Extra Mathematical Notes for Lectures and Classes*

Parley Ruogu Yang†

This version: Friday 21$^{\text{st}}$ January, 2022

Abstract

This document consists of notes for Lectures (section 2) and Classes (section 3).

Contents

*Latest version: `https://parleyyang.github.io/ST456/index.html`

†Faculty of Mathematics, University of Cambridge

1 General notes

1.1 Notations

The default meaning of $\mathbb{N}$ is the set of integers greater or equal to 1. For $n \in \mathbb{N}$, denote $[n] := \{1, 2, ..., n-1, n\} = [1, n] \cap \mathbb{N}$.

$N(\mu, \sigma^2)$ refers to a normal distribution with mean μ and variance σ^2, while a standard normal distribution refers to the case when $\mu = 0$ and $\sigma^2 = 1$.

Where ε or ε_i are written, the default meaning is that they are drawn from iid $N(0, \sigma^2)$ distribution with σ^2 unknown.

1.2 Activation functions

Let $\rho^{\texttt{sigmoid}} : \mathbb{R} \to \mathbb{R}$ be the sigmoid function, it is defined by

$$x \mapsto (1 + \exp(-x))^{-1} \tag{1.2.1}$$

2 Lectures

2.1 Lecture 2: Neural Networks Basics

3 Classes

3.1 Class 1: Linear and logistic regressions

3.1.1 Linear regression and MSE loss

Let $x \in \mathbb{R}^m$ be the input variable. Let $y \in \mathbb{R}$ be the output variable.

We consider $y = f(x) + \varepsilon$ where $f(x) = x^T w + b$

If we have data $\{(x_i, y_i) : i \in [n]\}$, the MSE loss takes the following form:

$$l(w, b) = n^{-1} \sum_{i \in [n]} (y_i - f(x_i))^2 = n^{-1} \sum_{i \in [n]} (y_i - x_i^T w - b)^2 \tag{3.1.2}$$

3.1.2 Gradient of linear regression with MSE loss

It will be useful later in subsection 3.3 to have the gradient ∇l in hand. In particular:

$$\nabla_w l(w, b) = n^{-1} \sum_{i \in [n]} (2x_i)(f(x_i) - y_i) \tag{3.1.3}$$

$$\nabla_b l(w, b) = n^{-1} \sum_{i \in [n]} 2(f(x_i) - y_i) \tag{3.1.4}$$

3.1.3 Logistic regression model and binary cross entropy

Let ρ be the sigmoid function, then we consider $y = f(x) + \varepsilon$ where $f(x) = \rho(x^T w + b)$

In the event of binary classification problem, in which $y \in \{0, 1\}$, we clearly do not have ε as a Normally distributed error. In this occasion, with data $\{(x_i, y_i) : i \in [n]\}$, we consider the binary cross entropy as

$$l(f) = -n^{-1} \left(\sum_{i \in [n]} y_i \log(f(x_i)) + (1 - y_i) \log(1 - f(x_i)) \right) \tag{3.1.5}$$

3.2 Class 2: The XOR Problem

3.3 Class 3: Gradient Descent and Stochastic Gradient Descent

3.3.1 Gradient Descent

Let $f : \mathbb{R}^{\ltimes} \to \mathbb{R}$ be a differentiable function, a gradient descent sequence $\{x_n\}_{n=0}^{\infty}$ with learning rate scheduling $\{\eta_n\}_{n=0}^{\infty}$ and initialisation x_{ini} is defined as

$$x_0 = x_{ini} \tag{3.3.6}$$

$$x_n = x_{n-1} - \eta_{n-1} \nabla f(x_{n-1}) \quad \forall n \in \mathbb{N} \tag{3.3.7}$$

3.4 Class 3: Option Pricing

3.4.1 Background

A (European) call option at maturity T gives the owner the right to buy an underlying asset at strike price K. This price of such an option is denoted as $V(S_t, t; K)$ at time $t \in [0, T]$, where S_t is the price of the underlying asset at time t. It is natural to relate this to various parameters in the market: in the Black-Scholes model, we relate this to the interest rate r and volatility σ. A PDE expression is provided as

$$\partial_t V + rS\partial_S V + \frac{1}{2}\sigma^2 S^2 \partial_S^2 V = rV \tag{3.4.8}$$

The solution of this is complicated and non-linear:

$$V(S_t, t; K) = S_t N(d_1) - Ke^{r(T-t)} N(d_2) \tag{3.4.9}$$

where $d_1 = (\sigma\sqrt{T-t})^{-1}(\log(S_t K^{-1}) + (r + \frac{\sigma^2}{2}(T-t))$ and $d_2 = d_1 - \sigma\sqrt{T-t}$

3.4.2 Class 3 Notebook 1

In this notebook, we keep other parameters the same and study the relationship between strike price K and the associated price of call option V . In particular, we select a number of strike prices, denoted $x_1, ..., x_n \in \mathbb{R}$ and generate the call option prices $y_1, ..., y_n \in \mathbb{R}$ in accordance with Equation 3.4.9. The dataset is hence $\{(x_i, y_i) : i \in [n]\}$ and that we would like to approximate a function $f : \mathbb{R} \to \mathbb{R}$ as we generate our data $y_i = f(x_i) \ \ \forall i$

3.4.3 Class 3 Notebook 2

In practice, one would be asked for the implied volatility σ given the data they receive — in this notebook, we fix 16 different strike prices and collect their corresponding call prices: for now, assume no noise. Then, for each $y_i = \sigma_i \in \mathbb{R}$, we have a 16-dimensional data $x_i \in \mathbb{R}^{16}$, so the dataset is $\{(x_i, y_i) : i \in [n]\}$ and that we would like to approximate a function $f : \mathbb{R}^{16} \to \mathbb{R}$ as we generate our data $y_i = f(x_i) \ \ \forall i$

3.4.4 Class 3 Homework

Realistically, the data contains noise. In the Homework, we will work with noisy data, in particular, we consider the same function $f : \mathbb{R}^{16} \to \mathbb{R}$ as was in Notebook 2, but that we generate $\varepsilon_i \sim N(0_{16}, \sigma^2 I_{16\times 16}) \forall i \in [n]$, and observe $\tilde{x_i} = \max\{x_i + \varepsilon_i, 0\}$ instead of x_i. The maximum is in place because the practical world would not accept a negative prices on an option — so whilst there are noises, there is an obvious truncation.

So, we are still in the business of approximating f, but this time we have data $\{(\tilde{x_i}, y_i) : i \in [n]\}$.