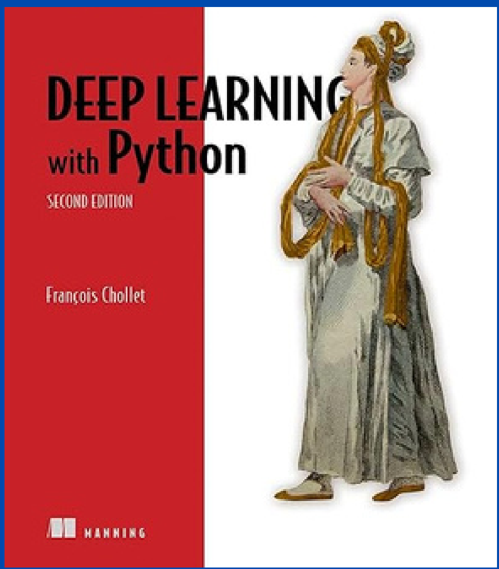
I.A X A.I

Lessons Learnt from Deep Learning with Python



FOUNDATIONS OF ARTIFICIAL INTELLIGENCE AND DEEP LEARNING

- Artificial Intelligence (AI) Overview:
  - Definition: AI encompasses efforts to automate tasks typically performed by humans.
  - Tasks: AI tasks can be categorized into abstraction, reasoning, pattern recognition, and intuition.

- Traditional Software Engineering vs. Machine Learning:
  - Traditional Software: Follows the Input + Process → Output model. It's efficient for tasks like calculations but limited in encoding only abstraction and reasoning.
  - Machine Learning: Adopts the Input + Output → Process approach. It's superior in pattern recognition and intuition, learning from examples to create rules.

- Concepts in Machine Learning:
  - Interpretations, Representations, Transformations: Key elements in understanding how machine learning models process data.

- Types of Machine Learning:
  - Shallow Machine Learning: Efficient for small datasets, limited to 1-2 data transformations. Examples include Naive Bayes, Support Vector Machines, and Decision Trees.
  - Deep Machine Learning (Deep Learning): Capable of numerous complex data transformations, using Neural Networks like Recurrent and Convolutional Neural Networks.

- Input and Output Modalities:
  - Computers convert various modalities (like images) into numerical values for processing.

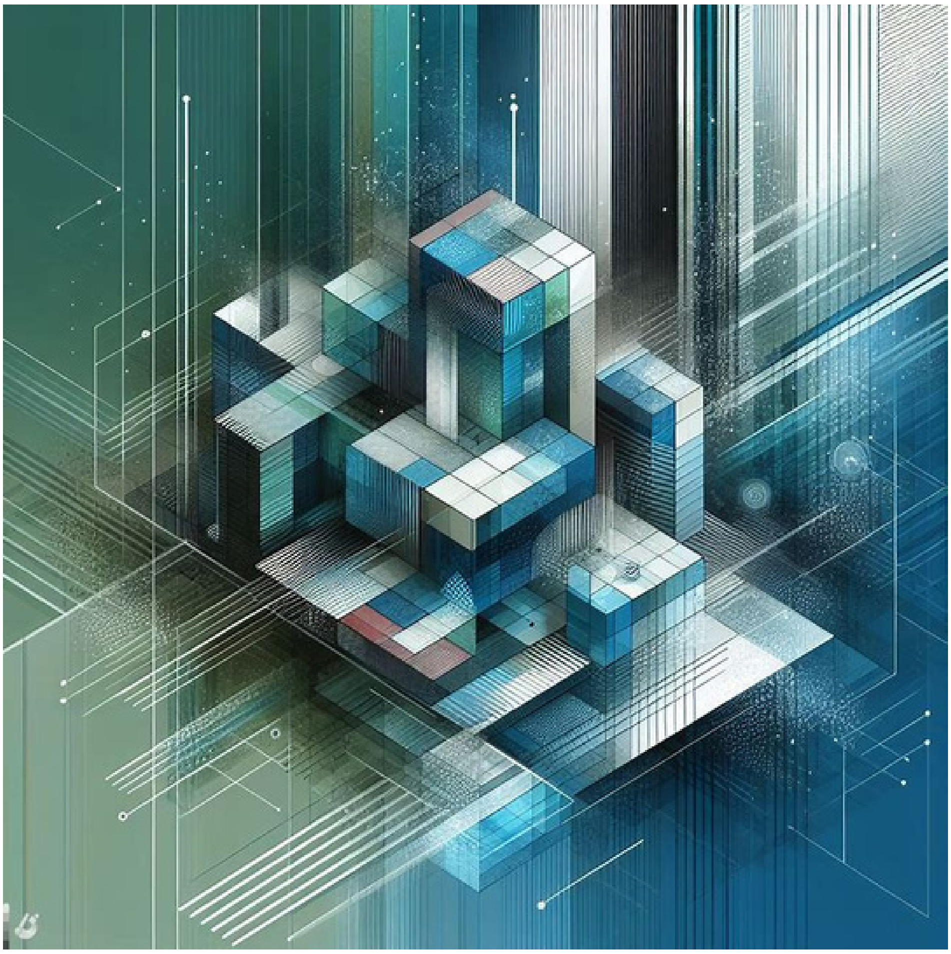
FOUNDATIONS OF ARTIFICIAL INTELLIGENCE AND DEEP LEARNING

- The Future of Machine Learning:
  - Integration of traditional software engineering methods for enhanced abstraction and reasoning.

- Deep Learning Fundamentals:
  - Tensor Operations: Basic operations in deep learning neural networks.
  - Layers and Functions: Neural networks consist of layers that perform specific tensor operations, leading to geometric transformations.

- Understanding Neural Networks:
  - A neural network is a series of geometric transformations in a high-dimensional space.
  - Importance of Non-linear Transformations: They allow for richer hypothesis spaces and complex data representations.

- Key Questions and Insights:
  - The significance of high-dimensional space and its impact on data representation.
  - The process of decomposing complex transformations into simpler ones in deep learning.



KEY CONCEPTS IN NEURAL NETWORKS

Basic Terminology:

Categories, data points, layers, dense layers, optimizers, loss functions, and metrics.

Tensors and Data Structures:

Understanding tensors, matrices, vectors, and their dimensionality.
Neural Network Development Process:

Steps include data loading, preparation, choosing architecture and optimizer, setting loss functions and metrics, fitting the model, making predictions, and evaluation.

Tensor Gradients:

Tensor gradients are crucial for calculating the necessary adjustments to each parameter tensor, aiming to minimize the loss function.
They account for the impact of each parameter on the function.

Weights and Learning:

Weights, stored as tensors, contain the knowledge of a model.
Learning involves finding weight values that minimize the loss function for given training data and targets.

Mini-batch Stochastic Gradient Descent (SGD):

Involves using random batches of data samples.
Computes gradients of model parameters against the loss on each batch.
Parameters are then adjusted in the direction opposite to the gradient, facilitating learning.

Fundamentals of Learning:

Learning is possible because all tensor operations in neural networks are differentiable.
This differentiability enables the calculation of gradients.

Backpropagation:

A key process in learning, where the chain rule is used to compute gradients of model parameters relative to the loss on a batch.
Backpropagation allows for the effective updating of model weights based on these gradients.

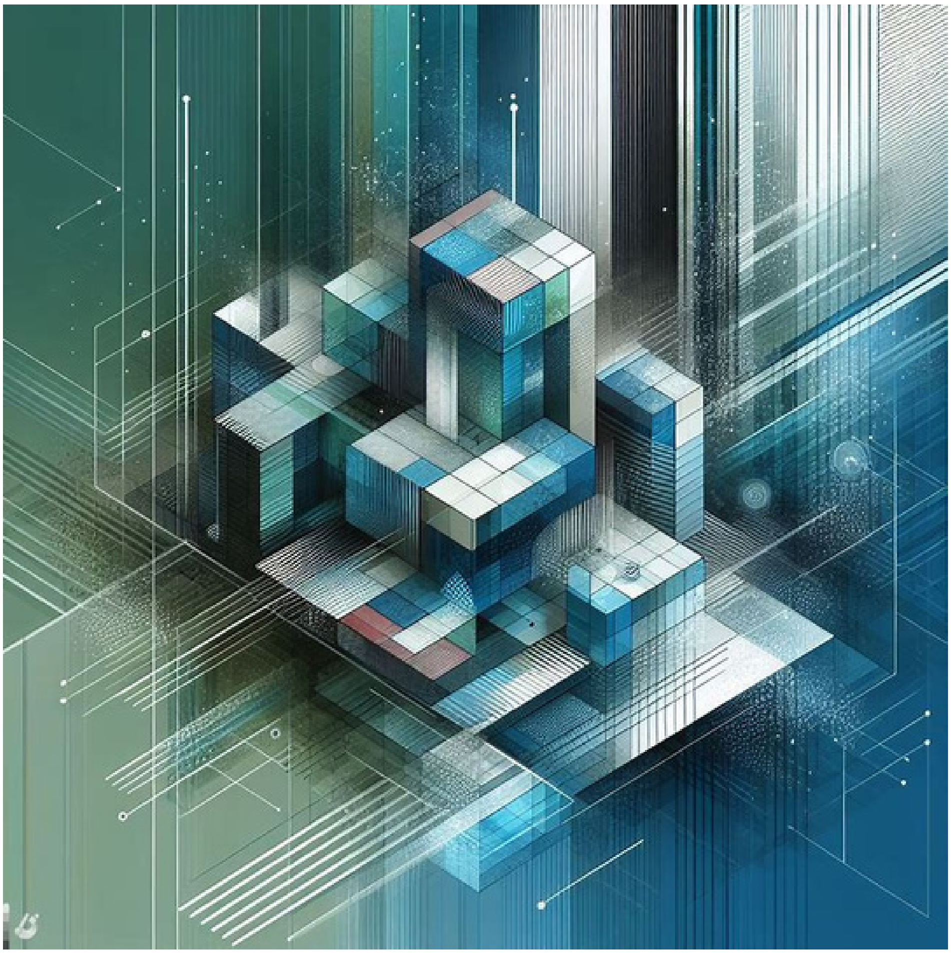
KEY CONCEPTS IN NEURAL NETWORKS

Understanding the Gradient Function:

The gradient function is a result of backpropagation.
It indicates how the loss function would change with slight variations in each network parameter.
This function varies with different batches of data and as network parameters update.

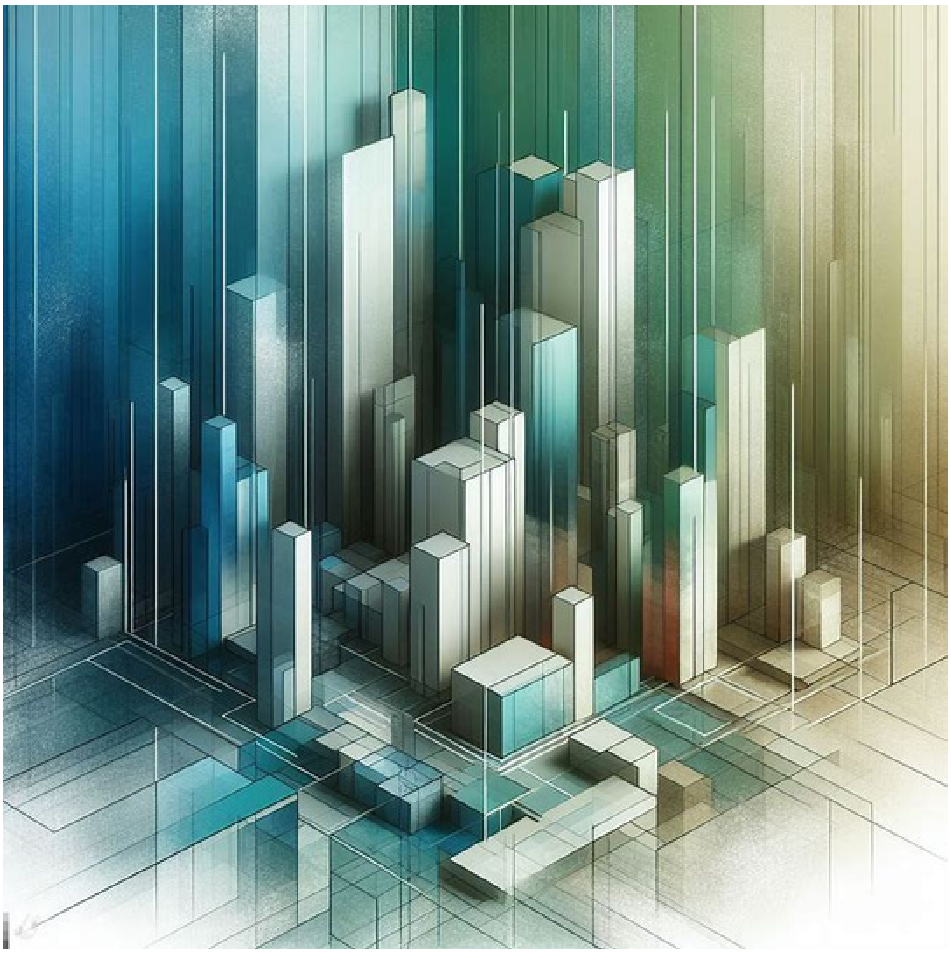
Loss and Optimizers:

Loss is a measure of success, the quantity that needs to be minimized during training.
The optimizer defines how the gradients of the loss will be used to update model parameters, guiding the learning process.



PREPROCESSING, LOSS FUNCTIONS, AND MODEL DESIGN

1. **Preprocessing and Feature Engineering:**
   - Scaling: If input data features have different ranges, scale each feature independently for consistency.
2. **Loss Functions:**
   - Regression: Specific loss functions are commonly used for regression tasks to measure how well the model predicts continuous values.
3. **Evaluation Metrics:**
   - Classification: Accuracy is typically the primary metric of interest.
   - Regression: Focus on error metrics like Mean Absolute Error (MAE) to assess prediction accuracy.
4. **Evaluating Models with Limited Data:**
   - Model Structure: With little data, use 1-2 intermediate layers to prevent overfitting.
   - Validation Method: Employ K-fold validation to mitigate the high variance issue in validation splits due to limited data.
     - Process: Split data into K partitions, train on K-1 partitions, and evaluate on the remaining one. Repeat and average the scores for robustness.
5. **Design Considerations for Complex Categories:**
   - Model Complexity: Ensure sufficient layers to handle the complexity when classifying several categories.
   - Avoid Information Bottlenecks: Prevent intermediary layers from being too small, which could restrict the necessary information flow from input to output.
6. **Additional Note:**
   - Logistic Regression: Despite being a shallow machine learning model, logistic regression is actually used for classification tasks.



FUNDAMENTALS OF MACHINE LEARNING

Optimization and Overfitting: Understanding and addressing overfitting is essential. It involves considering the influence of individual features and their relevance, which may sometimes lead to spurious correlations. Getting your model to overfit is key before picking a more robust model that generalises well.

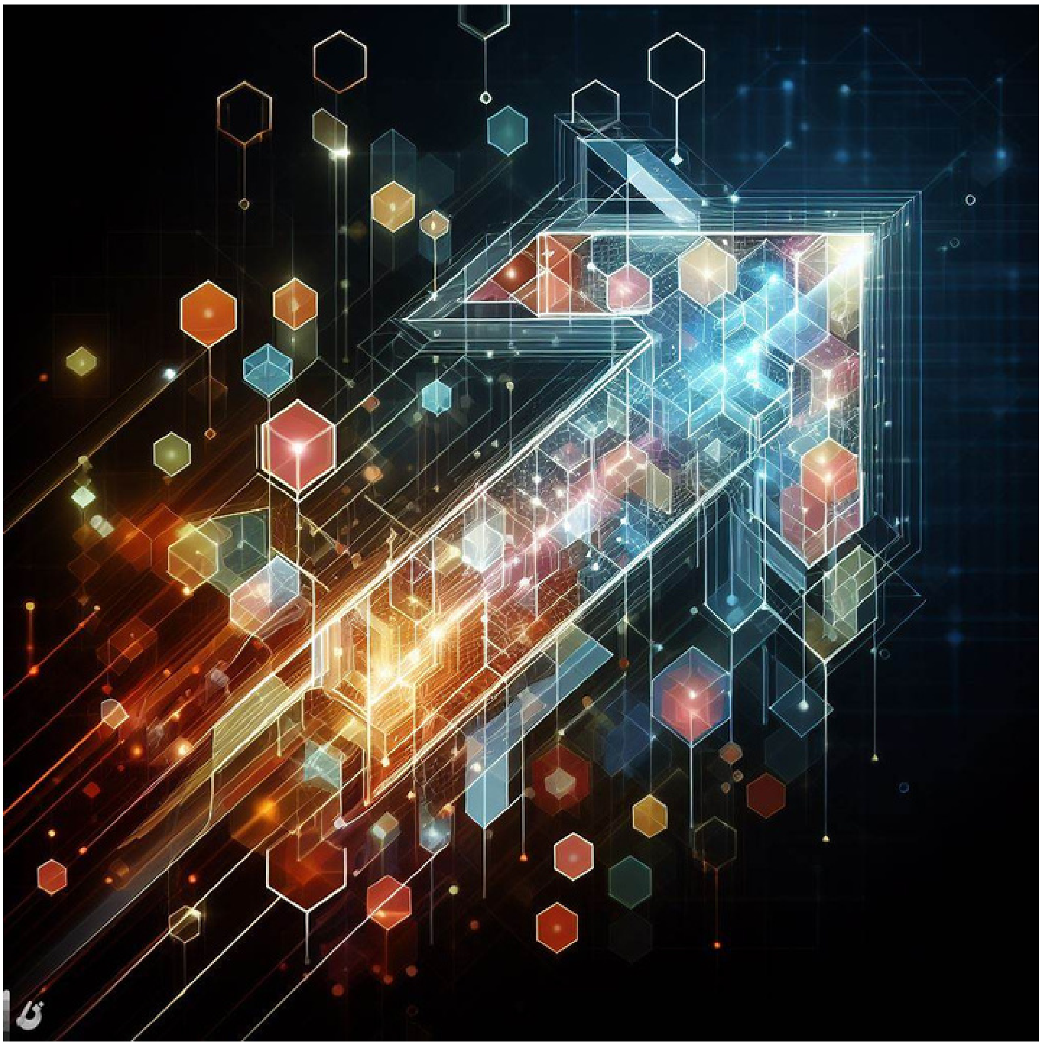
Feature Selection: Utilising techniques such as mutual information scores to select the most informative features, which is vital in datasets with numerous features.

Manifold Hypothesis: This complex concept involves understanding that high-dimensional data can often be represented in a simpler, lower-dimensional space.

Model Evaluation: The importance of representative training and testing sets, especially in time-series data, to avoid temporal leaks. The use of K-fold validation is emphasized, particularly in cases of limited data.

Regularization Techniques: Strategies like early stopping, reducing network size, weight regularization, and dropout are discussed. These techniques help in simplifying the model and avoiding overfitting, ensuring that the model generalizes well to new data.

Dropout: This concept is explained as a technique to improve generalization by randomly deactivating neurons during training, thus preventing over-reliance on specific neurons or pathways.



SPECIALIZATIONS

Chapter 8: Deep Learning for Computer Vision

- Focuses on combating overfitting in image data through data augmentation.

Chapter 9: Advanced Deep Learning for Computer Vision

- Highlights best practices like residual connections, batch normalization, and depthwise separable convolutions.
- Discusses inspecting convnets' visual representations and understanding image classification decisions using class activation maps.

Chapter 10: Deep Learning for Time Series

- Recommends RNNs, especially LSTM and GRU layers, as optimal for time series data.
- Stresses the importance of establishing a common baseline.

Chapter 11: Deep Learning for Text

- Compares bag-of-words models (like n-grams with dense layers) for unordered word processing and sequence models (like RNNs, 1D convnets, transformers) for ordered processing.
- Identifies transformers as state-of-the-art for complex sequence-to-sequence tasks such as machine translation.

Chapter 12: Generative AI

- Explores the possibility of transferring the style of one image to another.
- Introduces DeepDream and the application of convnet activation layers for generating visually striking images.

MACHINE LEARNING CONSULTANCY SHOP

1. Business Preparation:
   - Steps: Incorporation, website, and networking.
2. Ethical Consideration:
   - Assess the moral implications of projects before taking them on.
3. Project Flow Process:
   - Includes problem definition, data collection and considerations, understanding data, success metrics, data pre-processing, evaluation protocol, model development, stakeholder communication, deployment options, inference model optimization, and model maintenance.
4. Problem Definition:
   - Contextual Understanding: Understanding customer needs, data availability, and constraints.
   - Problem Framing: Determining inputs, targets, and machine learning tasks.
   - Existing Solutions: Researching current methods and solutions.
5. Data Collection and Annotation:
   - Challenges in data collection and the need for quality, representative data.
   - Options for data annotation, including in-house, crowdsourcing, and specialized companies.
6. Understanding and Preprocessing Data:
   - Data exploration, handling missing values, and ensuring data quality.
7. Choosing a Measure of Success:
   - Defining relevant metrics aligned with business goals.
8. Model Development:
   - Focusing on achieving statistical power, experimenting with architectures, and tuning hyperparameters.
9. Stakeholder Communication:
   - Setting realistic expectations and explaining model limitations.
10. Deployment Options:
    - Various deployment methods like REST API, on-device, and in-browser.
11. Inference Model Optimization:
    - Techniques like weight pruning and quantization for resource-constrained environments.
12. Monitoring and Maintaining Models:
    - Strategies for ongoing model evaluation, adaptation, and maintenance.

BEST PRACTICES

There are systematic methods for enhancing machine learning model performance, focusing on hyperparameter tuning and model ensembles.

How to improve model performance?

Model Ensembles

Ensembles works by pooling together different perspectives of the data so you can get a far more accurate description of the data.

Not so much How good your best model is but the share diversity of your set of candidates. Their ability to provide information that other models don't have access to. Need biases to cancel out.

Hyperparameter tuning

Automatic hyperparameter optimization using techniques like Bayesian optimization, genetic algorithms, and random search due to the complexity and resource-intensive nature of manual tweaking. Tools are available like KerasTuner for hyperparameter tuning and anticipates future advancements in AutoML for generating entire machine learning pipelines automatically. One issue at scale is validation-set overfitting.

Future of ML Engineers

As things get automated, ML Engineers will move up higher in value creation tasks. Focusing on data curation, crafting complex loss functions that are an accurate reflection of business objectives, and understanding the broader impact of models in digital ecosystems.

How to improve speed and scale up the model?

Speed

Strategies for speeding up and scaling model training, including multi-GPU training, TPU training, mixed precision, and leveraging cloud computing.

WORKING WITH KERAS - CUSTOM METRICS

Custom Metrics in Keras:

1. Keras, a powerful and flexible deep learning library, allows you to define and use your own custom metrics.
2. This feature is particularly useful when standard metrics do not fully capture the specific goals or nuances of your machine learning project.
3. Custom metrics can be tailored to the unique requirements of your task, providing more relevant and insightful evaluations of your model's performance.

THE FUTURE

The final chapter focuses on progression of machine learning towards more advanced capabilities, like formal reasoning and abstraction, and its implications for AI. It explores the concept of moving from local to extreme generalisation, akin to human cognition, and the potential for automating the development of AI systems.

It addresses the current limitations of deep learning models, particularly in understanding and generalising beyond their training data. The chapter also emphasizes the need for new metrics focused on generalization and the use of datasets like ARC to evaluate AI's ability to handle novel situations.

Additionally, it delves into the importance of analogy extraction and the relationship between discrete and continuous abstraction methods, highlighting the potential for blending these approaches in AI development. The chapter concludes by contemplating the future of AI, including the integration of program synthesis and genetic search, to create more adaptable and generalizable systems.

I.A X A.I