**MAT235 Tutorial 5**

TA: Jace Alloway (j.alloway@mail.utoronto.ca)
Office Hours: M14-16 PG101

This week: Method of gradient descent. Refer 14.4 Hughes-Hallett.

**Gradient Descent:** Consider a real-valued function $f : \mathbb{R}^n \to \mathbb{R}$ with some local minimum on a subset of it's domain. Suppose we want to approximate the location of the minimum numerically ($f$ may be an unknown function we want to minimize). Recall that the gradient of a function $f$ is a vector field which, at every point in space, points in the direction of maximal increase of $f$. This means that $\nabla f$ is a vector field which always points towards the maximum (maxima) of $f$. If we invert the sign of the field, $-\nabla f$ will then always point towards the minima of $f$; away from the maxima.

Suppose we choose a point $\mathbf{x_0} = (x_0, y_0, z_0, \dots)$ in the domain of $f$ close to a minimum (assuming a minimum exists). Then, at $\mathbf{x_0}$, the vector $-\nabla f(\mathbf{x_0})$ points towards said minimum. We can then translate $\mathbf{x_0}$ to a *new* point $\mathbf{x_1}$, *closer* to the minimum, by adding $\mathbf{x_0} - \nabla f(\mathbf{x_0})$,

$$\mathbf{x_1} = \mathbf{x_0} - \nabla f(\mathbf{x_0}). \tag{1}$$

The point $\mathbf{x_1}$ is now a new point closer to the minimum of $f$: which means we can repeat this process to obtain a second point $\mathbf{x_2}$ even closer to the minimum than $\mathbf{x_1}$ and $\mathbf{x_0}$:

$$\mathbf{x_2} = \mathbf{x_1} - \nabla f(\mathbf{x_1}). \tag{2}$$

We can continue to take new points $\mathbf{x_3}, \mathbf{x_4}, \dots$ until we get closer and closer to the minimum of $f$. Each of the new points is obtained by taking iterations, since this is an iterable process. The next point is a function of the previous point(s). That is, for every point $\mathbf{x_i}$, the next point closer to the minimum is given by

$$\mathbf{x_{i+1}} = \mathbf{x_i} - \nabla f(\mathbf{x_i}). \tag{3}$$

This however, it not entirely useful until we add a scaling parameter $\eta$ (called the 'learning rate') to our translating vector $-\nabla f(\mathbf{x_i})$, since this vector may be arbitrary in magnitude, causing the result to actually *diverge* instead of converge. We will investigate this further in the practice problems. Hence

$$\boxed{\mathbf{x_{i+1}} = \mathbf{x_i} - \eta \nabla f(\mathbf{x_i}), \quad \eta \in \mathbb{R}^+ \quad \text{(descent)}} \tag{4}$$

is called the *method of gradient descent*: an iterable process to numerically approximate the locations of minima of functions. One may also define the *method of gradient ascent*, to approximate the locations of maxima, just be inverting the sign of the vector.

$$\boxed{\mathbf{x_{i+1}} = \mathbf{x_i} + \eta \nabla f(\mathbf{x_i}), \quad \eta \in \mathbb{R}^+ \quad \text{(ascent)}} \tag{5}$$

Considering only gradient descent for now, we find that the method will produce a sequence of points $\{\mathbf{x_0}, \mathbf{x_1}, \mathbf{x_2}, \dots\} \subseteq \mathbb{R}^n$ such that, given sufficiently small $\eta$,

$$f(\mathbf{x_0}) \geq f(\mathbf{x_1}) \geq f(\mathbf{x_2}) \geq \dots, \tag{6}$$

that is, the values of $f(\mathbf{x_i})$ decrease (approaching a minimum). In practice, the input dimensionality $n$ is very large, and $f$ is a function which computes "loss" or the error of a statistical model (which we aim to minimize).

**Example:** Consider the function $f : \mathbb{R}^2 \to \mathbb{R}$ defined as $f(x, y) = x^2 + y^2$. The gradient of this function is $\nabla f(x, y) = \langle 2x, 2y\rangle$. Taking $\eta = 0.1$ and $\mathbf{x_0} = (1, 1)$, let us approximate the minimum of $f$ numerically for 3 iterations:

$$
\begin{aligned}
\mathbf{x_1} &= \langle 1, 1\rangle - (0.1)\nabla f(1, 1) && (7)\\
&= \langle 1, 1\rangle - \langle 0.2, 0.2\rangle && (8)\\
&= \langle 0.8, 0.8\rangle && (9)\\
\mathbf{x_2} &= \langle 0.8, 0.8\rangle - (0.1)\nabla f(0.8, 0.8) && (10)\\
&= \langle 0.8, 0.8\rangle - \langle 0.16, 0.16\rangle && (11)\\
&= \langle 0.64, 0.64\rangle && (12)\\
\mathbf{x_3} &= \langle 0.64, 0.64\rangle - (0.1)\nabla f(0.64, 0.64) && (13)\\
&= \langle 0.64, 0.64\rangle - \langle 0.128, 0.128\rangle && (14)\\
&= \langle 0.512, 0.512\rangle. && (15)
\end{aligned}
$$

With each iteration, notice that the point approaches $(0, 0)$, the actual minimum. One may continue to compute manually, but for a large number of iterations it is better to use code.

```
n = 0   [1 1]
n = 1   [0.8 0.8]
n = 2   [0.64 0.64]
n = 3   [0.512 0.512]
n = 4   [0.4096 0.4096]
        ...
n = 38  [0.00020769 0.00020769]
n = 39  [0.00016615 0.00016615]
```

**Problems:** Let $f : \mathbb{R}^2 \to \mathbb{R}$ be defined as $f(x, y) = x^2 + y^2 + 2$.

(a) Apple the method of gradient descent for 4 iterations to find $\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}$. Start with $\mathbf{x_0} = \langle 1, -1 \rangle$ with a learning rate of $\eta = 0.25$.

(b) Verify that $f(\mathbf{x_0}) \geq f(\mathbf{x_1}) \geq f(\mathbf{x_2}) \geq f(\mathbf{x_3}) \geq f(\mathbf{x_4})$. Does the sequence seem to converge to a particular value? What value should it converge to?

(c) Repeat parts (a) and (b) but with $\eta = 1$. What do you notice?

(d) Repeat parts (a) and (b) but with $\mathbf{x_0} = \langle 10, -10 \rangle$. What do you notice?