Recall that the NN optimization problem is

$$\min_\theta f(\theta)$$

where

$$f(\theta) = \frac{1}{2C} \theta^T \theta + \frac{1}{l} \sum_{i=1}^l \xi(z^{L+1,i}(\theta); y^i, Z^{1,i})$$

Let’s simplify the loss part

$$f(\theta) = \frac{1}{2C} \theta^T \theta + \frac{1}{l} \sum_{i=1}^l \xi(\theta; y^i, Z^{1,i})$$

The issue now is how to do the minimization.
This is one of the most used optimization method

First-order approximation

\[
f(\theta + \Delta \theta) \approx f(\theta) + \nabla f(\theta)^T \Delta \theta,
\]

where

\[
\nabla f(\theta) = \begin{bmatrix}
\frac{\partial f(\theta)}{\partial \theta_1} \\
\vdots \\
\frac{\partial f(\theta)}{\partial \theta_n}
\end{bmatrix}
\]

is the gradient of \( f(\theta) \)
Gradient Descent II

- Solve

\[
\min_{\Delta \theta} \quad \nabla f(\theta)^T \Delta \theta
\]

subject to \( \|\Delta \theta\| = 1 \) (1)

- The constraint \( \|\Delta \theta\| = 1 \) is needed. Otherwise, the above sub-problem goes to \(-\infty\).

- The solution of (1) is

\[
\Delta \theta = -\frac{\nabla f(\theta)}{\|\nabla f(\theta)\|} \quad (2)
\]
Gradient Descent III

- This is called the **steepest descent direction**

- However, because we only consider an approximation

\[ f(\theta + \Delta \theta) \approx f(\theta) + \nabla f(\theta)^T \Delta \theta \]

we may not have the strict decrease of the function value

- That is,

\[ f(\theta) < f(\theta + \Delta \theta) \]

may occur
Gradient Descent IV

- But in general we need the descent property to get the convergence
- We have

\[ f(\theta + \alpha \Delta \theta) = f(\theta) + \alpha \nabla f(\theta)^T \Delta \theta + \frac{1}{2} \alpha^2 \Delta \theta^T \nabla^2 f(\theta) \Delta \theta + \cdots, \]

where

\[ \nabla^2 f(\theta) = \begin{bmatrix} \frac{\partial^2 f}{\partial \theta_1 \partial \theta_1} & \cdots & \frac{\partial^2 f}{\partial \theta_1 \partial \theta_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial \theta_n \partial \theta_1} & \cdots & \frac{\partial^2 f}{\partial \theta_n \partial \theta_n} \end{bmatrix} \]
Gradient Descent V

is the Hessian of $f(\theta)$

- If
  \[ \nabla f(\theta)^T \Delta \theta < 0, \]
  then a small enough $\alpha$ can ensure
  \[ f(\theta + \alpha \Delta \theta) < f(\theta) \]

- Thus in optimization for any direction (not necessarily the steepest descent direction), it is called a descent direction if
  \[ \nabla f(\theta)^T \Delta \theta < 0 \]
The direction chosen in (2) is a descent direction:

$$-\nabla f(\theta)^T \frac{\nabla f(\theta)}{\|\nabla f(\theta)\|} < 0.$$
We have seen that we need a step size $\alpha$ such that

$$f(\theta + \alpha \Delta \theta) < f(\theta)$$

In optimization this is called a line search procedure.

Strict line search

$$\min_{\alpha} f(\theta + \alpha \Delta \theta)$$

This is a one-dimensional optimization problem.

In practice, people use backtracking line search.
We check

$$\alpha = 1, \beta, \beta^2, \ldots$$

with $\beta \in (0, 1)$ until

$$f(\theta + \alpha \Delta \theta) < f(\theta) + \nu \nabla f(\theta)^T (\alpha \Delta \theta)$$

Here

$$\nu \in (0, \frac{1}{2})$$

The convergence is well established.
For example, if the steepest descent direction is used with the backtracking line search, Corollary 1.1.2 at https://sites.math.washington.edu/~burke/crs/408/notes/nlp/line.pdf shows that for every limit point $\bar{\theta}$ of a convergent subsequence of $\{\theta^k\}$, where $k$ is the iteration index, we have

$$\nabla f(\bar{\theta}) = 0$$

This means we can reach a stationary point of a non-convex problem.
The back-tracking line search procedure is simple and useful in practice.
It is known that the convergence is slow for difficult problems.

Thus in many optimization applications, methods of using second-order information (e.g., quasi Newton or Newton) are preferred.

\[
f(\theta + \Delta \theta) \approx f(\theta) + \nabla f(\theta)^T \Delta \theta + \frac{1}{2} \Delta \theta^T \nabla^2 f(\theta) \Delta \theta
\]

These methods have fast final convergence.

An illustration (modified from Tsai et al. (2014))
Practical Use of Gradient Descent II

Slow final convergence  Fast final convergence

- But fast final convergence may not be needed in machine learning
The reason is that an optimal solution $\theta^*$ may not lead to the best model.

We will discuss such issues again later.