

Procesamiento del Lenguaje Natural, 2025-1

Facultad de Ingeniería, UNAM

Diseño y Regularización de Redes Neuronales para Clasificación
Multiclase de Texto (Programa 2)

1 Objetivos

El objetivo de este ejercicio es emplear las habilidades de programación de una red neuronal en Keras para llevar a cabo clasificación multiclase de tweets vinculados al ciberacoso. Para desarrollar el programa se debe preprocesar el texto, definir el pipeline y diseñar la arquitectura de la red neuronal, incluyendo funciones de pérdida, funciones de activación y optimizadores. Además, se requiere la implementación de técnicas de regularización, como dropout, para evitar el sobreajuste. Finalmente, es esencial guardar los pesos de la red neuronal para su posterior uso.

2 Desarrollo

Para el desarrollo del programa utilizar únicamente los archivos adjuntos proporcionados en las instrucciones del entregable del classroom, los cuales son: `cyberbullying_train.csv`, `cyberbullying_val.csv` y `cyberbullying_test.csv`.

El conjunto de datos (entrenamiento, validación y prueba) contiene más de 47000 tweets etiquetados según la clase de ciberacoso:

- Age
- Ethnicity
- Gender
- Religion
- Other type of cyberbullying
- Not cyberbullying

Los datos han sido balanceados para contener aproximadamente 8000 de cada clase.

Para el desarrollo del programa preprocesar el texto y utilizar vectores de tf-idf como entrada para la red neuronal. Diseñar la arquitectura neuronal y entrenar en el conjunto de datos `cyberbullying_train.csv`, utilizando

`cyberbullying_val.csv` como conjunto de validación. Utiliza el conjunto `cyberbullying_val.csv` para monitorear sobreajuste.

Para la red neuronal, definir una capa `Linear(keras.layers.Layer)`, agregar la matriz de pesos "W", el vector de sesgo "B", y las funciones de activación que se consideren necesarias.

Notas importantes que te pueden ayudar a realizar el programa (opcional):

- Puedes utilizar el método `map`¹ de `tf.data` para generar los vectores "one-hot" que corresponden a las etiquetas de clase.
- Utiliza el método `compile`² para especificar la función de pérdida y el optimizador.
- Utiliza el método `fit`³ para proporcionar los conjuntos de entrenamiento y validación, especificando el número de épocas y el `callback`⁴ para poder recuperar los mejores pesos para la evaluación en el conjunto de `cyberbullying_test.csv`.
- Utiliza la capa de `dropout`⁵ para evitar el sobreajuste.
- La evaluación en el conjunto de `cyberbullying_test.csv` se debe realizar utilizando el método `evaluate`⁶.

3 Entregables

1. Entregar el notebook con la ejecución del entrenamiento (mostrando gráficas de *accuracy* y *loss* para los conjuntos de entrenamiento y validación), conclusiones, el reporte de clasificación y la matriz de confusión. La matriz de confusión debe mostrar el nombre de las clases en el eje X y Y, como se muestra en la Figura 1.
2. Subir los pesos de la red neuronal en formato `.zip` para poder reproducir el entrenamiento y los resultados.

4 Evaluación

Para obtener una calificación aprobatoria, se requiere que el entrenamiento se pueda reproducir al ejecutar el notebook y que el modelo alcance al menos una precisión (*accuracy*) del 0.83 en el conjunto de datos `cyberbullying_test.csv`. El modelo debe entrenarse únicamente en el conjunto `cyberbullying_train.csv`, y no en el conjunto `cyberbullying_test.csv`.

¹https://www.tensorflow.org/api_docs/python/tf/data/Dataset#map

²https://keras.io/api/models/model_training_apis/

³https://keras.io/api/models/model_training_apis/

⁴<https://keras.io/api/callbacks/>

⁵https://keras.io/api/layers/regularization_layers/dropout/

⁶https://keras.io/api/models/model_training_apis/

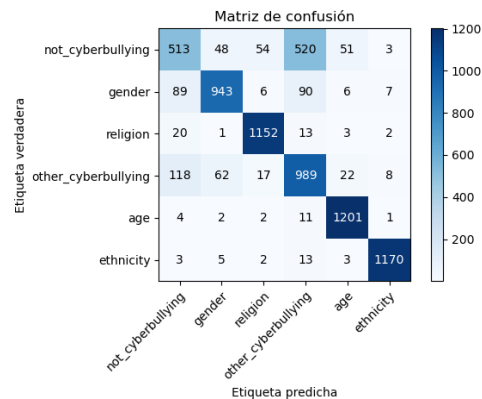


Figure 1: Matriz de confusión.

El ejercicio puede ser entregado de manera individual o en equipos de máximo dos personas y tiene un valor de **9 puntos**. El peso de la evaluación se distribuye de la siguiente manera:

- Preprocesamiento del texto, pipeline y arquitectura neuronal (definición de capa `Linear(keras.layers.Layer)`) adecuada para el problema (**1 punto**).
- Conclusiones, describiendo claramente los métodos empleados para resolver el problema y justificando claramente el preprocesamiento de los datos realizado, incluyendo la selección de parámetros y la arquitectura neuronal del modelo. Además, se debe presentar la matriz de confusión y el reporte de clasificación que muestre correctamente el nombre de cada clase (**1 punto**).
- Obtener una precisión (accuracy) de al menos 0.83 en el conjunto de datos `cyberbullying_test.csv` (**7 puntos**).

Notas importantes de la evaluación:

- En caso de detectar soluciones demasiado similares o de presentar el trabajo de otro como propio, los involucrados perderán el derecho a la calificación en el ejercicio.
- No se aceptan programas después de la fecha y hora de entrega (tomando en cuenta los días de retraso restantes).