

Herramientas y estrategias de parameter tuning aplicadas a la neurociencia computacional

Andrés Arco López

Universidad de Granada

Grado en Ingeniería Informática

22 de Noviembre de 2022

Trabajo tutorizado por Pablo Martínez Cañada

Contenidos

- 1 Motivación de este TFG
- 2 Métodos
- 3 Resultados
- 4 Conclusiones y vías futuras

Motivación de este TFG

¿Por qué usar modelos de simulación del cerebro?

Actualmente, conocemos menos del 10 % del funcionamiento del cerebro. La neurociencia computacional permite construir modelos de simulación del cerebro, que incluyen cientos de miles de neuronas para:

¿Por qué usar modelos de simulación del cerebro?

Actualmente, conocemos menos del 10 % del funcionamiento del cerebro. La neurociencia computacional permite construir modelos de simulación del cerebro, que incluyen cientos de miles de neuronas para:

- ▶ **Diseñar algoritmos** que descubren cómo resolver problemas neurológicos.

¿Por qué usar modelos de simulación del cerebro?

Actualmente, conocemos menos del 10 % del funcionamiento del cerebro. La neurociencia computacional permite construir modelos de simulación del cerebro, que incluyen cientos de miles de neuronas para:

- ▶ **Diseñar algoritmos** que descubren cómo resolver problemas neurológicos.
- ▶ **Crear nuevas herramientas** para el diagnóstico de enfermedades.

¿Por qué usar modelos de simulación del cerebro?

Actualmente, conocemos menos del 10 % del funcionamiento del cerebro. La neurociencia computacional permite construir modelos de simulación del cerebro, que incluyen cientos de miles de neuronas para:

- ▶ **Diseñar algoritmos** que descubren cómo resolver problemas neurológicos.
- ▶ **Crear nuevas herramientas** para el diagnóstico de enfermedades.
- ▶ Explorar **hipótesis sobre la función del cerebro** que la neurociencia experimental no es capaz con los métodos existentes de registro neuronal.

Objetivos principales

Conseguir **estimar los parámetros del modelo** (por ejemplo, el cociente entre excitación, E , e inhibición, I : E/I) que han dado lugar a las distintas propiedades de la señal del EEG.

Objetivos principales

Conseguir **estimar los parámetros del modelo** (por ejemplo, el cociente entre excitación, E , e inhibición, I : E/I) que han dado lugar a las distintas propiedades de la señal del EEG.

Desarrollar **herramientas** que podrían ser usadas para ayudar en el **diagnóstico clínico de condiciones médicas** y descodificar los parámetros del circuito cortical que son alterados.

Métodos

Modelo neuronal

Para poder llevar a cabo todas las simulaciones se ha usado NEST, que es un simulador de red neuronal de spikes utilizado para simular dinámicas de interacción entre neuronas.

Modelo neuronal

Para poder llevar a cabo todas las simulaciones se ha usado NEST, que es un simulador de red neuronal de spikes utilizado para simular dinámicas de interacción entre neuronas.

- ▶ El modelo cuenta con un total de 5000 neuronas, de las cuales, 4000 excitatorias y 1000 inhibitorias.

Modelo neuronal

Para poder llevar a cabo todas las simulaciones se ha usado NEST, que es un simulador de red neuronal de spikes utilizado para simular dinámicas de interacción entre neuronas.

- ▶ El modelo cuenta con un total de 5000 neuronas, de las cuales, 4000 excitatorias y 1000 inhibitorias.
- ▶ Se estudia el comportamiento de estas neuronas en intervalos de 1 ms (1000 puntos de actividad neuronal en cada simulación).

Modelo neuronal

Para poder llevar a cabo todas las simulaciones se ha usado NEST, que es un simulador de red neuronal de spikes utilizado para simular dinámicas de interacción entre neuronas.

- ▶ El modelo cuenta con un total de 5000 neuronas, de las cuales, 4000 excitatorias y 1000 inhibitorias.
- ▶ Se estudia el comportamiento de estas neuronas en intervalos de 1 ms (1000 puntos de actividad neuronal en cada simulación).
- ▶ Se han obtenido resultados a partir de la variación de 5 parámetros.

Modelo neuronal

```
31  #! =====
32  #! Parameters
33  #! =====
34
35  size_factor = 1.0 # downscale the size of the network
36  weight_factor = 1.0 # increase the synaptic weights to compensate it
37
38  Network_params = {
39      "N_exc": int(4000.0/size_factor),
40      "N_inh": int(1000.0/size_factor),
41      "P": 0.2,
42      "extent": 1.0,
43      "exc_exc_recurrent": 0.178*weight_factor,
44      "exc_inh_recurrent": 0.233*weight_factor,
45      "inh_inh_recurrent": -2.70*weight_factor,
46      "inh_exc_recurrent": -2.01*weight_factor,
47      "th_exc_external": 0.234*weight_factor,
48      "th_inh_external": 0.317*weight_factor,
49      "cc_exc_external": 0.187*weight_factor,
50      "cc_inh_external": 0.254*weight_factor
51  }
```

Figura: Código donde se representan los valores de referencia de las conductancias sinápticas.

Paralelización en el servidor

La paralelización puede mejorar la eficiencia de la ejecución de simulaciones a gran escala aprovechando máquinas multinúcleo/multiprocesador, clústeres de computadoras o supercomputadoras. Para la realización de las simulaciones se ha utilizado:

Paralelización en el servidor

La paralelización puede mejorar la eficiencia de la ejecución de simulaciones a gran escala aprovechando máquinas multinúcleo/multiprocesador, clústeres de computadoras o supercomputadoras. Para la realización de las simulaciones se ha utilizado:

- ▶ Un servidor con sistema operativo Linux que cuenta con 32 núcleos disponibles.

Paralelización en el servidor

La paralelización puede mejorar la eficiencia de la ejecución de simulaciones a gran escala aprovechando máquinas multinúcleo/multiprocesador, clústeres de computadoras o supercomputadoras. Para la realización de las simulaciones se ha utilizado:

- ▶ Un servidor con sistema operativo Linux que cuenta con 32 núcleos disponibles.
- ▶ Memoria RAM de 128GB y almacenamiento de hasta 2TB de información.

Paralelización en el servidor

La paralelización puede mejorar la eficiencia de la ejecución de simulaciones a gran escala aprovechando máquinas multinúcleo/multiprocesador, clústeres de computadoras o supercomputadoras. Para la realización de las simulaciones se ha utilizado:

- ▶ Un servidor con sistema operativo Linux que cuenta con 32 núcleos disponibles.
- ▶ Memoria RAM de 128GB y almacenamiento de hasta 2TB de información.
- ▶ Se han realizado un total de 480 simulaciones (4 minutos por simulación).

Paralelización en el servidor

La paralelización puede mejorar la eficiencia de la ejecución de simulaciones a gran escala aprovechando máquinas multinúcleo/multiprocesador, clústeres de computadoras o supercomputadoras. Para la realización de las simulaciones se ha utilizado:

- ▶ Un servidor con sistema operativo Linux que cuenta con 32 núcleos disponibles.
- ▶ Memoria RAM de 128GB y almacenamiento de hasta 2TB de información.
- ▶ Se han realizado un total de 480 simulaciones (4 minutos por simulación).
- ▶ Conjunto de datos compuesto por archivos: .spikes, .AMPA, .GABA.

Análisis de los datos

Para el cálculo de la señal del electroencefalograma EEG, se ha hecho uso de la siguiente fórmula, que en trabajos previos ha demostrado aproximar bien el EEG (Martínez-Cañada et al. [2021]).

$$EEG = |AMPA| + |GABA|$$

Análisis de los datos

Para el cálculo de la señal del electroencefalograma EEG, se ha hecho uso de la siguiente fórmula, que en trabajos previos ha demostrado aproximar bien el EEG (Martínez-Cañada et al. [2021]).

$$EEG = |AMPA| + |GABA|$$

Ahora se mostrarán los distintos tipos de gráficas obtenidas al variar los parámetros de las simulaciones.

Análisis de los datos

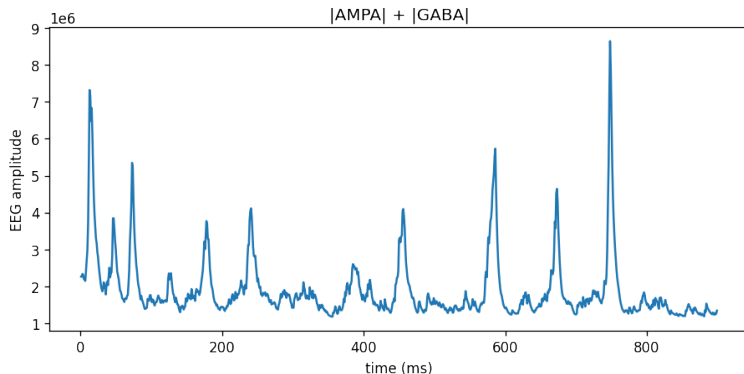


Figura: Gráfica del EEG con valores estándar

Análisis de los datos

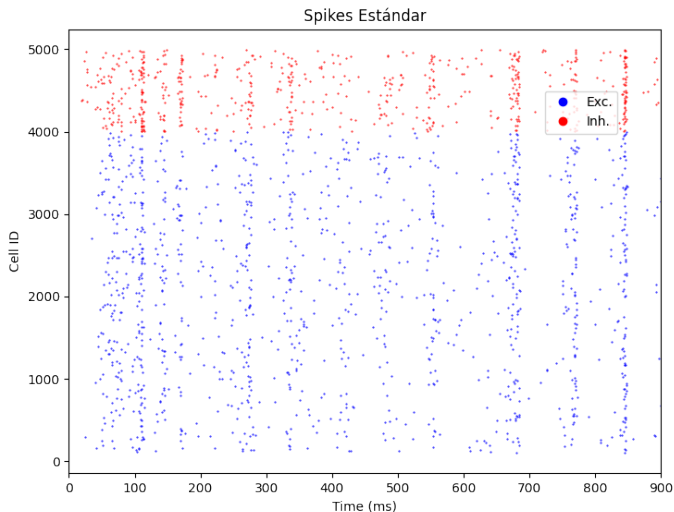


Figura: Gráfica de spikes con valores estándar

Análisis de los datos

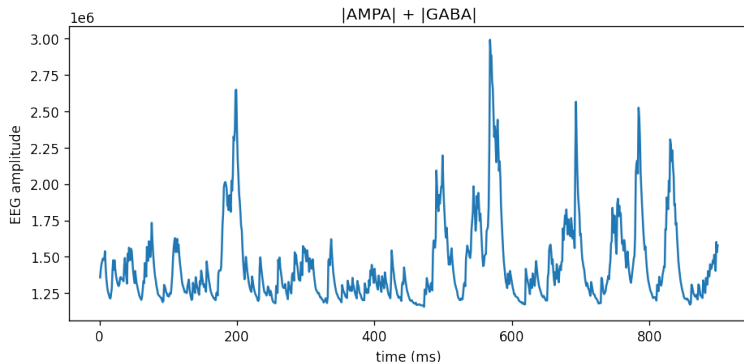


Figura: Gráfica del EEG con parámetro $\text{exc_exc} = -0.005$.

Análisis de los datos

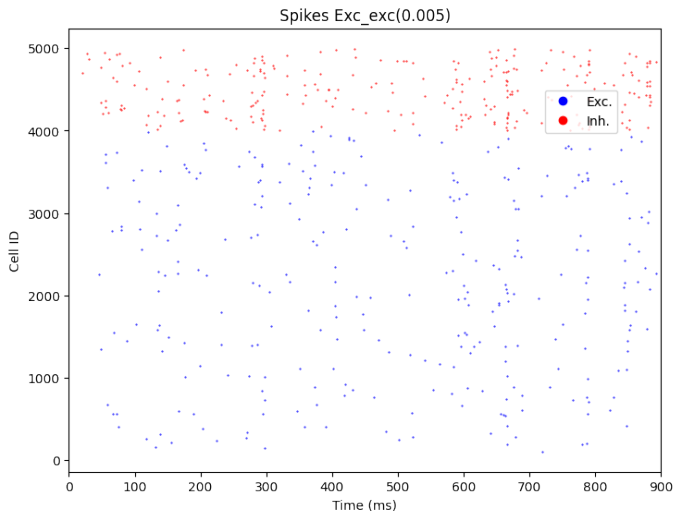


Figura: Gráfica de spikes con parámetro $\text{exc_exc} = -0.005$.

Conjunto de datos del estudio

Estos valores del EEG se han almacenado en una variable X en la que en la que en cada elemento tenemos un vector con todos los valores del EEG de esa simulación y otra variable Y multidimensional que se corresponden con los valores de los ratios de conductancias g calculados como:

$$g = \frac{g_{exc}}{g_{inh}}$$

y con con los valores de la entrada externa del modelo (que puede representar la entrada del tálamo) v_0 .

Algoritmos de Machine Learning

Los algoritmos que se han utilizado para estimar los parámetros del modelo, g y v_0 , han sido:

Algoritmos de Machine Learning

Los algoritmos que se han utilizado para estimar los parámetros del modelo, g y v_0 , han sido:

- ▶ Algoritmo de regresión lineal simple.

Algoritmos de Machine Learning

Los algoritmos que se han utilizado para estimar los parámetros del modelo, g y v_0 , han sido:

- ▶ Algoritmo de regresión lineal simple.
- ▶ Algoritmo de regresión de Ridge (variando el parámetro alfa).

Algoritmos de Machine Learning

Los algoritmos que se han utilizado para estimar los parámetros del modelo, g y v_0 , han sido:

- ▶ Algoritmo de regresión lineal simple.
- ▶ Algoritmo de regresión de Ridge (variando el parámetro alfa).
- ▶ Algoritmo no lineal - K-Nearest Neighbors.

Algoritmos de Machine Learning

Los algoritmos que se han utilizado para estimar los parámetros del modelo, g y v_0 , han sido:

- ▶ Algoritmo de regresión lineal simple.
- ▶ Algoritmo de regresión de Ridge (variando el parámetro alfa).
- ▶ Algoritmo no lineal - K-Nearest Neighbors.
- ▶ Algoritmo no lineal - Red Neuronal.

Algoritmos de Machine Learning

Los algoritmos que se han utilizado para estimar los parámetros del modelo, g y v_0 , han sido:

- ▶ Algoritmo de regresión lineal simple.
- ▶ Algoritmo de regresión de Ridge (variando el parámetro alfa).
- ▶ Algoritmo no lineal - K-Nearest Neighbors.
- ▶ Algoritmo no lineal - Red Neuronal.

Algoritmos de Machine Learning

Se ha hecho uso de la librería scikit-learn en Python, para la ejecución de todos ellos excepto para la Red Neuronal, que ha requerido la utilización de una función externa usada en otros estudios.

Para el estudio de los resultados obtenidos con cada algoritmo se han realizado 100 repeticiones de 10-fold cross validation, en las cuales, se seleccionan aleatoriamente los datos de entrenamiento y los datos de test.

Resultados

Regresión lineal simple

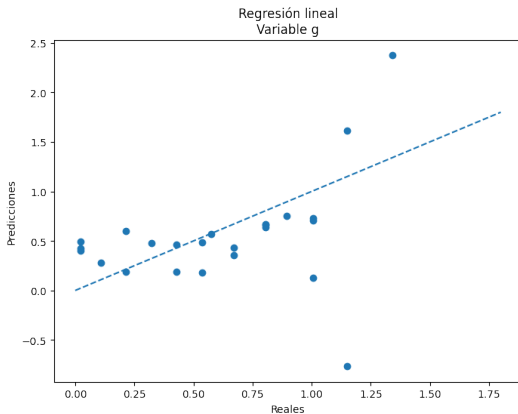


Figura: Comparativa entre los valores reales y las predicciones de la variable g para 1 fold de 10-fold cross validation.

Regresión lineal simple aplicando autocorrelación

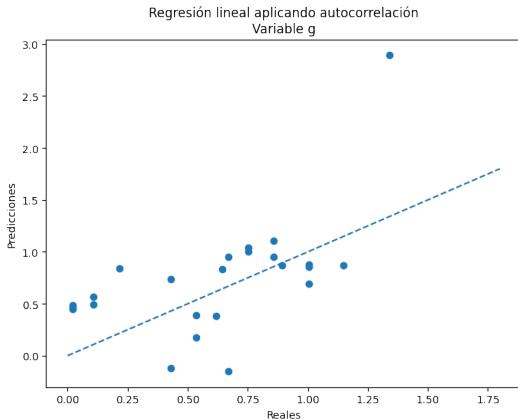


Figura: Comparativa entre los valores reales y las predicciones de la variable g después de aplicar autocorrelación para 1 fold de 10-fold cross validation.

Regresión de Ridge (alfa = 0.5)

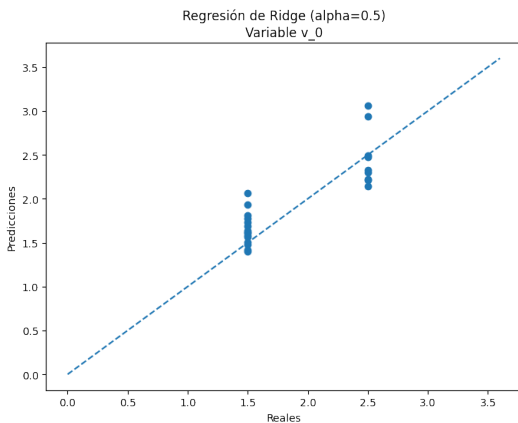


Figura: Comparativa entre los valores reales y las predicciones de la variable v_0 para 1 fold de 10-fold cross validation.

Regresión de Ridge aplicando autocorrelación ($\alpha = 0.5$)

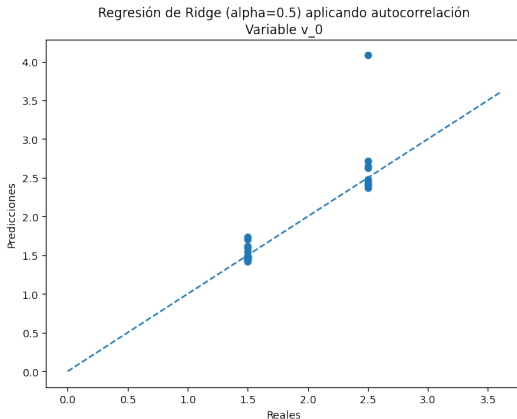


Figura: Comparativa entre los valores reales y las predicciones de la variable v_0 después de aplicar autocorrelación para 1 fold de 10-fold cross validation.

K-Nearest Neighbors

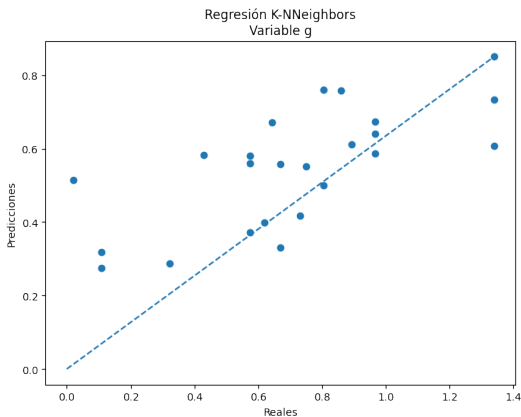


Figura: Comparativa entre los valores reales y las predicciones de la variable g para 1 fold de 10-fold cross validation.

K-Nearest Neighbors aplicando autocorrelación

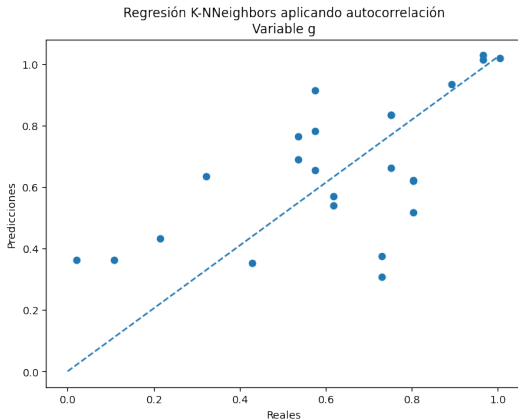
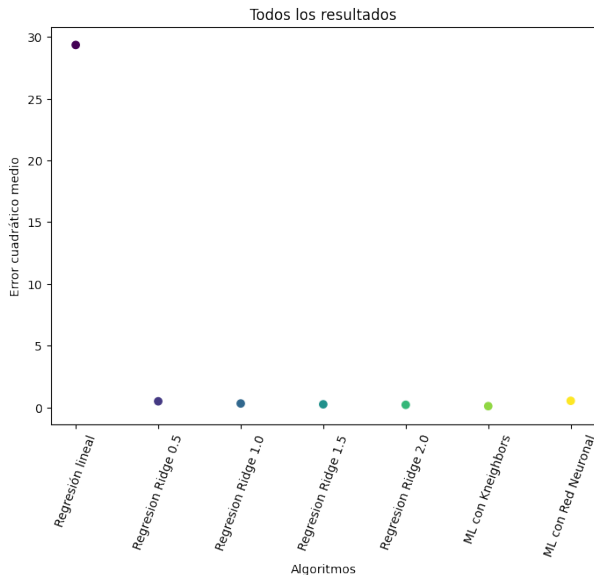


Figura: Comparativa entre los valores reales y las predicciones de la variable g después de aplicar autocorrelación para 1 fold de 10-fold cross validation.

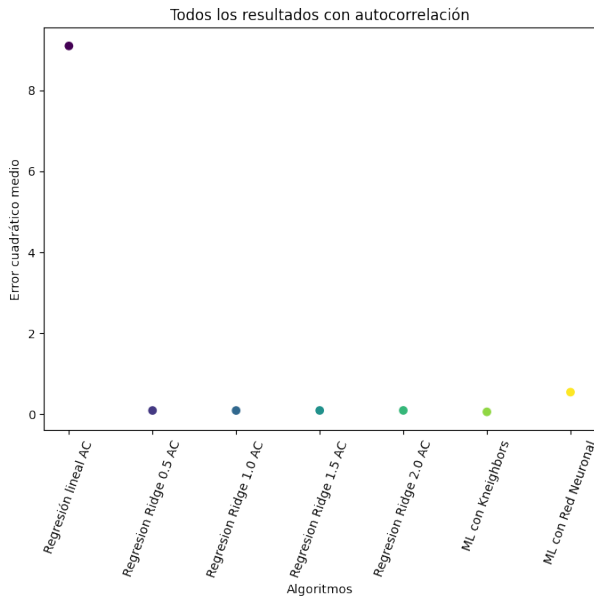
Resultados finales

Tras realizar un total de 1000 ejecuciones mediante cada uno de los algoritmos (excepto en el de Red Neuronal, debido a su complejidad) y almacenar promedio de todos los errores cuadráticos medios calculados en cada ejecución, entre los valores de test y los valores obtenidos en las predicciones, se han obtenido los siguientes resultados.

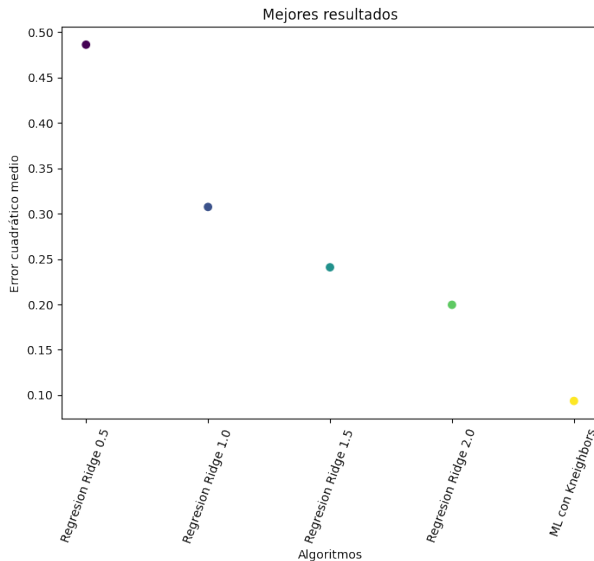
Todos los resultados



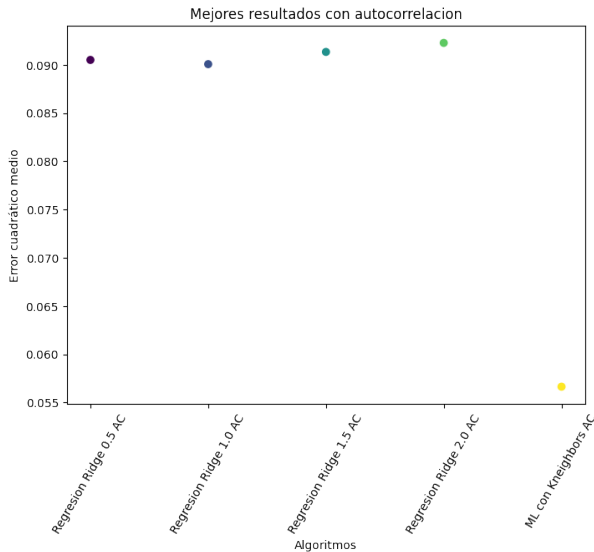
Todos los resultados con autocorrelación



Mejores resultados



Mejores resultados con autocorrelación



Conclusiones y vías futuras

Conclusiones

- Hemos obtenido un gran conjunto de datos, entre ellos la señal del electroencefalograma (EEG), que es una de las señales no-invasivas más conocida en el ámbito clínico.

Conclusiones

- Hemos obtenido un gran conjunto de datos, entre ellos la señal del electroencefalograma (EEG), que es una de las señales no-invasivas más conocida en el ámbito clínico.
- Analizando estas señales hemos sido capaces de detectar diferentes tipos de gráficas, algunas muy realistas.

Conclusiones

- Hemos obtenido un gran conjunto de datos, entre ellos la señal del electroencefalograma (EEG), que es una de las señales no-invasivas más conocida en el ámbito clínico.
- Analizando estas señales hemos sido capaces de detectar diferentes tipos de gráficas, algunas muy realistas.
- Hemos estimado variaciones del parámetro E/I , que según la literatura están relacionadas con patologías como la epilepsia o el autismo.

Conclusiones

- Hemos obtenido un gran conjunto de datos, entre ellos la señal del electroencefalograma (EEG), que es una de las señales no-invasivas más conocida en el ámbito clínico.
- Analizando estas señales hemos sido capaces de detectar diferentes tipos de gráficas, algunas muy realistas.
- Hemos estimado variaciones del parámetro E/I , que según la literatura están relacionadas con patologías como la epilepsia o el autismo.
- Gracias a estos algoritmos y usando un modelo del cerebro, hemos podido leer señales del EEG y producir estimaciones de cambios en E/I , obteniendo los mejores resultados con el algoritmo de K-Nearest Neighbors.

Limitaciones

- El modelo cerebral usado es un modelo simplificado de una pequeña parte del cerebro y no tiene en cuenta conexiones con otras regiones del cerebro.

Limitaciones

- El modelo cerebral usado es un modelo simplificado de una pequeña parte del cerebro y no tiene en cuenta conexiones con otras regiones del cerebro.
- El modelo usa neuronas de tipo de LIF, que son neuronas simplificadas, que facilitan el cálculo del potencial con vistas a su implementación hardware

Limitaciones

- El modelo cerebral usado es un modelo simplificado de una pequeña parte del cerebro y no tiene en cuenta conexiones con otras regiones del cerebro.
- El modelo usa neuronas de tipo de LIF, que son neuronas simplificadas, que facilitan el cálculo del potencial con vistas a su implementación hardware
- La fórmula utilizada para generar el EEG, es una fórmula simplificada, con la que hemos sido capaces de mantenernos cercanos a los valores esperados, pero no tienen en cuenta otros factores.

Trabajo futuro

- Mejorar el modelo superando las limitaciones mencionadas anteriormente, haciendo uso de una mayor cantidad de parámetros.

Trabajo futuro

- Mejorar el modelo superando las limitaciones mencionadas anteriormente, haciendo uso de una mayor cantidad de parámetros.
- Aplicar el modelo a registros reales de señales del EEG tomados en pacientes con patologías clínicas.

Bibliografía fundamental

- Noei S. Martínez-Cañada, P. and S. Panzeri. Methods for inferring neural circuit interactions and neuromodulation from local field potential and electroencephalogram measures. Brain Informatics, 8, 2021.
- Pablo Martínez-Cañada, Torbjørn V. Ness, Gaute T. Einevoll, Tommaso Fellin, and Stefano Panzeri. Computation of the electroencephalogram (eeg) from network models of point neurons. PLOS Computational Biology, 17:1–41, 04 2021.
- Alberto Mazzoni, Stefano Panzeri, Nikos K. Logothetis, and Nicolas Brunel. Encoding of naturalistic stimuli by local field potential spectra in networks of excitatory and inhibitory neurons. PLOS Computational Biology, 4, 12 2008.

Bibliografía fundamental

- Whittingstall K. Brunel N. Logothetis N. K. Panzeri S. Mazzoni, A. Understanding the relationships between spike rate and delta/gamma frequency bands of lfps and eegs using a local cortical network model. page 956–972, 2010.
- Merzenich M. M. Rubenstein, J. L. Model of autism: increased ratio of excitation/inhibition in key neural systems. page 255–267, 2003.

Gracias por su atención.