The Deep BSDE Method

Daniel Kovach
Missouri University of Science and Technology

THE DEEP BSDE METHOD

by

DANIEL GERALD KOVACH II

A THESIS

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

APPLIED MATHEMATICS

2023

Approved by:

Jason Murphy, Advisor
Daozhi Han
Yanzhi Zhang

ABSTRACT

The curse of dimensionality is the non-linear growth in computing time as the dimension of a problem increases. Using the Deep Backwards Stochastic Differential Equation (Deep BSDE) method developed in [1], I approximate the solution at an initial time to a one-dimensional diffusion equation. Although we only approximate a one-dimensional equation, this method extends well to higher dimensions because it overcomes the curse of dimensionality by evaluating the given partial differential equation along “random characteristics”. In addition to the implementation, I also present most of the mathematical theory needed to understand this method.

ACKNOWLEDGMENTS

I would like to thank Dr. Murphy for his expertise and guidance with partial differential equations and the theory and implementation behind the Deep BSDE method and neural networks. Additionally, I would like to thank Dr. Wenqing Hu for his expertise in deep learning and help in understanding the various optimization methods traditionally used in statistical learning, and Jacob Hauck, a PhD student and Kummer Fellow, who was kind enough to assist with the PyTorch implementation. I would also like to thank Dr. Chen-Murphy, whose applied matrix theory course and other pure math courses improved my ability to write proofs and understand the theory necessary for this paper. Last, but not least, I would like to thank the members of my family who supported me and without whom I could not have completed this. Namely, my parents and grandparents.

TABLE OF CONTENTS

Page

SECTION

LIST OF ILLUSTRATIONS

1. INTRODUCTION

The role of neural networks in solving the issues today has expanded at an even greater pace in recent years. In our setting, we apply them to partial differential equations by using stochastic differential equations as training data with a method that generalizes well to high dimensions. This method was first established in [1], so we are only presenting most of the theory required to understand it and an implementation of it. We emphasize that we did not discover this method, and due to technical limitations, we present only a one-dimensional model. A survey of related methods can be found in [2]. There are two branches of mathematics needed to understand the Deep BSDE method: stochastic differential equations and nonlinear optimization. We begin with the former, continue with the latter, then present the implementation of [1].

2. STOCHASTIC THEORY

The following assumes familiarity with measure theory. See the appendix for more information.

2.1. PROBABILITY THEORY

2.1.1. Preliminaries. We first review some definitions from probability theory.

Definition: A probability space is a triple $(\Omega, \mathcal{U}, \mathbb{P})$ where $\Omega \subset \mathbb{R}^n$ is a collection of open sets, $\mathcal{U}$ is a σ-algebra which acts on subsets of Ω, and $\mathbb{P}$ is a probability measure on $\mathcal{U}$. That is, a probability space is a measure space where the measure is a probability measure.

Definition: We say $\mathbb{P} : \mathcal{U} \to [0, 1]$ is a *probability measure* on the σ-algebra $\mathcal{U}$ if the following holds:

1. $\mathbb{P}(\emptyset) = 0, \mathbb{P}(\Omega) = 1.$
2. Countable sub-additivity: If $A_1, A_2, \ldots \in \mathcal{U}$, then

   $$\mathbb{P}(\cup_{k=1}^{\infty}) \leq \sum_{k=1}^{\infty} \mathbb{P}(A_k),$$

   with equality holding if A_k's are disjoint.
3. Monotonicity: If $A, B \in \mathcal{U} : A \subseteq B$, then $\mathbb{P}(A) \leq \mathbb{P}(B)$.

Definition: A mapping

$$\mathbf{X} : \Omega \to \mathbb{R}^n$$

is called an n-dimensional random variable if for each $B \in \mathcal{B}$, we have

$$\mathbf{X}^{-1}(B) \in \mathcal{U},$$

where $\mathcal{B}$ is the Borel subsets of $\mathbb{R}^n$. Note that

$$\mathbf{X}^{-1}(B) \in \mathcal{U} = \mathbb{P}(\mathbf{X} \in B)$$

and

$$\exists \Phi : Y = \Phi(\mathbf{X}) \iff \exists Y \in \mathcal{U}(\mathbf{X}).$$

Definition: The expected value of a vector-valued random variable is

$$\mathbb{E}(\mathbf{X}) := \int_\Omega \mathbf{X} d\mathbb{P}.$$

Definition: The variance of a vector-valued random variable is

$$\mathbb{V}(\mathbf{X}) := \int_\Omega |\mathbf{X} - \mathbb{E}(\mathbf{X})|^2 d\mathbb{P}.$$

Definition: The *distribution function* of **X** is the function

$$F_{\mathbf{X}} : \mathbb{R}^n \to [0, 1]$$

defined by

$$F_{\mathbf{X}} := \mathbb{P}(\mathbf{X} \leq x), \forall x \in \mathbb{R}^n.$$

If we have a collection of vector-valued random variables, we generalize the previous definition to

$$F_{\mathbf{X}_1,\ldots,\mathbf{X}_m} : (\mathbb{R}^n)^m \to [0, 1]$$

defined by

$$F_{\mathbf{X}_1,\ldots,\mathbf{X}_m} := \mathbb{P}(\mathbf{X}_1 \leq x, \ldots, \mathbf{X}_m \leq x_m), \forall x_k \in \mathbb{R}^n,\ k = 1, \ldots, m.$$

Definition: Let $\mathbf{X} : \Omega \longrightarrow \mathbb{R}^n$ be a random variable with distribution function $F = F_{\mathbf{X}}$. If there exists a non-negative, integrable function $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ such that

$$F(x) = F(x_1, \ldots, x_n) = \int_{-\infty}^{x_1} \ldots \int_{-\infty}^{x_n} f(y_1, \ldots, y_n) dy_n \ldots dy_1,$$

then f is called the *density* function for $\mathbf{X}$.

Then we also have

$$\mathbb{P}(\mathbf{X} \in B) = \int_B f(x) dx, \forall B \in \mathcal{B}.$$

Definition: A random variable $\mathbf{X} : \Omega \longrightarrow \mathbb{R}$ with density

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{|x-\mu|^2}{2\sigma^2}}\ (x \in \mathbb{R})$$

is called *Gaussian* with mean μ and variance σ^2.

Theorem: If the distribution of $\mathbf{X}$ is given by $\mathbb{P}$, with density $f(\mathbf{X})$, then for any function $g(\mathbf{X}) \in \mathcal{B} \subset \Omega$,

$$\mathbb{E}(g(\mathbf{X})) = \int_\Omega g(\mathbf{X}) d\mathbb{P} = \int_\Omega g(\mathbf{X}) f(\mathbf{X}) d\mathbf{X}.$$

So $\mathbb{E}(g(\mathbf{X}))$ is a linear functional $\ell_{\mathbf{X}} : C(\mathbb{R}^n; \mathbb{R}) \longrightarrow \mathbb{R}$. So by the Riecz Representation theorem (see appendix A), we know there exists a unique Borel measure $\mu_{\mathbf{X}}$ such that

$$\forall g \in C(\mathbb{R}^n), \mathbb{E}(g(\mathbf{X})) = \int_\Omega f d\mu_{\mathbf{X}}.$$

Definition: The characteristic function of a density function is the Fourier transform...

2.1.2. Independence. *Definition:* The *conditional probability* of an event $A \in \mathcal{U}$, given the occurrence of an event $B \in \mathcal{U}$, is denoted $\mathbb{P}(A|B)$, and defined to be

$$\mathbb{P}(A|B) := \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}, \mathbb{P}(B) > 0.$$

Definition: Events $A_1, A_2, \ldots \in \mathcal{U}$ are said to be independent if $\mathbb{P}(\bigcap_{i \in \mathbb{N}} A_i) = \prod_{i \in \mathbb{N}} \mathbb{P}(A_i)$. Note that $\mathbb{P}(B) = 0$ is allowed under this definition. Similarly, a collection of random variables

$\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_k$ are said to be independent if

$$\forall B_i \in \mathcal{B}, \mathbb{P}(\mathbf{X}_1 \in B_1, \mathbf{X}_2 \in B_2, \ldots, \mathbf{X}_k \in B_k) = \prod_{i=1}^{k} \mathbb{P}(\mathbf{X}_i \in B_i).$$

Equivalently, the set $\{\mathcal{U}(\mathbf{X}_i)\}_{i \in \mathbb{N}}$ of σ-alebgras are independent sets.

If $\{\mathbf{X}_i\}_{i \in \mathbb{N}}$ are independent, then

$$\mathbb{E}(\prod_{i=1}^{n} \mathbf{X}_i) = \prod_{i=1}^{n} \mathbb{E}(\mathbf{X}_i) \text{ and } \mathbb{V}(\sum_{i=1}^{n} \mathbf{X}_i) = \sum_{i=1}^{n} \mathbb{V}(\mathbf{X}_i).$$

If $\mathbf{X}_1 \sim N(\mu_1, \sigma_1^2), \mathbf{X}_2 \sim N(\mu_2, \sigma_2^2)$, then

$$\mathbf{X}_1 + \mathbf{X}_2 \sim N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2).$$

Proof. We have

$$\mu_1 = \mathbb{E}(\mathbf{X}_1) = \int_{\Omega} \mathbf{X}_1 d\mathbb{P},$$

$$\mu_2 = \mathbb{E}(\mathbf{X}_2) = \int_{\Omega} \mathbf{X}_2 d\mathbb{P}.$$

By linearity,

$$\mu_1 + \mu_2 = \mathbb{E}(\mathbf{X}_1 + \mathbf{X}_2).$$

Now,

$$\sigma_1 = \int_\Omega |\mathbf{X}_1 - \mu_1|^2 d\mathbb{P},$$

$$\sigma_2 = \int_\Omega |\mathbf{X}_2 - \mu_2|^2 d\mathbb{P}.$$

Again by linearity,

$$\begin{aligned}
\sigma_1 + \sigma_2 &= \int_\Omega |\mathbf{X}_1 - \mu_1|^2 + |\mathbf{X}_2 - \mu_2|^2 d\mathbb{P} \\
&= \int_\Omega \langle \mathbf{X}_1 - \mu_1, \mathbf{X}_1 - \mu_1 \rangle + \langle \mathbf{X}_2 - \mu_2, \mathbf{X}_2 - \mu_2 \rangle d\mathbb{P} \\
&= \int_\Omega \langle \mathbf{X}_1 - \mu_1 + \mathbf{X}_2 - \mu_2, \mathbf{X}_1 - \mu_1 + \mathbf{X}_2 - \mu_2 \rangle d\mathbb{P} \\
&= \int_\Omega |\mathbf{X}_1 - \mu_1 + \mathbf{X}_2 - \mu_2|^2 d\mathbb{P} \\
&= \int_\Omega |\mathbf{X}_1 + \mathbf{X}_2 - (\mu_1 + \mu_2)|^2 d\mathbb{P}
\end{aligned}$$

□

2.1.3. Conditional Expectation. *Definition:* $\mathbb{E}(X|B) = \frac{1}{\mathbb{P}(B)} \int_B X d\mathbb{P}$.

Definition: $L^2(\Omega, \mathcal{U})$ denotes the set of all square-integrable, $\mathcal{U}$-measurable functions supported on a set Ω acted on by a σ-algebra $\mathcal{U}$ and is endowed with the norm

$$\|\mathbf{X}\|_{L^2(\Omega)} := (\int_\Omega |\mathbf{X}|^2 d\mathbb{P})^{\frac{1}{2}} < \infty.$$

Note that each element of L^2 is actually an equivalence class of all functions with the same measure, not a particular function.

Remark: Let $\mathcal{V} \subset \mathcal{U}$, be another σ-algebra, and define $\mathcal{V} = L^2(\Omega; \mathcal{V})$. Then $\mathcal{V}$ is a closed subspace of $L^2(\Omega; \mathcal{U})$.

Remark: Given $\mathbf{X} \in L^2(\Omega; \mathcal{U})$. $\mathbb{E}(\mathbf{X}|\mathcal{V}) = \text{proj}_{\mathcal{V}}\mathbf{X}$. ($L^2$ projection formula).

2.2. STOCHASTIC PROCESSES

Definitions:

1. A collection $\{\mathbf{X}(t) | t \geq 0\}$ of random variables is called a stochastic process.
2. For each point $\omega \in \Omega$, $t \mapsto \mathbf{X}(t, \omega)$ is the corresponding sample path.

Definition: Let $X(\cdot)$ be a stochastic process. Then

$$\mathcal{U}(t) := \mathcal{U}(X(s) \mid 0 \leq s \leq t),$$

the σ-algebra generated by the random variables $X(s)$ for $0 \leq s \leq t$, is called the *history* of the process up to and including the time $t = 0$.

Definition: We call $X(\cdot)$ a *martingale* if $\forall t \in \mathbb{R}, \mathbb{E}(|X(t)|) < \infty$ and

$$\forall 0 \leq s \leq t, \mathbb{E}(X(t)|\mathcal{U}(s)) = X(s).$$

We have the following martingale inequalities, respectively called Doob's mertingale inequality and Doob's maximal inequality. (The martingale inequality theorems on p. 35-36 in [3].)

1. For all $\lambda > 0, t \geq 0$,

$$\mathbb{P}(\max_{0 \leq s \leq t} X(s) \geq \lambda) \leq \frac{1}{\lambda}\mathbb{E}(\max(0, X(t))).$$

2. For $1 < p < \infty$,

$$\mathbb{E}(\max_{0 \le s \le t} |X(s)|^p) \le (\frac{p}{p-1})^p \mathbb{E}(|X(t)|^p).$$

2.2.1. Brownian Motion.

Definition: A stochastic process $W(\cdot)$ is called a Brownian motion (or Wiener process) if

1. $W(0) = 0$ a.s.,
2. $W(t) - W(s)$ is $N(0, t-s)$ for all $t \ge s \ge 0$,
3. For all times $0 < t_1 < \ldots < t_n$, the random variables

$$W(t_1), W(t_2) - W(t_1), \ldots, W(t_n) - W(t_{n-1})$$

are independent.

Proposition: Using these definitions, we may show that the following properties of Brownian motion hold.

1. $\mathbb{E}(W(t)) = 0,\ \mathbb{E}(W^2(t)) = t \quad (t \ge 0)$ and
2. $\mathbb{E}(W(t)W(s)) = t \wedge s = \min\{s, t\}$ where $t, s \ge 0$.

Proof. Since $W(t) \sim \mathcal{N}(0, t)$, $\mathbb{E}(W(t)) = 0$ immediately follows. $t = (\mathbb{V}(W(t)))^{\frac{1}{2}} = \mathbb{E}((W(t) - E(W(t)))^2) = \mathbb{E}(W^2(t))$ follows from this as well. Next, without loss of generality, take $0 \le s \le t$. Then by adding zero, we see that

$$\begin{aligned}
\mathbb{E}(W(t)W(s)) &= \mathbb{E}([W(s) + W(t) - W(s)]W(s)) \\
&= \mathbb{E}(W^2(s) + [W(t) - W(s)]) \\
&= \mathbb{E}(W^2(s)) + \mathbb{E}[W(t) - W(s)] \\
&= s.
\end{aligned}$$

□

Construction:

Let $\{\psi_n\}_{n=0}^{\infty}$ be a complete, orthonormal basis of $L^2(0,1)$ such that the basis functions $\psi_n : [0,1] \to \mathbb{R}$ are deterministic. Then

$$\exists \text{ random } A_n \in \mathbb{R} : \forall \xi(t) \in L^2(0,1), \xi(t) = \sum_{n=0}^{\infty} A_n \psi_n(t).$$

Multiplying by ψ_m on both sides of the equality, integrating against time, and using orthonormality, we obtain

$$A_n = \int_0^1 \xi(t)\psi_n(t)dt.$$

We can think of a particular $\xi(t)$ as the one dimensional time-derivative of Brownian motion. Note that this is the derivative in the sense of we have integrate against the function to get the original function.

So then we have

$$W(t) = \int_0^t \xi(s)ds = \sum_{n=0}^{\infty} A_n \int_0^t \psi_n(s)ds.$$

Theoretically any orthonormal basis will work, but we will use a basis that is the family of Haar functions.

Definition: The family of Haar functions, $\{h_k\}_{k=0}^{\infty}$ are defined for $t \in [0,1]$ such that for some positive n with $2^n \leq k < 2^{n+1}$, we have

$$h_k(t) := \begin{cases} 2^{n/2} & \text{for } \frac{k-2^n}{2^n} \leq t \leq \frac{k-2^n+1/2}{2^n} \\ -2^{n/2} & \text{for } \frac{k-2^n+1/2}{2^n} < t \leq \frac{k-2^n+1}{2^n} \\ 0 & \text{otherwise.} \end{cases}$$

We also define $h_0(t) := 1$ for $t \in [0,1]$. See Figure 2.1 for a graph of some members of this family.

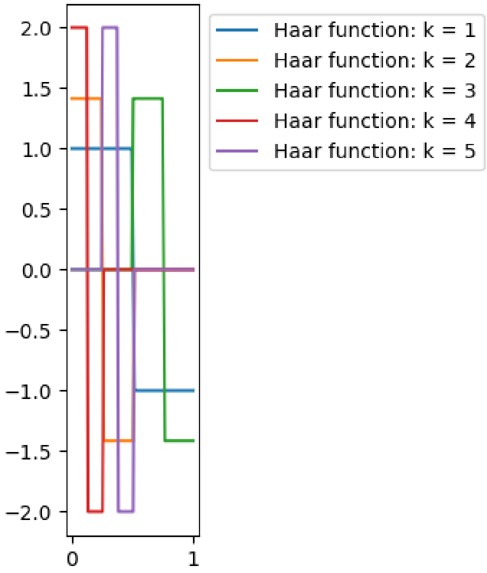

Figure 2.1. Haar Functions

Proposition: This family of functions forms an orthonormal basis.

Proof. Normality is fairly straightforward. For $k = 0$, $\int_0^1 h_0^2(t)dt = 1$ holds. For $k > 0$, we have

$$\int_0^1 h_k^2(t)dt = \int_{\frac{k-2^n}{2^n}}^{\frac{k-2^n+1}{2^n}} h_k^2(t)dt = \int_{\frac{k-2^n}{2^n}}^{\frac{k-2^n+1}{2^n}} 2^n dt = 2^n[\frac{k-2^n+1}{2^n} - \frac{k-2^n}{2^n}] = 1.$$

It's obvious that $\forall k > 0, \int_0^1 h_k dt = 0$.

We have two cases for the orthogonality proof.

1. If their supports are disjoint, then $\forall \ell > k \geq 0,\ h_k h_\ell = 0$.
2. If their supports are not disjoint, then h_k is constant on the support of h_ℓ. Thus we have

$$\int_0^1 h_\ell h_k dt = \pm 2^{n/2} \int_0^1 h_\ell dt = 0.$$

□

Definition: From Haar functions, we introduce the family of *Schauder functions*, depicted in Figure 2.2. The k-th *Schauder function* for $k = 0, 1, \ldots,$ and $t \in [0, 1]$, is

$$s_k(t) = \int_0^t h_k(s)ds.$$

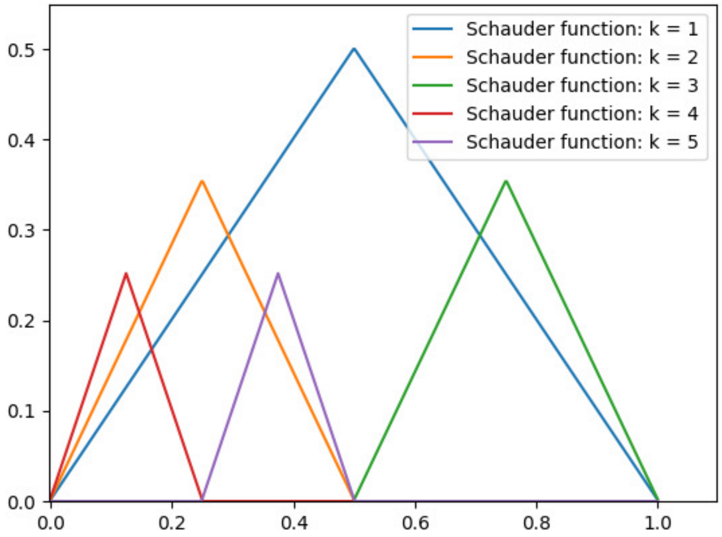

Figure 2.2. Schauder Functions

Proposition: A sample path of Brownian motion, $t \mapsto W(t, \omega)$ is uniformly Hölder continuous for each exponent $0 < \gamma < 1/2$, but nowhere Hölder continuous with exponent $1/2 < \gamma$. More specifically, this mapping is a.s. nowhere differentiable and is of infinite variation for each time interval.

It is possible to use the integral of Haar functions, known as Schauder functions, to form an orthonormal basis of L^2 and explicitly construct Brownian motion from this basis, but we will not do so here.

The lack of differentiability can be partially explained by the *Markov property* of Brownian motion.

Definitions:

1. Earlier we defined the conditional expectation. We will now define the *conditional probability*. That is, if $\mathcal{V} \subseteq \mathcal{U}$ is a σ-algebra with $\mathcal{V} \subseteq \mathcal{U}$, then the conditional probability of A given $\mathcal{V}$ is

$$P(A|\mathcal{V}) := \mathbb{E}(\chi_A|\mathcal{V}) \text{ for } A \in \mathcal{U}.$$

2. A stochastic process $\mathbf{X}(\cdot) \subset \mathbb{R}^n$ is a *Markov process* if it satisfies

$$\forall 0 \leq s \leq t, \forall \text{ Borel subsets } B, P(\mathbf{X}(t) \in B|\mathcal{U}(s)) = P(\mathbf{X}(t) \in B|\mathbf{X}(s)) a.s.$$

2.2.2. Introduction to Stochastic Integrals. We are going to need to work with a different kind of integral in order to integrate against Brownian motion. This integral is similar in nature to Riemann-Stieljes integrals, but Brownian motion is of infinite variation so this will not work. Therefore we have to use the *Itô* integral which is a Riemann sum. We will need some definitions to establish this. First we will use the *Paley-Wiener-Zygmund* integral.

Definition: Let $g : [0, T] \longrightarrow \mathbb{R}$ be continuously differentiable and deterministic such that $g(0) = g(T) = 0$ (g is zero on the boundary). Then we can define a formula similar to integration by parts:

$$\int_0^T g dW := -\int_0^T g' W dt.$$

Lemma: The following properties hold for the *Paley-Wiener-Zygmund* integral:

1. $\mathbb{E}(\int_0^T g dW) = 0,$

2. $\mathbb{E}((\int_0^T g dW)^2) = \int_0^T g^2 dt.$

Proof. From the previous definition and Fubini's theorem, we have

$$\mathbb{E}(\int_0^T gdW) = \mathbb{E}(\int_0^T g'Wdt) = \int_0^T \mathbb{E}(g'W)dt = \int_0^T (g')\mathbb{E}(W)dt = \int_0^T (g')0dt = 0.$$

Again, by Fubini's,

$$\mathbb{E}((\int_0^T gdW)^2) = \mathbb{E}((\int_0^T g'Wdt)(\int_0^T g'Wds)) = (\int_0^T \int_0^T g'(t)g'(s)\mathbb{E}(W(s)W(t))dtds).$$

We've previously found that $\mathbb{E}(W(s)W(t)) = \min(t,s)$. Take $s = \min(t,s)$ So using the above conditions on g and integrating by parts, we have

$$(\int_0^T \int_0^T g'(t)g'(s)\mathbb{E}(W(s)W(t))dtds) = \int_0^T \int_0^T g'(t)g'(s)sdtds =$$

$$\int_0^T g'(t)[\int_0^t g'(s)sds + \int_t^T g'(s)sds]dt = \int_0^T g'(t)[\int_0^t g'(s)sds + \int_t^T tg'(s)ds]dt =$$

$$\int_0^T g'(t)[(tg(t) - \int_0^t g(s)ds) + (\int_t^T tg'(s)ds)]dt = \int_0^T g'(t)[(tg(t) - \int_0^t g(s)ds) + t(-g(t))]dt =$$

$$\int_0^T g'(t)[-\int_0^t g(s)ds]dt = \int_0^T g^2dt.$$

□

Claim: If $g_n \longrightarrow g$ in L^2 with $g_n(0) = g_n(T) = 0$, then $\{\int_0^T g_n(t)dt\}_n^\infty$ is Cauchy in L^2.

Proof. Since $g_n \longrightarrow g$ in L^2, it is Cauchy in L^2. From the above lemma, we also know that

$$\int_0^T (g_n - g_m)^2 dt = \mathbb{E}((\int_0^T (g_n - g_m)dW)^2).$$

□

We can now define

$$\lim_{n\to\infty} \int_0^T g dW = \lim_{n\to\infty} \int_0^T g_n dW,$$

with the limit being understood in the $L^2(\Omega)$ sense. Our integral only works for deterministic functions.

Definitions:

1. Given a partition in time P, the mesh size of P is the largest gap between time intervals. That is,

$$|P| := \max_{0\leq k\leq m-1} |t_{k+1} - t_k|.$$

2. Let $0 \leq \lambda \leq 1$ be fixed and let P be a partition of $[0, T]$. Then we define

$$\tau_k := (1 - \lambda)t_k + \lambda t_{k+1}.$$

3. Given the previous two definitions, we can now construct the Riemann sum approximation of the integral

$$\int_0^T W dW$$

with the Riemann sum

$$R := R(P,\lambda) = \sum_{k=0}^{m-1} W(\tau_k)(W(t_{k+1}) - W(t_k)).$$

Depending on our choice of λ, we will have different Riemann approximations. If $\lambda = 0$, this will be the approximation used to for the Itô integral. If $\lambda = 1/2$, it will be the *Stratonovich* integral. We will eventually see that the choice of λ impacts the chain rule used in Stochasic calculus. Itô's chain rule is different from the typical chain rule seen in classical calculus, but Stratonovich's definition actually does not differ. In this paper, we will only be dealing with Itô's integral.

The Quadratic Variation Lemma:

Let $[T_0, T] \subset [0, \infty)$ be an interval with n (different) partitions collectively denoted by

$$P^n := \{T_0 = t_0^n < t_1^n < \ldots < t_{m_n}^n = T\} \text{ such that } |P^n| \to 0 \text{ as } n \to \infty.$$

Then

$$\sum_{k=0}^{m_n-1} (W(t_{k+1}^n) - W(t_k^n))^2 \to T - T_0 \; in \; L^2(\Omega).$$

Proof. Telescoping, we have

$$\sum_{k=0}^{m_n-1} (W(t_{k+1}^n) - W(t_k^n))^2 - (T - T_0) = \sum_{k=0}^{m_n-1} (W(t_{k+1}^n) - W(t_k^n))^2 - (t_{k+1}^n - t_k^n).$$

Taking the expectation of this squared, we have

$$\begin{aligned}
&\mathbb{E}([\sum_{k=0}^{m_n-1}(W(t_{k+1}^n)-W(t_k^n))^2-(T-T_0)]^2)\\
&=\sum_{k=0}^{m_n-1}\sum_{j=0}^{m_n-1}\mathbb{E}([(W(t_{k+1}^n)-W(t_k^n))^2-(t_{k+1}^n-t_k^n)][(W(t_{j+1})-W(t_j))^2-(t_{j+1}^n-t_j^n)]).
\end{aligned}$$

If $k \neq j$, we can see that

$$\begin{aligned}
\mathbb{E}([(W(t_{k+1}^n)-W(t_k^n))^2-(t_{k+1}^n-t_k^n)] &= \mathbb{E}[(W(t_{k+1}^n))^2-t_k+(W(t_k^n))^2-t_k-(t_{k+1}^n-t_k^n)])\\
&= \mathbb{E}[(W(t_{k+1}^n))^2-t_k^n+(W(t_k^n))^2-t_k^n-(t_{k+1}^n-t_k^n)]\\
&= \mathbb{E}(W(t_{k+1}^n))^2-t_k^n+\mathbb{E}(W(t_k^n))^2-t_k^n-(t_{k+1}^n-t_k^n)\\
&= t_{k+1}^n-t_k^n-(t_{k+1}^n-t_k^n)\\
&= 0.
\end{aligned}$$

So then the above summation reduces to

$$\sum_{k=0}^{m_n-1}\mathbb{E}([(W(t_{k+1}^n)-W(t_k^n))^2-(t_{k+1}^n-t_k^n)]^2).$$

Note that

$$\mathbb{E}([(W(t_{k+1}^n)-W(t_k^n))^2-(t_{k+1}^n-t_k^n)]^2)=\mathbb{E}([\frac{(W(t_{k+1}^n)-W(t_k^n))^2}{(t_{k+1}^n-t_k^n)}-1]^2[(t_{k+1}^n-t_k^n)]^2).$$

Let $X_k = \frac{(W(t_{k+1}^n)-W(t_k^n))}{(t_{k+1}^n-t_k^n)}$.

Since $W(t)-W(s) \sim N(0,t-s)$, we have that $X_k \sim N(0,1)$.

Then we have

$$\sum_{k=0}^{m_n-1} \mathbb{E}([(W(t^n_{k+1}) - W(t^n_k))^2 - (t^n_{k+1} - t^n_k)]^2) = \sum_{k=0}^{m_n-1} \mathbb{E}([X_k^2 - 1]^2)[(t^n_{k+1} - t^n_k)]^2.$$

Furthermore, we have that X_k is a discrete martingale, since it is just Brownian motion rescaled by the time steps.

Then by the Doob's maximal inequality, we have

$$\begin{aligned}
\mathbb{E}([X_k^2 - 1]^2) &= \mathbb{E}([X_k^4 - 2X_k^2 + 1]) \\
&= 1 + \mathbb{E}([|X_k|^4 - 2|X_k|^2]) \\
&\leq 1 + \mathbb{E}(\max_{1\leq i\leq k}[|X_j|^4)) \\
&\leq 1 + (4/3)^4\mathbb{E}(|X_k|^4)
\end{aligned}$$

Let $C = 1 + (4/3)^4\mathbb{E}(|X_k|^4)$. Then we have

$$\begin{aligned}
\sum_{k=0}^{m_n-1} \mathbb{E}([(W(t^n_{k+1}) - W(t^n_k))^2 - (t^n_{k+1} - t^n_k)]^2) &\leq \sum_{k=0}^{m_n-1} C(t^n_{k+1} - t^n_k)^2 \\
&\leq \sum_{k=0}^{m_n-1} C(t^n_{k+1} - t^n_k)^2 \\
&\leq C\sum_{k=0}^{m_n-1} |P_n|(t^n_{k+1} - t^n_k) \\
&= C|P_n|(T - T_0).
\end{aligned}$$

It follows that

$$\begin{aligned}\lim_{n\to\infty}|P_n| = 0 &\implies \lim_{n\to\infty} C|P_n|(T-T_0) = 0\\ &\implies \lim_{n\to\infty} C|P_n|(T-T_0) = 0\\ &\implies \lim_{n\to\infty}\sum_{k=0}^{m_n-1}\mathbb{E}([(W(t_{k+1}^n)-W(t_k^n))^2-(t_{k+1}^n-t_k^n)]^2) = 0.\end{aligned}$$

So we have

$$\sum_{k=0}^{m_n-1}(W(t_{k+1}^n)-W(t_k^n))^2 \to T-T_0 \; in \; L^2(\Omega).$$

□

2.2.3. Itô's Integral. Itô's integral is built from σ-algebras and sequences that agree with time-steps of stochastic processes known as step processes. We introduce the definitions below.

1. The σ-algebras

   $$\mathcal{W}(t) := \mathcal{U}(W(s)|0 \le s \le t)$$

   and

   $$\mathcal{W}^+(t) := \mathcal{U}(W(s)-W(t)|t \le s)$$

   are respectively called the *history* and *future* of the given Brownian motion.

2. A *filtration* is a *nonanticipating* family $\mathcal{F}(\cdot)$ of σ-algebras. It is nonanticipating with respect to $W(\cdot)$ if

   (a) For all $0 \le s \le t$, $\mathcal{F}(s) \subseteq \mathcal{F}(t)$,

   (b) For all $t \ge 0$, $\mathcal{W}(t) \subseteq \mathcal{F}(t)$,

   (c) For all $t \ge 0$, $\mathcal{F}(t)$ is independent of $\mathcal{W}^+(t)$.

3. Stochasic processes are called nonanticipating (or adapted) if they are measurable with respect to some filtration for all $t \geq 0$. Informally, if the process is jointly measurable with respect to all $\omega \in \Omega$ and $t \geq 0$, it is called *progressively measurable*.

4. $\mathbb{L}^2(0, T)$ is the space of all real-valued, progressively measurable stochastic processes $G(\cdot)$ such that
$$\mathbb{E}(\int_0^T G^2 dt) < \infty.$$

5. Similarly, $\mathbb{L}^1(0, T)$ is the space of all real-valued, progressively measurable stochastic processes $F(\cdot)$ such that
$$\mathbb{E}(\int_0^T |F| dt) < \infty.$$

6. A *step process* is a process $G \in \mathbb{L}^2(0, T)$ that if given a partition P defined as before, we have
$$G(t) \equiv G_k \text{ for } t_k \leq t < t_{k+1}.$$
Note that since $G(\cdot)$ is non-anticipating, we have that G_k is $\mathcal{F}(t_k)$-measurable.

7. The Itô *stochastic* integral is defined to be
$$\int_0^T G dW = \sum_{k=0}^{m-1} G_k(W(t_{k+1} - W(t_k))).$$

Lemma:

If $G \in \mathbb{L}^2(0, T)$, there exists a sequence of bounded step processes $G^n \in \mathbb{L}^2(0, T)$ such that as $m_n \to \infty$,
$$\mathbb{E}(\int_0^T |G - G^n|^2 dt) \to 0.$$

We omit the proof. See [3] p. 68.

Definition:

The *Itô integral* is defined as follows. Let $G \in \mathbb{L}^2(0,T)$ with the same step processes used in the lemma. Then as $n, m \to \infty$, we have

$$\mathbb{E}((\int_0^T G^n - G^m dW)^2) = \mathbb{E}(\int_0^T G^n - G^m dt) \to 0.$$

Thus we have an integral which makes sense in $L^2(\Omega)$. Namely,

$$\int_0^T GdW := \lim_{n\to\infty} \int_0^T G^n dW.$$

Itô Product Rule:

Suppose $dX_1 = F_1 dt + G_1 dW$ and $dX_2 = F_2 dt + G_2 dW$. Then

$$d(X_1 X_2) = X_1 dX_2 + X_2 dX_1 + G_1 G_2 dt.$$

Integrating both sides, we have the *integration by parts* formula

$$\int_s^r X_1 dX_2 = X_1(r)X_2(r) - X_1(s)X_2(s) - \int_s^r X_1 dX_2 - \int_s^r G_1 G_2 dt.$$

Itô Chain Rule:

Let $X(\cdot)$ be a real stochastic process satisfying

$$X(r) = X(s) + \int_s^r F dt + \int_s^r G dW.$$

Then we say $X(\cdot)$ has the stochastic differential

$$dX = F dt + G dW$$

for all times $0 \le s \le r \le T$ with $F \in \mathbb{L}^1(0,T), G \in \mathbb{L}^2(0,T)$.

Let $u : \mathbb{R} \times [0,T] \to \mathbb{R}$ be differenitable in time and twice differenitable in space such that these derivatives are continuous. Then if $Y(t) := u(X(t),t)$, we have

$$dY = du(X(t),t) = u_t dt + u_x dX + \frac{u_{xx}}{2} G^2 dt = (u_t + u_x F + \frac{u_{xx}}{2} G^2) dt + u_x G dW.$$

In the equivalent integral form (which exists by continuity), we have

$$Y(r) - Y(s) = u(X(r),r) - u(X(s),s) = \int_s^r (u_t + u_x F + \frac{u_{xx}}{2} G^2) dt + \int_s^r u_x G dW.$$

We omit the proof but give a sketch. For the full proof, see [3] p. 75-77.

The proof is broken into three steps. First, you consider the function $u(x) = x^m$. Then you consider a separation of variables of $u(x,t)$ into two polynomials. Then Stone-Weierstrass implies that this holds for all u.

2.2.4. Stochastic Differential Equations. *Definitions:*

$$\mathcal{F}(t) = \mathcal{U}(\mathbf{W}(s), \mathbf{X}_0), \text{ where } t \geq 0 \text{ and } 0 \leq s \leq t$$

is the σ- algebra generated by $\mathbf{X}_0 \in \mathbb{R}^n$ and the history of the Brownian motion up to and including time t.

Let $T > 0$ be given, and let the functions $\mathbf{b} : \mathbb{R}^n \times [0,T] \to \mathbb{R}^n, \mathbf{B} : \mathbb{R}^n \times [0,T] \to \mathbb{M}^{n \times m}$ also be given. Then $\forall t \in [0,T], \mathbf{X}(\cdot)$ is a solution of the Itô stochastic differential equation

$$\begin{cases} d\mathbf{X} = \mathbf{b}(\mathbf{X},t)dt + \mathbf{B}(\mathbf{X},t)d\mathbf{W} \\ \mathbf{X}(0) = \mathbf{X}_0 \end{cases}$$

if

1. $\mathbf{X}(\cdot)$ is progressively measurable with respect to $\mathcal{F}(\cdot)$,

2. $\mathbf{F} := \mathbf{b}(\mathbf{X}, t) \in \mathbb{L}^1_n(0, T)$

3. $\mathbf{G} := \mathbf{B}(\mathbf{X}, t) \in \mathbb{L}^2_{n\times m}(0, T)$,

4. $\forall t \in [0, T], \mathbf{X}_t = \mathbf{X}_0 + \int_0^t \mathbf{b}(\mathbf{X}(s), s)ds + \int_0^t \mathbf{B}(\mathbf{X}(s), s)dW$ a.s.

Examples:

Consider the stochastic differential equation

$$\begin{cases} d\mathbf{Y} = \lambda \mathbf{Y} dW \\ \mathbf{Y}(0) = 1, \end{cases}$$

where λ is a constant.

Consider the process $X(\cdot) = W(\cdot)$, and let

$$Y(\cdot) := u(X(t), t) = e^{\lambda X(t) - \frac{\lambda^2 t}{2}}.$$

Since $u_t = \frac{-u_{xx}}{2}$, applying Itô's chain rule yields the stochastic differential

$$du(X(t), t) = u_x dW = \lambda u dW.$$

Since $u(X(0), 0) = 1$, we have a solution process for the given SDE. Note that $u(x, t)$ under Itô's chain rule acts like $e^{\lambda x}$ under the deterministic chain rule.

Now for $k = 0, 1, \ldots$ consider the sequence of functions defined by

$$h_k(x, t) = \frac{(-t)^k}{k!} e^{\frac{x^2}{2t}} \frac{d}{dx^k}(e^{\frac{-x^2}{2t}}).$$

Then without proof, we claim

$$\sum_{k=0}^{\infty} \lambda^k h_k(x, t) = e^{\lambda x - \frac{\lambda^2 t}{2}}.$$

I.e., there is a generating function of $u(x,t)$ expressed in terms of $h_k(x,t)$, which are called *Hermite polynomials.*

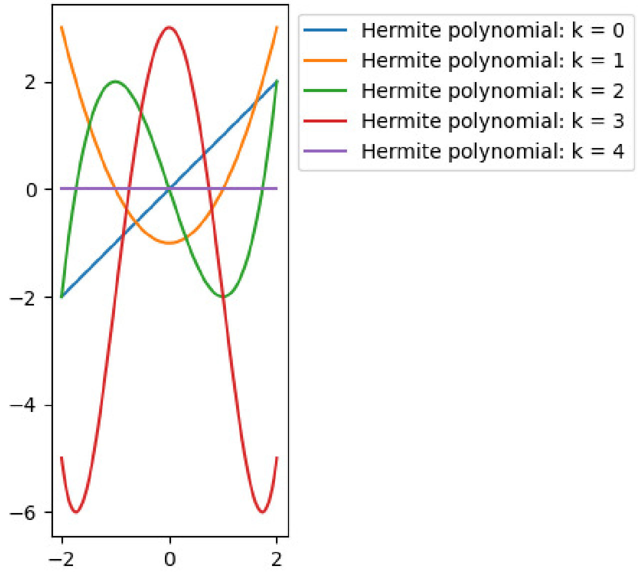

Figure 2.3. Hermite Polynomials

Just as in the Maclaurin expression $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$, we have that the components of the series can be resurviely defined through deterministic differentiation:

$$S_n(x) := \frac{x^n}{n!} \text{ with } \frac{d}{dx} S_n(x) = S_{n-1}(x),$$

a similar notion holds for the Hermite polynomials with respect to Itô's stochastic differentiation.

Theroem:

The stochastic *indefinite Itô integral* is defined to be

$$I(t) := \int_0^t G dw$$

where $G \in \mathbb{L}^2(0,T)$.

Then

$$\int_0^t h_n(W(s),s)dW = h_{n+1}(W(t),t) \text{ or } dh_{n+1}(W(t),t) = h_n(W(t),t)dW.$$

Proof. Consider the process $X(\cdot) = W(\cdot)$, and let

$$Y(\cdot) := u(X(t),t) = e^{\lambda X(t) - \frac{\lambda^2 t}{2}}.$$

Since

$$\begin{cases} d\mathbf{Y} = \lambda \mathbf{Y} dW \\ \mathbf{Y}(0) = 1, \end{cases}$$

we can integrate to obtain

$$Y(t) = 1 + \lambda \int_0^t Y dW \implies \sum_{n=0}^{\infty} \lambda^n h_n(W(t),t) = 1 + \lambda \int_0^t \sum_{n=0}^{\infty} \lambda^n h_n(W(t),t) dW.$$

Using absolute (though we only need uniform) convergence,

$$\lambda \int_0^t \sum_{n=0}^{\infty} \lambda^n h_n(W(t),t)dW = \sum_{n=0}^{\infty} \lambda^n \int_0^t \lambda h_n(W(t),t)dW = \sum_{n=0}^{\infty} \lambda^n \int_0^t h_{n-1}(W(t),t)dW.$$

Since λ is arbitrary, we must have that $h_n(W(t),t) = \int_0^t h_{n-1}(W(t),t)$.

□

Existence and Uniqueness of Solutions:

Uniqueness of solutions to stochastic differential equations means that given two stochastic processes $\mathbf{X}(\cdot)$ and $\mathbf{Y}(\cdot)$ which both solve the given equation, we have that

$$\text{For all } 0 \leq t \leq T, \mathbf{X}(t) = \mathbf{Y}(t) \; a.s.$$

There is a more general proof provided in [3], where the author also allows time dependence on $\boldsymbol{\mu}, \boldsymbol{\sigma}$ defined below.

The n-dimensional stochastic differential equation that describes stocks (with stationary volatility and drift) is given by the geometric Brownian motion and initial condition

$$\begin{cases} d\mathbf{X} = \boldsymbol{\mu}(\mathbf{X})dt + \boldsymbol{\sigma}(\mathbf{X})d\mathbf{W} \\ \mathbf{X}(0) = \boldsymbol{\xi}, \end{cases}$$

where $\boldsymbol{\mu} : \mathbb{R}^n \longrightarrow \mathbb{R}^n, \boldsymbol{\sigma} : \mathbb{R}^n \longrightarrow \mathbb{M}^{m \times n}$ satisfy for some $L \in \mathbb{R}$

$$|\boldsymbol{\mu}(\mathbf{X}_1) - \boldsymbol{\mu}(\mathbf{X}_2)| \leq L|\mathbf{X}_1 - \mathbf{X}_2|$$

and

$$|\boldsymbol{\sigma}(\mathbf{X}_1) - \boldsymbol{\sigma}(\mathbf{X}_2)| \leq L|\mathbf{X}_1 - \mathbf{X}_2|$$

for all $t \in [0, T]$ and $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{R}^n$. Suppose also that

$$|\boldsymbol{\mu}(\mathbf{X})| \leq L|\mathbf{X} + 1|$$

and

$$|\boldsymbol{\sigma}(\mathbf{X})| \leq L|\mathbf{X} + 1|$$

for all $t \in [0, T]$ and $\mathbf{X} \in \mathbb{R}^n$.

Lastly, let $\boldsymbol{\xi}$ be independent of the the future of the m-dimensional Browian motion starting from $t = 0$ and suppose the expected value of $\boldsymbol{\xi}$ is square-integrable.

Then the above SDE has a unique solution $\mathbf{X} \in \mathbb{L}_n^2(0, T)$.

Proof. We have continuity in time and Lipschitz continuity in space, so we will prove existence with successive approximations.

Let $\mathbf{X}^0(t) \equiv \boldsymbol{\xi}$ and define

$$\mathbf{X}^{k+1} := \xi + \int_0^t \boldsymbol{\mu}(\mathbf{X}^k) ds + \int_0^t \boldsymbol{\sigma}(\mathbf{X}^k) d\mathbf{W} \;\; (k = 0, 1, \ldots).$$

Let the expected value of the squared magnitude of differences between increments of the process be denoted by

$$D^k(t) := \mathbb{E}(|\mathbf{X}^{k+1}(t) - \mathbf{X}^k(t)|^2), \;\; k = 0, 1, \ldots.$$

Suppose for all k, $D^k(t) \leq \frac{(At)^{k+1}}{(k+1)!}$ for some $A \in \mathbb{R}$.

Then expanding the quadratic and applying the Cauchy-Schwartz inequality, we have

$$\begin{aligned}
\mathbb{E}(\max_{0 \leq t \leq T} |\mathbf{X}^{k+1}(t) - \mathbf{X}^k(t)|^2) &= \mathbb{E}(\max_{0 \leq t \leq T} | \int_0^t \boldsymbol{\mu}(\mathbf{X}^k(s)) - \boldsymbol{\mu}(\mathbf{X}^{k-1}(s)) ds \\
&\quad + \int_0^t \boldsymbol{\sigma}(\mathbf{X}^k(s)) - \boldsymbol{\sigma}(\mathbf{X}^{k-1}(s)) d\mathbf{W}(s)|^2) \\
&\leq 2\mathbb{E}(\max_{0 \leq t \leq T} | \int_0^t \boldsymbol{\mu}(\mathbf{X}^k(s)) - \boldsymbol{\mu}(\mathbf{X}^{k-1}(s)) ds|^2) \\
&\quad + 2\mathbb{E}(\max_{0 \leq t \leq T} | \int_0^t \boldsymbol{\sigma}(\mathbf{X}^k(s)) - \boldsymbol{\sigma}(\mathbf{X}^{k-1}(s)) d\mathbf{W}(s)|^2) \\
&\leq 2T\mathbb{E}(\max_{0 \leq t \leq T} \int_0^t |\boldsymbol{\mu}(\mathbf{X}^k(s)) - \boldsymbol{\mu}(\mathbf{X}^{k-1}(s))|^2 ds) \\
&\quad + 2\mathbb{E}(\max_{0 \leq t \leq T} | \int_0^t \boldsymbol{\sigma}(\mathbf{X}^k(s)) - \boldsymbol{\sigma}(\mathbf{X}^{k-1}(s)) d\mathbf{W}(s)|^2).
\end{aligned}$$

Continuing with using quadratic variation and the Lipschitz condition, we have

$$
\begin{aligned}
&= 2T\mathbb{E}(\max_{0\le t\le T}\int_0^t |\boldsymbol{\mu}(\mathbf{X}^k(s)) - \boldsymbol{\mu}(\mathbf{X}^{k-1}(s))|^2 ds)\\
&+ 2\mathbb{E}(\max_{0\le t\le T}\int_0^t |\boldsymbol{\sigma}(\mathbf{X}^k(s)) - \boldsymbol{\sigma}(\mathbf{X}^{k-1}(s))|^2 ds)\\
&\le 2TL^2\mathbb{E}(\int_0^T |\mathbf{X}^k(s) - \mathbf{X}^{k-1}(s)|^2 ds)\\
&+ 2\mathbb{E}(\max_{0\le t\le T}\int_0^t |\boldsymbol{\sigma}(\mathbf{X}^k(s)) - \boldsymbol{\sigma}(\mathbf{X}^{k-1}(s))|^2 ds).
\end{aligned}
$$

Then by Doob's maximal inequality,

$$
2\mathbb{E}(\max_{0\le t\le T}\int_0^t |\boldsymbol{\sigma}(\mathbf{X}^k(s)) - \boldsymbol{\sigma}(\mathbf{X}^{k-1}(s))|^2 ds) \le 8L^2\int_0^T \mathbb{E}(|\mathbf{X}^k(s) - \mathbf{X}^{k-1}(s)|^2)ds.
$$

Thus

$$
\begin{aligned}
\mathbb{E}(\max_{0\le t\le T}|\mathbf{X}^{k+1}(t) - \mathbf{X}^k(t)|^2) &\le 2L^2(4+T)\int_0^T \mathbb{E}(|\mathbf{X}^k(s) - \mathbf{X}^{k-1}(s)|^2)ds\\
&= 2L^2(4+T)\int_0^T D^{k-1}(s)ds\\
&\le 2L^2(4+T)T\frac{(AT)^k}{k!}\\
&= C\frac{(AT)^k}{k!}
\end{aligned}
$$

if we take $C = 2L^2(4+T)T$.

Then Chebyshev's inequality ([3] p. 21) tells us

$$
\mathbb{P}(\max_{0\le t\le T}|\mathbf{X}^{k+1}(t) - \mathbf{X}^k(t)|^2 > \frac{1}{2^k}) \le 4^k\mathbb{E}(\max_{0\le t\le T}|\mathbf{X}^{k+1}(t) - \mathbf{X}^k(t)|^2) \le 4^kC\frac{(AT)^k}{k!}.
$$

So we have

$$\sum_{k=1}^{\infty} \mathbb{P}(\max_{0\le t\le T} |\mathbf{X}^{k+1}(t) - \mathbf{X}^{k}(t)|^2 > \frac{1}{2^k}) \le Ce^{4AT}.$$

Therefore by the Borel-Cantelli lemma,

$$\mathbb{P}(\max_{0\le t\le T} |\mathbf{X}^{k+1}(t) - \mathbf{X}^{k}(t)|^2 > \frac{1}{2^k} \text{ i.o. }) = 0.$$

Therefore for a.e. ω, we have uniform convergence on $[0, T]$ as $k \to \infty$ for a telescoping sum

$$\mathbf{X}^k = \mathbf{X}^0 + \sum_{j=0}^{k-1} \mathbf{X}^{j+1}(t) - \mathbf{X}^{j}(t).$$

Let $\mathbf{X}(t) = \lim_{k\to\infty} \mathbf{X}^k(t)$. Then

$$\begin{aligned}
\mathbf{X}(t) &= \mathbf{X}^0 + \lim_{k\to\infty} \sum_{j=0}^{k-1} \mathbf{X}^{j+1}(t) - \mathbf{X}^{j}(t) \\
&= \boldsymbol{\xi} + \lim_{k\to\infty} \sum_{j=0}^{k-1} \int_0^t \boldsymbol{\mu}(\mathbf{X}^j(s))ds + \int_0^t \boldsymbol{\sigma}(\mathbf{X}^j(s))d\mathbf{W} \\
&- (\int_0^t \boldsymbol{\mu}(\mathbf{X}^{j-1}(s))ds + \int_0^t \boldsymbol{\sigma}(\mathbf{X}^{j-1}(s))d\mathbf{W}) \\
&= \boldsymbol{\xi} + \lim_{k\to\infty} \sum_{j=0}^{k-1} \int_0^t \boldsymbol{\mu}(\mathbf{X}^j(s)) - \boldsymbol{\mu}(\mathbf{X}^{j-1}(s))ds \\
&+ \int_0^t \boldsymbol{\sigma}(\mathbf{X}^j(s)) - \boldsymbol{\sigma}(\mathbf{X}^{j-1}(s))d\mathbf{W} \\
&= \boldsymbol{\xi} + \int_0^t \lim_{k\to\infty} \sum_{j=0}^{k-1} \boldsymbol{\mu}(\mathbf{X}^j(s)) - \boldsymbol{\mu}(\mathbf{X}^{j-1}(s))ds \\
&+ \int_0^t \lim_{k\to\infty} \sum_{j=0}^{k-1} \boldsymbol{\sigma}(\mathbf{X}^j(s)) - \boldsymbol{\sigma}(\mathbf{X}^{j-1}(s))d\mathbf{W}.
\end{aligned}$$

Note that we have made use of the uniform continuity for the interchanging of summation and integration. Of course, when $j = 0$, $\boldsymbol{\sigma}(\mathbf{X}^{j-1}(s))$ and $\boldsymbol{\mu}(\mathbf{X}^{j-1}(s))$ do not exist and are not present in the summation, so telescoping and taking the limit yields

$$\mathbf{X}(t) = \boldsymbol{\xi} + \int_0^t \boldsymbol{\mu}(\mathbf{X}(s))ds + \int_0^t \boldsymbol{\sigma}(\mathbf{X}(s))d\mathbf{W}(s).$$

We have proved that there exists a solution under the assumption that for all $k = 0, 1, \ldots$, there exists some $A \in \mathbb{R}$ such that

$$D^k(t) \leq \frac{(At)^{k+1}}{(k+1)!}.$$

We proceed with induction.

Note that in the previous argument with the inequalities on $\mathbb{E}(\max_{0\leq t\leq T} |\mathbf{X}^{k+1}(t) - \mathbf{X}^k(t)|^2)$, we already showed the first line of the following

$$\begin{aligned}
D^k(t) &\leq 2TL^2\mathbb{E}(\int_0^t D^{k-1}(s)ds) + 2\mathbb{E}(\int_0^t |\boldsymbol{\sigma}(\mathbf{X}^k(s)) - \boldsymbol{\sigma}(\mathbf{X}^{k-1}(s))|^2 ds) \\
&\leq 2TL^2\mathbb{E}(\int_0^t D^{k-1}(s)ds) + 2L^2\mathbb{E}(\int_0^t D^{k-1}(s)ds) \\
&= 2L^2(1+T)\mathbb{E}(\int_0^t D^{k-1}(s)ds) \\
&\leq 2L^2(1+T)\mathbb{E}(\int_0^t \frac{(As)^k}{k!}ds) \\
&\leq \frac{(At)^{k+1}}{(k+1)!},
\end{aligned}$$

taking $A \geq 2L^2(1+T)$.

For $k = 0$, it is trivial using the assumptions on $\boldsymbol{\mu}, \boldsymbol{\sigma}$ we have not yet used.

For uniqueness, we have similar inequalities to $D^k(t)$. That is, given two processes $\mathbf{X}_1(t), \mathbf{X}_2(t)$, we have

$$\begin{aligned}\mathbb{E}(|\mathbf{X}_1(t) - \mathbf{X}_2(t)|^2) \leq 2TL^2(\int_0^t \mathbb{E}(|\mathbf{X}_1(s) - \mathbf{X}_2(s)|^2)ds) \\ + 2L^2(\int_0^t \mathbb{E}(|\mathbf{X}_1(s) - \mathbf{X}_2(s)|^2)ds) \\ = 2L^2(1+T)(\int_0^t \mathbb{E}(|\mathbf{X}_1(s) - \mathbf{X}_2(s)|^2)ds).\end{aligned}$$

Since

$$\mathbb{E}(|\mathbf{X}_1(t) - \mathbf{X}_2(t)|^2) \leq 0 + 2L^2(1+T)(\int_0^t \mathbb{E}(|\mathbf{X}_1(s) - \mathbf{X}_2(s)|^2)ds),$$

by Gronwall's inequality, we must have that

$$\mathbb{E}(|\mathbf{X}_1(t) - \mathbf{X}_2(t)|^2) \leq 0e^{(2L^2(1+T))t} = 0.$$

We refer the reader to [3] p.92 for the remainder of the proof that the solution lies in $\mathbb{L}_n^2(0,T)$.

□

3. NEURAL NETWORKS

3.1. INTRODUCTION

The study of neural networks has exploded in the past decade with hundreds of thousands of papers having been written. As such, our presentation of the more recent topics shall be discussed later and only highlighting material relevant to the numerical estimation of high-dimensional PDE. Neural networks are formed from their “hyperparameters”, so one may define the study of the theory of neural networks as the study of these hyperparameters, given some domain. We restrict our attention to the domain of $\mathbb{R}^n$ and give a brief history of results concerning these hyperparameters as well as some notable applications.

3.2. GENERAL STRUCTURE

In its simplest form, a *neural network* $N : \mathbb{R}^P \times \mathbb{R}^n \to \mathbb{R}^m$ is a non-linear mapping from the parameters $\theta \in \mathbb{R}^P$ and an input vector $\mathbf{x} \in \mathbb{R}^n$ to an output vector $\hat{\boldsymbol{y}} \in \mathbb{R}^m$. This mapping uses some chosen *hyperparameters* to construct a solution from a specified *architecture*.

The *architecture hyperparameters* include

1. Depth: Number of hidden layers, H.
2. Widths: Number of parameters L_i within each layer. Note that parameters may also be referred to as neurons, units, or weights.
3. Activation function: A non-linear function $\sigma(\cdot)$ applied component-wise to the output of every layer except the output layer.

We will only be considering fully-connected feedforward neural networks. These neural networks are “feedforward” in the sense that information is fed in one-direction to obtain an output. They are “fully-connected” in the sense that each activated parameter is connected to every parameter in the next layer (there are $L_{i-1}L_i$ parameters here for each W_i). From the depth and widths, we know that $P = \sum_{i=1}^{H} L_{i-1}L_i + L_i$ where the additional L_i term represents the parameters of the bias vectors $\mathbf{b}_i$.

In Figure 3.1, we give the network diagram displaying the architecture of a neural network of depth H with layer widths L_i, we use the following notation to aid in readability. Let $S_j = \sum_{i=1}^{j} L_i$ and $\sigma(\theta_k) = \overline{\theta}_k$. The activated parameters are also indicated by nodes directed from red arrows used to denote $\sigma(\cdot)$. $N(\theta, x) = y$ is the output.

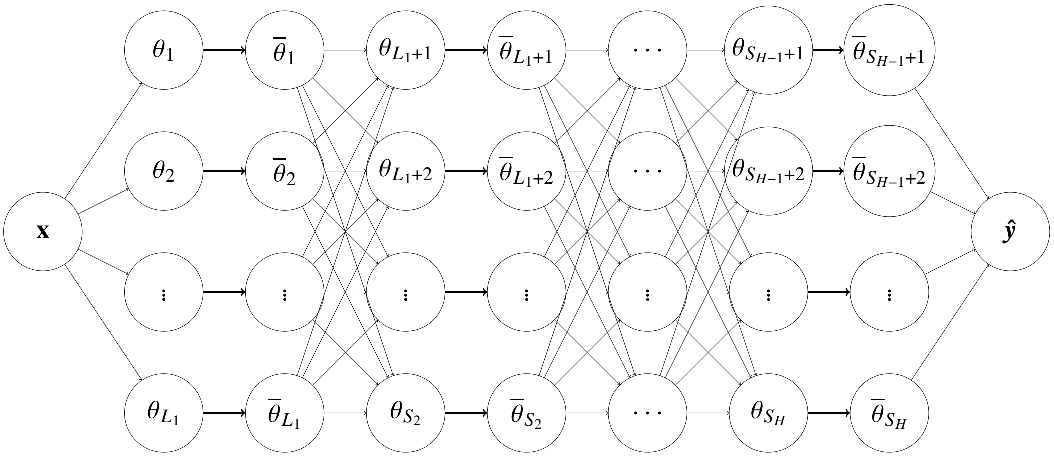

Figure 3.1. A Neural Network with Finite Width and Depth

Given the architecture hyperparameters, we have the remaining *learning hyperparameters*.

1. Loss function: A measurable function $\mathcal{L}$ that compares the target output to the output of the neural network.

2. Optimization algorithm: A mapping from the loss function to the parameters. This is also known as training.

3. Batch size: Amount of training data used in the optimization algorithm.

4. Epochs: Number of iterations used in training.

5. Learning rate: A scalar $0 < \eta < 1$ that sets the magnitude of the adjustment for each iteration.

For each epoch, the neural network performs a forward pass then a backward pass. Let $\theta = \{\theta_k\}_{k=1}^{S_H}$. The forward pass is the mapping simply the mapping $N(\theta, x)$ where

$$N(\theta, \mathbf{x}) = \Lambda_H \circ \sigma \circ \Lambda_{H-1} \circ \ldots \circ \sigma \circ \Lambda_2 \circ \sigma \circ \Lambda_1(\mathbf{x}).$$

For $1 < i \leq L$, where L is the number of layers, $\Lambda_i : \{\theta_k\}_{k=S_{i-2}}^{S_{i-1}} \mapsto \{\theta_k\}_{k=S_{i-1}}^{S_i}$ is a linear mapping which maps the input vector of dimension equal to the width of the previous layer to a vector of dimension equal to the width of the current layer. Specifically, $\lambda_i(\cdot) = W_i(\cdot) + b_i$, where W_i is a weight matrix, and b_i is a bias vector. Note that Λ_1 maps the input vector to a vector of dimension equal to the width of the first layer, and Λ_H is the only hidden layer that does not get activated. $\mathbf{x}$ is sometimes referred to as the *input layer* and $N(\theta, \mathbf{x})$ is sometimes referred to as the *output layer*.

The backward pass is a mapping $\mathcal{L}(N(\theta, \mathbf{x}), \theta) : \mathbb{R}^P \times \mathbb{R}^n \times \mathbb{R}^P \to \mathbb{R}^P$ with the forward pass and parameters as inputs to the loss function and updates the parameters based off the optimization algorithm.

3.2.1. Activation Functions. To list a few activation functions, we have

1. The linear activation function: $\sigma(x) = x$. That is, the identity mapping is the activation function.

2. The logistic/sigmoidal activation function: $\sigma(x) = 1/(1 + e^{-x})$.

3. The rectified linear unit (ReLU) or ramp function: $\sigma(x) = \max\{0, x\}$.

4. The smooth ReLU or softplus: $\sigma(x) = \log(1 + e^x)$.

5. The hyperbolic tangent: $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

6. The arc tangent function: $\sigma(x) = \arctan(x)$.

7. The bipolar function: $\sigma(x) = \text{sign}(x)$ if x is non-zero.

8. The bipolar sigmoid: $\sigma(x) = \frac{1-e^{-x}}{1+e^{-x}}$.

For ease of comparison, we have plotted each of the functions on the same plot in Figure 3.2, but we also include the plots of each individual activation function.

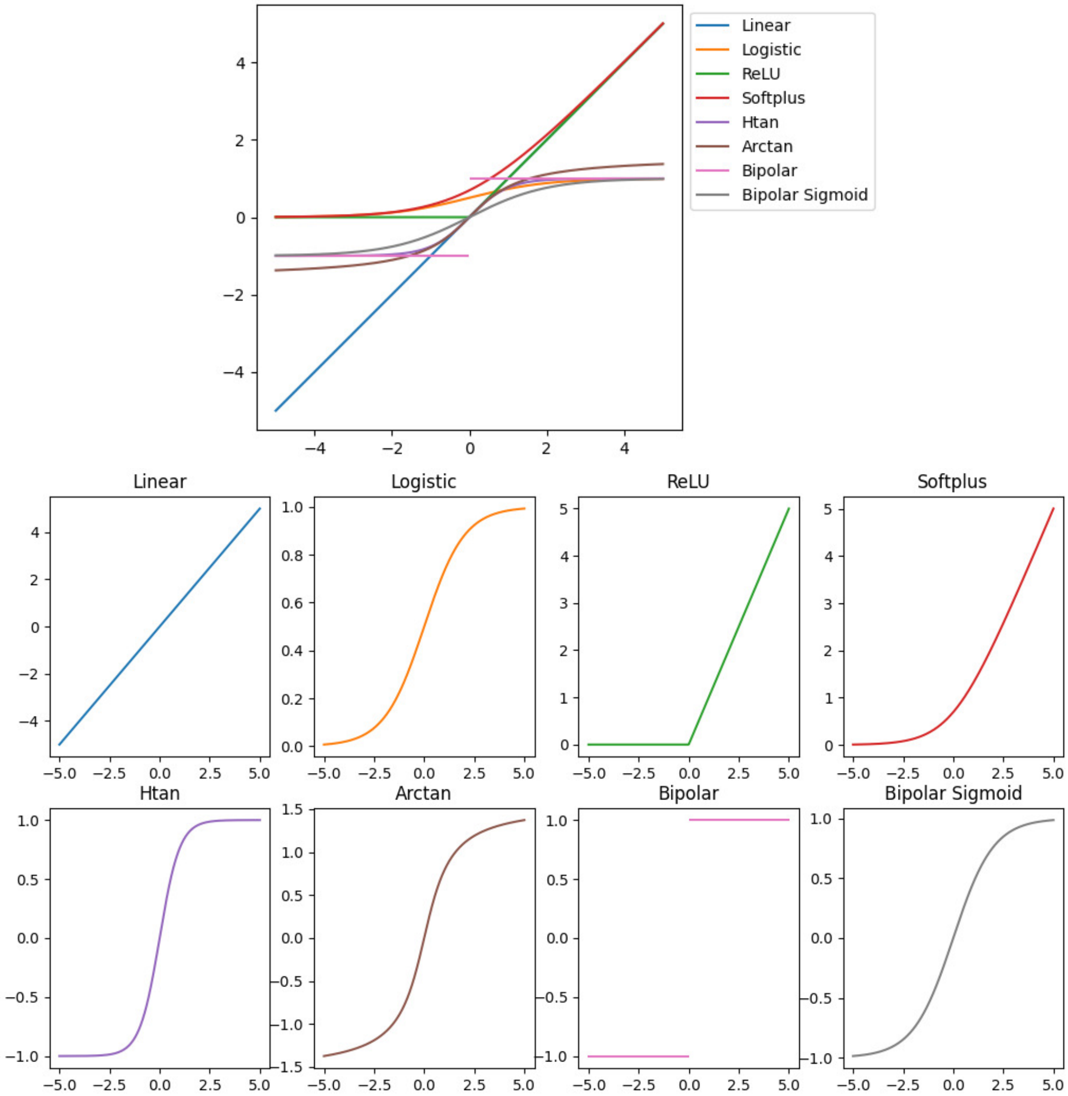

Figure 3.2. Activation Functions

Aside from the logistic activation function, all the S-shaped activation functions may be referred to as "sigmoidal." Except for the linear activation function, all the above functions have found uses in binary classification problems, but may also be used to approximate well-behaved functions. This was proven by Cybenko in [4]. Note that if the activation function between each layer is the linear activation function, then the network is reduced to a single-layer model. The linear activation function is always applied from the last second to last layer to the output layer.

3.2.2. Architectures. The role of architectures in neural networks is still not entirely understood, with many open questions surrounded the theory of infinite depth. However, for shallow neural networks, Cybenko's theorem in [4] was the first to show that two-layer neural networks of infinite width may be used as "universal appproximators." This was then improved by [5] who showed that one-layer networks of infinite-width are universal approximators. Hornik then went on to prove that the choice of activation function does not induce the approximation property, but rather this property is induced by the architecture [6]. For a further discussion of architectures, we refer the reader to [7].

Two notable theorems in neural network theory are the following:

Universal Approximation Theorem:

Cybenko and others went on to find more approximation theorems, culminating in the present-day theorem established in [8]. This theorem roughly states that continuous functions on a compact subset of $\mathbb{R}^n$ can be approximated to arbitrary precision by increasing the number of parameters.

No Free Lunch Theorem:

Developed by Wolbert in [9], this theorem states that when averaged across all data-generating distributions, every algorithm has the same error rate when classifying points outside the data. In some sense, there is no general algorithm that is better than the rest.

3.2.3. Loss Functions. The following loss functions are used for regression problems: *L^2 Loss:*

$$\mathcal{L}(\mathbf{y}, N(\theta, \mathbf{x})) = ||\mathbf{y} - N(\theta, \mathbf{x})||_2^2$$

L^1 Loss:

$$\mathcal{L}(\mathbf{y}, N(\theta, \mathbf{x})) = ||\mathbf{y} - N(\theta, \mathbf{x})||_1^2$$

L^0 *Loss:* This loss function counts the number of non-zero elements in the vector.

The following loss functions are used for classification problems:

Hinge Loss:

$$\mathcal{L}(\mathbf{y}, N(\theta, \mathbf{x})) = \max(0, 1 - \mathbf{y} \cdot N(\theta, \mathbf{x}))$$

Exponential Loss:

$$\mathcal{L}(\mathbf{y}, N(\theta, \mathbf{x})) = e^{-\mathbf{y} \cdot N(\theta, \mathbf{x})}$$

Cross-Entropy Loss:

Let $p(\cdot), q(\cdot)$ be the respective densities of probability distributions $\mathbb{P}, \mathbb{Q}$ with support $\mathcal{X}$. Then cross-entropy between the distributions is given by

$$\mathcal{L}(\mathbf{y}, N(\theta, \mathbf{x})) = -\sum_{\mathbf{x} \in \mathcal{X}} p(\mathbf{x}) \log(q(\mathbf{x})).$$

3.2.4. Optimization Algorithms. We shall only discuss the regime of empirical risk minimization (ERM). That is, when minimizing the loss function, we take an average over the various losses

$$\theta_m^* = \text{argmin}_\theta \frac{1}{m} \sum_{i=1}^{m} L(y_i, N(\theta, \mathbf{x}_i)).$$

We will note that there is also the regime of regularized risk minimization, or (R-ERM) that is ERM with an additional penalty/reward term, but we will not discuss it here. The parameters are trained with

$$\theta_m^* = \arg\min_\theta \frac{1}{m} \sum_{i=1}^{m} L(y_i, N(\theta, \mathbf{x}_i)) + \lambda R(\theta).$$

We will also only be focusing on gradient-based methods. In practice, these methods are very fast to compute because of the autograd algorithm developed in [10] which calculates the gradient of the neural network (with respect to its paramters) based on the computational graph. For more information on computational graphs, see [11].

Mini-batch:

Mini-batching is when the program takes random samples from the data and trains random parameters based off this data. This is more efficient than training on the whole dataset, but we do not explore this here.

3.2.5. Gradient-Based Methods.

Gradient-based methods use the gradient of the loss function with respect to the parameters of the neural network and update the parameters by taking a small step (η_k) in the direction of the parameters that would produce true solution.

Consider a neural network with three hidden layers and the L^2 loss function used for training, as we will use later in the Deep BSDE method. Then

$$\begin{aligned} N(\theta, x) &= \Lambda_3 \circ \sigma \circ \Lambda_2 \circ \sigma \circ \Lambda_1(\mathbf{x}) \\ &= W_3(\sigma(W_2(\sigma(W_1(\mathbf{x}) + \mathbf{b}_1)) + \mathbf{b}_2)) + \mathbf{b}_3. \end{aligned}$$

Thus

$$\begin{aligned} \frac{\partial}{\partial\theta}\mathcal{L}(y, N(\theta, x)) &= \frac{\partial}{\partial\theta}\mathcal{L}(y, N(\theta, x)) \\ &= -2\frac{\partial N(\theta, x)}{\partial\theta} \\ &= -2\frac{\partial}{\partial\theta}[W_3(\sigma(W_2(\sigma(W_1(\mathbf{x}) + \mathbf{b}_1)) + \mathbf{b}_2)) + \mathbf{b}_3]. \end{aligned}$$

Substituting for brevity and using the chain rule, we have

$$\begin{aligned}\frac{\partial}{\partial\theta}\mathcal{L}(y, N(\theta, x)) &= -2\frac{\partial}{\partial\theta}[W_3(\sigma(W_2(\sigma(W_1(\mathbf{x}) + \mathbf{b}_1)) + \mathbf{b}_2)) + \mathbf{b}_3] \\ &= -2[\frac{\partial W_3}{\partial\theta}[\sigma'(\Lambda_2)]\frac{\partial}{\partial\theta}[\sigma(\Lambda_2)] + \frac{\partial\mathbf{b}_3}{\partial\theta}] \\ &= -2[\frac{\partial W_3}{\partial\theta}[\sigma'(\Lambda_2)][\frac{\partial W_2}{\partial\theta}[\sigma'(\Lambda_1)]\frac{\partial}{\partial\theta}[\sigma(\Lambda_1)] + \frac{\partial\mathbf{b}_2}{\partial\theta}] + \frac{\partial\mathbf{b}_3}{\partial\theta}] \\ &= -2[\frac{\partial W_3}{\partial\theta}[\sigma'(\Lambda_2)][\frac{\partial W_2}{\partial\theta}[\sigma'(\Lambda_1)][\frac{\partial W_1}{\partial\theta}[\sigma'(\mathbf{x})] + \frac{\partial\mathbf{b}_1}{\partial\theta}] + \frac{\partial\mathbf{b}_2}{\partial\theta}] + \frac{\partial\mathbf{b}_3}{\partial\theta}].\end{aligned}$$

Seeing as how complicated this can be with more layers, autograd has significantly improved the ease of implementing gradient-based learning algorithms.

```
Algorithm 1: Gradient Descent
    Input: $N(\theta_0, \cdot), \eta_k, K$;
    Output: $\theta_K$;
    for $k \leftarrow 0$ to $K$ do
        Calculate $\nabla_\theta N(\theta, \cdot; )$;
        $\theta_{k+1} = \theta_k - \eta_k \nabla_\theta \mathcal{L}(y, N(\theta, \cdot))$;
    end
```

```
Algorithm 2: Stochastic Gradient Descent (SGD)
    Origin: [12];
    Require: $\varphi_k \sim \mathbb{P}$ i.i.d.;
    Require: $\mathbb{E}(M(\theta_k, \cdot; \varphi_k)) = \nabla_\theta N(\theta_k, \cdot)$.;
    Input: $N(\theta_0, \cdot), \eta_k, K$;
    Output: $\theta_K$;
    for $k \leftarrow 0$ to $K$ do
        Sample $\varphi_k$;
        Calculate $\nabla_\theta M(\theta_k, \cdot; \phi_k)$;
        $\theta_{k+1} = \theta_k - \eta_k \nabla_\theta \mathcal{L}(y, M(\theta, \cdot; \varphi_k))$;
    end
```

Other popular gradient-based methods include

1. Adaptive Moment Estimation (ADAM) developed in [13].

2. Heavy-Ball Method developed in [14].
3. Accelerated Gradient Descent (AGD) developed in [15].

3.3. APPLICATIONS

Neural networks have found countless applications in today's society. Google's AlphaZero is one of the best chess engines in the world, consistently beating Deep Blue, the chess engine that was once considered a point of no return for humans against programs since its famous match Deep Blue vs Kasparov (1997). Games aside, neural networks have aided in reducing pesticides for farmers [16], finding faster matrix multiplication algorithms [17], and finding new protein structures that might have otherwise taken years [18]. Of course, generative a.i. models such as ChatGPT have been growing in popularity ever since GPT-3. In some instances, these models confidently present incorrect information; a problem known as "hallucinations." Despite these issues, popularity surrounding the recent advancements has led to neural networks often being used without rigorous justification of their effectiveness. With artificial general intelligence still in the distant future, it's important to question whether the neural network approach is best for the given task.

Neural networks have found fruitful applications in the following categories of tasks:

1. Classification: The program estimates a function that maps an input to a category.
2. Regression: The program estimates a function that performs a least-squares regression to predict the outcome of a given input.
3. Transcription: Take in unstructured data and output text. For example, reading handwritten digits.
4. Machine Translation: Take in a sequence of symbols of one language and convert it to a sequence of symbols of another language.

5. Anomaly Detection: In this type of task, the program searches for statistical outliers with respect to its training data.

6. Synthesis and sampling: Generation of data given the training data. One example is Stable Diffusion.

7. Imputation of missing values: Given incomplete data, recover the missing values.

8. Denoising: Given corrupted data, recover the clean data.

9. Density estimation: Recover the probability density function of the distribution of the training data.

4. THE DEEP BSDE METHOD

4.1. AN INTRODUCTION TO DEEP BSDE

In [1], the authors present a Deep learning algorithm which uses the Feynman-Kac formula to solve semi-linear PDE. We follow their work but specialize to the one-dimensional case. Their work deals with semi-linear parabolic PDE in their stochastic representation given by

$$\frac{\partial u}{\partial t} + \frac{1}{2}\mathrm{Tr}(\boldsymbol{\sigma}\boldsymbol{\sigma}^T \mathrm{Hess}_x u) + \nabla u \cdot \boldsymbol{\mu} + f = 0,$$

where $u : \mathbb{R}^n \times \mathbb{R} \longrightarrow \mathbb{R}$, $f(t, x, u, \boldsymbol{\sigma}^T \boldsymbol{\nabla} u)$ is a nonlinear function known as the *generator* of the backwards process, and $\boldsymbol{\mu} : \mathbb{R}^n \longrightarrow \mathbb{R}$ and $\boldsymbol{\sigma} : \mathbb{R}^{n \times n} \longrightarrow \mathbb{R}$ satisfy the forward process

$$d\mathbf{X} = \boldsymbol{\mu}(\mathbf{X}(t), t)dt + \boldsymbol{\sigma}(\mathbf{X}(t), t)d\mathbf{W}.$$

Aside from $f, \boldsymbol{\sigma}$, and $\boldsymbol{\mu}$, we are also given T and $u(x, T) = g(x)$ (known as the *terminal condition*) where $t \in [0, T]$. Note that because Brownian motion is nowhere differentiable, the above stochastic differential can only be interpreted in the sense of integrals. That is, satisfying the forward process means we have an *adapted solution*

$$\mathbf{X}(t) = \boldsymbol{\xi} + \int_0^t \boldsymbol{\mu}(\mathbf{X}(s), s)ds + \int_0^t \boldsymbol{\sigma}(\mathbf{X}(s), s)d\mathbf{W}(s).$$

The vector $\mathbf{X}(0) = \boldsymbol{\xi} \in \mathbb{R}^n$ is independent of the filtration associated to the process, and will be chosen. The idea is to approximate $u(0, x)$, along with its gradient at intermediate time steps, by neural networks, which will be trained to minimize the difference between $u(T, x)$ and the given terminal condition. More precisely, we will solve the equation along well-chosen random paths that reduce the PDE to a first-order equation via the Ito chain rule; the loss function will then involve the expected difference between u and the given terminal condition.

Suppose $Y(t) = u(\mathbf{X}(t), t)$ and $\mathbf{Z}(t) = \boldsymbol{\sigma}^T(\mathbf{X}(t), t)\nabla u(\mathbf{X}(t), t)$ hold almost surely. If all the above holds, then the associated backwards process satisfies

$$Y(t) = g(\mathbf{X}(T)) - \int_t^T (\mathbf{Z}(s))^T d\mathbf{W}(s) + \int_t^T f(s, \mathbf{X}(s), \mathbf{Y}(s), \mathbf{Z}(s))ds.$$

There are numerous subtleties associated with forward-backward stochastic differential equations (FBSDEs), but we shall refer the reader to the references mentioned therein the paper we follow for more detail. However, we shall note that the solution to a FBSDE is only adapted to the forward filtration and that one interpretation of FBSDE is as a two-point boundary value problem. The aforementioned description can be found in [19], where they also provide examples of non-solvable FBSDE as well as some conditions for well-posedness (in the stochastic sense). We forgo the details and again refer the reader to the references of [1] when we claim that the backward process in their setting has a unique solution (up to $g(X(T))$) and unique in the sense of SDE. Substituting and reversing the backward process, we have

$$\begin{aligned} u(\mathbf{X}(t), t) = u(\mathbf{X}(0), 0) &+ \int_0^t [\nabla u(\mathbf{X}(s), s)]^T \boldsymbol{\sigma}(\mathbf{X}(s), s) d\mathbf{W}(s) \\ &- \int_0^t f(\mathbf{X}(s), s, u(\mathbf{X}(s), s), \boldsymbol{\sigma}^T(\mathbf{X}(s), s)\nabla u(\mathbf{X}(s), s))ds. \end{aligned}$$

As in [1], we use the Euler-Maruyama method to discretize the stochastic integrals, producing the approximations

$$\mathbf{X}(t_{n+1}) - \mathbf{X}(t_n) \approx \boldsymbol{\mu}(\mathbf{X}(t_n), t_n)\Delta t_n + \boldsymbol{\sigma}(\mathbf{X}(t_n), t_n)\Delta \mathbf{W}_n$$

and

$$u(\mathbf{X}_{t_{n+1}}, t_{n+1}) - u(\mathbf{X}_{t_n}, t_n) \approx -f(u(\mathbf{X}_{t_n}, t_n), t_n, u(\mathbf{X}_{t_n}, t_n), \boldsymbol{\sigma}^T(\mathbf{X}_{t_n}, t_n)\nabla u(\mathbf{X}_{t_n}, t_n))\Delta t_n \\ + [\nabla u(\mathbf{X}_{t_n}, t_n)]^T \boldsymbol{\sigma}(\mathbf{X}_{t_n}, t_n)\Delta \mathbf{W}_n.$$

where

$$\Delta \mathbf{W}_n = \mathbf{W}(t_{n+1}) - \mathbf{W}(t_n), \quad \Delta t_n = t_{n+1} - t_n.$$

Note that this is essentially the stochastic equivalent of constructing a telescoping sum.

With $\boldsymbol{\sigma}, \boldsymbol{\mu}, f, \mathbf{X}(t), t$ and $u(x, T)$ known, if we can estimate ∇u, then we can add all of these pieces together to obtain an approximation to $u(\mathbf{X}, 0)$ as given by the integral formula, for any chosen $\boldsymbol{\xi} = \mathbf{X}(0)$. Using a collection of different $\boldsymbol{\xi}$ inputs, we can construct a pointwise approximation to $u(x, 0)$.

The idea of [1] is to use a deep neural network consisting of a subnetwork for each piece of the telescoping sum to approximate the gradient, given some ξ. The forward iteration initially uses an estimate for the gradient, then uses the previous telescoping sum to compare the estimate $\hat{u}(x, T)$ against the true solution $g(x)$. The loss function $\mathcal{L}(g, \hat{u}) := \mathbb{E}[(g(X_{T_N}) - \hat{u}(\{X_{t_n}\}_{n=1}^N, \{W_{t_n}\}_{n=1}^N))^2]$. Because this loss function uses all the subnetwork approximations, we are simultaneously training all the subnetworks for each iteration, so mini-batching is critical to speed up the computation. However, due to technical limitations, we were unable to implement the mini-batch method into our network.

Since $Y(t) = u(\mathbf{X}(t), t)$, we can rewrite the loss function.

$$\begin{aligned} g(\mathbf{X}(T)) - Y(0) &= \int_0^T (\mathbf{Z}(s))^T d\mathbf{W}(s) - \int_0^T f(s, \mathbf{X}(s), \mathbf{Y}(s), \mathbf{Z}(s)) ds \\ &= u(\mathbf{X}(0), 0) - u(\boldsymbol{\xi}, 0). \end{aligned}$$

Thus

$$\mathcal{L}(g, \hat{u}) := \mathbb{E}[(g(X_{T_N}) - \hat{u}(\{X_{t_n}\}_{n=1}^N, \{W_{t_n}\}_{n=1}^N))^2] = \mathbb{E}[(\hat{u}(\{X_{t_n}\}_{n=1}^N, \{W_{t_n}\}_{n=1}^N) - u(\boldsymbol{\xi}, 0))^2].$$

By using the full range of t values in the BSDE, we are able to approximate the adapted solution at the initial forward position which recovers the value of the solution to the PDE at its initial position. Although we are using the loss function to adjust the gradient, we are using the gradient in the forward iteration in the network to estimate u then using the loss function as a way to compare the estimate to the expected value at the forward iteration.

Despite the high-dimensionality of the problem, the curse of dimensionality is overcome by solving the PDE along a one-dimensional curve estimated by averaging so-called “random characteristics” given by the telescoping equation derived from the reversed BSDE. These are “characteristics” in the sense that we reduce the problem to one-dimension.

4.2. EXPLICIT PDE SOLUTION

In order to implement the ideas of [1], we select specific choices of the functions σ and μ, namely $\boldsymbol{\sigma}(x) = \boldsymbol{\mu}(x) = x$ and $f \equiv 0$ so that our equation has the form

$$u_t = -\nabla \cdot (\frac{x^2}{2}\nabla u)$$

and

$$d\mathbf{X} = \mathbf{X}dt + \mathbf{X}dW.$$

This is the process that governs stock prices with unit volatility and drift.

Note that $\boldsymbol{\sigma}$, $\boldsymbol{\mu}$ are uniformly Lipschitz, so there exists a unique forward solution this SDE as proven previously.

Consider the equation

$$d\mathbf{X} = \boldsymbol{\mu}\mathbf{X}dt + \boldsymbol{\sigma}\mathbf{X}dW.$$

Then we have

$$\boldsymbol{\mu}dt + \boldsymbol{\sigma}dW = \frac{d\mathbf{X}}{\mathbf{X}}.$$

That is, the distribution is log-normally distributed, producing a geometric Brownian motion.

Consider the adapted filtration $\mathbf{Y}(\mathbf{X}(t)) := u(\mathbf{X}(t), t) \equiv \log(\mathbf{X}(t))$.

By the Itô chain rule, in one dimension, we have

$$\begin{aligned} dY &= d\log(X) \\ &= u_t dt + u_x dX + \frac{u_{xx}}{2}(\sigma X)^2 dt \\ &= \frac{dX}{X} - \frac{1}{2X^2}(\sigma X)^2 dt \\ &= \mu dt + \sigma dW - \frac{1}{2}(\sigma)^2 dt \\ &= (\mu - \frac{\sigma^2}{2})dt + \sigma dW. \end{aligned}$$

Therefore

$$\begin{aligned} \int_0^t d\log(X(s))ds &= \log(X(t)) - \log(X(0)) \\ &= \int_0^t (\mu - \frac{\sigma^2}{2})ds + \int_0^t \sigma dW(s) \\ &= (\mu - \frac{\sigma^2}{2})t + \sigma W(t). \end{aligned}$$

So then

$$e^{\log(X(t)) - \log(X(0))} = \frac{X(t)}{\xi} = e^{(\mu - \frac{\sigma^2}{2})t + \sigma W(t)}.$$

Thus,

$$X(t) = \xi e^{(\mu - \frac{\sigma^2}{2})t + \sigma W(t)}.$$

Revisiting the original equation, we know

$$\mathbb{E}(X(t)) = \xi + \int_0^t \mu \mathbb{E}(X(s))ds,$$

so that

$$\mathbb{E}(X(t)) = \xi e^{\mu t}.$$

Our particular solution is

$$X(t) = \xi e^{\frac{t}{2}+W(t)}$$

with

$$\mathbb{E}(X(t)) = \xi e^{t}.$$

Recall our PDE

$$u_t = -\nabla \cdot (\frac{x^2}{2}\nabla u).$$

Then in the one-dimensional case, our equation becomes

$$u_t = -\frac{d}{dx}(\frac{x^2}{2}u_x) = -(\frac{x^2}{2}u_{xx} + xu_x),$$

with $x \in \mathbb{R}$. Separating variables, let $u(x,t) = j(x)h(t)$.

Then we arrive to two ordinary differential equations, with $\lambda \in \mathbb{C}$,

$$h(t) = -\lambda h'(t)$$

and

$$\frac{x^2}{2}j''(x) + xj(x) - \lambda j(x) = 0.$$

So we have a first order constant coefficient ODE, with $h(t) = e^{-\lambda t}$, and an equidimensional equation.

We use the ansatz $j(x) = x^m$ for some $m \in \mathbb{R}$. Then the above equation becomes

$$(\frac{m(m-1)}{2} + m - \lambda)x^m = 0.$$

Since $x \not\equiv 0$, we have $m = \frac{-1\pm\sqrt{1+8\lambda}}{2}$.

We would like to choose λ such that $m \in \mathbb{N}$. Consider $\lambda_n = \frac{n^2+n}{2}$. Then the roots of the quadratic are $n, -n-1$. Because we would not like the solution to blow-up at $x = 0$, we only consider the positive root. Then by the principle of superposition,

$$u(x,t) = \sum_{n=1}^{\infty} C_n x^n e^{-\frac{n^2+n}{2}t} = \sum_{n=1}^{\infty} (C_n^{1/n} x e^{-\frac{n+1}{2}t})^n,$$

where $C_n \in \mathbb{R}$ is dependent on n.

Then we have a geometric series which only converges when $xe^{-\frac{n+1}{2}t}C_n^{1/n} < 1$.

For our particular solution, we take $C_n = 0$ for all $n \neq 2$ and $C_2 = 1$. Therefore, we have the solution

$$e^{-3t}x^2 = u(x,t) \text{ with } e^{-3T}x^2 = u(x,T) = g(x).$$

4.3. A 1-D IMPLEMENTATION

We now list the steps necessary to perform Deep BSDE and provide examples in the one-dimension. We emphasize that we do not claim to have efficient code, but merely correct code. The only packages used for computations were PyTorch and NumPy.

4.3.1. Step 1: Simulate Brownian Motion. We draw from a Bernoulli distribution to sample Brownian motion. Note that in Figure 4.1, the time-axis is partitioned into 90 intervals, but $0 \leq t \leq 1$, and there are 90 sample paths (also called simulations) of Brownian motion. Also note how the approximation $dW \approx \sqrt{dt}$ (due to quadratic variation) is used in the implementation below.

```
#Approximate Brownian motion
```

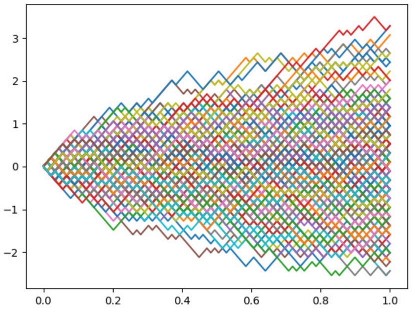

Figure 4.1. Brownian Motions

```
def BM_approx(N,M,T):

    dt = T/(N-1)
    dx = np.sqrt(dt)
    BM = torch.zeros((N, M))

    #random -1 or 1 for matrix of size N x M-1
    steps = 1-2*torch.bernoulli(\
    torch.empty(N, M).uniform_(0, 1))

    for m in range(0,M):
        for n in range(1,N):
            BM[n,m] = BM[n-1,m] + dx*(steps[n-1][m])

    return BM
```

4.3.2. Step 2: Simulate an X-Path. Given samples of Brownian motion, we can construct corresponding sample paths, which we call “X-paths”. Figure 4.2 only includes the plot of the second output of the function of the code below, as the first output is for computational purposes later.

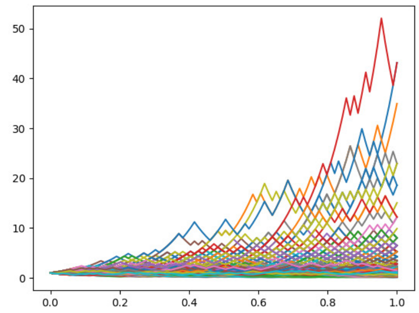

Figure 4.2. Sample Paths

Note that because ξ is independent of the filtration, we initialize each sample path with the given ξ.

```
#Takes in a choice of gamma/sigma/mu and
#produces the X(t) paths at times
#t_n corresponding to a discrete set of omega's
def X_path(N,M,T,xi):


    dt = T/(N-1)
    X = torch.zeros((N, M))
    BM = BM_approx(N,M,T)
    dW = torch.zeros((N, M))
    for m in range(M):
        X[0,m]=xi
        for n in range(1,N):
            dW[n,m] = BM[n,m]-BM[n-1,m]
            X[n,m] = X[n-1,m] + mu(X[n-1,m])*dt \
                + sigma(X[n-1,m])*dW[n,m]

    return dW, X
```

4.3.3. Step 3: Solve the PDE along the X-Paths and Recover ξ. For the following code, we use the sample paths as the training data for the neural network. The forward pass of each subnetwork is that which you would expect, but the forward pass of the whole network uses the telescoping sum mentioned above.

```
def estimate_xi(N,M,T,xi,layer_size,learning_rate,\
    iteration_num):

#Training Data
    dt = T/(N-1)
    X = X_path(N,M,T,xi)
    dW = X[0]
    X_p = X[1]
    true_solution = solution_u(xi,0)
    initialguess = torch.tensor([0.5*true_solution], \
    dtype = torch.float32)
    true_grad = grad_u(xi,0)
    initialgradguess = torch.tensor([0.5*true_grad], \
    dtype = torch.float32)

#Stack of Neural Networks
    class Subnet(torch.nn.Module):
            def __init__(self, num_neurons_per_layer):
                super().__init__()
                self.fc1 = torch.nn.Linear(1, \
                   num_neurons_per_layer)
                self.fc2 = torch.nn.Linear(\
                   num_neurons_per_layer, \
                   num_neurons_per_layer)
                self.fc3 = torch.nn.Linear(\
                   num_neurons_per_layer, 1)

            def forward(self, x):
                x = self.fc1(x)
                #x = torch.nn.functional.relu(x)
                x = torch.nn.functional.sigmoid(x)
                x = self.fc2(x)
                #x = torch.nn.functional.relu(x)
                x = torch.nn.functional.sigmoid(x)
                x = self.fc3(x)
                return x


    class Net(torch.nn.Module):
        def __init__(self, num_neurons_per_layer):
            super().__init__()
            self.subnets = torch.nn.ModuleList(\
               [Subnet(num_neurons_per_layer) \
               for i in range(N)])
            self.soln_t0 = torch.nn.Parameter(initialguess)
            self.grad_u_0_xi = torch.nn.Parameter(\
               initialgradguess)

        def forward(self):
            dt = T/(N-1)
            U = self.soln_t0.tile(M)
            G_U = self.grad_u_0_xi.tile(M)
            dW, X = X_path(N,M,T, xi)
```

```
            U = U + (G_U*dW[0])

            for n in range(1,N):
                G_U = self.subnets[n-1](\
                 torch.reshape(X[n-1],(M,1)))
                U = U + (G_U*dW[n])
            return U

    net = Net(layer_size)
    optimizer = torch.optim.SGD(\
       net.parameters(), lr=learning_rate)

    loss_fcn = torch.nn.MSELoss()

    for n in range(iteration_num):
        optimizer.zero_grad()

        y =  net()

        loss = loss_fcn(g(X_p[-1],T), y)

        loss.backward()

        optimizer.step() # apply gradients

return net.soln_t0.item()
```

4.3.4. Step 4: Repeat 1-4 Until $u(x, 0)$ is Recovered. For Figure 4.3, we ran 40 estimations of ξ with 90 time steps, 300 simulations of X-paths, a learning rate $\eta = 0.001$, no momentum, $T = 1$ and 2000 iterations per subnetwork. It is worth noting that the training data consisted of the same number of simulations and time-steps as the previous plots.

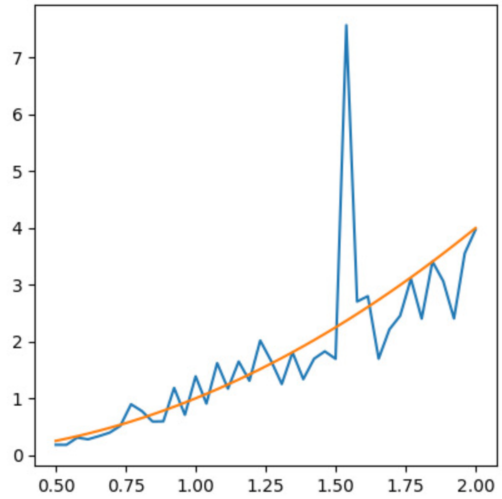

Figure 4.3. A Pointwise Reconstruction of $u(x, 0)$

4.4. CONCLUSION

The increase in computational power of computers has led to the design of algorithms that were once impractical. The accuracy of the Deep BSDE method relies on a small number of parameters that, in some cases, require high values. But due to the computation time only increasing linearly, this is still more efficient than using a nonlinear method. If we had used 100 dimensions instead of one dimension or increase the number of neural networks, the average error of the approximations of each of the $u(\xi, 0)$ values should be closer to zero. Perhaps with the implementation of the mini-batch method, we could have used more iterations and obatined a solution that was computed faster. But the accuracy of the program would not have changed much as the neural networks would still have converged to the same solutions generated by the same stochastic training data (given a random seed).

However, implementing mini-batch would have allowed us to train more neural networks in the same amount of time, thereby indirectly increasing the accuracy if given the same amount of training time.

APPENDIX

1. MEASURE THEORY

This section provides background information needed to understand the probability theory. We followed the work of [20] and refer the reader to that text for more details.

1.1. EXISTENCE OF THE BOREL σ-ALGEBRA. *Definition:* Given a set Ω, a collection $\mathcal{U}$ of Ω is called a σ-algebra (of subsets of X) provided

1. $\emptyset \in \mathcal{U}$.
2. If $\Omega \supset A \in \mathcal{U}$, then $A^c \in \mathcal{U}$. Note that $A^c = \Omega - A = \{x | x \in \Omega, x \notin A\}$ is the set of all elements in Ω that are not a member of A. Here, $-$ denotes the operation of set difference.
3. The union of a countable collection of sets in $\mathcal{U}$ also belongs to $\mathcal{U}$.

De Morgan's Identities: Let $\Omega \supseteq \cup_{i \in \mathbb{N}} B_i$ for some subsets B_i of Ω. Then

$$\Omega - \cup_{i \in \mathbb{N}} B_i = \cap_{i \in \mathbb{N}} [\Omega - B_i] \text{ and } \Omega - \cap_{i \in \mathbb{N}} B_i = \cup_{i \in \mathbb{N}} [\Omega - B_i].$$

From these identities, we can see that σ-algebras are closed with respect to the operations of intersection and union of open sets.

Proposition: Let $\mathcal{S}$ be a collection of subsets of Ω. Then the intersection $\mathcal{U}$ of all σ-algebras of subsets of Ω that contain $\mathcal{U}$ is a σ-algebra that contains $\mathcal{U}$. Moreover, it is the smallest σ-algebra of subsets of Ω that contains $\mathcal{S}$ in the sense that any σ-algebra that contains $\mathcal{S}$ also contains $\mathcal{U}$.

Let $\{A_n\}_{n=1}^{\infty}$ be a countable collection of sets that belong to $\mathcal{U}$. By closure under intersections and unions, we must have that

$$\limsup\{A_n\}_{n=1}^{\infty} = \bigcap_{k=1}^{\infty}[\bigcup_{n=k}^{\infty} A_n], \liminf\{A_n\}_{n=1}^{\infty} = \bigcup_{k=1}^{\infty}[\bigcap_{n=k}^{\infty} A_n] \in \mathcal{U}.$$

Definition: The collection $\mathcal{B}$ of Borel sets of real numbers is the smallest σ-algebra of sets of real numbers that contains all of the open sets of real numbers. Every open set and every closed set is a Borel set. A countable intersection of open sets is called a G_δ set while a countable union of closed sets is an F_σ set. The aforementioned lim inf and lim sup sets are also Borel sets. $\mathcal{B}$ is the Borel σ-algebra.

1.2. GENERAL MEASURE SPACES. A *Measurable space* is a couple $(\Omega, \mathcal{U})$, where Ω is a set and $\mathcal{U}$ is a σ-algebra on subsets of Ω. A subset A of Ω is called *measurable* provided $A \in \mathcal{U}$.

Definition: A *measure* $\mathbb{P}$ on a measurable space $(\Omega, \mathcal{U})$ is a set function $\mathbb{P}$ (function which maps a collection of sets to the extended real numbers) such that $\mathbb{P}(\emptyset) = 0$ and $\mathbb{P}$ is countably additive, i.e. if $\{A_k\}_{k\in\mathbb{N}}$ is a collection of disjoint sets, then $\mathbb{P}(\cup_{k\in\mathbb{N}} A_k) = \sum_{k=1}^{\infty} \mathbb{P}(A_k)$.

Note: The use of this notation for the measure space is typically reserved for probability spaces, which we define later for stochastic differential equations.

Then a *measure space* is a triple $(\Omega, \mathcal{U}, \mathbb{P})$, where $\mathbb{P}$ is a measure on a σ-algebra $\mathcal{U} \subseteq \Omega$. We have the following properties:

1. Finite Additivity: $\{A_k\}_{k=1}^{n}$ is a collection of disjoint sets, then $\mathbb{P}(\cup_{k=1}^{n} A_k) = \sum_{k=1}^{n} \mathbb{P}(A_k)$.

2. Monotonicity: If A and B are measurable $A \subseteq B$, then $\mathbb{P}(A) \leq \mathbb{P}(B)$

3. Excision: If $A \subseteq B$ and $\mathbb{P} < \infty$, then $\mathbb{P}(B - A) = \mathbb{P}(B) - \mathbb{P}(A)$.

4. Countable monotonicity: For any countable collection $\{A_k\}_{k=1}^{\infty}$ of measurable functions that covers a measurable set A,

$$\mathbb{P}(A) \leq \sum_{k=1}^{\infty} \mathbb{P}(A_k).$$

Proof. From the property of countable additivity, we know that if $\{A_k\}_{k\in\mathbb{N}}$ is a collection of disjoint sets, then $\mathbb{P}(\cup_{k\in\mathbb{N}} A_k) = \sum_{k=1}^{\infty} \mathbb{P}(A_k)$. Then if given a finite subset $\{A_k\}_{k\in\mathbb{N}}$, say $\{A_k\}_{k=1}^{n}$, we can, without loss of generality, let $\{A_k\}_{k=n+1}^{\infty}$ consist entirely of empty sets. Then

$$\mathbb{P}(\cup_{k\in\mathbb{N}} A_k) = \sum_{k=1}^{\infty} \mathbb{P}(A_k) = \sum_{k=1}^{n} \mathbb{P}(A_k) + \sum_{k=n+1}^{\infty} \mathbb{P}(A_k) = \sum_{k=1}^{n} \mathbb{P}(A_k) = \mathbb{P}(\cup_{k=1}^{n} A_k).$$

Then from finite additivity, we have that if $A \subseteq B$

$$\mathbb{P}(B) = \mathbb{P}(A) + \mathbb{P}(B - A),$$

as the right-hand-side consists of distjoint sets. This implies the excision property. Since $\mathbb{P}(B - A) \geq 0$, this also implies the monotonicity property.

For the countable monotonicity property, let $C_1 := A_1$ and for all $n > 1$, define $C_n = A_n - \cup_{k=1}^{n-1}$. Then C_n is disjoint from C_i for all $i < n$. So $\{C_k\}_{k=1}^{\infty}$ is a collection of disjoint sets. Also note that $\forall n, C_n \subseteq A_n$ but $\cup_{k=1}^{\infty} C_n = \cup_{k=1}^{\infty} A_n$. Then we have by monotonicity then countable additivity,

$$\mathbb{P}(A) \leq \mathbb{P}(\cup_{k=1}^{\infty} A_n) = \mathbb{P}(\cup_{k=1}^{\infty} C_n) = \sum_{k=1}^{\infty} \mathbb{P}(C_k) \leq \sum_{k=1}^{\infty} \mathbb{P}(A_k).$$

□

Continuity of Measure: Let $(\Omega, \mathcal{U}, \mathbb{P})$ be a measure space. Then we say a sequence of sets $\{A_k\}$ are *ascending* provided $A_k \subset A_{k+1}$ and *descending* if $A_{k+1} \subset A_k$.

1. If $\{A_k\}_{k=1}^{\infty}$ is ascending, then

$$\mathbb{P}(\cup_{k=1}^{\infty} A_k) = \lim_{k \to \infty} \mathbb{P}(A_k).$$

2. If $\{A_k\}_{k=1}^{\infty}$ is descending, then

$$\mathbb{P}(\cap_{k=1}^{\infty} A_k) = \lim_{k \to \infty} \mathbb{P}(A_k).$$

Proof. First we prove the first assertion for ascending sets. Suppose $\exists j : \mathbb{P}(A_j) = \infty$. Then by monotonicity, $\mathbb{P}(\cup_{k=1}^{\infty} A_k) = \infty = \lim_{k\to\infty} \mathbb{P}(A_k)$. Now suppose $\forall k, \mathbb{P}(A_k) < \infty$. Then define $A_0 = \emptyset$, and let $C_k = A_k - A_{k-1}$. Since $\{A_k\}_{k=1}^{\infty}$ is ascending, $C_k = A_k - \cup_{k=1}^{\infty} A_k$. By a previous argument, $\{C_k\}_{k=1}^{\infty}$ is disjoint and $\cup_{k=1}^{\infty} A_k = \cup_{k=1}^{\infty} C_k$. Thus by countable additivity of $\mathbb{P}$ then the excision property, we have

$$\mathbb{P}(\cup_{k=1}^{\infty} A_k) = \mathbb{P}(\cup_{k=1}^{\infty} C_k) = \sum_{k=1}^{\infty} \mathbb{P}(A_k - A_{k-1}) = \sum_{k=1}^{\infty} \mathbb{P}(A_k) - \mathbb{P}(A_{k-1}) =$$

$$\lim_{n\to\infty} \sum_{k=1}^{n} \mathbb{P}(A_k) - \mathbb{P}(A_{k-1}) = \lim_{n\to\infty} (\mathbb{P}(A_n) - \mathbb{P}(A_0)) = \lim_{n\to\infty} \mathbb{P}(A_n).$$

Next, we prove the second assertion for descending sets. Let $\{A_k\}_{k=1}^{\infty}$ be descending. Define for all k, $C_k = A_1 - A_k$. Since the sequence is descending,

$$C_k = A_1 - \cap_{n=k}^{\infty} A_k \subset A_1 - \cap_{n=k+1}^{\infty} A_k = C_{k+1},$$

so C_k is ascending.

It follows that

$$\mathbb{P}(\cup_{k=1}^{\infty} C_k) = \lim_{k\to\infty} \mathbb{P}(C_k).$$

By De Morgan's identities,

$$\cup_{k=1}^{\infty} C_k = \cup_{k=1}^{\infty} A_1 - A_k = A_1 - \cap_{k=1}^{\infty} A_k.$$

By the excision property,

$$\mathbb{P}(A_1 - \cap_{k=1}^{\infty} A_k) = \lim_{n\to\infty} (\mathbb{P}(A_1) - \mathbb{P}(A_n)).$$

Note that

$$\lim_{n\to\infty} (\mathbb{P}(A_1) - \mathbb{P}(A_n)) = \lim_{k\to\infty} \mathbb{P}(C_k) = \mathbb{P}(A_1 - \cap_{k=1}^{\infty} A_k).$$

Then by excision, we have

$$\lim_{n\to\infty} (\mathbb{P}(A_n)) = \mathbb{P}(\cap_{k=1}^{\infty} A_k).$$

□

Borel-Cantelli Lemma:

Let $(\Omega, \mathcal{U}, \mathbb{P})$ be a measure space and $\{A_k\}_{k=1}^{\infty}$ a countable collection of measurable sets such that $\sum_{k=1}^{\infty} \mathbb{P}(A_k) < \infty$. Then *almost all* $\omega \in \Omega$ belong to at most a finite number of A_k's. Note that we say that the collection

$$\cap_{k=1}^{\infty} \cup_{m=k}^{\infty} A_m = \{\omega \in \Omega | \omega \text{ belongs to infinitely many of the } A_k\}$$

is called "A_n *infinitely often*" (A_n i.o.). We also say that a property of a subset $A \subset \Omega$ holds *almost everywhere* on A provided it holds on $A - A_0 : \mathbb{P}(A_0) = 0$. If $\mathbb{P}$ is a probability measure, we say that this property holds *almost surely*.

Proof. By continuity of $\mathbb{P}$ then by countable monotonicity, we have

$$\mathbb{P}(\cap_{k=1}^{\infty} \cup_{m=k}^{\infty} A_m) = \lim_{n\to\infty} \mathbb{P}(\cup_{k=n}^{\infty} A_k) \leq \lim_{n\to\infty} \sum_{k=n}^{\infty} \mathbb{P}(A_k) = 0.$$

□

Therefore the Borel-Cantelli lemma tells us that if A_n i.o., then $\mathbb{P}(A_n \text{i.o.}) = 0$.

Definitions: Let $(\Omega, \mathcal{U}, \mathbb{P})$ be a measure space.

1. If $\mathbb{P}(\Omega) < \infty$, the measure $\mathbb{P}$ is called finite.

2. If $\Omega = \cup_{k=1}^{\infty} A_k : \forall k, \mathbb{P}(A_k) < \infty$, then we say $\mathbb{P}$ is σ-finite.

3. The previous definitions may also be applied to measurable sets, provided the sets satisfy the corresponding properties with respect to $\mathbb{P}$.

4. The measure space is said to be *complete* provided $\mathcal{U}$ contains all subsets of sets of measure 0.

Proposition: Every measure space can be completed. Let $(\Omega, \mathcal{U}, \mathbb{P})$ be a measure space. Define $\mathcal{U}_0$ to be the collection of subsets A of the form $A = B \cup C$ where $C \in \mathcal{U}$ and $B \subset D$ for some $D \in \mathcal{U}$ such that $\mathbb{P}(D) = 0$. Let $\mathbb{P}_0$ be a measure on $\mathcal{U}_0$ such that $\mathbb{P}_0(A) = \mathbb{P}(C)$. Then $\mathcal{U}_0$ is a σ-algebra that contains $\mathcal{U}$, $\mathbb{P}_0$ is a measure that extends $\mathbb{P}$, and $(\Omega, \mathcal{U}_0, \mathbb{P}_0)$ is a complete measure space.

We omit the proof but refer the reader to [21] theorem 1.36.

1.3. MEASUREABLE FUNCTIONS. *Proposition*: Let $(\Omega, \mathcal{U})$ be a measurable space and let X be an extended real-valued function defined on Ω. Then the following equivalent statements hold:

1. $\forall c \in \mathbb{R}, \{\omega \in \Omega | X(\omega) < c\} \in \mathcal{U}(\Omega)$.
2. $\forall c \in \mathbb{R}, \{\omega \in \Omega | X(\omega) \leq c\} \in \mathcal{U}(\Omega)$.
3. $\forall c \in \mathbb{R}, \{\omega \in \Omega | X(\omega) > c\} \in \mathcal{U}(\Omega)$.
4. $\forall c \in \mathbb{R}, \{\omega \in \Omega | X(\omega) \geq c\} \in \mathcal{U}(\Omega)$.

If any of the above conditions hold, then we have $\forall c \in \mathbb{R}, \{\omega \in \Omega | X(\omega) = c\} \in \mathcal{U}(\Omega)$.

Proof. Suppose Ω is measurable. Then $\Omega \in \mathcal{U} \implies \Omega^c \in \mathcal{U}$. Therefore, statements (1) and (4) are equivalent, and statements (2) and (3) are equivalent. So it suffices to show that (1) $\iff$ (2) holds.

Suppose (1) is true. Then

$$\{\omega \in \Omega | X(\omega) \leq c\} = \bigcap_{k=1}^{\infty} \{\omega \in \Omega | X(\omega) < c + 1/k\}.$$

A countable intersection of measurable sets is measurable, so (1) $\implies$ (2).

Suppose (2) is true. Then

$$\{\omega \in \Omega | X(\omega) < c\} = \bigcup_{k=1}^{\infty} \{\omega \in \Omega | X(\omega) \leq c - 1/k\}.$$

A countable union of measurable sets is measurable, so (2) $\implies$ (1).

The intersection of two measurable sets is measurable, so we have

$$\forall c \in \mathbb{R}, \{\omega \in \Omega | X(\omega) \leq c\} \cap \{\omega \in \Omega | X(\omega) \geq c\} = \{\omega \in \Omega | X(\omega) = c\} \in \mathcal{U}(\Omega).$$

If $c = \pm\infty$, it can be shown that $\{\omega \in \Omega | X(\omega) = c\} \in \mathcal{U}$.

□

Definition: If a function f satisfies one of the above conditions and its domain is measurable, then f is measurable.

Proposition:

f is measurable if and only if for each $B \in \mathscr{B}$, where $\mathscr{B}$ denotes the Borel σ-algebra, we have $f^{-1}(B) \in \mathcal{U}$. Note that because $\mathscr{B}$ is the smallest σ-algebra of open sets of $\mathbb{R}^n$, we must have that $\mathscr{B} \subset \mathcal{U} \subseteq \Omega$.

Proof. Suppose f is measurable and $B = \bigcup_{i=1}^{\infty} I_i$ is a union of open, bounded intervals such that

$$I_i = S_i \cap T_i \text{ where } S_i = (s_i, \infty), \quad T_i = (-\infty, s_i).$$

Since $S_i, T_i \subset I_i$, which is bounded, and f is measurable, $f^{-1}(S_i) \cap f^{-1}(T_i)$ is measurable. Therefore

$$f^{-1}(B) = f^{-1}(\bigcup_{i=1}^{\infty} S_k \cap T_k) \text{ is measurable.}$$

On the other hand, suppose for every open B, $f^{-1}(B) = \{\omega \in \Omega | f(x) \in \mathscr{B}\}$ is measurable. Then we can express B as a union of open intervals, so that we have a set defined by a function over a union of measurable sets. So f is measurable.

□

1.4. RIEMANN INTEGRATION. Let f be a bounded real-valued function on the closed, bounded interval $[a, b]$ that has a *partition* P, where P is a totally-ordered set $P = \{x_0, x_1, \ldots, x_n\}$ where $a = x_0 < x_1 < \ldots < x_n = b$.

Then we have upper and lower Riemann sums respectively defined by

$$U(f,P) = \sum_{i=1}^{n} M_i(x_i - x_{i-1}), \text{ where } M_i = \sup_{x\in(x_{i-1},x_i)}\{f(x)\}$$

and

$$L(f,P) = \sum_{i=1}^{n} m_i(x_i - x_{i-1}), \text{ where } m_i = \inf_{x\in(x_{i-1},x_i)}\{f(x)\}.$$

These notions naturally lead to the notion of upper and lower Riemann integrals, defined respectively by

$$\overline{\int_a^b} f(x)\,\mathrm{d}x = \sup_P\{L(f,P)\}$$

and

$$\underline{\int_a^b} f(x)\,\mathrm{d}x = \inf_P\{U(f,P)\}.$$

Then we say f is *Riemann integrable* provided

$$\overline{\int_a^b} f(x)\,\mathrm{d}x = \underline{\int_a^b} f(x)\,\mathrm{d}x$$

and we just write

$$\int_a^b f dx.$$

REFERENCES

[1] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, aug 2018. doi: 10.1073/pnas.1718942115. URL `https://doi.org/10.1073%2Fpnas.1718942115`.

[2] Jared Chessari, Reiichiro Kawai, Yuji Shinozaki, and Toshihiro Yamada. Numerical methods for backward stochastic differential equations: A survey. *Probability Surveys*, 20(none), jan 2023. doi: 10.1214/23-ps18. URL `https://doi.org/10.1214%2F23-ps18`.

[3] Lawrence C. Evans. *An Introduction to Stochastic Differential Equations*. American Mathematical Society, 2013.

[4] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, Dec 1989. ISSN 1435-568X. doi: 10.1007/BF02551274. URL `https://doi.org/10.1007/BF02551274`.

[5] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(89)90020-8. URL `https://www.sciencedirect.com/science/article/pii/0893608089900208`.

[6] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(91)90009-T. URL `https://www.sciencedirect.com/science/article/pii/089360809190009T`.

[7] Julius Berner, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. The modern mathematics of deep learning. *arXiv preprint arXiv:2105.04026*, 2021.

[8] Patrick Kidger and Terry J. Lyons. Universal approximation with deep narrow networks. *CoRR*, abs/1905.08539, 2019. URL `http://arxiv.org/abs/1905.08539`.

[9] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.*, 1:67–82, 1997.

[10] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*, 2017. URL `https://openreview.net/forum?id=BJJsrmfCZ`.

[11] Aaron Courville Ian Goodfellow, Yoshua Bengio. *Deep Learning*. MIT Press, 2017.

[12] Herbert E. Robbins. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] Boris Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr Computational Mathematics and Mathematical Physics*, 4:1–17, 12 1964. doi: 10.1016/0041-5553(64)90137-5.

[15] Yurii Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983.

[16] Tanha Talaviya, Dhara Shah, Nivedita Patel, Hiteshri Yagnik, and Manan Shah. Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides. *Artificial Intelligence in Agriculture*, 4:58–73, 2020. ISSN 2589-7217. doi: https://doi.org/10.1016/j.aiia.2020.04.002. URL `https://www.sciencedirect.com/science/article/pii/S258972172030012X`.

[17] Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis, and Pushmeet Kohli. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022. doi: 10.1038/s41586-022-05172-4.

[18] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, Augustin Žídek, Tim Green, Kathryn Tunyasuvunakool, Stig Petersen, John Jumper, Ellen Clancy, Richard Green, Ankur Vora, Mira Lutfi, Michael Figurnov, Andrew Cowie, Nicole Hobbs, Pushmeet Kohli, Gerard Kleywegt, Ewan Birney, Demis Hassabis, and Sameer Velankar. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Research*, 50(D1):D439–D444, 11 2021. ISSN 0305-1048. doi: 10.1093/nar/gkab1061. URL `https://doi.org/10.1093/nar/gkab1061`.

[19] Jiongmin Yong Jin Ma. *Forward-Backward Stochastic Differential Equations and their Applications*. Springer, 2007.

[20] P.M. Fitzpatrick H.L. Royden. *Real Analysis*. Pearson Education, Inc., 2010.

[21] Walter Rudin. *Real and Complex Analysis*. McGraw-Hill Science/Engineering/Math, May 1986. ISBN 0070542341.

VITA

Daniel G. Kovach II was born in St. Louis, MO on March 10, 1999. They attended the University of Alabama from Fall 2017 to Spring 2021 with a Presidential Scholarship. In Fall 2018, they were named to the Dean's List, and in Spring 2020, they were named to the President's List. In May 2021, they graduated with honors, obtaining a B.S. in mathematics with a concentration in statistics and minor in economics. In August 2021, they attended the Missouri University of Science and Technology, where they worked as a graduate teaching assistant from Spring 2022 to Spring 2023 while studying as a graduate student of applied mathematics. In the Summer of 2023, they then worked as a graduate research assistant. In December 2023, they received their M.S. in applied mathematics from the Missouri University of Science and Technology.