

# CPSC 340 – Tutorial 1

Lironne Kurzman lironnek@cs.ubc.ca

University of British Columbia

September 13th, 2021

### GITHUB - tl;dr

```
Clone a repo:
    git clone <repo url>
See changes:
    git status
    git diff <file name>
Commit changes:
    git add <file name>
    git commit -m "<commit message>" git push
Pull remote changes:
    git pull
```

### **Notation and Gradients**

- o Define:  $R^n$ ,  $x^T A x$ , ||x||,  $\nabla f(x)$
- If  $x \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^m$ ,  $A \in \mathbb{R}^{n \times m}$  find the dimensions of:

$$\circ X^T A$$
,  $X^T A Y$ ,  $X X^T$ .

○ If  $a, x ∈ \mathbb{R}^n$ , compute the gradients of:

$$f(x) = a^T x,$$
  
 $f(x) = \log a^T x.$ 

A helpful sanity check:

If x is a length-n (column) vector, and  $f: \mathbb{R}^n \to \mathbb{R}$  then  $\nabla f(x)$  is a length-n (row) vector.

o Entropy: Given a probability vector  $p = [p_1, ..., p_n]^T$ ,

$$entropy(p) = - \underset{i=1}{n} \sum p_i \log(p_i).$$

### Python/NumPy Basics

o numpy Arrays:

$$A = np. array(...)$$

- Array slicing
- Element-wise operations:

- Array shapes
- o Matrix/vector operations:

Other useful things:

```
np. ones, np. zeros
```

```
mirror_object
                                   ror object to mirror
        peration == "MIRROR_X/:
    mirror_mod.use_x = Tryle
   mirror_mod.use_y = F lse
_operation == "MIRROR Y"
   lrror_mod.use_x = False
lrror_mod.use_y = True
     rror_mod.use_x = valse
               rror_mod.use_y = False
               _rror_mod.use_z = True
              election at the end -add
                        ob.select= 1
                   er ob.select=1
            "Selected" + str(modification of the structure of the str
               wint("please select exact)
                 -- OPERATOR CLASSES
```

### **Python Demo**

**Numpy Basics** 

```
x mirror to the selected
ject.mirror_mirror_x"
ror X"
```

## Bonus: We can check that the gradient implementation is correct

Finite-difference approximation of the gradient

$$\epsilon = 1e - 4$$

Gradient definion:

$$\frac{\partial f(x)}{\partial x_i} = \lim_{\epsilon \to 0} \frac{f(x_i + \epsilon) - f(x_i)}{\epsilon}$$

Therefore, 
$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x_i + \epsilon) - f(x_i)}{\epsilon}$$

### **Bonus: Code to approximate the gradient**

$$\frac{\partial f(x)}{\partial x_i} \approx \frac{f(x_i + \epsilon) - f(x_i)}{\epsilon}$$

### Create a simple gradient check for single-variable function

```
def forward_diff(x, f):
    # Approximates gradient eps = 1e-4
    g_approx = f(x + eps) - f(x) g_approx /= eps
```

### **Bonus: Code to approximate the gradient**

### Using scipy:

```
import numpy as np
from scipy.optimize
import approx_fprime

approx = approx_fprime(x0, foo, 1e-4)
exact = foo_grad(x0)

print("My gradient : %s" % exact)
print("Scipy's gradient: %s" % approx)

# Assert that the two are almost equal
np. testing.assert almost equal(approx, exact, 3)
```