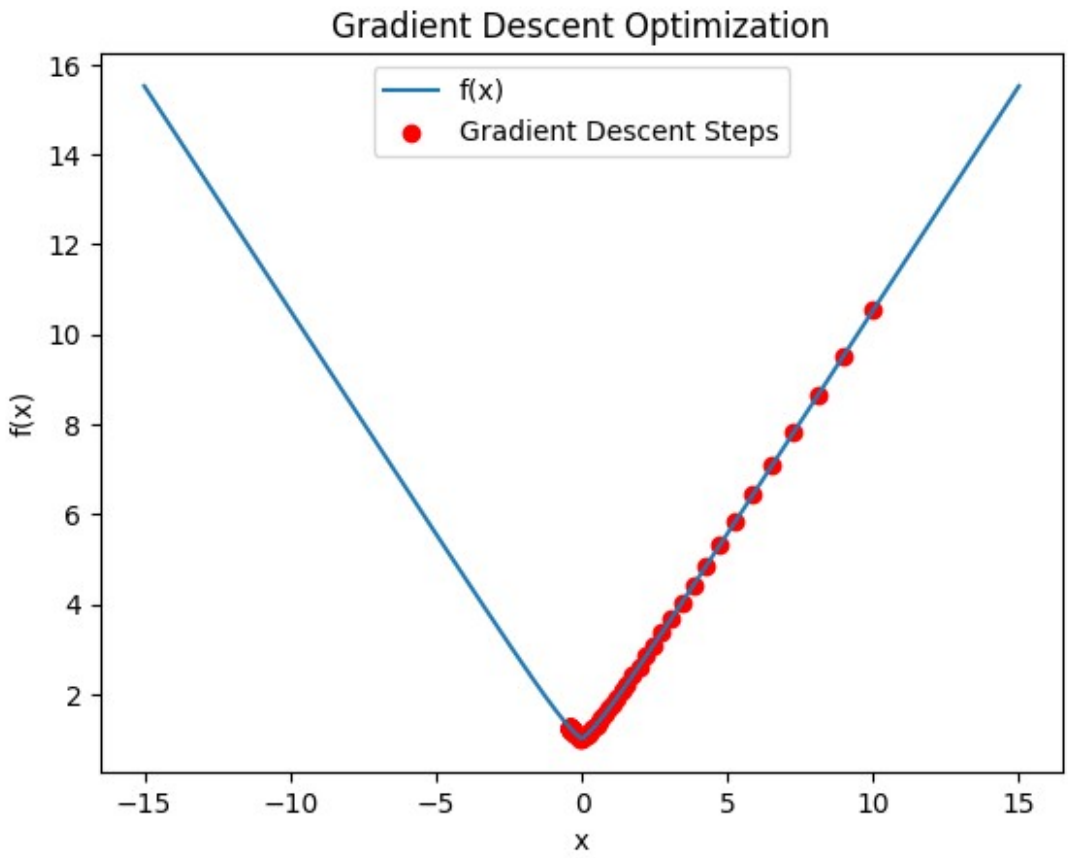
```python
import numpy as np
import matplotlib.pyplot as plt

def gradient_descent(f, grad_f, theta_init, learning_rate,
iterations):
    theta = theta_init
    history = [theta]
    for i in range(iterations):
        theta -= learning_rate * grad_f(theta)
        history.append(theta)
    return theta, history

# Definicja funkcji i jej pochodnej
f = lambda x: np.sqrt(np.abs(x) + 1 + x**2)
grad_f = lambda x: (0.5 * (2*x + 1/np.sqrt(np.abs(x) + 1 + x**2)))

# Parametry
theta_init = 10.0
learning_rate = 0.1
iterations = 100

# Optymalizacja
optimal_theta, history = gradient_descent(f, grad_f, theta_init,
learning_rate, iterations)

# Wizualizacja
x = np.linspace(-15, 15, 400)
y = f(x)

plt.plot(x, y, label='f(x)')
plt.scatter(history, f(np.array(history)), color='red',
label='Gradient Descent Steps')
plt.legend()
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title('Gradient Descent Optimization')
plt.show()

print("Optymalne theta:", optimal_theta)
```




```
Optymalne theta: -0.40020749387593874
```

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Wczytanie danych California Housing
california = fetch_california_housing()
X, y = california.data, california.target

# Przekształcenie jednej cechy (np. standaryzacja)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Podział na dane treningowe i testowe
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
test_size=0.2, random_state=42)

# Budowa modelu
model = Sequential([
```

```
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dense(64, activation='relu'),
    Dense(1)  # Warstwa wyjściowa
])

# Kompilacja
model.compile(optimizer='adam', loss='mse')

# Trening
model.fit(X_train, y_train, epochs=100, batch_size=32,
validation_split=0.2)

# Ocena modelu
loss = model.evaluate(X_test, y_test)
print("Strata na danych testowych:", loss)
```

```
Epoch 1/100

c:\Users\szymo\AppData\Local\Programs\Python\Python312\Lib\site-
packages\keras\src\layers\core\dense.py:87: UserWarning: Do not pass
an `input_shape`/`input_dim` argument to a layer. When using
Sequential models, prefer using an `Input(shape)` object as the first
layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer,
**kwargs)

413/413 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step - loss: 1.5865 - val_loss:
0.4615
Epoch 2/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.3997 - val_loss:
0.4267
Epoch 3/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.3872 - val_loss:
0.4439
Epoch 4/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.3895 - val_loss:
0.3734
Epoch 5/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.3286 - val_loss:
0.3685
Epoch 6/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.3171 - val_loss:
0.3444
Epoch 7/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.3217 - val_loss:
0.3446
Epoch 8/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.3102 - val_loss:
0.3332
Epoch 9/100
```

```
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2952 - val_loss:
0.3341
Epoch 10/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2892 - val_loss:
0.3311
Epoch 11/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2971 - val_loss:
0.3199
Epoch 12/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2931 - val_loss:
0.3424
Epoch 13/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2835 - val_loss:
0.3295
Epoch 14/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2907 - val_loss:
0.3206
Epoch 15/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2800 - val_loss:
0.3221
Epoch 16/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2788 - val_loss:
0.3030
Epoch 17/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2781 - val_loss:
0.2987
Epoch 18/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2676 - val_loss:
0.3034
Epoch 19/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2709 - val_loss:
0.3162
Epoch 20/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2863 - val_loss:
0.2918
Epoch 21/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2571 - val_loss:
0.2944
Epoch 22/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2582 - val_loss:
0.2903
Epoch 23/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2552 - val_loss:
0.3036
Epoch 24/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2574 - val_loss:
0.2900
Epoch 25/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2768 - val_loss:
```

```
0.2894
Epoch 26/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2621 - val_loss:
0.3020
Epoch 27/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2558 - val_loss:
0.2867
Epoch 28/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2529 - val_loss:
0.2850
Epoch 29/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2448 - val_loss:
0.2985
Epoch 30/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2552 - val_loss:
0.3061
Epoch 31/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2627 - val_loss:
0.2893
Epoch 32/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2510 - val_loss:
0.2893
Epoch 33/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2502 - val_loss:
0.2770
Epoch 34/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2857 - val_loss:
0.2838
Epoch 35/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2478 - val_loss:
0.2807
Epoch 36/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2485 - val_loss:
0.2860
Epoch 37/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2450 - val_loss:
0.2864
Epoch 38/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2511 - val_loss:
0.2886
Epoch 39/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2343 - val_loss:
0.2804
Epoch 40/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2503 - val_loss:
0.2803
Epoch 41/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2363 - val_loss:
0.2701
```

```
Epoch 42/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2455 - val_loss:
0.2735
Epoch 43/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2451 - val_loss:
0.2765
Epoch 44/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2458 - val_loss:
0.2738
Epoch 45/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2500 - val_loss:
0.2696
Epoch 46/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2416 - val_loss:
0.2778
Epoch 47/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2445 - val_loss:
0.2772
Epoch 48/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2368 - val_loss:
0.2712
Epoch 49/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2304 - val_loss:
0.2826
Epoch 50/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2297 - val_loss:
0.2760
Epoch 51/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2298 - val_loss:
0.2809
Epoch 52/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2279 - val_loss:
0.2718
Epoch 53/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2303 - val_loss:
0.2684
Epoch 54/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2339 - val_loss:
0.2709
Epoch 55/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2296 - val_loss:
0.2703
Epoch 56/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2336 - val_loss:
0.2711
Epoch 57/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2278 - val_loss:
0.2701
Epoch 58/100
```

```
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2318 - val_loss:
0.2756
Epoch 59/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2387 - val_loss:
0.2703
Epoch 60/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2255 - val_loss:
0.2671
Epoch 61/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2245 - val_loss:
0.2841
Epoch 62/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2301 - val_loss:
0.2801
Epoch 63/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2276 - val_loss:
0.2759
Epoch 64/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2313 - val_loss:
0.2656
Epoch 65/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2247 - val_loss:
0.2806
Epoch 66/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2279 - val_loss:
0.2665
Epoch 67/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2282 - val_loss:
0.2718
Epoch 68/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2258 - val_loss:
0.2796
Epoch 69/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2243 - val_loss:
0.2736
Epoch 70/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2250 - val_loss:
0.2772
Epoch 71/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2160 - val_loss:
0.2798
Epoch 72/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2178 - val_loss:
0.2781
Epoch 73/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2179 - val_loss:
0.2723
Epoch 74/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2163 - val_loss:
```

```
0.2720
Epoch 75/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2254 - val_loss:
0.2727
Epoch 76/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2218 - val_loss:
0.2696
Epoch 77/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2231 - val_loss:
0.2714
Epoch 78/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2167 - val_loss:
0.2801
Epoch 79/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2190 - val_loss:
0.2706
Epoch 80/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2145 - val_loss:
0.2707
Epoch 81/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2161 - val_loss:
0.2822
Epoch 82/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2201 - val_loss:
0.2750
Epoch 83/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2194 - val_loss:
0.2781
Epoch 84/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2217 - val_loss:
0.2775
Epoch 85/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2171 - val_loss:
0.2690
Epoch 86/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2160 - val_loss:
0.2722
Epoch 87/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step - loss: 0.2188 - val_loss:
0.2709
Epoch 88/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2313 - val_loss:
0.2691
Epoch 89/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2169 - val_loss:
0.2688
Epoch 90/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2159 - val_loss:
0.2712
```

```
Epoch 91/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2147 - val_loss:
0.2708
Epoch 92/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2196 - val_loss:
0.2709
Epoch 93/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2118 - val_loss:
0.2791
Epoch 94/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2157 - val_loss:
0.2715
Epoch 95/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2148 - val_loss:
0.2727
Epoch 96/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2046 - val_loss:
0.2688
Epoch 97/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2165 - val_loss:
0.2675
Epoch 98/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2206 - val_loss:
0.2660
Epoch 99/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2133 - val_loss:
0.2701
Epoch 100/100
413/413 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.2081 - val_loss:
0.2691
129/129 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step - loss: 0.2642
Strata na danych testowych: 0.272561252117157
```

```python
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Flatten, Dense,
MaxPooling2D
from tensorflow.keras.datasets import cifar100
from tensorflow.keras.utils import to_categorical

# Wczytanie danych CIFAR-100
(X_train, y_train), (X_test, y_test) = cifar100.load_data()
y_train = to_categorical(y_train, 100)
y_test = to_categorical(y_test, 100)

# Budowa modelu
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
```

```
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(512, activation='relu'),
    Dense(100, activation='softmax')
])

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

# Trening
model.fit(X_train, y_train, epochs=20, batch_size=64,
validation_split=0.2)

# Ocena modelu
loss, accuracy = model.evaluate(X_test, y_test)
print("Strata na danych testowych:", loss)
print("Dokładność na danych testowych:", accuracy)
```

```
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-100-
python.tar.gz
169001437/169001437 ━━━━━━━━━━━━━━━━━━━━ 9s 0us/step

c:\Users\szymo\AppData\Local\Programs\Python\Python312\Lib\site-
packages\keras\src\layers\convolutional\base_conv.py:107: UserWarning:
Do not pass an `input_shape`/`input_dim` argument to a layer. When
using Sequential models, prefer using an `Input(shape)` object as the
first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer,
**kwargs)

Epoch 1/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 19s 29ms/step - accuracy: 0.0200 - loss:
8.9081 - val_accuracy: 0.0584 - val_loss: 4.2730
Epoch 2/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 18s 28ms/step - accuracy: 0.0795 - loss:
4.1209 - val_accuracy: 0.1158 - val_loss: 3.9113
Epoch 3/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 23s 32ms/step - accuracy: 0.1401 - loss:
3.7364 - val_accuracy: 0.1450 - val_loss: 3.8003
Epoch 4/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 20s 31ms/step - accuracy: 0.2176 - loss:
3.3101 - val_accuracy: 0.1634 - val_loss: 3.6641
Epoch 5/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 18s 29ms/step - accuracy: 0.3041 - loss:
2.8781 - val_accuracy: 0.1950 - val_loss: 3.5932
Epoch 6/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 18s 30ms/step - accuracy: 0.4101 - loss:
2.3665 - val_accuracy: 0.2132 - val_loss: 3.8154
Epoch 7/20
```

```
625/625 ━━━━━━━━━━━━━━━━━━━━ 18s 29ms/step - accuracy: 0.5191 - loss:
1.8705 - val_accuracy: 0.2120 - val_loss: 4.1251
Epoch 8/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 19s 30ms/step - accuracy: 0.6266 - loss:
1.4187 - val_accuracy: 0.2097 - val_loss: 4.8582
Epoch 9/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 18s 29ms/step - accuracy: 0.7117 - loss:
1.0648 - val_accuracy: 0.2094 - val_loss: 5.5740
Epoch 10/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 18s 29ms/step - accuracy: 0.7688 - loss:
0.8525 - val_accuracy: 0.2035 - val_loss: 6.2577
Epoch 11/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 18s 29ms/step - accuracy: 0.8066 - loss:
0.6924 - val_accuracy: 0.2073 - val_loss: 6.8779
Epoch 12/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 18s 29ms/step - accuracy: 0.8365 - loss:
0.5657 - val_accuracy: 0.2062 - val_loss: 7.6468
Epoch 13/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 20s 31ms/step - accuracy: 0.8569 - loss:
0.4997 - val_accuracy: 0.1989 - val_loss: 8.8903
Epoch 14/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 20s 32ms/step - accuracy: 0.8644 - loss:
0.4734 - val_accuracy: 0.1909 - val_loss: 9.0912
Epoch 15/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 19s 31ms/step - accuracy: 0.8759 - loss:
0.4383 - val_accuracy: 0.2057 - val_loss: 9.6393
Epoch 16/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 20s 31ms/step - accuracy: 0.8815 - loss:
0.4167 - val_accuracy: 0.1972 - val_loss: 11.2183
Epoch 17/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 19s 31ms/step - accuracy: 0.8903 - loss:
0.3936 - val_accuracy: 0.1978 - val_loss: 10.8628
Epoch 18/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 19s 31ms/step - accuracy: 0.8989 - loss:
0.3690 - val_accuracy: 0.1898 - val_loss: 11.7778
Epoch 19/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 19s 30ms/step - accuracy: 0.9031 - loss:
0.3511 - val_accuracy: 0.2037 - val_loss: 11.7037
Epoch 20/20
625/625 ━━━━━━━━━━━━━━━━━━━━ 19s 30ms/step - accuracy: 0.9138 - loss:
0.3195 - val_accuracy: 0.1954 - val_loss: 13.3479
313/313 ━━━━━━━━━━━━━━━━━━━━ 2s 6ms/step - accuracy: 0.1965 - loss:
13.0540
Strata na danych testowych: 13.029065132141113
Dokładność na danych testowych: 0.20010000467300415
```