Rotterdam School of Management
Erasmus University

RSM Erasmus

Deep Learning Lecture

Guest Lecture Turing Students Rotterdam 2022-04-04

RSM - a force for positive change

Introduction

- A bit about me

  Assistant Professor of Marketing, RSM

  Scientific Advisor Schwarz Group (German retail group, Lidl, Kaufland)

  Before: Founder of AI startups

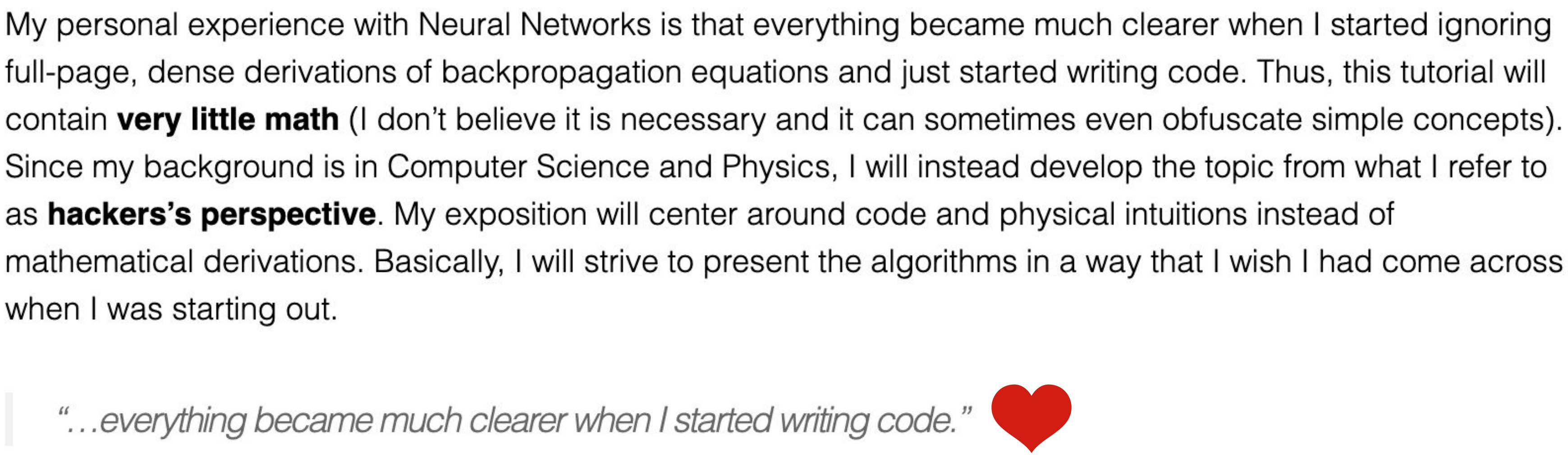
  Research: Deep learning applications in marketing

  sebastiangabel.com

- GitHub repository for this lecture: https://github.com/sbstn-gbl/dl-lecture
- Other courses: Learning from Big Data (Bachelor Minor)

Lecture Philosophy

- Hacker's guide to Neural Networks (Karpathy)



My personal experience with Neural Networks is that everything became much clearer when I started ignoring full-page, dense derivations of backpropagation equations and just started writing code. Thus, this tutorial will contain **very little math** (I don't believe it is necessary and it can sometimes even obfuscate simple concepts). Since my background is in Computer Science and Physics, I will instead develop the topic from what I refer to as **hackers's perspective**. My exposition will center around code and physical intuitions instead of mathematical derivations. Basically, I will strive to present the algorithms in a way that I wish I had come across when I was starting out.

> *"…everything became much clearer when I started writing code."*

❤

- ***Combine*** mathematical concepts with code
  - Understand what you do (i.e., study theory)
  - Build it yourself ("Don't use a model you have not implemented from scratch yourself")
  - Let more skilled developers build the software that you use (e.g., PyTorch)

Learning Objectives

- How do neural networks work?
  - Notebook 1: Gradient Descent
  - Notebook 2: Backpropagation
- How do I implement a neural network in PyTorch?
  - Notebook 3: Spiral
  - Notebook 4: Tensorboard
- How do I make sure that my neural network learns well?
  - Notebook 5: Weight initialization

Lessons Learned

- How do neural networks work?

  Notebook 1: Gradient Descent, Notebook 2: Backpropagation

  → Understanding backpropagation and gradient descent is an important foundation.

- How do I implement a neural network in PyTorch?

  Notebook 3: Spiral, Notebook 4: Tensorboard

  → We only scratched the surface today, and there are much better ways to implement NNs in PyTorch. Check out additional frameworks (e.g., PyTorch Lightning).

- How do I make sure that my neural network learns well?

  Notebook 5: Weight initialization

  → We discussed one topic today. Read additional papers and books, but most the important thing is still to implement models. And: Don't just execute code you find on the internet (everyone can do that), implement the models yourself.

Some Literature Recommendations

- Deep Learning Book (Goodfellow, Bengio, Courville)
- Deep learning (LeCun, Bengio, Hinton)
- Deep Neural Networks for YouTube Recommendations
- Neural Networks and Deep Learning
- GitHub: ML-From-Scratch
- Hacker's guide to Neural Networks
- Random Search for Hyper-Parameter Optimization
- Stanford CS230 Deep Learning and CS231n: Deep Learning for Computer Vision