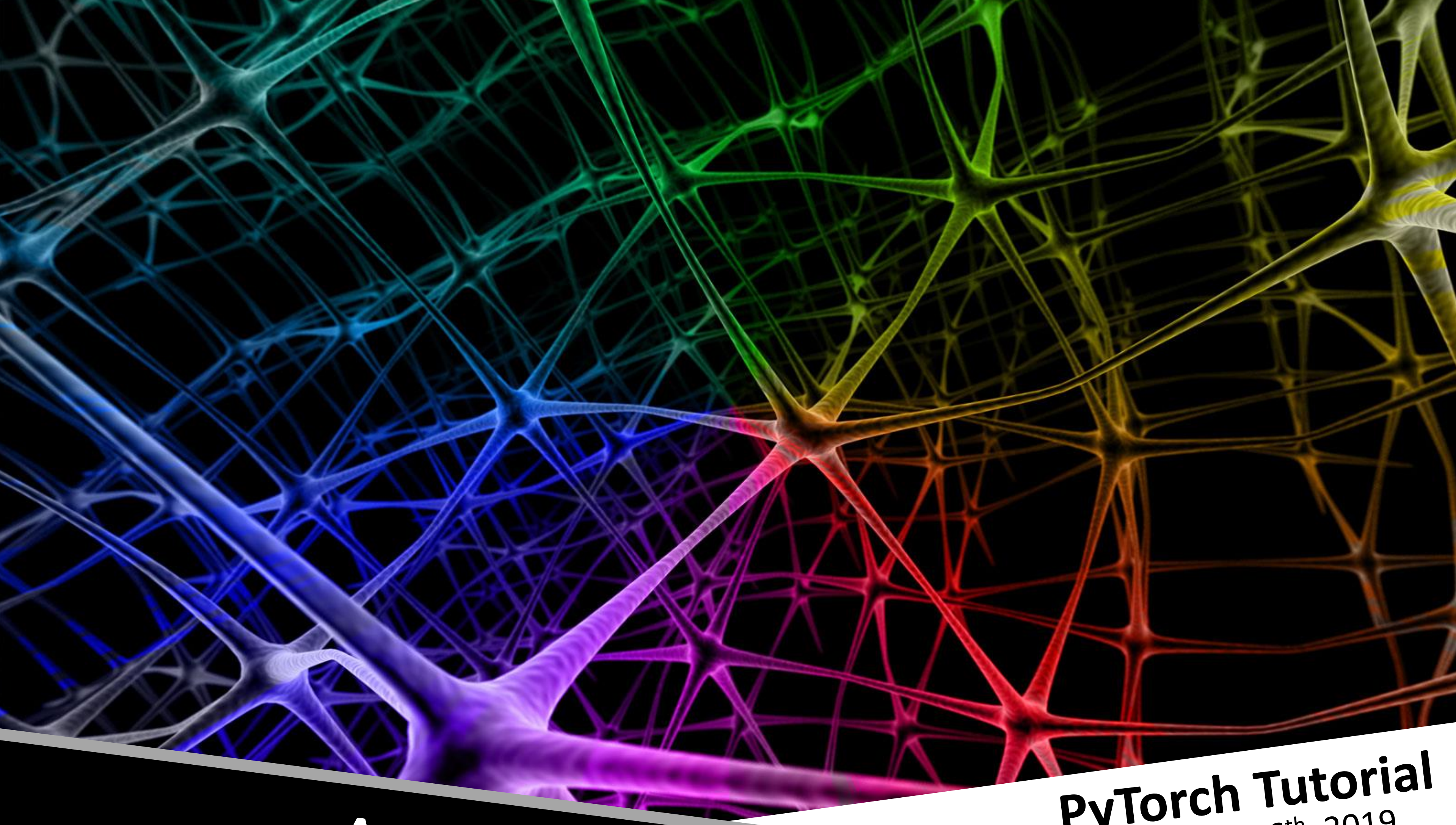


PyTorch Tutorial

Feb 26th, 2019

Applied Deep Learning

SHANG-YU SU

HTTP://ADL.MIULAB.TW

國立臺灣大學
National Taiwan University

Deep Learning Framework

Chainer mxnet Caffe2

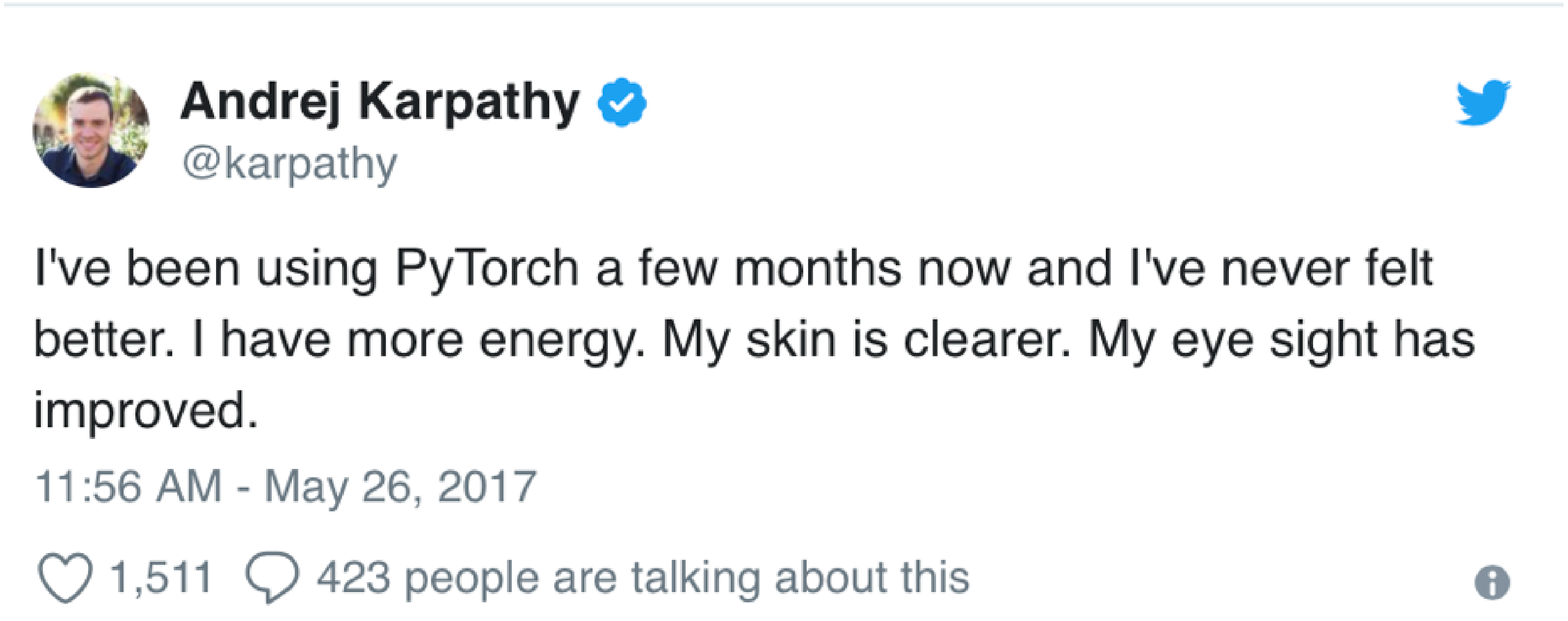
Microsoft CNTK TensorFlow K Keras

GLUON PYTORCH theano

What is PyTorch?

- A replacement for NumPy to use the power of GPUs
- A deep learning research platform that provides maximum flexibility and speed
- Developed by Facebook

Why PyTorch?


Andrej Karpathy
@karpathy

I've been using PyTorch a few months now and I've never felt better. I have more energy. My skin is clearer. My eye sight has improved.

11:56 AM - May 26, 2017

1,511 423 people are talking about this

Why PyTorch?

- The TAs of this course are more familiar with this toolkit
- Flexibility, easy-to-use, ecosystem…

Tensors

- Tensors are similar to NumPy's ndarrays, with the addition being that Tensors can also be used on a GPU to accelerate computing.

Data type	dtype	CPU tensor	GPU tensor
32-bit floating point	`torch.float32` or `torch.float`	`torch.FloatTensor`	`torch.cuda.FloatTensor`
64-bit floating point	`torch.float64` or `torch.double`	`torch.DoubleTensor`	`torch.cuda.DoubleTensor`
16-bit floating point	`torch.float16` or `torch.half`	`torch.HalfTensor`	`torch.cuda.HalfTensor`
8-bit integer (unsigned)	`torch.uint8`	`torch.ByteTensor`	`torch.cuda.ByteTensor`
8-bit integer (signed)	`torch.int8`	`torch.CharTensor`	`torch.cuda.CharTensor`
16-bit integer (signed)	`torch.int16` or `torch.short`	`torch.ShortTensor`	`torch.cuda.ShortTensor`
32-bit integer (signed)	`torch.int32` or `torch.int`	`torch.IntTensor`	`torch.cuda.IntTensor`
64-bit integer (signed)	`torch.int64` or `torch.long`	`torch.LongTensor`	`torch.cuda.LongTensor`

Computational Graph

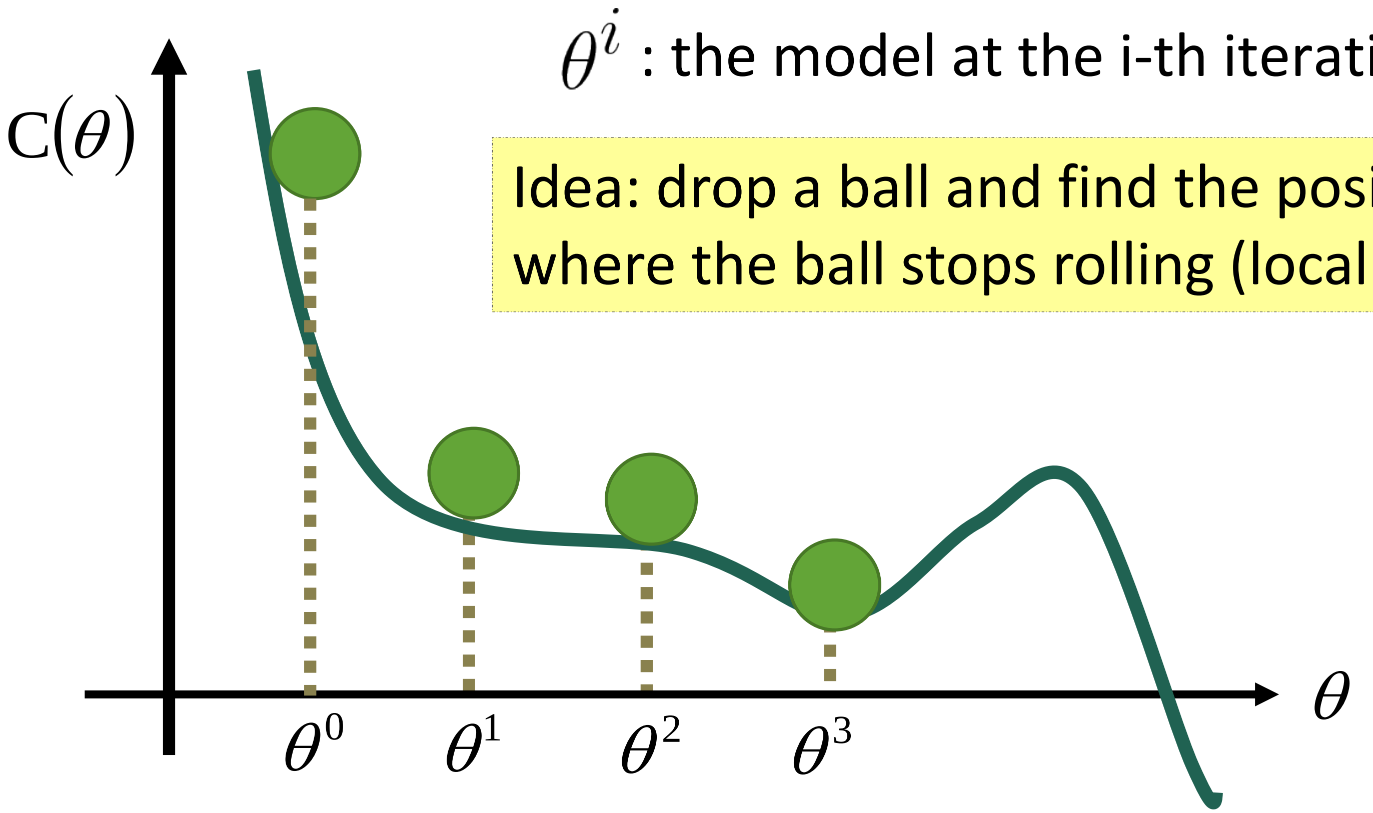
- A computation graph is a DAG in which nodes represent variables (tensors, matrix, scalars, etc.) and edge represent some mathematical operations (for example, summation, multiplication).
- The computation graph has some leaf variables, the root variables of the graph are computed according to operations defined by the graph.
- During the optimization step, we combine the chain rule and the graph to compute the derivative of the output w.r.t the learnable variable in the graph and update these variables to make the output close to what we want.
- In neural networks, these learnable variables are often called weight and bias.

torch.autograd

- torch.autograd provides classes and functions implementing automatic differentiation of arbitrary scalar valued functions. It requires minimal changes to the existing code - you only need to declare Tensors for which gradients should be computed with the requires_grad=True keyword.

Gradient Descent for Optimization

Assume that θ has only one variable

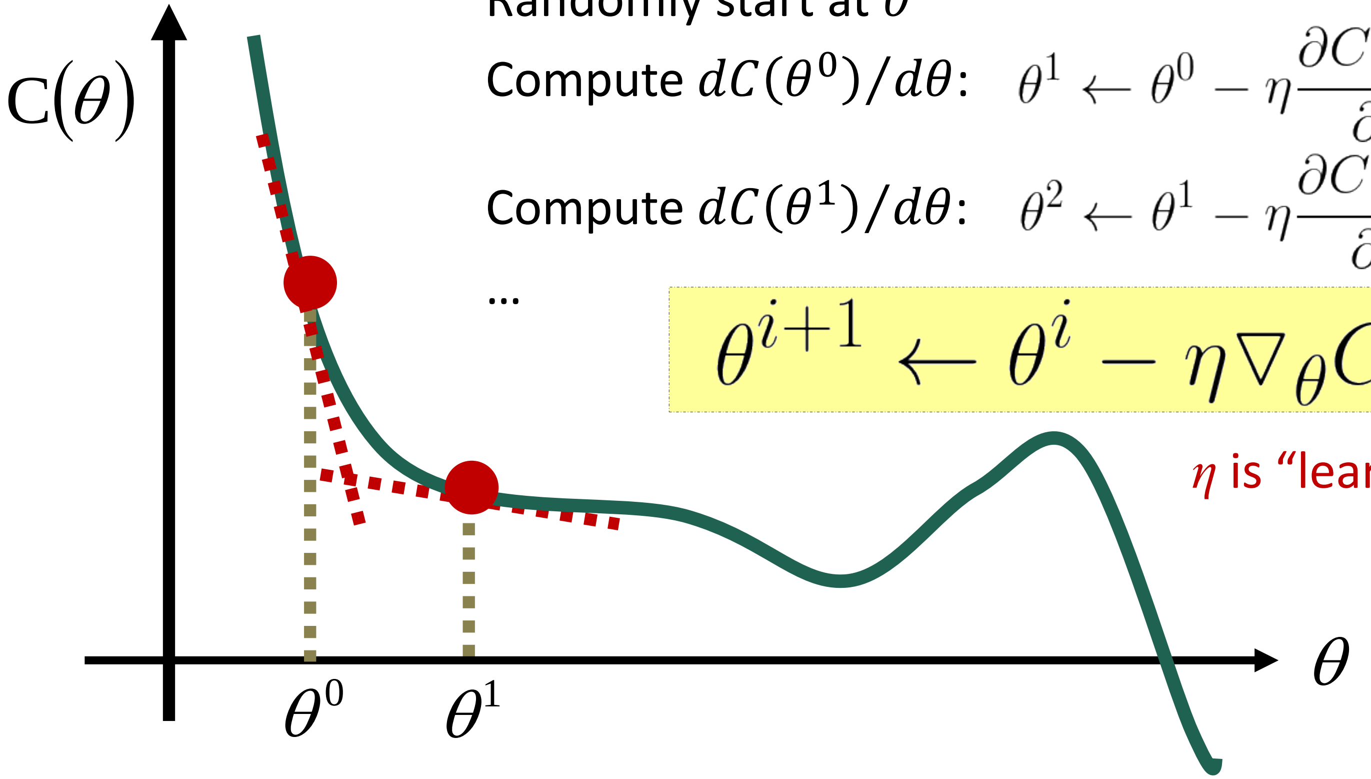


θ^i : the model at the i-th iteration

Idea: drop a ball and find the position where the ball stops rolling (local minima)

Gradient Descent for Optimization

Assume that θ has only one variable



Randomly start at θ^0

Compute $dC(\theta^0)/d\theta$: $\theta^1 \leftarrow \theta^0 - \eta \frac{\partial C(\theta^0)}{\partial \theta}$

Compute $dC(\theta^1)/d\theta$: $\theta^2 \leftarrow \theta^1 - \eta \frac{\partial C(\theta^1)}{\partial \theta}$

…

$$\theta^{i+1} \leftarrow \theta^i - \eta \nabla_\theta C(\theta^i)$$

η is “learning rate”

torch.nn

- Neural networks can be constructed using the torch.nn package.
- nn depends on autograd to define models and differentiate them.
- An nn.Module contains layers, and a method forward(input)that returns the output

torch.optim

- A package implementing various optimization algorithms.
- Most commonly used methods are already supported
- SGD, Adam, RMSProp, …etc.

Resources

- https://pytorch.org/tutorials/
- https://github.com/yunjey/pytorch-tutorial