1 April 23

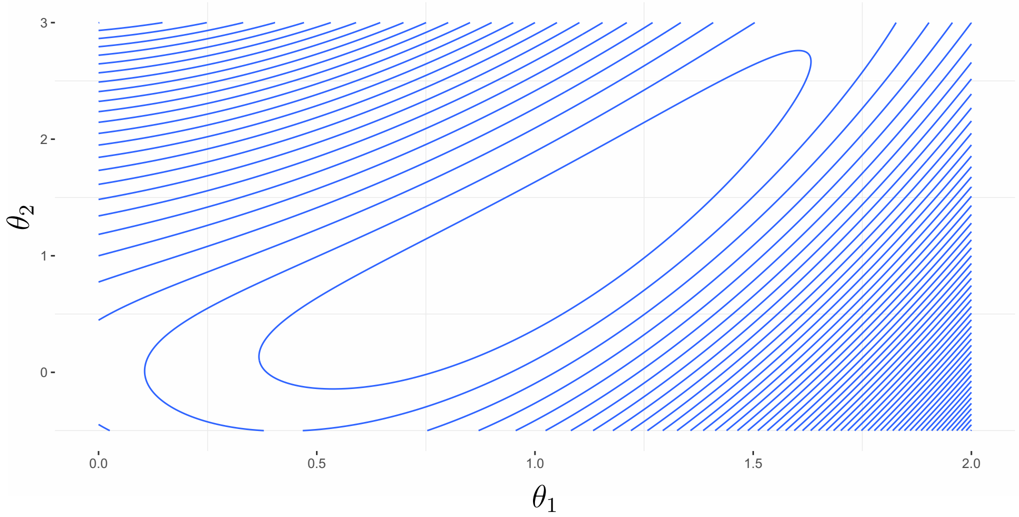
1.1 How to adjust Learning Rate

Assume the objective function,

$$\underset{\theta}{\text{minimize}} \quad f(\theta) = 0.5(\theta_1^2 - \theta_2)^2 + 0.5(\theta_1 - 1)^2 \tag{1.1}$$

where θ includes θ_1 and θ_2.

The figure for the 2D model looks like,



Next, in order to update θ, we should find the gradients for both θ_1, θ_2,

$$\frac{\partial}{\partial \theta_1} f(\theta) = 2\theta_1^3 - 2\theta_1\theta_2 + \theta_1 - 1$$
$$\frac{\partial}{\partial \theta_2} f(\theta) = -\theta_1^2 + \theta_2$$

Therefore, the update of θ takes the form,

$$\theta_1^{(n+1)} = \theta_1^{(n)} - \alpha_1 \frac{\partial}{\partial \theta_1^{(n)}} f(\theta) = \theta_1^{(n)} - \alpha_1 \big(2(\theta_1^{(n)})^3 - 2\theta_1^{(n)}\theta_2^{(n)} + \theta_1^{(n)} - 1\big)$$
$$\theta_2^{(n+1)} = \theta_2^{(n)} - \alpha_2 \frac{\partial}{\partial \theta_2^{(n)}} f(\theta) = \theta_2^{(n)} - \alpha_2 \big(- (\theta_1^{(n)})^2 + \theta_2^{(n)}\big)$$

From the definition of fixed point theory, which given a function f defined on the real numbers with real values and given a point x_0 in the domain of f, the fixed point iteration is

$$x_{n+1} = f(x_n), n = 0, 1, \ldots \tag{1.2}$$

which gives rise to the sequence $x_0, x_1, \ldots$, which is hoped to converge to a point x.

By implementing fixed point iteration in our update model, we hope $\theta^{(0)}, \theta^{(1)}, \ldots$ at any iteration (Superscript indicates iteration) will converge to a point θ

$$\theta_1^{(n+1)} = f(\theta_1^{(n)}) \to \theta_1^* = f(\theta_1^*)$$
$$\theta_2^{(n+1)} = f(\theta_2^{(n)}) \to \theta_2^* = f(\theta_2^*)$$

By applying Taylor series, we can approximate $f(\theta^*)$

$$f(\theta^*) = f'(\theta^*)|\theta^{(n)} - \theta^*|$$

We hope $0 < |f'(\theta^*)| < 1$, so $f(\theta^*)$ will converge. Following the definitions defined above, we derived the range for the learning rate α_1,

$$\frac{\partial}{\partial \theta_1^*} f(\theta_1^*) = 1 - \alpha_1 (6\big(\theta_1^*)^2 - 2\theta_2^* + 1\big)$$

Lower Bound : $0 < |f'(\theta_1^*)|$,

$$0 < 1 - \alpha_1 \big(6(\theta_1^*)^2 - 2\theta_2^* + 1\big)$$
$$\alpha_1 < \frac{1}{6(\theta_1^*)^2 - 2\theta_2^* + 1}$$

Upper Bound : $|f'(\theta_1^*)| < 1$,

$$-1 < 1 - \alpha_1\big(6(\theta_1^*)^2 - 2\theta_2^* + 1\big) < 1$$
$$0 < \alpha_1 < \frac{2}{6(\theta_1^*)^2 - 2\theta_2^* + 1}$$

α_1 boundary,

$$0 < \alpha_1 < \frac{1}{6(\theta_1^*)^2 - 2\theta_2^* + 1}$$

Following the same logic, we derive the bellow formula to extract the range for the learning rate α_2,

$$\frac{\partial}{\partial\theta_2^*} f(\theta_2^*) = 1 - \alpha_2$$

Lower Bound : $0 < |f'(\theta_2^*)|$,

$$0 < 1 - \alpha_2$$
$$\alpha_2 < 1$$

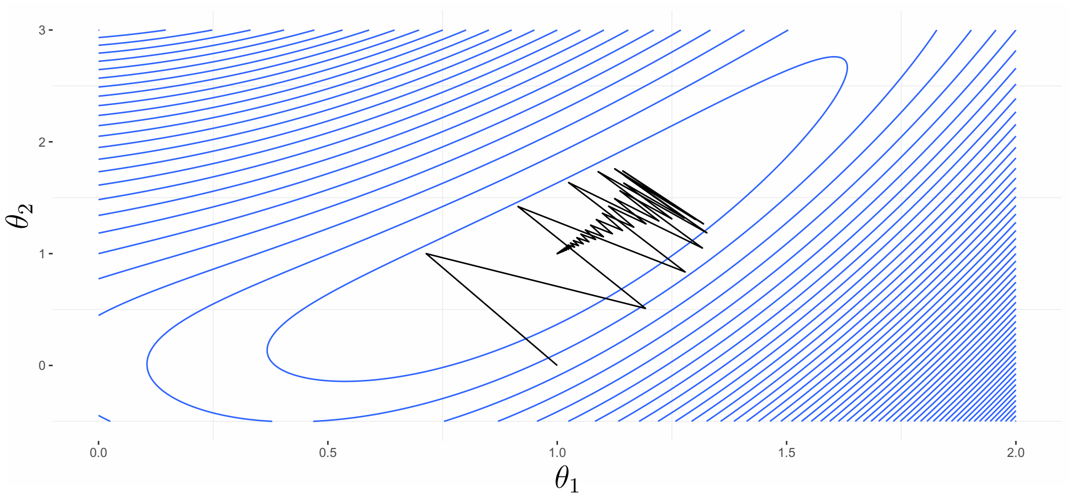
Upper Bound : $|f'(\theta_2^*)| < 1$,

$$-1 < 1 - \alpha_2 < 1$$
$$0 < \alpha_2 < 2$$

α_2 boundary,

$$0 < \alpha_1 < 1$$

Applying the learning rate $\alpha_1 = \frac{1}{6(\theta_1^*)^2 - 2\theta_2^* + 1}$ and $\alpha_2 = 1$. The searching path,



R code:

```
library(ggplot2)

mytheme = theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank())

theta1_seq <- seq(0, 2,length.out = 1000)
theta2_seq <- seq(-0.5,3,length.out = 1000)
data = expand.grid(x = theta1_seq,y = theta2_seq,KEEP.OUT.ATTRS = FALSE)


response = function(theta1,theta2){
  return(0.5*(theta1^2-theta2)^2+0.5*(theta1-1)^2)
}

data = cbind(data, response = response(data$x,data$y))

plot = ggplot(data, aes(x = x, y = y, z=response)) + stat_contour(binwidth = 0.2)+mytheme


searching = data.frame(theta1=rep(0,100),theta2=rep(0,100),response=rep(0,100))

theta1 = 0
theta2 = 0

for(i in 1:100){
  theta1_new = theta1 - ((1/((6*(theta1^{2}))-(2*theta2)+1)))*(2*theta1^{3}-2*theta1*theta2+theta1-1)
  theta2_new = theta2 - (1)*(-(theta1^2)+theta2)
  response_value = response(theta1_new,theta2_new)

  theta1 = theta1_new
  theta2 = theta2_new

  cat("Iteration:", i,"\n")
  cat("theta1: ", theta1, "\n")
  cat("theta2: ", theta2, "\n")
  cat("Response: ", response_value,"\n")
  searching[i,1] = theta1
```

```
  searching[i,2] = theta2
  searching[i,3] = response_value
}

plot + geom_path(data=searching, aes(x=theta1,y=theta2))
```