Tutorial

Send an email at the following address: [samuel.rochette@epfedu.fr](http://samuel.rochette@epfedu.fr)

It must contain a HTML export of the python notebook you’ll create.

Open a new python notebook for each of the 6 sessions we will have together. Please use markdown and make it comfortable to read.

# Going deeper in the model

1. Install tensorflow\_datasets library and download the “cats\_vs\_dogs” dataset
2. Display the shape of the images (use the “take” method on your dataset and iterate over it). Is it always the same shape?
3. Define the following architecture, normalize, resize and add batch dimension to your input data and train your model (select only a fraction of dataset for the training so it’s not too long [🐢](https://emojipedia.org/turtle/) and go to the next question while the gradient finds its way [🚀](https://emojis.wiki/fr/fusee/))

| classifier = Sequential()  classifier.add(Convolution2D(32, (3 ,3), input\_shape=(..., ..., 3), activation='relu')) classifier.add(MaxPool2D(pool\_size=(2, 2)))  classifier.add(Convolution2D(20, (3, 3), activation='relu')) classifier.add(MaxPool2D(pool\_size=(2, 2)))  classifier.add(Convolution2D(12, (3, 3), activation='relu')) classifier.add(MaxPool2D(pool\_size=(2, 2)))  classifier.add(Convolution2D(8, (3, 3), activation='relu')) classifier.add(MaxPool2D(pool\_size=(2, 2)))  classifier.add(Flatten())  classifier.add(Dense(activation='relu', units=128)) classifier.add(Dense(activation='sigmoid', units=1))  classifier.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy']) |
| --- |

1. Explain the behaviour of the MaxPool layers
2. How is padding handled in the Convolution2D layers?
3. Compute the number of trainable parameters (write calculation details in a markdown cell, help can be found [here](https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks#overview))
4. Compare the result with the output of classifier.summary() Do you find the same result? (you should :p)
5. This model is very close to LeNet architecture (model used by LeCun in 1989 to recognise digits for the US Postal). Compare the number of parameters with AlexNet and with Inception v3. What are your thoughts about that?
6. What are the different activation functions? What is important for those activation and why isn’t the sigmoid chosen everywhere?
7. What is the stride parameter in conv layer and why is it used?
8. Add a dropout layer and explain what it does and why it can be useful

# Monitoring

1. Explain what a callback is and visit the [tf.keras.callback](https://www.tensorflow.org/api_docs/python/tf/keras/callbacks) page
2. Add an [early stopping callback](https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping) to your previous model. How is this useful for the training?
3. Add a [learning rate scheduler callback](https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/LearningRateScheduler). Explain the behaviour.
4. Add a [tensorboard callback](https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/TensorBoard) and open a tensorboard server.
5. Add a lambda callback