# Problem Set 3 – Loss Functions and Fitting Models

DS542 – DL4DS

Spring, 2025

**Note:** Refer to the equations in the *Understanding Deep Learning* textbook to solve the following problems.

## Problem 5.9

Consider a multivariate regression problem in which we predict the height of an individual in meters and their weight in kilos from some data $x$. Here, the units take quite different values. What problems do you see this causing? Propose two solutions to these problems.

Answer: Both height and weight have different scales and magnitudes due to their various units and ranges. For instance, height might range from 5 to 6 feet, while weight can range from 100 to 200 lbs. Even this generalization can be different depending on the country a population is from and other environmental factors. The various scales therefore will cause one of the variables like weight to dominate the model's predictions and overshadow the effect of the other variable like height. This can lead to biased and imbalanced regression coefficients. Furthermore, since the variables are on different scales, the gradient descent may struggle with numerical stability where larger numbers might cause slower convergence, leading to poor model performance.

A solution is to use **standardization** for the features by subtracting the mean and dividing by the standard deviation which addresses the scale issue by transforming both height and weight to have a mean of 0 and standard deviation of 1. Another solution is to scale the variables to a fixed range (for instance [0,1]) using **min-max scaling** which will preserve the relative distance between values while ensuring both variables are on the same scale which will improve convergence in optimization algorithms.

# Problem 6.6

Which of the functions in Figure 6.11 from the book is convex? Justify your answer. Characterize each of the points 1–7 as (i) a local minimum, (ii) the global minimum, or (iii) neither.

Answer:

1. Plot (a): This function is not convex because I can draw a straight line between points on the curve where the local maxima lie above this straight line. Point 1 is a local minimum, point 2 is a global minimum, and point 3 is a local minimum.
2. Plot (b): This function is convex because I can draw a straight line through any 2 points on the curve and the line is always above the curve. Point 4 is neither a global nor local minimum and point 5 is a global minimum.
3. Plot (c): This function is not convex because I can draw a straight line through 2 points where the curve is above the drawn straight line. Point 6 is a global minimum and point 7 is neither a local nor global minimum.

# Problem 6.10

Show that the momentum term $m_t$ (equation (6.11)) is an infinite weighted sum of the gradients at the previous iterations and derive an expression for the coefficients (weights) of that sum.

Answer:

## Momentum Term as an Infinite Weighted Sum of Gradients

Consider the momentum update rule for the momentum term $m_t$ in the algorithm:

$$m_{t+1} = \beta \cdot m_t + (1 - \beta) \sum_{i \in B_t} \frac{\partial \ell_i[\phi_t]}{\partial \phi}$$

where:

- $m_t$ is the momentum term at iteration $t$,
- $\beta$ is the momentum hyperparameter (a scalar),

- $\ell_i[\phi_t]$ is the loss function for the $i$-th data point at time $t$,

- $\frac{\partial \ell_i[\phi_t]}{\partial \phi}$ is the gradient of the loss with respect to the parameters $\phi_t$,

- $B_t$ is the set of data points used at iteration $t$.

The parameter update rule is:

$$\phi_{t+1} = \phi_t - \alpha \cdot m_{t+1}$$

where $\alpha$ is the learning rate.

We want to express the momentum term $m_t$ as an infinite weighted sum of gradients.

## Unfolding the Recursion

Start by expanding the recursion for $m_{t+1}$:

$$m_{t+1} = \beta \cdot m_t + (1-\beta) \sum_{i \in B_t} \frac{\partial \ell_i[\phi_t]}{\partial \phi}$$

Next, expand $m_t$ recursively using the same formula:

$$m_t = \beta \cdot m_{t-1} + (1-\beta) \sum_{i \in B_{t-1}} \frac{\partial \ell_i[\phi_{t-1}]}{\partial \phi}$$

Substituting this into the expression for $m_{t+1}$:

$$m_{t+1} = \beta \left( \beta \cdot m_{t-1} + (1-\beta) \sum_{i \in B_{t-1}} \frac{\partial \ell_i[\phi_{t-1}]}{\partial \phi} \right) + (1-\beta) \sum_{i \in B_t} \frac{\partial \ell_i[\phi_t]}{\partial \phi}$$

This process continues recursively, so at the $k$-th iteration back, we have:

$$m_{t+1} = \beta^k m_{t-k} + (1-\beta) \sum_{i \in B_{t-k}} \frac{\partial \ell_i[\phi_{t-k}]}{\partial \phi}$$

## Infinite Weighted Sum of Gradients

By continuing this recursion infinitely, the momentum term $m_{t+1}$ becomes an infinite weighted sum of gradients:

$$m_{t+1} = (1-\beta) \sum_{k=0}^{\infty} \beta^k \sum_{i \in B_{t-k}} \frac{\partial \ell_i[\phi_{t-k}]}{\partial \phi}$$

## Deriving the Weights

The weight for the gradient from iteration $t - k$ is given by:

$$w_k = (1 - \beta)\beta^k$$

This weight is the product of two factors:

- The decay factor $\beta^k$, which decreases the influence of older gradients,
- The scaling factor $(1 - \beta)$, which ensures that the sum of weights across all iterations is 1.

## Final Expression

Thus, the momentum term $m_{t+1}$ can be expressed as an infinite weighted sum of gradients:

$$m_{t+1} = \sum_{k=0}^{\infty}(1 - \beta)\beta^k \sum_{i \in B_{t-k}} \frac{\partial \ell_i[\phi_{t-k}]}{\partial \phi}$$

This shows that $m_{t+1}$ is an infinite weighted sum of the gradients at previous iterations, where the weight for the gradient at iteration $t - k$ is $(1 - \beta)\beta^k$.