Título

Detección de incoherencias en una evaluación por pares

Tareas

- Realizar una aplicación práctica basada en redes de **Deep Learning** para contrastar la **información numérica** y **la textual** para detectar incoherencias.
- Diseñar un **regresor** (modelo que predice valores numéricos) de forma que a partir de una observación textual (observación, sugerencia, comentario) se predice el **valor numérico correspondiente.** Es un problema similar al de **análisis de sentimiento** (sentiment análisis) dentro del área de lenguaje natural (NLP - Natural Language Processing). En este **contexto** tiene sentido aplicar **técnicas de NLP** debido a su aplicación en un **contexto restringido** (número de palabras reducido y relacionadas con cierta actividad).
- Se realizará la **memoria del desafío** (individual o en parejas) detallando los **pasos** realizados y los **resultados** obtenidos.
- Para la implementación de la práctica se usará el lenguaje Python usando básicamente la biblioteca de Deep Learning de tf.Keras (documentación).

Objetivos

- Conocer e implementar soluciones a problemas en el ámbito del procesamiento de lenguaje natural (NLP) usando redes neuronales;
- Conocer y solucionar algunos problemas comunes relacionados con el aprendizaje automático cuando se aplica a entradas de modelos en forma de texto. Algunos ejemplos serían el preprocesado (eliminación de caracteres especiales, extracción de lemas, etc.) y codificación de los mismos.

Contenidos

- El **conjunto de datos** que se usará consiste en una serie de comentarios (sugerencias) pertenecientes a dos actividades (actividad 1 con 4.735 sugerencias y actividad 2 con 6.783 sugerencias), etiquetadas con sus correspondientes secciones.

- El **texto original** está sin tratar, es decir, sin filtrar errores de sintaxis, con caracteres especiales, etc.
- En la **evaluación entre iguales** se realizan evaluaciones de actividades entre compañeros. Cada sección contiene:
    - un valor numérico de 0 a 3. Siendo 0 no realizado o mal resuelto y 3 completamente resuelto;
    - un campo de sugerencias al autor (texto en castellano) contiene los comentarios que el evaluador realiza al trabajo. Este campo puede estar en blanco en ausencia de comentarios cuando el evaluador encuentra toda la sección correcta.
- Se han usando dos tipos de actividades:
    - **1 - Creative Commons**: Se trata de elegir, referenciar e integrar en un tema 5 imágenes con ciertas restricciones de uso;
    - **2 - Webquest**: Diseño de un sistema basado de la metodología de la Webquest siguiendo un esquema determinado;
    - Todas las actividades anteriores se crearon con herramientas Web 2.0 y se entregaron mediante una URL pública y accesible.
- Tenemos que **probar** el ejemplo básico de la red como punto de partida para intentar mejorar su MAE (Mean Absolute Error) o RMSE (Root Mean Square Error) con 10-CV (cross-validation) ajustando algunos de sus parámetros como el número de capas, neuronas por capa, tipo de capa, o bien, cambiando la arquitectura general de la red (ver apartado de Criterios de evaluación).
- Se realizará una **corrección personal** (de forma presencial o remota síncrona) en la que se revisará la memoria entregada, el código implementado y los resultados obtenidos.

Esta práctica está basada en el siguiente artículo:

- Rico-Juan, J. R., Gallego, A. J., Valero-Mas, J. J., and Calvo-Zaragoza, J. (2018). **Statistical semi-supervised system for grading multiple peer-reviewed open-ended works**. *Computers & Education*, 126(1):264–282. Enlaces a la revista y al servidor GRFIA;

Entrega

Para entregar la práctica se rellenará el siguiente formulario con los siguientes campos:

- Memoria en formato PDF (con usando un capítulo para cada desafío con los apartados correspondientes realizados según el apartado de criterios de evaluación);
- `codigo.py`: Contendrá el código completo de la práctica con comentarios sobre cada apartado y las funciones o clases correspondientes.

La fecha límite:

- para la para la convocatoria C2 (enero) será el 6 de diciembre 2024 hasta las 23:59h y
- para la convocatoria C4 (julio) será el 19 de junio de 2025 hasta las 23:59h.

Criterios de evaluación

Se valorará la calidad de la memoria final evaluando elementos como que sea clara, resumida, con los gráficos necesarios y completa en cuanto a los experimentos realizados; además de los ajustes y mejoras realizadas sobre la red básica inicial.

Niveles de la evaluación:

Básico (hasta 5 puntos)

Realización de los siguientes apartados:

- Preprocesado básico de palabras (minúsculas, lemmas, caracteres especiales y stopwords);
- Aplicación de la validación cruzada 10-CV;
- Aplicación de un par de redes neuronales. Una de ellas basada en el concepto de conjunto de palabras (bag of words) bien usando 0/1 si la palabra está o no en el texto, usando el número repeticiones o frecuencia, o usando el índice TF-IDF. La otra aproximación consistiría en usar una secuencia de palabras donde el orden de las mismas se tiene presente (previamente convertidas a índices en un diccionario);
- Evaluación de la redes por actividad (1 o 2);
- Algún estudio comparativo usando el test estadísticos (Wilcoxon);
- Redacción correcta de la memoria a entregar.

Medio (hasta 3 puntos)

Realización de la mayoría de apartados:

- Aplicar más arquitecturas de redes distintas a la del apartado básico;
- Ajustes combinados (número de capas, tipo de capas, normalización, etc.);
- Preprocesado avanzado (eliminación de palabras poco útiles [stop words], detección de errores de sintaxis, aumentado de datos atendiendo a sinónimos o mezcla entre recomendaciones de similar valor);
- Ajuste del algoritmo de optimización (mejores resultados con menos épocas);
- Uso de la notación funcional de tf.Keras (functional API) o subclasses (subclassing API) en toda la práctica;
- Uso de redes ya entrenadas para aplicarlas directamente al texto, o bien, ajustarlas (fine tuning) al vocabulario.

- Detectar en qué palabras se centra el modelo para explicar su predicción para ello se puede usar el paquete SHAP (SHapley Additive exPlanations);
- Comparativas entre los diferentes ajustes con verificación de resultados (test estadísticos).

Avanzado (hasta 2 puntos)

Cambios significativos respecto de la red inicial que implicaría la realización de alguno de los siguientes apartados:

- Aumentado de datos teniendo que equilibrar el número de ejemplos entre diferente valores (generador personalizado para equilibrar los batches);
- Usar modelos de redes más modernas para aprendizaje semántico (para aplicar similitud y en este caso predicción de un valor) como los tipo Transformer, BERT u otros tipos más recientes.
- Uso de redes GAN (Generative Adversarial Networks) para generar nuevos ejemplos y permitir reducir el error final de la predicción;
- Aportaciones propias en alguna de las etapas de la construcción de la red o de la experimentación.