FIRST EDITION

MATHEMATICAL FOUNDATIONS OF NEURAL NETWORKS

From CNNs to PINNs, QINNs, and Deep Reinforcement Learning

RAPHAEL NORIEGA

Contents

Contents iii

I. Mathematical Preliminaries 1

1. Introduction 3
1.1. Historical context of AI and neural networks . . . 3
1.2. AI winters and the deep learning revolution . . . 3
1.3. Why mathematics matters in deep learning . . . 3

2. Linear Algebra Essentials 5
2.1. Vector spaces and inner products . . . 5
2.2. Eigenvalues, eigenvectors, diagonalization . . . 5
2.3. Singular value decomposition (SVD) . . . 5
2.4. Tensor notation and operations . . . 5

3. Multivariable Calculus and Analysis 7
3.1. Gradients, Jacobians, Hessians . . . 7
3.2. Taylor expansions in multiple variables . . . 7
3.3. Divergence, curl, and Laplacians . . . 7
3.4. Variational principles . . . 7

4. Probability, Statistics, and Information Theory 9
4.1. Random variables and distributions . . . 9
4.2. Expectation, variance, covariance . . . 9
4.3. Gaussian and exponential families . . . 9
4.4. Entropy, KL divergence, mutual information . . . 9

5. Optimization Theory 11
5.1. Convexity, duality, and Lagrangians . . . 11
5.2. Gradient descent and its variants . . . 11
5.3. Stochastic optimization and convergence . . . 11
5.4. Newton and quasi-Newton methods . . . 11

6. Functional Analysis Foundations 13
6.1. Normed spaces, Banach and Hilbert spaces . . . 13
6.2. Orthogonal polynomials (Hermite, Laguerre, Legendre) . . . 13
6.3. Special functions (Gamma, Beta, Bessel) . . . 13
6.4. Operator theory foundations for PINNs and QINNs . . . 13

II. Classical Neural Networks 15

7. The Perceptron and Linear Models 17
7.1. McCulloch–Pitts neurons . . . 17
7.2. Rosenblatt's perceptron and linear separability . . . 17
7.3. Logistic regression as probabilistic perceptron . . . 17

8. Feedforward Networks (MLPs) 19
8.1. Activation functions (ReLU, sigmoid, tanh, GELU, softmax) . . . 19
8.2. Universal Approximation Theorem . . . 19
8.3. Forward and backward propagation . . . 19
8.4. Vanishing and exploding gradients . . . 19

9. Convolutional Neural Networks (CNNs) **21**
9.1. Mathematical basis of convolution . . . 21
9.2. Feature maps, receptive fields, pooling . . . 21
9.3. Modern CNN architectures (AlexNet, VGG, ResNet) . . . 21
9.4. Applications in vision, audio, and physics . . . 21

10. Recurrent Neural Networks (RNNs) **23**
10.1. Sequences as dynamical systems . . . 23
10.2. Gradient vanishing and exploding . . . 23
10.3. LSTMs and GRUs . . . 23
10.4. Applications in NLP, speech, time-series . . . 23

11. Autoencoders and Representation Learning **25**
11.1. Linear autoencoders and PCA . . . 25
11.2. Nonlinear autoencoders . . . 25
11.3. Variational Autoencoders (VAEs) . . . 25
11.4. Latent space geometry . . . 25

12. Graph Neural Networks (GNNs) **27**
12.1. Graph Laplacians and spectral methods . . . 27
12.2. Message passing frameworks . . . 27
12.3. Applications in chemistry, materials, biology . . . 27

III. Neural Networks for Differential Equations **29**

13. Mathematical Methods for Differential Equations **31**
13.1. Classification of ODEs and PDEs . . . 31
13.2. Boundary and initial conditions . . . 31
13.3. Separation of variables . . . 31
13.4. Sturm–Liouville problems and orthogonal expansions . . . 31
13.5. Fourier and Laplace transforms . . . 31
13.6. Spectral methods (Chebyshev, Legendre) . . . 31
13.7. Galerkin and Finite Element Methods (FEM) . . . 31
13.8. Method of Frobenius and special functions . . . 31

14. Physics-Informed Neural Networks (PINNs) **33**
14.1. Embedding PDEs into loss functions . . . 33
14.2. Collocation and weak formulations . . . 33
14.3. Elliptic, parabolic, and hyperbolic PDEs . . . 33
14.4. Applications: fluids, electromagnetism, quantum mechanics . . . 33
14.5. Extensions: XPINNs, VPINNs, Bayesian PINNs . . . 33
14.6. Inverse problems (intro to iPINNs) . . . 33

15. Inverse Physics-Informed Neural Networks (iPINNs) **35**
15.1. Motivation: the role of inverse problems . . . 35
15.2. Formulation with unknown coefficients . . . 35
15.3. Loss functions for parameter estimation . . . 35
15.4. Applications: heat, waves, Schrödinger, materials . . . 35
15.5. Ill-posedness and regularization . . . 35
15.6. Sensitivity to noise and data quality . . . 35

16. Quantum Neural Networks (QINNs) **37**
16.1. Hilbert spaces and Dirac notation . . . 37
16.2. Quantum perceptron and gates as layers . . . 37
16.3. Variational quantum circuits (VQE, QAOA) . . . 37
16.4. Quantum Boltzmann Machines, QCNNs, Quantum Reservoirs . . . 37
16.5. Parameter-shift rule for gradients . . . 37

16.6. Challenges: barren plateaus, NISQ hardware . . . 37
16.7. Applications in optimization, chemistry, cryptography . . . 37

17. Neural Operators and DeepONets **39**
17.1. Learning operators between function spaces . . . 39
17.2. Comparison with PINNs, iPINNs, FEM . . . 39
17.3. Applications in PDEs and scientific computing . . . 39

IV. Reinforcement Learning **41**

18. Classical Reinforcement Learning **43**
18.1. Agents, environments, states, actions, rewards . . . 43
18.2. Markov Decision Processes (MDPs) . . . 43
18.3. Value functions and Bellman equations . . . 43
18.4. Tabular methods: SARSA, Q-learning . . . 43

19. Deep Reinforcement Learning **45**
19.1. Deep Q-Networks (DQN) . . . 45
19.2. Policy gradient methods (REINFORCE, PPO) . . . 45
19.3. Actor–Critic architectures (A2C, A3C) . . . 45
19.4. Landmark systems: AlphaGo, AlphaZero, MuZero . . . 45

V. Modern Architectures **47**

20. Generative Models **49**
20.1. GANs and minimax optimization . . . 49
20.2. Wasserstein GANs, StyleGAN . . . 49
20.3. Diffusion models and stochastic processes . . . 49
20.4. Applications in synthesis and design . . . 49

21. Transformers and Attention Mechanisms **51**
21.1. Self-attention: queries, keys, values . . . 51
21.2. Multi-head attention . . . 51
21.3. Positional encodings . . . 51
21.4. Transformer architectures: BERT, GPT, multimodal . . . 51
21.5. Applications in PDEs and symbolic regression . . . 51

VI. Advanced Topics **53**

22. Optimization Beyond Gradient Descent **55**
22.1. Variational inference . . . 55
22.2. Expectation-Maximization (EM) . . . 55
22.3. Federated optimization challenges . . . 55

23. Mathematical Frontiers of Neural Networks **57**
23.1. Neural Tangent Kernels (NTK) . . . 57
23.2. Infinite-width limits and mean-field theory . . . 57
23.3. Geometry of loss landscapes . . . 57
23.4. Generalization bounds and capacity . . . 57

24. Meta-Learning and Transfer Learning **59**
24.1. Few-shot learning . . . 59
24.2. Pretraining and fine-tuning . . . 59
24.3. Continual learning . . . 59

25. Explainability and Interpretability **61**
25.1. Saliency maps and Grad-CAM 61
25.2. SHAP and LIME 61
25.3. Interpretable PINNs and DRL policies 61

26. Ethical and Societal Aspects **63**
26.1. Bias and fairness in AI 63
26.2. Privacy and security 63
26.3. AI regulation and governance 63

VII. Practical Implementation **65**

27. Computational Frameworks **67**
27.1. PyTorch fundamentals 67
27.2. TensorFlow and Keras 67
27.3. JAX and differentiable programming 67

28. Efficient Training and Scaling **69**
28.1. Hardware acceleration: GPUs, TPUs 69
28.2. Parallelization and distributed training 69
28.3. Memory-efficient backpropagation 69

29. Case Studies in Scientific Machine Learning **71**
29.1. Navier–Stokes with PINNs 71
29.2. QINNs for quantum chemistry 71
29.3. DRL for robotics and control 71
29.4. CNNs/GNNs for materials science 71

A. Mathematical Notation and Symbols **73**

B. Linear Algebra Toolbox **75**

C. Probability Distributions **77**

D. Special Functions (Gamma, Beta, Bessel, etc.) **79**

E. Implementations in PyTorch and TensorFlow **81**

Part I.

Mathematical Preliminaries

Introduction 1.

1.1. Historical context of AI and neural networks

1.2. AI winters and the deep learning revolution

1.3. Why mathematics matters in deep learning

Linear Algebra Essentials 2.

2.1. Vector spaces and inner products

2.2. Eigenvalues, eigenvectors, diagonalization

2.3. Singular value decomposition (SVD)

2.4. Tensor notation and operations

Multivariable Calculus and Analysis

3.

3.1. Gradients, Jacobians, Hessians

3.2. Taylor expansions in multiple variables

3.3. Divergence, curl, and Laplacians

3.4. Variational principles

Probability, Statistics, and Information Theory 4.

4.1. Random variables and distributions

4.2. Expectation, variance, covariance

4.3. Gaussian and exponential families

4.4. Entropy, KL divergence, mutual information

Optimization Theory 5.

5.1. Convexity, duality, and Lagrangians

5.2. Gradient descent and its variants

5.3. Stochastic optimization and convergence

5.4. Newton and quasi-Newton methods

Functional Analysis Foundations | 6.

6.1. Normed spaces, Banach and Hilbert spaces

6.2. Orthogonal polynomials (Hermite, Laguerre, Legendre)

6.3. Special functions (Gamma, Beta, Bessel)

6.4. Operator theory foundations for PINNs and QINNs

Part II.

Classical Neural Networks

7. The Perceptron and Linear Models

7.1. McCulloch–Pitts neurons

7.2. Rosenblatt's perceptron and linear separability

7.3. Logistic regression as probabilistic perceptron

Feedforward Networks (MLPs) 8.

8.1. Activation functions (ReLU, sigmoid, tanh, GELU, softmax)

8.2. Universal Approximation Theorem

8.3. Forward and backward propagation

8.4. Vanishing and exploding gradients

Convolutional Neural Networks (CNNs) 9.

9.1. Mathematical basis of convolution

9.2. Feature maps, receptive fields, pooling

9.3. Modern CNN architectures (AlexNet, VGG, ResNet)

9.4. Applications in vision, audio, and physics

10. Recurrent Neural Networks (RNNs)

10.1. Sequences as dynamical systems

10.2. Gradient vanishing and exploding

10.3. LSTMs and GRUs

10.4. Applications in NLP, speech, time-series

11. Autoencoders and Representation Learning

11.1. Linear autoencoders and PCA

11.2. Nonlinear autoencoders

11.3. Variational Autoencoders (VAEs)

11.4. Latent space geometry

Graph Neural Networks (GNNs) 12.

12.1. Graph Laplacians and spectral methods

12.2. Message passing frameworks

12.3. Applications in chemistry, materials, biology

Part III.

Neural Networks for Differential Equations

Mathematical Methods for Differential Equations 13.

13.1. Classification of ODEs and PDEs

13.2. Boundary and initial conditions

13.3. Separation of variables

13.4. Sturm–Liouville problems and orthogonal expansions

13.5. Fourier and Laplace transforms

13.6. Spectral methods (Chebyshev, Legendre)

13.7. Galerkin and Finite Element Methods (FEM)

13.8. Method of Frobenius and special functions

14. Physics-Informed Neural Networks (PINNs)

14.1. Embedding PDEs into loss functions

14.2. Collocation and weak formulations

14.3. Elliptic, parabolic, and hyperbolic PDEs

14.4. Applications: fluids, electromagnetism, quantum mechanics

14.5. Extensions: XPINNs, VPINNs, Bayesian PINNs

14.6. Inverse problems (intro to iPINNs)

15. Inverse Physics-Informed Neural Networks (iPINNs)

15.1. Motivation: the role of inverse problems

15.2. Formulation with unknown coefficients

15.3. Loss functions for parameter estimation

15.4. Applications: heat, waves, Schrödinger, materials

15.5. Ill-posedness and regularization

15.6. Sensitivity to noise and data quality

Quantum Neural Networks (QINNs) 16.

16.1. Hilbert spaces and Dirac notation

16.2. Quantum perceptron and gates as layers

16.3. Variational quantum circuits (VQE, QAOA)

16.4. Quantum Boltzmann Machines, QCNNs, Quantum Reservoirs

16.5. Parameter-shift rule for gradients

16.6. Challenges: barren plateaus, NISQ hardware

16.7. Applications in optimization, chemistry, cryptography

17. Neural Operators and DeepONets

17.1. Learning operators between function spaces

17.2. Comparison with PINNs, iPINNs, FEM

17.3. Applications in PDEs and scientific computing

Part IV.

Reinforcement Learning

18. Classical Reinforcement Learning

18.1. Agents, environments, states, actions, rewards

18.2. Markov Decision Processes (MDPs)

18.3. Value functions and Bellman equations

18.4. Tabular methods: SARSA, Q-learning

19. Deep Reinforcement Learning

19.1. Deep Q-Networks (DQN)

19.2. Policy gradient methods (REINFORCE, PPO)

19.3. Actor–Critic architectures (A2C, A3C)

19.4. Landmark systems: AlphaGo, AlphaZero, MuZero

Part V.

Modern Architectures

Generative Models | 20.

20.1. GANs and minimax optimization

20.2. Wasserstein GANs, StyleGAN

20.3. Diffusion models and stochastic processes

20.4. Applications in synthesis and design

21. Transformers and Attention Mechanisms

21.1. Self-attention: queries, keys, values

21.2. Multi-head attention

21.3. Positional encodings

21.4. Transformer architectures: BERT, GPT, multimodal

21.5. Applications in PDEs and symbolic regression

Part VI.

Advanced Topics

22. Optimization Beyond Gradient Descent

22.1. Variational inference

22.2. Expectation-Maximization (EM)

22.3. Federated optimization challenges

Mathematical Frontiers of Neural Networks 23.

23.1. Neural Tangent Kernels (NTK)

23.2. Infinite-width limits and mean-field theory

23.3. Geometry of loss landscapes

23.4. Generalization bounds and capacity

Meta-Learning and Transfer Learning

24.

24.1. Few-shot learning

24.2. Pretraining and fine-tuning

24.3. Continual learning

25. Explainability and Interpretability

25.1. Saliency maps and Grad-CAM

25.2. SHAP and LIME

25.3. Interpretable PINNs and DRL policies

Ethical and Societal Aspects 26.

26.1. Bias and fairness in AI

26.2. Privacy and security

26.3. AI regulation and governance

Part VII.

Practical Implementation

Computational Frameworks | 27.

27.1. PyTorch fundamentals

27.2. TensorFlow and Keras

27.3. JAX and differentiable programming

28. Efficient Training and Scaling

28.1. Hardware acceleration: GPUs, TPUs

28.2. Parallelization and distributed training

28.3. Memory-efficient backpropagation

Case Studies in Scientific Machine Learning 29.

29.1. Navier–Stokes with PINNs

29.2. QINNs for quantum chemistry

29.3. DRL for robotics and control

29.4. CNNs/GNNs for materials science

A. Mathematical Notation and Symbols

Linear Algebra Toolbox | B.

Probability Distributions C.

Special Functions (Gamma, Beta, Bessel, etc.) D.

E. Implementations in PyTorch and TensorFlow