Activity 2

Deep Learning Lab

September 27, 2019

1 Exercises

1. Execute each of the examples presented during the lecture (Slides 21, 22, 23, 25, and 28-29).

2. 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.

```
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) Adapt the snippet in Slide 25 to find the global minimum of f. Use 20 iterations of gradient descent with a learning rate of 0.1. Initialize both x and y to -1000. Tip: represent x and y by a two-dimensional vector (x, y).

   (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? Tip: use the marker style 'r.-' with plt.plot.

3. Consider the task of minimizing a function f. In momentum-based gradient descent, each parameter x has a corresponding velocity v. The initial velocity v_0 is given by $v_0 = 0$, and

$$v_{t+1} = \mu v_t - \eta f'(x_t),$$

where $0 \leq \mu \leq 1$ is the so-called momentum coefficient, η is the learning rate, and f' is the derivative of f with respect to x. Each parameter is updated according to $x_{t+1} = x_t + v_{t+1}$.

Adapt your solution to the previous exercise to use momentum-based gradient descent. Use $\mu = 0.5$. What happens if the momentum coefficient μ is too large? What happens if the momentum coefficient is zero? Tip: use an additional tf.Variable to store the velocities, and update velocities and parameters separately.