

Introducción a PyTorch

Enrique Escalante-Notario

Outline

- D Regresión y varianza.
- D Instalación
- D Manejo básico de PyTorch
- D Ejemplo
 - Red de convolución de imágenes.

PyTorch

From Wikipedia, the free encyclopedia

PyTorch is an [open source](#) machine learning library based on the [Torch](#) library,^{[3][4][5]} used for applications such as computer vision and natural language processing,^[6] primarily developed by Facebook's AI Research lab (FAIR).^{[7][8][9]} It is free and open-source software released under the [Modified BSD license](#). Although the [Python](#) interface is more polished and the primary focus of development, PyTorch also has a [C++](#) interface.^[10]

A number of pieces of Deep Learning software are built on top of PyTorch, including Tesla Autopilot^[11], Uber's Pyro,^[12] HuggingFace's Transformers,^[13] PyTorch Lightning^{[14][15]}, and Catalyst.^{[16][17]}

PyTorch provides two high-level features:^[18]

- Tensor computing (like [NumPy](#)) with strong acceleration via [graphics processing units \(GPU\)](#)
- Deep neural networks built on a tape-based [automatic differentiation](#) system

PyTorch

- ▷ Es un framework Python 3.X (X=7)
- ▷ Alguno de sus puntos "Tensorflow Killer"
- ▷ Basado en numpy
- ▷ Es usado en el Deep Learning;
"yo" se programó flexible y sencillo.

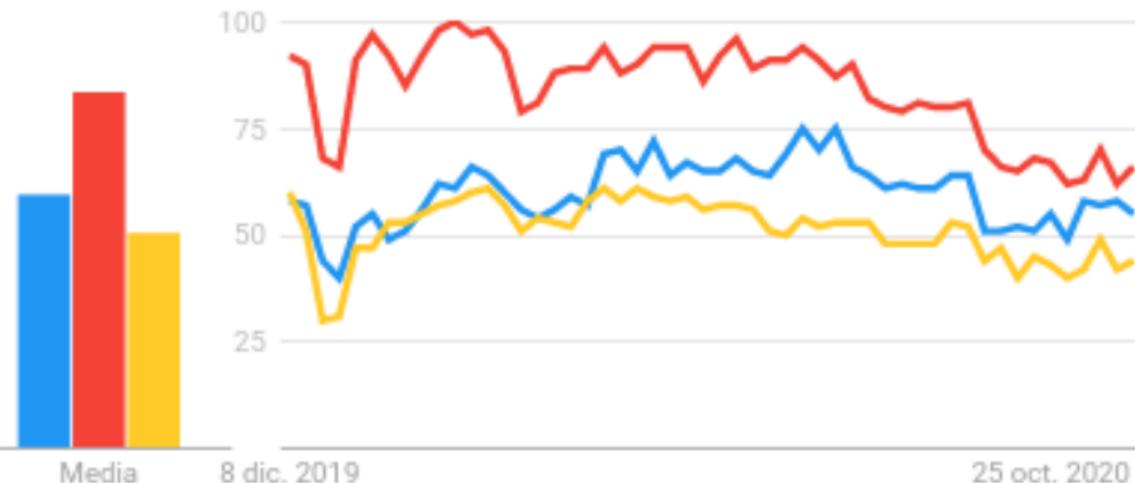
Alternatives to PyTorch

- ③ TensorFlow {
 - calculates neurons
 - mediante graph
- ④ Theano {
 - calcula simbolica
 - de forma.
- ⑤ Keras {
 - Rede neural

Interés a lo largo del tiempo

Google Trends

● PyTorch ● TensorFlow ● Scikit-learn



¿Por qué PyTorch?

- ▷ Python
- ▷ Rápido crecimiento
- ▷ Un mundo de soporte para la ciencia
- ▷ Sencillas implementaciones
- ▷ Trabajar con gráficos dinámicos
- ▷ Soporte para CUDA

Instalaciones

▷ Python 3.x ✓

▷ Pip, cardo o un instalador de paquetes
de Python ✓

(1) X Si, tenues una GPU se pueblan procesos

(-1) X Podemos usar Pytorch sobre C++/Java.

▷ Podemos elegir entre la rama de
desarrollo o estable ✓

Instalaciones

- ▷ Python 3.x ✓
- ▷ Pip, conda o un instalador de paquetes de Python ✓
- ▷ Puedes elegir entre la rama de desarrollo o estable ✓

```
> sudo pip3 install torch torchvision torchaudio  
matplotlib numpy utils
```

Instalación

- ▷ Python 3.x ✓
 - ▷ Pip, conda o una instalación de paquetes de Python ✓
 - ▷ Podemos elegir entre la rama de desarrollo o estable ✓
- > sudo pip3 install jupyter
- Alternativo, pero útil para aprender

Terminal - python3

```
→ ~ python3
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> import torchvision
>>> x = torch.rand(5,3)
>>> print(x)
tensor([[0.5280, 0.9970, 0.7704],
       [0.1556, 0.6320, 0.7765],
       [0.2484, 0.7608, 0.1847],
       [0.1539, 0.1815, 0.6031],
       [0.4293, 0.2727, 0.5217]])
>>> torch.cuda.is_available()
/usr/local/lib/python3.8/dist-packages/torch/cuda/__init__.py:52: UserWarning: CUDA initialization: Found no NVIDIA driver on your system. Please check that you have an NVIDIA GPU and installed a driver from http://www.nvidia.com/Download/index.aspx (Triggered internally at /pytorch/c10/cuda/CUDAFunctions.cpp:100.)
    return torch._C._cuda_getDeviceCount() > 0
False
>>> █
```

Voy a ver en libreta de Jupyter.

```
Terminal - jupyter notebook
→ ~ jupyter notebook
[I 10:31:47.082 NotebookApp] Serving notebooks from local directory: /home/enrique
[I 10:31:47.083 NotebookApp] Jupyter Notebook 6.1.4 is running at:
[I 10:31:47.083 NotebookApp] http://localhost:8888/?token=a17b7562d0f2e42d46e381
1817285d6636d8d1a1008fbefd
[I 10:31:47.083 NotebookApp] or http://127.0.0.1:8888/?token=a17b7562d0f2e42d46
e3811817285d6636d8d1a1008fbefd
[I 10:31:47.083 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[C 10:31:47.133 NotebookApp]

To access the notebook, open this file in a browser:
file:///home/enrique/.local/share/jupyter/runtime/nbserver-337117-open.h
tml
Or copy and paste one of these URLs:
http://localhost:8888/?token=a17b7562d0f2e42d46e3811817285d6636d8d1a1008
fbefd
or http://127.0.0.1:8888/?token=a17b7562d0f2e42d46e3811817285d6636d8d1a1008
fbefd
```

Home Page - Select or create a new notebook

localhost:8888/tree

jupyter

Files Running Clusters

Duplicate Rename Move Download View Edit

Upload New

| | Name | Last Modified | File size |
|-------------------------------------|---------------------------|----------------|-----------|
| <input type="checkbox"/> | Descargas | hace 2 minutos | |
| <input type="checkbox"/> | Documentos | hace 16 horas | |
| <input type="checkbox"/> | Downloads | hace 5 días | |
| <input type="checkbox"/> | Escritorio | hace 5 días | |
| <input type="checkbox"/> | Imágenes | hace un minuto | |
| <input type="checkbox"/> | Plantillas | hace 2 meses | |
| <input type="checkbox"/> | snap | hace 10 días | |
| <input type="checkbox"/> | 2020-11-27-Note-16-05.pdf | hace un minuto | 5.81 MB |
| <input type="checkbox"/> | 2020-11-27-Note-17-28.xoj | hace un minuto | 726 kB |
| <input checked="" type="checkbox"/> | orbi2ml | hace un mes | 384 B |

Values a copy for.

Evolución de las redes neuronales

Etapas I

1958 - Perceptron

1965 - Multilayer perceptron.

1980

- Neuronas sigmoidales
- Redes Feed forward
- Fully connected

$$d(z) = \frac{1}{1 + e^{-z}}$$

Biases ||
Pesos ||
Manojo ||
Multicapa ||
Difícil de ||
calcular

faltantes ||
Primeras funciones
de activación ||
- Sigmoidal
- Bias ||
Control ||
los estados en ||
aproximadas personas

Etapas 2

1986 - Back propagation

No hay loops !!
Todos los
neuronas ↴
Conectadas.

Rodar
entrena de
modo supervisado
a fondo los
pesos !!

1989 - Convolutional Neural Network

{ Una capa extenderá info y luego clasificarán.
Se reduce el número de features manteniendo la info más importante.
Las últimas capas tiene nodos para cada clasificación.

1997 Learning short term
Memory / Recurrent
Neural Network

Conexión hetero
áreas ~ Señal temporal
Alta tasa de
niveles de
memoria

Etapas 3

Deep Learning

2006 Deep Belief Network.

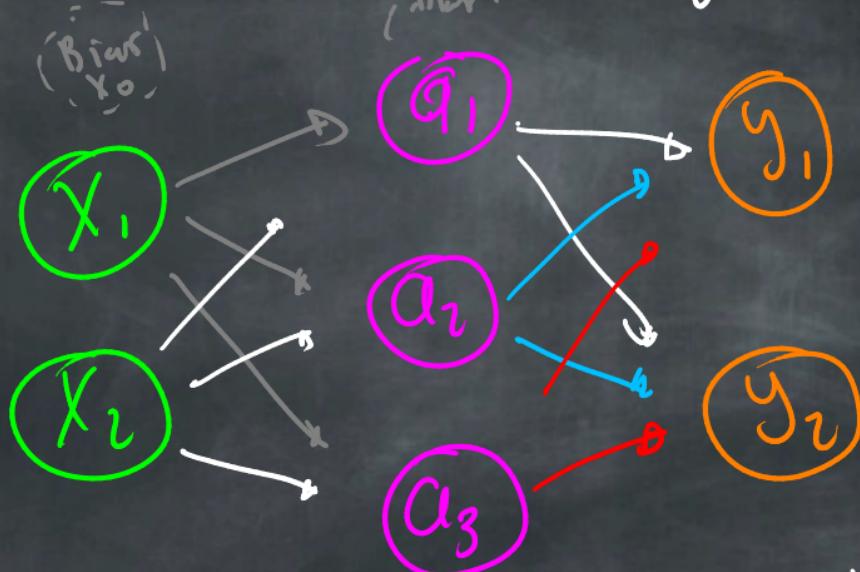
Autoencoder con Restricted
Boltzmann Machine

No tiene el
poder de generalizar
Aprendizamiento
no supervisado
Evitar caer en
minimos locales
por Gradi Decent-

2014 Generative Adversarial
Network

Se enthalten 2
re^ct generat.,
discriminat.

Considerar el siguiente ejemplo



Entrada
 x_i

activación
 a_i

salida
 y_j :

Existe

$O(j)$ Transformación lineal con
los pesos que mapan entre
los pesos j y $j+1$

Entonces

$$a_i = g(O_i^T X)$$

g es una función de activación
{Sigmoid, tanh, ReLU ...}

Feed Propagation

Resumo de "Técnicas a Direita"

① $X = (x_0, x_1, x_2)$

② $z^{cap^2} = \theta_1 X$

③ $a_i^{cap^2} = g(z^{cap^2})$

④ $z^{cap^3} = \theta_2 a^{cap^2}$

⑤ $y = g(z^{cap^3})$

$g's$ podem ser
diferentes

Jinapir

Back propagation (función descendiente)

- * El modelo tiene forma a aprender
- * Vemos de "descender a la gavilla" para mejorar los procedimientos.
- * Dos formas:
 - ① Propagación
 - ② Actualizar pesos

① Proponer

①.1 Entrenar los FP supervisando para activar
los salidas de la red.

①.2 Hacer BP, para generar los errores (Δ)
de todos los pesos de salida
y de los capas ocultas.

② Adulterar puros

Para calcular se requiere de los puros

②.1 Multiplicar D de su licor por su
aditivo en la medida para obtener
el guardaante del puro.

②.2 Quitar un porcentaje del guardaante
de ese puro

El porcentaje se conoce como "Learning rate"

- Si es grande, aprende más rápido
pero impreciso.

- Si es pequeño, aprende más lento
y podrían no terminar nunca

Repetimos
en distancias

el mismo punto cada
vez al aprendizaje

Si $\Delta(l)$ es el error de la capa l .

$$\Delta(3) = a^{capa\ 3} - y$$

$$\Delta(2) = O_2^T \Delta(3) \cdot g'(z^{capa\ 2})$$

Buscando minimizar

$$J = a^{capa} \Delta^{capa+1}$$

Este valor juntado con el learning rate, Buscar el minimo global.

Reynolds a Superstar