Memoria de Proyecto

Decodificación cerebral basado en LSTM

Isabel Amaya Rodríguez

Héctor Rivas Pagador

Ángel Serrano Alarcón

# Índice

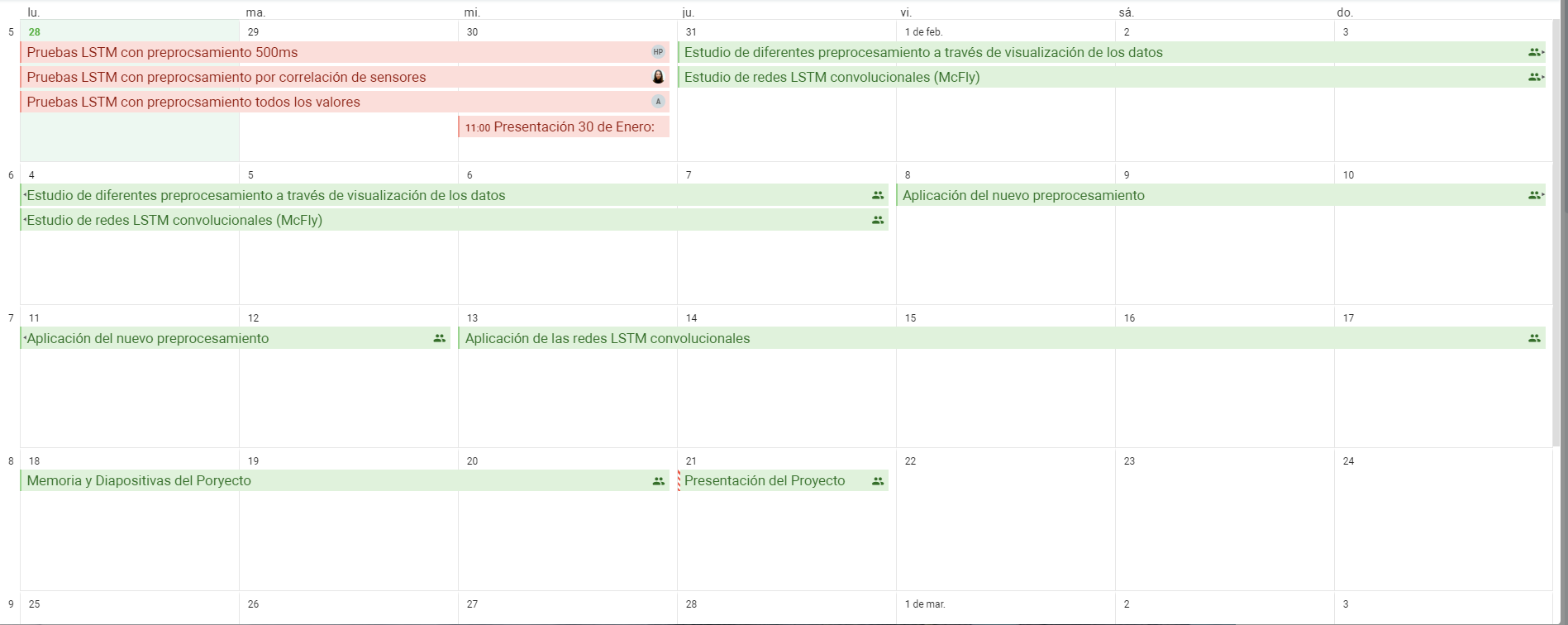
1. Objetivos Finales
2. Planificación
3. Resultados de participantes de la competición
4. Solución del problema hasta el momento
   1. Preprocesamiento
   2. Modelo
   3. Evaluación de resultados
5. Conclusiones
6. Bibliografía

# Objetivos Finales

Entre los objetivos que nos hemos marcado en este proyecto, los principales son los siguientes:

* **Solución al problema mediante diferentes modelos:** realizar una solución con diferentes modelos de clasificación y realizar una comparación de los datos obtenidos
* **Crear un modelo de predicción de los estímulos visuales mediante redes LSTM:** realizar una solución mediante redes recurrentes LSTM con la librería de Keras.
* **Utilización de la arquitectura LSTM convolucional.**

# Planificación



Para la planificación del proyecto, ponemos como objetivo tener 2 “plazos de entrega”.

* 30 de Enero: en este día, el principal objetivo es haber realizado las siguientes tareas:
  + Modelo básico con redes LSTM al problema planteado
  + Aplicación de varios modelos de clasificación al problema planteado
  + Realización de 3 diferentes métodos de preprocesamiento sobre los datos del problema
  + Realización de pruebas del modelo básico de redes LSTM sobre los 3 diferentes datos resultados de haberles aplicado lo métodos de preprocesamiento

La realización de las diferentes tareas las realizamos todos los integrantes del grupo juntos, excepto la realización de las pruebas del modelo básico de redes LSTM, donde cada uno realiza una prueba sobre un tipo de preprocesamiento diferente.

* 21 de Febrero: para este día, queremos haber realizado las siguientes tareas:
  + Mejorar la solución con redes LSTM realizando diferentes pruebas para conseguir unos parámetros óptimos para la red LSTM.
  + Aplicación de redes convolucionales LSTM.
  + Estudio de los datos para agregar algún preprocesamiento de los mismo que nos permita mejorar la solución

# Resultados de participantes de la competición

En el sitio web de la competición, no hay ningún Kernel, pero se encuentran compartidos los modelos de los 3 equipos que quedaron en las primeras 3 posiciones de la competición.

Las accuracy obtenidas por los 3 primeros equipos son las siguientes:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | — | Alexandre Barachant |  | 0.75501 |  |  |
| 2 | — | Heikki Huttunen |  | 0.72667 |  |  |
| 3 | 21 | Nathan Hammes |  | 0.71316 |  |  |

**Alexandre Barachant:** la solución de este equipo está compuesta por dos pasos de clasificación. El primero es supervisado y utiliza todos los datos de train de los sujetos. Este modelo se aplica en cada sujeto de train para obtener una primera estimación de las etiquetas de test. El segundo paso es aprendizaje no supervisado y se aplica de forma independiente en cada sujeto test. Utiliza las etiquetas del primer paso como inicialización de un algoritmo iterativo sin supervisión, similar a un clustering de k-means.

Respecto al preprocesamiento, coge los valores correspondientes al intervalo de tiempo 0.5s-1.5s, ya que a partir de 0.5s es cuando se empieza a visualizar estímulos. También realiza un filtro de paso banda entre 0-20Hz.

**Heikki Huttunen**: Este equipo realizó una combinación jerárquica de regresión logística y random forest. La primera capa consiste en una colección de 337 clasificadores de regresión logística, donde cada uno utiliza datos de un solo sensor o datos de un solo punto de tiempo. Las resultantes estimaciones de probabilidades se alimentan de un random forest de 1000 árboles, el cual hace la decisión final.

**Nathan Hammes**: este equipo realizó una clasificación agrupada utilizando máquinas de vectores de soporte estándar de Lasso y SVMlight. Realiza un paso de reclasificación realizando una clasificación teniendo como entrada la salida de 3 clasificadores. Respecto al preprocesamiento, filtran los datos mediante un Filtro de elimina banda a 50 Hz para eliminar el ruido eléctrico, luego un filtro paso bajo y realizan un promediado. Eliminan los sensores del 1 al 162 y sólo cogen la información correspondiente al intervalo de tiempo 0-0.5s.

# Solución del problema hasta el momento

En este apartado se explica los diferentes pasos que hemos realizado para tener una primera aproximación a la solución del problema.

## Preprocesamiento

Tras realizar un estudio de las soluciones de los diferentes participantes de la competición, hemos decido realizar 3 tipos de preprocesamiento para posteriormente aplicar dichos datos en un modelo de red LSTM.

Estos preprocesamientos realizados nos sirven para tener una primera visualización y comparar los resultados para posteriormente decidir el tipo de preprocesamiento a utilizar.

* **Coger los valores de las series temporales a partir de 500ms y normalización**: el primer clasificado de la competición observó que los estímulos empiezan a aparecer a partir de los 500ms, por lo que filtramos estos valores y nos quedamos con los restantes valores que corresponden a 1s. Normalizamos los valores mediante z-score.
* **Coger todos los valores y los normalizamos:** realizamos la prueba con todos los valores para comparar los resultados con los anteriores.
* **Eliminar los sensores con mayor correlación:** uno de los puntos que nos planteamos fue estudiar la correlación que tienen los sensores a la hora de coger los valores de las series temporales. Si varios sensores tienen alta correlación, significa que nos está dando la misma información a la hora de visualizar los estímulos, por lo que estos sensores podríamos filtrarlos.

Lo primero que hicimos fue coger todos los valores de correlación de los sensores a partir de los valores de sus correspondientes series temporales. Estos datos de correlación los visualizamos en un **diagrama de calor** [Figura 1].

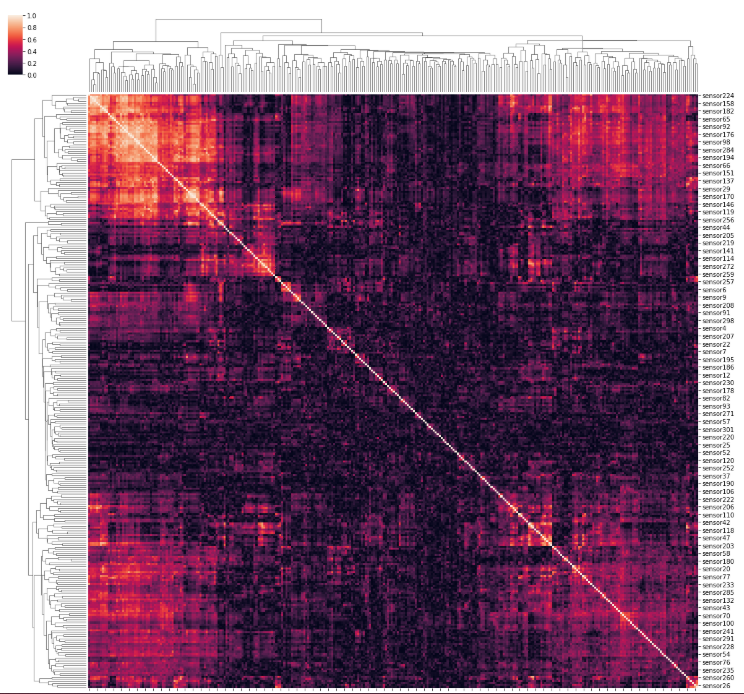
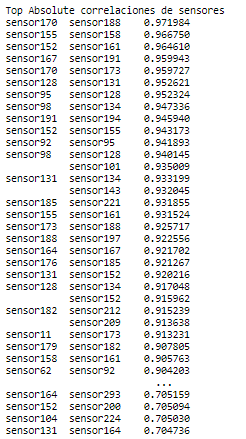


Figura 1 - Diagrama de calor de correlaciones

En este diagrama observamos que varios sensores tienen una alta correlación. A parte de esto, creamos una lista con las correlaciones más altas que hay:



Como podemos comprobar, hay sensores con una alta correlación. Mirando la lista general de correlaciones, observamos que las correlaciones más altas se encuentran en los primero 180 sensores, por lo que decidimos filtrar estos y quedarnos con los 126 sensores restantes.

## Modelo con red LSTM

Para crear el modelo, utilizamos la librería de Keras. A partir de esta librería, creamos un modelo secuencial que nos permite crear y agregar capas de modelos.

Utilizamos dos modelos para realizar las primeras pruebas. Como primera capa utilizamos una red LSTM que tendrá como entrada los datos y clases de entrenamiento del dataset.

El entrenamiento de la red LSTM se realiza mediante validación cruzada con 7 épocas (iteraciones)

Esta red está conectada a una función de activación sigmoide que nos devuelve un resultado binario (0 o 1)

## Evaluación

# Conclusiones

# Bibliografía

[1] URL del competición en Kernel: <https://www.kaggle.com/c/decoding-the-human-brain>

[2] Repositorio oficial de la competición: <https://github.com/FBK-NILab/DecMeg2014>

[3] Modelos del 1º y 3º clasificado: <https://www.kaggle.com/c/decoding-the-human-brain/discussion/9913>

[4] Modelo del 2º clasificado: <https://github.com/mahehu/decmeg>

[5] Librería red LSTM en Keras: <https://keras.io/layers/recurrent/#lstm>