Métodos de Aprendizaje Automático para decodificar información contenida en potenciales de acción neuronales

Tesis de Grado para la obtención
del título de Ingeniero en Informática
realizada por

Matías Leonel Egea

Dirigida por el Doctor Ingeniero
Sergio Lew

Facultad de Ingeniería
Universidad de Buenos Aires

Julio 2015

Métodos de Aprendizaje Automático para decodificar información contenida en potenciales de acción neuronales

Tesis de grado

Ingeniería en Informática

2015/7

Facultad de Ingeniería
Universidad de Buenos Aires

Julio 2015

Resumen

El sistema nervioso de los mamíferos está constituido por neuronas que se comunican emitiendo potenciales de acción. Cuanta información es posible rescatar de la actividad de una población de neuronas es un problema que aún no está definitivamente resuelto. Se sabe que las distintas áreas del cerebro codifican la información de maneras diferentes, con mayor o menor grado de redundancia e inmunidad a las perturbaciones y con diferentes topologías, lo que le confiere a cada área propiedades emergentes únicas.

El Área Tegmental Ventral (VTA) y la Corteza PreFrontal (PFC) son áreas claves para la toma de decisiones y el aprendizaje de conductas que conducen a la obtención de refuerzos apetitivos (comida). Mientras que la primera procesa información concerniente a la obtención y la predicción de refuerzos, la segunda está involucrada en el procesamiento de reglas. Dada la masiva inervación dopaminérgica de la primera sobre la segunda, este par de áreas cuenta con la información necesaria, tanto para aprender reglas de conducta que conllevan al refuerzo, como para eliminar aquellas que resultan superfluas para la tarea.

En este trabajo se estudió cuánta información acarrean los potenciales de acción *(spikes)* provenientes de las neuronas del Área Tegmental Ventral (VTA) y de la Corteza PreFrontal (PFC) del cerebro de la rata, cuando el animal ejecuta una tarea de discriminación auditiva. Se utilizaron diferentes métodos de aprendizaje automático para predecir, en base a los potenciales de acción emitidos, cual fué el estímulo auditivo previamente presentado. Se comparó el rendimiento de dichos algoritmos tomando en cuenta la cantidad de neuronas utilizadas para la predicción, la longitud de la ventana de tiempo en la que se observa la actividad y el momento de la observación.

Los resultados encontrados ayudan a definir el esquema de aprendizaje y clasificación más adecuado, cuando la decodificación de la información neuronal debe ser realizada de manera on-line y en electrónica portátil, con el objetivo de minimizar la invasividad del método.

Agradecimientos

En primer lugar deseo agradecer al Lic. Camilo Mininni, al Dr. Silvano Zanutto y al Instituto de Ingeniería Biomédica de la Facultad de Ingeniería de la Universidad de Buenos Aires por facilitar los datos de los registros electrofisiológicos, sobre los cuales pude realizar esta Tesis.

Considero necesario agradecer a la Universidad de Buenos Aires y en particular a la Facultad de Ingeniería, por brindarme la posibilidad de tener una educación gratuita, de calidad y de nivel mundial.

Por otro lado, quiero agradecerle a mi tutor, el Dr. Ing. Sergio Lew, quien supo guiarme durante todo el desarrollo de este trabajo. Me gustaría agradecerle especialmente por la disponibilidad, la velocidad en las respuestas, las correcciones, la dedicación y el tiempo invertido.

Agradezco a mis padres, Ana y Daniel, por la educación que me brindaron, por estar siempre que los necesité durante todo el transcurso de mi carrera y por brindarme una base sólida de conocimientos, que me permitió aprovechar y disfrutar de la instrucción universitaria.

Por último me gustaría agradecer a mi mujer, Elizabeth, quien me acompañó y apoyó incondicionalmente, día y noche, e hizo posible la finalización de esta Tesis de grado.

Índice

Resumen **iv**

Agradecimientos **v**

1 Introducción **1**

2 Métodos de clasificación **4**
2.1 Discriminante Lineal de Fisher 4
2.2 Bayes Naive 5
2.3 SVM 7
2.4 Random Forest 9

3 Set de datos **12**
3.1 Origen de los Datos 12
3.2 Selección de Features 13
3.2.1 Análisis de neurona única 14
3.2.2 Análisis poblacional 17

4 Resultados **20**
4.1 Evolución temporal 20
4.2 Variabilidad en la decodificación 23
4.2.1 Área Tegmental Ventral 24
4.2.2 Corteza PreFrontal 26
4.2.3 Conclusiones 28
4.3 Dependencia tamaño poblacional 29
4.3.1 Conclusión 35
4.4 Impacto al variar la longitud de ventana 36
4.4.1 Conclusión 37
4.5 Predicción del comportamiento 38
4.5.1 Conclusión 39

5 Conclusiones **40**

Apéndice A Set de datos **43**
A.1 Origen 43
A.1.1 Procedimientos experimentales 43
A.1.2 Animales 43
A.1.3 Manipulación Pre-quirúrgica 43
A.1.4 Dispositivos de fijación a la cabeza 43
A.1.5 Cirugía 44
A.1.6 Electrodos y adquisición de los datos 44
A.2 Formato y distribución 45
A.3 Observaciones y análisis 45

Apéndice B Configuración de los algoritmos **47**
B.1 LDA 47
B.2 Bayes-Naive 47
B.3 Análisis con SVM 48
B.4 Análisis con Random Forest 48

Bibliografía **49**

Glosario de palabras **53**

Índice de Figuras

2.1 Separación lineal de 2 clases. 5
2.2 Ejemplo de clasificación con Bayes Naive 6
2.3 Separación con margen. Los vectores p1 y p3 son los llamados *support vectors* 8
2.4 Proceso de clasificación de una muestra con Random Forest . 10

3.1 Matriz X y Vector Y 14
3.2 Matriz de confusión de una neurona LDA con las cuatro clases, donde se observa la poca cantidad de trials GOi y NOGOi, en comparación con las otras dos clases. 15
3.3 Performance de las neuronas VTA - Corridas individuales. Cada par X-Y corresponde a una neurona con su precisión de clasificación (media para las 500 corridas). - LDA 16
3.4 Performance de las neuronas PFC - Corridas individuales. Cada par X-Y corresponde a una neurona con su precisión de clasificación (media para las 500 corridas). - LDA 16
3.5 Figura a modo de ejemplo (datos ficticios) que ilustra un problema no linealmente separable si se considera cada neurona por separado, pero que se vuelve linealmente clasificable tomando como *feature* a cada neurona. 18
3.6 Matriz X y Vector Y - Análisis poblacional - $x_{i,j}$ representa la suma de todos los *spikes* para el *trial* i de la neurona j . . 19

4.1 División de la neurona en ventanas 20
4.2 Composición de la matriz X_w y el vector Y_w 21
4.3 LDA - Performance en función del tiempo. 22
4.4 Performance en la decodificación en función del tiempo para 500 re-muestreos en las neuronas VTA. a) Análisis de discriminante lineal. b) Análisis con Naive Bayes. c) Análisis con Support Vector Machine. d) Análisis con Random Forest. 24
4.5 Comparación para todas las neuronas VTA en barras a partir del milisegundo 5000 25

4.6 Significancia en la que la media de un método es mejor que la de otro a partir del milisegundo 5000 para las neuronas VTA. 25
4.7 Performance en la decodificación en función del tiempo para 500 re-muestreos en las neuronas PFC. a) Análisis de discriminante lineal. b) Análisis con Naive Bayes. c) Análisis con Support Vector Machine. d) Análisis con Random Forest. . . 26
4.8 Comparación apareada para todas las neuronas PFC en barras a partir del milisegundo 6500. 27
4.9 Comparación apareada para todas las neuronas PFC en barras entre los milisegundos 4100 y 4400. 27
4.10 Significancia en la que la media de un método es mejor que la de otro a partir del milisegundo 6500 para las neuronas PFC. 28
4.11 Gráficos performance/tiempo. Cada sub-gráfico corresponde a un porcentaje de neuronas: a) 100%, b) 75%, c) 50%, d) 25%, e) 10%, f) 5% 29
4.12 Gráfico de barras promediando la performance a partir de 5000ms. Se grafica la performance en función del porcentaje de neuronas y del método de clasificación utilizado. 30
4.13 Comparación para las neuronas VTA que su performance individual no supera en media al 70%. 31
4.14 Comparación para todas las neuronas VTA cuya performance individual no supera en media al 70%. Gráfico de barras a partir del milisegundo 5000. 32
4.15 Comparación para las neuronas PFC cuya performance individual no supera en media al 70%. 33
4.16 Comparación para todas las neuronas PFC cuya performance individual no supera en media al 70%. Gráfico de barras a partir del milisegundo 6500. 34
4.17 Significancia en la que la media de un método es mejor que la de otro a partir del milisegundo 6500 para las neuronas PFC, sin las cuatro neuronas que superan el 70% de performance en su clasificación individual. 34
4.18 Random Forest - Gráfico de la performance en función al tamaño de ventana, desde el milisegundo 4000 - 500 corridas . 36
4.19 Random Forest - Ventana de tamaño fijo en 300ms (Verde) vs Ventana ampliándose a medida que avanza el tiempo (Rojo). 37
4.20 Gráfico de la performance en función del tiempo separando las clases L (saca lengua) y NO-L (no saca lengua) para la ventana entre 4000ms y 4300ms 38
4.21 Significancia en el que la performance de un método es mayor a 0.5 para la ventana entre 4000ms y 4300ms 39

Introducción

El principal mecanismo por el cual la información se codifica, almacena y transmite en el sistema nervioso de diferentes especies (desde C. Elegans hasta el cerebro de los primates) es la generación de potenciales de acción. En este sentido, distintas áreas del cerebro de los mamíferos codifican la información de manera diferente. Un ejemplo claro de codificación de información visual se observa en la corteza visual primaria de los felinos y primates, en donde se registran neuronas cuya frecuencia de disparos de potenciales de acción aumenta, cuando se presentan estímulos visuales consistentes en lineas con una cierta orientación (Hubel y Wiesel, 1962). En otras áreas corticales y subcorticales los estímulos externos no impactan tan claramente sobre la actividad neuronal, aunque ciertos fenómenos comunes a la dinámica neuronal se mantienen (Churchland et al., 2010). Entender la forma en que los estímulos externos son codificados, procesados y mantenidos en memoria por lapsos de tiempo cortos, forma parte del estudio de lo que se conoce como código neuronal (Quiroga y Panzeri, 2009).

Más allá del hecho que entender estos mecanismos brindan la posibilidad de comprender el funcionamiento de la máquina más compleja del mundo, las aplicaciones prácticas de este conocimiento resultan de gran importancia en el área de la clínica médica. La utilidad inmediata de contar con técnicas eficientes de decodificación aportan a la neuroprostética, donde a través de las Interfaces Cerebro Computadora (Brain Computer Interfaces BCI) (Lebedev y Nicolelis, 2006) se demostró que es posible mover aparatos robóticos decodificando la "intención de movimiento del paciente". Con el registro simultáneo de la actividad neuronal de decenas de neuronas pertenecientes a la corteza premotora, fue posible entender el código neuronal contenido en esa población y entrenar decodificadores lineales para predecir las respuestas motoras (Aflalo et al., 2015). Tal vez el trabajo más impactante de los últimos años, sea el que conecta dos cerebros de ratas, para demostrar que se puede extraer la información táctil (vibrisas) proveniente de la corteza de barriles, decodificarla y luego re-inyectarla por medio de estimulación eléc-

trica a otro animal, creando así la primera interfaz cerebro a cerebro (Brain To Brain Interface BTBI).(Pais-Vieira et al., 2013)

Sin embargo, existen problemas de origen práctico que limitan la performace de la decodificación de información neuronal. Es fácil notar que cuanto mayor sea la densidad de electrodos implantados en el sistema nervioso, más neuronas se podrán registrar simultáneamente y tanto menor será el error de estimación cometido. No obstante, cuando se implantan electrodos profundos en el cerebro, el sistema inmune reacciona de diferentes maneras: una de ellas es rechazando el cuerpo extraño, mientras que otra es creando tejido degenerado alrededor de los electrodos, que contiene a las neuronas originales y que perturba el modo de disparo. En consecuencia, la opción de implantar miles de electrodos para acercarse a una estimación precisa de la información contenida, no parece ser una opción que, a largo plazo, pueda subsistir intacta dentro de un cerebro (Polikov et al., 2005). Por esta razón se torna imprescindible mejorar los métodos de decodificación, con el fin de minimizar la cantidad de electrodos implantados, manteniendo niveles de performance aceptables.

La aplicación de modelos probabilísticos para la decodificación de información contenida en los potenciales de acción, es la alternativa más utilizada actualmente (Brown et al., 2004). Estos métodos utilizan aprendizaje automático con el objetivo de extraer (si existiera) la información indispensable para la decodificación en un proceso de entrenamiento, para luego emplearlas durante el uso del dispositivo (Duda et al., 1973).

En esta tesis, se estudiará la información contenida en los trenes de potenciales de acción en registros del Área Tegmental Ventral (VTA) y la Corteza PreFrontal (PFC) del cerebro de la rata. Se hará foco en el estudio del área VTA, dado que se tienen mas neuronas registradas. Con este objetivo, se utilizarán diferentes Algoritmos estadísticos de aprendizaje automático para clasificar los datos registrados en cuatro animales, durante una tarea de discriminación auditiva. Se comparará el funcionamiento del Discriminante Lineal de Fisher (LDA), Bayes Naive, Máquinas de Soporte Vectorial (SVM) y Random Forest. En el capítulo 2 se explicarán estos cuatro métodos de clasificación.

En el capítulo 3 se describirá el proceso de obtención y procesamiento de los datos utilizados en el análisis. Se expondrá el proceso de selección de *features* (variables de entrada en los métodos de clasificación), y se explorará la posibilidad de realizar la clasificación utilizando solamente la información de neuronas individuales, es decir, en un análisis unidimensional.

En el capítulo 4, se analizarán distintos aspectos del problema:

- Evolución de la performance de clasificación a medida que transcurre el tiempo, tanto antes como después del estímulo.
- Comparación de los algoritmos de aprendizaje automático y determinación del más adecuado.
- Variación de la performance de clasificación en función del número de neuronas utilizadas.
- Determinación del tiempo mínimo de análisis para poder extraer información de los potenciales de acción.
- Predicción del comportamiento del animal, conociendo solamente su actividad neuronal.

Por último en el capítulo 5 se presentarán las conclusiones del presente trabajo.

Métodos de clasificación

En esta sección se describirán los aspectos fundamentales del funcionamiento de los cuatro métodos de clasificación con que se analizarán los datos: Discriminante Lineal de Fisher (LDA), Bayes Naive, Máquinas de Soporte Vectorial (SVM) y Random Forest.

2.1 Discriminante Lineal de Fisher

El método del discriminante lineal utilizado es una generalización del descripto originalmente por Fisher (Fisher, 1936). El mismo resulta óptimo en el caso de variables con distribución normal y permite encontrar el hiperplano que mejor separa las dos clases, siendo posible la generalización para un número mayor de clases (Rao, 1948).
En la figura 2.1 se ilustra el caso de dos poblaciónes diferentes (A y B) que se pretende separar con un hiperplano.

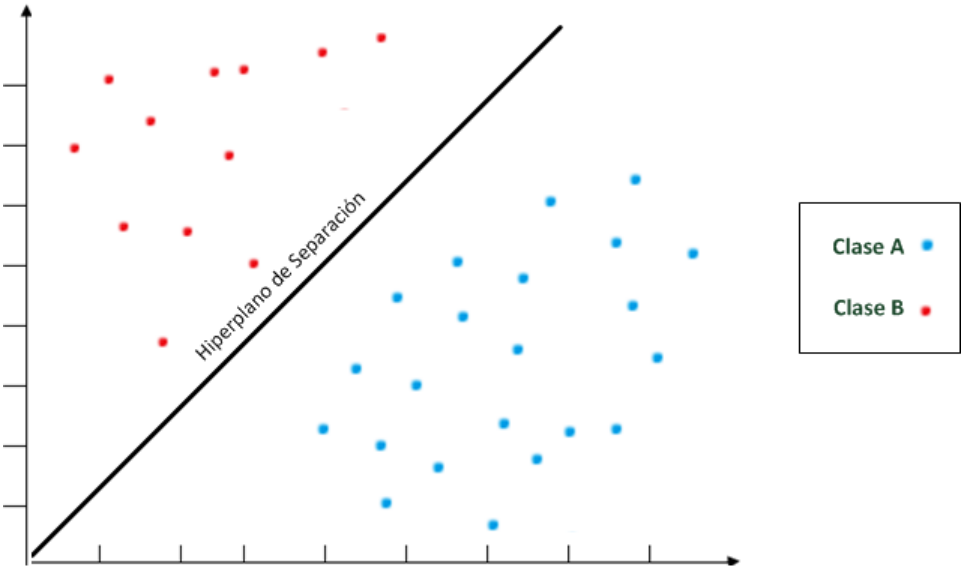

Figure 2.1: Separación lineal de 2 clases.

En el trabajo de Fisher se demuestra que el vector que determina al hiperplano que mejor separa las clases es:

$$w = (\mu_1 - \mu_2)(\Sigma_1 + \Sigma_2)^{-1} \tag{2.1}$$

siendo μ_i la media de la clase i y Σ_i la matriz de covarianza de dicha clase.
Uno de los problemas que surgen al utilizar esta herramienta para separar clases es la inversión de la suma de las Matrices de Covarianza. Cuando el número de muestras es pequeño (como en el caso de la cantidad de *trials* que un animal realiza en una sesión de entrenamiento), la inversión de dicha matriz suele estar mal condicionada. En estos casos, se puede utilizar la pseudoinversa de la matriz suma, dando como resultado una discriminación pseudolineal del problema (Liu et al., 2007). Dada la sencillez del procedimiento para encontrar el hiperplano que mejor separa dos clases, este método es ampliamente aplicado en gran variedad de señales de origen neuronal (Aggarwal et al., 2013; Velliste et al., 2014).

2.2 Bayes Naive

En este método se asume que las variables (*features*) son independientes entre sí. Por esta suposición que generalmente no se cumple pero que simplifica radicalmente el proceso de estimación, se considera al método "ingenuo"

(*naive*).
Bayes Naive utiliza una función de distribución a priori, el teorema de Bayes y la suposición previa para poder predecir la función de distribución a posteriori de las variables (Duda et al., 1973). Los mejores resultados con *Bayes Naive*, se obtienen cuando la dimensión del espacio de *features* es alta, debido a que la estimación de la función densidad de probabilidad se vuelve poco eficiente (Hastie et al., 2009). El método resulta rápido, incremental y apto para el trabajo con atributos discretos y continuos (Kononenko, 1993), y tiene excelente performance cuando es aplicado a datos provenientes de variables continuas y normalmente distribuidas. Sin embargo, su ingenuidad puede derivar en un bajo rendimiento cuando existen dependencias fuertes entre las variables que fueron originalmente consideradas como independientes.

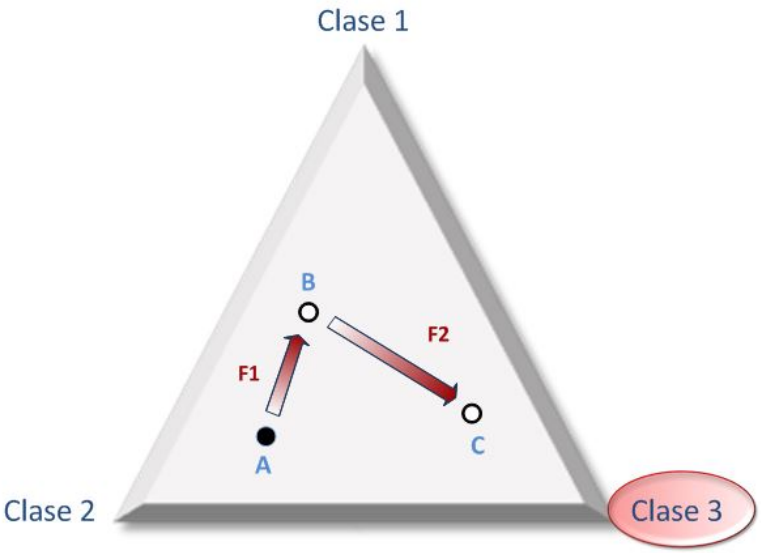

Figure 2.2: Ejemplo de clasificación con Bayes Naive

En la figura 2.2 se muestra una clasificación entre tres clases con Bayes Naive con el objetivo de explicar el funcionamiento del método.
Se tiene una muestra "m" que se quiere clasificar. En la misma se encuentran presentes dos *features* "F1" y "F2".
Al empezar a clasificar, se utiliza la probabilidad a priori. Generalmente es

determinada por la cantidad de apariciones de cada clase en el corpus de entrenamiento. El punto "A" es determinado por esta probabilidad. Luego se evalúan cada uno de los *features* de la muestra a clasificar. La aparición del *feature* "F1" indica una cierta probabilidad favorable a la clase 1, este *feature* mueve el resultado hacia el punto "B". La aparición del *feature* "F2" vuelve a correr la probabilidad, esta vez hacia el punto "C". Dado que en la muestra, los únicos dos *features* que se encontraron son "F1" y "F2", se da la clasificación por terminada. La Clase 3 es la que está mas cerca del punto "C", por lo tanto el resultado devuelto por el clasificador será que la muestra "m" pertenece a dicha clase.

Este método fue utilizado para clasificación de *spikes* en forma semiautomática en Harris et al. (2000) con buenos resultados, superando la clasificación manual. A su vez, en Hu et al. (2005) se lo utilizó para predecir la decisión de una rata clasificando los *spikes* de las neuronas del área cortical motora. En esa ocasión se lo combinó con un análisis de componentes principales para extraer las mejores *features* y de esta manera competir con la performance obtenida con clasificadores mas robustos.
Al igual que con el Clasificador Lineal de Fisher, este método es ampliamente utilizado debido a su velocidad de entrenamiento y clasificación.
Se pueden encontrar mas ejemplos de su aplicación en procesamiento de señales neuronales en Schmuker et al. (2014); Valenti et al. (2006)

2.3 SVM

El método *Support Vector Machine* (SVM) se basa en la idea de proyectar los datos en un espacio dimensional mayor, con el fin de encontrar un hiperplano que permita separar las clases. Las características del hiperplano aseguran una alta capacidad de generalización de este método de aprendizaje automático (Cortes y Vapnik, 1995).
Una vez encontrado el hiperplano, se busca un margen m que maximice el espacio entre las clases. Los vectores que se encuentran a distancia m del hiperplano son llamados *support vectors* (vectores de soporte).

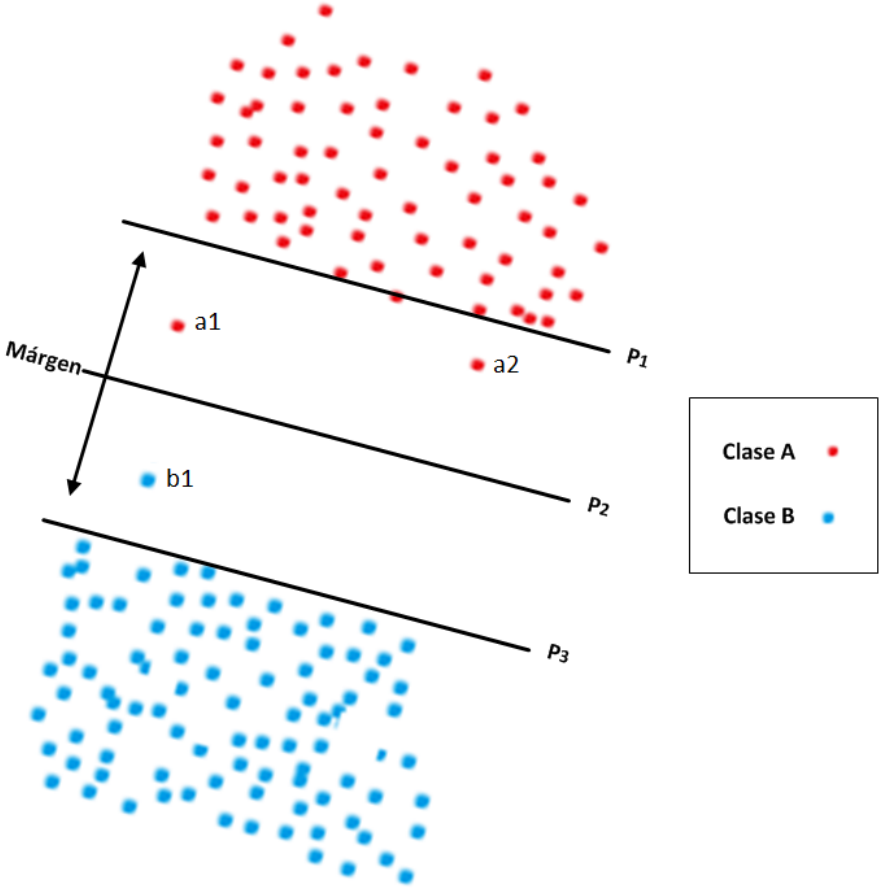

Figure 2.3: Separación con margen. Los vectores p1 y p3 son los llamados *support vectors*

Se busca que el margen sea ancho para una mayor capacidad de generalización. Para lograrlo, SVM utiliza una condición de márgenes relajados *"soft margin"* que permite que haya algunos elementos de las clases a separar dentro de los márgenes (puntos a1, a2 y b1 en la figura 2.3). El balance entre maximizar el margen y minimizar la cantidad de elementos dentro del mismo impacta en la performance y en el error de clasificación.

Trabajar con espacios de grandes dimensiones es muy costoso tanto a nivel de procesamiento como de memoria. En vez de proyectar cada punto en la nueva dimensión, se utiliza lo que se conoce como *kernel trick* (el truco del kernel). El "truco" se encuentra en que no es necesario realizar toda la transformación a la nueva dimensión, simplemente basta con poder calcular el producto interno en dicho espacio.

Para ello, se utilizan funciones de kernel que simplifican el cálculo y permiten computar el producto interno de dos vectores pertenecientes a una dimensión N en una dimensión M, con $M \geq N$, trabajando en la misma dimensión N. Se define $K : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$ y $\phi : \mathbb{R}^N \to \mathbb{R}^M$ tal que $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle_M$ donde $\langle \cdot, \cdot \rangle_M$ es el producto interno de $\mathbb{R}^M$ y $\phi(x)$ transforma x a $\mathbb{R}^M$. Dado que K trabaja exclusivamente en $\mathbb{R}^N$ y el resultado es un escalar, se está computando el producto interno de M en N.
La construcción de la función de Kernel ideal para un problema particular puede ser compleja, sin embargo existen varias funciones genéricas que se adaptan a muchos problemas del mundo real (Press, 2007):

1. Lineal: $K(x_i, x_j) = x_i \cdot x_j + c$
2. Potencia: $K(x_i, x_j) = (x_i \cdot x_j)^d$
3. Polinómico: $K(x_i, x_j) = (ax_i \cdot x_j + c)^d$
4. Sigmoide: $K(x_i, x_j) = tanh(ax_i \cdot x_j + c)$
5. Función Básica Radial Gaussiana: $K(x_i, x_j) = exp(-\frac{1}{2}|x_i - x_j|^2/\sigma^2)$

Debido a que este es un método de clasificación mas robusto que los métodos enunciados anteriormente (Hu et al., 2005), es ampliamente utilizado para el procesamiento de señales neuronales. En Vogelstein et al. (2004) se lo utiliza en el proceso de ordenamiento de *spikes* y reducción de ruido, con resultados superiores a las técnicas de ordenamiento de *spikes* basadas en "template-matching". Otro ejemplo es el de resolución de superposiciones en ordenamiento de *spikes* descripto en Ding y Yuan (2008).
También es utilizado en la detección de *spikes* en señales electroencefalográficas (Acır y Güzeliş, 2004).

Este método se enuncia en Cortes y Vapnik (1995).

2.4 Random Forest

El método de *Random Forest* utiliza la técnica divide y conquista (*divide and conquer*). Se trata de un ensamble de árboles de decisión entrenados con valores provenientes de un vector de muestras aleatorias, tomadas de forma independiente y con la misma distribución para todos los árboles (Breiman, 2001). De esta manera, mediante un sistema de votos, se logra asignar la clase correspondiente a cada una de las muestras. Como los árboles de decisión son predictores sin sesgo pero tienen alta varianza, Random Forest busca reducir la varianza promediando un alto número de árboles de decisión (Hastie et al., 2009). De esta forma, el error de generalización tiende a un límite cuanto

mayor sea el número de árboles (Breiman, 2001). El método es un caso particular de *bagging*, cuya idea principal es promediar muchos modelos con ruido y muy poco sesgo para conseguir reducir la varianza. Los árboles de decisión son candidatos ideales para *bagging*, debido a que pueden capturar estructuras con interacción compleja en los datos y tienen relativamente bajo sesgo cuando crecen lo suficientemente profundo. Adicionalmente, los árboles se benefician en gran medida con el promedio, dado que son muy sensibles al ruido en los datos (Hastie et al., 2009). En la figura 2.4 se ejemplifica el uso de Random Forest para la determinación de la probabilidad que una muestra pertenezca a la clase $\boldsymbol{c}$ dado que se observó la variable $\boldsymbol{f}$.

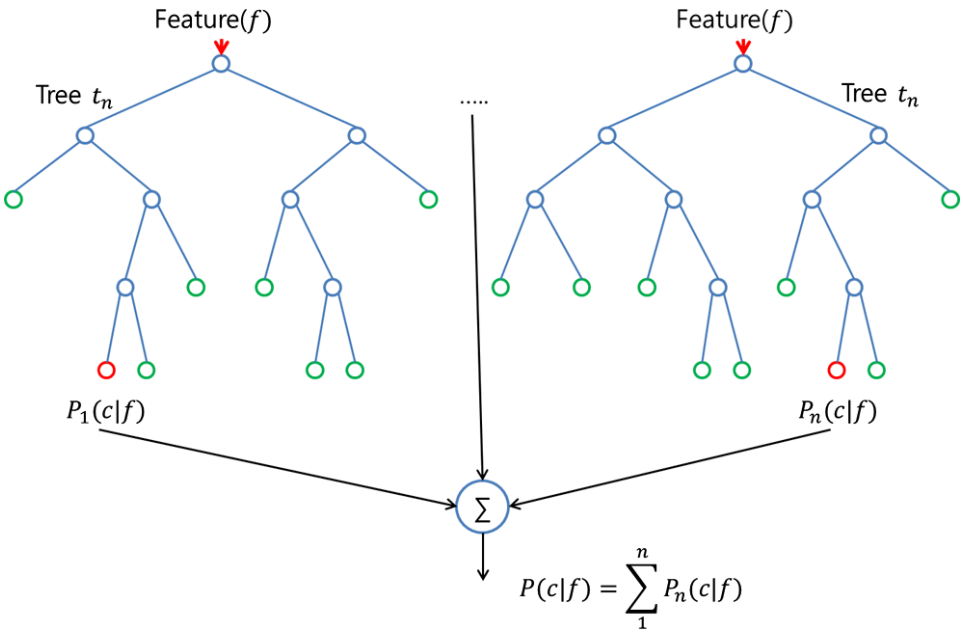

Figure 2.4: Proceso de clasificación de una muestra con Random Forest

Cada variable (*feature*) "f" de la muestra a clasificar es evaluada en cada árbol, recorriendo en forma binaria las ramas, hasta llegar a una hoja. Estas hojas determinan la probabilidad que la muestra pertenezca a la clase "c", dado que esta posee el *feature* "f". Esta probabilidad constituye el voto de cada árbol para la clase c. Recorriendo el vector de *features* y sometiendo cada *feature* de la muestra al mismo proceso de votación, se obtiene, mediante la suma de todos los votos, la decisión acerca de la clase a la que pertenece la muestra. (Kim y Kang, 2013; Breiman, 2001).

Este método se ha utilizado tanto para la clasificación, como para la selección de las *features* mas relevantes en el procesamiento de señales neuronales.

En Oh et al. (2003) fue empleado para reducir el número de variables en un ensamble de tren de *spikes*. A su vez en Lehmann et al. (2007); Fraiwan et al. (2012) se utilizó Random Forest en señales electroencefalográficas, para detectar la enfermedad del Alzheimer.

Set de datos

3.1 Origen de los Datos

El set de datos analizado proviene del registro de la actividad neuronal correspondiente a cuatro ratas macho adulto Long Evans. Las mismas fueron entrenadas para discriminar dos estímulos auditivos de frecuencias diferentes ($1KHz$ y $8KHz$). Luego que alcanzaran una performace mayor a 80%, probadas en el paradigma de discriminación, se registró la actividad neuronal del Área Tegmental Ventral (VTA) y de la Corteza PreFrontal (PFC). Se realizaron 30 sesiones de registro (con los 4 animales), obteniéndose luego de un proceso de *spike* sorting 153 neuronas VTA y 95 neuronas PFC. (Quiroga et al., 2004).

Para el entrenamiento se utilizó el paradigma GO/NOGO. Dicho paradigma consiste en una tarea de discriminación, en la cual los estímulos son presentados en un flujo continuo y los participantes deben realizar una decisión binaria para cada uno de ellos. Se presentan 2 tipos de estímulo distintos, en un caso se espera de los participantes una respuesta motora (caso GO), mientras que en el otro dichos participantes deben contener la respuesta (caso NOGO) (Donders, 1969).
En dos de los animales entrenados, la presentación del estímulo auditivo de mayor frecuencia fue asociada con la respuesta GO (sacar la lengua en una ventana temporal de 2 segundos posteriores al estímulo), mientras que el de menor frecuencia fue asociado con la respuesta NOGO (esconder la lengua por 2 segundos). En los otros dos animales los estímulos se asociaron a las respuestas de manera opuesta, para evitar cualquier sesgo en los resultados. En todos los casos la respuesta GO correcta (sacar la lengua cuando el tono presentado era el correcto) fue recompensada con una gota de agua. La respuesta de *lick* fue filmada con una cámara y se determinó a través del procesamiento de imágenes de la zona bucal de cada animal.

Los animales fueron registrados una vez por día (sesión) y en cada sesión

se repitió la secuencia Tono > Respuesta > Refuerzo tantas veces (trials) como el animal quiso, hasta alcanzar la saciedad. Los trials de cada sesión se dividieron de acuerdo a la respuesta que el animal tuvo que realizar y de acuerdo a si la misma fue o no correcta:

- **GOc**: trial GO correcto, el animal tuvo que sacar la lengua una vez (lick) y recibió agua
- **GOi**: trial GO incorrecto, el animal tuvo que sacar la lengua una vez (lick) y no lo hizo.
- **NOGOc**: trial NOGO correcto, el animal tuvo que esconder la lengua durante 2 segundos (no-lick) y así lo hizo.
- **NOGOi**: trial NOGO incorrecto, el animal tuvo que esconder la lengua durante 2 segundos (no-lick) y sin embargo la sacó.

Como cada tipo de trial tiene unívocamente asociado un estímulo auditivo determinado, la relación estímulo-respuesta es unívoca. Por este motivo hablar de trial GOc en el procesamiento de datos es equivalente a hablar de un estímulo auditivo determinado, presentado durante ese trial.

Para el análisis se tomó una ventana de $-4000\ ms$ a $+4001\ ms$ para las neuronas VTA y de $-4000\ ms$ a $+3000\ ms$ para las neuronas PFC, siendo $0ms$ el inicio del estímulo. La duración del mismo es de $1000ms$.

La información relativa al formato y distribución de los datos, junto con el análisis de los mismos se detalla en las secciones A.2 y A.3 del Apéndice A.

3.2 Selección de Features

La técnica utilizada para entrenar los algoritmos estadísticos elegidos fue aprendizaje supervisado (Mohri et al., 2012). Esta técnica consiste en predecir la clase a la que pertenece una muestra después de haber aprendido con una serie de ejemplos (datos de entrenamiento). Para el aprendizaje supervisado se tomó en forma aleatoria el 70% de los datos para el entrenamiento y se validó el resultado del entrenamiento con el 30% restante de los mismos (etapa de test). Como característica de la señal (*feature*) se utilizó la actividad media de las neuronas en una ventana temporal determinada, esto es, la tasa de disparo promedio. Podrían haberse empleado otras características de la señal neuronal como el Inter Spike Interval o incluso la variabilidad con respecto al promedio, pero la tasa de disparo es una medida directa de actividad, lo que simplifica la validación de los resultados obtenidos.

3.2.1 Análisis de neurona única

Como primera aproximación se intentará separar las clases utilizando la actividad de una sola neurona por clasificador, repitiendo el proceso para todas las neuronas de la población registrada. En otros términos, se estudiará cuanta información en promedio tiene el conjunto de neuronas, sin considerar que puede haber interacción entre ellas.

Para entrenar los clasificadores se debe generar un espacio de *features*, que permita al clasificador utilizado, encontrar correlaciones entre los mismos y las respuestas esperadas.
Se sabe, de acuerdo a lo observado en la sección 3.1, que cada *trial* se corresponde unívocamente con una respuesta del animal. Por ende, es posible deducir que la respuesta esperada para cada *trial* será la respuesta del animal (GOc, GOi, NOGOc, NOGOi).
Los clasificadores requieren una matriz de *features* y un vector de salidas esperadas para poder realizar el entrenamiento. De ahora en adelante se llamará a esta matriz X y al vector de respuestas esperadas Y.
Las columnas de X serán los *features*, y las filas los *trials*. Cada fila de X tendrá una salida esperada, que se incluirá en el vector Y. Lo que denota que la fila n de X se corresponde con la posición n de Y.

El análisis entonces consiste en determinar cuales serán los *features* que se utilizarán en el entrenamiento.

MATRIZ X						VECTOR Y
Trials\Features	Feature 1	Feature 2	...	Feature n		Resultados
Trial 1					➡	r1
Trial 2					➡	r2
Trial 3					➡	r3
...					➡	...
Trial n					➡	rn

Figure 3.1: Matriz X y Vector Y

Utilizando solamente una neurona, se explorarán 2 opciones:

A. Tomar como columna de la matriz X a cada milisegundo, teniendo entonces 8001 columnas. Para cada intersección con cada fila (trial), habrá un 1 si hubo un *spike* y un 0 si no lo hubo.

B. Tomar como columna de la matriz X la suma de todos los *spikes* durante los $8001ms$. De esta forma se tendrá una única columna en la matriz X. Esta opción representa una medida de cuanto disparó la neurona.

El primer problema observado radica en la poca cantidad de trials GOi y NOGOi, lo que apareja que los clasificadores no encuentren un patrón para poder diferenciarlos de los demás.

Supervised Learning LDA

Classifier performances

Error rate	0,3338

Values prediction

Value	Recall	1-Precision
GOc	0,6185	0,2595
NOGOi	0,0000	0,0000
NOGOc	0,7855	0,3765
GOi	0,4500	0,5263

Confusion matrix

	GOc	NOGOi	NOGOc	GOi	Sum
GOc	214	0	131	1	346
NOGOi	9	0	19	0	28
NOGOc	62	0	260	9	331
GOi	4	0	7	9	20
Sum	289	0	417	19	725

Figure 3.2: Matriz de confusión de una neurona LDA con las cuatro clases, donde se observa la poca cantidad de trials GOi y NOGOi, en comparación con las otras dos clases.

Por el motivo explicado y los resultados obtenidos, se decidió de ahora en mas utilizar solamente los trials de GOc y NOGOc para simplificar el trabajo de los clasificadores.

Con esta simplificación, se intentó separar las clases GOc y NOGOc.

Para el set de neuronas LDA:
Utilizando la opción A, la performance obtenida fue de 0.5354 ± 0.0956.
Utilizando la opción B, la performance fue de 0.5910 ± 0.1179.

La performance total se calculó como el promedio de la performance individual de clasificación entre las 153 neuronas $\pm$ su desvío estándar.

Para el set de neuronas PFC:
Utilizando la opción A, la performance obtenida fue de 0.5072 ± 0.0485.
Utilizando la opción B, la performance fue de 0.5433 ± 0.0755.

La performance total se calculó como el promedio de la performance individual de clasificación entre las 95 neuronas $\pm$ su desvío estándar.

A efectos de tener una mejor comprensión del set de datos, se muestra a continuación la clasificación individual de cada neurona.
Se entrenó entonces, utilizando la opción B, un clasificador LDA para cada neurona corriendo el algoritmo 500 veces y se calculó la media.

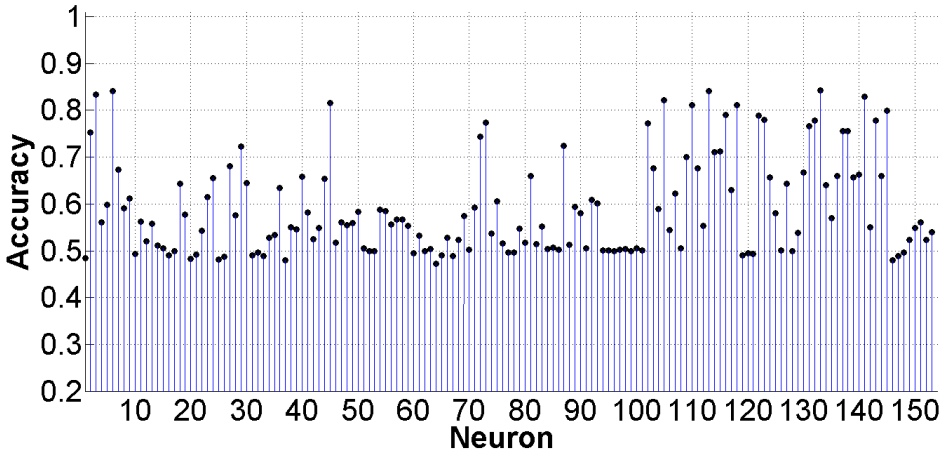

Figure 3.3: Performance de las neuronas VTA - Corridas individuales. Cada par X-Y corresponde a una neurona con su precisión de clasificación (media para las 500 corridas). - LDA

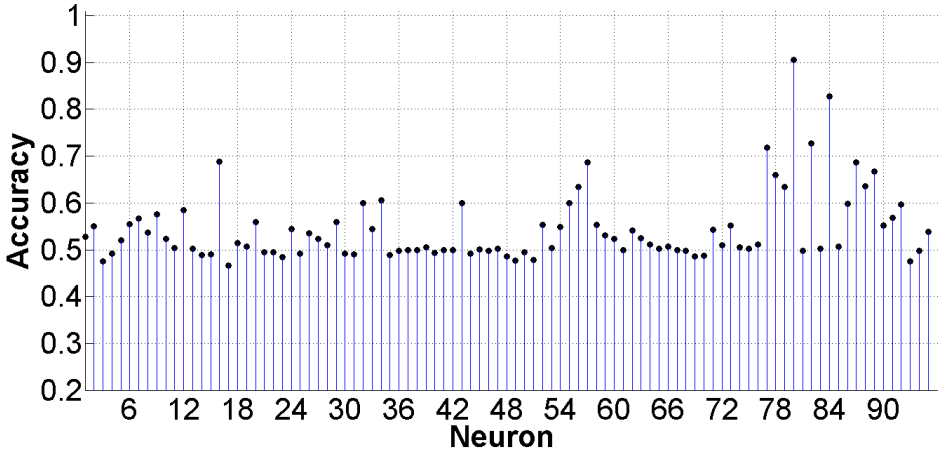

Figure 3.4: Performance de las neuronas PFC - Corridas individuales. Cada par X-Y corresponde a una neurona con su precisión de clasificación (media para las 500 corridas). - LDA

Se pueden observar neuronas que por si solas clasifican el set de datos. En el caso de PFC se aprecia como la media de la neurona 80 supera el 90% y la 84 el 80%. Estas neuronas son casos aislados, que si bien separan correctamente las clases, no representan el comportamiento global.
En VTA, en cambio, no hay ninguna neurona que sobrepase el 90%, pero se observan varias que están por arriba del 80%.

Dado que la media de la performance de clasificación ± su desvio estándar con LDA se solapan con el 0.5, se puede inferir que el resultado no es significativo. Por ende, descartando los casos aislados, se concluyó que con una sola neurona no hay suficiente información para diferenciar los trials.

3.2.2 Análisis poblacional

3.2.2.1 Análisis

Como se vio en la sección anterior, la predicción del estímulo presentado en base a la información obtenida de neuronas individuales no es significativa. Es por este motivo, que se intentará analizar si el contenido de información de la población de neuronas aumenta en función del número de neuronas registradas.
Si las neuronas fuesen procesos estocásticos independientes, es esperable que el contenido de información no aumente, pero si existen interacciones de segundo o mayor orden probablemente la predicción del estímulo sea posible.

Al analizar varias neuronas, al igual que en el caso anterior, se debe definir con que *features* se entrenarán los clasificadores.
Aprovechando el análisis de neurona única realizado en la sección anterior, se puede elaborar un análisis similar para el caso poblacional. La opción B del análisis de neurona única ahora parece mas interesante, ya que de esta manerá se tiene un *feature* por cada neurona. De este modo la matriz X queda armada de la siguiente forma: cada neurona es una columna y cada trial una fila. La misma esta completa con la suma de los *spikes* para cada neurona en cada trial.
Para las salidas esperadas Y, al igual que en el caso anterior, se toman las respuestas de los trials.

Definir un *feature* por cada neurona implica tener una dimensión distinta para cada neurona, lo cual puede ayudar a facilitar la clasificación. En la figura 3.5 se puede apreciar un ejemplo de 2 neuronas, donde individual-

mente la clasificación sería muy difícil, pero al graficar las 2 dimensiones, se vuelve un problema linealmente separable.

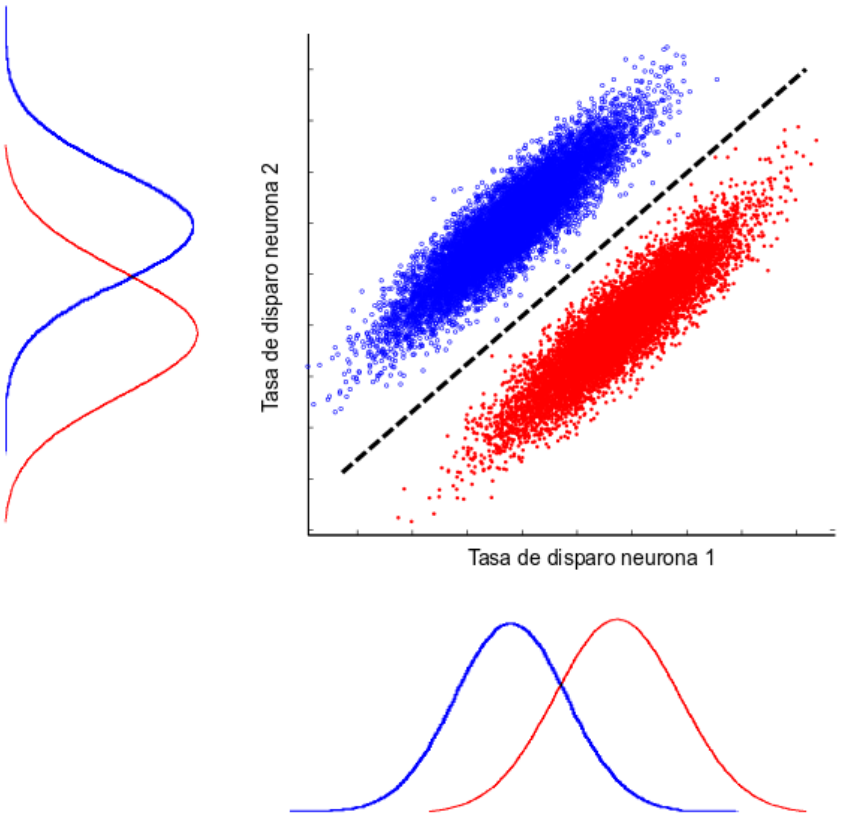

Figure 3.5: Figura a modo de ejemplo (datos ficticios) que ilustra un problema no linealmente separable si se considera cada neurona por separado, pero que se vuelve linealmente clasificable tomando como *feature* a cada neurona.

3.2.2.2 Preparación

La construcción se realiza para el área VTA para simplificar la explicación, pero el procedimiento aplica de igual manera al área PFC.

La matriz X enseñada al clasificador tiene a las neuronas como columnas y los distintos trials de GOc y NOGOc como filas. El vector Y contiene las respuestas esperadas.

Para armar dicha matriz, se debe buscar la máxima cantidad de trials GOc y NOGOc que aparecen mínimamente en todas las neuronas. Se cuenta con 31 trials GOc y 23 trials NOGOc disponibles como mínimo en todas las sesiones. Se elije entonces el mínimo valor de los dos (23) para conseguir que la matriz tenga la misma cantidad de trials de uno y de otro y que el análisis resulte balanceado.

Se tiene la matriz X con 153 columnas (cantidad total de neuronas VTA que se registraron en las 30 sesiones) y 46 filas que representan 23 trials GOc y 23 trials NOGOc. También se cuenta con el vector Y con 46 posiciones donde en cada posición se colocará un 1 o un 2, dependiendo si el trial (fila de X) correspondiente a este número de posición corresponde a la clase 1 (GOc) o a la clase 2 (NOGOc).

MATRIZ X						VECTOR Y
Trials\Features	Neurona 1	Neurona 2	...	Neurona 153		Resultados
Trial 1	$x_{1,1}$	$x_{1,2}$	...	$x_{1,153}$	➡	GO
...	...	...	...	...	➡	GO
Trial 23	$X_{23,1}$	$X_{23,2}$	...	$X_{23,153}$		GO
Trial 24	$X_{24,1}$	$X_{24,2}$	...	$X_{24,153}$	➡	NOGO
...	...	...	...	...	➡	NOGO
Trial 46	$X_{46,1}$	$X_{46,2}$	...	$X_{46,153}$	➡	NOGO

Figure 3.6: Matriz X y Vector Y - Análisis poblacional - $x_{i,j}$ representa la suma de todos los *spikes* para el *trial* i de la neurona j

A su vez, se toma el 70% de las filas de la matriz X para entrenamiento y el 30% restante para medir la performance de clasificación.

3.2.2.3 Resultados

Se entrenó un LDA con las matrices mencionadas anteriormente y se consiguió una performance alrededor del 90%.
Estos resultados son significativos, mas adelante se analizarán las implicancias de los mismos.

Resultados

4.1 Evolución temporal: análisis por ventanas

Como se mostró en el capítulo 3, la información contenida en la población de neuronas ayuda a predecir el estímulo presentado con mayor performance que el promedio de lo que consigue cada neurona individualmente. Sin embargo, en ese análisis exploratorio sobre la actividad de las neuronas de VTA, se consideró la actividad total de cada neurona dentro de la ventana de análisis $(-4000ms, 4000ms)$ alrededor del inicio del tono. Para entender en que momento comienzan las neuronas a tener información sobre el estímulo presentado, en esta sección se realizará un análisis en ventanas pequeñas $(300ms)$ que se deslizan cada $10ms$ a lo largo de la duración del trial.

La actividad total en cada ventana por neurona se utilizó como *feature* para entrenar los clasificadores y, como antes, la salida deseada para cada trial fue el tipo GO correcto o NOGO correcto. En la figura 4.1 se muestra un ejemplo de ventana aplicado sobre la matriz de 0 y 1 original.

NEURONA A

Milisegundos

Trial/Milisegundos	1	2	...	x	x+1	...	...	x+Wsize-1	x+Wsize	...	n-2	n-1	n	Respuesta
Trial 1	1	1	...	1	0	...	...	0	1	...	1	0	0	NOGOc
Trial 2	0	1	...	1	1	...	...	0	1	...	0	1	1	GOc
Trial 3	1	0	...	1	0	...	...	0	1	...	0	1	0	NOGOc
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Trial m	0	1	...	0	1	...	...	1	0	...	0	0	1	GOc

Wi

Figure 4.1: División de la neurona en ventanas

De esta manera, para cada neurona, en un instante determinado puede generarse el vector X_w que contiene la actividad en la ventana w y tiene asociado un vector Y_w con la información acerca del tipo de trial. La figura 4.2 muestra un ejemplo de la estructura de datos construida para entrenar los

clasificadores.

NEURONA A

Trial/Ventana	w1	w2	…	wi	…	wn	Respuesta
Trial 1	3	6	…	3	…	5	NOGOc
Trial 2	4	10	…	2	…	6	GOc
Trial 3	15	0	…	10	…	10	NOGOc
…	…	…	…	…	…	…	…
Trial m	8	3	…	11	…	4	GOc

NEURONA B

Trial/Ventana	w1	w2	…	wi	…	wn	Respuesta
Trial 1	7	3	…	4	…	9	GOc
Trial 2	9	16	…	3	…	3	GOc
Trial 3	12	4	…	5	…	17	NOGOc
…	…	…	…	…	…	…	…
Trial m	6	0	…	1	…	5	NOGOc

MATRIZ X (Wi) **VECTOR Y (Wi)**

Trial/Neurona	Neurona A	Neurona B	…	Neurona N	Respuesta
Trial 1	2	4	…	6	GOc
Trial 2	11	3	…	8	GOc
Trial 3	3	5	…	3	NOGOc
…	…	…	…	…	…
Trial m	10	1	…	5	NOGOc

Figure 4.2: Composición de la matriz X_w y el vector Y_w

La figura 4.3 muestra el resultado del análisis con el algoritmo LDA, entrenando un clasificador por cada ventana, tomando 70 % de los trials al azar y verificando con el 30 % restante.

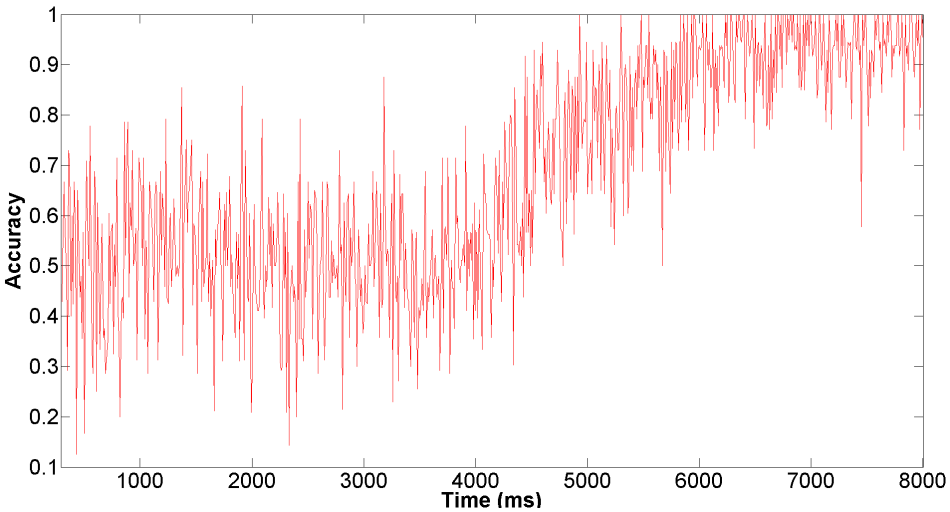

Figure 4.3: LDA - Performance en función del tiempo.

En la figura 4.3 se puede apreciar claramente como a partir del inicio del tono (4000 ms) la performance comienza a incrementarse. Puede observarse que la performance del clasificador es cercana a lo esperado por azar cuando el tono no fue todavía presentado.También puede notarse la gran fluctuación que hay entre ventanas contiguas, esto se debe principalmente a la variabilidad en los datos. Para determinar si el incremento en la performance es significativo, se realizó un análisis con re-muestreo con el fin de determinar la variabilidad alrededor del valor promedio

4.2 Análisis de la variabilidad en la decodificación

En la sección anterior se mostró que entrenando un clasificador lineal con un porcentaje de los trials (70%), la performance de la predicción acerca de cual es el estímulo presentado, crece luego de la aparición del mismo. De esta manera, calculando el promedio de los aciertos en los trials de prueba (30%), se tiene una idea de cual es la performance media esperada, pero no de cuanta variabilidad existe en esta medición. Por lo tanto, no es posible saber cuan esperable sería un resultado parecido, si se utilizara otro conjunto de trials para entrenar el decodificador. Para resolver este problema y obtener una estimación de la variabilidad, se empleará en este capítulo una técnica de re-muestreo denominada Bootstrapping (Efron, 1979).

Se realizará el análisis utilizando LDA, Bayes Naive, Support Vector Machine (SVM) y Random Forest y se comparará si existen diferencias significativas en el desempeño de estas herramientas.
El análisis se efectuará utilizando ventanas solapadas (10 ms) de 300 ms de longitud. Para cada ventana se creará la matriz X y el vector Y, luego se entrenará a la herramienta tomando 70% de los trials de forma aleatoria y se calculará la performace (número de aciertos / número de trials) con el 30% restante. Se repetirá este procedimiento 500 veces con el objetivo de calcular la performance promedio y el Error Estándar a la Media del Bootstrapping (B.S.E.M) de la performace de clasificación. En cada una de las corridas se utilizará el mismo conjunto de muestras para entrenar y probar los clasificadores, de esta manera, se podrán realizar comparaciones apareadas sobre los resultados obtenidos.

En el Apéndice B se especifican los parámetros utilizados en cada uno de los métodos.

4.2.1 Área Tegmental Ventral

En la figura 4.4 se puede observar el desempeño de los cuatro métodos. Se grafica la performance promedio de las 500 corridas por cada ventana y el B.S.E.M.

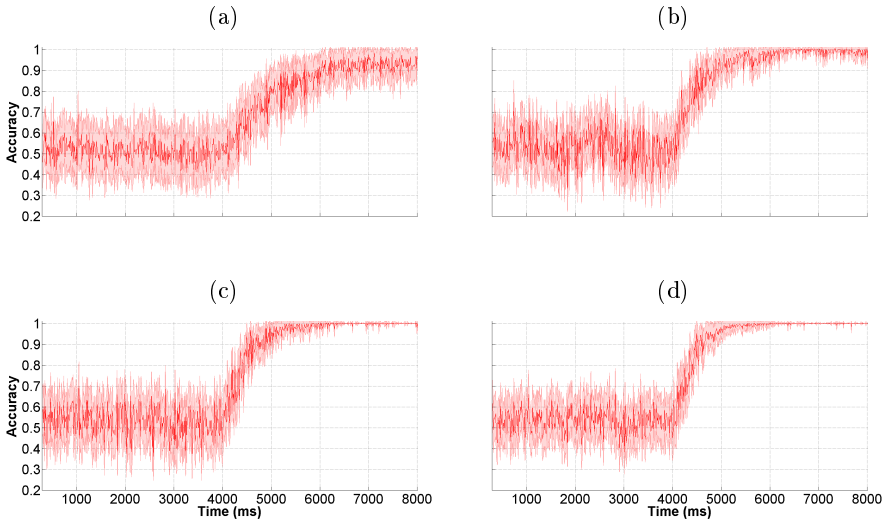

Figure 4.4: Performance en la decodificación en función del tiempo para 500 re-muestreos en las neuronas VTA. a) Análisis de discriminante lineal. b) Análisis con Naive Bayes. c) Análisis con Support Vector Machine. d) Análisis con Random Forest.

Se observa que la performance de clasificación para LDA es mayor a 50%, un segundo después de la presentación del estímulo ($5000ms$). Para los otros métodos esto sucede aproximadamente $500ms$ después del inicio del estímulo, lo que indica que estos últimos necesitan menos exposición al estímulo para decodificar correctamente.
Si la población neuronal mantiene información acerca de un estímulo que ya no está presente, debería ser posible predecirlo sólo en base a la actividad post-estímulo.

En la figura 4.4 se ve que la performance en la decodificación crece incluso luego de la desaparición del estímulo (5000 ms). Por esta razón, y para comparar el desempeño de los métodos, se hizo un análisis de las performances obtenidas entre 5000 ms y 8000 ms. En la figura 4.5 se puede apreciar la performance promedio y el B.S.E.M para cada método.

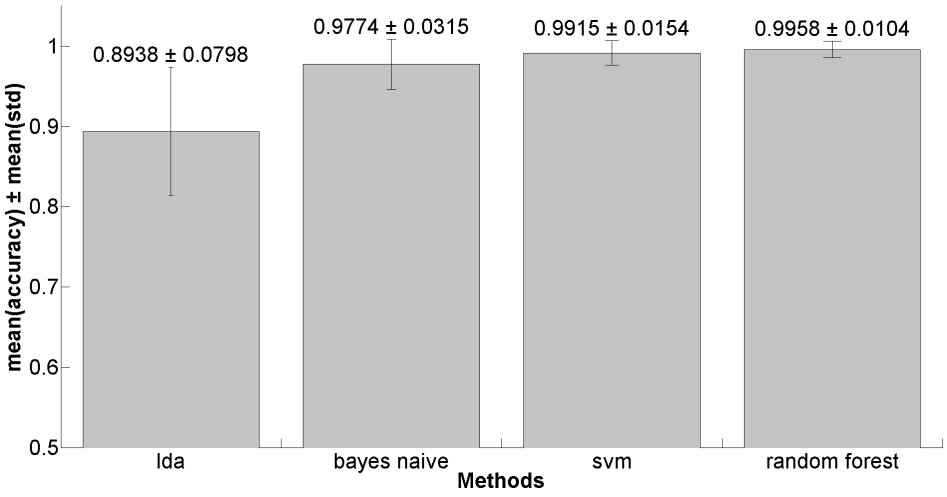

Figure 4.5: Comparación para todas las neuronas VTA en barras a partir del milisegundo 5000

En la Tabla 4.6 se muestra el valor p con el cual se puede evidenciar que un método M_i funciona mejor que otro M_j. Esta significancia se calculó contando las veces en que la media de la performance de una ventana para uno de los métodos, fue más alta que la del otro, es decir, sin asumir ninguna distribución paramétrica:

$$p = 1 - \frac{\#\ Perf(M_i) > Perf(M_j)}{Número\ de\ ventanas} \tag{4.1}$$

	p
Random Forest mayor que LDA	0
Random Forest mayor que Bayes Naive	0,122924
Random Forest mayor que SVM	0,468439
SVM mayor que LDA	0
SVM mayor que Bayes Naive	0,149502
Bayes Naive mayor que LDA	0,006645

Figure 4.6: Significancia en la que la media de un método es mejor que la de otro a partir del milisegundo 5000 para las neuronas VTA.

En la tabla 4.6 se ve que LDA presenta una performance inferior a los otros métodos, mientras que no hay diferencia significativa entre Bayes Naive, SVM y Random Forest. Un análisis con más datos podría determinar si existen diferencias significativas entre estos tres últimos métodos.

4.2.2 Corteza PreFrontal

El análisis precedente se repitió sobre una población de neuronas registradas simultáneamente en la corteza PreFrontal. En la figura 4.7 se puede ver el desempeño de los cuatro métodos. Se grafica la performance promedio de las 500 corridas por cada ventana y el B.S.E.M.

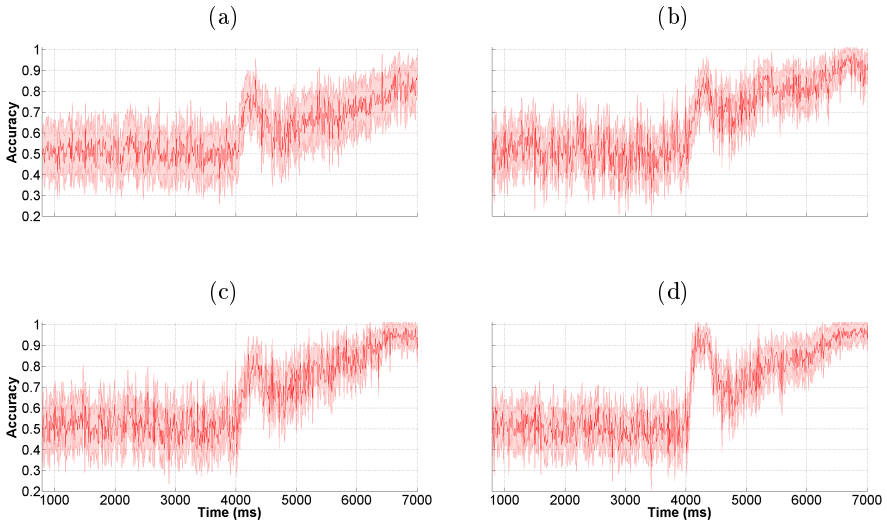

Figure 4.7: Performance en la decodificación en función del tiempo para 500 re-muestreos en las neuronas PFC. a) Análisis de discriminante lineal. b) Análisis con Naive Bayes. c) Análisis con Support Vector Machine. d) Análisis con Random Forest.

Se puede observar un incremento importante de la performance $100ms$ después de la presentación del estímulo, a diferencia de lo que se midió en VTA, donde la performance en la decodificación se incrementa a medida que transcurre el tiempo y, una vez que la rata toma la decisión, la misma se ve reflejada en la performance creciente.

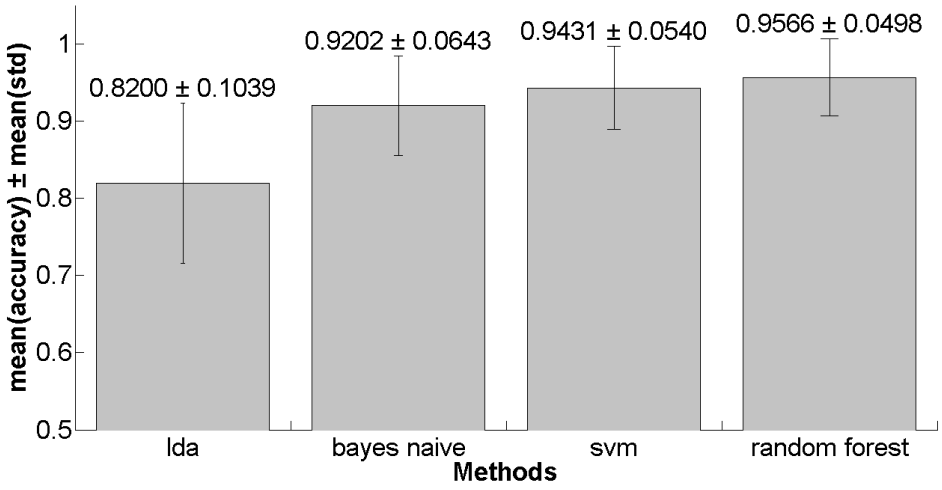

Figure 4.8: Comparación apareada para todas las neuronas PFC en barras a partir del milisegundo 6500.

En la figura 4.8 se aprecia como la performance en la decodificación alcanza un 0.9566 ± 0.0498 utilizando *Random Forest*. Si bien es menor que la obtenida en el Área VTA, igualmente alcanza para demostrar que en PFC también hay información acerca del tono. En la siguiente figura se puede observar el incremento que tiene lugar un instante posterior a la presentación del estímulo.

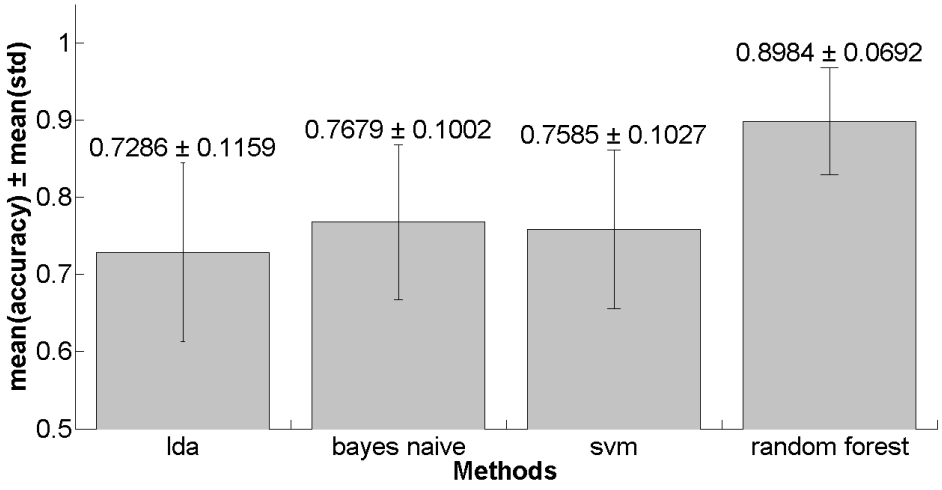

Figure 4.9: Comparación apareada para todas las neuronas PFC en barras entre los milisegundos 4100 y 4400.

En la sección 4.3 se analizará este fenómeno.

	p
'Random Forest mayor que LDA'	0
'Random Forest mayor que Bayes Naive'	0,137255
'Random Forest mayor que SVM'	0,27451
'SVM mayor que LDA'	0
'SVM mayor que Bayes Naive'	0,235294
'Bayes Naive mayor que LDA'	0,019608

Figure 4.10: Significancia en la que la media de un método es mejor que la de otro a partir del milisegundo 6500 para las neuronas PFC.

En la tabla 4.10 se observa que, al igual que en VTA, los métodos *Random Forest*, SVM y *Bayes Naive* son mejores que LDA. Los resultados muestran que hay suficiente información en el conjunto de neuronas de PFC, para predecir el estímulo auditivo con una performance de 0.9566 ± 0.0498

4.2.3 Conclusiones

El análisis del discriminante lineal (LDA), aunque más simple, resulta ser menos eficiente que los otros métodos analizados. Esto ya fue demostrado en la literatura (Lee et al., 2005; Lehmann et al., 2007; Maroco et al., 2011) pero nunca con un conjunto de datos de actividad neuronal de neurona única. Sin embargo, la simplicidad de LDA lo hace más atractivo para implementaciones portátiles que requieran volúmenes pequeños (implementaciones en FPGA) y bajo consumo de energía, como es el caso de las Interfaces Cerebro Computadora portátiles. Estos requerimientos (volumen y consumo) pueden ser minimizados atacando varios factores. Uno de ellos es el número de electrodos que el sistema debe procesar, cuanto menor sea el número de electrodos menor serán los requerimientos computacionales, pero disminuye el número de neuronas promedio registradas, lo que implicaría contar con menor volumen de información.
En la sección siguiente se hará un análisis de como la cantidad de neuronas utilizadas para la decodificación, impacta en la performance de los clasificadores.

4.3 Dependencia con el tamaño de la población

Como se mencionó en la sección anterior, la necesidad de reducir el volumen y consumo de las interfaces portátiles, obligan a reducir el número de electrodos a ser registrados y procesados. Por otro lado, la reducción del número de electrodos acarrea otras ventajas asociadas a los problemas de rechazo.
En este capítulo se estudia la robustez de cada método de decodificación en función del número de neuronas intervinientes en el análisis. Se utilizó nuevamente la técnica de *bootstrapping* (500 re-muestreos con reemplazo) y en cada re-muestreo se tomó un porcentaje fijo (100%, 75%, 50%, 25%, 10% y 5%) y aleatorio de neuronas del set VTA. La figura siguiente muestra el resultado de la performance de clasificación para los cuatro métodos y diferentes porcentajes de neuronas utilizadas en el análisis.

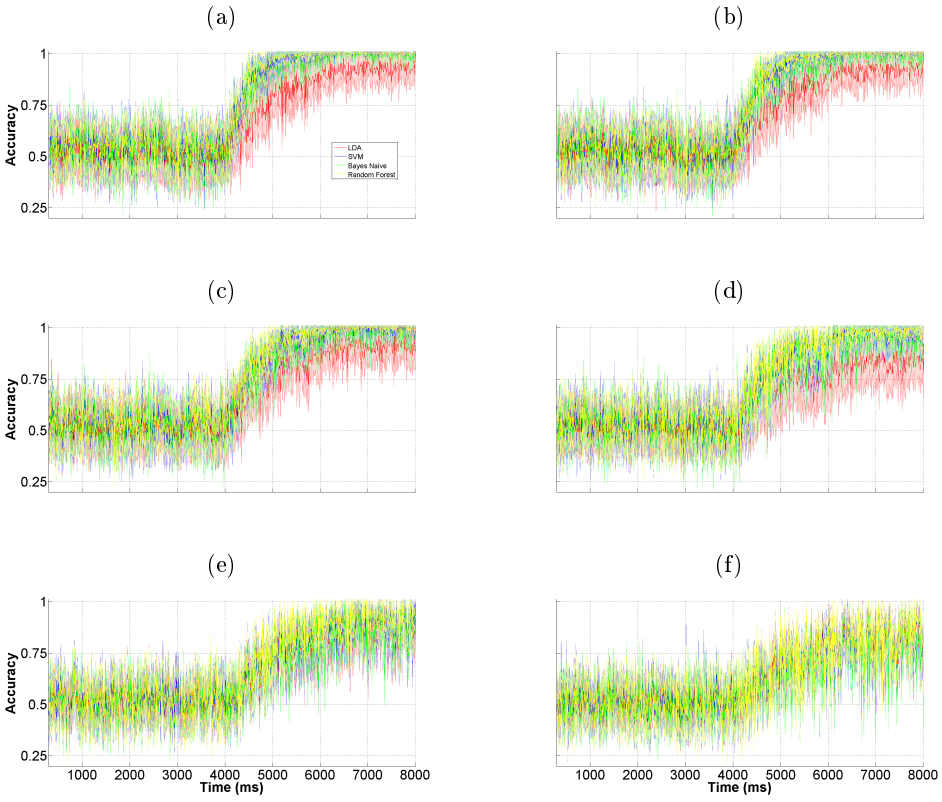

Figure 4.11: Gráficos performance/tiempo. Cada sub-gráfico corresponde a un porcentaje de neuronas: a) 100%, b) 75%, c) 50%, d) 25%, e) 10%, f) 5%

En la figura 4.11 se puede ver como, cuando la cantidad de neuronas involucradas disminuye, decrece la performance de clasificación en todos los métodos. Observando los últimos $2500ms$ se ve que para las cuatro primeras

figuras, LDA tiene una performance menor a los otros métodos, diferencia que desaparece cuando el porcentaje de neuronas utilizadas es menor o igual a 10%.

La figura siguiente muestra la variación de la performance promedio, para los cuatro métodos, tomando la información posterior a la desaparición del estímulo.

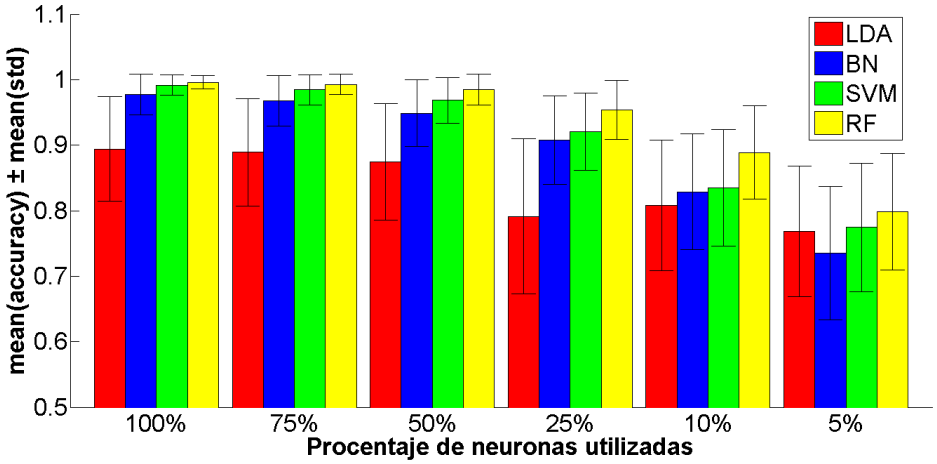

Figure 4.12: Gráfico de barras promediando la performance a partir de $5000ms$. Se grafica la performance en función del porcentaje de neuronas y del método de clasificación utilizado.

Se observa que independientemente del método, cuando la cantidad de neuronas es mayor o igual a 77 (50%) la performance se mantiene casi constante, independientemente de cuantas neuronas mas se agreguen. Por el contrario, debajo del 50% la performance decrece linealmente a medida que el número de neuronas disminuye.

Como se ha visto en los análisis de neurona única (sección 3.2.1), en el conjunto de datos existen neuronas cuya performance de clasificación es alta. Esto puede traer aparejado que el decrecimiento de la performance en alguno de los conjuntos reducidos anteriores, se deba a la eliminación de dichas neuronas. Por este motivo se decidió realizar un nuevo análisis excluyendo aquellas neuronas cuya performance individual supere el 70%. En VTA, se eliminaron 27 neuronas, de acuerdo con el gráfico 3.3. Las neuronas eliminadas son las siguientes: [2 3 6 29 45 72 73 87 102 105 109 110 113 114 115 116 118 122 123 131 132 133 137 138 141 143 145].

La figura 4.13 muestra la performace de clasificación con este conjunto reducido de datos.

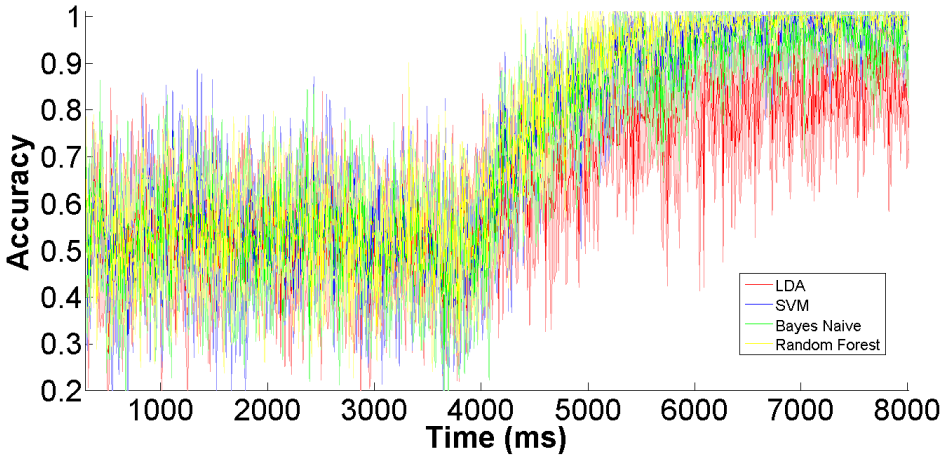

Figure 4.13: Comparación para las neuronas VTA que su performance individual no supera en media al 70%.

Comparando la figura 4.13 con la figura 4.4 se puede observar que la forma de la curva para los cuatro métodos continúa siendo exactamente la misma.

Para poder observar si la performance disminuyó se realizó el siguiente gráfico de barras.

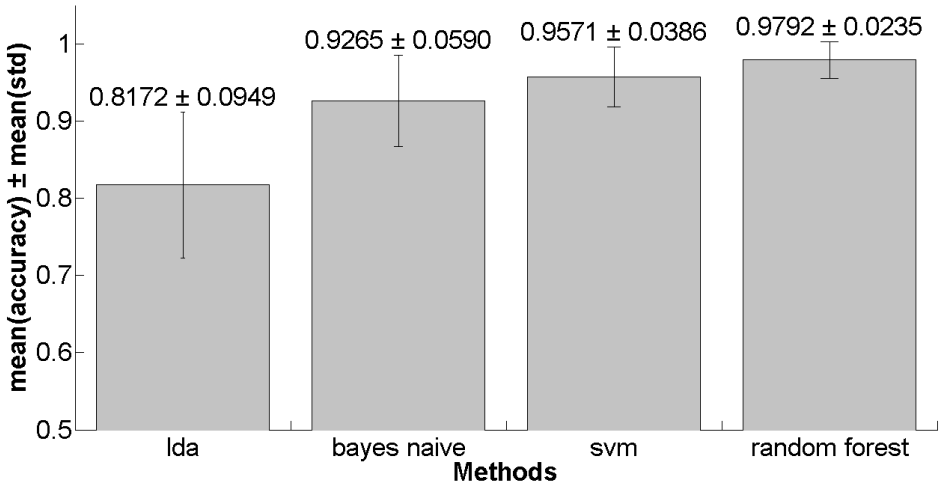

Figure 4.14: Comparación para todas las neuronas VTA cuya performance individual no supera en media al 70%. Gráfico de barras a partir del milisegundo 5000.

Al compararlo con el gráfico 4.5 se puede apreciar una disminución en la performance del 1.66% con *Random Forest*. A su vez, eliminando 38 neuronas al azar (corrida con el 75% de neuronas), la performance con Random Forest disminuyó apenas en 0.32%. Por consiguiente se demuestra que esas 27 neuronas eliminadas son mas importantes que la media de las demás para la clasificación. Asimismo se logra probar que la información está almacenada en el conjunto de las neuronas y no depende de ninguna de ellas individualmente.

Para el área PFC se realizó el mismo procedimiento, corriendo nuevamente los métodos y descartando las neuronas cuya performance individual sea mayor en media al 70% de acuerdo con el gráfico 3.4. Las neuronas eliminadas son cuatro: (77 80 82 84).

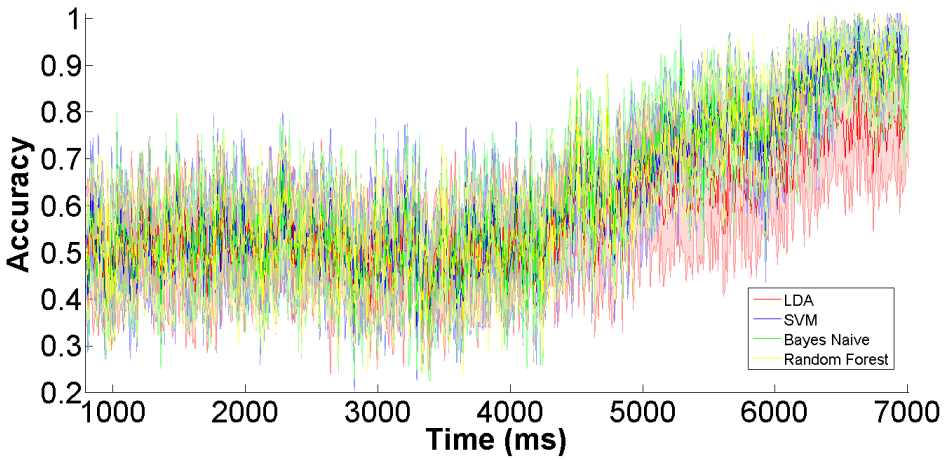

Figure 4.15: Comparación para las neuronas PFC cuya performance individual no supera en media al 70%.

Si se compara la figura 4.15 con la figura 4.7, se observa como el pico posterior al estímulo desapareció y la performance se incrementa lentamente luego del estímulo.

De esta manera queda demostrado que el pico posterior al estímulo, era un efecto de unas pocas neuronas que separaban las clases $100ms$ después de escuchado el tono.

En el gráfico de barras 4.16 se aprecia que hubo una disminución de aproximadamente 5% en todos los métodos con respecto al gráfico 4.8.

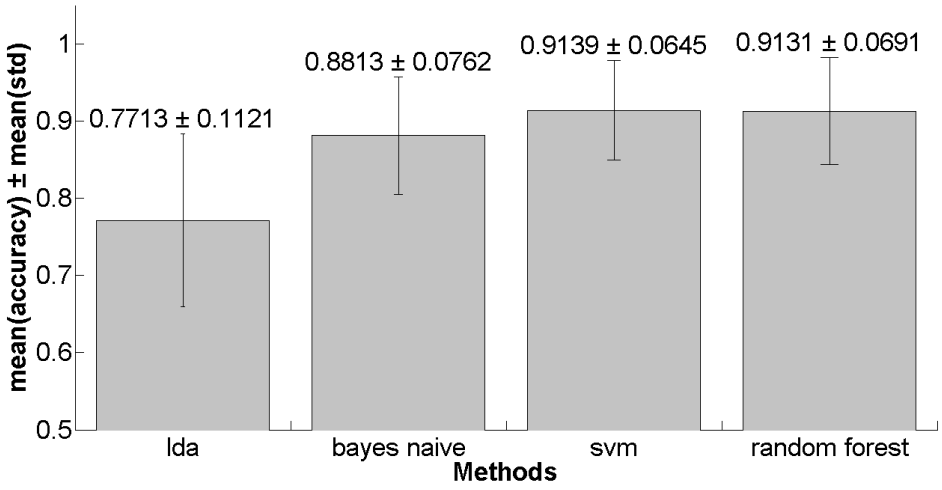

Figure 4.16: Comparación para todas las neuronas PFC cuya performance individual no supera en media al 70%. Gráfico de barras a partir del milisegundo 6500.

Dado que el resultado varió en un 5% aproximadamente, se realizó nuevamente la comparación de las medias ventana a ventana.

	p
'Random Forest mayor que LDA'	0
'Random Forest mayor que Bayes Naive'	0,196078
'Random Forest mayor que SVM'	0,490196
'SVM mayor que LDA'	0
'SVM mayor que Bayes Naive'	0,156863
'Bayes Naive mayor que LDA'	0

Figure 4.17: Significancia en la que la media de un método es mejor que la de otro a partir del milisegundo 6500 para las neuronas PFC, sin las cuatro neuronas que superan el 70% de performance en su clasificación individual.

En la tabla 4.17 se observa que las conclusiones expuestas en la sección anterior no cambian. La única diferencia con la tabla 4.10 radica en entre Random Forest y SVM no se aprecian diferencias significativas.

4.3.1 Conclusión

Los métodos de clasificación analizados requieren una cantidad mínima de neuronas para clasificar con performance significativamente mayor al 50%.
Es importante notar que con sólo 8 neuronas se puede predecir el estímulo presentado (con *Random Forest*) con una performance de 0.798 ± 0.089. Este resultado puede ser aceptable dependiendo del problema que se quiera resolver, pero resulta importante cuando se desea minimizar la cantidad de electrodos implantados.
Asimismo es para destacar la distribución poblacional de la información, dado que la performance de la misma sobrepasa el 90%, aunque se tomen neuronas cuya performance individual no supere el 70%.

4.4 Impacto de la longitud de la ventana de análisis sobre la performance de clasificación

Dada la baja frecuencia de disparo que tienen en promedio las neuronas de VTA y PFC (10 Hz y 5 Hz respectivamente), cabe esperar que el análisis sobre ventanas temporales largas provea resultados mas estables que con ventanas cortas. Sin embargo, poder decidir rápidamente en función de los estímulos presentados que respuesta ejecutar es fundamental para la implementación de Interfaces Cerebro Computadora. Aparece entonces, un compromiso entre performance en la decodificación y velocidad de respuesta que debe optimizarse. En esta sección se estudiará el impacto que tiene variar el tamaño de ventana sobre la performance del método Random Forest. Se eligió este método dado que es el que mejor funciona entre los estudiados en secciones anteriores y con el objetivo de tener una cota superior de performance en la decodificación.

Dado que el estímulo es presentado en el milisegundo 4000 de la escala temporal, se amplió paulatinamente el tamaño de ventana desde ese instante (comenzando con una de 100 ms) y se buscó cuál es el mínimo tamaño que resulta en una performance significativamente diferente del azar. La figura 4.18 muestra el resultado de este análisis, se puede observar que con un tamaño de ventana aproximado de 400 ms se logra una performace promedio de 80%.

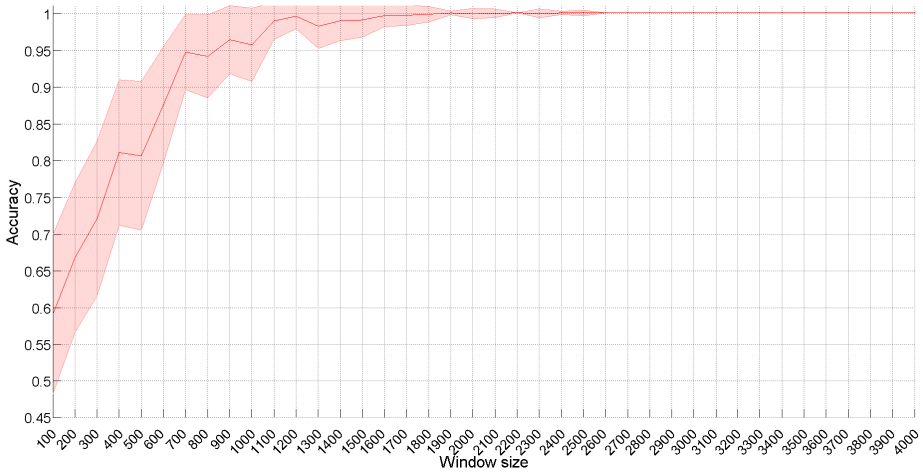

Figure 4.18: Random Forest - Gráfico de la performance en función al tamaño de ventana, desde el milisegundo 4000 - 500 corridas

Surge entonces la pregunta sobre cuando es equivalente utilizar una ventana larga o una ventana deslizante corta. El primer caso asegura obtener toda la información existente a expensas de un mayor costo computacional, con el consecuente incremento en el consumo energético.
La figura 4.19 muestra la performance promedio del análisis con ventanas incrementales y con una ventana de 300 ms deslizante.

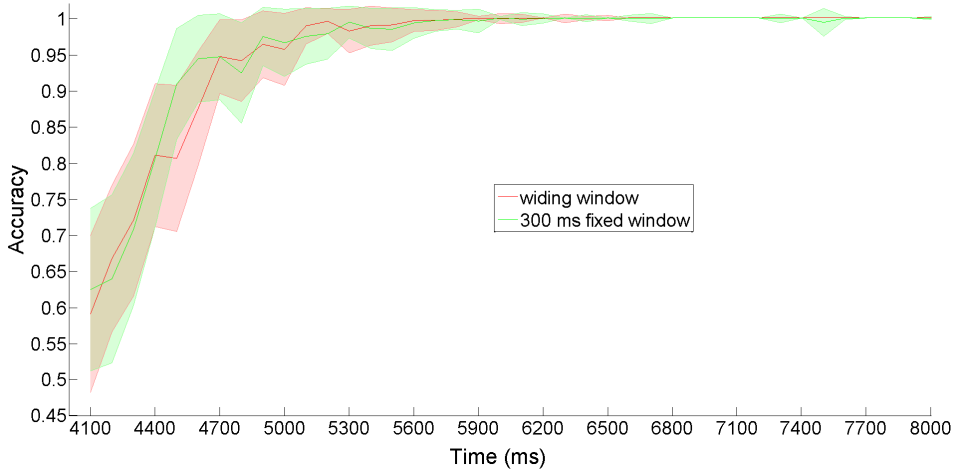

Figure 4.19: Random Forest - Ventana de tamaño fijo en $300ms$ (Verde) vs Ventana ampliándose a medida que avanza el tiempo (Rojo).

4.4.1 Conclusión

Se puede observar que no existen diferencias significativas, más allá del hecho de que en el primer caso la performance fue $100\% \pm 0$ a partir de un tamaño de ventana de 1500 ms aproximadamente. También se puede deducir que es preferible reducir la carga de procesamiento utilizando una ventana corta deslizante, esperando el tiempo necesario para lograr una performance mínima aceptable.

4.5 Predicción del comportamiento

En un experimento futuro, en el momento de registrar los *spikes* de las ratas, se podría predecir si sacarán o no la lengua, incluso antes de la reacción del propio animal.
Con este propósito, se agruparon los trials GOc y NOGOi en una nueva clase llamada L (saca lengua). A su vez, los trials GOi y NOGOc fueron agrupados en una nueva clase llamada NO-L (no saca lengua). El análisis se realizó sobre el Área VTA.
Se prepararon los datos tal y como se especificó en la sección 3.2.2.2, con la salvedad que a la matriz de entrada de los métodos se le agregaron cinco *trials* NOGOi y cinco GOi, indicando que pertenecen a las clases L y NO-L respectivamente.

En la figura 4.20 se muestra la performance para la ventana de amplitud $300ms$ que va desde $4000ms$ a $4300ms$. Se tomó está ventana, debido a que se quiere predecir como responderá el animal al estímulo antes que se ejecute la respuesta motora.

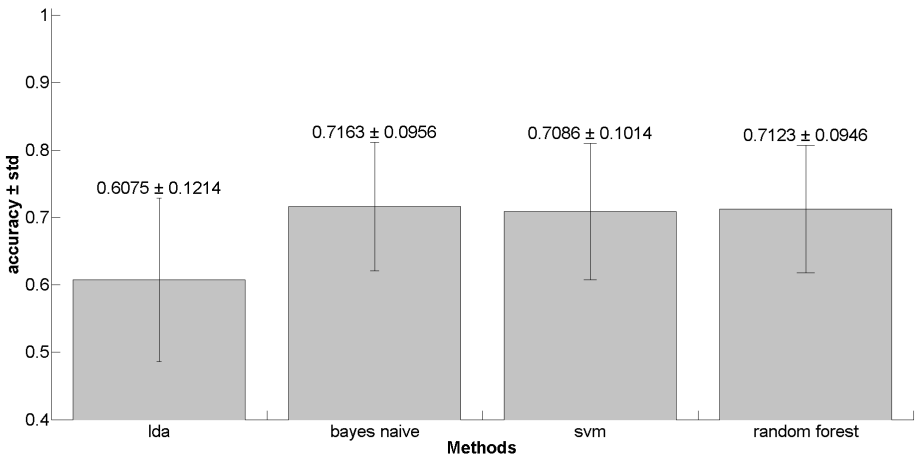

Figure 4.20: Gráfico de la performance en función del tiempo separando las clases L (saca lengua) y NO-L (no saca lengua) para la ventana entre $4000ms$ y $4300ms$

Para obtener la significancia de la clasificación, en la tabla 4.21 se calculó el valor p, el cual indica de las 500 iteraciones realizadas, cuantas veces la performance de clasificación se encuentra por sobre 0.5.

$$p = 1 - \frac{\#\ Perf(M_i) > 0.5}{500} \tag{4.2}$$

	p
LDA	0,192
Bayes Naive	0,016
SVM	0,028
Random Forest	0,014

Figure 4.21: Significancia en el que la performance de un método es mayor a 0.5 para la ventana entre $4000ms$ y $4300ms$

4.5.1 Conclusión

La performance de clasificación para cualquiera de los 3 mejores métodos es aproximadamente 0.7 ± 0.1. La significancia para los mismos es aceptable, demostrando que se alejan del azar.
Si bien la precisión obtenida no es demasiada, hay que tener en cuenta que se está prediciendo la respuesta antes que ocurra con un 70% de probabilidad de acierto.
Ademas se pudo observar que para el caso de los primeros milisegundos, con todas las respuestas posibles, es factible utilizar Bayes Naive.

Conclusiones

En este trabajo de tesis se analizó cuanta información acerca de un estímulo auditivo (previamente presentado) hay contenida en los trenes de potenciales de acción del Área Tegmental Ventral (VTA) y la Corteza PreFrontal (PFC) del cerebro de la rata. La aproximación se realizó mediante el uso de clasificadores con aprendizaje automático, y se comparó la performance del Discriminante Lineal de Fisher (LDA), Bayes Naive, Máquinas de Soporte Vectorial (SVM) y Random Forest. En una primera aproximación, en la sección 3.2.1 se consideró a las neuronas registradas como independientes, donde sólo en pocos casos aislados, se obtuvo una performance de clasificación aceptable ($p < 0.05$). Si un animal logra tasas de acierto mayores al 80% es de suponer que, o bien se trata del consenso de un número grande de neuronas que individualmente aciertan, o es una propiedad emergente de una población que contribuye débilmente a un consenso fuerte en la decisión. Por este motivo se analizó la población de neuronas, para verificar si la interacción entre ellas brinda más información sobre el estímulo presentado que la suma de las contribuciones individuales. Los resultados obtenidos con los cuatro métodos de clasificación empleados confirman esta suposición, la información codificada en la población de neuronas es mayor que la obtenida por contribuciones individuales, llegando a decodificar cual fue el estímulo presentado, aún 3 segundos después de su desaparición, con una tasa de acierto superior al 90%.

Con el objetivo de estudiar la variabilidad de la performance en un análisis trial a trial, se implementó el método de *bootstrapping* y de esta forma se estimó el error estándar a la media. Con esta medida de variabilidad, se pudo analizar que tan buena es la clasificación de los estímulos para los cuatro métodos utilizados. En todos los casos SVM y Random Forest tuvieron performances más altas que LDA y Bayes Naive, a expensas de una mayor complejidad en el cálculo de la superficie que separa las clases.

Desde el punto de vista de la utilidad de los métodos, la eficiencia en

la decodificación debe evaluarse como un compromiso entre la cantidad de neuronas y la cantidad de tiempo necesaria para asegurar una predicción. La cantidad de neuronas que se pueden acceder desde un registro de electrodos profundos, depende principalmente del número de electrodos implantados. Aumentar dicho número acarrea dos problemas: el tamaño del implante crece según el número de electrodos y el sistema inmune reacciona al cuerpo extraño. En la sección 4.3 se estudió el efecto que produce reducir el número de neuronas en el análisis. Si bien la performance en la decodificación decrece, se pudo observar que con un 10% de las neuronas (del área VTA), la misma ronda el 90% (utilizando Random Forest). Para asegurar que este porcentaje no es simplemente el resultado del muestreo (dado que hay neuronas individuales cuya performance en la decodificación es alta) se removieron del análisis las neuronas con performance mayor a 70%, y el resultado no se alteró significativamente. Se puede concluir que el efecto poblacional potencia la decodificación aún en un tamaño de población reducida (una decena de neuronas).

Una de las principales restricciones impuestas sobre un sistema de decodificación portátil y on-line es el consumo energético. Toda interfaz cerebro computadora debe estar alimentada por baterías y su autonomía está directamente ligada al peso e indirectamente relacionada con el consumo. Para reducir el consumo es necesario ahorrar en cómputo, esto se logra minimizando el número de canales que se analizan (número de electrodos) o minimizando el tiempo de análisis. En la sección 4.4 se estudió como el tamaño de la ventana de decodificación altera la performance y se determinó que una ventana de tamaño $400ms$ (a partir de la aparición del estímulo), es suficiente para lograr una performace promedio mayor a 80%. Se comparó esta estrategia con la de ventana deslizante y se determinó que las respectivas performances no fueron diferentes, concluyendo que es más eficiente trabajar con una ventana corta ($300ms$) y no procesar al comienzo del estímulo para ahorrar recursos.

Finalmente, como resultado de este trabajo se ha determinado: Cuál herramienta resulta más apropiada para decodificar la información contenida en un tren de potenciales de acción; que el efecto poblacional tiene más información que la contenida en neuronas individuales; que hay un límite superior en el número de neuronas a partir del cual no se puede obtener más información; que también existe un límite inferior que ronda la decena de neuronas y permite decodificar con buena performance (90%); que con ventanas temporales de aproximadamente $300ms$ se pueden decodificar los estímulos, con un retardo máximo, que ronda los $300ms$, después de la presentación del estímulo; que analizando solamente los $300ms$ posteriores al tono, se puede predecir el comportamiento del animal (si sacará o no la lengua) antes que

el mismo ocurra, con un 70% de probabilidad de acierto.

Quedan como líneas futuras de investigación, el estudio más profundo de cómo los diferentes parámetros que configuran el método más eficiente (*Random Forest*) impactan en la performance de decodificación y la estimación del gasto energético necesario para decodificar instantáneamente esta información de manera portátil.

Set de datos

A.1 Origen

A.1.1 Procedimientos experimentales

Todos los protocolos animales utilizados en este estudio fueron aprobados por el Comité de Ética y Cuidado Animal del Instituto de Biología y Medicina Experimental - Consejo Nacional de Investigaciones Científicas y Técnicas (IBYME - CONICET) , y fueron realizados de acuerdo con la Guía para el Cuidado y Uso de Animales de Laboratorio del Instituto Nacional de Salud (NIH).

A.1.2 Animales

Todas las ratas utilizadas en este experimento fueron ratas macho adultas Long Evans, proporcionadas por el IBYME - CONICET y mantenidas en ciclos de $12h$ de oscuridad/luz. Los animales fueron alojados en jaulas individuales con agua y comida *ad libitum*. Su peso se mantuvo entre $270g$ y $330g$.

A.1.3 Manipulación Pre-quirúrgica

La manipulación pre-quirúrgica comenzó dos semanas antes de la cirugía para que los animales consiguieran habituarse al operador. Los animales fueron levantados por el operador, primero por un corto tiempo ($30s$), aumentando gradualmente hasta $10min$. Fueron puestos en libertad sólo cuando estaban calmos y quietos, teniendo especial cuidado de no liberarlos mientras intentaran escapar de la contención.

A.1.4 Dispositivos de fijación a la cabeza

Los dispositivos de fijación eran piezas de aluminio ($2g$) en forma de cruz, fabricados a partir de una chapa de aluminio de $2mm$ de espesor. Los cuatro

extremos del dispositivo fueron atornillados a dos adaptadores de plástico, que a su vez estaban colocados a los titulares de la barra del oído de un aparato estereotáxico Kopf.

A.1.5 Cirugía

Los animales fueron anestesiados usando ketamina/xilazina ($75mg/kg$, 10 mg/kg, respectivamente). El estado adecuado de la anestesia fue probado mediante la observación de la ausencia del reflejo de la pata. A lo largo de la cirugía , los ojos se cubrieron de ungüento para evitar la desecación . La temperatura corporal se midió mediante una sonda rectal y se mantuvo constante a 37℃, utilizando una almohadilla controlada.
La piel de la cabeza fue afeitada y el cráneo limpiado y desinfectado ($H2O2$, 3%). Una vez que el cráneo estaba expuesto, se perforaron dos agujeros de $2mm$ de diámetro sobre las áreas PFC y VTA (PFC coordenadas : $AP = +2,7mm$, $L = 2mm$, $DV = 2mm$, VTA coordenadas: $AP = -4,8mm$, $L = 1mm$, $DV = 8mm$; Bregma como cero).
Se colocó alrededor de cada agujero una cámara de grabación plástica cilíndrica de $3mm$ de diámetro por $4mm$ de profundidad. Se colocaron dos tornillos de acero inoxidable en cada uno de los huesos parietales (4 tornillos en total), después de ser desinfectados en etanol al 70%. Finalmente, el dispositivo de fijación fue mantenido en su lugar, y fijado a los tornillos y las cámaras de grabación con acrílico dental. Las cámaras de grabación se llenaron con solución antibiótica (neomicina $3,5mg/ml$, polimixina B $5000UI$, gramicidina USP $0,025mg$; OFTAL 3, Holliday - Scott, AR), y se sellaron con un tapón de algodón.
Inmediatamente después de la cirugía las ratas fueron inyectadas subcutáneamente con $1mg/kg$ del analgésico meloxicam (Mobic, Boehringer Ingelheim, AR). Durante el post-operatorio, las ratas fueron tratadas con antibióticos (enrofloxacina en el agua potable a $0,05mg/ml$; Floxacin, Afford, AR) y analgésicos (3 gotas de Tramadol 5% por cada $100ml$ de agua potable; Calmador, Finadiet ,AR) durante al menos 5 días.

A.1.6 Electrodos y adquisición de los datos

El registro extracelular se obtuvo utilizando tetrodos de acuerdo con el siguiente procedimiento.(Gray et al., 1995)
Basicamente se juntaron cuatro alambres enrollados de nychrome de 12µm de diámetro (Kantal, Palm Coast). A continuación, cada tetrodo se introdujo dentro de una cánula de acero inoxidable de 230µm de diámetro externo. Cada alambre se aisló con una funda de poliamida, y su impedancia (a $1KHz$) se ajustó entre $0,5M\Omega$ y $0,8M\Omega$ por electrodeposición de oro en la punta.
Los haces de electrodos fueron construidos juntando tres cánulas con ciano-

acrilato en una configuración de triángulo, con una separación entre ellas de $250\mu m$. Un cable conectado a la cánula de cada conjunto de tetrodos fue utilizado como tierra.
Las señales fueron pre-amplificadas $x10$ y amplificadas $x1000$. Los datos fueron adquiridos con un dispositivo de National Instruments a una frecuencia de muestreo de $30KHz$.

A.2 Formato y distribución

El set de datos fue adquirido en un archivo de matlab dividido en 3 estructuras, las cuales a su vez están subdivididas en 30 sesiones. Cada sesión se subdivide en trials.

Las tres estructuras son:

dataBEH: Contiene los resultados, las respuestas de las ratas a los estímulos auditivos. Se trata de 4 vectores para cada sesión, cada uno asociado a un resultado (GOc, GOi, NOGOc, NOGOi). Cada vector contiene los números de trials para esa sesión que se corresponden con la respuesta que el vector representa.

dataVTA y dataPFC: Contiene los *spikes* registrados del Área Tegmental Ventral y de la Corteza Pre Frontal respectivamente. Poseen neuronas (representadas en matrices) para cada sesión. Estas matrices tienen como filas los trials (representado por sus números consecutivos) y como columnas el tiempo (representado en milisegundos). En su interior hay un 0 o un 1 indicando si en ese par (trial, milisegundo) hubo o no un *spike*.

A.3 Observaciones y análisis

Se observa que el número de neuronas es variable por cada sesión, variando entre 1 y 13, con un promedio de 5 neuronas por sesión para VTA y variando entre 1 y 8, con un promedio de 3 neuronas por sesión para PFC.

El numero de trials también es variable entre 64 y 725 dependiendo de la sesión.

El hecho de que los trials y las neuronas sean variables en cada sesión, es una cuestión a considerar a la hora de decidir la forma de aplicar un algoritmo clasificador.

Se puede observar que la distribución de los resultados es bastante despareja:
GOc: 3169

GOi: 367
NOGOc: 2701
NOGOi: 686

Esto se debe, a que las ratas fueron previamente entrenadas hasta alcanzar un 80% de efectividad antes de empezar a registrar las mediciones.

Una distribución tan despareja puede generar dificultades para los clasificadores.

Configuración de los algoritmos estadísticos de aprendizaje automático

Se utilizó el paquete *Statistics and Machine Learning Toolbox* de la herramienta MATLAB para el uso de los algoritmos LDA, Bayes Naive y SVM. Para Random Forest, no se utilizó la versión nativa de la herramienta debido a la lentitud de la misma. En su lugar se optó una implementación optimizada para MATLAB del código original en R. La misma es pública, de código abierto y se encuentra disponible para descargar en https://code.google.com/p/randomforest-matlab/

Para los comandos de MATLAB, "Xtrain" y "Xtest" serán la matrices de *features* de entrenamiento y test respectivamente. A su vez "Ytrain" e "Ytest" serán los vectores resultados esperados y obtenidos respectivamente.

B.1 LDA

En esté método solamente se configuró el uso de la versión pseudolineal.

Los comandos MatLab para ejecutarlo son:
daClassifier = ClassificationDiscriminant.fit(Xtrain,Ytrain,'discrimType', 'pseudoLinear');
Ytest = daClassifier.predict(Xtest);

B.2 Bayes-Naive

Se entrenó un clasificador Bayes-Naive con una distribución de datos multinomial, dado que esta distribución suele tener buenos resultados para características discretas.
La probabilidad a priori de las clases se configuro como "empírica", es decir que la misma será igual a la frecuencia relativa de distribución de cada clase.

Los comandos MatLab para ejecutarlo son:
nbClassifier = NaiveBayes.fit(Xtrain,Ytrain,'Distribution','mn');
Ytest = nbClassifier.predict(Xtest);

B.3 Análisis con SVM

Se entrenó un clasificador SVM con una función de kernel lineal.
El método utilizado para usar SVM fue SMO (*Sequencial minimal optimization*) (Platt et al., 1998).
El parámetro de ajuste C fue configurado en 1 que es el valor por defecto de la librería utilizada. (No se varió ya que la performance obtenida ronda el 100% sin necesidad de ajustar el método).
Se configuraron 30000 iteraciones del método como máximo por cuestiones de velocidad.

Los comandos MatLab para ejecutarlo son:
opts = statset('MaxIter',30000,'UseParallel','Always');
svmClassifier = svmtrain(Xtrain,Ytrain,'options',opts);
Ytest = svmclassify(svmClassifier,Xtest);

Se probó con otras funciones de kernel (cuadrática, polinomial (grado 3), Gaussiana radial básica y perceptrón multicapa) pero con el set de datos utilizado, la performance de cada una de ellas era igual o peor que la obtenida con un kernel lineal. Dado que el mismo es computacionalmente menos costoso, se utilizó dicho kernel durante todo el trabajo.

B.4 Análisis con Random Forest

Se entrenó un clasificador *Random Forest* con 300 árboles. El "mtry" que indica el número máximo de variables en cada árbol se configuró en 12. El mismo se calculó como el piso de la raíz cuadrada de la cantidad de neuronas. Se aconseja calcular el "mtry" de esta forma por primera vez, y luego si fuera necesario ir variando el valor hasta obtener la mejor performance.

Los comandos MatLab para ejecutarlo son:
rfClassifier = classRF_train(Xtrain,Ytrain, 300);
Ytest = classRF_predict(Xtest,rfClassifier);

Bibliografía

Acir, N. y Güzeliş, C. Automatic spike detection in eeg by a two-stage procedure based on support vector machines. *Computers in Biology and Medicine*, vol. 34(7), páginas 561–575, 2004.

Aflalo, T., Kellis, S., Klaes, C., Lee, B., Shi, Y., Pejsa, K., Shanfield, K., Hayes-Jackson, S., Aisen, M., Heck, C. et al. Decoding motor imagery from the posterior parietal cortex of a tetraplegic human. *Science*, vol. 348(6237), páginas 906–910, 2015.

Aggarwal, V., Mollazadeh, M., Davidson, A. G., Schieber, M. H. y Thakor, N. V. State-based decoding of hand and finger kinematics using neuronal ensemble and lfp activity during dexterous reach-to-grasp movements. *Journal of neurophysiology*, vol. 109(12), páginas 3067–3081, 2013.

Breiman, L. Random forests. *Machine learning*, vol. 45(1), páginas 5–32, 2001.

Brown, E. N., Kass, R. E. y Mitra, P. P. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature neuroscience*, vol. 7(5), páginas 456–461, 2004.

Churchland, M. M., Byron, M. Y., Cunningham, J. P., Sugrue, L. P., Cohen, M. R., Corrado, G. S., Newsome, W. T., Clark, A. M., Hosseini, P., Scott, B. B. et al. Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nature neuroscience*, vol. 13(3), páginas 369–378, 2010.

Cortes, C. y Vapnik, V. Support-vector networks. *Machine learning*, vol. 20(3), páginas 273–297, 1995.

Ding, W. y Yuan, J. Spike sorting based on multi-class support vector machine with superposition resolution. *Medical & biological engineering & computing*, vol. 46(2), páginas 139–145, 2008.

Donders, F. C. On the speed of mental processes. *Acta psychologica*, vol. 30, páginas 412–431, 1969.

DUDA, R. O., HART, P. E. ET AL. *Pattern classification and scene analysis*, vol. 3. Wiley New York, 1973.

EFRON, B. Bootstrap methods: another look at the jackknife. *The annals of Statistics*, páginas 1–26, 1979.

FISHER, R. A. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, vol. 7(2), páginas 179–188, 1936.

FRAIWAN, L., LWEESY, K., KHASAWNEH, N., WENZ, H. y DICKHAUS, H. Automated sleep stage identification system based on time–frequency analysis of a single eeg channel and random forest classifier. *Computer methods and programs in biomedicine*, vol. 108(1), páginas 10–19, 2012.

GRAY, C. M., MALDONADO, P. E., WILSON, M. y MCNAUGHTON, B. Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex. *Journal of neuroscience methods*, vol. 63(1), páginas 43–54, 1995.

HARRIS, K. D., HENZE, D. A., CSICSVARI, J., HIRASE, H. y BUZSÁKI, G. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *Journal of neurophysiology*, vol. 84(1), páginas 401–414, 2000.

HASTIE, T., TIBSHIRANI, R., FRIEDMAN, J., HASTIE, T., FRIEDMAN, J. y TIBSHIRANI, R. *The elements of statistical learning*, vol. 2. Springer, 2009.

HU, J., SI, J., OLSON, B. P. y HE, J. Feature detection in motor cortical spikes by principal component analysis. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 13(3), páginas 256–262, 2005.

HUBEL, D. H. y WIESEL, T. N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, vol. 160(1), páginas 106–154, 1962.

KIM, O. y KANG, D.-J. Fire detection system using random forest classification for image sequences of complex background. *Optical Engineering*, vol. 52(6), páginas 067202–067202, 2013.

KONONENKO, I. Successive naive bayesian classifier. *Informatica*, vol. 17, páginas 167–74, 1993.

LEBEDEV, M. A. y NICOLELIS, M. A. Brain–machine interfaces: past, present and future. *TRENDS in Neurosciences*, vol. 29(9), páginas 536–546, 2006.

LEE, J. W., LEE, J. B., PARK, M. y SONG, S. H. An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics & Data Analysis*, vol. 48(4), páginas 869–885, 2005.

LEHMANN, C., KOENIG, T., JELIC, V., PRICHEP, L., JOHN, R. E., WAHLUND, L.-O., DODGE, Y. y DIERKS, T. Application and comparison of classification algorithms for recognition of alzheimer's disease in electrical brain activity (eeg). *Journal of neuroscience methods*, vol. 161(2), páginas 342–350, 2007.

LIU, J., CHEN, S., TAN, X. y ZHANG, D. Efficient pseudoinverse linear discriminant analysis and its nonlinear form for face recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 21(08), páginas 1265–1278, 2007.

MAROCO, J., SILVA, D., RODRIGUES, A., GUERREIRO, M., SANTANA, I. y DE MENDONÇA, A. Data mining methods in the prediction of dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests. *BMC research notes*, vol. 4(1), página 299, 2011.

MILLER, J., SCHÄFFER, R. y HACKLEY, S. A. Effects of preliminary information in a go versus no-go task. *Acta Psychologica*, vol. 76(3), páginas 241–292, 1991.

MOHRI, M., ROSTAMIZADEH, A. y TALWALKAR, A. *Foundations of machine learning*. MIT press, 2012.

OH, J., LAUBACH, M. y LUCZAK, A. Estimating neuronal variable importance with random forest. En *Bioengineering Conference, 2003 IEEE 29th Annual, Proceedings of*, páginas 33–34. IEEE, 2003.

PAIS-VIEIRA, M., LEBEDEV, M., KUNICKI, C., WANG, J. y NICOLELIS, M. A. A brain-to-brain interface for real-time sharing of sensorimotor information. *Scientific reports*, vol. 3, 2013.

PLATT, J. ET AL. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

POLIKOV, V. S., TRESCO, P. A. y REICHERT, W. M. Response of brain tissue to chronically implanted neural electrodes. *Journal of neuroscience methods*, vol. 148(1), páginas 1–18, 2005.

PRESS, W. H. *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

Quiroga, R. Q., Nadasdy, Z. y Ben-Shaul, Y. Unsupervised spike sorting with wavelets and superparamagnetic clustering. *Neural Computation*, vol. 16, páginas 1661–1687, 2004.

Quiroga, R. Q. y Panzeri, S. Extracting information from neuronal populations: information theory and decoding approaches. *Nature Reviews Neuroscience*, vol. 10(3), páginas 173–185, 2009.

Rao, C. R. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 10(2), páginas 159–203, 1948.

Schmuker, M., Pfeil, T. y Nawrot, M. P. A neuromorphic network for generic multivariate data classification. *Proceedings of the National Academy of Sciences*, vol. 111(6), páginas 2081–2086, 2014.

Valenti, P., Cazamajou, E., Scarpettini, M., Aizemberg, A., Silva, W. y Kochen, S. Automatic detection of interictal spikes using data mining models. *Journal of neuroscience methods*, vol. 150(1), páginas 105–110, 2006.

Velliste, M., Kennedy, S. D., Schwartz, A. B., Whitford, A. S., Sohn, J.-W. y McMorland, A. J. Motor cortical correlates of arm resting in the context of a reaching task and implications for prosthetic control. *The Journal of Neuroscience*, vol. 34(17), páginas 6011–6022, 2014.

Vogelstein, R. J., Murari, K., Thakur, P. H., Diehl, C., Chakrabartty, S. y Cauwenberghs, G. Spike sorting with support vector machines. En *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, vol. 1, páginas 546–549. IEEE, 2004.

Glosario de palabras

Ad Libitum Ad libitum es una expresión del latín que significa literalmente «a placer, a voluntad». 43

Algoritmos estadísticos de aprendizaje automático . 2

Análisis con re-muestreo . 22

Análisis de componentes principales . 7

Aparato estereotáxico Kopf Instrumento diseñado originalmente por David Kopf Instruments en 1963, su facilidad de uso y versatilidad permiten una precisa alineación de los animales pequeños para la colocación estereotáxica de electrodos, micropipetas, cánulas y otros dispositivos. 44

Bootstrapping Es un método de remuestreo propuesto por Bradley Efron en 1979 (Efron, 1979). Se utiliza para aproximar la distribución en el muestreo de un estadístico muestral.. 29

C. Elegans . 1

Clasificador . 14

Comparaciones Apareadas . 23

Corteza de Barriles . 1

Corteza PreFrontal (PFC) . iv, 12

Corteza Premotora . 1

Código neuronal . 1

Decodificadores Lineales . 1

Distribución Multinomial La distribución multinomial es una generalización de la distribución binomial. La misma permite que pueda haber múltiples resultados, en lugar de solo dos posibles, para cada ensayo. La función de probabilidad de la distribución multinomial es:

$$f(x_1 \dots x_k; n, p_1 \dots p_k) = \begin{cases} \dfrac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k} & \text{cuando } \sum_{i=1}^{k} x_i = n \\ \\ 0 & \text{en otros casos} \end{cases} \tag{5.1}$$

. 47

Distribución paramétrica . 25

Electrodos . 2

Error de generalización . 9

Error Estándar a la Media del Bootstrapping (B.S.E.M) . 23

Feature . 2

FPGA Una FPGA (del inglés Field Programmable Gate Array) es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada 'in situ' mediante un lenguaje de descripción especializado. La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas en un chip.. 28

Hiperplano . 4, 5, 7

Inervación dopaminérgica . iv

Inter Spike Interval . 13

Matriz de confusión . viii, 15

Matriz de Covarianza . 5

Matriz mal condicionada . 5

Métodos de Aprendizaje Automático . i, ii, iv

Neuroprostética . 1

Paradigma de discriminación . 12

Potenciales de acción neuronales . i, ii, 54

Pseudoinversa de una matriz . 5

Significancia . 25

Spikes potenciales de acción neuronales. 7, 9, 11, 12

Técnica divide y conquista . 9

Vibrisas . 1

Árboles de decisión . 9

Área cortical motora . 7

Área Tegmental Ventral (VTA) . iv, 12

Áreas corticales . 1