

Model

For this homework we will use Convolutional Neural Network (CNN). Like in the lectures, we'll use Keras.

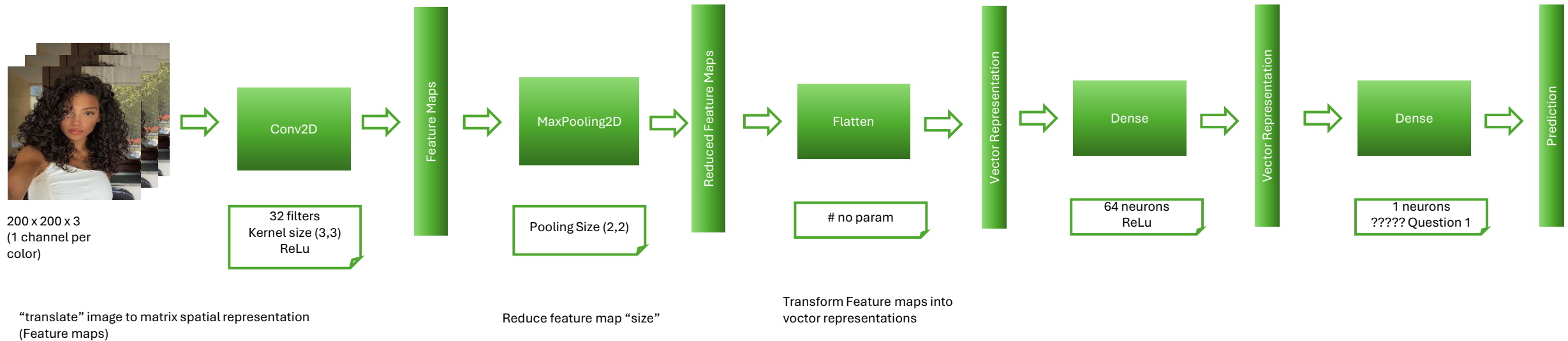
You need to develop the model with following structure:

- The shape for input should be `(200, 200, 3)`
- Next, create a convolutional layer (`Conv2D`):
 - Use 32 filters
 - Kernel size should be `(3, 3)` (that's the size of the filter)
 - Use `'relu'` as activation
- Reduce the size of the feature map with max pooling (`MaxPooling2D`)
 - Set the pooling size to `(2, 2)`
- Turn the multi-dimensional result into vectors using a `Flatten` layer
- Next, add a `Dense` layer with 64 neurons and `'relu'` activation
- Finally, create the `Dense` layer with 1 neuron - this will be the output
 - The output layer should have an activation - use the appropriate activation for the binary classification case

As optimizer use `SGD` with the following parameters:

Notas:

- Los filtros (también llamados pesos o kernels) son aprendidos y se quedan guardados en el modelo.
- La reducción de los feature maps con Maxpooling ayuda a disminuir la complejidad del procesamiento
- El diagrama muestra una operación, pero estas librerías trabajan en batches, es decir de manera paralela



```
... Found 201 images belonging to 2 classes.
```

```
model.fit(train_ds, epochs=10, validation_data=val_ds, batch_size=20)
```

[21] ✓ 1m 58.2s

Python

```
... Epoch 1/10
40/40 [=====] - 13s 315ms/step - loss: 0.6929 - accuracy: 0.4963 - val_loss: 0.6934 - val_accuracy: 0.5075
Epoch 2/10
40/40 [=====] - 12s 293ms/step - loss: 0.6915 - accuracy: 0.5150 - val_loss: 0.6943 - val_accuracy: 0.5174
Epoch 3/10
40/40 [=====] - 11s 279ms/step - loss: 0.6909 - accuracy: 0.5200 - val_loss: 0.6948 - val_accuracy: 0.5075
Epoch 4/10
40/40 [=====] - 12s 289ms/step - loss: 0.6901 - accuracy: 0.5188 - val_loss: 0.6946 - val_accuracy: 0.5124
Epoch 5/10
40/40 [=====] - 12s 301ms/step - loss: 0.6885 - accuracy: 0.5312 - val_loss: 0.6974 - val_accuracy: 0.5075
Epoch 6/10
40/40 [=====] - 11s 281ms/step - loss: 0.6867 - accuracy: 0.5325 - val_loss: 0.6954 - val_accuracy: 0.5174
Epoch 7/10
40/40 [=====] - 12s 310ms/step - loss: 0.6858 - accuracy: 0.5325 - val_loss: 0.7003 - val_accuracy: 0.5025
Epoch 8/10
40/40 [=====] - 11s 284ms/step - loss: 0.6840 - accuracy: 0.5387 - val_loss: 0.7030 - val_accuracy: 0.5124
Epoch 9/10
40/40 [=====] - 11s 282ms/step - loss: 0.6793 - accuracy: 0.5500 - val_loss: 0.7050 - val_accuracy: 0.5124
Epoch 10/10
40/40 [=====] - 11s 276ms/step - loss: 0.6596 - accuracy: 0.6125 - val_loss: 0.6732 - val_accuracy: 0.6517
```

1st run

```
... <tensorflow.python.keras.callbacks.History at 0x1c22fbaf2e0>
```

▷ ◁

Python

File

Edit

Selection

View

Go

Run

Terminal

Help

←

→

ml_zc_w8_lap (Workspace)

hw_notebook.ipynb

ml_zc_w8_lap > hw_notebook.ipynb > empty cell

Code

Markdown

Run All

Restart

Clear All Outputs

Variables

Outline

tf_gpu_py39 (Python 3.9.15)

Found 201 images belonging to 2 classes.

history_normalized = model.fit(train_ds, epochs=10, validation_data=val_ds, batch_size=20)

[22] ✓ 1m 56.1s

Python

Epoch 1/10

40/40 [=====] - 13s 323ms/step - loss: 0.6237 - accuracy: 0.6913 - val_loss: 0.6750 - val_accuracy: 0.6965

Epoch 2/10

40/40 [=====] - 11s 276ms/step - loss: 0.5884 - accuracy: 0.6900 - val_loss: 0.6611 - val_accuracy: 0.6517

Epoch 3/10

40/40 [=====] - 11s 275ms/step - loss: 0.5423 - accuracy: 0.7350 - val_loss: 0.6758 - val_accuracy: 0.6667

Epoch 4/10

40/40 [=====] - 11s 278ms/step - loss: 0.5159 - accuracy: 0.7600 - val_loss: 0.6514 - val_accuracy: 0.6418

Epoch 5/10

40/40 [=====] - 11s 273ms/step - loss: 0.4857 - accuracy: 0.7725 - val_loss: 0.6488 - val_accuracy: 0.6468

Epoch 6/10

40/40 [=====] - 11s 279ms/step - loss: 0.4613 - accuracy: 0.8000 - val_loss: 0.8004 - val_accuracy: 0.6667

Epoch 7/10

40/40 [=====] - 12s 290ms/step - loss: 0.4269 - accuracy: 0.8200 - val_loss: 0.6464 - val_accuracy: 0.6766

Epoch 8/10

40/40 [=====] - 11s 276ms/step - loss: 0.4609 - accuracy: 0.8125 - val_loss: 0.6205 - val_accuracy: 0.6816

Epoch 9/10

40/40 [=====] - 12s 290ms/step - loss: 0.4454 - accuracy: 0.8037 - val_loss: 0.6331 - val_accuracy: 0.6517

Epoch 10/10

40/40 [=====] - 11s 280ms/step - loss: 0.4129 - accuracy: 0.8375 - val_loss: 0.5907 - val_accuracy: 0.6468

2nd run

[]

Python

main*

0 0 0

0

Spaces: 4

Cell 34 of 34

11:28 a.m.

04/12/2024

