

07MIAR_Proyecto_Programacion

October 9, 2021

1 07MIAR - Redes Neuronales y Deep Learning: Proyecto de programación "*Deep Vision in classification tasks*"

1.1 Enunciado

En esta actividad, el alumno debe **evaluar y comparar dos estrategias** para la **clasificación de imágenes** empleando el **dataset asignado**. El/La alumnox deberá resolver el reto proponiendo una solución válida **basada en aprendizaje profundo**, más concretamente en redes neuronales convolucionales (CNNs). Será indispensable que la solución propuesta siga el **pipeline visto en clase** para resolver este tipo de tareas de inteligencia artificial:

1. **Carga** del conjunto de datos
2. **Inspección** del conjunto de datos
3. **Acondicionamiento** del conjunto de datos
4. Desarrollo de la **arquitectura** de red neuronal y **entrenamiento** de la solución
5. **Monitorización** del proceso de **entrenamiento** para la toma de decisiones
6. **Evaluación** del modelo predictivo y planteamiento de la siguiente prueba experimental

1.1.1 Estrategia 1: Entrenar desde cero o *from scratch*

La primera estrategia a comparar será una **red neuronal profunda** que el **alumno debe diseñar, entrenar y optimizar**. Se debe **justificar empíricamente** las decisiones que llevaron a la selección de la **arquitectura e hiperparámetros final**. Se espera que el alumno utilice todas las **técnicas de regularización** mostradas en clase de forma justificada para la mejora del rendimiento de la red neuronal (*weight regularization, dropout, batch normalization, data augmentation, etc.*).

1.1.2 Estrategia 2: Red pre-entrenada

La segunda estrategia a comparar debe incluir la utilización de una **red preentrenada** con el dataset ImageNet, llevando a cabo tareas de **transfer learning** y **fine-tuning** para resolver la tarea de clasificación asignada. Deben **compararse al menos dos tipos de arquitecturas** (VGGs, ResNet50, Xception, InceptionV3, InceptionResNetV2, MobileNetV2, DenseNet, ResNet) y se debe **seleccionar la que mayor precisión proporcione** (información sobre las arquitecturas disponibles en <https://keras.io/applications/>). Se espera que el/la alumnox utilice todas las **técnicas de regularización** mostradas en clase de forma justificada para la mejora del rendimiento de la red neuronal (*weight regularization, dropout, batch normalization, data augmentation, etc.*).

1.2 Normas a seguir

- Será **indispensable** realizar el **trabajo por parejas**. Dichas parejas de alumnos se generarán **de manera automática** teniendo en cuenta el país de residencia con el objetivo de facilitar el trabajo en equipo.
- Se debe entregar un **ÚNICO FICHERO PDF POR ALUMNO** que incluya las instrucciones presentes en el Colab Notebook y su **EJECUCIÓN!!!**. Debe aparecer todo el proceso llevado a cabo en cada estrategia (i.e. carga de datos, inspección de datos, acondicionamiento, proceso de entrenamiento y proceso de validación del modelo).
- **La memoria del trabajo** (el fichero PDF mencionado en el punto anterior) deberá **subirla cada integrante del grupo** (aunque se trate de un documento idéntico) a la actividad que se habilitará en **CampusVIU**.
- Se recomienda trabajar respecto a un directorio base (**BASE_FOLDER**) para facilitar el trabajo en equipo. En este notebook se incluye un ejemplo de cómo almacenar/cargar datos utilizando un directorio base.
- Las **redes propuestas** deben estar **entrenadas** (y **EVIDENCIAR este proceso en el documento PDF**). La entrega de una **red sin entrenar** supondrá **perdida de puntos**.
- Si se desea **evidenciar alguna métrica** del proceso de entrenamiento (precisión, pérdida, etc.), estas deben ser generadas.
- Todos los **gráficos** que se deseen mostrar deberán **generarse en el Colab Notebook** para que tras la conversión aparezcan en el documento PDF.

1.3 Tips para realizar la actividad con éxito

- Los **datos** se cargarán directamente **desde** la plataforma **Kaggle** mediante su API (<https://github.com/Kaggle/kaggle-api>). En este Notebook se incluye un ejemplo de cómo hacerlo. Se recomienda generar una función que aborde esta tarea.
- El **documento PDF a entregar** como solución de la actividad se debe **generar automáticamente desde el fichero ".ipynb"**. En este Notebook se incluye un ejemplo de cómo hacerlo.
- **Generar secciones y subsecciones en el Colab Notebook** supondrá que el documento **PDF generado** queda totalmente **ordenado** facilitando la evaluación al docente.
- Se recomienda encarecidamente **incluir comentarios concisos pero aclaratorios**.
- Es muy recomendable crear una **última sección** de texto en el Colab Notebook en la que se discutan los diferentes modelos obtenidos y se extraigan las conclusiones pertinentes.

1.4 Criterios de evaluación

- **Seguimiento** de las **normas establecidas** en la actividad (detalladas anteriormente).
- Creación de una **solución que resuelva la tarea de clasificación**, i.e. que las figuras de mérito empleadas para medir la bondad del modelo predictivo evidencien un *performance* superior al rendimiento estocástico.
- **Claridad** en la creación de la solución, en las justificaciones sobre la toma de decisiones llevada a cabo así como en las comparativas y conclusiones finales.
- **Efectividad** al presentar las comparaciones entre métricas de evaluación de ambas estrategias.
- **Demostración** de la utilización de **técnicas de regularización** para mejorar el rendimiento de los modelos.

1.5 Ejemplo de carga de datos desde la plataforma Kaggle

```
[ ]: # En primer lugar se debe generar un API de Kaggle personal, para ello:  
# 1. Registrarse en https://www.kaggle.com  
# 2. Ir a 'Account tab' de tu perfil y seleccionar 'Create API Token'  
# 3. Almacenar en tu ordenador local el fichero kaggle.json  
  
[1]: # Nos aseguramos que tenemos instalada la última versión de la API de Kaggle en Colab  
!pip install --upgrade --force-reinstall --no-deps kaggle
```

```
Collecting kaggle  
  Downloading kaggle-1.5.12.tar.gz (58 kB)  
    || 58 kB 4.2 MB/s  
Building wheels for collected packages: kaggle  
  Building wheel for kaggle (setup.py) ... done  
  Created wheel for kaggle: filename=kaggle-1.5.12-py3-none-any.whl size=73051  
sha256=141b9f2c0d152797fa8b04866dcd7e2566a5e3efef0ffaf2c9fcded76124c14a  
  Stored in directory: /root/.cache/pip/wheels/62/d6/58/5853130f941e75b2177d281e  
b7e44b4a98ed46dd155f556dc5  
Successfully built kaggle  
Installing collected packages: kaggle  
  Attempting uninstall: kaggle  
    Found existing installation: kaggle 1.5.12  
    Uninstalling kaggle-1.5.12:  
      Successfully uninstalled kaggle-1.5.12  
Successfully installed kaggle-1.5.12
```

```
[2]: %%capture  
# Seleccionar el API Token personal previamente descargado (fichero kaggle.json)  
from google.colab import files  
files.upload()
```

<IPython.core.display.HTML object>

Saving kaggle.json to kaggle.json

```
[2]: {'kaggle.json':  
b'{"username": "adcogra89", "key": "93999d4b45d3ec7128e876bf17c7f5f4"}'}
```

```
[3]: # Creamos un directorio en el que copiamos el fichero kaggle.json  
!mkdir ~/.kaggle  
!cp kaggle.json ~/.kaggle/  
!chmod 600 ~/.kaggle/kaggle.json
```

```
[4]: # Ya podemos listar los datasets disponibles en kaggle para su descarga  
!kaggle datasets list
```

ref	size	lastUpdated	downloadCount	voteCount	usabilityRating	title
gpreda/reddit-vaccine-myths	235KB	2021-10-01 17:39:01	13441	1094	1.0	Reddit Vaccine Myths
crowww/a-large-scale-fish-dataset	3GB	2021-04-28 17:03:01				A Large Scale Fish Dataset
616	0.9375					8074
imsparsh/musicnet-dataset	22GB	2021-02-18 14:12:19	3556	253	1.0	MusicNet Dataset
fatiimaezzahra/famous-iconic-women	838MB	2021-02-28 14:56:00	1534	117	0.75	Famous Iconic Women
dhruvildave/wikibooks-dataset	2GB	2021-07-03 18:37:20	3104	235	1.0	Wikibooks Dataset
nickuzmenkov/nih-chest-xrays-tfrecords	TFRecords	11GB	2021-03-09 04:49:23			NIH Chest X-rays
72	0.9411765					1287
promptcloud/careerbuilder-job-listing-2020	Listing 2020	42MB	2021-03-05 06:59:52			Careerbuilder Job Listing 2020
77	1.0					2113
mathurinache/twitter-edge-nodes	342MB	2021-03-08 06:43:04	1033	90	1.0	Twitter Edge Nodes
coloradokb/dandelionimages	4GB	2021-02-19 20:03:47	915	43	0.75	DandelionImages
alsgroup/end-als	Challenge	12GB	2021-04-08 12:16:37			End ALS Kaggle Challenge
137	0.9375					952
simiotic/github-code-snippets	7GB	2021-03-03 11:34:39	373	70	1.0	GitHub Code Snippets
stuartjames/lights	Specularity Dataset	18GB	2021-02-18 14:32:26			LightS: Light Specularity Dataset
32	0.6875					169
mathurinache/the-lj-speech-dataset	Dataset	3GB	2021-02-15 09:19:54			The LJ Speech Dataset
50	1.0					377
nickuzmenkov/ranzcr-clip-kfold-tfrecords	TFRecords	2GB	2021-02-21 13:29:51			RANZCR CLiP KFold
24	0.875					138
imsparsh/accentdb-core-extended	Extended	6GB	2021-02-17 14:22:54			AccentDB - Core & Extended
31	0.875					150
landrykezebou/lvzhdr-tone-mapping-benchmark-dataset-tmonet	Benchmark Dataset (TMO-Net)	24GB	2021-03-01 05:03:40			LVZ-HDR Tone Mapping Benchmark Dataset (TMO-Net)
29	0.75					212
zynicide/wine-reviews	51MB	2017-11-27 17:08:04	145394	3163	0.7941176	Wine Reviews

datasnaek/youtube-new				Trending YouTube
Video Statistics	201MB	2019-06-03 00:56:47		153628
4190 0.7941176				
datasnaek/chess				Chess Game Dataset
(Lichess)	3MB	2017-09-04 03:09:09		22700
843 0.8235294				
residentmario/ramen-ratings				Ramen Ratings
40KB 2018-01-11 16:04:39	27891	675	0.7058824	

```
[5]: # IMPORTANTE ANTES DE DESCARGAR UN DATASET !!!
# Antes de descargar el dataset debemos ir a https://www.kaggle.com/c/
    → $nameCompetition/data y aceptar las "Competition Rules"
# En caso de no realizar el paso anterior al descargar el dataset obtendremos
    → el siguiente mensaje "403 - Forbidden"
# AHORA SI: Descarguemos un dataset de cierta competición
!kaggle competitions download -c dogs-vs-cats-redux-kernels-edition

#NOTA: El nombre del dataset de competición a descargar se informará en el
    → documento en el que se presenten las parejas formadas
```

```
Downloading dogs-vs-cats-redux-kernels-edition.zip to /content
100% 810M/814M [00:21<00:00, 48.9MB/s]
100% 814M/814M [00:21<00:00, 39.3MB/s]
```

```
[6]: # Creemos un directorio para descomprimir los datos
!mkdir my_dataset
```

```
[7]: # Descomprimos los datos y los dejamos listos para trabajar
!unzip dogs-vs-cats-redux-kernels-edition.zip -d my_dataset
```

```
Archive: dogs-vs-cats-redux-kernels-edition.zip
  inflating: my_dataset/sample_submission.csv
  inflating: my_dataset/test.zip
  inflating: my_dataset/train.zip
```

```
[ ]: !unzip my_dataset/test.zip
```

```
[57]: %%capture
!ls test/
```

1.6 Ejemplo de escritura de datos tomando como referencia un BASE_FOLDER

```
[34]: # Conectamos con nuestro Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

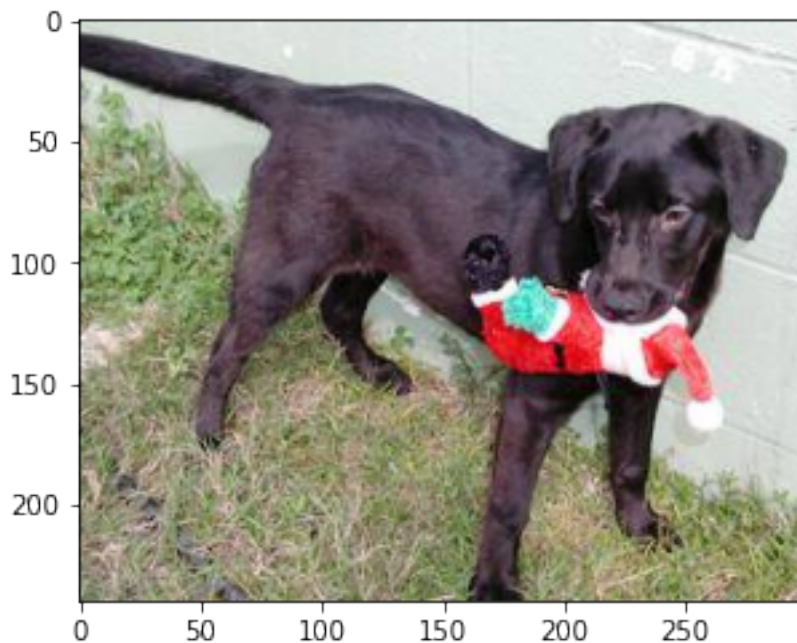
```
Mounted at /content/drive
```

```
[35]: # Establezco una ruta absoluta a un directorio existente de mi Google Drive
BASE_FOLDER = "/content/drive/MyDrive/07MIAR_Proyecto_Programacion/"
```

```
[38]: import matplotlib.pyplot as plt
import numpy as np
import cv2

# Escogiendo y mostrando una imagen al azar del conjunto de test
idx = np.random.randint(1, 12000)
img = cv2.imread('test/' + str(idx) + '.jpg', cv2.COLOR_BGR2RGB)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img)
```

```
[38]: <matplotlib.image.AxesImage at 0x7f8d1a8dcd10>
```



```
[39]: # Almacenando una imagen aleatoria de test en mi BASE_FOLDER
cv2.imwrite(BASE_FOLDER + 'my_pet.png', img)
```

```
[39]: True
```

1.7 Ejemplo de generación de documento PDF a partir del Colab Notebook (archivo ".ipynb")

```
[ ]: # Ejecutando los siguientes comandos en la última celda del Colab Notebook se
    → convierte de ".ipynb" a PDF
# En caso de querer ocultar la salida de una celda puesto que no tenga
    → relevancia se debe insertar
```

```
# el comando %%capture al inicio de la misma. Véase la celda que contiene !ls
→test en este Notebook.
```

```
[60]: name_IPYNB_file = '07MIAR_Proyecto_Programacion.ipynb'
get_ipython().system(
    "apt update >> /dev/null && apt install texlive-xetex
    →texlive-fonts-recommended texlive-generic-recommended >> /dev/null"
)
get_ipython().system(
    "jupyter nbconvert --output-dir='$BASE_FOLDER'
    →'$BASE_FOLDER'$name_IPYNB_file' --to pdf"
)
```

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

```
[NbConvertApp] Converting notebook /content/drive/MyDrive/07MIAR_Proyecto_Programacion/07MIAR_Proyecto_Programacion.ipynb to pdf
[NbConvertApp] Support files will be in 07MIAR_Proyecto_Programacion_files/
[NbConvertApp] Making directory ./07MIAR_Proyecto_Programacion_files
[NbConvertApp] Writing 43374 bytes to ./notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: [u'xelatex', u'./notebook.tex',
'-quiet']
[NbConvertApp] Running bibtex 1 time: [u'bibtex', u'./notebook']
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 179341 bytes to /content/drive/MyDrive/07MIAR_Proyecto_Programacion/07MIAR_Proyecto_Programacion.pdf
```