
farabio

Release 0.0.1

San Askaruly

Aug 18, 2021

CONTENTS

1	Overview	3
1.1	Getting started	3
1.2	Reference	3
1.3	Handbook	9
2	Indices and tables	23
	Bibliography	25
	Index	27

farabio is a minimal PyTorch toolkit for out-of-the-box deep learning support in biomedical imaging. out-of-the-box deep learning support in biomedical imaging.

Features:

- Biomedical datasets
- Common DL models
- Flexible trainers (in progress)

OVERVIEW

1.1 Getting started

1.1.1 How to install

1.1.2 1. Activate conda environment

```
$ conda create -n myenv python=3.8  
$ conda activate myenv
```

1.1.3 2. Install farabio

A. With pip:

```
$ pip install farabio -f https://download.pytorch.org/whl/torch_stable.html
```

B. Setup from source:

```
$ git clone https://github.com/tuttelikz/farabio.git && cd farabio  
$ python -m pip install --upgrade pip setuptools wheel  
$ pip install . -f https://download.pytorch.org/whl/torch_stable.html
```

1.2 Reference

1.2.1 data Package

biodatasets Module

biodatasets module provides classes to load public biomedical datasets in a PyTorch friendly manner.

ChestXrayDataset class

```
class farabio.data.biodatasets.ChestXrayDataset(root: str = '.', download: bool = False, mode: str =  
    'train', shape: int = 256, transform:  
    Optional[Callable] = None, target_transform:  
    Optional[Callable] = None, show: bool = True)
```

PyTorch friendly ChestXrayDataset class

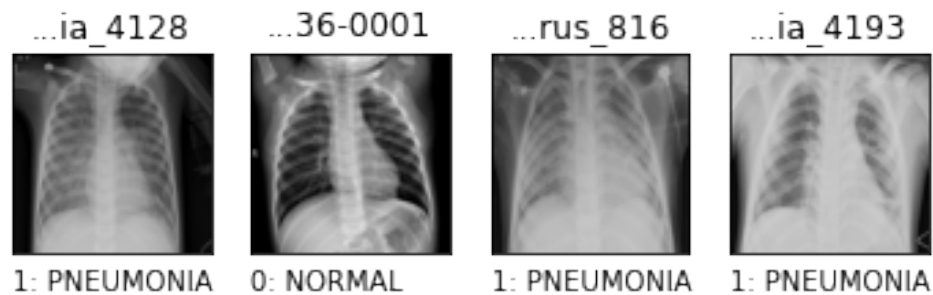
Dataset is loaded using Kaggle API. For further information on raw dataset and pneumonia detection, please refer to [1].

References

[1]

Examples

```
>>> valid_dataset = ChestXrayDataset(root=_path, download=True, mode="val",  
    ↪ show=True)
```



```
visualize_batch()
```


DSB18Dataset class

```
class farabio.data.biodatasets.DSB18Dataset(root: str = '.', download: bool = False, mode: str = 'train',  
                                             shape: int = 512, transform: Optional[Callable] = None,  
                                             target_transform: Optional[Callable] = None, show: bool  
                                             = True)
```

PyTorch friendly DSB18Dataset class

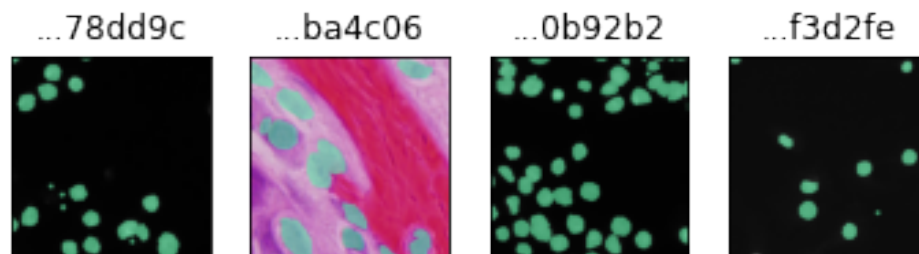
Dataset is loaded using Kaggle API. For further information on raw dataset and nuclei segmentation, please refer to [1].

References

[1]

Examples

```
>>> train_dataset = DSB18Dataset(_path, transform=None, download=False, show=True)
```



`visualize_batch()`

HistocancerDataset class

```
class farabio.data.biodatasets.HistocancerDataset(root: str = '.', mode: str = 'train', transform:
Optional[Callable] = None, target_transform:
Optional[Callable] = None, download: bool =
False, show: bool = True)
```

PyTorch friendly HistocancerDataset class

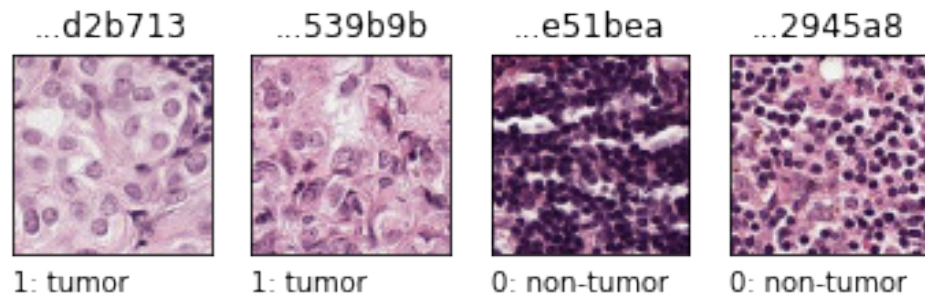
Dataset is loaded using Kaggle API. For further information on raw dataset and tumor classification, please refer to [1].

References

[1]

Examples

```
>>> train_dataset = HistocancerDataset(root=".", download=False, mode="train")
```



`visualize_batch()`

RANZCRDataset class

```
class farabio.data.biodatasets.RANZCRDataset(root: str = '.', mode: str = 'train', shape: int = 256,
                                             transform: Optional[Callable] = None, target_transform:
                                             Optional[Callable] = None, download: bool = False,
                                             show: bool = True)
```

PyTorch friendly RANZCRDataset class

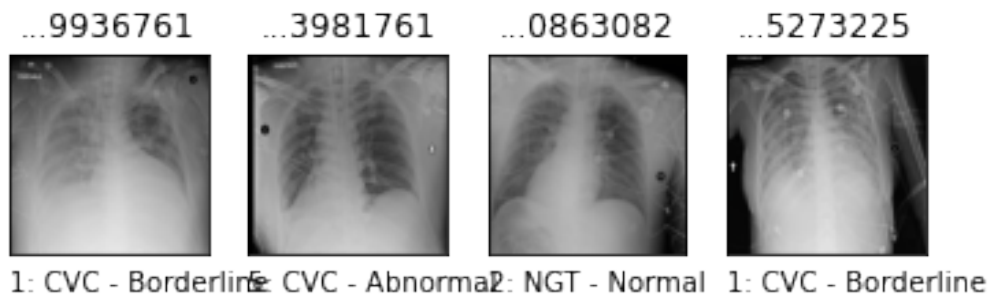
Dataset is loaded using Kaggle API. For further information on raw dataset and catheters presence, please refer to [1].

References

[1]

Examples

```
>>> train_dataset = RANZCRDataset(_path_ranzcr, show=True, shape=512)
```



visualize_batch()

RetinopathyDataset class

```
class farabio.data.biodatasets.RetinopathyDataset(root: str = '.', mode: str = 'train', shape: int = 256,  
transform: Optional[Callable] = None,  
target_transform: Optional[Callable] = None,  
download: bool = False, show: bool = True)
```

PyTorch friendly RetinopathyDataset class

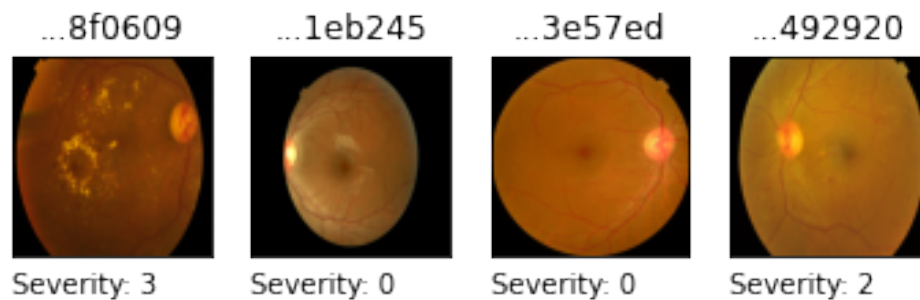
Dataset is loaded using Kaggle API. For further information on raw dataset and blindness detection, please refer to [1].

References

[1]

Examples

```
>>> train_dataset = RetinopathyDataset(".", mode="train", show=True)
```



`visualize_batch()`

1.3 Handbook

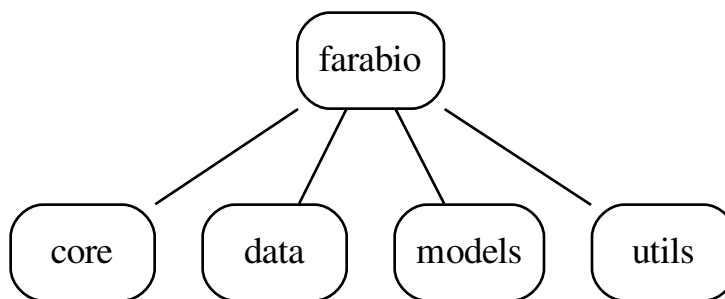
1.3.1 Overview

Deep learning has transformed many aspects of industrial pipelines recently. Scientists involved in biomedical imaging research are also benefiting from the power of AI to tackle complex challenges. Although academic community has widely accepted image processing tools, such as scikit-image, ImageJ, there is still a need for a tool which integrates deep learning into biomedical image analysis. We propose a minimal, but convenient Python package based on PyTorch with biomedical datasets, common deep learning models, and extended by flexible trainers.

What can I do with this package?

- Load public biomedical datasets
- Load common deep learning models
- Do basic image preprocessing and transformations
- Customize training loops to your own needs

Package structure



How to contribute?

You can contribute to this package by reporting issues and/or by sending pull request.

If you find a bug, please report it by opening an [issue on Git](#).

1.3.2 Machine learning for Biomedical Engineers

Table of Contents

- *Activation functions*
 - *Binary Step*
 - *Piecewise Linear*
 - *Bipolar*
 - *Sigmoid*
 - *Bipolar Sigmoid*
 - *Hyperbolic Tangent, TanH*
 - *Arctangent, ArcTan*
 - *Rectified Linear Units, ReLU*
 - *Leaky Rectified Linear Units, Leaky ReLU*
 - *Exponential Linear Units, ELU*
 - *SoftPlus*

Activation functions

Binary Step

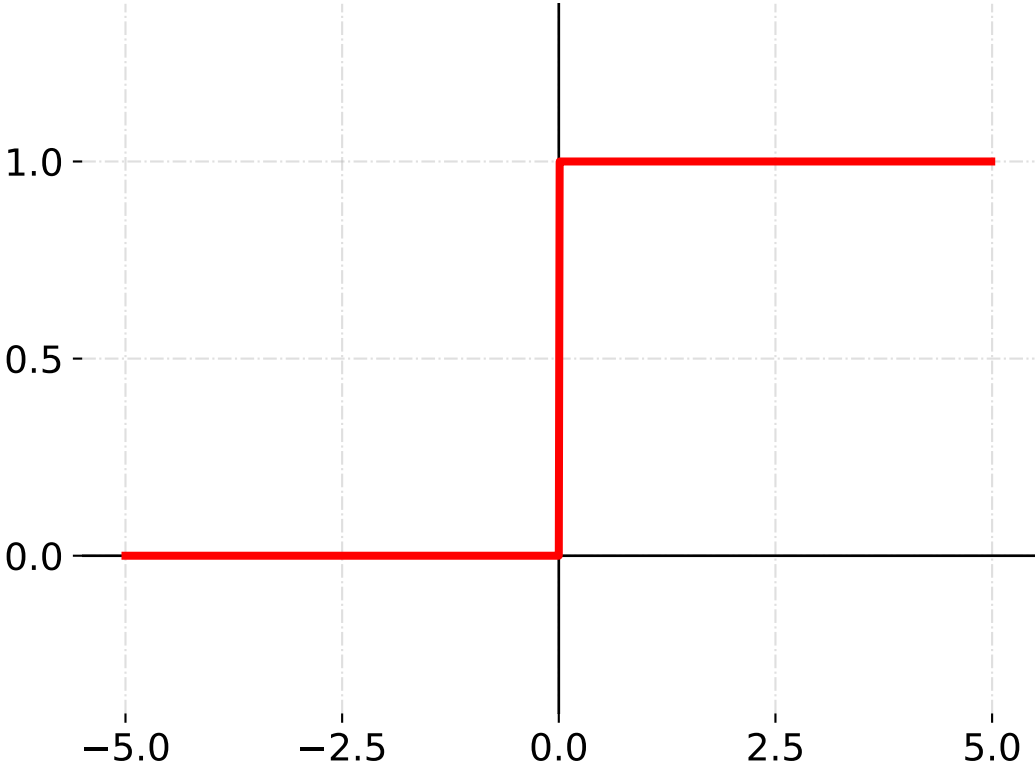
$$f(x) = \begin{cases} 0 & \text{for } x < x_{min} \\ mx + b & \text{for } x_{min} \leq x \leq x_{max} \\ 1 & \text{for } x > x_{max} \end{cases}$$

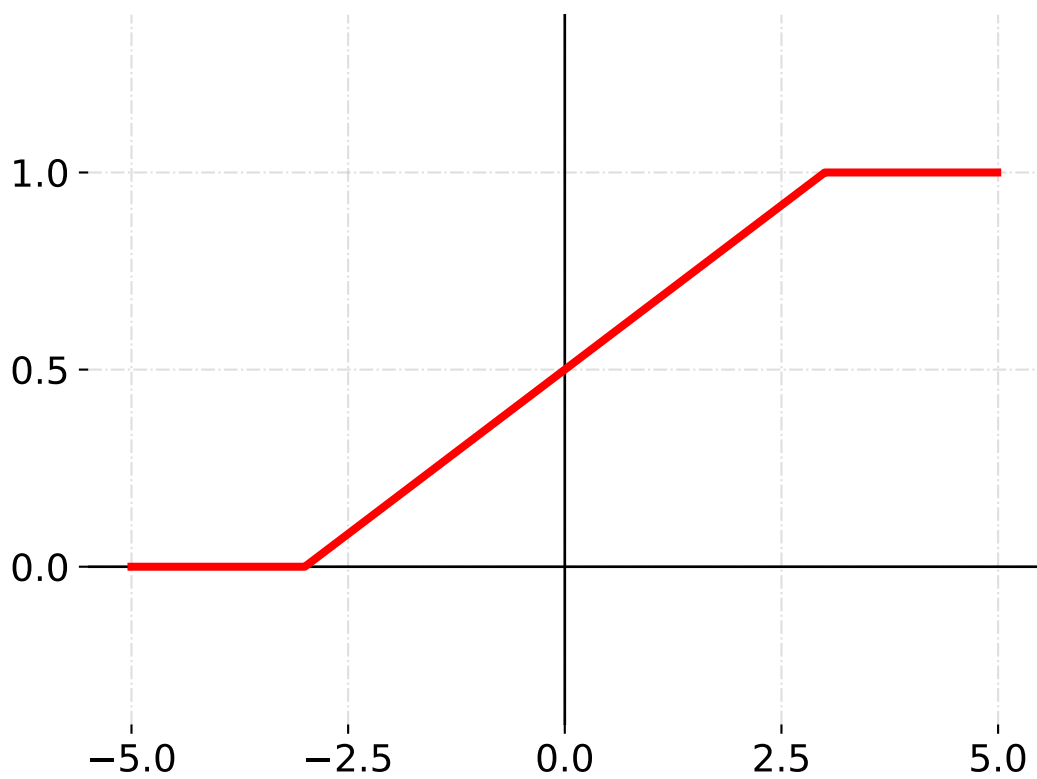
$$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$$

Piecewise Linear

$$f(x) = \begin{cases} 0 & \text{for } x < x_{min} \\ mx + b & \text{for } x_{min} \leq x \leq x_{max} \\ 1 & \text{for } x > x_{max} \end{cases}$$

$$f'(x) = \begin{cases} 0 & \text{for } x < x_{min} \\ m & \text{for } x_{min} \leq x \leq x_{max} \\ 0 & \text{for } x > x_{max} \end{cases}$$

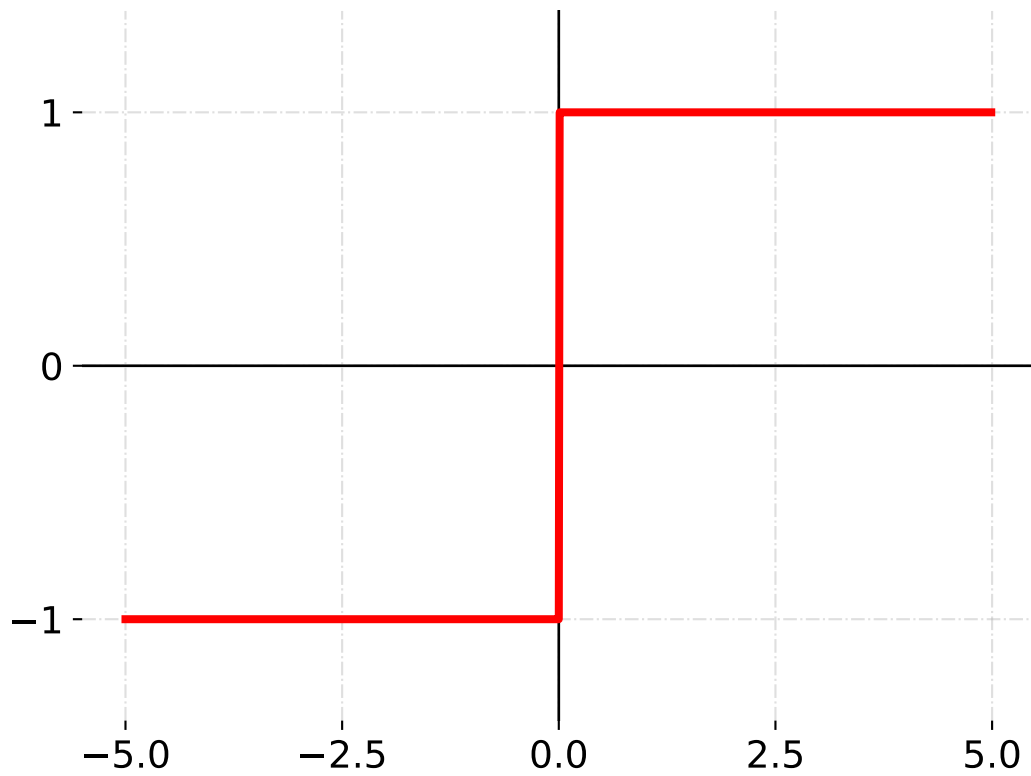




Bipolar

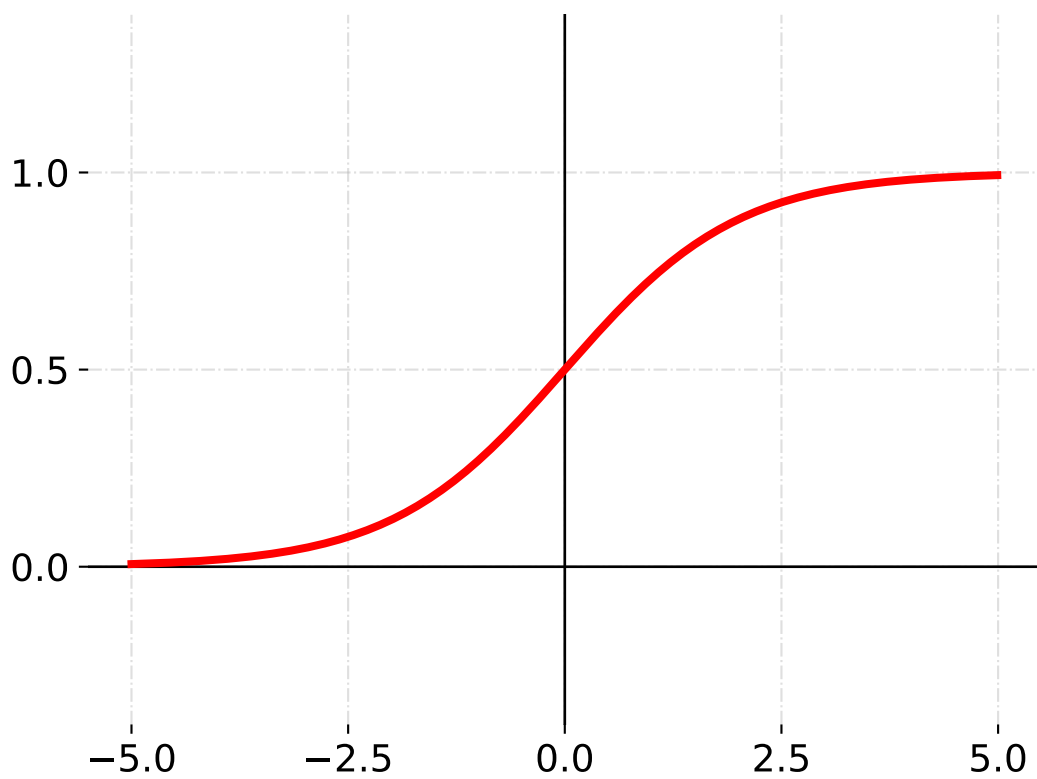
$$f(x) = \begin{cases} -1 & \text{for } x \leq 0 \\ 1 & \text{for } x > 0 \end{cases}$$

$$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$$

**Sigmoid**

$$f(x) = \frac{1}{1 + e^{-x}}$$

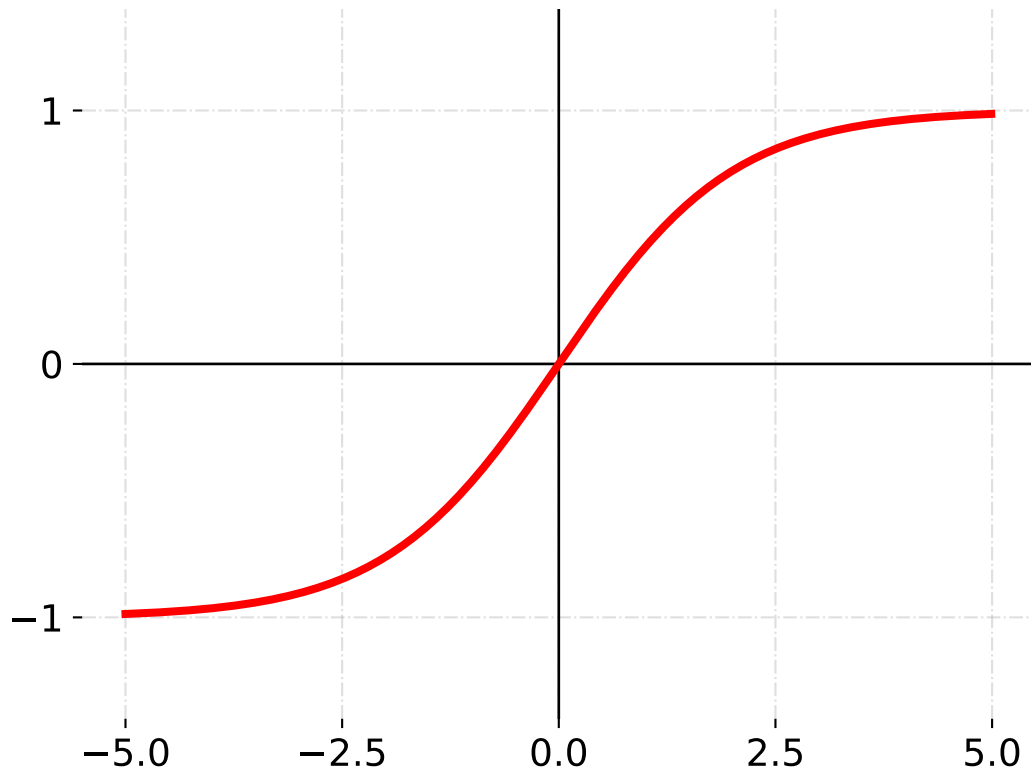
$$f'(x) = f(x)(1 - f(x))$$



Bipolar Sigmoid

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

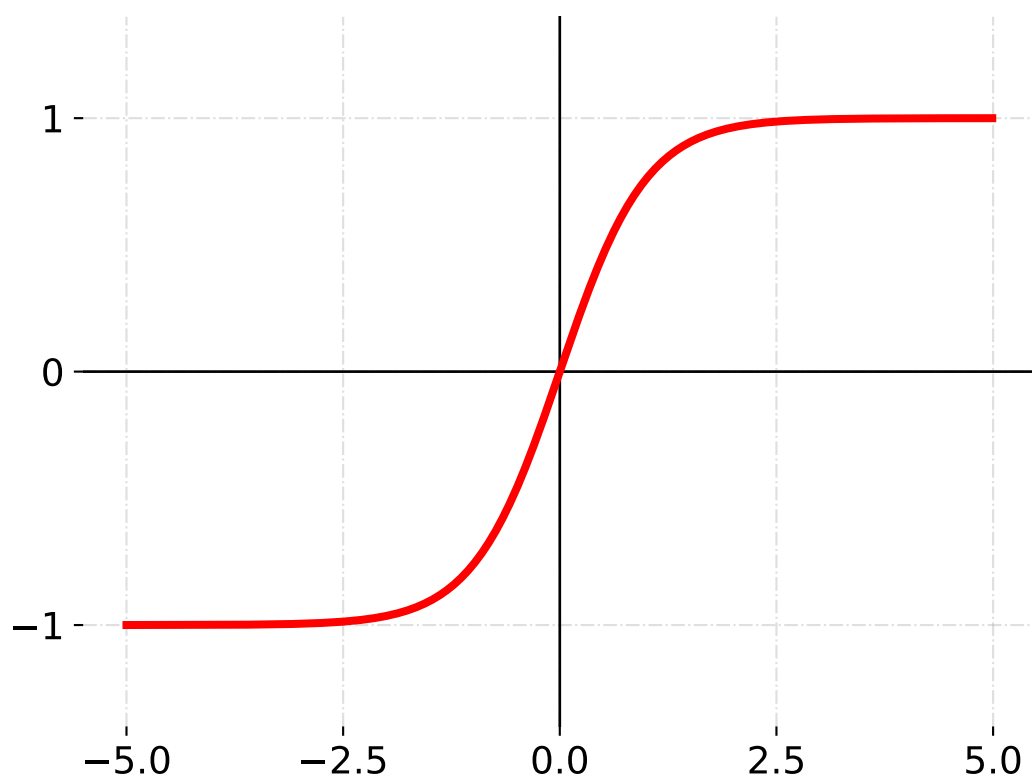
$$f'(x) = \frac{2e^x}{(e^x + 1)^2}$$



Hyperbolic Tangent, TanH

$$f(x) = \frac{2}{1 + e^{-2x}} - 1$$

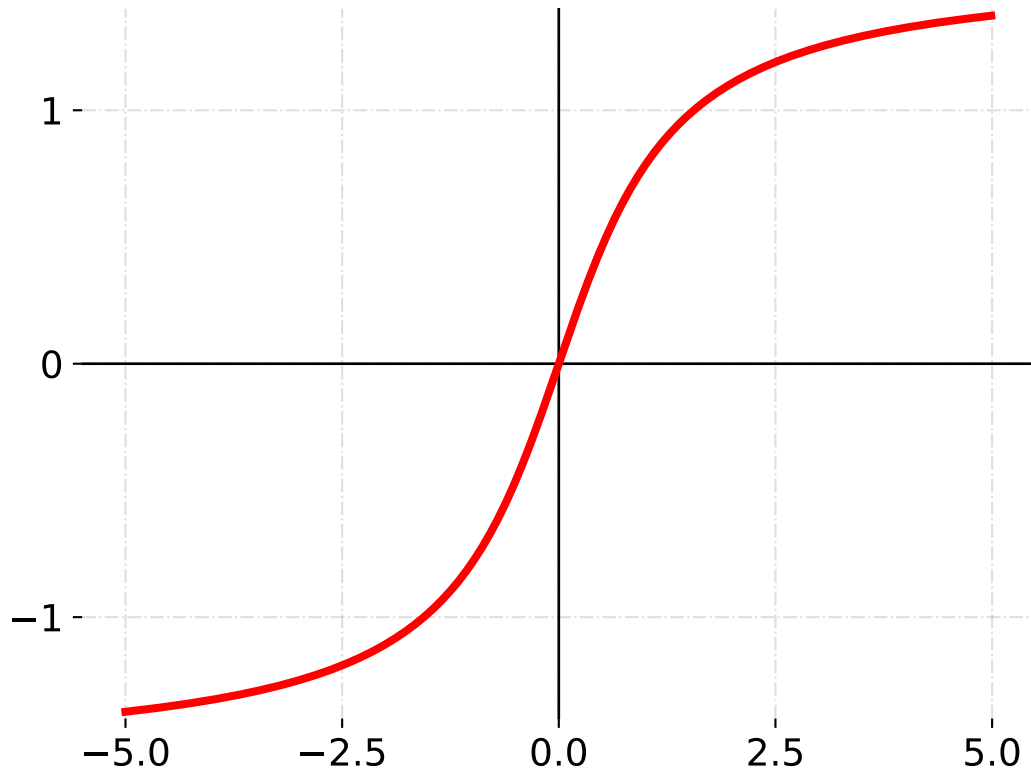
$$f'(x) = 1 - f(x)^2$$



Arctangent, ArcTan

$$f(x) = \tan^{-1}(x)$$

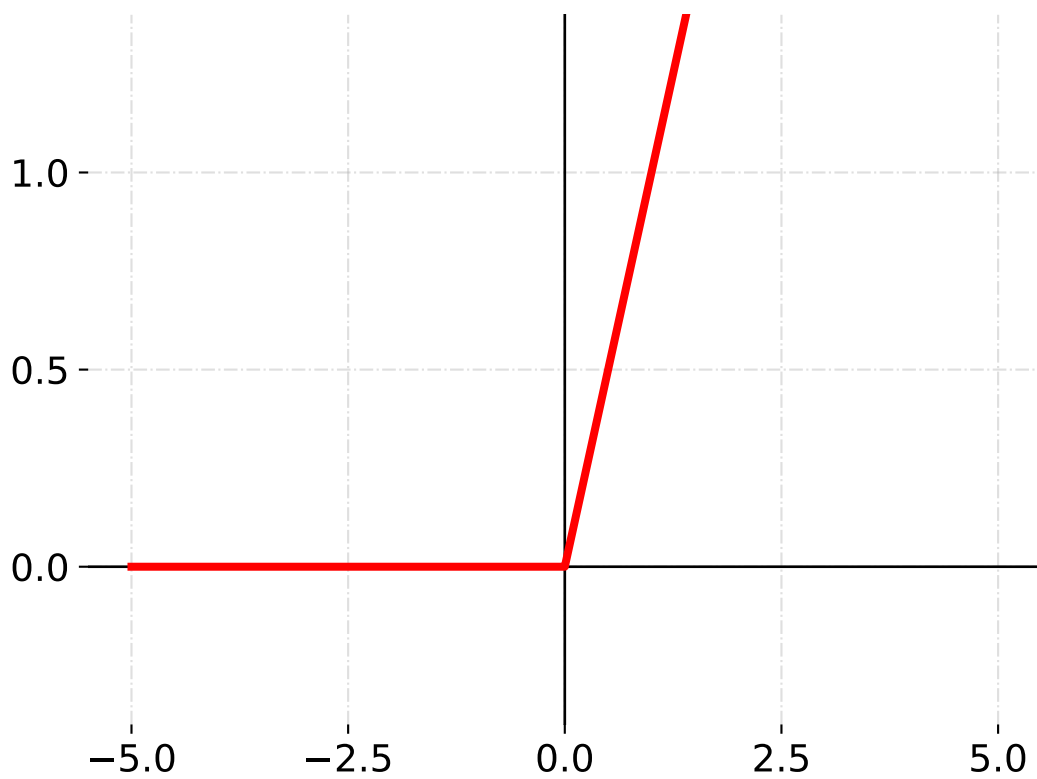
$$f'(x) = \frac{1}{1+x^2}$$



Rectified Linear Units, ReLU

$$f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}$$

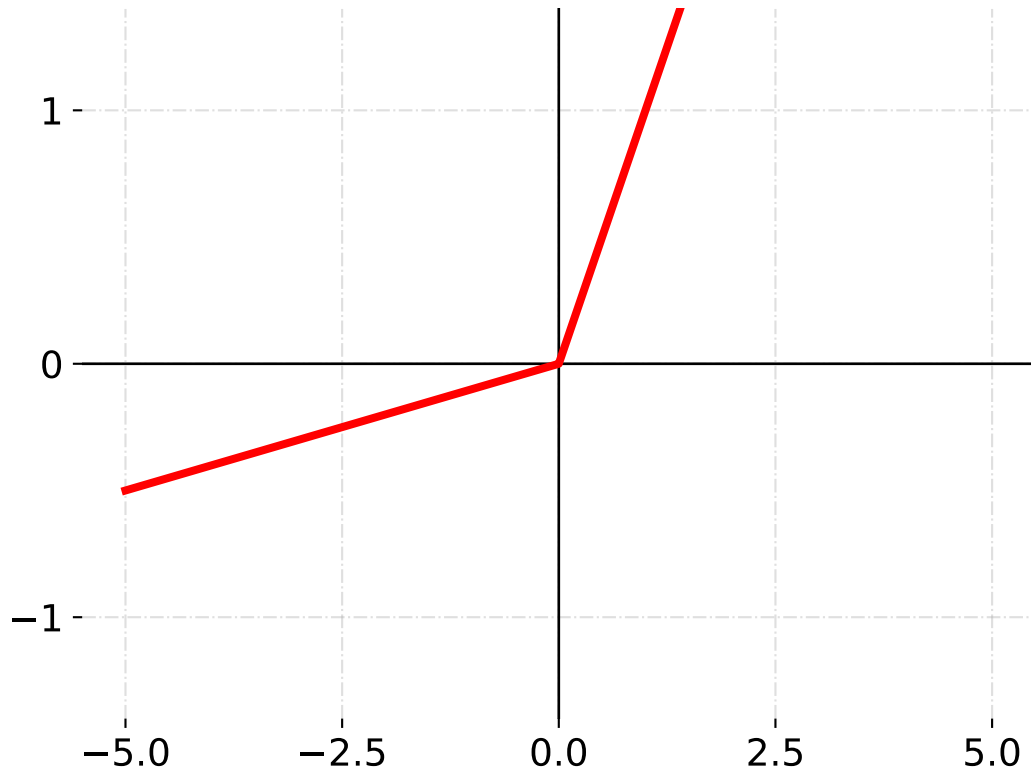
$$f'(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ 1 & \text{for } x > 0 \end{cases}$$



Leaky Rectified Linear Units, Leaky ReLU

$$f(x) = \begin{cases} ax & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}$$

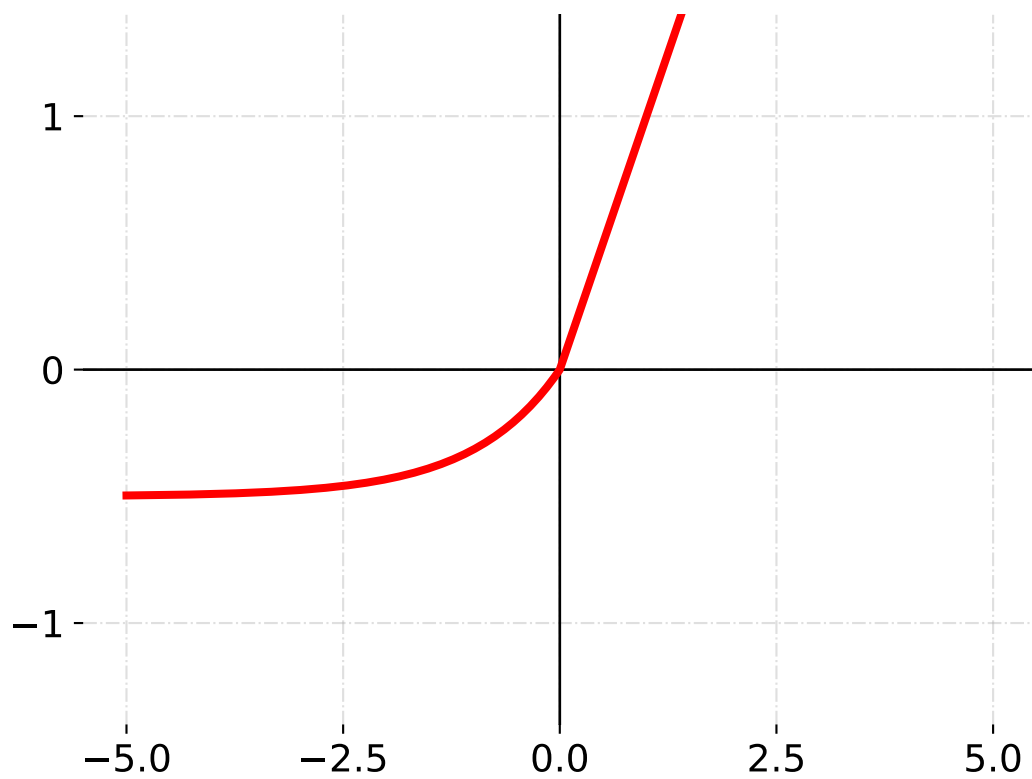
$$f'(x) = \begin{cases} a & \text{for } x \leq 0 \\ 1 & \text{for } x > 0 \end{cases}$$



Exponential Linear Units, ELU

$$f(x) = \begin{cases} a(e^x - 1) & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}$$

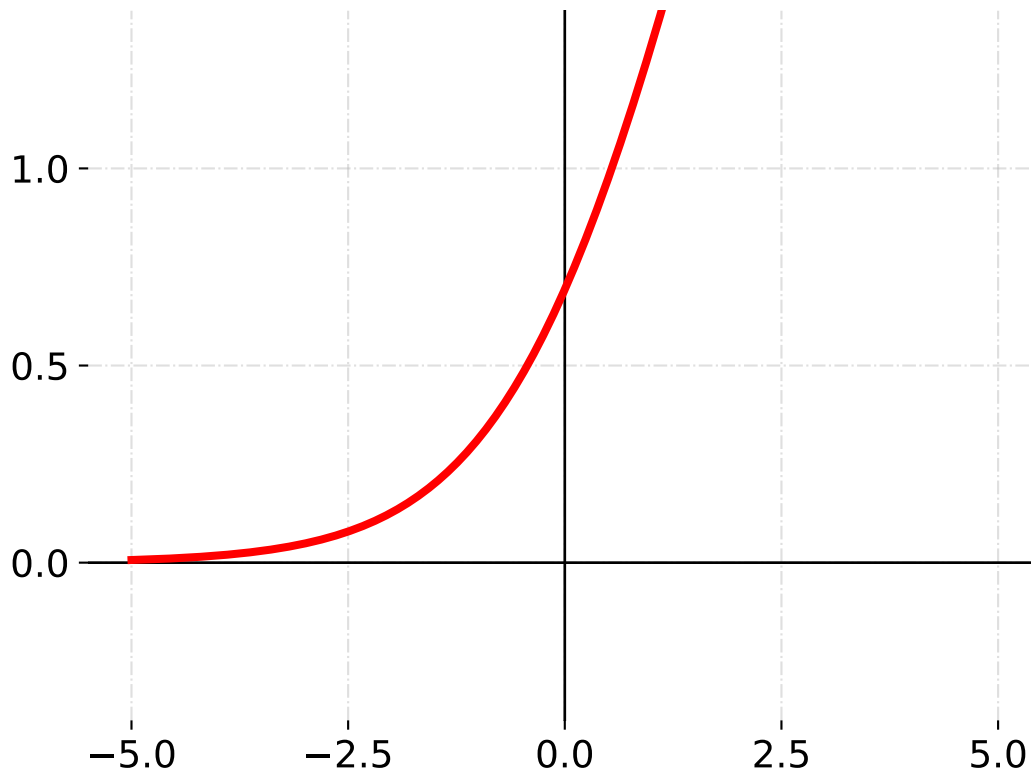
$$f'(x) = \begin{cases} f(x) + a & \text{for } x \leq 0 \\ 1 & \text{for } x > 0 \end{cases}$$



SoftPlus

$$f(x) = \ln(1 + e^x)$$

$$f'(x) = \frac{1}{1 + e^{-x}}$$



1.3.3 Changelog

Recent Changes

- **fix on spacings** by *San Askaruly* at 2021-08-18 08:56:55
- **hide members except visu_batch in biodatasets** by *San Askaruly* at 2021-08-17 18:14:16
- **biodatasets images change in docs** by *San Askaruly* at 2021-08-17 17:56:41

Commit [1acf1452f4](#)

Warning: There were uncommitted changes when this was compiled.

1.3.4 FAQ

- How to install pip package to python that uses conda?
 - Solution from [Github issue](#).
- How do i know on which GPU is my model?
 - Solution from [Stackoverflow answer](#).

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

BIBLIOGRAPHY

- [1] <https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia>
- [1] <https://www.kaggle.com/c/data-science-bowl-2018/overview>
- [1] <https://www.kaggle.com/c/histopathologic-cancer-detection/data> `_
- [1] <https://www.kaggle.com/c/ranzcr-clip-catheter-line-classification/data>
- [1] <https://www.kaggle.com/c/aptos2019-blindness-detection/data> `_

INDEX

C

ChestXrayDataset (class in *farabio.data.biodatasets*),
[4](#)

D

DSB18Dataset (class in *farabio.data.biodatasets*), [5](#)

H

HistocancerDataset (class in *farabio.data.biodatasets*), [6](#)

R

RANZCRDataset (class in *farabio.data.biodatasets*), [7](#)

RetinopathyDataset (class in *farabio.data.biodatasets*), [8](#)

V

visualize_batch() (farabio.data.biodatasets.ChestXrayDataset
method), [4](#)

visualize_batch() (farabio.data.biodatasets.DSB18Dataset
method), [5](#)

visualize_batch() (farabio.data.biodatasets.HistocancerDataset
method), [6](#)

visualize_batch() (farabio.data.biodatasets.RANZCRDataset
method), [7](#)

visualize_batch() (farabio.data.biodatasets.RetinopathyDataset
method), [8](#)