

Sieci neuronowe i sztuczna inteligencja – laboratorium 8

Monika Błyszcz, 236623

Kod:

```
1 # import os utilities
2 import os
3
4 # import numpy
5 import numpy as np
6
7 # images
8 import skimage
9 from skimage import transform
10 from skimage.color import rgb2gray
11
12 # Import the `pyplot` module
13 import matplotlib.pyplot as plt
14
15
16 # function to load data
17 def load_data(data_directory):
18     """Loads sign images data from their folder.
19
20     Returns:
21         images: list of images, i.e., signs
22         labels: list of labels, i.e., signs IDs
23     """
24     # We need back labels and the row images
25     images = []
26     labels = []
27
28     # We have one folder per sign type
29     directories = []
30     for d in os.listdir(data_directory):
31         if os.path.isdir(os.path.join(data_directory, d)):
32             directories.append(d)
33
34     # In each folder there are not only images but also csv
description
35     # files
36     for d in directories:
37         label_directory = os.path.join(data_directory, d)
38         file_names = [
39             os.path.join(label_directory, f)
40             for f in os.listdir(label_directory)
41             if f.endswith(".ppm")
42         ]
43
44         for f in file_names:
45             images.append(skimage.io.imread(f))
```

```

46         labels.append(int(d))
47
48     return images, labels
49
50
51 ROOT_PATH = os.getcwd()
52
53 # Download training data
54
55 train_data_directory = os.path.join(ROOT_PATH, "Training")
56 test_data_directory = os.path.join(ROOT_PATH, "Testing")
57
58 images, labels = load_data(train_data_directory)
59 test_images, test_labels = load_data(test_data_directory)
60
61 print(labels)
62
63 ## The following commented lines were reported in the
DataCamp materials
64 ## but they does not work here
65 print(images.ndim)
66 print(images.size)
67 images[0]
68 print(len(images))
69 print(len(labels))
70
71 # this should be a bar plot but an histogram with the same
number of
72 # bins that that unique levels of the labels list should
be fine :-)
73 unique_labels = set(labels)
74 n_labels = max(unique_labels) + 1
75
76 # Make a histogram with 62 bins of the `labels` data
77 plt.hist(labels, n_labels)
78
79 # Show the plot
80 plt.show()
81
82 # Determine the (random) indexes of the images that you
want to see
83 traffic_signs = [300, 2250, 3650, 4000]
84
85 # Fill out the subplots with the random images that you
defined
86 for i in range(len(traffic_signs)):

```

```

87     plt.subplot(1, 4, i + 1)
88     plt.axis('off')
89     plt.imshow(images[traffic_signs[i]])
90     plt.subplots_adjust(wspace=0.5)
91     plt.show()
92     print(
93         "shape: {0}, min value: {1}, max value: {2}".
format(
94         images[traffic_signs[i]].shape,
95         images[traffic_signs[i]].min(),
96         images[traffic_signs[i]].max()
97     )
98 )
99
100 # Plot a grid with a sample of all the signs
101 plt.figure(figsize=(15, 15))
102
103 i = 1
104
105 for label in unique_labels:
106     # pick the first image for the label.
107     #
108     # The index() method searches an element in the list
and returns its
109     # index. In simple terms, index() method finds the
given element in
110     # a list and returns its position. However, if the
same element is
111     # present more than once, index() method returns its
smallest/first
112     # position.
113     image = images[labels.index(label)]
114
115     # We have 62 images. Hence, define a 64 grid sub-
plots
116     plt.subplot(8, 8, i)
117
118     # Don't include axes
119     plt.axis('off')
120
121     # Add a title to each subplot
122     #
123     # The count() method returns the number of elements
with the
124     # specified value.
125     plt.title("Label {0} ({1})".format(label, labels.

```

```
125 count(label)))
126
127     # Add 1 to the counter
128     i += 1
129
130     # Plot this first image
131     plt.imshow(image)
132
133 plt.show()
134
135 # To tackle the differing image sizes, you're going to
    rescale the images
136 images_28 = [
137     transform.resize(image, (28, 28))
138     for image in images
139 ]
140
141 # Convert `images28` to an array
142 images_28 = np.array(images_28)
143
144 # Convert `images28` to grayscale
145 images_28 = rgb2gray(images_28)
146
147 for i in range(len(traffic_signs)):
148     plt.subplot(1, 4, i + 1)
149     plt.axis('off')
150     plt.imshow(images_28[traffic_signs[i]], cmap="gray")
151     plt.subplots_adjust(wspace=0.5)
152
153 plt.show()
154
155 # Test set
156 # Transform the images to 28 by 28 pixels
157 test_images_28 = [
158     transform.resize(image, (28, 28))
159     for image in test_images
160 ]
161 # Convert to grayscale
162 test_images_28 = rgb2gray(np.array(test_images_28))
163
```

W trakcie wykonywania zadania, w pierwszym kroku wczytano dane testowe i dane treningowe. Uzyskano dzięki temu zbiór obrazów i ich indeksów (etykiety). Następnie utworzono histogram obrazujący ilość znaków danej kategorii. Potem wyszukano 4 losowe kategorie obrazów. Po czym wykorzystując cały zbiór pokazano po 1 przykładzie z każdej kategorii. Na końcu 4 obrazy z klas wybranych wcześniej ukazano w odcieniach szarości

[illegible]

A bar chart showing the frequency of the number of children per family. The x-axis is labeled 'Number of children' and ranges from 0 to 60 with major ticks every 10 units. The y-axis is labeled 'Frequency' and ranges from 0 to 350 with major ticks every 50 units. The bars represent the count of families for each number of children. The distribution is unimodal and slightly right-skewed, with a peak at 22 children (frequency ~375). There are several smaller peaks, notably at 33 children (frequency ~315) and 38 children (frequency ~285).

Number of children	Frequency
0	15
1	110
2	15
3	15
4	15
5	15
6	15
7	15
8	15
9	15
10	15
11	15
12	15
13	15
14	15
15	15
16	15
17	15
18	15
19	15
20	15
21	15
22	375
23	15
24	15
25	15
26	15
27	15
28	15
29	15
30	15
31	15
32	15
33	315
34	15
35	15
36	15
37	15
38	285
39	15
40	15
41	15
42	15
43	15
44	15
45	15
46	15
47	15
48	15
49	15
50	15
51	15
52	15
53	15
54	15
55	15
56	15
57	15
58	15
59	15
60	15

Label 0 (15) Label 1 (110) Label 2 (13) Label 3 (15) Label 4 (15) Label 5 (11) Label 6 (18) Label 7 (157)

Label 8 (27) Label 9 (18) Label 10 (21) Label 11 (7) Label 12 (18) Label 13 (90) Label 14 (43) Label 15 (9)

Label 16 (9) Label 17 (79) Label 18 (81) Label 19 (231) Label 20 (42) Label 21 (43) Label 22 (375) Label 23 (15)

Label 24 (48) Label 25 (42) Label 26 (6) Label 27 (18) Label 28 (125) Label 29 (33) Label 30 (37) Label 31 (63)

Label 32 (316) Label 33 (12) Label 34 (46) Label 35 (60) Label 36 (18) Label 37 (98) Label 38 (285) Label 39 (196)

Label 40 (242) Label 41 (148) Label 42 (35) Label 43 (30) Label 44 (48) Label 45 (74) Label 46 (44) Label 47 (147)

Label 48 (11) Label 49 (12) Label 50 (15) Label 51 (27) Label 52 (27) Label 53 (199) Label 54 (118) Label 55 (12)

Label 56 (95) Label 57 (78) Label 58 (15) Label 59 (42) Label 60 (9) Label 61 (282)

Skala szarości:

