/01

outil de visualisation pour un réseau neuronal convolutif

Sommaire

/03

Creation du repo

Progrès

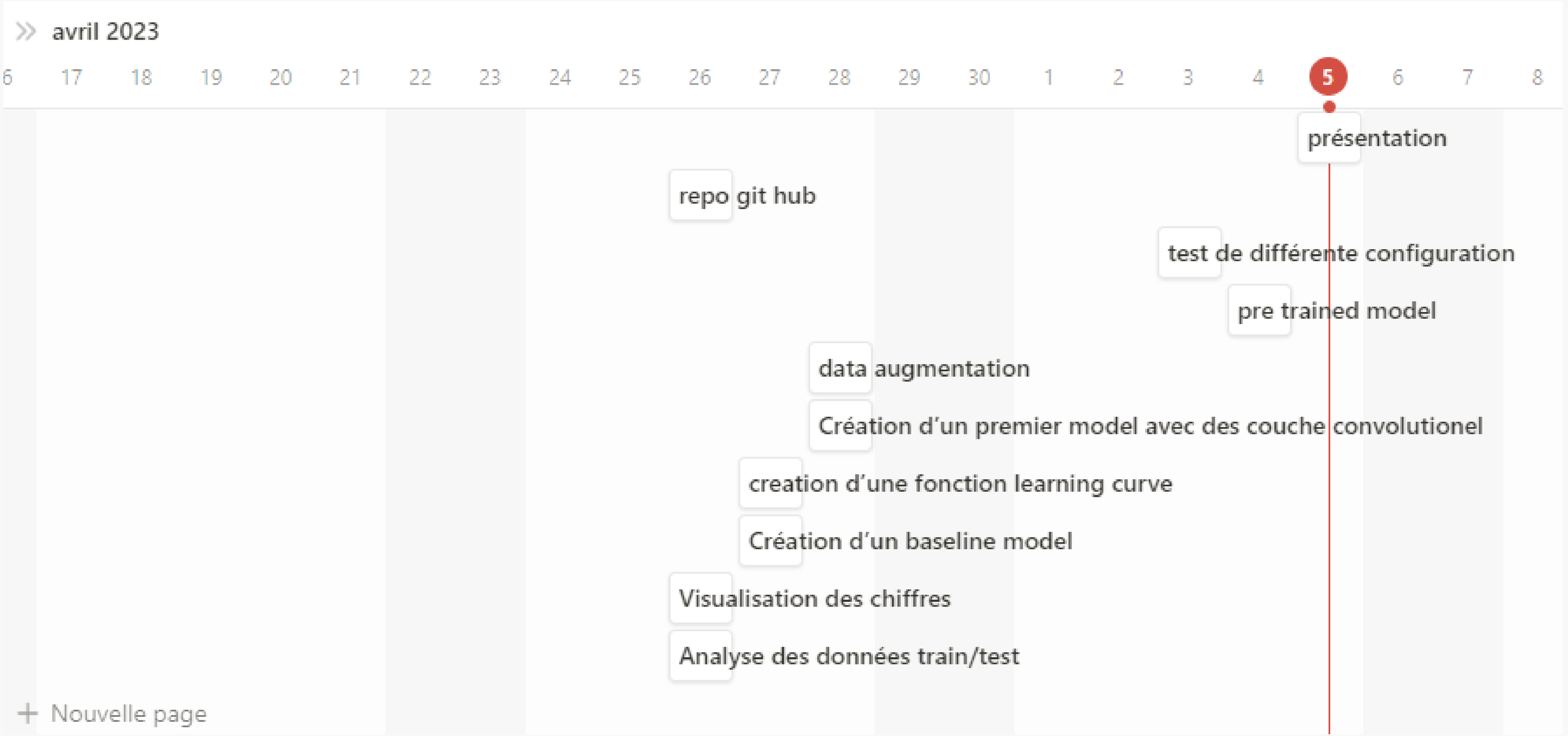
Les présentations sont des outils de communication pouvant être utilisés en tant que conférences.

Résultats importants

Les présentations sont des outils de communication pouvant être utilisés en tant que conférences.

/04

Planning —


avril 2023
6
17
18
19
20
21
22
23
24
25
26
27
28
29
30
1
2
3
4
5
6
7
8
présentation
repo git hub
test de différente configuration
pre trained model
data augmentation
Création d'un premier model avec des couche convolutionel
creation d'une fonction learning curve
Création d'un baseline model
Visualisation des chiffres
Analyse des données train/test
Nouvelle page

/06

Baseline

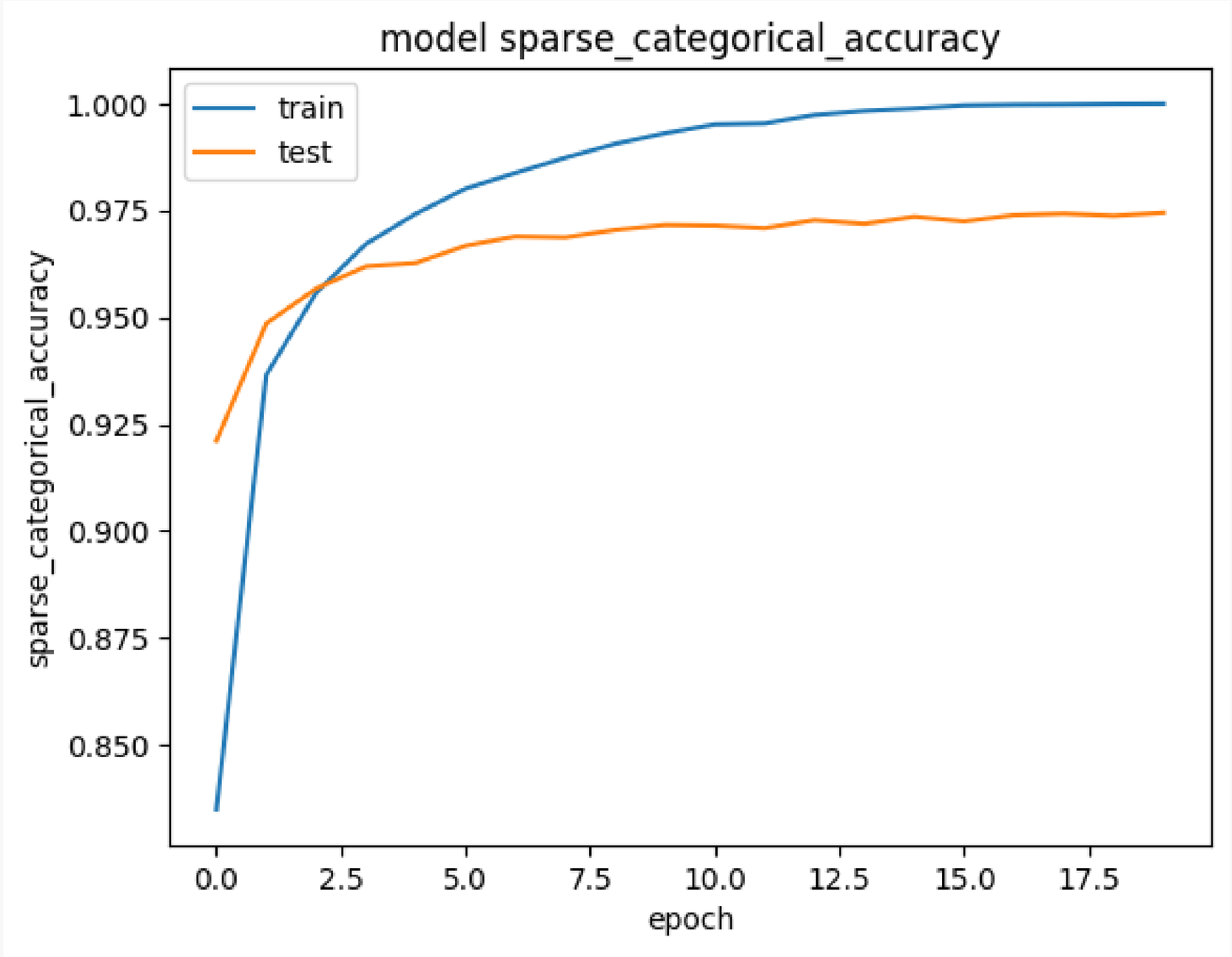
```
model = Sequential()
model.add(Input(784))
# model.add(Flatten(input_shape = (28,28)))
# model.add(RandomFlip("horizontal"))
# model.add(RandomRotation(0.1))
model.add(Dense(397, activation='relu'))
model.add(Dense(100, activation='relu'))
model.add(Dense(10, activation='softmax'))

# Compile model
model.compile(loss = sparse_categorical_crossentropy, optimizer='adam', metrics=["sparse_categorical_accuracy"])

# Fit the model
history = model.fit(x = X_train, y = y_train,epochs=20, batch_size= 512, validation_split= 0.33 )

# Save model
model.save("data/model_baseline.h5")

learning_curve_dl(history,"sparse_categorical_accuracy")
```


model sparse_categorical_accuracy
train
test
1.000
0.975
0.950
0.925
0.900
0.875
0.850
sparse_categorical_accuracy
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
epoch

/09

—

Convolutional 1st iter

Data augmentation

```
datagen = ImageDataGenerator(
    rotation_range=10,       # randomly rotate the images by up to 10 degrees
    zoom_range=0.1,          # randomly zoom the images by up to 10%
    width_shift_range=0.1,   # randomly shift the images horizontally by up to 10%
    height_shift_range=0.1, # randomly shift the images vertically by up to 10%
    horizontal_flip=True,    # randomly flip the images horizontally
    vertical_flip=False,     # don't randomly flip the images vertically
    fill_mode='nearest'      # fill any empty pixels with the nearest value
)
```

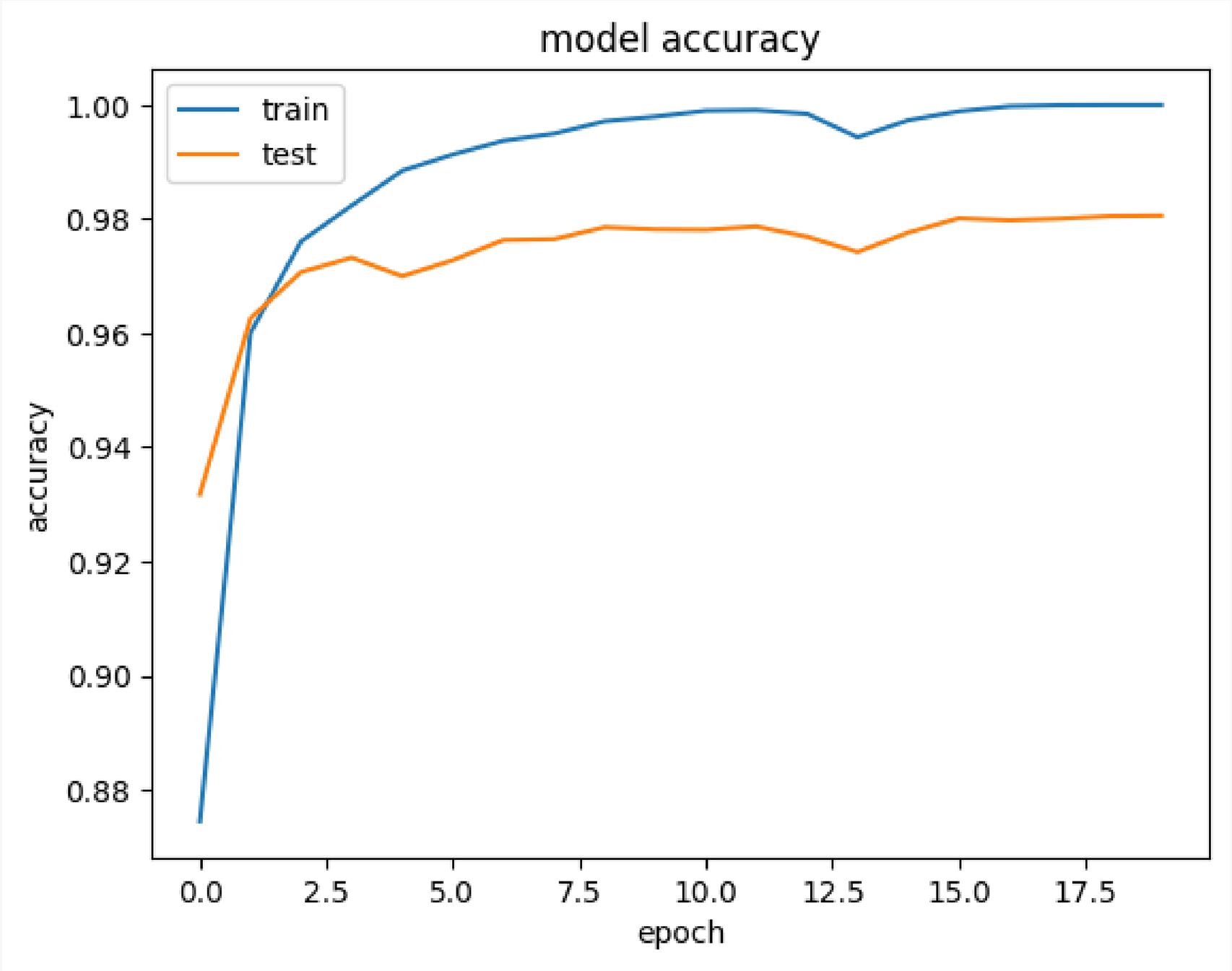
Architecture

```
model = Sequential()
# model.add(Input(784))
model.add(Conv2D(32, (3,3), activation = 'relu', input_shape=(28, 28, 1)))
model.add(Flatten())
# model.add(RandomFlip("horizontal"))
# model.add(RandomRotation(0.1))

model.add(Dense(32, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(10, activation='softmax'))

# Compile model
model.compile(loss = sparse_categorical_crossentropy, optimizer='adam', metrics=["accuracy"])

# Fit the model
history = model.fit(x = x_train, y = y_train, epochs=20, validation_split= 0.33,batch_size = 128 )
```


model accuracy
train
test
1.00
0.98
0.96
0.94
0.92
0.90
0.88
accuracy
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
epoch

/09

—

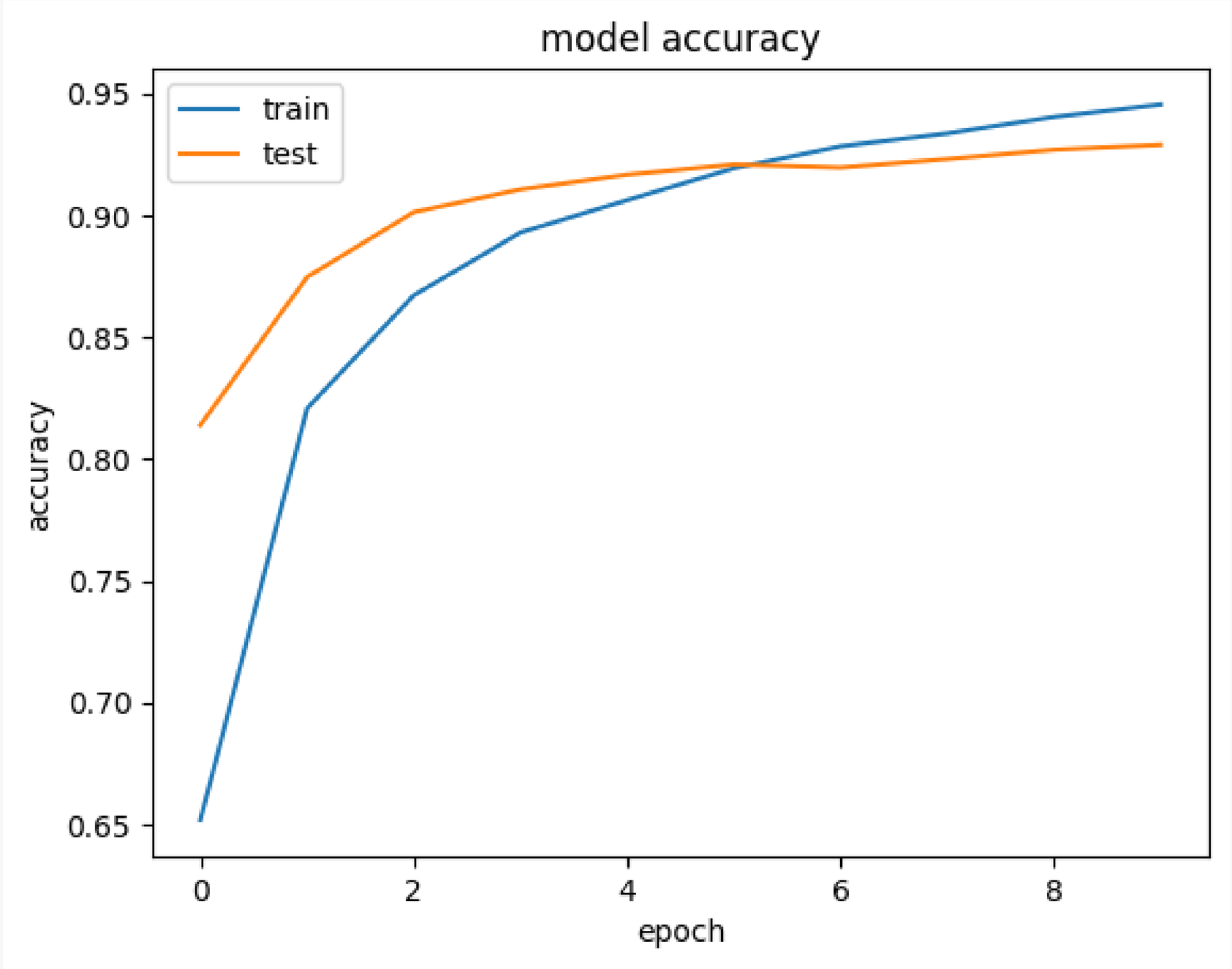
Convolutional 2nd iter

Data augmentation

```
# Create a data generator for data augmentation
datagen = ImageDataGenerator(
    rotation_range=90,       # randomly rotate the images by up to 10 degrees
    zoom_range=0.3,          # randomly zoom the images by up to 10%
    width_shift_range=0.1,   # randomly shift the images horizontally by up to 10%
    height_shift_range=0.1, # randomly shift the images vertically by up to 10%
    horizontal_flip=True,    # randomly flip the images horizontally
    vertical_flip=False,     # don't randomly flip the images vertically
    fill_mode='nearest'      # fill any empty pixels with the nearest value
)
```

Architecture

```
model = Sequential()
# model.add(Input(784))
model.add(BatchNormalization(input_shape=(28, 28, 1)))
model.add(Conv2D(32, (3,3), activation = 'relu'))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64, (3,3), activation = 'relu'))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(64, (3,3), activation = 'relu'))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dense(10, activation='softmax'))
```


model accuracy
train
test
0.95
0.90
0.85
0.80
0.75
0.70
0.65
accuracy
0
2
4
6
8
epoch

Perspective

- **Tester d'autres modèles**
- **Repenser le design de l'application**

Merci !