Exercise 2

Deep Learning Lab

September 23, 2021

1 PyTorch Basics

The objective of this exercise is to get familiar with standard operations in PyTorch.

1.1 Tensor Basics

Make sure that you can:

- Create tensors and inspect their shape and data type.
- Create random tensors with a shape you specify.
- Perform element-wise arithmetic operations between tensors.
- Perform arithmetic operations between tensors and scalars.
- Perform matrix multiplications.
- Perform unary operations on tensors (e.g., `max`, `sum`) along different axes.
- Apply functions element-wise to a tensor (e.g., `exp`).
- Slice tensors, index using lists of elements, and index using Boolean arrays.
- Use build-in constructors to generate standard tensors (`zeros`, `eyes`, ...).
- Use `fill_`.
- Transpose tensors and permute their dimensions.
- Reshape tensors. Reshape using -1 as an index.
- Concatenate tensors. Split tensors.
- Implement a bit more complex operations using multiple basic operations; e.g. write a function which, given two tensors A and B, returns `True` if A and B are equal or if all elements of A and B are equal except where B's element is 1.
- Construct your own example to test the function above.

1.2 Gradient Descent by Hand

1. Consider the function f given by

$$f(x, y) = \frac{x^2}{2} + 2y^2.$$

   Using Matplotlib, it is possible to create a contour plot for f:

Listing 1: Contour plot using Matplotlib.

```
import matplotlib.pyplot as plt

def f(x, y):
    return ((x**2) / 2.) + (2 * y**2)

def create_contour_plot(low=-1000, high=1000, points=50):
    f_range = np.linspace(low, high, points)

    X, Y = np.meshgrid(f_range, f_range)
    Z = f(X, Y)

    plt.contour(X, Y, Z, colors='b')
```

It is possible to visualize the result using `plt.show`, or even draw on top of this contour plot (for instance, using `plt.plot`).

(a) Use 20 iterations of gradient descent with a learning rate of 0.1 to find the global minimum of f. Initialize both x and y to -1000. Hint: represent x and y by a two-dimensional vector (x, y), and express f using a dot product.

(b) Use Matplotlib to plot the optimization trajectory (sequence of coordinates) on top of the contour plot for f. Did the optimization procedure succeed? What happens if the learning rate is too low/high? Use the marker style 'r.-' with `plt.plot`.