

Le jeu de données porte sur 1558 caractéristiques d'images. La variable d'intérêt classe les images en deux groupes selon qu'elles ont des images de publicité (1/ad) ou non (0/no ad). Les caractéristiques/prédicteurs sont tous quantitatifs.

```
In [7]: print(df)
      height  width  aratio  ...  caption.for  caption.you  class
0         125    125  1.0000  ...             0             0       1
1          57    468  8.2105  ...             0             0       1
2          33    230  6.9696  ...             0             0       1
3          60    468  7.8000  ...             0             0       1
4          60    468  7.8000  ...             0             0       1
...      ...    ...    ...  ...      ...      ...      ...
3274       170     94  0.5529  ...             0             0       0
3275       101    140  1.3861  ...             0             0       0
3276         23    120  5.2173  ...             0             0       0
3277          0      0  0.0000  ...             0             0       0
3278         40     40  1.0000  ...             0             0       0

[3279 rows x 1559 columns]
```

L'architecture du RN conçu est la suivante :

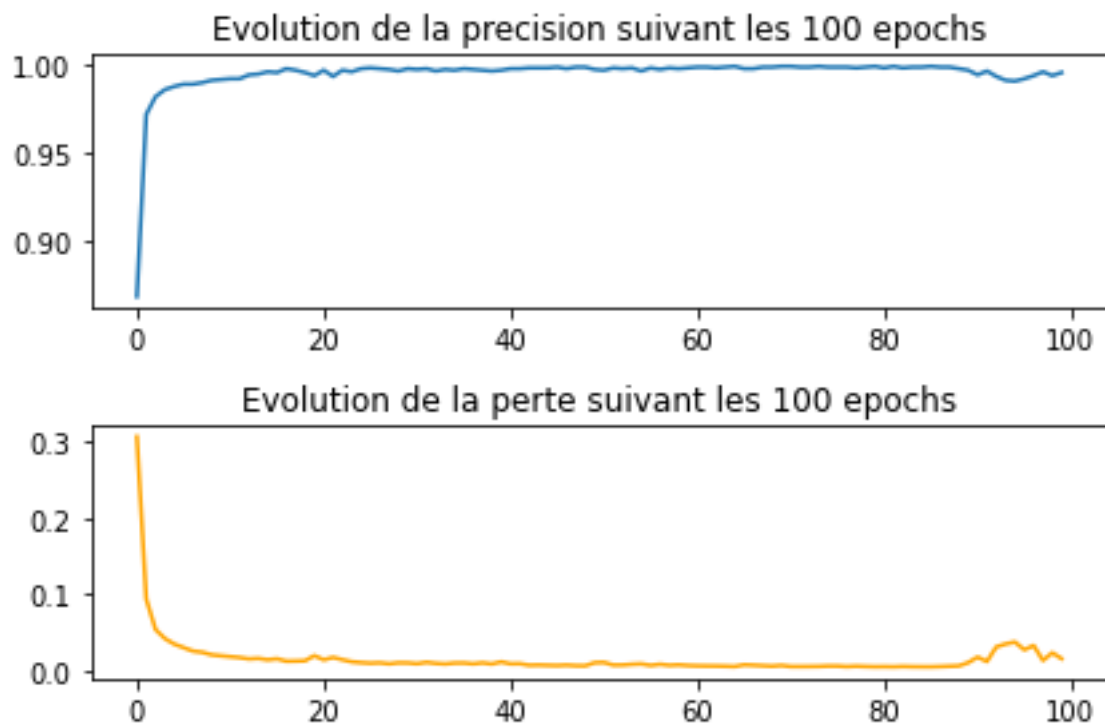
```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping
model = Sequential()

nb_entries = 1558
nb_neurons_hidden_layer_1, nb_neurons_hidden_layer_2 = 64, 16 # décroissant

model.add(Dense(nb_neurons_hidden_layer_1, input_dim = nb_entries, activation = "relu"))
model.add(Dense(nb_neurons_hidden_layer_2, activation = "softplus"))
model.add(Dense(1, activation = "sigmoid"))

model.compile(loss="binary_crossentropy",
              optimizer=Adam(learning_rate=0.01),
              metrics=["accuracy"])
```

Le dataset est réparti en train set et test set. Pas de validation croisée. Le graphique suivant donne l'évolution de la métrique accuracy et du loss suivant les epochs.



Pour plus une lecture plus précise, les valeurs optimales pour le jeu d'apprentissage ont été isolés :

```
=====
Train set
Précision = 99.52069520950317 %
Fonction de perte = 0.011794157326221466
=====
Test set
Précision = 97.86585569381714 %
Fonction de perte = 0.11407627910375595
=====
meilleure epoch = 64
meilleure précision (train set) = 0.9986928105354309
loss correspondant = 0.00538247637450695
```