión multiclas.

Para calcular los valores *VP*, *VN*, *FP*, *FN*, por clase se propone el siguiente pseudo-código:

```
MC = getMC(y_salidas, y_true)

VP = VN = FP = FN = [0,0,0,0,0]

for i in range(len(MC)):
    for j in range(len(MC[0])):
        FN[i] = FN[i] + MC[i][j] if (i != j) else FN[i]
        FP[i] = FP[i] + MC[j][i] if (i != j) else FP[i]
        VP[i] = MC[i][j] if (i == j) else VP[i]
        VN[i] = sum(MC) - FN[i] - FP[i] - VP[i]
```

Si quisiéramos saber los valores *VP*, *VN*, *FP*, *FN*, para todo el desempeño total, simplemente sumamos lo obtenido en cada clase así:

```
VP_total = sum(VP)
VN_total = sum(VN)
FP_total = sum(FP)
FN_total = sum(FN)
```

Ahora los valores de cada celda nos representan lo siguiente:

- VP_{total} , toda la diagonal de la matriz. La salida del modelo coincide con lo etiquetado con el ejemplar.
- VN_{total} , todas las veces que no era la clase i y dijó que no era de la clase i.
- FN_{total} , todas las veces que era $clase_i$ y dijó que era $clase_x$. (deseado $FN = 0$)
- FP_{total} , todas las veces que predijo que era $clase_i$ y era $clase_x$. (deseado $FP = 0$)

Así que, si los valores que no están en la diagonal de la matriz son cero o todas nuestras clasificaciones están en la diagonal, podemos decir que nuestro modelo aprendió a clasificar correctamente todas las clases.

Precisión y recuperación : La precisión (*precision*) nos dice la proporción de ejemplares que se logró clasificar correctamente. Mientras que recuperación (*recall*) nos dice cuantas asignaciones de lo que nos interesa pudo clasificar correctamente en otras palabras donde del total de las respuestas correctas que se pueden tener, cuantas respuestas positivas acertadas se tuvo.

$$P = \frac{VP}{VP + FP} = \frac{\text{VerdaderosPositivos}}{\text{ValoresEsperados}} \quad (4.4)$$

$$R = \frac{VP}{VP + FN} = \frac{\text{VerdaderosPositivos}}{\text{ValoresPredichos}} \quad (4.5)$$

| Valores para la clase i | | Salidas y del clasificador | | | |
|-------------------------|---------|----------------------------|---------|---------|---------|
| | | clase 1 | clase 2 | clase i | clase n |
| Etiquetas ytrue | clase 1 | VN | VN | FP | VN |
| | clase 2 | VN | VN | FP | VN |
| | clase i | FN | FN | VP | FN |
| | clase n | VN | VN | FP | VN |

Recall = $VP / (VP+FN)$
Cuantos valores esperados fueron asignados realmente.

Precisión = $VP / (VP+FP)$
Cuento de lo que clasifico fue correcto

- Cuando el modelo detecta los ejemplares, pero los incluye en otras clases también: P es bajo y R es alto.
- Cuando el modelo no detectó bien los ejemplares, pero tampoco los incluyó en otras clases: P es alto y R es bajo.

4. Perceptrón simple

- Cuando el modelo detecta los ejemplares y no los incluye en otras clases: P y R es alto.
- Cuando el modelo no detecta los ejemplares: P y R es bajo.

En ocasiones le daremos más importancia al recall y en otras a la precisión. Retomando el ejemplo previo 4.1, no nos importa tanto los casos de error tipo 1, donde el modelo se equivoca con los ejemplares negativos, nos importa los errores de tipo 2, los Falsos Negativos, en este caso le daremos especial atención al Recall y que este sea alto. Pero si bien nuestra tarea es detectar cuando un correo es spam, no impacta tanto que aunque en ocasiones un correo sea spam, no lo clasifique como tal (error tipo 2 FN), nos es crucial que un correo que no sea spam nos lo clasifique como tal (error tipo 1 FP). Así deseando que los falsos positivos se acerquen a cero, dandonos como resultado una precisión alta aunque el recall sea bajo.

Exactitud y medida f : La exactitud (*accuracy*) es una medida de cuántas predicciones correctas en total hizo el modelo para el conjunto de datos completo (no se recomienda usar si tienes clases desbalanceadas, es decir, muchos elementos de una clase y poco de otra pues nos puede fallar totalmente con las clases pequeñas y aun así su valor sería alto). La medida f (*f score*) se utiliza para combinar las medidas de precisión y recall en un solo valor. Es práctico porque hace más fácil el poder comparar el rendimiento combinado de la precisión y la recall. Se dan las ecuaciones a continuación:

$$\text{Accuracy} = A = \frac{VP + VN}{VP + VN + FP + FN} = \frac{VP + VN}{\text{TodosLosValoresClasificados}} \quad (4.6)$$

$$F = \frac{2}{\frac{1}{P} + \frac{1}{R}} = 2 \frac{P * R}{P + R} \quad (4.7)$$

La medida f, también la podemos escribir (con un poco de aritmética) como:

$$F = \frac{2VP}{2VP + FP + FN} \quad (4.8)$$

5 | Perceptrón multicapa

Intro

Perdí el archivo :CCCC

XOR

Anteriormente, habíamos logrado separar clases de ejemplares siempre y cuando estos se pudieran modelar en un plano linealmente separable. Hasta ahora había sido un reto importante (no logrado) hacer clasificaciones de ejemplares distribuidos en un plano no separable linealmente, como lo es *la función XOR* donde, tenemos que las respuestas positivas se encuentran en una diagonal opuesta a las respuestas negativas y no hay manera de separar a los blancos de los negros con una sola frontera lineal, lo que hace imposible separar con un solo perceptrón. Entonces notamos que necesitamos de dos líneas que nos permitan separar el plano. Así llegamos a la idea que necesitamos más de una capa, que nos permita hacer la siguiente separación del plano.

| Entrada x_1 | Entrada x_2 | Salida y |
|---------------|---------------|------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Figura 5.1 Tabla de verdad para la función XOR.

5. Perceptrón multicapa

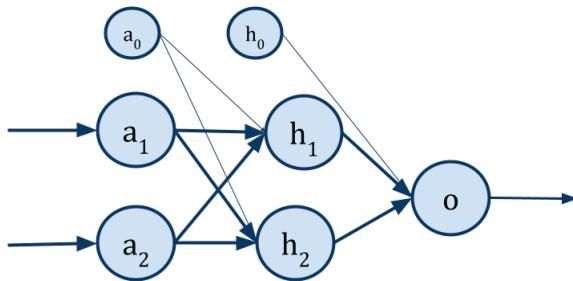


Figura 5.2 Perceptrón para la función XOR, con una capa oculta.

En la tabla de verdad notamos que para la función XOR, cuando la suma de las entradas es:

- par (tomando en cuenta al cero como par) la salida es 0.
- impar la salida es 1.

Para poder aprender la función únicamente tenemos que añadir una capa intermedia, la llamada capa oculta. Esta capa será la responsable de tomar las características de los vectores de entrada.

Ahora para la función tenemos dos entradas, una capa oculta (con dos neuronas) y una salida. Las neuronas en la capa oculta nos permitirán dividir el plano. Así, una solución sería que las neuronas ocultas denotadas por h (*hidden* de oculto en inglés) comparten pesos, la primera neurona h_1 haría la función OR, haciendo la distinción entre las entradas 00, la segunda neurona h_2 haría la función NAND haciendo posible distinguir el vector de entrada 11, una vez obtenido estos resultados de la capa oculta la neurona de salida o de output se encargara de distinguir la intersección entre estas, ejecutando la función AND. En resumen, la función **XOR** la podemos escribir como:

$$\begin{aligned} \text{XOR} &= (x_1 \vee x_2) \wedge \neg(x_1 \wedge x_2) \\ \text{XOR} &= \text{AND}(\text{OR}(x_1, x_2), \text{NAND}(x_1, x_2)) \end{aligned} \quad (5.1)$$

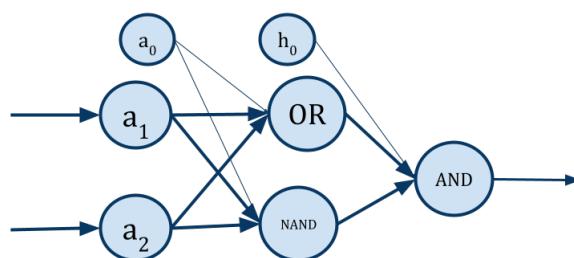


Figura 5.3 Solución para perceptrón de la función XOR.

Propagación hacia adelante manual

En esta parte vamos a ver como podemos evaluar la red para la solución que se dio en la sección anterior. (Recordemos que también se pueden dar otras soluciones para el XOR).

Entonces el procedimiento matemático general para poder evaluar una red cuando tenemos más de un perceptrón. Veamos primero que pasa con el **xor**, y el **nand**, por el momento no tomaremos en cuenta las neuronas de entrada a puesto que solo son usadas para almacenar las entradas. La capa oculta está formada realmente por dos percetrones, que dan su salida a una neurona más que es la capa de salida de nuestra red. Para calcular sus salidas debemos aplicar la suma ponderada de las entradas a una función activación, así podemos ver que la salida de una neurona oculta es la siguiente:

$$z_j = g(\sum_i i w_{ij} a_i) \quad (5.2)$$

A estos perceptrones a su vez se le están aplicando la función de activación sigmoide:

$$h_j = \frac{1}{1 + e^{-z}} \quad (5.3)$$

Así la neurona de salida recibe a los perceptrones ya evaluados y listos para aplicar pesos a estos y una función de activación, que en este caso son dados de la siguiente forma:

$$z_o = g(\sum_j j w_{jo} h_j) \quad (5.4)$$

$$o = \frac{1}{1 + e^{z_o}} \quad (5.5)$$

Así tomando de ejemplo la función XOR, los valores de la capa oculta son evaluados de la siguiente forma, donde x_1 y x_2 son evaluados por 0 o 1 según sea necesario:

$$\begin{aligned} h_0 &= 1 \\ h_1 &= g(1w_{01} + x_1 w_{11} + x_2 w_{21}) \\ h_2 &= g(1w_{02} + x_1 w_{12} + x_2 w_{22}) \end{aligned} \quad (5.6)$$

Entonces hasta aquí ya tenemos los valores de la capa oculta, una vez que ya tenemos este conjunto de valores podemos empezar a trabajar con el tercer perceptor, sus valores de entrada van a estar dados por los valores de activación de h_1 y h_2 y por un sesgo,

pues recordemos que es la función Nand, por tanto necesitamos movernos ligeramente del origen. Lo que vamos a tener aquí la fórmula se ve similar, lo único es que ahora los valores de entrada fueron los valores que obtuvimos en el cálculo de la capa anterior:

$$o = g(h_0w_{0o} + h_1w_{1o} + h_2w_{2o}) \quad (5.7)$$

Como estamos trabajando capa por capa, primero entran los valores con los que van a trabajar todos los perceptrones, después calculamos todos los de la capa oculta que son independientes entre sí, aunque tengan en común las mismas entradas y finalmente hacemos el cálculo de la siguiente capa, que en ese caso es la salida, pero bien podría ser otra oculta.

Por esto estamos hablando de un algoritmo llamado de *alimentación hacia adelante*, este hay que recorrerlo de izquierda hacia derecha, para ir obteniendo los diferentes valores de activación en cada una de las capas. Una característica de este tipo de arquitectura es que, al estar las neuronas conectadas, siempre se está conectando de las neuronas en la capa anterior hacia las neuronas en la capa siguiente. Cuando tenemos la evaluación de las capas ocultas, podemos obtener finalmente la salida final, con una evaluación final.

Si bien esta forma de evaluación nos lleva al resultado correcto, este método se puede simplificar sobretodo en el caso que estemos manejando más entradas y más perceptrones en las capas ocultas. Así nos daremos cuenta que es posible usar matrices, esta forma de evaluación la veremos en la siguiente sección.

Propagación hacia adelante vectorizada (con matrices)

La sección anterior si bien nos da la idea somera de como van a ser las operaciones para el aprendizaje ahora veamos lo de forma matricial. Esto nos ayudará a en un momento dado escribirlo en el lenguaje de nuestra convención.

Tomemos en cuenta de nuevo a la red neuronal de la sección anterior y recordamos que tenemos nuestros valores de entrada 0 y hoy vamos a imaginarnos exactamente como están escritos aquí y vamos a pensar que los metemos todos en un solo vector que va a entrar precisamente a la capa de entrada que en este momento la llamamos a y eso es exactamente lo que vamos a tener aquí pensemos en nuestras entradas como un vector ahora para poder hacer los siguientes cálculos vamos a acomodar ahora los pesos correspondientes a cada perceptor para el primer perceptor tomamos los pesos que están conectados desde la neurona de origen a 0 a 1 a 2 hacia la neurona oculta 1 por eso los índices 0 1 1 2 1 índice origen hacia índice destino y vamos a colocar todos esos en un renglón de la matriz de pesos aquí estaría la representación de nuestro primer perceptor para el segundo percepción vamos a tomar también su respectivo conjunto de pesos y le vamos a colocar en el segundo renglón de la matriz observen entonces que los valores que estoy tomando menos 0.5 1 y 1 que son éstos menos 0.5 uno y uno son términos que

había utilizado anteriormente para hacer mi primer cálculo bueno 0.51 y ahora pasó el segundo perceptor 1.511 1.5 menos 11 son también los pesos de las conexiones con este otro persona 1.5 menos uno y menos uno entonces ahí está cada renglón es un perceptor qué pasa ahora si tomamos las dos matrices y colocamos los pesos acá en nuestro valor de activación del lado derecho observen que entonces la forma de resolver la operación de multiplicación para matrices me dice que voy a multiplicar menos 0.5 por 1 uno por cero y uno por uno muy bien regresemos acá y recordemos que este siempre vale uno éste valía cero y éste valía uno entonces lo que tengo es exactamente la misma operación que tenía que arriba y lo mismo me va a pasar con los otros dos valores 1.5 menos son en menos uno pues aquí es el menos el 1.5 y los demás parece igualmente y entonces tenemos una misma entrada operando con los dos diferentes percepciones el resultado de haber hecho esta primera multiplicación me va a quedar en la parte de arriba que correspondería al valor de activación de mi primer perceptor en la cama oculta este por este me va a dar ahora al segundo perceptor que serían este y este recordemos entonces que este de acá el sesgo es por definición entonces por eso no lo tuvimos que introducir en nuestros cálculos para poder pasar a la siguiente etapa entonces lo que hacemos ahora es convertir este resultado de acá que es el vector que habíamos obtenido con esta multiplicación pero ya con la función de activación y ahora éste se va a convertir en nuestro nuevo vector de entrada entonces para obtener agua lo que haríamos sería lo siguiente habría que especificar ahora que estamos agregando el vector de sesgo le ponemos un 1 más tendríamos que obtener también los pesos y afectan a nuestro percepción en este caso copiamos este acá y hacemos la multiplicación y aplicamos nuestra función de activación y con eso ya podemos resolver la red observen que este formato nos puede funcionar en general para cualquier red y este le vamos a llamar la convención 1 básicamente la seguí para que fuera fácil visualizar la relación entre la red que tenemos a ver y la forma en la que estoy escribiendo los vectores el 10 1 y el 1 lo dejamos acá y vendría la siguiente en verdad la forma en que lo escribí ahorita solamente me va a funcionar cuando tengo un ejemplar que estoy alimentando ahora necesito modificarla un poco para poder trabajar con varios valores de entrada digamos que ya no nada más quiero evaluaré 0,1 digamos que quiero evaluar 0 0 0 1 1 0 y 1 existen varias formas equivalentes para hacerlo vamos a utilizar ahorita una que sea fácil de utilizar con la forma en que nos entregan las tablas de datos.

Lo que nos intriga ahora es saber que vamos a hacer en el caso que tengamos varias entradas a trabajar al mismo tiempo.

Existen varias opciones una opción sería tomar esta misma representación e inmediatamente comenzar a añadir cosas acá por ejemplo podríamos decir que tendríamos el 100 aquí el 110 acá el 111 desarrollar todas nuestras matemáticas ver cómo acomodar nuestras matrices y al final obtener un resultado que tenga todos los resultados sobre las columnas se puede hacer y seguramente encontrarán quien lo haga así por el momento vamos a seguir otra convención y este le voy a llamar la convención 2 lo que estamos haciendo en la convención 2 es más bien poner nuestro ejemplar de manera horizontal como renglón de ventaja va a tener esto ahora cuando yo quiera poner los demás ejemplares bueno como se va a ver recordemos que este que tenemos a la izquierda son nuestros

sesgos entonces va a ser una matriz con los primeros unos y después de eso voy a poder escribir mi tabla como estamos acostumbrados a verlos en circuitos digitales por ejemplo más adelante cuando estemos trabajando con redes neuronales sobre diferentes tipos de características como edades número de hijos tamaño de la casa etcétera etcétera entonces podremos llegar a tener un conjunto de datos donde hacia acá estamos enumerando todas las características estamos evaluando y así acá tenemos los ejemplares con los que vamos a trabajar muy bien entonces de hecho si ustedes les dieron una pequeña base en excel es precisamente así como se los van a dar en la parte de arriba está bien en los títulos que es lo que se está poniendo y es ahí abajo vienen los valores y que de renglón representa hay un empleado alguna estudiante etcétera etcétera entonces esta anotación es mucho más directa para la forma en que vamos a recibir la información y tal vez esté más fácil de interpretar qué está pasando con los renglones por eso vamos a utilizar esta versión muy bien entonces si ahora en nuestro ejemplar de entrenamiento viene de forma horizontal vamos a tener que trabajar al revés con los pesos cada percepción va a ser ahora una columna entonces ahora vemos colocados aquí los mismos pesos de los percepciones del ejercicio anterior bueno un ejemplo entonces vemos como esta matriz es en realidad la matriz transpuesta de la versión anterior entonces si quieren pasar entre una versión y otra en darnos hay que tener cuidado con las transpuestas si coloquemos así ahora los pesos para poder hacer la multiplicación también vamos a necesitar poner al revés los valores de entrada y los pesos ahora la a nos está quedando del lado izquierdo y los pesos del lado derecho ojo esta doble voy a estar no son lo mismo que relaciona y entre ésta y ésta de tres tandas de dibujar esto ahora porque aquí tenemos doble uva y aquí tenemos a w algo que les va a servir mucho cuando encuentren diferentes textos con diferentes convenciones es recordar esta regla matemática a transpuesta de respuesta es igual transpuesta muy bien entonces vamos a seguirlo de esta manera vamos a ponerla a la izquierda los pesos a la derecha pesos sobre las columnas y podremos repetir otra vez la notación la diferencia es que en esta ocasión los resultados ahora nos van a quedar sobre los renglones prueben hacer esto con un pipe y resuelvan lo para todas las posibles entradas del short y verán que pueden obtener acá al final su tablita con los resultados correctos no olviden cada vez que cambien de capa que a la izquierda hay que agregar los sesgos aquí tenemos entonces ya cómo se vería el final de la operación es poner ahora los resultados de activación de h aquí ahorita nada más está un ejemplar si quieren poner los demás solamente se agregan como más renglones ahora aquí están los pesos para la conexión con el último perfecto y realizamos la misma operación y obtenemos aquí la salida para este valor y aquí tuviéramos otros renglones entonces nos saldrían aquí también los otros resultados resumiendo entonces esta es la forma en la que veríamos nuestros datos de entrada en las cuatro entradas para la compuerta digital al agregar los sesgos es equivalente a agregar aquí al inicio una columna de 12 entonces podemos sustituir pesos [Música] tenemos aquí entonces los valores de activación del lado izquierdo los pesos del lado derecho y al realizar las multiplicaciones correspondientes lo que estamos obteniendo aquí son los valores de activación para la capa oculta para calcular h es lo que tenemos aquí aquí todavía no tenemos los sesgos para dar el siguiente paso observen que es básicamente como repetir este ciclo comenzaríamos con las hachas siendo estos valores que están acá y luego habría que agregar sesgos calcular los pesos para la siguiente capa

hacer la multiplicación etc vamos a hacer un rápido análisis de qué fue lo que pasó con la función sort también en el mundo matemático que pasó entre capa y capa recuerden que mencionaba varias veces que ya no iban a ser las mismas entradas cuando se aplicará el último and que se hubiera estado en la primera capa.

Interpretación matemática del mapeo no lineal

En esta sección vamos a ver por qué agregar una capa más nos permite calcular funciones que no era posible calcular antes es decir no basta con tener más percepciones en realidad necesitamos tener percepciones que reciban como entrada la salida de percepciones anteriores y esto lo vamos a notar aquí por eso el selector es una función tan útil para poder explicar estos conceptos de entrada que teníamos al inicio de la función sort eran los valores 0 0 0 1 1 0 y 1 1 estos solamente los podría bueno puede utilizar perceptor únicamente para evaluar funciones que sean linealmente separables entonces cuando hemos aplicado la primera capa donde pudimos poner tantas operaciones como que hicimos lo que hicimos fue tomar estas entradas y mapear las a un conjunto distinto y esté ocurriendo en este nuevo conjunto

Observen que las entradas ahora son 01 11 11 10 es decir estas dos están duplicadas ya no tengo el 0 0 por eso es que se dice que trabajamos con un mapeo no lineal nuestra función sigmoide o nuestra función escalón transformaron nuestro espacio de manera que no tenemos ahora una relación 1 a 1 qué efecto va a tener. Aquí tenemos una vez más cuáles son las entradas originales las entradas que se obtuvieron es bueno salidas de las capas ocultas que van a hacer ahora las entradas para el último perceptor entonces veamos acá qué pasó con este cuadro aquí teníamos cuatro datos que no eran linealmente separables después de la primera capa fue como haberlo plegado a otro espacio de dos dimensiones, pero observen ahora que como que esto es bueno vamos a ver quién se mató a quién el 0 0 se convirtió en 0,1 es decir este para acá el 0,1 se convirtió en 1,1 este fue para acá el 1010 también se convirtió en 1,1 es otro de acá y el 11 se convirtió en el 10 está aquí entonces nuestros puntos negros como que giraron un poquito y quedaron aquí y nuestros dos puntos blancos quedaron mapeados uno encima del otro en el mismo punto de este lado.

Entonces ahora sí lo puedo separar con un plano qué pasa en medio de los dos cualquiera que lo logre es bueno entonces estos tres valores quedaron mapeados ahora a un nuevo espacio que por cierto tiene una dimensión solo hay un valor. Entonces el paso fundamental radicó en que pudimos mapear este espacio hacia un espacio nuevo donde si es posible separar a nuestros datos linealmente y sobre esto pues ya simplemente aplicamos lo que podía hacer cualquier perceptual matemáticamente ese es el poder que nos está añadiendo una capa de en medio y básicamente lo podemos generalizar ahora si a cualquier función. Es posible quitar los sesgos si utilizamos la función escalón como función de activación y definimos precisamente el menor o igual para la parte del escalón en esta ocasión vamos a obtener valores que están exactamente en este punto de salto

así es que vamos a tener que decir quién queda a la izquierda quien quiere a la derecha y entonces podemos hacer esto igual se necesitan dos capas, pero podemos quitarnos la parte de los sesgos porque nuestras fronteras si están pasando por ser un coma se hace solo por curiosidad bien aquí tenemos entonces ya esa interpretación vemos que los ceros negativos se van a tomar como ceros y solamente los positivos van a quedar como unos y automáticamente se puede hacer esta versión resumida.

Propagación hacia adelante multicapa

Ahora para el modelo de una red en general tenemos otra vez nuestra nuestra arquitectura base que va a ser una red en capas también se le conoce como el perceptor multicapa en esta primera versión tenemos la capa de entrada que decíamos es la capa de chocolate porque realmente no son percepciones solamente reciben sesgos si los vamos a ocupar y nuestros datos o características para nuestros ejemplares y aquí tenemos nuestros datos de entrada esto que estamos dibujando aquí de hecho es una función una función que viene de x y evaluado hacia digamos xy son funciones sin moldes podríamos pensar en los reales donde tendremos que llamar aquí por ejemplo m y queremos referirnos al número de neuronas en la capa de salida.

One hot encoding. Falta todo esto.

6 | Entrenamiento por retropropagación

Función de error

Gradiente de la función de error

Descenso por el gradiente

Otros algoritmos de optimización

7 | Optimización del entrenamiento

Problemas en redes profundas

Gradiente desvaneciente (o que explota)

Entrenamiento en línea vs en lotes

Normalización y normalización por lotes

Regularización

8 | Caso de análisis e interpretación

Red Hinton árbol familiar con numpy (entrenamiento)

Red Hinton árbol familiar con pytorch

9

Entrenamiento con genéticos

Algoritmos genéticos

Neuroevolución

Antecedentes: Aprendizaje por refuerzo en videojuegos

Arquitectura para estimar la función de recompensa

Entrenamiento

10 | Mapeos autoorganizados

Aprendizaje no supervisado

Mapeos auto-organizados

Kohonen

11 | Redes Neuronales Convolucionales

Convolución

Redes Convolucionales

Softmax

MNIST

Parte III

Redes con ciclos

12 | Redes Neuronales Recurrentes

Derivadas ordenadas

Retropropagación en el tiempo

Sistemas dinámicos y despliegue del grafo

Arquitectura recurrente universal

Función de error

Forzamiento del profesor

13 | Atención

14 | LSTM

15 | GRU

16 | Casos de análisis: etiquetado de palabras y conjugación de verbos

16. CASOS DE
ANÁLISIS: ETIQUETADO
DE PALABRAS Y
CONJUGACIÓN DE
VERBOS

Parte IV

Redes no dirigidas

17 | Redes de hopfield

Entrenamiento

18 | Máquinas de Boltzman

Entrenamiento

Partículas y partículas de fantasía

Máquinas de Boltzman Restringidas

19 | Redes adversarias

GANs

A | Ecuaciones diferenciales