

Proyecto EL4106 - Semestre Otoño 2019

Profesor: Javier Ruiz del Solar

Auxiliar: Patricio Loncomilla

Ayudantes: Gabriel Azócar, Nicolás Cruz, Francisco Leiva, Giovanni Pais

Publicación enunciado: Jueves 06 de Junio de 2019

Presentación oral: Martes 25 y jueves 27 de Junio de 2019

Informe técnico: Viernes 12 de Julio de 2019

El objetivo de este proyecto es implementar un clasificador de gestos de manos. Se usará el conjunto de datos "EMG data for gestures Data Set", la cual se puede encontrar en el siguiente sitio web:

<https://archive.ics.uci.edu/ml/datasets/EMG+data+for+gestures>

Este conjunto de datos contiene mediciones de personas realizando distintos gestos con las manos (7 gestos más posición de reposo). Los datos a usar corresponden a capturas de ocho señales, tomadas en el antebrazo usando electromiografía (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 indicado.

Se les pide diseñar, implementar, entrenar y validar un sistema de clasificación que reciba las 8 señales de entrada, una por cada sensor y que clasifique el gesto realizado. Exista la posibilidad de que usted determine que no necesariamente las 8 señales deben ser usadas, sino un número menor.

Para lograr el objetivo del proyecto deberá:

- (1) Elegir conjuntos de entrenamiento, validación y test. El conjunto de test debe corresponder a los 8 últimos sujetos del conjunto de datos. El alumno tiene libertad para decidir cómo usar los conjuntos de entrenamiento y validación. Sin embargo, el **conjunto de test no debe usarse** de ningún modo para decidir si un clasificador es mejor que otro. El conjunto de test sólo se usará en la parte final de la tarea, ya que representa datos que sólo estarán disponibles cuando el sistema funcione después de haber sido diseñado.
- (2) Realizar un programa que permita leer la base de datos. Esta consiste en un conjunto de archivos .txt (separados por comas). Se recomienda leerlo usando el paquete *pandas*. Los datos con label '7' deben ser ignorados, ya que hay muy pocas muestras de esta clase en la base de datos.
- (3) Graficar algunas características del conjunto de datos, así como la señal que indica las etiquetas. Esto puede facilitar la comprensión del problema a resolver.
- (4) Investigue el aporte de cada señal de entrada y en caso de ser necesario tome decisiones respecto de la exclusión de alguna(s) de esta(s).
- (5) Evaluar si se requiere aplicar algún tipo de filtrado a las señales o algún tipo de pre-procesamiento. En el caso que se requiera, implementar estas operaciones.
- (6) Definir las características a ser utilizadas. Se recomienda usar características calculadas sobre cada canal, como las medias, varianzas, valores mínimos y máximos, rangos de las variables, etcétera (ver a modo de ejemplos las características usadas en [1]). El extractor deberá considerar distintos tamaños de ventana de análisis y distintos pasos entre ventanas. Para las pruebas preliminares, se recomienda un ancho de ventana 200 y un paso de 200 entre las ventanas. Se recomienda normalizar las características, usando las estadísticas del conjunto de prueba (no el de validación).
- (7) Como el número de características puede ser muy grande evalúe la necesidad de hacer selección de características y/o de aplicar algún método de reducción de dimensionalidad. En el caso que estime es necesario realizar la selección de características y/o la reducción de dimensionalidad, implementarla.
- (8) 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, etc.). Note que las ventanas pueden contener gestos o pausas, en este punto las pausas deben ignorarse.
- (9) Las señales contienen tanto gestos como pausas entre gestos consecutivos. Se debe diseñar e implementar un detector de gestos (clasificador gesto versus pausa), el cual es un clasificador que pueda predecir cuando hay una pausa (clase igual a cero) o un gesto (clase distinta de cero) en una ventana dada. Al igual que en el punto (8), usted es libre de usar los clasificadores más apropiados y que resuelvan de mejor manera el problema.
 - (10) La ventana propuesta inicialmente es de un ancho de 200 muestras. Analice disminuir o aumentar el ancho de la ventana. Analice también aumentar o disminuir el paso usado entre las ventanas.
 - (11) Usando el mejor clasificador obtenido, con la mejor división temporal obtenida, calcular matrices de confusión finales y *accuracies* para el sistema usando el conjunto de prueba. Se debe calcular matrices de confusión y *accuracies* para: (i) sólo el detector de gestos, (ii) sólo el clasificador de gestos y (iii) ambos en conjunto.

Importante: No necesariamente los pasos (4)–(10) deben realizarse en el orden solicitado, pues pudiera ser que haya que iterar entre estos.

El proyecto debe realizarse en Python 3, con bibliotecas de machine learning estándares (scikit-learn, pandas, etc.), instaladas en forma local o usando *colaboratory*. En este último caso, se recomienda primero bajar el archivo .zip comprimido, y luego descomprimirlo usando la siguiente línea en el notebook:

```
!unzip EMG_data_for_gestures-master.zip
```

Entregas:

1. Presentación oral (50%), donde se expliquen las metodologías que utilizó para resolver el problema, las características, el método de selección de características, resultados de clasificación etc. (25 y 27 de Junio)
2. Código e informe escrito (50%), donde se expliquen las metodologías que utilizó para resolver el problema, las características, el método de selección de características, resultados de clasificación etc. (12 de Julio)

La extensión de las presentaciones será de aproximadamente 5 minutos. La información del tiempo exacto de éstas se entregará una semana antes de las presentaciones, considerando el número de estudiantes que presentará. 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) 2 horas antes de la presentación oral. 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. En las fechas en que hay presentaciones orales, suya o de sus compañeros, su **asistencia es obligatoria**. Las entregas atrasadas serán penalizadas con un punto de descuento por cada día de atraso. Se abrirá un tema en el foro para consultas.

La evaluación considerará el correcto funcionamiento del programa, 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.

[1] Macro-Class Selection for Hierarchical K-NN Classification of inertial sensor data.
http://crcv.ucf.edu/papers/PECCS_2012.pdf