

INSTITUTO TECNOLÓGICO DE COSTA RICA
Ingeniería en Computación, Alajuela (IC-Alajuela)
I Semestre 2023
Inteligencia Artificial
Tarea programada 03
Profesora: María Auxiliadora Mora

Tema: Aprendizaje profundo con redes neuronales convolucionales y recurrentes (LSTM)

Entrega: Un archivo .zip que contenga un documento en formato Jupyter notebook bien documentado que incluya los ejercicios. A través del TEC-digital.

Modo de trabajo: Grupos de 2 personas máximo.

Tecnología a utilizar: Python, PyTorch, entre otras bibliotecas.

Introducción:

En esta tarea se aplicarán conceptos básicos de aprendizaje automático utilizando redes neuronales convolucionales y recurrentes (LSTM) para resolver problemas que involucran clasificación tanto de imágenes como de textos de lenguaje natural, este último caso más conocido como análisis de sentimientos.

Las y los estudiantes deberán completar dos secciones de ejercicios. La primera sección está orientada a trabajar con clasificación de imágenes, en este caso se utilizarán los conjuntos de datos propuestos por la profesora para entrenar una red convolucional para la clasificación de imágenes. En el segundo caso se realizará procesamiento de lenguaje natural, se debe implementar una red neuronal Long Short-Term Memory (LSTM) aplicada a un problema de clasificación de textos de opinión sobre algún producto o servicio. Los conjuntos de datos serán propuestos también por la profesora.

El **objetivo del trabajo** es poner en práctica las **habilidades de investigación y el conocimiento adquirido durante el curso** sobre aprendizaje profundo por medio de ejercicios prácticos que permitan a las y los estudiantes experimentar con algoritmos de aprendizaje automático.

En ambos ejercicios se debe utilizar la biblioteca PyTorch, de no ser así, el ejercicio no aportará al cálculo de la nota de la tarea.

Objetivos de aprendizaje:

1. Poner en práctica habilidades de investigación y documentación de resultados.
2. Aplicar el conocimiento teórico sobre aprendizaje profundo en la implementación de un ejemplo aplicado a la clasificación de imágenes.
3. Experimentar con el flujo completo de trabajo requerido en proyectos de aprendizaje automático para realizar análisis de sentimientos a partir de datos en lenguaje natural.

4. Fortalecer capacidades en los estudiantes en el uso de bibliotecas de aprendizaje automático como PyTorch y otras.

Ejercicios

Sección 1. Redes neuronales convolucionales (CNN).

Utilice PyTorch para implementar una red neuronal profunda para clasificar imágenes utilizando capas convolucionales.

Realice las siguientes actividades vistas en clase:

1. Describa el problema y el objetivo del ejercicio.
2. Referencie formalmente los datos utilizados en el ejercicio.
3. Cargue, escale y explore los datos.
4. Calcule algunas estadísticas importantes, por ejemplo cantidad de registros por clase para verificar si las clases están balanceadas y un histograma del largo de los textos.
5. Defina la red convolucional.
6. Defina los hiper-parámetros de entrenamiento, por ejemplo, función de pérdida, el optimizador, entre otros.
7. Separe las muestras en datos de entrenamiento y validación y entrene el modelo.
8. Evalúe el modelo resultante con métricas como Accuracy, Precision, Recall y F1. Despliegue de forma gráfica la matriz de confusión para el cálculo de las métricas y explique los resultados obtenidos.
9. Grafique la curva de error, explique los resultados obtenidos y ajuste el modelo o el proceso de entrenamiento apropiadamente (por ejemplo el número de épocas).
10. Analice los resultados, proponga mejoras y explique los cambios realizados al flujo de trabajo del proyecto para mejorar el rendimiento de la red (aplique al menos dos cambios que efectivamente mejoren el rendimiento).
11. Presente al menos cuatro conclusiones.
12. Todas las secciones del ejercicio deben estar bien documentadas.
13. Incluya referencias en formato APA al final del documento.

Para el ejercicio deben utilizar al menos el 30% de las clases del siguientes conjunto de datos:

- BIRDS 525 SPECIES- IMAGE CLASSIFICATION:
<https://www.kaggle.com/datasets/gpiosenka/100-bird-species>

Sección 2. Análisis de emociones con redes neuronales recurrentes LSTM utilizando Pytorch.

El reconocimiento de emociones a partir de texto es un campo de investigación relacionado con el análisis de sentimientos. El análisis de sentimientos tiene como objetivo detectar sentimientos positivos, negativos o neutros en un texto, mientras que el análisis de emociones tiene como objetivo detectar y reconocer tipos de sentimientos presentes en el texto como entusiasmo, odio, felicidad, tristeza, entre otros.

En este ejercicio, para realizar el análisis de emociones se utilizará una colección de tweets etiquetados con las emociones expresadas en ellos. El conjunto de datos fue publicado por data.world. Se adjunta una copia del archivo que contiene tres columnas:

tweet_id: Un identificador.

sentiment: Emoción presente en el tuit, o sea la clase a la que pertenece el tweet.

content: El texto del tweet.

El objetivo del ejercicio de modelización es clasificar los textos de los tweets utilizando cuatro de las clases con más registros.

Realice las siguientes actividades:

1. Describa el problema y el objetivo del ejercicio.
2. Preprocese el conjunto de datos, es decir:
 1. (1 punto) Verifique si existen registros con valores faltantes y de ser así elimínelos.
 2. (1 punto) Utilice expresiones regulares para eliminar los caracteres especiales característicos de los tweets como @, #, entre otros.
 3. (1 punto) Elimine las "stop words".
 4. (1 punto) Convierta el texto del campo content a minúsculas.
3. (2 puntos) Explore y visualice algunas estadísticas con gráficos de barras o pastel. Por ejemplo, cuente cuántos registros hay en cada clase y haga un histograma con el largo de los tweets.
4. (1 punto) Seleccione un subconjunto de los registros que corresponda a las cuatro clases con más datos.
5. Utilizando **PyTorch** defina una red recurrente LSTM para procesar el conjunto de datos y clasificar los comentarios de usuario.
6. Defina los hiper-parámetros del proceso de entrenamiento, por ejemplo, función de pérdida, el optimizador, entre otros.
7. Separe las muestras en datos de entrenamiento y evaluación y entrene el modelo.
8. Grafique la curva de error, explique los resultados obtenidos y ajuste el modelo o el proceso de entrenamiento apropiadamente.
9. Evalúe el modelo resultante utilizando una matriz de confusión y métricas extraídas a partir de esta (ie. exactitud, precisión, exhaustividad y F1). Despliegue de forma gráfica la matriz de confusión para el cálculo de las métricas y explique los resultados obtenidos.
10. Analice los resultados, proponga mejoras y explique los cambios realizados al flujo de trabajo del proyecto para mejorar el rendimiento de la red (aplique al menos dos cambios que efectivamente mejoren el rendimiento).

11. Genere y documente sus conclusiones (incluya al menos cuatro conclusiones importantes).
12. Todas las secciones del ejercicio deben estar bien documentadas (con encabezado en las funciones que describen qué hace cada una y descripción de los parámetros, además, porciones internas del código deben estar documentadas también).
13. Incluya una sección de referencias en formato APA al final del documento.

Rúbrica

Rubro	Puntos
Se cargaron, escalaron y visualizaron los datos para ser introducidos a la red convolucional.	2
Calcule algunas estadísticas importantes, por ejemplo cantidad de registros por clase para verificar si las clases están balanceadas.	1
Se definió una red convolucional, utilizando PyTorch.	5
Se definieron los hiper-parámetros de entrenamiento, por ejemplo, función de pérdida, el optimizador.	2
Se separaron las muestras en datos de entrenamiento y evaluación y se entrenó el modelo apropiadamente como hemos visto en clase.	1
Se grafica la curva de error y se explican los resultados obtenidos y se ajusta el modelo o el proceso de entrenamiento apropiadamente (por ejemplo el número de épocas).	6
Se evaluó el modelo resultante con métricas como Accuracy, Precision, Recall y F1 y se documentó la evaluación apropiadamente. Se despliega la matriz de confusión de forma gráfica.	6
Se aplicaron mejoras al flujo de trabajo del proyecto que tuvieron impacto positivo en el rendimiento de la red.	5
Se generó y documentó todas las conclusiones (al menos 4 conclusiones)	2
2) Clasificación de textos con redes neuronales recurrentes LSTM.	
Se cargaron, visualizaron y prepararon los datos para ser introducidos a la red LSTM.	4
Explore y visualice algunas estadísticas con gráficos de barras o pastel: cuente cuántos registros hay en cada clase y haga un histograma con el largo de los tweets.	2
Se seleccionaron un subconjunto de los registros que corresponden a las cuatro clases con más datos.	1
Se definió una red recurrente convolucional utilizando PyTorch	5
Se definieron los hiper-parámetros de entrenamiento, por ejemplo, función de pérdida, el optimizador.	1
Se separaron las muestras en datos de entrenamiento y evaluación y se entrenó el modelo apropiadamente como hemos visto en clase.	1
Grafique la función de error con datos entrenamiento y prueba de todas las épocas (como vimos en clase).	6
Se evaluó el modelo resultante utilizando una matriz de confusión y métricas extraídas a partir de esta (ie. exactitud, precisión, exhaustividad y F1) y se desplegó de forma gráfica la matriz de confusión para el cálculo de las métricas.	6
Se aplicaron mejoras al flujo de trabajo del proyecto que tuvieron impacto positivo en el	5

rendimiento de la red.	
Se generó y documentó todas las conclusiones (al menos 4 conclusiones interesantes)	2
Documentación de ambos ejercicios	
Se describe el problema y el objetivo de cada uno de los proyectos.	2
Se describen los datos utilizados en cada proyecto.	2
Todas las secciones del código están debidamente documentadas (con encabezado en las funciones que describen qué hace cada una y descripción de los parámetros, además, porciones internas del código están documentadas también).	4
Se incluyen referencias en formato APA.	1