



The Global PyTorch Summer Hackathon is here!

Calling developers and researchers to hack together packages, demos and applications on top of PyTorch and submit online.

We'll help you reach the community, amplify your vision and have \$60K in prizes!

Read more at <https://pytorch.devpost.com>

## How does transforms.ToTensor() work and computation of mean and std values

varghese\_alex

Oct '17

Oct 2017

Hi

1 / 5

Oct 2017

1. I am currently using the transforms.ToTensor(). As per the document it converts data in the range 0-255 to 0-1.  
However, the transform work on data whose values ranges between negative to positive values?  
Any ideas how this transform work. And the transformed values no longer strictly positive.
2. In most tutorials regarding the finetuning using pretrained models, the data is normalized with [0.485, 0.456, 0.406], [0.229, 0.224, 0.225]). I would like to know how these values are computed? Are they computed by attained by using mean & std value of each channel from the entire training data?

WERush Xinge

Oct '17

Aug 18

ToTensor() works for the image, whose elements are in range 0 to 255. You can write your custom Transforms to suit your needs.

[0.485, 0.456, 0.406] is the normalized mean value of ImageNet, and [0.229, 0.224, 0.225] denotes the std of ImageNet.

Yes, it is computed per channels.

varghese\_alex

Oct '17

Oki I got it.

[http://pytorch.org/docs/0.2.0/\\_modules/torchvision/transforms.html#ToTensor](http://pytorch.org/docs/0.2.0/_modules/torchvision/transforms.html#ToTensor) explains. The entire array is converted to torch tensor and then divided by 255. This is how it is forces the network to be between 0 and 1.

---

LvJC

Dec '18

Actually, I found it is different from  $*1.0/255$  in C++. The results between `transforms.ToTensor()` in Python and  $*1.0/255$  in C++ are not same.

I am wondering why...

---

neonb88 Nathan Bendich

Aug 18

My gut instinct was  $* 1.0 / 256$  instead of  $* 1.0 / 255$ . Does this fix the problem, @LvJC ?