

AI4M_C1_W3_lecture_ex_02

May 4, 2020

1 AI4M Course 1 week 3 lecture notebook

2 Extract a sub-section

In the assignment you will be extracting sub-sections of the MRI data to train your network. The reason for this is that training on a full MRI scan would be too memory intensive to be practical. To extract a sub-section in the assignment, you will need to write a function to isolate a small “cube” of the data for training. This example is meant to show you how to do such an extraction for 1D arrays. In the assignment you will apply the same logic in 3D.

```
In [1]: import numpy as np
```

```
In [2]: # Define a simple one dimensional "image" to extract from
        image = np.array([10,11,12,13,14,15])
        image
```

```
Out[2]: array([10, 11, 12, 13, 14, 15])
```

```
In [3]: # Compute the dimensions of your "image"
        image_length = image.shape[0]
        image_length
```

```
Out[3]: 6
```

2.0.1 Sub-sections

In the assignment, you will define a “patch size” in three dimensions, that will be the size of the sub-section you want to extract. For this exercise, you only need to define a patch size in one dimension.

```
In [4]: # Define a patch length, which will be the size of your extracted sub-section
        patch_length = 3
```

To extract a patch of length `patch_length` you will first define an index at which to start the patch.

Run the next cell to define your start index

```
In [5]: # Define your start index
        start_i = 0
```

At the end of the next cell you are adding 1 to the start index. Run cell a few times to extract some one dimensional sub-sections from your "image"

What happens when you run into the edge of the image (when start_index is > 3)?

```
In [7]: # Define an end index given your start index and patch size
print(f"start index {start_i}")
end_i = start_i + patch_length
print(f"end index {end_i}")
# Extract a sub-section from your "image"
sub_section = image[start_i: end_i]
print(sub_section)
# Add one to your start index
start_i +=1
```

```
start index 1
end index 4
[11 12 13]
```

You'll notice when you run the above multiple times, that eventually the sub-section returned is no longer of length patch_length.

In the assignment, your neural network will be expecting a particular sub-section size and will not accept inputs of other dimensions. For the start indices, you will be randomly choosing values and you need to ensure that your random number generator is set up to avoid the edges of your image object.

The next few code cells include a demonstration of how you could determine the constraints on your start index for the simple one dimensional example.

```
In [8]: # Set your start index to 3 to extract a valid patch
start_i = 3
print(f"start index {start_i}")
end_i = start_i + patch_length
print(f"end index {end_i}")
sub_section = image[start_i: end_i]
print(sub_section)
```

```
start index 3
end index 6
[13 14 15]
```

```
In [9]: # Compute and print the largest valid value for start index
print(f"The largest start index for which "
      f"a sub section is still valid is "
      f"{image_length - patch_length}")
```

The largest start index for which a sub section is still valid is 3

```
In [10]: # Compute and print the range of valid start indices
print(f"The range of valid start indices is:")
# Compute valid start indices, note the range() function excludes the upper bound
valid_start_i = [i for i in range(image_length - patch_length + 1)]
print(valid_start_i)
```

The range of valid start indices is:
[0, 1, 2, 3]

2.0.2 Random selection of start indices

In the assignment, you will need to randomly select a valid integer for the start index in each of three dimensions. The way to do this is by following the logic above to identify valid start indices and then selecting randomly from that range of valid numbers.

Run the next cell to select a valid start index for the one dimensional example

```
In [11]: # Choose a random start index, note the np.random.randint() function excludes the upper bound
start_i = np.random.randint(image_length - patch_length + 1)
print(f"randomly selected start index {start_i}")
```

randomly selected start index 2

```
In [12]: # Randomly select multiple start indices in a loop
for _ in range(10):
    start_i = np.random.randint(image_length - patch_length + 1)
    print(f"randomly selected start index {start_i}")
```

randomly selected start index 2
randomly selected start index 2
randomly selected start index 2
randomly selected start index 2
randomly selected start index 1
randomly selected start index 2
randomly selected start index 2
randomly selected start index 1
randomly selected start index 2
randomly selected start index 0

That's all for this lab, now you have the basic tools you need for sub-section extraction in this week's graded assignment!