Neural Networks Hello World + Assignments 2, 3

(Neural Networks Implementation and Application Tutorial)

Vilém Zouhar, Noon Pokaratsiri Goldstein

24th, 25th November 2021

Overview

- Organization - Bonus points
- Assignment 2
- Gradient
- PyTorch's Autograd
- NN Hello World
- Assignment 3

Organization - Bonus points

- 2 points for solution presentation
- 1-2 points for session contribution
- Max 2 points per session

Assignment 2

- *Tutor cue:* go through the assignment
- Typesetting:
  - $testaccuracy = 0.5$ or $\text{test accuracy} = 0.5$?
    - `$\text{test accuracy} = 0.5$`
  - The $*$ symbol is used for convolution. Use `\cdot` instead.
    - $a * b$, $a \cdot b$
- Cheating
  - Discussion vs. blatant copying
- What were the biggest issues? Coding or theory?
- Do you feel they are too easy/hard?
- Do you feel they are unrelated to the lecture content?

Optimization

Gradient 🤔

- What is it?

Optimization

Gradient 🤔

- What is it?
- How do we denote it?

Optimization

Gradient 🤔

- What is it?
- How do we denote it?
  - $\nabla f(p) = [\frac{\delta f}{\delta x_1}(p), \ldots, \frac{\delta f}{\delta x_k}(p)]$

Optimization

Gradient 🤔

- What is it?
- How do we denote it?
  - $\nabla f(p) = [\frac{\delta f}{\delta x_1}(p), \ldots, \frac{\delta f}{\delta x_k}(p)]$
- Why is it important?

Optimization

Gradient 🤔

- What is it?
- How do we denote it?
  - $\nabla f(p) = [\frac{\delta f}{\delta x_1}(p), \ldots, \frac{\delta f}{\delta x_k}(p)]$
- Why is it important?
  - Optimalization

Optimization

Gradient 🤔

- What is it?
- How do we denote it?
  - $\nabla f(p) = [\frac{\delta f}{\delta x_1}(p), \dots, \frac{\delta f}{\delta x_k}(p)]$
- Why is it important?
  - Optimalization

Optimization

Gradient 🤔

- What is it?
- How do we denote it?
  - $\nabla f(p) = [\frac{\delta f}{\delta x_1}(p), \ldots, \frac{\delta f}{\delta x_k}(p)]$
- Why is it important?
  - Optimalization

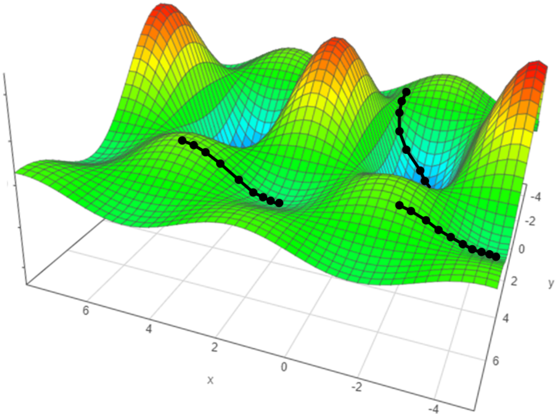

Figure 1: Function parameter landscape from [1]

Optimization

Few questions 🧐

- How does step/gradient-based optimization work?

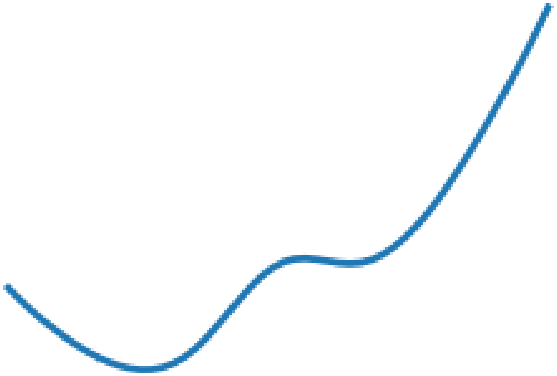

Figure 2: Function parameter landscape from [2]

Optimization

Few questions 🧐

- How does step/gradient-based optimization work?
- How is the step size determined?

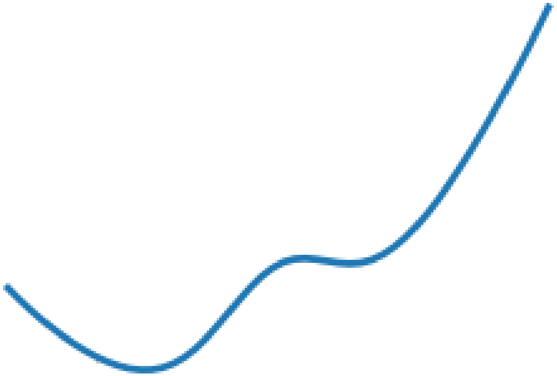
Figure 2: Function parameter landscape from [2]

Optimization

Few questions 🧐

- How does step/gradient-based optimization work?
- How is the step size determined?
- Why do we subtract the gradient and not add it?

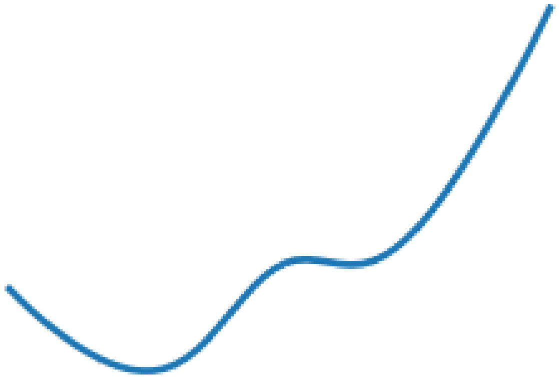

Figure 2: Function parameter landscape from [2]

Optimization

Few questions 🧐

- How does step/gradient-based optimization work?
- How is the step size determined?
- Why do we subtract the gradient and not add it?
- If we start in different places will we always find the same spot?

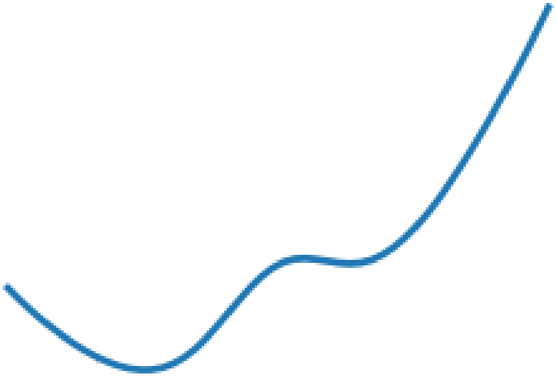

Figure 2: Function parameter landscape from [2]

Optimization

Few questions 🧐

- How does step/gradient-based optimization work?
- How is the step size determined?
- Why do we subtract the gradient and not add it?
- If we start in different places will we always find the same spot?
- Will we always find the global minimum?

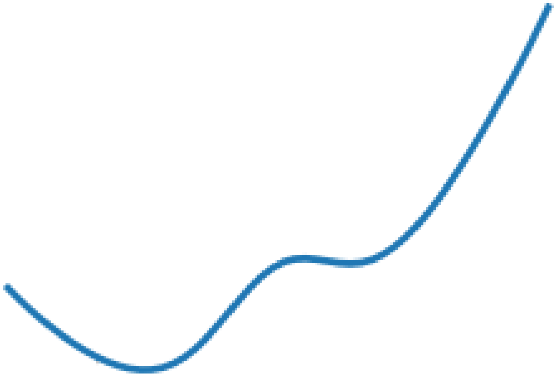

Figure 2: Function parameter landscape from [2]

Autograd & PyTorch

How to get the gradient at $(x, y) = (2, 3)$ of $x \cdot y + \sin(\pi \cdot x)$?

- By hand 😩

Autograd & PyTorch

How to get the gradient at $(x, y) = (2, 3)$ of $x \cdot y + \sin(\pi \cdot x)$?

- By hand 😩
- Autograd 🤩

Autograd & PyTorch

How to get the gradient at $(x, y) = (2, 3)$ of $x \cdot y + \sin(\pi \cdot x)$?

- By hand 😩
- Autograd 🤩

Autograd & PyTorch

How to get the gradient at $(x, y) = (2, 3)$ of $x \cdot y + \sin(\pi \cdot x)$?

- By hand 😩
- Autograd 🤩

By hand

$$\frac{\delta}{\delta x} = y + \pi \cdot \cos(\pi \cdot x) \rightarrow 3 + \pi$$
$$\frac{\delta}{\delta y} = x \rightarrow 2$$
$$\nabla f(2, 3) \rightarrow (3 + \pi, 2)$$

Autograd & PyTorch

How to get the gradient at $(x, y) = (2, 3)$ of $x \cdot y + \sin(\pi \cdot x)$?

- By hand 😩
- Autograd 🤩

By hand

$$\frac{\delta}{\delta x} = y + \pi \cdot \cos(\pi \cdot x) \rightarrow 3 + \pi$$
$$\frac{\delta}{\delta y} = x \rightarrow 2$$
$$\nabla f(2, 3) \rightarrow (3 + \pi, 2)$$

Autograd

```
import torch
import numpy as np
x = torch.tensor(2.0, requires_grad=True)
y = torch.tensor(3.0, requires_grad=True)
out = x*y + torch.sin(np.pi*x)
out.backward() # trigger gradient computation
assert np.isclose(x.grad, 3+np.pi)
assert np.isclose(y.grad, 2)
```

Assignment 3

- Any questions?

Resources

- [1] Optimization & landscapes offconvex.org/2018/11/07/optimization-beyond-landscape/
- [2] Optimization Introduction by Scipy scipy-lectures.org/advanced/mathematical_optimization/