# EE2211 Tutorial 8

## Question 1

Suppose we are minimizing $f(x) = x^4$ with respect to $x$. We initialize $x$ to be 2. We perform gradient descent with learning rate 0.1. What is the value of $x$ after the first iteration?

**Answer:**

- The gradient of $f(x)$ is $4x^3$.
- At $x = 2$, the gradient is $4 \times 2^3 = 32$
- After first iteration of gradient descent, value of $x$ will be $x = 2 - 0.1 \times 32 = -1.2$

## Question 2

Please consider the csv file (government-expenditure-on-education.csv), which depicts the government's educational expenditure over the years. We would like to predict expenditure as a function of year. To do this, fit an exponential model $f(\mathbf{x}, \mathbf{w}) = \exp(-\mathbf{x}^T\mathbf{w})$ with squared error loss to estimate $\mathbf{w}$ based on the csv file and gradient descent. In other words, $C(\mathbf{w}) = \sum_{i=1}^{m}(f(\mathbf{x}_i, \mathbf{w}) - y_i)^2$.

Note that even though year is one dimensional, we should add the bias term, so $\mathbf{x} = [1 \ \text{year}]^T$. Furthermore, optimizing the exponential function is tricky (because a small change in $\mathbf{w}$ can lead to large change in $f$). Therefore for the purpose of optimization, divide the "year" variable by the largest year (2018) and divide the "expenditure" by the largest expenditure, so that the resulting normalized year and normalized expenditure variables are between 0 and 1. Use a learning rate of 0.03 and run gradient descent for 2000000 iterations.

(a) Plot the cost function $C(\mathbf{w})$ as a function of the number of iterations.

(b) Use the fitted parameters to plot the predicted educational expenditure from year 1981 to year 2023.

(c) Repeat (a) using a learning rate of 0.1 and learning rate of 0.001. What do you observe relative to (a)?

The goal of this question is for you to code up gradient descent, so I will provide you with the gradient derivation. First, please note that in general, $\nabla_{\mathbf{w}}(\mathbf{x}^T\mathbf{w}) = \mathbf{x}$. To see this:

$$\nabla_{\mathbf{w}}(\mathbf{x}^T\mathbf{w}) = \begin{bmatrix} \frac{\partial(\mathbf{x}^T\mathbf{w})}{\partial w_1} \\ \frac{\partial(\mathbf{x}^T\mathbf{w})}{\partial w_2} \\ \vdots \\ \frac{\partial(\mathbf{x}^T\mathbf{w})}{\partial w_d} \end{bmatrix} = \begin{bmatrix} \frac{\partial(w_1x_1+w_2x_2+\cdots+w_dx_d)}{\partial w_1} \\ \frac{\partial(w_1x_1+w_2x_2+\cdots+w_dx_d)}{\partial w_2} \\ \vdots \\ \frac{\partial(w_1x_1+w_2x_2+\cdots+w_dx_d)}{\partial w_d} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} = \mathbf{x} \tag{1}$$

The above equality will be very useful for the other questions as well. Now, going back to our question,

$$\begin{aligned}
\nabla_{\mathbf{w}} C(\mathbf{w}) &= \nabla_{\mathbf{w}} \sum_{i=1}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^2 && (2)\\
&= \sum_{i=1}^{m} \nabla_{\mathbf{w}} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^2 && (3)\\
&= \sum_{i=1}^{m} 2(f(\mathbf{x}_i, \mathbf{w}) - y_i) \nabla_{\mathbf{w}} f(\mathbf{x}_i, \mathbf{w}) \quad \text{chain rule} && (4)\\
&= \sum_{i=1}^{m} 2(f(\mathbf{x}_i, \mathbf{w}) - y_i) \nabla_{\mathbf{w}} \exp(-\mathbf{x}_i^T\mathbf{w}) && (5)\\
&= -\sum_{i=1}^{m} 2(f(\mathbf{x}_i, \mathbf{w}) - y_i) \exp(-\mathbf{x}_i^T\mathbf{w}) \nabla_{\mathbf{w}} (\mathbf{x}_i^T\mathbf{w}) \quad \text{chain rule} && (6)\\
&= -\sum_{i=1}^{m} 2(f(\mathbf{x}_i, \mathbf{w}) - y_i) \exp(-\mathbf{x}_i^T\mathbf{w}) \mathbf{x}_i && (7)\\
&= -\sum_{i=1}^{m} 2(f(\mathbf{x}_i, \mathbf{w}) - y_i) f(\mathbf{x}_i, \mathbf{w}) \mathbf{x}_i && (8)
\end{aligned}$$

**Answer:**

Please see code Tut8_yeo.py.

(a) See Figure 1 below. The cost function decreases rapidly at first and then converges to a final value.

(b) See Figure 2 below.

(c) See Figures 3 and 4 below. A learning rate of 0.1 is too big, so the cost function does not decrease monotonically with increasing iterations, but instead fluctuate a lot without convergence. The final cost function value is much worse that (a). On the other hand, a learning rate of 0.001 is too small. So even though the cost function decreases monotonically with increasing iterations, gradient descent has not converged even after 2000000 iterations. The final cost function value is much worse that (a).

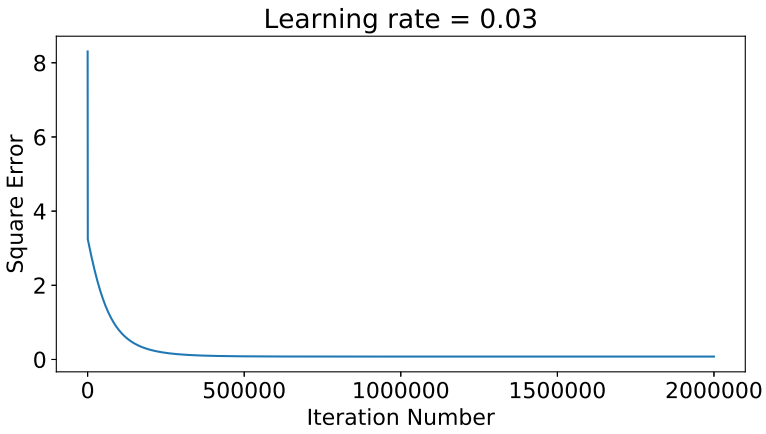


Figure 1: Cost function value as a function of iterations.

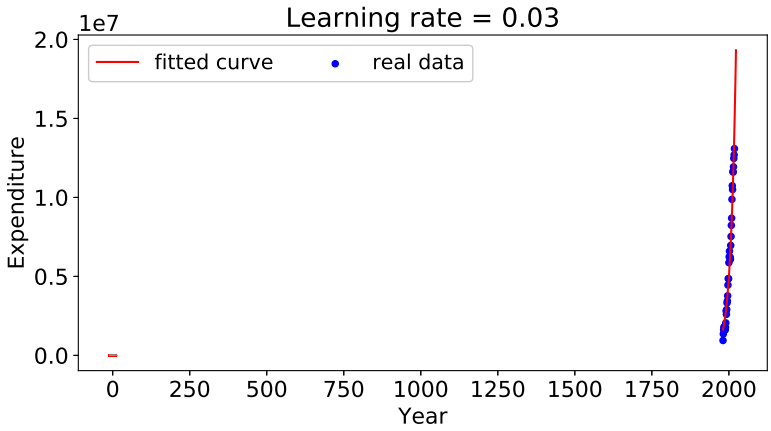


Figure 2: Fitted curve from 1981 to 2023

## Question 3

Given the linear learning model $f(\mathbf{x}, \mathbf{w}) = \mathbf{x}^T\mathbf{w}$, where $\mathbf{x} \in \mathbb{R}^d$. Consider the loss function $L(f(\mathbf{x}_i, \mathbf{w}), y_i) = (f(\mathbf{x}_i, \mathbf{w}) - y_i)^4$, where $i$ indexes the $i$-th training sample. The final cost function is $C(\mathbf{w}) = \sum_{i=1}^{m} L(f(\mathbf{x}_i, \mathbf{w}), y_i)$, where $m$ is the total number of training samples. Derive the gradient of the cost function with respect to $\mathbf{w}$.

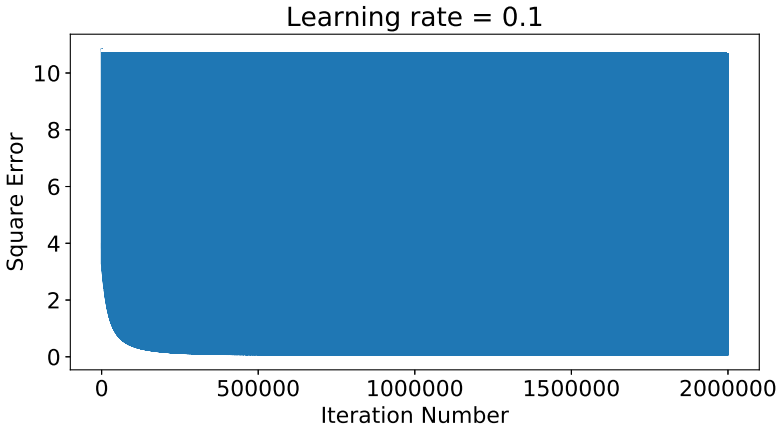


Figure 3: Cost function value as a function of iterations.

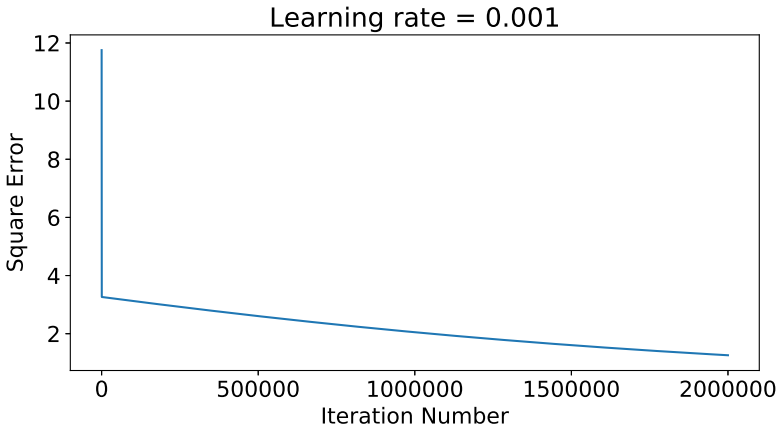


Figure 4: Cost function value as a function of iterations.

**Answer:**

$$\nabla_{\mathbf{w}} C(\mathbf{w}) = \nabla_{\mathbf{w}} \sum_{i=1}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^4 \tag{9}$$

$$= \sum_{i=1}^{m} \nabla_{\mathbf{w}} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^4 \tag{10}$$

$$= \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \nabla_{\mathbf{w}} f(\mathbf{x}_i, \mathbf{w}) \quad \text{chain rule} \tag{11}$$

$$= \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \nabla_{\mathbf{w}} (\mathbf{x}_i^T \mathbf{w}) \tag{12}$$

$$= \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \mathbf{x}_i \tag{13}$$

## Question 4

Repeat Question 3 using $f(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{x}^T\mathbf{w})$, where $\sigma(a) = \frac{1}{1+\exp(-\beta a)}$

**Answer:**

$$\nabla_{\mathbf{w}} C(\mathbf{w}) = \nabla_{\mathbf{w}} \sum_{i=1}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^4 \tag{14}$$

$$= \sum_{i=1}^{m} \nabla_{\mathbf{w}} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^4 \tag{15}$$

$$= \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \nabla_{\mathbf{w}} f(\mathbf{x}_i, \mathbf{w}) \quad \text{chain rule} \tag{16}$$

$$= \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \nabla_{\mathbf{w}} \sigma(\mathbf{x}_i^T\mathbf{w}) \tag{17}$$

$$= \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \frac{\partial \sigma(a)}{\partial a} \nabla_{\mathbf{w}} (\mathbf{x}_i^T\mathbf{w}) \quad \text{chain rule} \tag{18}$$

$$= \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \frac{\partial \sigma(a)}{\partial a} \mathbf{x}_i \tag{19}$$

So we just have to evaluate $\frac{\partial \sigma(a)}{\partial a}$ and plug it into the above equation. Note that $\frac{\partial \sigma(a)}{\partial a}$ is evaluated at $a = \mathbf{x}_i^T\mathbf{w}$, so

$$\frac{\partial \sigma(a)}{\partial a} = \frac{\partial}{\partial a} \left( \frac{1}{1 + \exp(-\beta a)} \right) \tag{20}$$

$$= -\frac{1}{(1 + e^{-\beta a})^2} \frac{\partial (1 + e^{-\beta a})}{\partial a} \tag{21}$$

$$= \frac{\beta}{(1 + e^{-\beta a})^2} e^{-\beta a} \tag{22}$$

$$= \frac{\beta}{(1 + e^{-\beta a})^2} (1 + e^{-\beta a} - 1) \tag{23}$$

$$= \beta \left( \frac{1}{1 + e^{-\beta a}} - \frac{1}{(1 + e^{-\beta a})^2} \right) \tag{24}$$

$$= \beta \left( \sigma(a) - \sigma^2(a) \right) \tag{25}$$

$$= \beta \sigma(a)(1 - \sigma(a)) \tag{26}$$

$$= \beta \sigma(\mathbf{x}_i^T\mathbf{w})(1 - \sigma(\mathbf{x}_i^T\mathbf{w})) \tag{27}$$

Therefore,

$$\nabla_{\mathbf{w}} C(\mathbf{w}) = \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \beta \sigma(\mathbf{x}_i^T\mathbf{w})(1 - \sigma(\mathbf{x}_i^T\mathbf{w}))\mathbf{x}_i \tag{28}$$

## Question 5

Repeat Question 3 using $f(\mathbf{x}, \mathbf{w}) = \sigma(\mathbf{x}^T\mathbf{w})$, where $\sigma(a) = \max(0, a)$

**Answer:**

$$\nabla_{\mathbf{w}} C(\mathbf{w}) = \nabla_{\mathbf{w}} \sum_{i=1}^{m} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^4 \tag{29}$$

$$= \sum_{i=1}^{m} \nabla_{\mathbf{w}} (f(\mathbf{x}_i, \mathbf{w}) - y_i)^4 \tag{30}$$

$$= \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \nabla_{\mathbf{w}} f(\mathbf{x}_i, \mathbf{w}) \quad \text{chain rule} \tag{31}$$

$$= \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \nabla_{\mathbf{w}} \sigma(\mathbf{x}_i^T\mathbf{w}) \tag{32}$$

$$= \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \frac{\partial \sigma(a)}{\partial a} \nabla_{\mathbf{w}} (\mathbf{x}_i^T\mathbf{w}) \quad \text{chain rule} \tag{33}$$

$$= \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \frac{\partial \sigma(a)}{\partial a} \mathbf{x}_i \tag{34}$$

So we just have to evaluate $\frac{\partial \sigma(a)}{\partial a}$ and plug it into the above equation. Note that $\frac{\partial \sigma(a)}{\partial a}$ is evaluated at $a = \mathbf{x}_i^T\mathbf{w}$. When $a < 0$, $\sigma(a) = 0$, so $\frac{\partial \sigma(a)}{\partial a} = 0$. When $a > 0$, $\sigma(a) = a$, so $\frac{\partial \sigma(a)}{\partial a} = 1$. Let us define $\delta(\mathbf{x}_i^T\mathbf{w} > 0) = \begin{cases} 1 & \text{if } \mathbf{x}_i^T\mathbf{w} > 0 \\ 0 & \text{if } \mathbf{x}_i^T\mathbf{w} < 0 \end{cases}$, so we get

$$\frac{\partial \sigma(a)}{\partial a} = \delta(\mathbf{x}_i^T\mathbf{w} > 0) \tag{35}$$

Therefore, we get

$$\nabla_{\mathbf{w}} C(\mathbf{w}) = \sum_{i=1}^{m} 4(f(\mathbf{x}_i, \mathbf{w}) - y_i)^3 \mathbf{x}_i \delta(\mathbf{x}_i^T\mathbf{w} > 0), \tag{36}$$