3 Implementarea rețelelor neuronale folosind Keras

Keras[1] este un API de învățare profundă de nivel înalt care ne permite să construim, să antrenăm, să evaluăm și să utilizăm cu ușurință diferite tipuri de rețele neuronale.

3.1 Utilizarea modelului secvențial

În continuare, vom folosi API-ul secvențial din Keras pentru a crea un model de rețea neuronală pentru regresie, având două straturi ascunse și un strat de ieșire:

```
from tensorflow.keras import models, layers

model = models.Sequential()
model.add(layers.Dense(units=32, activation='relu',
                       input_shape=X_train.shape[1:]))
model.add(layers.Dense(units=16, activation='relu'))
model.add(layers.Dense(units=1))
```

În loc să adăugăm straturile unul câte unul, putem furniza o listă de straturi atunci când creăm modelul secvențial:

```
model = models.Sequential([
    layers.Dense(units=32, activation='relu',
                 input_shape=X_train.shape[1:]),
    layers.Dense(units=16, activation='relu'),
    layers.Dense(units=1)
    ])
```

API-ul secvențial din Keras este destul de ușor de utilizat. Deși modelele secvențiale sunt foarte uzuale, uneori este util să se construiască rețele neuronale cu arhitecturi mai complexe, sau cu intrări sau ieșiri multiple. În acest scop, Keras oferă API-ul funcțional.

[1] https://keras.io/

3.2 Utilizarea modelului funcțional

Exemplul 1. *Un model simplu cu o singură intrare și o singură ieșire*

Vom crea din nou modelul folosit anterior pentru regresie utilizând, de data aceasta, API-ul funcțional.

```
input = layers.Input(shape=X_train.shape[1:])
hidden1 = layers.Dense(units=32, activation='relu')(input)
hidden2 = layers.Dense(units=16, activation='relu')(hidden1)
output = layers.Dense(units=1)(hidden2)
model = models.Model(inputs=[input], outputs=[output])
```

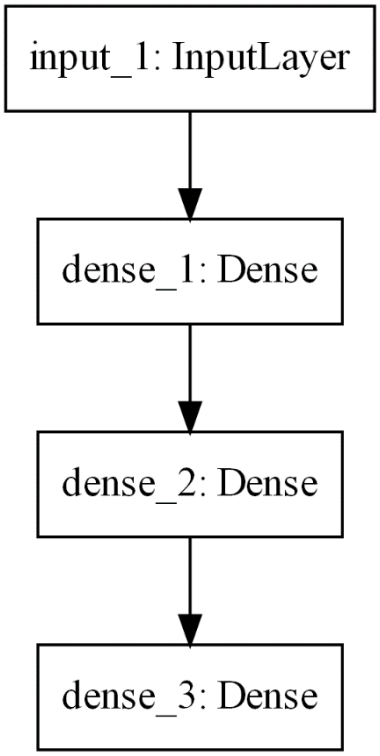

Fig. 3.1 Exemplu de model cu o singură intrare și o singură ieșire

Exemplul 2. *Modelul de rețea neuronală Wide & Deep*

Rețeaua neuronală Wide & Deep (Cheng et al., 2016) conectează toate sau o parte din intrări direct la stratul de ieșire.

Calea scurtă poate fi, de asemenea, utilizată pentru a furniza rețelei neuronale caracteristici proiectate manual.

Vom crea un astfel de model pentru aceeași problemă de regresie:

```
input = layers.Input(shape=X_train.shape[1:])
hidden1 = layers.Dense(units=32, activation='relu')(input)
hidden2 = layers.Dense(units=16, activation='relu')(hidden1)
concat = layers.concatenate([input, hidden2])
output = layers.Dense(units=1)(concat)
model = models.Model(inputs=[input], outputs=[output])
```

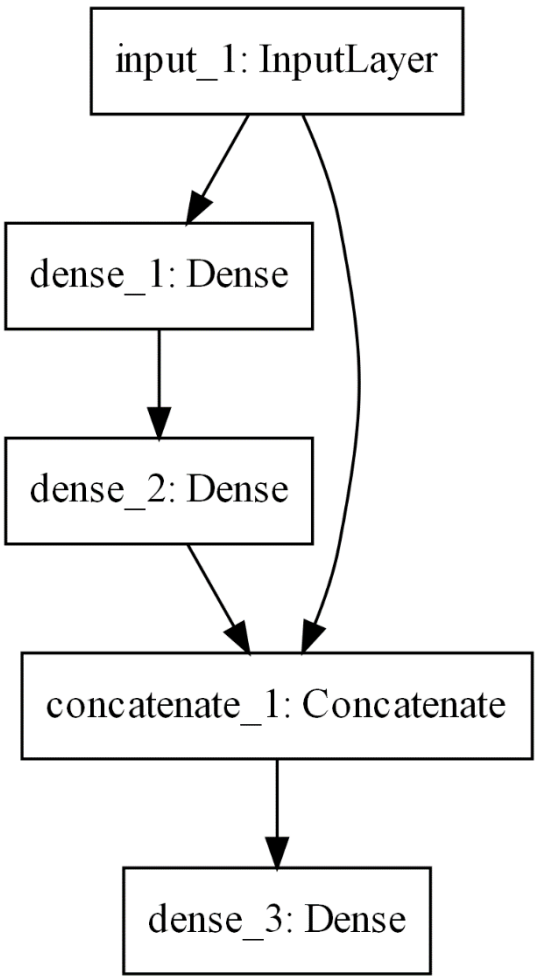

Fig. 3.2 Exemplu de rețea neuronală Wide & Deep

Exemplul 3. *Un model cu mai multe intrări și o singură ieșire*

Un model de predicție a prețului mărfurilor care constă din două intrări: imaginea produsului și marca produsului.

```
img_input = layers.Input(shape=(128, 128, 3),
                         name='Image_Input')
info_input = layers.Input(shape=(1, ),
                          name='Information_Input')
hidden1_1 = layers.Conv2D(filters=64, kernel_size=(5, 5),
                          strides=(2, 2),
                          activation='relu')(img_input)
hidden1_2 = layers.Conv2D(filters=32, kernel_size=(5, 5),
                          strides=(2, 2),
                          activation='relu')(hidden1_1)
hidden1_2_ft = layers.Flatten()(hidden1_2)
hidden1_3 = layers.Dense(units=64,
                         activation='relu')(info_input)
concat = layers.concatenate([hidden1_2_ft, hidden1_3])
hidden2 = layers.Dense(units=64, activation='relu')(concat)
output = layers.Dense(units=1, name='Output')(hidden2)
model = models.Model(inputs=[img_input, info_input],
                     outputs=output)
```

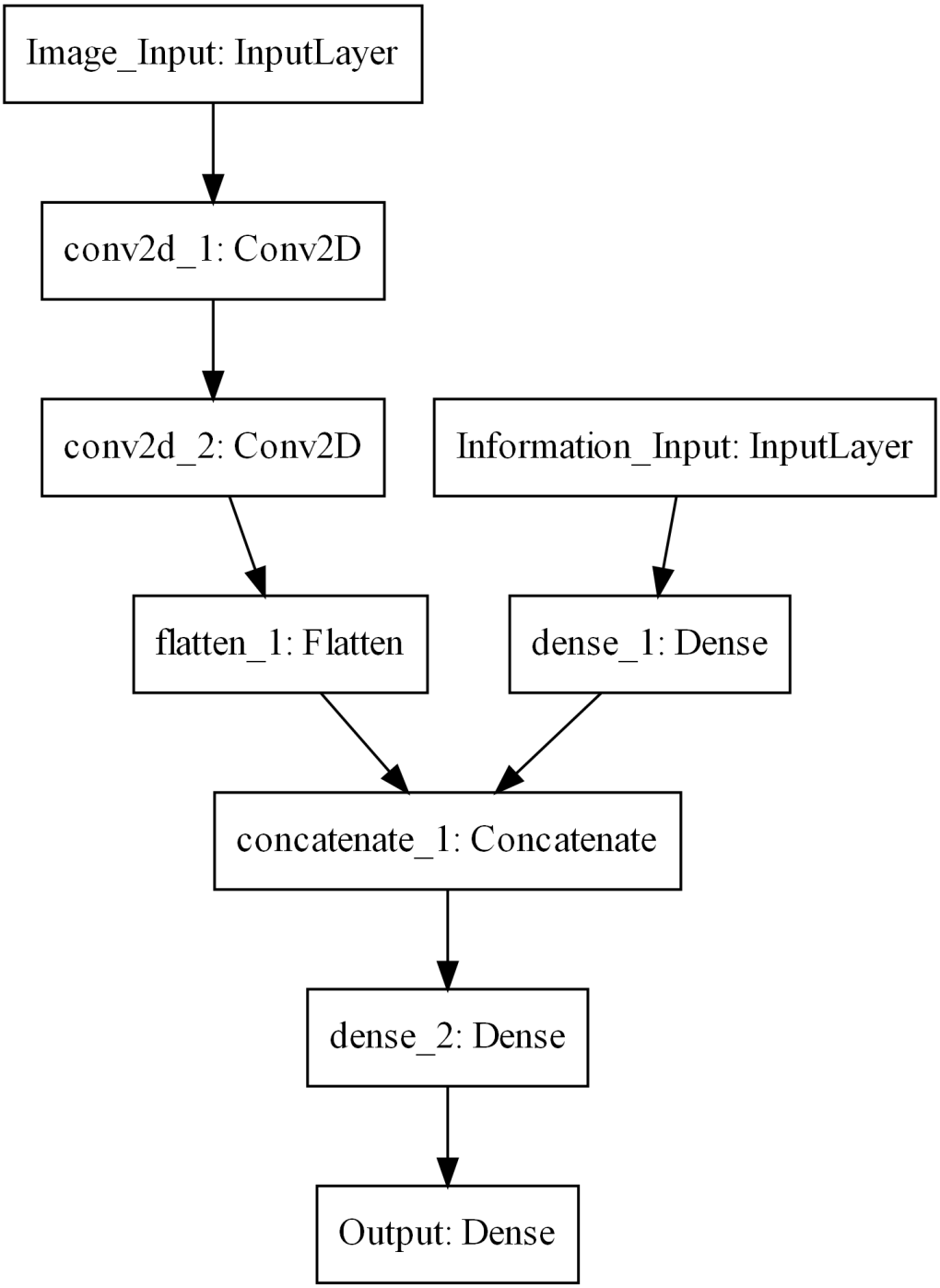

Fig. 3.3 Exemplu de model cu mai multe intrări și o singură ieșire

Exemplul 4. *Un model cu o singură intrare și mai multe ieșiri*

Intrarea modelului este imaginea feței unei persoane, având dimensiunea de (128, 128, 3).

```
input = layers.Input(shape=(128, 128, 3), name='Input')
hidden1 = layers.Conv2D(filters=64, kernel_size=(3, 3),
                        activation='relu')(input)
hidden2 = layers.Conv2D(filters=64, kernel_size=(3, 3),
                        strides=(2, 2),
                        activation='relu')(hidden1)
hidden3 = layers.Conv2D(filters=64, kernel_size=(3, 3),
                        strides=(2, 2),
                        activation='relu')(hidden2)
flatten = layers.Flatten()(hidden3)
age_output = layers.Dense(units=1, name='Age_Output')(flatten)
gender_output = layers.Dense(units=1, activation='sigmoid',
                             name='Gender_Output')(flatten)
model = models.Model(inputs=input,
                     outputs=[age_output, gender_output])
```

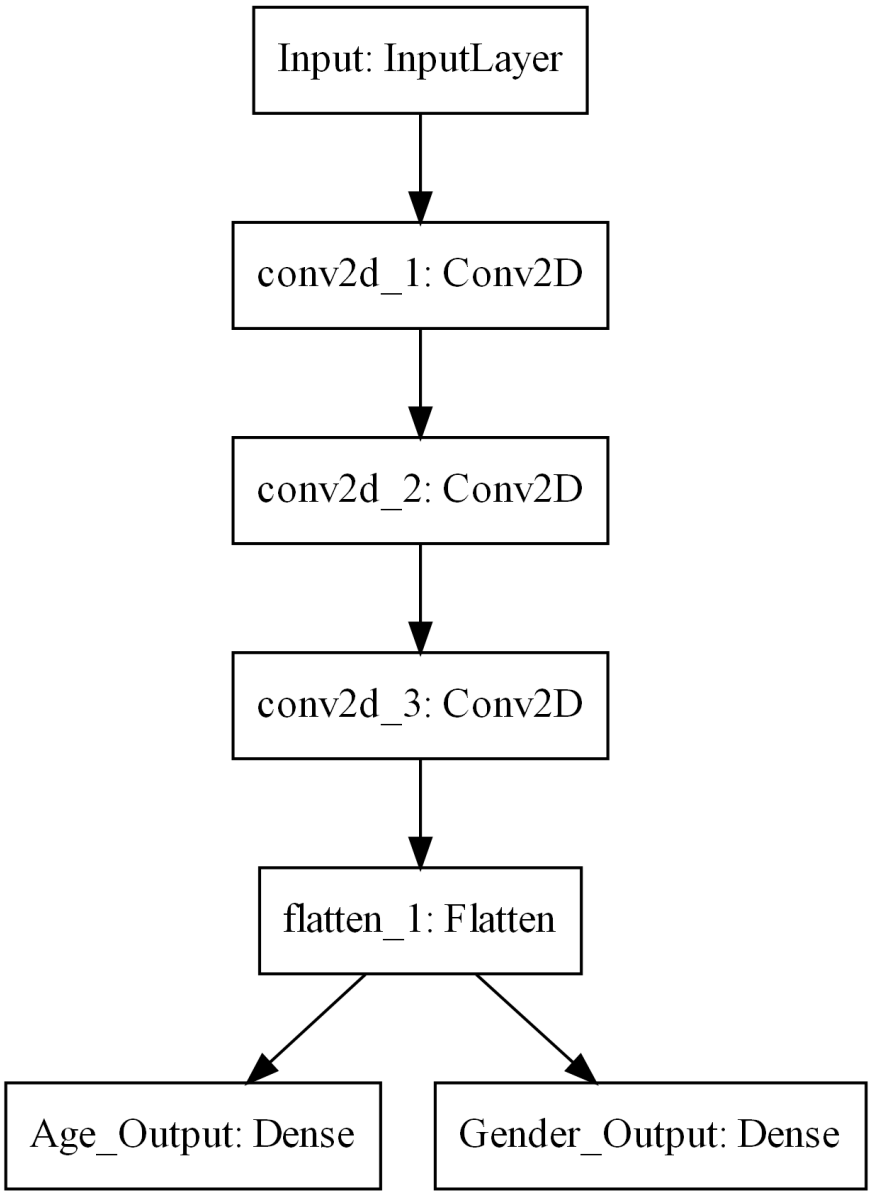

Fig. 3.4 Exemplu de model cu o singură intrare şi mai multe ieşiri

Exemplul 5. *Un model cu intrări multiple și ieșiri multiple*

Un model de predicție a stării vremii are două intrări, imaginea norilor din satelit și informații climatice și trei ieșiri pentru probabilitatea de precipitații, temperatură și umiditate.

```
image_input = layers.Input(shape=(256, 256, 3),
                           name='Image_Input')
info_input = layers.Input(shape=(10, ), name='Info_Input')
hidden1 = layers.Conv2D(filters=64, kernel_size=(3, 3),
                        activation='relu')(image_input)
hidden2 = layers.Conv2D(filters=64, kernel_size=(3, 3),
                        strides=(2, 2),
                        activation='relu')(hidden1)
hidden3 = layers.Conv2D(filters=64, kernel_size=(3, 3),
                        strides=(2, 2),
                        activation='relu')(hidden2)
flatten = layers.Flatten()(hidden3)
hidden4 = layers.Dense(units=64)(info_input)
concat = layers.concatenate([flatten, hidden4])
weather_output = layers.Dense(units=1,
                              name='Output1')(concat)
temp_output = layers.Dense(units=1,
                           name='Output2')(concat)
humidity_output = layers.Dense(units=1,
                               name='Output3')(concat)
model = models.Model(inputs=[image_input, info_input],
                     outputs=[weather_output, temp_output,
                              humidity_output])
```

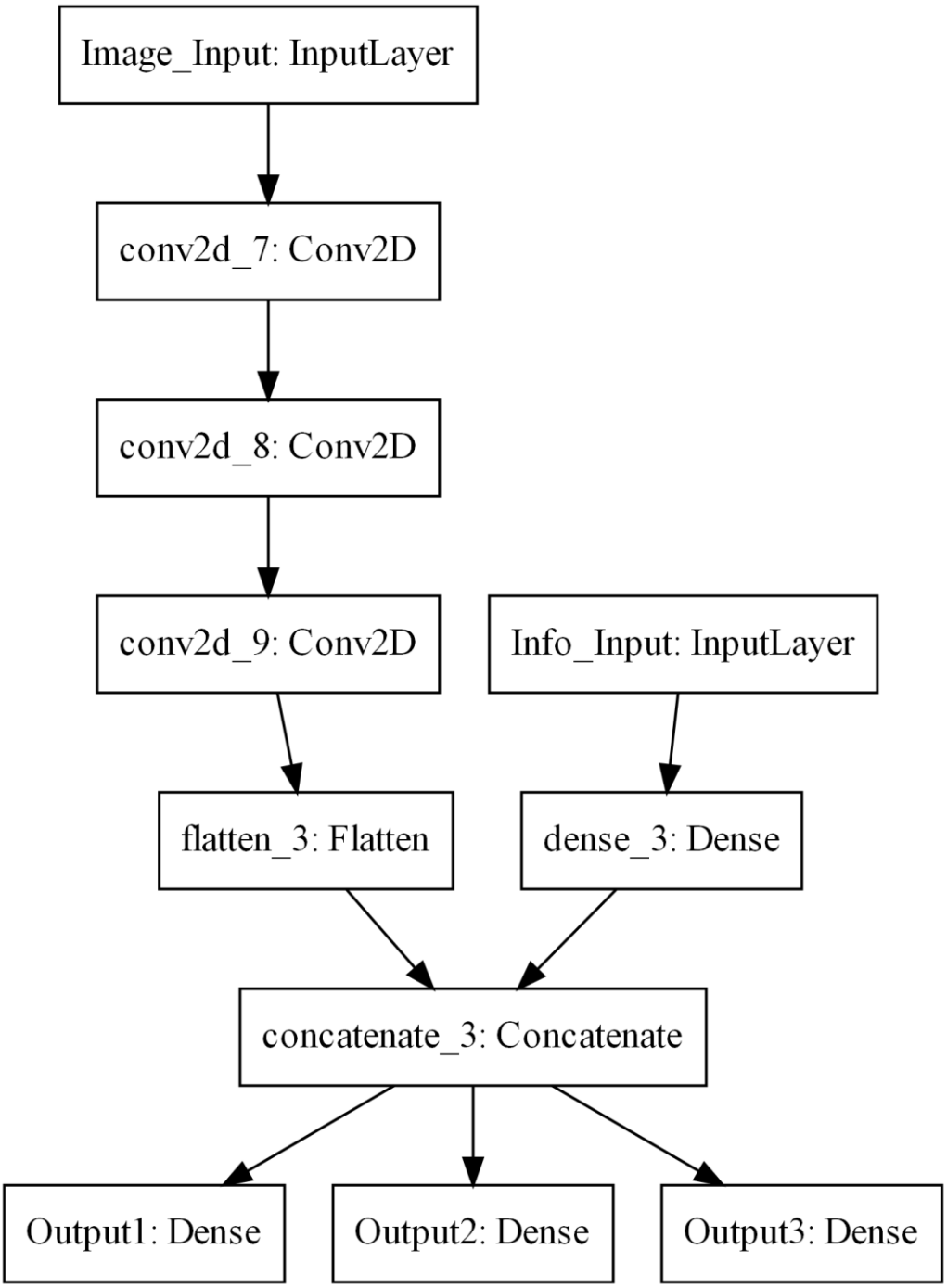

Fig. 3.5 Exemplu de model cu intrări multiple și ieșiri multiple