Outline

Mathematical Background
- Lecture 4: Intro to Optimization
- Lecture 5: Gradient Descent

Lecture 5: Gradient Descent
- Gradient Descent
  - Descent Direction
  - Step Size
  - Convergence
- Stochastic Gradient Descent
  - Difference between GD and SGD
  - Why does SGD work?
Most loss functions in machine learning problems are separable:

\[
J(\theta) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}(g_\theta(x^n), y^n) = \frac{1}{N} \sum_{n=1}^{N} J_n(\theta).
\]

(1)

For example,

- **Square-loss:**
  \[
  J(\theta) = \sum_{n=1}^{N} (g_\theta(x^n) - y^n)^2
  \]

- **Cross-entropy loss:**
  \[
  J(\theta) = - \sum_{n=1}^{N} \left\{ y^n \log g_\theta(x^n) + (1 - y^n) \log(1 - g_\theta(x^n)) \right\}
  \]

- **Logistic loss:**
  \[
  J(\theta) = \sum_{n=1}^{N} \log(1 + e^{-y^n \theta^T x^n})
  \]
Full Gradient VS Partial Gradient

Vanilla gradient descent:

$$\theta^{t+1} = \theta^t - \eta^t \nabla J(\theta^t).$$ \hspace{1cm} (2)

The full gradient of the loss is

$$\nabla J(\theta) = \frac{1}{N} \sum_{n=1}^{N} \nabla J_n(\theta).$$ \hspace{1cm} (3)

Stochastic gradient descent:

$$\nabla J(\theta) \approx \frac{1}{|\mathcal{B}|} \sum_{n \in \mathcal{B}} \nabla J_n(\theta)$$ \hspace{1cm} (4)

where $\mathcal{B} \subseteq \{1, \ldots, N\}$ is a random subset. $|\mathcal{B}| = \text{batch size.}$
Algorithm (Stochastic Gradient Descent)

1. Given \{({x}^n, {y}^n) \mid n = 1, \ldots, N\}.
2. Initialize \( \theta \) (zero or random)
3. For \( t = 1, 2, 3, \ldots \)
   - Draw a random subset \( B \subseteq \{1, \ldots, N\} \).
   - Update
     \[
     \theta^{t+1} = \theta^t - \eta^t \frac{1}{|B|} \sum_{n \in B} \nabla J_n(\theta)
     \]

   - If \( |B| = 1 \), then use only one sample at a time.
   - The approximate gradient is **unbiased**: (See Appendix for Proof)

\[
\mathbb{E} \left[ \frac{1}{|B|} \sum_{n \in B} \nabla J_n(\theta) \right] = \nabla J(\theta).
\]
Interpreting SGD

- Just showed that the SGD step is unbiased:

\[
\mathbb{E} \left[ \frac{1}{|\mathcal{B}|} \sum_{n \in \mathcal{B}} \nabla J_n(\theta) \right] = \nabla J(\theta).
\]

- Unbiased gradient implies that each update is

\[
\text{gradient } + \text{ zero-mean noise}
\]

- Step size: SGD with constant step size does not converge.
- If \( \theta^* \) is a minimizer, then \( J(\theta^*) = \frac{1}{N} \sum_{n=1}^{N} J_n(\theta^*) = 0 \). But

\[
\frac{1}{|\mathcal{B}|} \sum_{n \in \mathcal{B}} J_n(\theta^*) \neq 0, \quad \text{since } \mathcal{B} \text{ is a subset.}
\]

- Typical strategy: Start with large step size and gradually decrease:

\( \eta^t \to 0 \), e.g., \( \eta^t = t^{-a} \) for some constant \( a \).
Perspectives of SGD

Classical optimization literature have the following observations.

- Compared to GD in **convex** problems:
- SGD offers a **trade-off** between accuracy and efficiency
- More iterations
- Less gradient evaluation per iteration
- Noise is a by-product

Recent studies of SGD for **non-convex** problems found that

- SGD for training deep neural networks works
- SGD finds solution faster
- SGD find a better local minima
- Noise matters
GD compared to SGD

Convex

Non-Convex

GD

SGD

GD

SGD
Smoothing the Landscape

Analyzing SGD is an active research topic. Here is one by Kleinberg et al. (https://arxiv.org/pdf/1802.06175.pdf ICML 2018)

- The SGD step can be written as GD + noise:

\[
x^{t+1} = x^t - \eta (\nabla f(x^t) + w^t) = x^t - \eta \nabla f(x^t) - \eta w^t.
\]

- \(y^t\) is the “ideal” location returned by GD.
- Let us analyze \(y^{t+1}\):

\[
y^{t+1} \overset{\text{def}}{=} x^{t+1} - \eta \nabla f(x^{t+1})
\]
\[
= (y^t - \eta w^t) - \eta \nabla f(y^t - \eta w^t)
\]

- Assume \(\mathbb{E}[w] = 0\), then

\[
\mathbb{E}[y^{t+1}] = y^t - \eta \nabla \mathbb{E}[f(y^t - \eta w^t)]
\]
Smoothing the Landscape

- Let us look at $\mathbb{E}[f(y^t - \eta w^t)]$:

$$
\mathbb{E}[f(y - \eta w)] = \int f(y - \eta w)p(w) \, dw,
$$

where $p(w)$ is the distribution of $w$.
- $\int f(y - \eta w)p(w) \, dw$ is the convolution between $f$ and $p$.
- $p(w) \geq 0$ for all $w$, so the convolution always smooths the function.
- Learning rate controls the smoothness
  - Too small: Under-smooth. You have not yet escaped from bad local minimum.
  - Too large: Over-smooth. You may miss a local minimum.
Smoothing the Landscape
Reading List

**Gradient Descent**


**Stochastic Gradient Descent**