Gradient Descent Variants Explained from First Principles

Purpose of This Document

The goal of this document is to explain how neural networks are trained using gradient descent, with a particular focus on the difference between:

- Stochastic Gradient Descent (SGD)
- Mini-Batch Gradient Descent
- Full-Batch Gradient Descent

This document is written for *learning*, not publication. All symbols are explicitly defined, and no steps are skipped.

Dataset and Indexing Conventions

We assume a supervised learning setting and introduce explicit indexing to avoid ambiguity.

- N : total number of training samples
- $x^{(i)} \in \mathbb{R}^d$: input vector of the i-th dataset sample
- $y^{(i)}$: true label of the i-th dataset sample

The dataset is written as:

$$\mathcal{D} = \left\{ (x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)}) \right\}$$

Mini-Batch Indexing (Critical)

During training, the dataset is split into mini-batches.

- m : batch size
- $b \in \{1, 2, \ldots, K\}$: batch index
- $k \in \{1, 2, \ldots, m\}$: index of a sample *within* a batch

Here,

$$K = \frac{N}{m}$$

is the number of mini-batches per epoch (assuming m divides N).

Each mini-batch B_b is a set of dataset indices:

$$B_b = \{\, i_{b,1}, i_{b,2}, \ldots, i_{b,m} \,\}$$

Thus, $i_{b,k}$ refers to the **dataset index** of the k-th sample in batch b.

Model Definition

Let

$$f(x;\theta)$$

be the neural network model, where:

- θ denotes *all trainable parameters*
- θ includes every weight and bias in all layers

For a dataset sample with index $i_{b,k}$, the model prediction is:

$$\hat{y}^{(i_{b,k})} = f(x^{(i_{b,k})};\theta)$$

Loss Function (Per-Sample)

The loss function measures how incorrect a single prediction is.

Let:

$$\ell(\hat{y}, y)$$

denote the loss for one input–label pair.

For a specific sample in batch b:

$$\ell^{(i_{b,k})} = \ell\Big(\hat{y}^{(i_{b,k})}, y^{(i_{b,k})}\Big)$$

Each sample produces **its own loss value**.

Total Training Objective (One Loss Only)

The training objective over the full dataset is defined as:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell\Big(f(x^{(i)};\theta), y^{(i)}\Big)$$

Important clarifications:

- There is exactly **one** loss function $J(\theta)$
- Hidden layers do **not** have separate losses
- All gradients originate from this single scalar value

Goal of Training

The goal is to find parameters θ that minimize $J(\theta)$:

$$\theta^* = \arg\min_{\theta} J(\theta)$$

This is achieved using gradient descent:

$$\theta \leftarrow \theta - \eta\, \nabla_\theta J(\theta)$$

Where:

- $\nabla_\theta J(\theta)$ is the gradient of the loss
- $\eta > 0$ is the learning rate

Mini-Batch Loss

For a given batch B_b, the mini-batch loss is defined as the average of the per-sample losses in that batch:

$$\boxed{J_{B_b}(\theta) = \frac{1}{m} \sum_{k=1}^{m} \ell\Big(f(x^{(i_{b,k})}; \theta), y^{(i_{b,k})}\Big)}$$

This is the loss used to update parameters during batch b.

Gradient of a Mini-Batch

Differentiate the mini-batch loss:

$$\nabla_\theta J_{B_b}(\theta) = \nabla_\theta \left(\frac{1}{m} \sum_{k=1}^{m} \ell^{(i_{b,k})} \right)$$

Using linearity of differentiation:

$$\boxed{\nabla_\theta J_{B_b}(\theta) = \frac{1}{m} \sum_{k=1}^{m} \nabla_\theta \ell\Big(f(x^{(i_{b,k})}; \theta), y^{(i_{b,k})}\Big)}$$

This shows explicitly that:

- A loss is computed for *every* sample
- Gradients are computed for *every* sample
- Gradients are summed and averaged
- Parameters are updated once per batch

Three Gradient Descent Variants

1. Stochastic Gradient Descent (SGD)

SGD uses a batch size of:

$$m = 1$$

Each batch contains a single dataset index $i_{b,1}$.
Update rule:

$$\boxed{\theta \leftarrow \theta - \eta \nabla_\theta \ell\Big(f(x^{(i_{b,1})}; \theta), y^{(i_{b,1})}\Big)}$$

Updates per epoch:

$$N$$

2. Mini-Batch Gradient Descent

Mini-batch gradient descent uses:

$$1 < m < N$$

Update rule for batch b:

$$\boxed{\theta \leftarrow \theta - \eta \nabla_\theta J_{B_b}(\theta)}$$

Updates per epoch:

$$\frac{N}{m}$$

Example:

$$N = 60000, \quad m = 32 \quad \Rightarrow \quad 1875 \text{ updates per epoch}$$

3. Full-Batch Gradient Descent

Full-batch gradient descent uses:

$$m = N$$

The batch contains all dataset indices.
Update rule:

$$\boxed{\theta \leftarrow \theta - \eta \nabla_\theta J(\theta)}$$

Updates per epoch:

$$1$$

Key Conceptual Takeaways

1. There is exactly **one loss**.
2. Every sample produces its own loss and gradient.
3. Mini-batches average gradients, not losses after the fact.
4. Weight updates occur **once per batch**.
5. Batch indexing is essential to avoid conceptual errors.