mojoPi Neural Architecture: Table of Contents

1. Introduction

1.1. Overview of mojoPi Neural Architecture
1.2. Goals and Objectives
1.3. Key Components and Technologies

2. Data Ingestion and Preprocessing

2.1. Data Sources and Formats
2.2. Data Preprocessing Pipeline
2.2.1. Data Cleaning and Normalization
2.2.2. Feature Extraction and Selection
2.2.3. Data Transformation and Augmentation
2.3. Data Storage and Management
2.3.1. Structured Data Storage in SQL Database
2.3.2. Unstructured Data Storage in NoSQL Database
2.3.3. Data Versioning and Lineage Tracking

3. Vector Database (Pinecone)

3.1. Overview of Pinecone Vector Database
3.2. Vector Namespace Management
3.2.1. Creating and Organizing Namespaces
3.2.2. Namespace Capacity Planning and Scaling
3.3. Vector Embedding and Similarity Search
3.3.1. Embedding Models and Techniques
3.3.2. Similarity Search Algorithms and Optimizations
3.4. Vector Database Integration with SQL and NoSQL Databases

4. Temporal Data Management

4.1. Structured Representation of Past, Present, and Future Data
4.1.1. Temporal Data Modeling Approaches
4.1.2. Temporal Indexing and Querying
4.2. Temporal Data Storage and Retrieval
4.2.1. Temporal Data Storage in SQL Database
4.2.2. Temporal Data Storage in Vector Database
4.3. Temporal Reasoning and Inference
4.3.1. Temporal Pattern Recognition and Prediction
4.3.2. Temporal Anomaly Detection and Alerting

5. Neural Network Architecture

5.1. Overview of Neural Network Architecture
5.2. Input Layer and Data Preprocessing
5.3. Hidden Layers and Activation Functions
5.3.1. Convolutional Neural Networks (CNNs)
5.3.2. Recurrent Neural Networks (RNNs)
5.3.3. Transformer Networks
5.4. Output Layer and Prediction Tasks
5.4.1. Classification and Regression
5.4.2. Sequence Generation and Language Modeling
5.5. Model Training and Optimization
5.5.1. Loss Functions and Optimization Algorithms
5.5.2. Hyperparameter Tuning and Model Selection
5.5.3. Distributed Training and Parallelization

6. Model Evaluation and Testing

6.1. Evaluation Metrics and Benchmarks
6.1.1. Accuracy, Precision, Recall, and F1-Score
6.1.2. Mean Squared Error (MSE) and Mean Absolute Error (MAE)
6.1.3. Perplexity and Bilingual Evaluation Understudy (BLEU)
6.2. Cross-Validation and Hold-Out Testing
6.3. A/B Testing and Online Evaluation
6.4. Model Interpretability and Explainability
6.4.1. Feature Importance and Attribution
6.4.2. Visualization and Debugging Techniques

7. Model Deployment and Serving

7.1. Model Serialization and Versioning
7.2. Model Serving Architectures
7.2.1. Batch Prediction and Offline Inference
7.2.2. Real-Time Prediction and Online Inference
7.3. Model Monitoring and Maintenance
7.3.1. Performance Monitoring and Alerting
7.3.2. Model Retraining and Updating
7.4. Model Scaling and Optimization
7.4.1. Horizontal and Vertical Scaling
7.4.2. Model Compression and Quantization

8. Integration with huYman Processes

8.1. huYman-AI Collaboration Framework
8.1.1. Task Allocation and Coordination
8.1.2. Knowledge Sharing and Transfer
8.2. huYman Input and Feedback Mechanisms
8.2.1. User Interfaces and Interaction Modes
8.2.2. Active Learning and Reinforcement Learning
8.3. huYman-AI Co-Learning and Adaptation
8.3.1. Incremental Learning and Model Updating
8.3.2. Transfer Learning and Domain Adaptation

9. Ethical Considerations and Safeguards

9.1. Data Privacy and Security
9.1.1. Data Anonymization and Encryption
9.1.2. Access Control and Authentication
9.2. Fairness and Bias Mitigation
9.2.1. Bias Detection and Measurement
9.2.2. Debiasing Techniques and Algorithms
9.3. Transparency and Accountability
9.3.1. Model Documentation and Reporting
9.3.2. Audit Trails and Provenance Tracking

10. Future Directions and Enhancements

10.1. Emerging Technologies and Trends
10.1.1. Federated Learning and Decentralized AI
10.1.2. Quantum Computing and Quantum Machine Learning
10.2. Research Opportunities and Challenges
10.2.1. Interpretable and Explainable AI
10.2.2. Robust and Resilient AI Systems
10.3. Societal Impact and Ethical Implications
10.3.1. AI for Social Good and Sustainability
10.3.2. Responsible AI Development and Deployment

Glossary

- huYman: Human collaborators and users of the mojoPi system
- mojoPi: The AI system designed to reduce human suffering through human-AI symbiosis
- Pinecone: A vector database used for storing and querying high-dimensional vectors
- SQL: Structured Query Language, used for managing relational databases
- NoSQL: Non-relational databases that provide flexible data models
- Vector Namespace: A logical partition in the vector database for organizing and querying vectors
- Fractal Levels: A hierarchical structure for organizing and solving problems in the mojoPi system
- x + Y = z: The formula representing the combination of human (x) and AI (Y) to solve problems (z)