

Introduction to Intelligence Systems

**Intelligence System
Development**

2024 – 2025
Y4E1 – DCS – NU

By: SEK SOCHEAT

Advisor to DCS and Lecturer

Mobile: 017 879 967

Email: socheat.sek@gmail.com

Table of Contents

1. Overview of Intelligent Systems

- What are Intelligent Systems?
- Types of Intelligent Systems
- Applications of Intelligent Systems
- Key Components of Intelligent Systems

2. Software Development Tools and Frameworks for IS

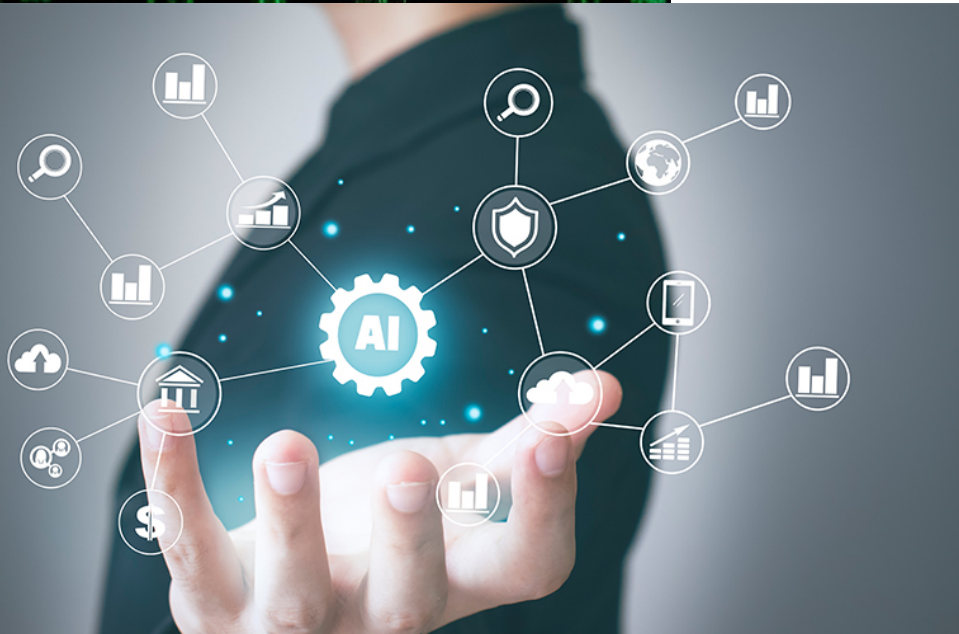
- Programming Languages for Intelligent Systems
- AI and Machine Learning Frameworks
- Data Processing Tools
- Integrated Development Environments (IDEs)
- Version Control and Collaboration Tools
- Cloud-based AI Development Platforms
- Other Tools for Specific AI Applications

3. Practical Activity

Overview of Intelligent Systems

Types, Applications, and Key Components

What are Intelligent Systems?



- Intelligent systems are computational systems that use techniques from artificial intelligence (AI) to perceive their environment, learn from data, and act or make decisions autonomously.
- They can be designed to solve complex problems by simulating human-like intelligence, such as reasoning, decision-making, pattern recognition, and adapting to changing conditions.

Types of Intelligent Systems

Rule-based Systems: These systems rely on predefined rules (if-then logic) to make decisions. They were some of the earliest forms of intelligent systems.

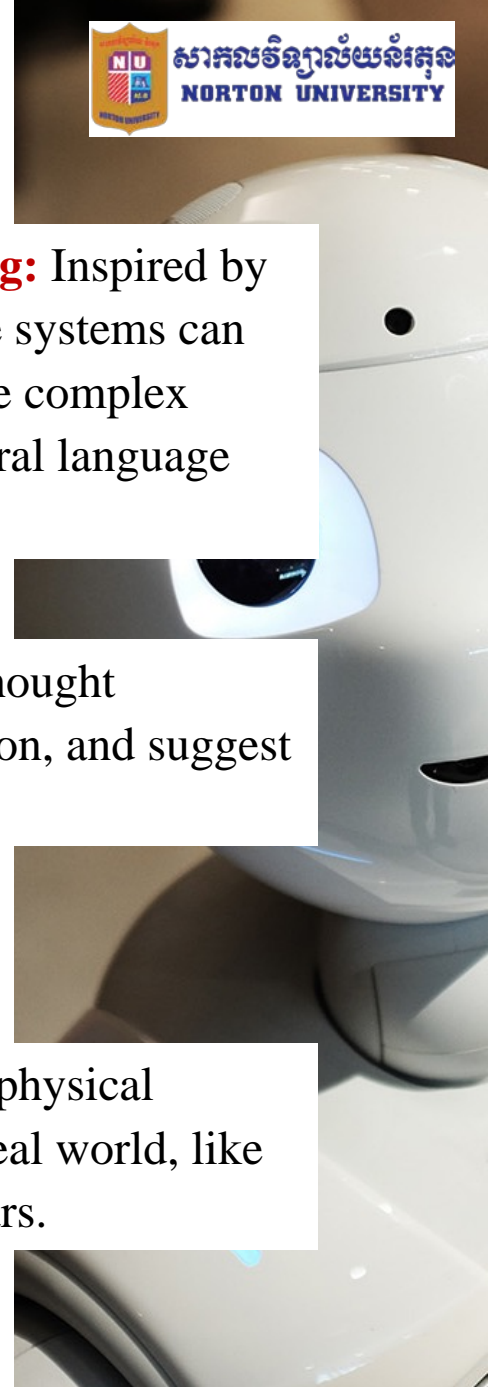
Expert Systems: Designed to replicate the decision-making ability of a human expert in a specific domain, using knowledge bases and inference engines.

Machine Learning Systems: These systems use data to learn and improve from experience over time. Examples include recommendation systems, speech recognition, and image classification.

Neural Networks and Deep Learning: Inspired by the structure of the human brain, these systems can process vast amounts of data and solve complex problems like image recognition, natural language processing, and game playing.

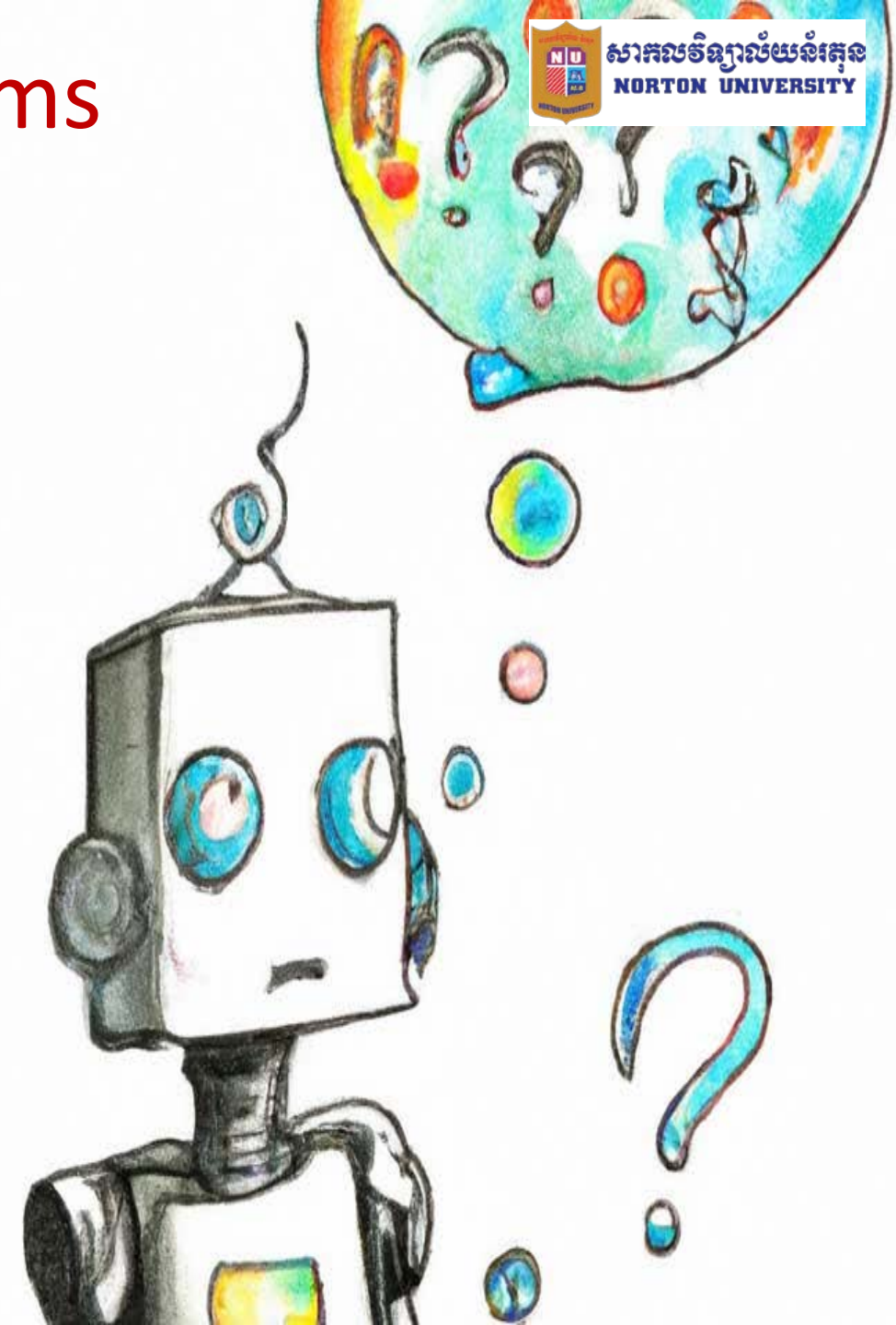
Cognitive Systems: Mimic human thought processes to understand context, reason, and suggest solutions (e.g., IBM Watson).

Robotic Systems: Combine AI with physical components to perform tasks in the real world, like autonomous drones or self-driving cars.



Applications of Intelligent Systems

- **Healthcare:** Diagnosis systems, predictive models for patient outcomes, and robotic surgery.
- **Finance:** Fraud detection, algorithmic trading, and credit risk evaluation.
- **Manufacturing:** Predictive maintenance, automated quality control, and robotics for assembly lines.
- **Transportation:** Self-driving vehicles, traffic management systems, and route optimization.
- **Customer Service:** Chatbots and virtual assistants for answering queries and resolving issues.
- **Smart Homes/IoT:** Home automation systems (smart thermostats, lighting, etc.) and predictive systems for energy management.



Key Components of Intelligent Systems

1. **Perception:** Systems must first perceive their environment using sensors, cameras, or other input devices (e.g., visual sensors for a self-driving car).
2. **Data Processing:** Raw data collected by sensors needs to be cleaned, structured, and processed before it can be used (e.g., feature extraction, normalization in machine learning).
3. **Learning/Inference Engine:** Core AI algorithms enable the system to learn from data and make decisions (e.g., neural networks in deep learning, decision trees in machine learning).
4. **Knowledge Base:** For systems like expert systems, the knowledge base contains domain-specific information and rules for decision-making.
5. **Actuation:** Systems act upon their environment (e.g., a robot moving its arms) based on the decisions or predictions they make.
6. **Feedback Loop:** Intelligent systems often improve over time by incorporating feedback from the environment or user interactions (reinforcement learning, adaptive systems).

Software Development Tools and Frameworks for Intelligence Systems

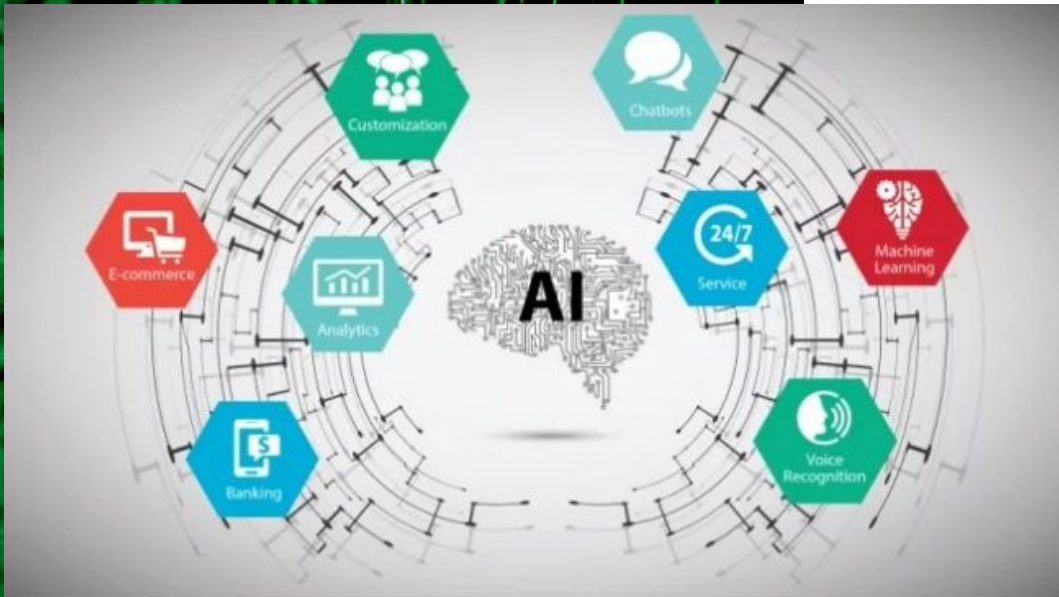
Programming Languages for Intelligent Systems

Python Programming:

- Python is the most widely used language for AI and machine learning due to its simplicity and extensive ecosystem of libraries.

Advantages: Easy to learn, strong community support, and rich libraries for scientific computing (e.g., NumPy, SciPy, Pandas).

Python's role in AI development: Its flexibility allows it to be used in everything from data preprocessing to model training and deployment.



AI and Machine Learning Frameworks

1. TensorFlow

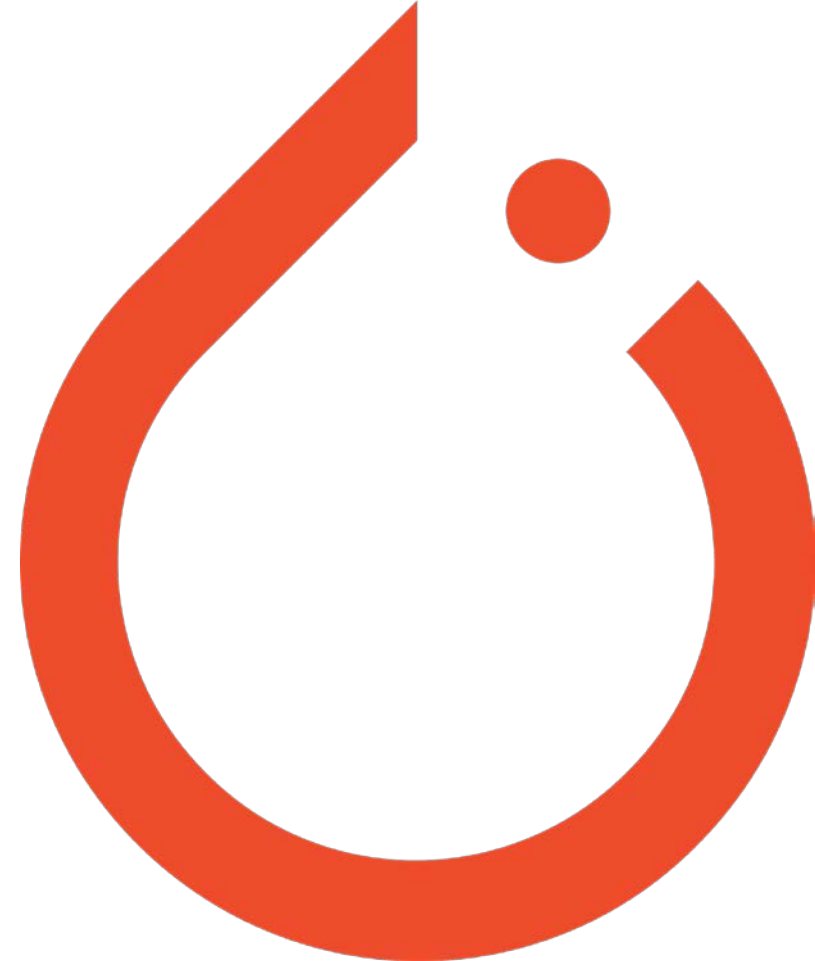
- Developed by Google, TensorFlow is one of the most popular deep learning frameworks.
- Provides tools for building and training machine learning models, especially for neural networks.
- TensorFlow supports both low-level operations (for research purposes) and high-level APIs like Keras for quick prototyping.
- **Use cases:** Image recognition, NLP, time series analysis.



AI and Machine Learning Frameworks

2. PyTorch

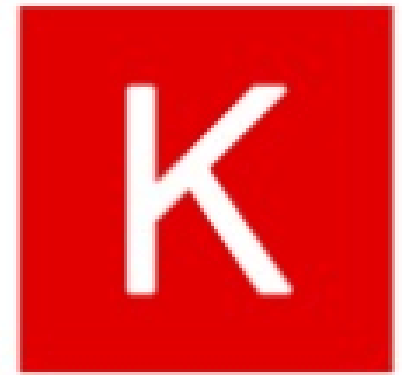
- Developed by Facebook, PyTorch is another popular framework known for its flexibility and ease of use, particularly favored in research.
- Provides dynamic computational graphs, which makes it easier to experiment with model architectures.
- PyTorch is well-suited for rapid development, and is increasingly used in production environments.
- **Use cases:** Deep learning research, NLP, reinforcement learning.



AI and Machine Learning Frameworks

3. Keras

- An open-source library that provides a high-level API to build and train deep learning models with minimal coding.
- Keras can run on top of TensorFlow, simplifying complex machine learning workflows.



Keras

Data Processing Tools



1. NumPy

- A fundamental package for numerical computing in Python, providing support for arrays, matrices, and high-level mathematical functions.
- Essential for handling large datasets and performing matrix operations, which are common in machine learning.

Data Processing Tools

2. Pandas

Pandas

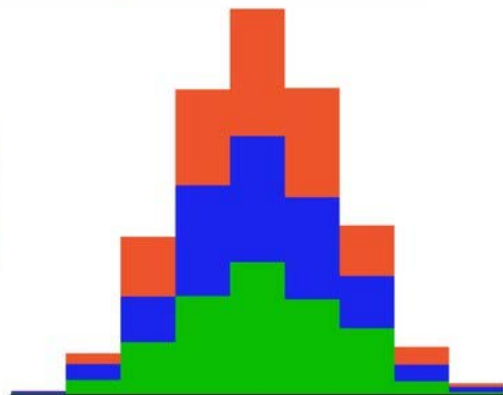
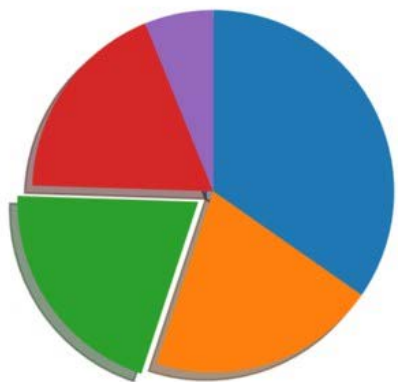


- A powerful data manipulation library used to handle structured data in tables (e.g., data frames).
- Pandas makes it easy to preprocess data, clean datasets, and perform data analysis.

Data Processing Tools

3. Matplotlib/Seaborn

matplotlib



- Visualization libraries that help create graphs, charts, and plots to better understand datasets and model performance.

Integrated Development Environments (IDEs)

Jupyter Notebook



- Jupyter Notebook is an interactive web-based development environment that supports Python.
- Popular in data science and AI development for its ability to combine code, text, and visualizations in a single document.
- Useful for experimenting with machine learning models, prototyping, and sharing research.

Integrated Development Environments (IDEs)

VS Code

A lightweight, flexible IDE with strong support for Python development and machine learning extensions (e.g., TensorFlow and PyTorch plugins).



Version Control and Collaboration Tools

Git and GitHub

- Git is a version control system that helps track code changes, enabling collaboration in teams and across projects.
- GitHub (or GitLab/Bitbucket) is an online platform for hosting and sharing code, with features like pull requests, issue tracking, and project management.
- Essential for collaboration in AI development, where multiple team members may work on the same model or datasets.

The GitHub logo, featuring the word "GitHub" in a white, bold, sans-serif font, set against a black background.

Cloud-based AI Development Platforms

What is Google Colab?



- A cloud-based platform that allows you to run Jupyter Notebooks in the cloud with free access to GPUs.
- Ideal for students and small projects that require powerful computation without needing dedicated hardware.

Cloud-based AI Development Platforms

AWS SageMaker



Amazon SageMaker

Amazon's machine learning service that provides an environment to build, train, and deploy machine learning models at scale.

Cloud-based AI Development Platforms

Microsoft Azure Machine Learning Studio



AZURE
MACHINE LEARNING
STUDIO

Azure's machine learning platform, providing pre-built models, drag-and-drop interfaces, and support for Python and R.

Cloud-based AI Development Platforms



OpenCV

- A powerful library used for computer vision applications such as object detection, image recognition, and facial recognition.
- Provides tools to capture, process, and analyze visual data.



NLTK (Native Language Tool Kit)

A toolkit for building natural language processing systems, including tools for text processing, tokenization, and sentiment analysis.

Practical Activity

Installation and Setup

- Install Python, TensorFlow, and PyTorch in the development environment (e.g., Jupyter Notebook, VS Code).
- Demonstrate how to set up a project structure and manage dependencies using tools like pip or Anaconda.

Hands-On Example

- Implement a simple machine learning model (e.g., linear regression or image classification) using TensorFlow or PyTorch.
- Visualize the data using Matplotlib/Seaborn, and display training accuracy and loss using Jupyter Notebook.





Thank you