CNN – MNIST Digits Classification

Objectif : entraîner un CNN sur MNIST (10 chiffres) et analyser l'impact du nombre de convolutions (2 vs 3).

In [5]:
```
import numpy as np
import matplotlib.pyplot as plt
import os

from tensorflow import keras
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropou
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
```

In [6]:
```
IMG_SIZE = (28, 28)
BATCH_SIZE = 32
EPOCHS = 20
```

In [7]:
```
(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()

x_train = x_train.astype("float32") / 255.0
x_test  = x_test.astype("float32") / 255.0

x_train = np.expand_dims(x_train, axis=-1)  # (60000, 28, 28, 1)
x_test  = np.expand_dims(x_test, axis=-1)   # (10000, 28, 28, 1)

y_train_cat = keras.utils.to_categorical(y_train, num_classes=10)
y_test_cat  = keras.utils.to_categorical(y_test, num_classes=10)
```

```
x_train: (60000, 28, 28) uint8
y_train: (60000,) uint8
x_test : (10000, 28, 28) uint8
y_test : (10000,) uint8
x_train: (60000, 28, 28, 1)
y_train_cat: (60000, 10)
```

In [8]:
```
def build_cnn(nb_conv):
    model = Sequential()

    # 1ère convolution
    model.add(Conv2D(32, (3,3), activation='relu', input_shape=(28,28,1)))
    model.add(MaxPooling2D((2,2)))

    # 2ème convolution
    model.add(Conv2D(32, (3,3), activation='relu'))
    model.add(MaxPooling2D((2,2)))

    # 3ème convolution OPTIONNELLE
    if nb_conv == 3:
        model.add(Conv2D(64, (3,3), activation='relu'))
        model.add(MaxPooling2D((2,2)))

    model.add(Flatten())
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5))
```

```
    model.add(Dense(10, activation='softmax'))  # 10 classes (0..9)

    model.compile(
        optimizer='adam',
        loss='categorical_crossentropy',
        metrics=['accuracy']
    )

    return model
```

In [9]:
```
def get_callbacks(nb_conv):
    ckpt_dir = f"checkpoints_mnist_cnn_{nb_conv}"
    os.makedirs(ckpt_dir, exist_ok=True)

    checkpoint = ModelCheckpoint(
        filepath=os.path.join(ckpt_dir, "weights_epoch_{epoch:02d}.h5"),
        monitor="val_accuracy",
        save_weights_only=True,
        save_best_only=True,
        verbose=1
    )

    early_stop = EarlyStopping(
        monitor="val_accuracy",
        patience=5,
        restore_best_weights=True,
        verbose=1
    )

    return checkpoint, early_stop, ckpt_dir
```

In [10]:
```
model_2 = build_cnn(nb_conv=2)
checkpoint, early_stop, ckpt_dir = get_callbacks(2)

history_2 = model_2.fit(
    x_train, y_train_cat,
    batch_size=BATCH_SIZE,
    epochs=EPOCHS,
    validation_data=(x_test, y_test_cat),
    callbacks=[checkpoint, early_stop],
    verbose=1
)
```

```
Epoch 1/20
1870/1875 [============================>.] - ETA: 0s - loss: 0.2266 - accuracy:
0.9315
Epoch 1: val_accuracy improved from -inf to 0.98620, saving model to checkpoints_
mnist_cnn_2\weights_epoch_01.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.2265 - accura
cy: 0.9316 - val_loss: 0.0459 - val_accuracy: 0.9862
Epoch 2/20
1873/1875 [============================>.] - ETA: 0s - loss: 0.0857 - accuracy:
0.9738
Epoch 2: val_accuracy improved from 0.98620 to 0.98830, saving model to checkpoin
ts_mnist_cnn_2\weights_epoch_02.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0856 - accura
cy: 0.9738 - val_loss: 0.0345 - val_accuracy: 0.9883
Epoch 3/20
1869/1875 [============================>.] - ETA: 0s - loss: 0.0634 - accuracy:
0.9807
Epoch 3: val_accuracy improved from 0.98830 to 0.99010, saving model to checkpoin
ts_mnist_cnn_2\weights_epoch_03.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0635 - accura
cy: 0.9807 - val_loss: 0.0310 - val_accuracy: 0.9901
Epoch 4/20
1868/1875 [============================>.] - ETA: 0s - loss: 0.0499 - accuracy:
0.9846
Epoch 4: val_accuracy did not improve from 0.99010
1875/1875 [==============================] - 12s 6ms/step - loss: 0.0500 - accura
cy: 0.9846 - val_loss: 0.0292 - val_accuracy: 0.9899
Epoch 5/20
1873/1875 [============================>.] - ETA: 0s - loss: 0.0458 - accuracy:
0.9861
Epoch 5: val_accuracy improved from 0.99010 to 0.99150, saving model to checkpoin
ts_mnist_cnn_2\weights_epoch_05.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0458 - accura
cy: 0.9861 - val_loss: 0.0289 - val_accuracy: 0.9915
Epoch 6/20
1870/1875 [============================>.] - ETA: 0s - loss: 0.0384 - accuracy:
0.9883
Epoch 6: val_accuracy improved from 0.99150 to 0.99230, saving model to checkpoin
ts_mnist_cnn_2\weights_epoch_06.h5
1875/1875 [==============================] - 12s 6ms/step - loss: 0.0385 - accura
cy: 0.9883 - val_loss: 0.0232 - val_accuracy: 0.9923
Epoch 7/20
1874/1875 [============================>.] - ETA: 0s - loss: 0.0340 - accuracy:
0.9894
Epoch 7: val_accuracy did not improve from 0.99230
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0340 - accura
cy: 0.9894 - val_loss: 0.0243 - val_accuracy: 0.9916
Epoch 8/20
1871/1875 [============================>.] - ETA: 0s - loss: 0.0295 - accuracy:
0.9908
Epoch 8: val_accuracy did not improve from 0.99230
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0295 - accura
cy: 0.9908 - val_loss: 0.0286 - val_accuracy: 0.9914
Epoch 9/20
1868/1875 [============================>.] - ETA: 0s - loss: 0.0264 - accuracy:
0.9918
Epoch 9: val_accuracy improved from 0.99230 to 0.99280, saving model to checkpoin
ts_mnist_cnn_2\weights_epoch_09.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0263 - accura
cy: 0.9918 - val_loss: 0.0235 - val_accuracy: 0.9928
```

```
Epoch 10/20
1870/1875 [============================>.] - ETA: 0s - loss: 0.0257 - accuracy:
0.9920
Epoch 10: val_accuracy improved from 0.99280 to 0.99330, saving model to checkpoi
nts_mnist_cnn_2\weights_epoch_10.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0257 - accura
cy: 0.9919 - val_loss: 0.0225 - val_accuracy: 0.9933
Epoch 11/20
1872/1875 [============================>.] - ETA: 0s - loss: 0.0229 - accuracy:
0.9926
Epoch 11: val_accuracy did not improve from 0.99330
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0228 - accura
cy: 0.9926 - val_loss: 0.0271 - val_accuracy: 0.9918
Epoch 12/20
1867/1875 [============================>.] - ETA: 0s - loss: 0.0218 - accuracy:
0.9928
Epoch 12: val_accuracy improved from 0.99330 to 0.99340, saving model to checkpoi
nts_mnist_cnn_2\weights_epoch_12.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0217 - accura
cy: 0.9929 - val_loss: 0.0247 - val_accuracy: 0.9934
Epoch 13/20
1873/1875 [============================>.] - ETA: 0s - loss: 0.0193 - accuracy:
0.9938
Epoch 13: val_accuracy did not improve from 0.99340
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0193 - accura
cy: 0.9938 - val_loss: 0.0225 - val_accuracy: 0.9933
Epoch 14/20
1866/1875 [============================>.] - ETA: 0s - loss: 0.0188 - accuracy:
0.9939
Epoch 14: val_accuracy improved from 0.99340 to 0.99360, saving model to checkpoi
nts_mnist_cnn_2\weights_epoch_14.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0187 - accura
cy: 0.9940 - val_loss: 0.0270 - val_accuracy: 0.9936
Epoch 15/20
1873/1875 [============================>.] - ETA: 0s - loss: 0.0178 - accuracy:
0.9939
Epoch 15: val_accuracy did not improve from 0.99360
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0177 - accura
cy: 0.9939 - val_loss: 0.0238 - val_accuracy: 0.9933
Epoch 16/20
1866/1875 [============================>.] - ETA: 0s - loss: 0.0172 - accuracy:
0.9945
Epoch 16: val_accuracy did not improve from 0.99360
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0172 - accura
cy: 0.9945 - val_loss: 0.0263 - val_accuracy: 0.9924
Epoch 17/20
1874/1875 [============================>.] - ETA: 0s - loss: 0.0165 - accuracy:
0.9946
Epoch 17: val_accuracy did not improve from 0.99360
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0165 - accura
cy: 0.9946 - val_loss: 0.0285 - val_accuracy: 0.9924
Epoch 18/20
1866/1875 [============================>.] - ETA: 0s - loss: 0.0138 - accuracy:
0.9954
Epoch 18: val_accuracy did not improve from 0.99360
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0138 - accura
cy: 0.9954 - val_loss: 0.0275 - val_accuracy: 0.9931
Epoch 19/20
1868/1875 [============================>.] - ETA: 0s - loss: 0.0151 - accuracy:
0.9950
```

```
Epoch 19: val_accuracy did not improve from 0.99360
Restoring model weights from the end of the best epoch: 14.
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0151 - accura
cy: 0.9950 - val_loss: 0.0286 - val_accuracy: 0.9926
Epoch 19: early stopping
```

In [11]:
```
model_3 = build_cnn(nb_conv=3)
checkpoint, early_stop, ckpt_dir = get_callbacks(3)

history_3 = model_3.fit(
    x_train, y_train_cat,
    batch_size=BATCH_SIZE,
    epochs=EPOCHS,
    validation_data=(x_test, y_test_cat),
    callbacks=[checkpoint, early_stop],
    verbose=1
)
```

```
Epoch 1/20
1870/1875 [============================>.] - ETA: 0s - loss: 0.2852 - accuracy:
0.9129
Epoch 1: val_accuracy improved from -inf to 0.97750, saving model to checkpoints_
mnist_cnn_3\weights_epoch_01.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.2849 - accura
cy: 0.9131 - val_loss: 0.0744 - val_accuracy: 0.9775
Epoch 2/20
1875/1875 [==============================] - ETA: 0s - loss: 0.1011 - accuracy:
0.9706
Epoch 2: val_accuracy improved from 0.97750 to 0.98080, saving model to checkpoin
ts_mnist_cnn_3\weights_epoch_02.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.1011 - accura
cy: 0.9706 - val_loss: 0.0629 - val_accuracy: 0.9808
Epoch 3/20
1868/1875 [============================>.] - ETA: 0s - loss: 0.0729 - accuracy:
0.9787
Epoch 3: val_accuracy improved from 0.98080 to 0.98180, saving model to checkpoin
ts_mnist_cnn_3\weights_epoch_03.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0727 - accura
cy: 0.9787 - val_loss: 0.0627 - val_accuracy: 0.9818
Epoch 4/20
1866/1875 [============================>.] - ETA: 0s - loss: 0.0623 - accuracy:
0.9818
Epoch 4: val_accuracy improved from 0.98180 to 0.98240, saving model to checkpoin
ts_mnist_cnn_3\weights_epoch_04.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0624 - accura
cy: 0.9818 - val_loss: 0.0612 - val_accuracy: 0.9824
Epoch 5/20
1875/1875 [==============================] - ETA: 0s - loss: 0.0535 - accuracy:
0.9839
Epoch 5: val_accuracy improved from 0.98240 to 0.98540, saving model to checkpoin
ts_mnist_cnn_3\weights_epoch_05.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0535 - accura
cy: 0.9839 - val_loss: 0.0478 - val_accuracy: 0.9854
Epoch 6/20
1874/1875 [============================>.] - ETA: 0s - loss: 0.0455 - accuracy:
0.9864
Epoch 6: val_accuracy improved from 0.98540 to 0.98610, saving model to checkpoin
ts_mnist_cnn_3\weights_epoch_06.h5
1875/1875 [==============================] - 12s 6ms/step - loss: 0.0455 - accura
cy: 0.9863 - val_loss: 0.0524 - val_accuracy: 0.9861
Epoch 7/20
1871/1875 [============================>.] - ETA: 0s - loss: 0.0387 - accuracy:
0.9890
Epoch 7: val_accuracy did not improve from 0.98610
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0387 - accura
cy: 0.9890 - val_loss: 0.0531 - val_accuracy: 0.9845
Epoch 8/20
1868/1875 [============================>.] - ETA: 0s - loss: 0.0364 - accuracy:
0.9891
Epoch 8: val_accuracy improved from 0.98610 to 0.98840, saving model to checkpoin
ts_mnist_cnn_3\weights_epoch_08.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0364 - accura
cy: 0.9891 - val_loss: 0.0444 - val_accuracy: 0.9884
Epoch 9/20
1870/1875 [============================>.] - ETA: 0s - loss: 0.0317 - accuracy:
0.9904
Epoch 9: val_accuracy did not improve from 0.98840
1875/1875 [==============================] - 12s 6ms/step - loss: 0.0316 - accura
```

```
cy: 0.9904 - val_loss: 0.0466 - val_accuracy: 0.9874
Epoch 10/20
1866/1875 [============================>.] - ETA: 0s - loss: 0.0280 - accuracy:
0.9913
Epoch 10: val_accuracy improved from 0.98840 to 0.98880, saving model to checkpoi
nts_mnist_cnn_3\weights_epoch_10.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0281 - accura
cy: 0.9913 - val_loss: 0.0469 - val_accuracy: 0.9888
Epoch 11/20
1866/1875 [============================>.] - ETA: 0s - loss: 0.0255 - accuracy:
0.9923
Epoch 11: val_accuracy did not improve from 0.98880
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0254 - accura
cy: 0.9923 - val_loss: 0.0504 - val_accuracy: 0.9872
Epoch 12/20
1871/1875 [============================>.] - ETA: 0s - loss: 0.0246 - accuracy:
0.9925
Epoch 12: val_accuracy improved from 0.98880 to 0.98950, saving model to checkpoi
nts_mnist_cnn_3\weights_epoch_12.h5
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0247 - accura
cy: 0.9924 - val_loss: 0.0521 - val_accuracy: 0.9895
Epoch 13/20
1874/1875 [============================>.] - ETA: 0s - loss: 0.0227 - accuracy:
0.9926
Epoch 13: val_accuracy did not improve from 0.98950
1875/1875 [==============================] - 12s 6ms/step - loss: 0.0226 - accura
cy: 0.9926 - val_loss: 0.0495 - val_accuracy: 0.9885
Epoch 14/20
1870/1875 [============================>.] - ETA: 0s - loss: 0.0195 - accuracy:
0.9936
Epoch 14: val_accuracy did not improve from 0.98950
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0195 - accura
cy: 0.9936 - val_loss: 0.0630 - val_accuracy: 0.9872
Epoch 15/20
1871/1875 [============================>.] - ETA: 0s - loss: 0.0198 - accuracy:
0.9940
Epoch 15: val_accuracy did not improve from 0.98950
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0198 - accura
cy: 0.9940 - val_loss: 0.0591 - val_accuracy: 0.9885
Epoch 16/20
1874/1875 [============================>.] - ETA: 0s - loss: 0.0184 - accuracy:
0.9940
Epoch 16: val_accuracy did not improve from 0.98950
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0184 - accura
cy: 0.9940 - val_loss: 0.0563 - val_accuracy: 0.9864
Epoch 17/20
1872/1875 [============================>.] - ETA: 0s - loss: 0.0167 - accuracy:
0.9948
Epoch 17: val_accuracy did not improve from 0.98950
Restoring model weights from the end of the best epoch: 12.
1875/1875 [==============================] - 11s 6ms/step - loss: 0.0167 - accura
cy: 0.9948 - val_loss: 0.0566 - val_accuracy: 0.9868
Epoch 17: early stopping
```

In [12]:

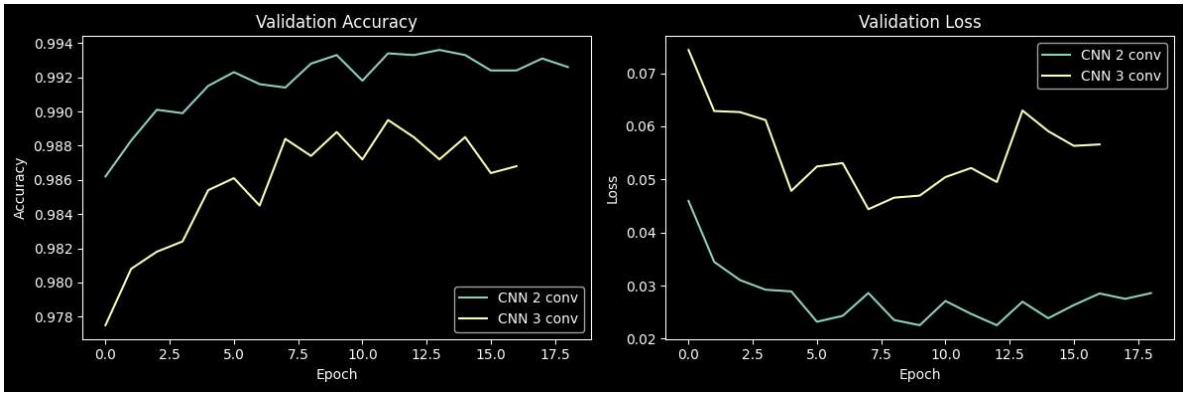
```
plt.figure(figsize=(12,4))

plt.subplot(1,2,1)
plt.plot(history_2.history["val_accuracy"], label="CNN 2 conv")
plt.plot(history_3.history["val_accuracy"], label="CNN 3 conv")
plt.title("Validation Accuracy")
```

```
plt.xlabel("Epoch")
plt.ylabel("Accuracy")
plt.legend()

plt.subplot(1,2,2)
plt.plot(history_2.history["val_loss"], label="CNN 2 conv")
plt.plot(history_3.history["val_loss"], label="CNN 3 conv")
plt.title("Validation Loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()

plt.tight_layout()
plt.show()
```



Conclusion

Les résultats montrent que les deux modèles atteignent une précision élevée sur l'ensemble de validation, confirmant que même une architecture relativement simple est suffisante pour capturer les caractéristiques principales des chiffres MNIST. Toutefois, le CNN à 3 couches de convolution présente une légère amélioration des performances par rapport au modèle à 2 couches, traduisant une meilleure capacité à extraire des caractéristiques plus fines et plus abstraites.

L'ajout de la troisième couche de convolution permet notamment une convergence plus stable et une réduction plus rapide de la perte, sans provoquer de surapprentissage significatif. Cette amélioration reste cependant modérée, ce qui s'explique par la relative simplicité du dataset MNIST, où les motifs visuels sont bien structurés et peu complexes.

En conclusion, pour le dataset MNIST, le CNN à 2 convolutions constitue un très bon compromis entre simplicité et performance, tandis que le CNN à 3 convolutions offre un gain marginal de précision au prix d'une complexité légèrement supérieure. Ce comportement illustre que l'augmentation de la profondeur du réseau est particulièrement bénéfique lorsque la complexité visuelle des données augmente.

In [ ]: