Inria
INVENTEURS DU MONDE NUMÉRIQUE

AgroParisTech
INSTITUT DES SCIENCES ET INDUSTRIES DU VIVANT ET DE L'ENVIRONNEMENT
PARIS INSTITUTE OF TECHNOLOGY FOR LIFE, FOOD AND ENVIRONMENTAL SCIENCES

Deep Learning

Teacher: Alexis Joly (Inria, Pl@ntNet)

Source: course of François Fleuret at EPFL
https://fleuret.org/dlc/ (cc-by-nc-sa)

EPFL

idiap
RESEARCH INSTITUTE

Plan du cours

Source: course of François Fleuret at EPFL https://fleuret.org/dlc/ (cc-by-nc-sa)

Séance 1

Introduction

1.1. – From neural networks to deep learning. (18 slides) (slides)
1.2. – Current applications and success. (25 slides) (slides)
1.3. – What is really happening? (10 slides) (slides)
1.4. – Tensor basics and linear regression. (13 slides) (slides)
1.5. – High dimension tensors. (20 slides) (slides)

From perceptron to deep convolutional neural networks

3.1. – The perceptron. (16 slides) (slides)
3.3. – Linear separability and feature design. (10 slides) (slides)
3.4. – Multi-Layer Perceptrons. (10 slides) (slides)
4.4. – Convolutions. (23 slides) (slides)
4.5. – Pooling. (7 slides) (slides)
8.2. – Networks for image classification. (34 slides) (slides)

Plan du cours

Source: course of François Fleuret at EPFL https://fleuret.org/dlc/ (cc-by-nc-sa)

Séance 2: Training deep neural networks

Back-propagation

- 3.5. – Gradient descent. (13 slides) (slides)
- 3.6. – Back-propagation. (11 slides) (slides)
- 4.1. – DAG networks. (11 slides) (slides)

Initialization, optimization and regularization

- 5.1 – Cross-entropy loss. (9 slides) (slides)
- 5.2 – Stochastic gradient descent. (17 slides) (slides)
- 5.5 – Parameter initialization. (22 slides) (slides)
- 5.6 – Architecture choice and training protocol. (9 slides) (slides)
- 6.1 – Rectifiers. (7 slides) (slides)
- 6.3 – Dropout. (12 slides) (slides)
- 6.4 – Batch normalization. (15 slides) (slides)

TD: Practical 3 - implement a MLP with one hidden layer

Plan du cours

Source: course of François Fleuret at EPFL https://fleuret.org/dlc/ (cc-by-nc-sa)

Séance 3: Pytorch practicals

Pytorch NN implementation

- 4.2 – Autograd. (9 slides) (slides)
- 4.3 – PyTorch modules and batch processing. (5 slides) (slides)
- 4.6 – Writing a PyTorch module. (10 slides) (slides)
- 5.3 – PyTorch optimizers. (8 slides) (slides)
- 5.7 – Writing an autograd function. (7 slides) (slides)
- 6.6 – Using GPUs. (19 slides) (slides)

TP: Practical 4 - Implement a convolutional network