Proyecto EL4106 - Semestre Otoño 2021

Profesor: Javier Ruiz del Solar
Auxiliar: Patricio Loncomilla.

Publicación enunciado 1° etapa:	15 de Junio de 2021
Primer avance:	9 de Julio de 2021
Presentación final:	27 de Julio de 2021

El objetivo de este proyecto es implementar un clasificador de actividades físicas. Se utilizará una versión modificada de la base de datos *Human Activity Recognition Using Smartphones Data Set*, cuya versión original está disponible en:

https://archive.ics.uci.edu/ml//datasets/Human+Activity+Recognition+Using+Smartphones

Esta base de datos contiene mediciones de personas realizando diferentes actividades físicas (6 diferentes actividades), medidas con una IMU (unidad de mediciones inerciales, sigla del inglés) de un Samsung Galaxy S II. La IMU incluye un acelerómetro 3D y un giroscopio de 3 ejes. A partir de esas 6 señales, se calculan 9 finales, aplicando un filtro sobre las señales del acelerómetro para dividirlas en dos componentes (9=3xgiróscopo+2x3xacelerómetro). Las actividades físicas (clases) para las cuales se realizaron las mediciones son:

- Caminar
- Subir escaleras caminando
- Bajar escaleras caminando
- Estar sentado
- Estar parado
- Estar acostado

Para cada actividad se tomaron muestras a 50 [Hz] de cada uno de los sensores, las cuales fueron reducidas a ventanas de 128 muestras (2.56 [s]). A partir del conjunto de datos original, se generó un nuevo conjunto para poder desarrollar el proyecto. Dicho conjunto tiene un total de 2800 muestras de entrenamiento, 2000 de validación y 2000 de prueba. Las etiquetas del conjunto de prueba no estarán disponibles para el desarrollo del proyecto y serán utilizadas para su evaluación.

Se les pide diseñar, implementar, entrenar y validar un sistema de clasificación que reciba estas 9 señales de entrada, una por cada sensor y que determine la acción realizada (clase). Existe la posibilidad de que usted determine que no necesariamente las 9 señales deben ser usadas, sino un número menor.

Para lograr esto necesitará:

(1) Realizar un programa que permita leer la base de datos. Esta consiste en un conjunto de archivos .csv (separados por comas). Se recomienda leerlo usando el paquete *pandas*. Se debe notar que las etiquetas del conjunto de prueba no están disponibles.

(2) Graficar algunas señales del conjunto de datos. Esto puede facilitar la comprensión del problema a resolver.

(3) 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.

(4) 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.

(5) 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. Se recomienda normalizar las características, usando las estadísticas del conjunto de entrenamiento (no el de validación)

(6) Aplique algún método de selección de características para determinar las características a ser usadas para entrenar el clasificador.
(7) Grafique clusters sobre un espacio de 2 dimensiones. Para esto, se debe aplicar PCA sobre las características y luego aplicar DBSCAN sobre los datos resultantes. El gráfico obtenido puede servir como guía para ver cuáles características permiten obtener una mejor separación de las clases.
(8) Elegir y entrenar un clasificador o cascada de clasificadores para determinar la actividad física realizada, 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.). Se debe usar un mínimo de 2 clasificadores para el proyecto y al menos uno de ellos no debe estar basado en deep learning.
(9) 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).

Importante: No necesariamente los pasos (4)-(8) 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 dataset_full_v03_alumnos.zip
```

Entregas:

1. Avance – 9 de Julio (1/3 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.
2. Final – 27 de Julio (2/3 de la nota del proyecto)
    - 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 y 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, 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. Habrá una competencia para la entrega inicial y otra para la entrega final. Dado que Kaggle no entrega matrices de confusión, almacene en su computador el archivo con las etiquetas predichas antes de subirlas. Se entregarán las etiquetas reales después de que el leaderboard privado de la entrega final se haga visible, momento en el cual se considera que el proyecto ha finalizado. De este modo, el alumno podrá generar matrices de confusión usando las etiquetas reales junto con las predichas que fueron subidas a la plataforma. Se debe notar que durante el desarrollo del proyecto sólo se interactuará con las etiquetas reales a través de los leaderboards.