## PyTorch tensor advanced indexing

Asked 2 years, 6 months ago Modified 2 years, 6 months ago Viewed 11k times



Say i have one matrix and one vector as follows:

```
11
```

```
x = torch.tensor([[1, 2, 3],
                  [4, 5, 6],
                  [7, 8, 9]])
```



```
y = torch.tensor([0, 2, 1])
```



is there a way to slice it x[y] so the result is:

```
res = [1, 6, 8]
```

so basically i take the first element of y and take the element in x that corresponds to the first row and the elements' column.

## Cheers

```
numpy pytorch Edit tags
python
```

Share Edit Follow Close Flag

edited Apr 8, 2020 at 9:06

asked Apr 8, 2020 at 8:33



yatu **82.8k** 12 72 119



- 1 Added brackets in definition of x, not sure if it was just a typo here or also in your code.
  - FBruzzesi Apr 8, 2020 at 8:58
- 1 it was a typo, thanks Dr. Prof. Patrick Apr 10, 2020 at 10:16

## 2 Answers

Sorted by: Reset to default

Date modified (newest first)





Advanced indexing in pytorch works just as <code>NumPy's</code>, i.e the indexing arrays are broadcast together across the axes. So you could do as in FBruzzesi's answer.



Though similarly to np.take\_along\_axis, in pytorch you also have <u>torch.gather</u>, to take values along a specific axis:



x.gather(1, y.view(-1,1)).view(-1)# tensor([1, 6, 8])



yatu

**82.8k** 12 72 119



You can specify the corresponding row index as:



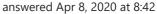


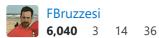






Share Edit Follow Flag







- 3 Im able to reproduce this answer withough any problem. Please revise your question @yarin yatu Apr 8, 2020 at 8:54
- 2 sorry i accidentally mistyped the parenthesis, now it works Dr. Prof. Patrick Apr 8, 2020 at 9:00 /
- 1 Good! it's simpler than torch.gather(). tom\_cat Apr 6 at 8:03