Program Structure

- Math for AI
- AI Engineering

AI Academy

Linear Algebra

Week 1

Introduction to Vectors, Matrices

1. Vectors and linear combinations
2. Lengths and dot products
3. Matrices, definition, matrix operations

Week 2

Linear Equations and Gaussian Elimination

1. Vectors and linear equations
2. Gaussian elimination
3. Elimination using matrices

Matrix Operations and Decomposition

1. Rules for matrix operations

2. Inverse matrices, conditions for existence, Gauss-Jordan elimination, Determinant

3. LU decomposition; Transposes, symmetric, permutation matrices

Vector Spaces and Subspaces

1. Vector spaces, subspaces

2. The nullspace of A: Solving $Ax = 0$

3. The complete solution to $Ax = b$

Independence, Basis and Dimension, Orthogonality

1. Linear independence, basis, dimension of vector spaces
2. Four fundamental subspaces (column, null, row, left null) and their dimensions
3. Orthogonal subspaces, orthogonal complements

Orthogonality cont'd

1. Orthogonal bases, Gramm-Schmidt process
2. Projection of vectors, orthogonal projections
3. Least squares approximation, fitting linear models

Eigenvalues and Eigenvectors

1. Eigenvalues, eigenvectors, characteristic equation

2. Diagonalization, conditions for diagonalization

3. Iterative estimates for eigenvalues/ eigenvectors

Linear Transformations

1. Definitions and properties of linear transformations
2. Representation of linear transformations as matrices
3. SVD, computation and applications

Final Exam and Projects

1. Review session
2. Comprehensive assessment covering all topics (final exam)
3. Presenting final projects that apply multiple linear algebra concepts

Calculus + Optimization methods

Fundamentals of Calculus for AI/ML

1. Introduction to Limits, definition, limit laws
2. Continuity, types of discontinuities
3. Limits at Infinity and Infinite Limits

Differential Calculus and Its Applications

1. Introduction to derivatives, definition, rules of differentiation
2. Applications of derivatives, tangent lines, rates of change
3. Second and higher-order derivatives, Taylor expansion

Integral Calculus and Its Applications

1. Introduction to integrals, definition, fundamental theorem of calculus
2. Applications of integrals, areas, volumes
3. Techniques of integration (integration by parts, substitution)

Multivariable Calculus for AI/ML

1. Partial Derivatives, chain rule
2. Gradient vectors, Hessian matrices, linear and quadratic approximations
3. Double integrals and applications

Optimization Techniques in AI/ML (part1)

1. Gradient Descent Variants - stochastic, mini-batch

2. Gradient Descent with Momentum, Adaptive Momentum (Adam); Newton's method

3. BFGS

Optimization Techniques in AI/ML (part2)

1. Conjugate gradient
2. Trust-region methods

Optimization Techniques in AI/ML (part3)

1. Constrained optimization, Lagrange multipliers
2. KKT conditions

Advanced AI/ML Applications

1. Case Studies in AI/ML
2. Integrating techniques in AI/ML, Combining calculus and optimization
3. Review and preparation for the final project

Probability theory
+ Statistics

Introduction to Probability Theory

1. **Fundamentals of Probability**
   - Basics of probability, event spaces

2. **Conditional probability, independence, law of total probability, chain rule (product rule)**

3. **Random Variables**
   - Discrete random variables, probability mass function (PMF)

Intro cont'd

1 Random Variables cont'd

- Continuous random variables, probability density function (PDF)

2 Probability Distributions

- common distributions (Bernoulli, beta, binomial, exponential, gamma, normal, Poisson)

3 Probability Distributions cont'd

- bivariate, marginal/conditional distributions, independent random variables

Expectation, Variance, and Moments

1 Expectation and Variance

- Definition and properties of expectation and variance

2 Moments and Moment Generating Functions

- Higher-order moments, moment generating functions

3 Covariance and Correlation

- Covariance, correlation, and their properties

Bayesian Probability

1. **Bayes Theorem**
   - Bayes theorem, applications, and examples

2. **Bayesian Inference**
   - Bayesian inference, prior and posterior distributions

3. **Bayesian Networks**
   - intro to bayesian networks

Hypothesis Testing and Confidence Intervals

1 Hypothesis Testing Basic

- Null and alternative hypotheses, types of errors

2 Confidence Intervals

- Constructing and interpreting confidence intervals

3 Advanced Hypothesis Testing

- ANOVA, chi-square tests

Statistical Inference and Estimation

1 **Point Estimation**

- Methods of point estimation, properties of estimators

2 **Interval Estimation**

- Constructing and interpreting interval estimates

3 **Maximum Likelihood Estimation (MLE)**

- Principles of MLE, applications

Markov Chain and Sampling methods

1. Markov Chain

2. Standard distributions
   - Multinoulli, Gaussian

3. Importance sampling, MCMC, Gibbs sampling

Review and Final Project

1 Review of the content

2 Final project

Software engineering

01 Python basics and data structures

1 Introduction, control structures, functions

- Introduction to Python and its applications
- Python syntax and indentation
- Variables and data types
- Control structures: if-else statements, loops (for, while)
- Functions: defining functions, arguments, return values

2 Lists, tuples, dictionaries, sets

- Lists: creation, indexing, slicing, methods (append, remove, etc.)
- Tuples: creation, indexing, immutability
- Dictionaries: key-value pairs, methods (get, keys, values, etc.)
- Sets: creation, methods (add, remove, union, intersection, etc.)

3 List comprehensions, generators, decorators

- List comprehensions: syntax and use cases
- Generators: yield keyword, creating and using generators
- Decorators: function decorators, applying multiple decorators

02 Python advanced features

1 File I/O, exception handling

- File I/O: reading from and writing to files
- Exception handling: try, except, finally blocks, raising exceptions

2 Modules and packages, standard library

- Modules: creating and importing modules
- Packages: structuring code with packages, __init__.py
- Standard library: overview of commonly used modules (os, sys, math, etc.)

3 Working with APIs, web scraping

- Working with APIs: making HTTP requests, handling responses
- Web scraping: BeautifulSoup, requests library

03 Python object-oriented programming (OOP)

1 Introduction to OOP

- Introduction to OOP: classes and objects
- Defining and using classes, instance methods, and attributes
- Inheritance and polymorphism

2 Advanced OOP concepts

- Advanced OOP concepts: class methods, static methods, properties
- Using decorators with classes
- Designing with inheritance and composition

3 Concurrency (threading and multiprocessing)

- Introduction to concurrency: threading and multiprocessing
- The threading module: creating and managing threads
- The multiprocessing module: creating and managing processes

04 Python concurrency mechanisms (Asynchronous programming)

1 Introduction to asynchronous programming

- Introduction to asynchronous programming: asyncio module
- Defining and running asynchronous tasks
- Using async and await keywords

2 Advanced asynchronous techniques

- Advanced concurrency techniques: futures, coroutines
- Handling exceptions in asynchronous code
- Using concurrent.futures for parallelism

3 Integration and deployment

- Integrating Python applications with databases
- Deployment strategies: packaging and distributing Python applications
- Using Docker for containerization

05 Code quality, testing, debugging

1 Code quality (writing clean code, code reviews)

- Principles of clean code: readability, maintainability, simplicity
- Code reviews: best practices, conducting effective reviews

2 Unit testing (writing test cases,TDD)

- Unit testing: importance, frameworks (unittest, pytest)
- Test-Driven Development (TDD): concepts, workflow

3 Debugging techniques and tools

- Debugging: strategies and techniques
- Tools: using IDEs, debuggers, logging

Machine Learning

01 Intro and supervised learning basics

1 Introduction to ML

- What it is and why it is needed, compare traditional “AI” against traditional software
- Paradigms of ML

2 Supervised learning

- KNN, for classification and regression tasks
- Approximate NN

3 Model validation and evaluation

- Data splitting, bias-variance tradeoff, validation/evaluation, metrics

02 Simplest models and what the “learning” process looks like

1 Supervised learning for regression

Linear regression

- algebraic interpretation
- probabilistic interpretation
- polynomial fitting

2 How to improve

- Hyperparameters, regularization, grid search

3 Supervised learning for classification

- Logistic regression vs Naive Bayes
- Transforming text into numbers

03 Practical issues in data preparation

1 Data cleaning

- Data cleaning, handling missing values, encoding categorical values

2 Feature engineering

- Techniques and real examples

3 Feature scaling

- Min/max, max/abs, transformations (z-score, log and etc)

04 Practical issues (cont'd) and other model families

1 Imbalanced datasets

- What to do at different steps: data preparation, training and metrics

2 Other families for supervised learning

- SVM (SVC, SVR)
- Decision trees, random forests, gradient boosting

3 Supervised learning wrap-up

05 Unsupervised learning

1. PCA
2. Density estimation, histograms, KDE
3. Clustering, Kmeans, hierarchical

06 Unsupervised learning (cont'd) and intro to RL

1. DBScan
2. GMM, EM
3. Reinforcement learning basics

07 Review and practical exercises

1 Review and practical exercises on machine learning algorithms

- Review of key concepts and algorithms learned so far

2 Practical session (applying supervised learning models)

3 Practical session (applying unsupervised learning models)

Deep Learning and common applications

AI Academy

01 Introduction to neural networks

1 What are NNs, why DL got so popular

- Issue of feature engineering
- General info about most common architectures (Perceptron-MLP, CNN, RNN, Transformers)

2 Basic MLP under the lens

- MLE framework, forward propagation, loss construction, univariate regression, binary/multiclass classification, multitask learning

3 Learning with backpropogation

- Derivation of update formulas
- Cases for MSE and binary CE

02 Neural network basics

1 Activation functions

- What makes NN non-linear
- Commonly used functions

2 Initialization and optimization

- Gradient descent variations

3 Regularization

- Common and DL specific regularization methods

03 CNNs and intro to transfer learning

1 CNN basics

- CNNs on 1d input
- CNNs on 2d input
- Downsampling and upsampling

2 CNN architectures, transfer learning basics

- AlexNet, Inception, VGG, ResNet
- Transfer learning with CNNs

3 Applications

- Object detection, segmentation
- YOLO, Faster-RCNN, U-Net, Mask R-CNN

04 Transfer learning for texts, RNNs

1 Text representation

- Word, subword embeddings and how to learn them
- Word2vec training, negative sampling

2 How to encode sequential information

- Logistic regression and MLP on embeddings
- CNNs for texts

3 RNNs

- Vanilla RNN
- Vanishing / exploding gradients
- LSTM, GRU, BiLSTM

05 RNNs (cont'd)

1 RNNs in practice

- Sequence classification, sequence labeling tasks

2 Sequence generation

- MLP vs RNN approach for language modeling
- Seq2seq: Machine translation

3 Improved seq2seq

- Introducing attention block for seq2seq

06 Attention is all you need!

1 Further improved seq2seq

- Birth of transformer architecture
- Building blocks

2 BERT, pretraining/ finetuning

- Sequence classification
- Sequence pair classification
- Sequence labeling

3 Decoder-only setup, aka GPT

- Pretraining / finetuning
- GPT family, scaling
- Birth of new paradigm: in-context learning

07 Transformers for vision, multi-modal learning, practical problems

1 Transformers in computer vision

- Vision transformers, masked autoencoders

2 Multi-modal learning

- CLIP

3 Model compression

- Distillation and quantization

08 Guest lectures

1 Learning on graphs

2 RAG

3 Multi-modal learning

Review and final project