

## Tarea 5 EL4106 - Semestre Otoño 2019

Profesor: Javier Ruiz del Solar

Auxiliar: Patricio Loncomilla

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

Fecha enunciado: 17 de junio de 2019.

Plazo entrega tarea: 1 de julio de 2019.

El objetivo de esta tarea es clasificar imágenes usando redes convolucionales. Se usará una base de datos llamada fashion-mnist, la cual contiene imágenes de tamaño 28x28, cada una de las cuales está asociada a una clase, la cual corresponde a una prenda de vestir entre 10 posibles. Además, hay 60.000 imágenes para entrenamiento y 10.000 para prueba. Se usará la biblioteca Keras para poder entrenar clasificadores, **usando gpu**.

Se usará el siguiente código como base para el proyecto:

[https://github.com/keras-team/keras/blob/master/examples/mnist\\_cnn.py](https://github.com/keras-team/keras/blob/master/examples/mnist_cnn.py)

Dado que se requiere gpu para realizar esta tarea, en el caso que el alumno no tenga acceso local a una, debe usar *colaboratory*, el cual es un sistema de Google que permite ejecutar un notebook de forma remota. El acceso a la gpu se obtiene del siguiente modo:

Menú “entorno de ejecución” -> cambiar entorno de ejecución -> gpu

Para poder realizar la tarea, se recomienda seguir el siguiente procedimiento:

- 1) Marco teórico
  - a. Defina brevemente en qué consiste una capa convolucional
  - b. Defina brevemente en qué consiste una capa fully connected
  - c. Defina brevemente en qué consiste max-pooling
  - d. Defina brevemente en qué consiste dropout
  - e. Indique qué es el accuracy
  - f. Indique qué es una función de pérdida (loss)
  - g. Indique cómo las curvas de accuracy de entrenamiento y validación permiten evaluar el sobreajuste de la red
  - h. Indique cómo las curvas de loss de entrenamiento y validación permiten evaluar el sobreajuste de la red
- 2) Descargue el código desde la dirección web indicada. Modifíquelo para que lea el dataset fashion-mnist en vez de mnist y ejecute un primer entrenamiento para asegurarse de que el sistema funcione
- 3) Describa la arquitectura de la red usada como base
- 4) El código usado como base sólo considera conjuntos de entrenamiento y prueba. Modifíquelo de modo que, a partir del conjunto de entrenamiento original, se puedan obtener nuevos conjuntos de entrenamiento y validación. Se recomienda que la validación contenga un 25% de los datos originales de entrenamiento.
- 5) Implemente un código que permita imprimir los *accuracies* y *losses* en los conjuntos de entrenamiento, validación y prueba, una vez que el sistema ha sido entrenado.
- 6) Implemente un código que permita calcular e imprimir el tiempo de entrenamiento
- 7) Implemente un código que permita graficar el *accuracy* y *loss* obtenidos durante el entrenamiento. Puede hacer este gráfico después de que el entrenamiento ha sido realizado.
- 8) Realizar las siguientes pruebas y reportar *accuracies* y *losses* para cada una:
  - a. Con red original:
    - i. Entrenar con 4 épocas
    - ii. Entrenar con 12 épocas
    - iii. Entrenar con 30 épocas y graficar *accuracy* y *loss*
  - b. Con red sin la segunda capa convolucional:

- i. Entrenar con 4 épocas
    - ii. Entrenar con 12 épocas
    - iii. Entrenar con 30 épocas y graficar *accuracy* y *loss*
  - c. Con red sin dropout
    - i. Entrenar con 4 épocas
    - ii. Entrenar con 12 épocas
    - iii. Entrenar con 30 épocas y graficar *accuracy* y *loss*
- 9) Para cada arquitectura, indique el número de épocas que obtuvo mejor desempeño en el test *accuracy* y test *loss*
- 10) Indique la mejor arquitectura/número de épocas según dos métricas: test *accuracy* y test *loss*
- 11) Analice la velocidad de entrenamiento obtenida por las tres arquitecturas de red, considerando 30 episodios
- 12) A partir de los gráficos obtenidos para la red original y la red con una capa convolucional menos, indique cuál es el efecto que genera sacar una capa convolucional.
- 13) A partir de los gráficos obtenidos para la red original y la red sin dropout, indique cuál es el efecto que genera el uso de dropout.
- 14) Genere un código que permita graficar los pesos (máscaras) de la primera capa convolucional. Si usa algún código como base para este punto en particular, referéncielo.
- 15) Modifique el tamaño de las máscaras de la primera capa para que tengan tamaño 7x7. Grafique los pesos antes de entrenar y después de entrenar, usando 50 épocas.

Los informes y códigos deben ser subidos a u-cursos a más tardar el día 1 de julio a las 23:59. Incluir un corto archivo de texto explicando cómo se utiliza su programa. Las tareas atrasadas serán penalizadas con un punto base por cada día de atraso.

Los informes deben además incluir introducción y conclusiones, de un modo similar a cómo se ha realizado en las tareas anteriores.

**Importante:** La evaluación de la tarea considerará el correcto funcionamiento del programa, la inclusión de los resultados de los pasos pedidos en el informe, la calidad de los experimentos realizados y de su análisis, la inclusión de las partes importantes del código en el informe, así como la forma, prolijidad y calidad de este. Se subirá a u-cursos una pauta indicando la estructura del informe.