THE DEVELOPER'S CONFERENCE

Trilha – Machine Learning

Uso de PyTorch para Aplicações de Visão Computacional

Fulvio Mascara
Cientista-chefe @ Foursys
Community Manager @ AI Brasil

THE DEVELOPER'S CONFERENCE

CARA...

- Corredor de media e longa distância
- Diabético
- Programo desde os 11 anos
- "Tocador" de violão e guitarra
- Gosto de cozinhar e viajar

... E CRACHÁ!

- 30 anos de carreira
- Cientista-chefe (P&D&I) da Foursys
- Community Manager – AI Brasil
- Tecnológo em Proc. Dados (Mackenzie)
- Pós-Graduado em Solution Architect (FIAP)
- Pós-Graduado em Estatística Aplicada (FMU)

Agenda

THE DEVELOPER'S CONFERENCE

- Frameworks Deep Learning
- PyTorch
- Visão Computacional no PyTorch
- Demonstração
  - Notebook Jupyter
- Expectativas
- Links úteis

Frameworks Deep Learning

Panorama atual

THE DEVELOPER'S CONFERENCE

TensorFlow

Keras

Caffe

DL4J

Caffe2

Sonnet

mxnet

PYTORCH

ONNX

fast.ai

GLUON

Microsoft CNTK

Chainer

theano

Frameworks Deep Learning

Principais Características – Visão Mercado

THE DEVELOPER'S CONFERENCE

- Otimizado para performance
- Fácil de entender e codificar
- Bom suporte da comunidade
- Paralelizar processos para reduzir os cálculos
- Automaticamente calcule os gradientes

- Design de Código conveniente
- Exemplos / referências de código de alta qualidade
- Velocidade do treinamento e da inferência
- Quantidade de funções / métodos oferecidos "out of the box"
- Compatibilidade nas atualizações de versão
- Suporte a múltiplas linguagens de programação

Fonte: https://www.analyticsvidhya.com/blog/2019/03/deep-learning-frameworks-comparison/

Fonte: https://medium.com/apache-mxnet/a-way-to-benchmark-your-deep-learning-framework-on-premise-4f7a0f475726

Frameworks Deep Learning

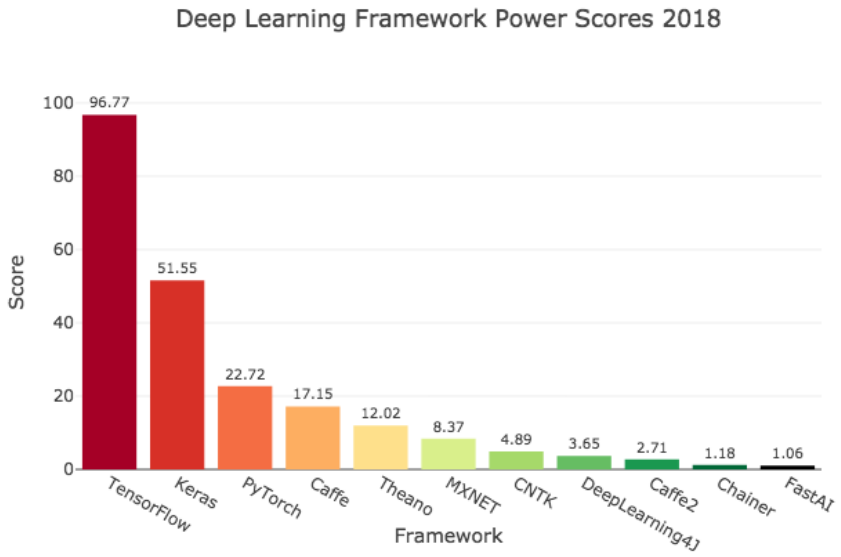
Principais Características – Visão pessoal

THE DEVELOPER'S CONFERENCE

- Constante evolução
- Comunidade vibrante, usando o framework tanto pra pesquisa quanto produção
- Rápida curva de aprendizado
- Ter recursos que permitam depurar o processo de treinamento das redes
- Modelos prontos para facilitar o processo de Transfer Learning
- Exportar o resultado para formatos que acelerem a inferência para Mobile e Edge Computing
- Velocidade no treinamento e na inferência

Frameworks Deep Learning

Adoção / Uso – Visão Setembro/2018



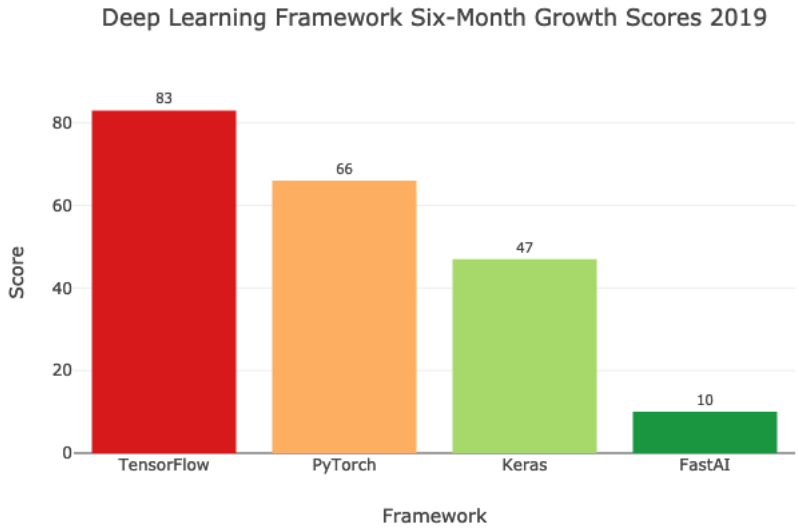
Critérios:

- Online Job Listings
- KDnuggets Usage Survey
- Google Search Volume
- Medium Articles
- Amazon Books
- ArXiv Articles
- GitHub Activity

Fonte: https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a

Frameworks Deep Learning

Adoção / Uso – Visão Abril/2019



Critérios:

- Online Job Listings
- Google Search Interest
- Medium Articles
- Quora Followers
- ArXiv Articles
- GitHub Activity

Fonte: https://towardsdatascience.com/which-deep-learning-framework-is-growing-fastest-3f77f14aa318

PyTorch
Histórico

THE DEVELOPER'S CONFERENCE

- Criado em Jan/16 pelo grupo de pesquisa de AI do Facebook
  - Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan
- Baseado no Torch
- Escrito em Python, C++ e CUDA
- PyTorch 1.0 lançado em Dez/18, com foco de se consolidar como um framework único "the path from research to production"

- Atualmente está na versão 1.1, lançada em Abril/19

PyTorch
Características

THE DEVELOPER'S CONFERENCE

- Fácil e muito flexível para prototipação
- Apesar das diversas features da v1.0, ainda é um framework muito mais voltado pra pesquisa
- Computação por Tensores, que acabam sendo uma abstração de arrays do numpy, mas com os ganhos computacionais do uso de GPU
- O fato de herdar as características do numpy, facilita a curva de aprendizado
- Estrutura de programação imperativa e não declarativa
- Dynamic Computational Graphs, ou seja, você pode alterar a arquitetura de uma rede durante sua execução, pois o "graph" é gerado "on the fly"
- Os DCGs são o que viabilizam um outro recurso do PyTorch: autograd. O autograd (módulo) usa um método de AD (Automatic Differentiation) pra "gravar" as operações de forward-pass e realizar um replay das mesmas pra calcular os gradientes no backprop
- Funcionalidades pra escalar o processo de treinamento das redes, provendo ganho de performance
- Possui compilador jit (just in time), que com o uso de Torch Script, torna possível rodar um código salvo no PyTorch, de forma compilada e sem necessidade do Python.
- Suporte nativo ao ONNX (Open Neural Network Exchange), o que facilita na integração com outras ferramentas compatíveis com este padrão
- Front-end híbrido, que permite transitar entre "eager mode" e "graph mode"
- Possui API e Front-end em C++
- Conta com datasets e modelos prontos pra uso, acelelerando o processo de prototipação e desenvolvimento
- Pronto para os principais vendors e plataformas de cloud AI/ML do mercado (AWS Sagemaker, Azure, GCP Kubeflow e Tensorboard)

PyTorch

Principais Estruturas

THE DEVELOPER'S CONFERENCE

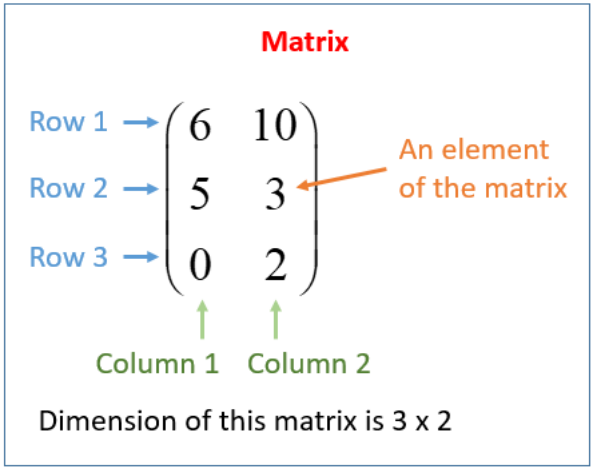
- Matrizes ou Tensores
- Operações com Tensores
- Variáveis e Gradientes (Autograd)

PyTorch

Principais Estruturas

THE DEVELOPER'S CONFERENCE

- Matrizes ou Tensores




```
import numpy as np
import torch

# 1) Create a PyTorch Tensor an array

arr = [[3, 4], [8, 5]] # python array
pyt_tensor = torch.Tensor(arr) # PyTorch tensor

# 2) Create a tensor

ones_tensor = torch.ones((2, 2)) # tensor containing all ones
torch.manual_seed(0)             # to have same values for random
generation
rand_tensor = torch.rand((2, 2)) # tensor containing random values

# if running on GPU, set random seed value as follows
if torch.cuda.is_available():
    torch.cuda.manual_seed_all(0)

# 3) Create a tensor from numpy array (dtype must be either double,
float, int64, int32, or uint8)

np_arr = np.ones((2, 2))
pyt_tensor = torch.from_numpy(np_arr)
np_arr_from_tensor = pyt_tensor.numpy() # convert tensor to numpy
array
```

Fonte: https://towardsdatascience.com/pytorch-fundamentals-50af6121d4a3

PyTorch

Principais Estruturas

THE DEVELOPER'S CONFERENCE

- Operações com Tensores

```
import numpy as np
import torch

# 1) Resizing a tensor

pyt_tensor = torch.ones((2, 2))
print(pyt_tensor.size())        # shows the size of this tensor
pyt_tensor = pyt_tensor.view(4) # resizing 2x2 tensor to 4x1

# 2) Mathematical Operations

pyt_tensor_a = torch.ones((2, 2))
pyt_tensor_b = torch.ones((2, 2))
res_tensor = pyt_tensor_a + pyt_tensor_b               # simple
element wise addidtion
res_tensor = torch.add(pyt_tensor_a, pyt_tensor_b)     # another way
of addidtion
pyt_tensor_a.add_(pyt_tensor_b)                         # In-place
addition

# Operation     operator    function_name
#  => Addition        +           add
#  => Subtraction     -           sub
#  => Multiplication  *           mul
#  => Divide          /           div
```

```
# 3) Mean and Standart deviation

pyt_tensor = torch.Tensor([1, 2, 3, 4, 5])
mean = pyt_tensor.mean(dim=0)                          # if multiple
rows then dim = 1
std_dev = pyt_tensor.std(dim=0)                          # if multiple
rows then dim = 1
```

Fonte: https://towardsdatascience.com/pytorch-fundamentals-50af6121d4a3

PyTorch

Principais Estruturas

THE DEVELOPER'S CONFERENCE

- Variáveis e Gradientes (Autograd)

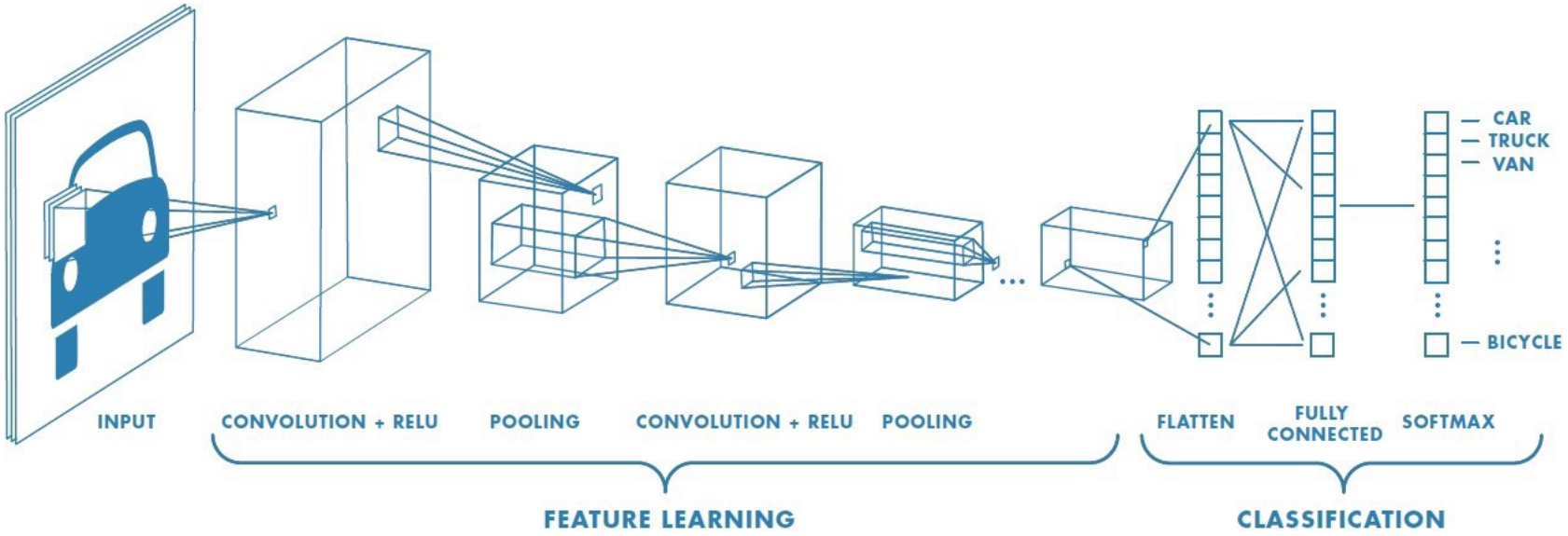
```
import numpy as np
import torch
from torch.autograd import Variable

pyt_var = Variable(torch.ones((2, 2)), requires_grad = True)

# behaves exactly the same as tensors, so we can apply all
operations in the same way
```

```
import numpy as np
import torch
from torch.autograd import Variable

# let's consider the following equation
# y = 5(x + 1)^2

x = Variable(torch.ones(1), requires_grad = True)
y = 5 * (x + 1) ** 2        # implementing the equation.
y.backward()                # calculate gradient
print(x.grad)               # get the gradient of variable x
# differentiating the above mentioned equation
# => 5(x + 1)^2 = 10(x + 1) = 10(2) = 20
```

Fonte: https://towardsdatascience.com/pytorch-fundamentals-50af6121d4a3

PyTorch

CNN – Convolutional Neural Networks



https://miro.medium.com/max/2625/1*vkQ0hXDaQv57sALXAJquxA.jpeg

PyTorch

Visão Computacional e CNNs

THE DEVELOPER'S CONFERENCE

torch.nn

Convolution Layers

- Conv1d
- Conv2d
- Conv3d
- ConvTranspose1d
- ConvTranspose2d
- ConvTranspose3d
- Unfold
- Fold

Pooling Layers

- MaxPool1d
- MaxPool2d
- MaxPool3d
- MaxUnpool1d
- MaxUnpool2d
- MaxUnpool3d
- AvgPool1d
- AvgPool2d
- AvgPool3d
- FractionalMaxPool2d
- LPPool1d
- LPPool2d
- AdaptiveMaxPool1d
- AdaptiveMaxPool2d
- AdaptiveMaxPool3d
- AdaptiveAvgPool1d
- AdaptiveAvgPool2d
- AdaptiveAvgPool3d

Padding Layers

- ReflectionPad1d
- ReflectionPad2d
- ReplicationPad1d
- ReplicationPad2d
- ReplicationPad3d
- ZeroPad2d
- ConstantPad1d
- ConstantPad2d
- ConstantPad3d

Dropout Layers

- Dropout
- Dropout2d
- Dropout3d
- AlphaDropout

Vision Layers

- PixelShuffle
- Upsample
- UpsamplingNearest2d
- UpsamplingBilinear2d

PyTorch

Visão Computacional e CNNs

THE DEVELOPER'S CONFERENCE

torch.nn.functional

Convolution Functions

- conv1d
- conv2d
- conv3d
- conv_transpose1d
- conv_transpose2d
- conv_transpose3d
- unfold
- fold

Pooling Functions

- avg_pool1d
- avg_pool2d
- avg_pool3d
- max_pool1d
- max_pool2d
- max_pool3d
- max_unpool1d
- max_unpool2d
- max_unpool3d
- lp_pool1d
- lp_pool2d
- adaptive_max_pool1d
- adaptive_max_pool2d
- adaptive_max_pool3d
- adaptive_avg_pool1d
- adaptive_avg_pool2d
- adaptive_avg_pool3d

Dropout Functions

- dropout
- alpha_dropout
- dropout2d
- dropout3d

Vision Functions

- pixel_shuffle
- pad
- interpolate
- upsample
- upsample_nearest
- upsample_bilinear
- grid_sample
- affine_grid

PyTorch

Visão Computacional e CNNs

THE DEVELOPER'S CONFERENCE

torchvision

datasets

- MNIST
- Fashion-MNIST
- KMNIST
- EMNIST
- FakeData
- COCO
  - Captions
  - Detection
- LSUN
- ImageFolder
- DatasetFolder
- ImageNet
- CIFAR
- STL10
- SVHN
- PhotoTour
- SBU
- Flickr
- VOC
- Cityscapes
- SBD

models

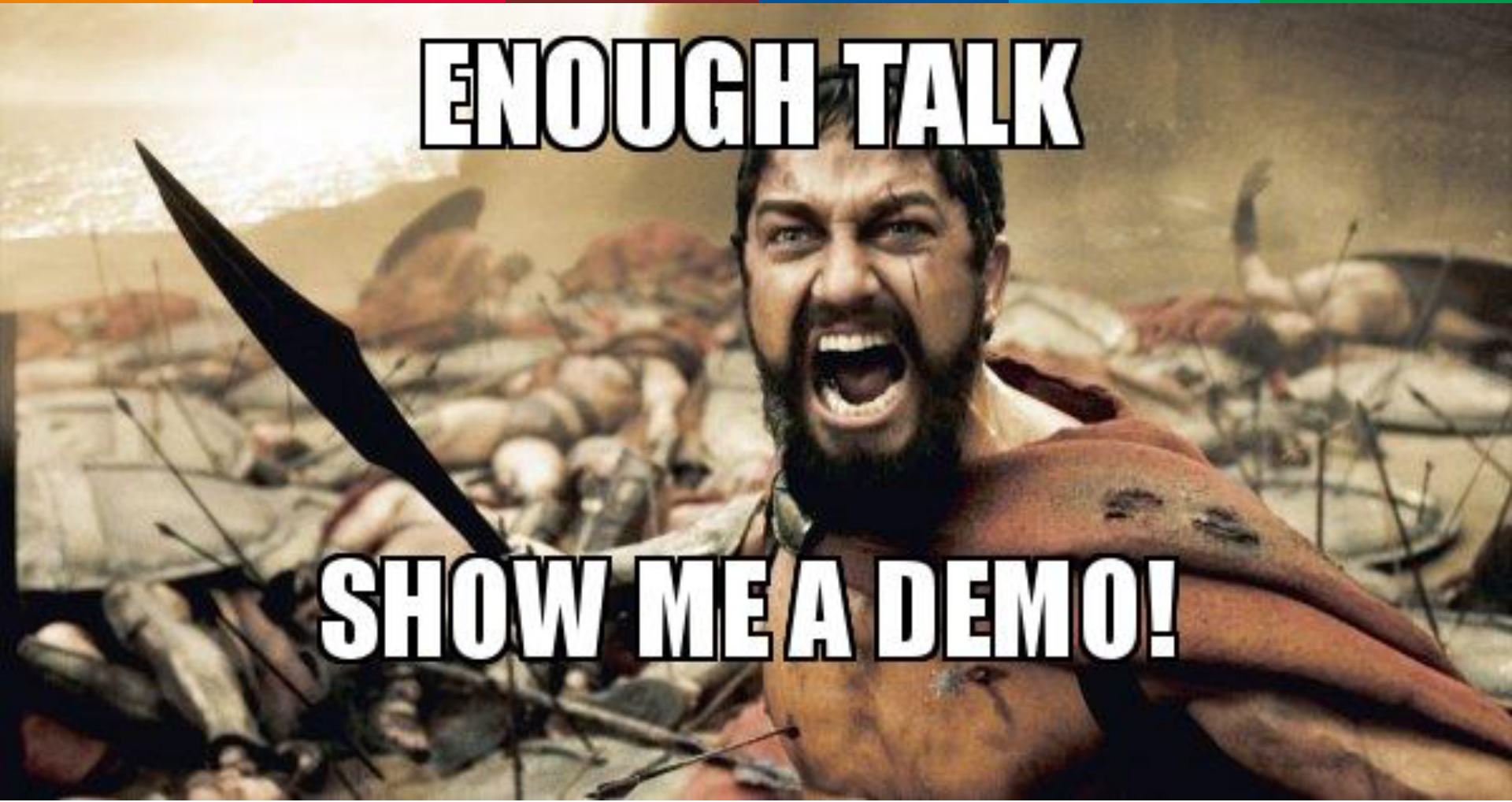
- Classification
  - Alexnet
  - VGG
  - ResNet
  - SqueezeNet
  - DenseNet
  - Inception v3
  - GoogLeNet
  - ShuffleNet v2
  - MobileNet v2
  - ResNext
- Semantic Segmentation
  - Fully Convolutional Networks
  - DeepLabV3
- Object Detection, Instance Segmentation an Keypoint Detection
  - Runtime characteristics
  - Faster R-CNN
  - Mask R-CNN
  - Keypoint R-CNN

transforms

- Crop
- Resize
- Scale
- Flip
- Rotate
- Color
- Brightness / Contrast / Hue
- Affine

- Compose (Pipeline)

- PIL Image
- Tensor


ENOUGH TALK
SHOW ME A DEMO!

Expectativas

THE DEVELOPER'S CONFERENCE

- Aumentar a adoção em ambientes produtivos
- PyTorch Lite
- Ter uma versão não imperativa

Conclusões

Links úteis

THE DEVELOPER'S CONFERENCE

Awesome List – PyTorch
https://github.com/bharathgs/Awesome-pytorch-list

The Incredible PyTorch
https://www.ritchieng.com/the-incredible-pytorch/

Fast.ai
https://docs.fast.ai/index.html

Chainer
https://chainer.org

Contatos

THE DEVELOPER'S CONFERENCE

LinkedIn

in/fulviomascara

twitter

@fulviomascara

Gmail

fulvio.mascara

GitHub

fulviomascara

AI Brasil
an artificial intelligence community

THE DEVELOPER'S CONFERENCE

LinkedIn

Meetup

WhatsApp

facebook

Muito obrigado !

THE DEVELOPER'S
CONFERENCE