

## Proyecto EL4106 - Semestre Otoño 2022

Profesor: Javier Ruiz del Solar  
Auxiliar: Patricio Loncomilla

Publicación enunciado: 10 de Junio de 2022  
Informe intermedio: 26 de Junio de 2022  
Informe y presentación final: 12 de Julio de 2022

El objetivo de este proyecto es implementar un clasificador de gestos de manos a partir de señales mioeléctricas. Se usará el conjunto de datos *EMG data for gestures Data Set*.

Este conjunto de datos contiene mediciones de personas realizando distintos gestos con las manos (7 gestos incluyendo la posición de reposo). Los datos a usar corresponden a capturas de ocho señales, tomadas en el antebrazo usando electromiogramas (EMG), además de una etiqueta en cada instante de tiempo indicando el gesto realizado. Un total de 36 sujetos realizan los gestos indicados, con pausas intermedias. Para cada sujeto se realizaron dos capturas. Información más detallada se puede encontrar en el sitio web del conjunto de datos original (<https://archive.ics.uci.edu/ml/datasets/EMG+data+for+gestures>).

Para efectos de la tarea, los 36 sujetos fueron divididos en 30 sujetos para entrenamiento y validación, y 6 sujetos para el conjunto de prueba. Se entregarán las capturas completas para el conjunto de entrenamiento y validación. Sin embargo, para el conjunto de prueba se entregará un conjunto de 672 ventanas, cada una de tamaño 800 x 8. Las etiquetas de las ventanas del conjunto de prueba no serán entregadas.

Además de lo anterior, la clase 0 será ignorada debido a que corresponde a zonas no etiquetadas, mientras que la clase 7 también será ignorada, ya que no todos los sujetos realizan la actividad correspondiente.

**Se pide diseñar, implementar y entrenar un sistema de clasificación que reciba las 8 señales mioeléctricas (se usan 8 electodos) y que clasifique el gesto realizado.** Exista la posibilidad de que usted decida/determine que no necesariamente las 8 señales deben ser usadas, sino un número menor.

Para lograr esto necesitará (todos los pasos son obligatorios y deben ser reportados):

- (1) Implementar un código que permita leer la base de datos de entrenamiento + validación.
- (2) Implementar un código que permita leer las ventanas del conjunto de prueba. Cada fila del conjunto de prueba corresponde a una ventana de tamaño 800, con 8 canales, concatenados horizontalmente. Es decir, cada fila tiene tamaño 6400. El código debe reconstruir las ventanas con los 8 canales (dimensión 800 x 8) para poder trabajar con ellas. Se recuerda que las etiquetas del conjunto de prueba no están disponibles.
- (3) Definir conjuntos de entrenamiento y validación. Ambos conjuntos deben contener personas distintas.
- (4) Graficar algunas señales del conjunto de datos. Esto puede facilitar la comprensión del problema a resolver.
- (5) Investigue el aporte de cada señal de entrada en la solución del problema, y en caso de ser necesario, tome decisiones respecto de la exclusión de alguna(s) de esta(s). Debe justificarse claramente el hecho de eliminar señales.
- (6) Evaluar si se requiere aplicar algún tipo de filtrado a las señales (pasabajos/pasaaltos/pasabandas) o algún tipo de pre-procesamiento. En el caso que se requiera, implementar estas operaciones.
- (7) Implementar un código que permita generar ventanas a partir del conjunto de entrenamiento y de validación. Se recomienda inicialmente un tamaño de ventana 800, con un paso de tamaño 250, considerando los 8 canales disponibles. El paso corresponde a la separación usada para generar

ventanas consecutivas. Se recomienda considerar ventanas que contengan un solo label, y que correspondan a una única captura. **Las ventanas que contengan labels 0 o 7 deben ignorarse.**

- (8) Definir las características a ser utilizadas e implementar su cálculo. Se recomienda usar características calculadas sobre cada canal, como las medias, varianzas, valores mínimos y máximos, rangos de las variables, etcétera. Posibles características se indican en [1] (ver Tabla 3). Se recomienda normalizar las características, usando las estadísticas del conjunto de entrenamiento (no el de validación).
- (9) Aplique algún método de selección de características para determinar las características a ser usadas para entrenar el clasificador.
- (10) Elegir y entrenar un clasificador o cascada de clasificadores para determinar el gesto realizado, usando los conjuntos de entrenamiento y validación. Usted es libre de utilizar el o los clasificadores que le parezcan más apropiados y que resuelvan de mejor manera el problema (Bayes, Redes Neuronales, SVM, Adaboost, Random Forest, redes deep, árbol de clasificadores, etc.).
- (11) La ventana propuesta inicialmente tienen un ancho de 800 muestras. Analice disminuir el ancho de la ventana. Analice también aumentar o disminuir el paso usado entre las ventanas. En el caso en que se use un tamaño de ventana reducido, se debe analizar una estrategia para clasificar las ventanas del conjunto de prueba.
- (12) Aplique el mejor clasificador obtenido sobre el conjunto de prueba. Luego, suba el archivo con las etiquetas a la plataforma Kaggle (se indicará el link de la competencia en el foro, así como las instrucciones de uso).

**Importante:** No necesariamente los pasos (5)–(11) deben realizarse en el orden solicitado, pues pudiera ser que haya que iterar entre estos.

**Importante:** En el caso en que se quiera usar redes convolucionales 2D, no es necesario abordar la generación de características, selección de características ni el entrenamiento de clasificadores como ítems separados, ya que la red convolucional reemplaza dichos pasos. Los alumnos que elijan esta metodología tendrán un “bono” de 0,5 puntos en nota final del proyecto (con saturación en 7,0), y además en caso que su nota promedio en tareas sea inferior a 5,5, no deberán dar examen.

El proyecto debe realizarse en Python 3, con bibliotecas de *machine learning* estándares (scikit-learn, pandas, pytorch, tensorflow, etc.), instaladas en forma local o usando colab. Los archivos con los datos necesarios para desarrollar el proyecto serán subidos a la plataforma u-cursos.

Entregas:

1. Avance – 26 de Junio (30% de la nota del proyecto)
  - Presentación en archivo formato Powerpoint o PDF, donde se expliquen las metodologías que utilizará para resolver el problema, las características usadas, las señales seleccionadas y el método de selección de características implementado, además de resultados preliminares de clasificación obtenidos usando las características seleccionadas en el conjunto de prueba. En el caso de usar redes CNN, mostrar como los datos de los 8 canales son utilizados por la red y resultado preliminares.
2. Final – 12 de Julio (70% de la nota del proyecto, de los cuales el 20% se derivará del resultado obtenido en kaggle y el 50% del informe escrito y presentación oral)
  - Presentación oral (grabada) e informe escrito, donde se explique la metodología utilizada, los resultados finales obtenidos con los clasificadores escogidos, las mejoras de los clasificadores luego de filtrar las características con el método de selección utilizado, las métricas (matriz de confusión y accuracy) sobre el conjunto de validación, el score en el conjunto de prueba, los problemas encontrados y posibles mejoras al sistema.

Las presentaciones orales grabadas deben ser de máximo 3 minutos. Recuerde considerar el público objetivo, no pierda tiempo explicando la base de datos u otra información que todos conocen. Las presentaciones deben ser entregadas de forma electrónica (en u-cursos), así como el informe final en

formato PDF. Además, el informe final en PDF debe ser subido a la plataforma Turnitin. Para la evaluación final se deberán entregar todos los códigos, los cuales deben ser subidos a u-cursos. Incluir un corto archivo de texto explicando cómo se utiliza su programa.

**El proyecto debe realizarse en forma individual.** Las entregas atrasadas serán penalizadas con un punto de descuento por cada día de atraso. Se abrirá un tema en el foro para consultas.

**Importante:** La evaluación considerará el correcto funcionamiento del programa, **la calidad del clasificador obtenido**, la inclusión de los resultados en el informe, la calidad de los experimentos realizados y de su análisis, así como la forma, prolijidad y calidad del mismo.

**Recomendación:** Si se quiere usar pytorch (sin usar redes convolucionales), se recomienda trabajar inicialmente con scikit-learn para poder familiarizarse con el problema y poder realizar tareas como la selección de características, señales y normalización de datos.

**Nota:** Las etiquetas del conjunto de prueba no están disponibles. Se usará Kaggle para implementar un *leaderboard* público y otro privado que permita rankear los resultados subidos por los alumnos. Dado que Kaggle no entrega matrices de confusión, almacene en su computador el archivo con las etiquetas predichas antes de subirlas. Se debe notar que durante el desarrollo del proyecto sólo se interactuará con las etiquetas reales a través de los *leaderboards*.

[1] Macro-Class Selection for Hierarchical K-NN Classification of inertial sensor data.  
<http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=4A350743038752220482F311B681E303?doi=10.1.1.228.9382&rep=rep1&type=pdf>