



UNIVERSIDAD DEL SINÚ
Elías Bechara Zainúm
Seccional Cartagena

**MODELO DE RED NEURONAL CONVOLUCIONAL PARA EL DIAGNÓSTICO
DE NEUMONÍA EN IMÁGENES RADIOLÓGICAS**

Manual Técnico

Estudiantes

AGUSTIN NAAR PEREZ
FABIAN BARRETO MARTINEZ

UNIVERSIDAD DEL SINÚ ELÍAS BECHARÁ ZAINÚM SECCIONAL CARTAGENA

ESCUELA DE INGENIERÍA DE SISTEMAS

CARTAGENA-COLOMBIA

JUNIO 2019

Tabla de contenido

INTRODUCCIÓN.....	4
1. Ambiente de desarrollo (Anaconda)	4
1.1 Júpiter notebook	5
2. Librerías	6
3. Cargar notebook de trabajo y ubicación de los datos.....	7
4. Notebook de modelado de la red neuronal convolucional	10

TABLA DE ILUSTRACIONES

Ilustración 1 página oficial para descargar anaconda	5
Ilustración 2 ambiente de desarrollo anaconda	5
Ilustración 3 panel de navegación de júpiter notebook	6
Ilustración 4. Forma de utilizar el comando para instalación de librerías	6
Ilustración 5 ubicación de la librería "cnn_utils.py"	7
Ilustración 6. Los notebook con la fuente de código.	7
Ilustración 7. Lectura de los archivos dicom	8
Ilustración 8. Ubicación de los archivos dicom	9
Ilustración 9. Extracción de variables dependientes.....	9
Ilustración 10. Ubicación para guardar el modelo	10

INTRODUCCIÓN

En el presente manual se detalla paso a paso como está organizado el ambiente de trabajo del algoritmo para comprender cada componente y herramientas usadas.

El ambiente de desarrollo usado fue anaconda para Windows el cual incorpora jupyter notebook con el lenguaje de programación python 3.7 y lo recomendable en capacidad de hardware para el uso de los datos es una DSVM(Data Scencie Virtual Machine) de la plataforma de servicios Azure Microsoft

1. Ambiente de desarrollo (Anaconda)

La distribución de fuente abierta Anaconda es la forma más fácil de realizar la ciencia de datos y el aprendizaje automático de Python / R en Linux, Windows y Mac OS X. Con más de 11 millones de usuarios en todo el mundo, es el estándar de la industria para el desarrollo, prueba y capacitación en una sola máquina, que permite a los científicos de datos individuales:

Descarga rápidamente más de 1,500 paquetes de Python / R data science

Administre bibliotecas, dependencias y entornos con Conda.

Desarrolle y capacite modelos de aprendizaje automático y aprendizaje profundo con scikit-learn , TensorFlow y Theano

Analice los datos con escalabilidad y rendimiento con Dask , NumPy , pandas y Numba

Visualice los resultados con Matplotlib , Bokeh , Datashader y Holoviews.

En la ilustración 1 se observa el portal web original para descargar anaconda.

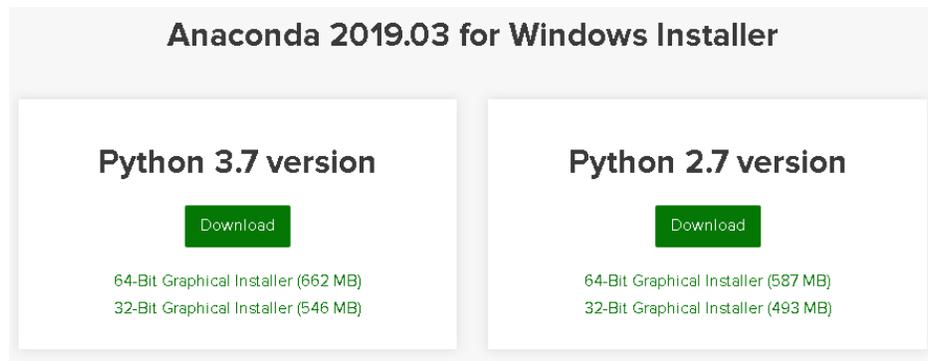


Ilustración 1 página oficial para descargar anaconda

Link de descargar para el entorno de desarrollo <https://www.anaconda.com/distribution/#download-section>

Luego de haber instalado anaconda se podrá acceder jupyter notebook para comenzar a instalar las librerías de desarrollo, en la ilustración 2 se observa al ambiente de desarrollo de anaconda.

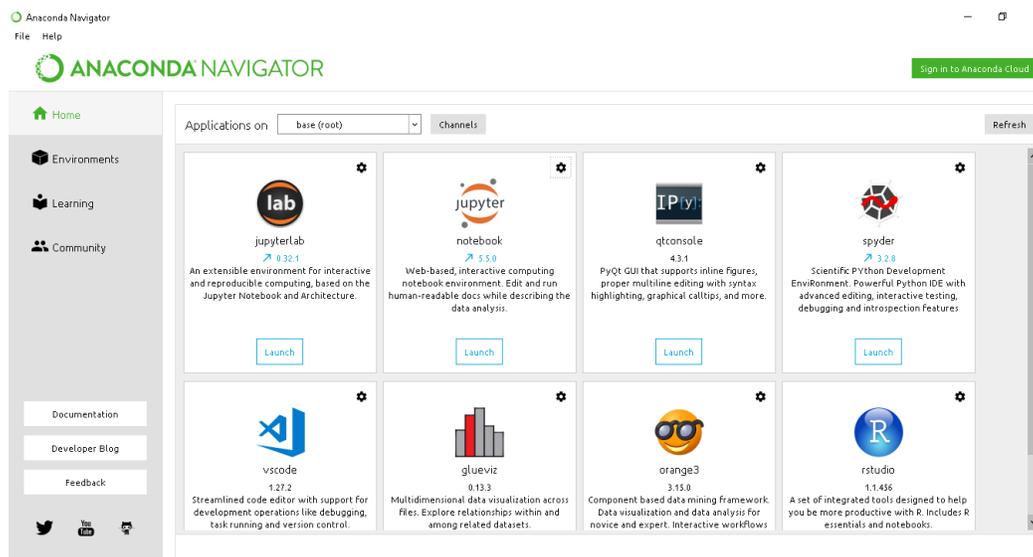


Ilustración 2 ambiente de desarrollo anaconda

1.1 Jupyter notebook

El cuaderno Jupyter es una aplicación web de código abierto que le permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Los usos incluyen: limpieza y transformación de datos, simulación

numérica, modelado estadístico, visualización de datos, aprendizaje automático y mucho más.

Luego de podrá observar el panel de navegación como se muestra en la ilustración 3, para cargar y crear cuadernos jupyter (notebook jupyter) en el botón desplegable con el nombre “new” se debe seleccionar para crear un nuevo cuaderno y para cargar el cuaderno con la fuente del código se buscar en la ubicación donde se encuentre.

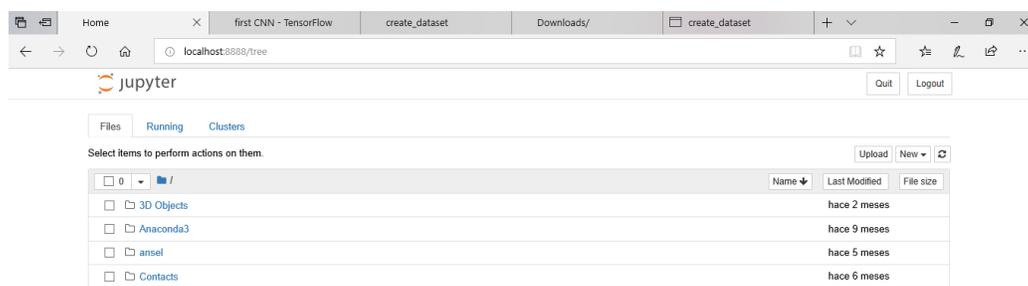


Ilustración 3 panel de navegación de jupyter notebook

2. Librerías

Tenido en cuenta que el entorno de desarrollo anaconda luego de haberse instalado integra python, pues permitirá instalar las librerías con el comando “!pip install” seguido del nombre de la librería como se muestra en la ilustracion4.

```
In [1]: !pip install dicom

Collecting dicom
  Downloading https://files.pythonhosted.org/packages/6c/0e/488becc361f1416f6682a9b9c223eb6a9db2d5d09-0.9.9.post1-py3-none-any.whl (766kB)
Installing collected packages: dicom
Successfully installed dicom-0.9.9.post1
```

Ilustración 4. Forma de utilizar el comando para instalación de librerías

Las librerías de uso libre que se deben instalar son:

Dicom, H5py, Numpy, Pandas, matplotlib.pyplot, sklearn.metrics, sklearn.model_selection, skimage.transform y tensorflow en la version 1.8

Otra librería que se necesaria es “cnn_utils.py” la cual no es publica pero está junto de los entregables y para poder importarla en el ambiente de desarrollo debe ser copiado el archivo “cnn_utils.py” en la ruta que se muestra en la ilustración 5.

Nombre	Fecha de modifica...	Tipo	Tamaño
_compat_pickle	28/03/2018 5:19 a. ...	Python File	9 KB
_compression	28/03/2018 5:19 a. ...	Python File	6 KB
_dummy_thread	28/03/2018 5:19 a. ...	Python File	5 KB
_markupbase	28/03/2018 5:19 a. ...	Python File	15 KB
_nsis	20/05/2018 4:35 a. ...	Python File	7 KB
_osx_support	28/03/2018 5:19 a. ...	Python File	19 KB
_pydecimal	28/03/2018 5:19 a. ...	Python File	225 KB
_pyio	28/03/2018 5:19 a. ...	Python File	86 KB
_sitebuiltins	28/03/2018 5:19 a. ...	Python File	4 KB

Ilustración 5 ubicación de la librería "cnn_utils.py"

3. Cargar notebook de trabajo y ubicación de los datos

Hasta este momento se ha creado el entorno de trabajo con las librerías implementadas en el algoritmo. Luego se puede cargar los cuadernos jupyter "create_dataset.ipynb" para lectura de imágenes dicom y "first CNN – TensorFlow.ipynb" para observar el modelamiento de la red neuronal convolucional con sus métricas de evaluación.

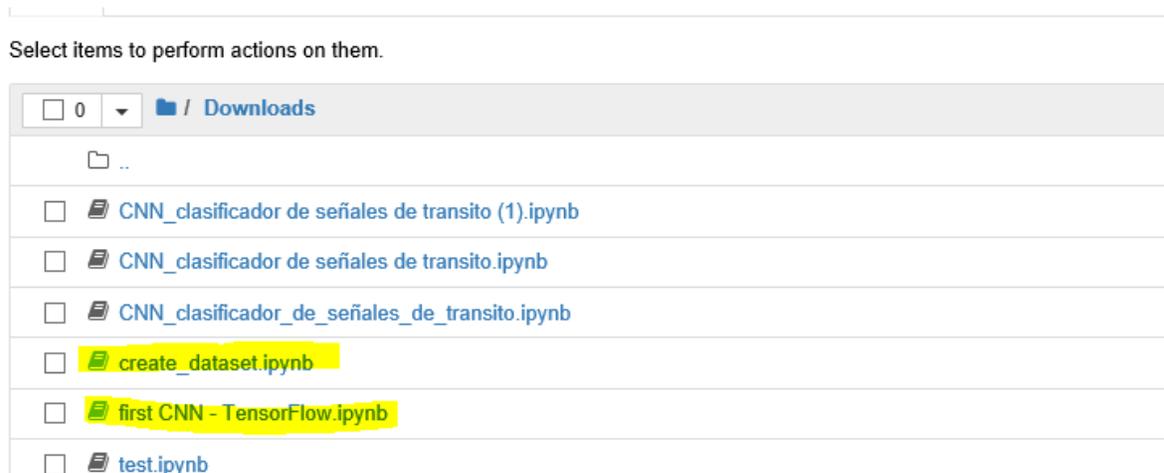


Ilustración 6. Los notebook con la fuente de código.

En el primer bloque de código del notebook “create_dataset.ipynb” que se puede observar en la ilustración 7 se debe establecer la ruta de ubicación de los archivos dicom, es recomendable tener todos los archivos dicom en una carpeta ubicada en “C:/”, y por asignación la ruta de destino para el conjunto de datos es la misma que la ubicación origen del notebook “create_dataset.ipynb”.



The image shows a Jupyter Notebook interface with the title "create_dataset" and a status bar indicating "Last Checkpoint: el martes pasado a las 19:03 (autosaved)". The notebook has a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Below the menu bar is a toolbar with icons for saving, adding cells, undo, redo, and running code. The main area contains a code cell with the following Python code:

```
X=np.array([])

for m in range(1,10001):

    imagen=(dicom.dcmread("C:\stage_2_train_images/"+str(m)+".dcm"))
    imagen_dicom=np.array(imagen)# FICHERO DICOM*****
    pixel_arr=np.array(imagen.pixel_array)
```

Ilustración 7. Lectura de los archivos dicom

La ilustración 8 muestra la ubicación de las imágenes dicom.

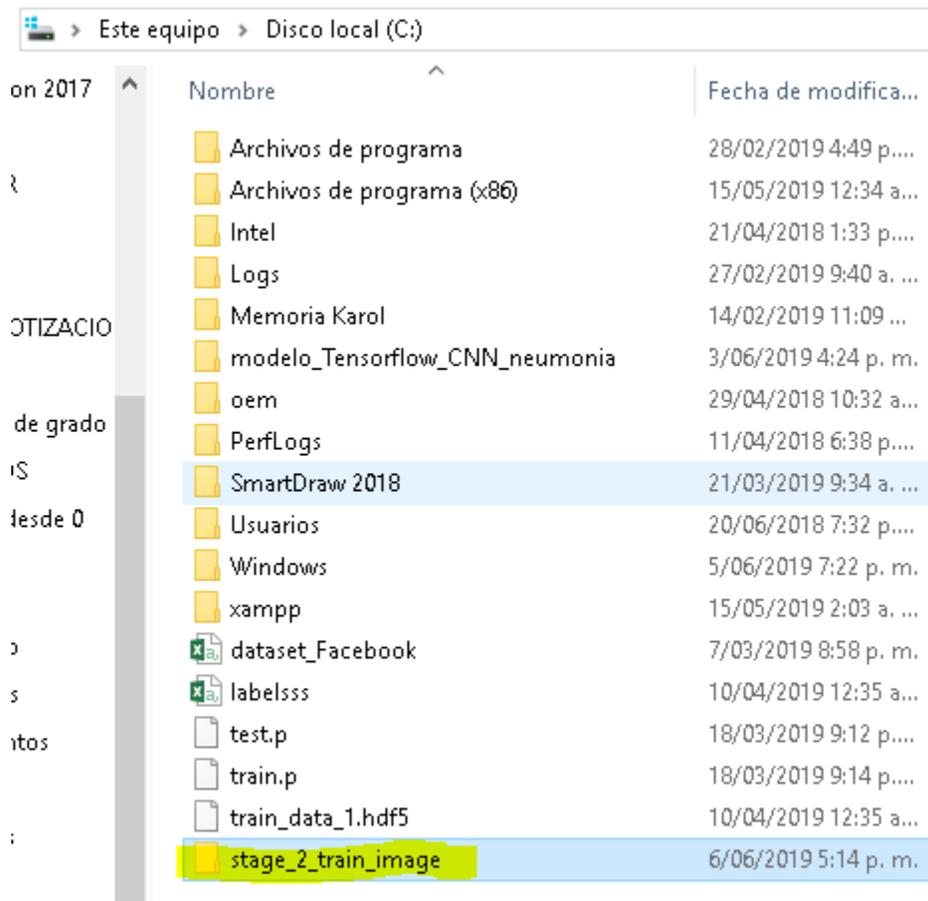


Ilustración 8. Ubicación de los archivos dicom

En el notebook “create_dataset.ipynb” también contiene el bloque de código encargado de la extracción de las variables dependientes como se muestra en la ilustración 9.

```
import numpy as np
import pandas as pd
import pydicom as dicom
import h5py
nodupli=np.array([])

labels_ordenado=np.array([])# donde guardar Los Leables que coninsiden con Los feat

labels = pd.read_csv('C:/labels_unicos.csv')
Y=np.array([])
print("-----labels cruzados-----")
for d in range(1,20780):

    imagen=(dicom.dcmread("C:/stage_2_train_images/"+str(d)+".dcm"))
    pac=np.array(int(labels.Target[imagen.PatientID==labels.patientId]))
```

Ilustración 9. Extracción de variables dependientes.

4. Notebook de modelado de la red neuronal convolucional

Hasta este punto fueron cargados todos los datos (variables dependientes e independientes), para poder construir los subconjunto de datos y a su vez utilizarlos en el notebook “First CNN – TensorFlow.ipynb” que permite construir el modelo de red convolucional.

En la ilustración 10 se puede observar la ruta de destino para el modelo entrenado

```
saver.save(sess, './my_test_model')
# plot the cost
plt.plot(np.squeeze(costs))
plt.ylabel('cost')
plt.xlabel('iterations (per tens)')
plt.title("Learning rate =" + str(learning_rate))
plt.show()

# Calculate the correct predictions
correct_prediction = tf.equal(tf.argmax(y_pred,1), tf.argmax(y_true,1))
```

Ilustración 10. Ubicación para guardar el modelo