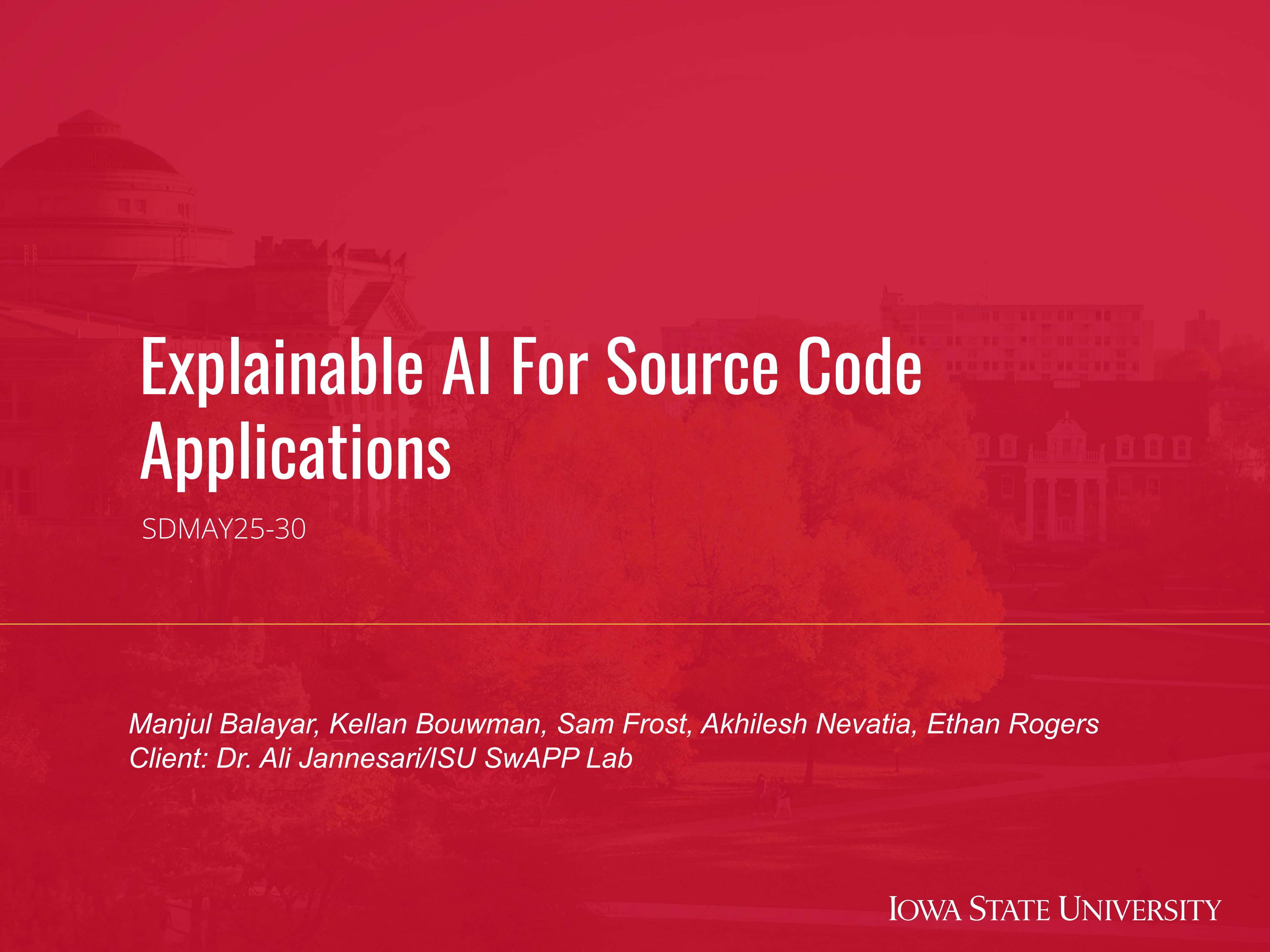


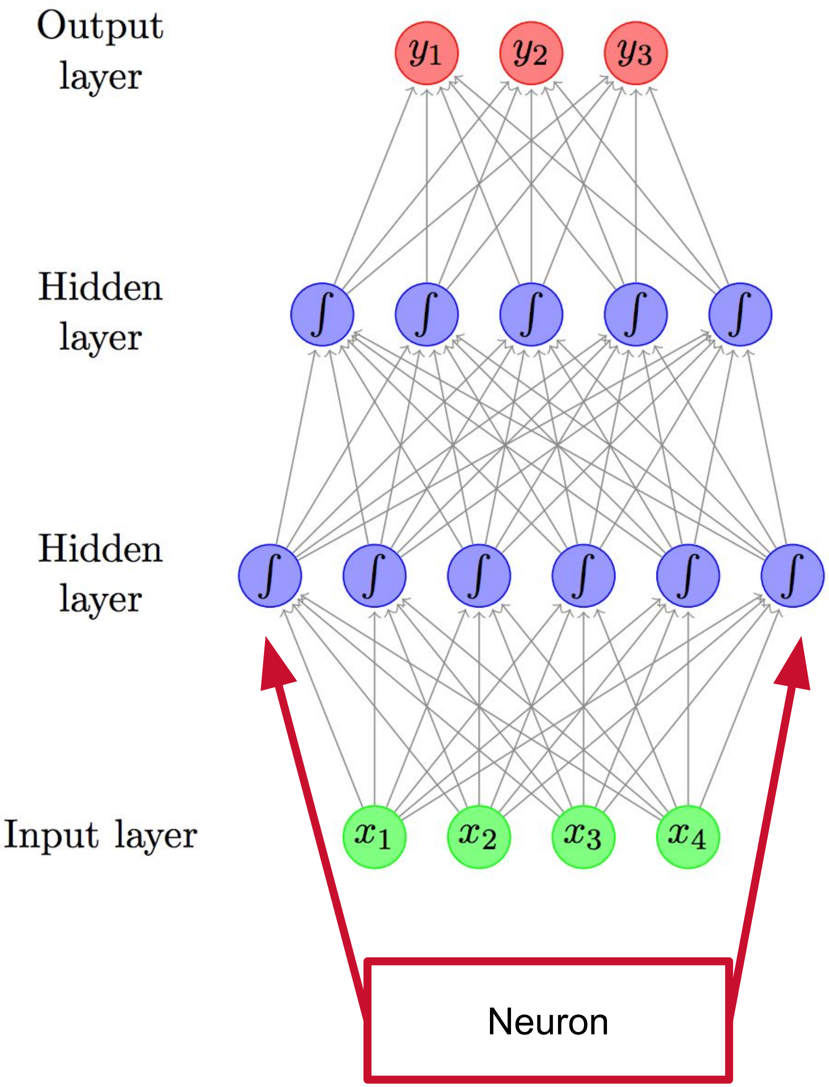
Explainable AI For Source Code Applications

SDMAY25-30

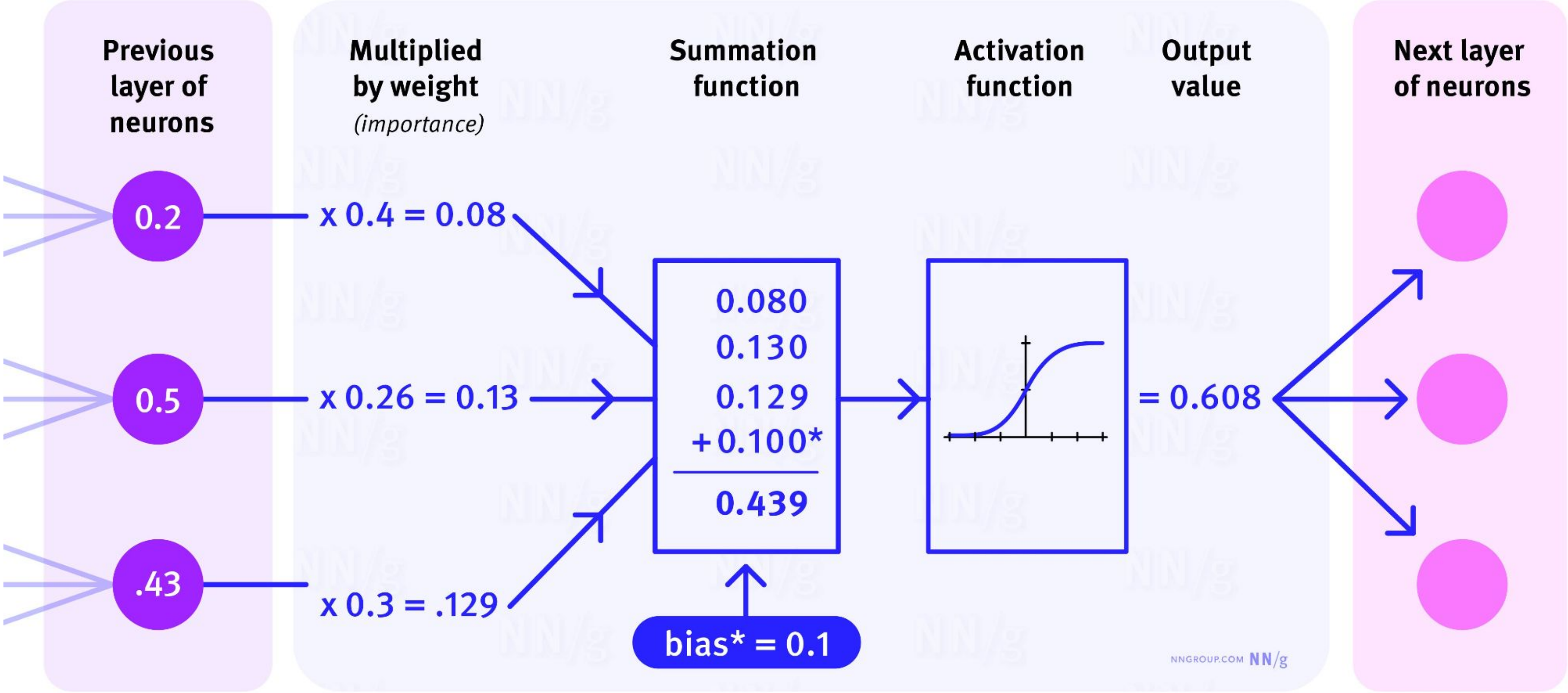
Manjul Balayar, Kellan Bouwman, Sam Frost, Akhilesh Nevatia, Ethan Rogers
Client: Dr. Ali Jannesari/ISU SwAPP Lab

IOWA STATE UNIVERSITY

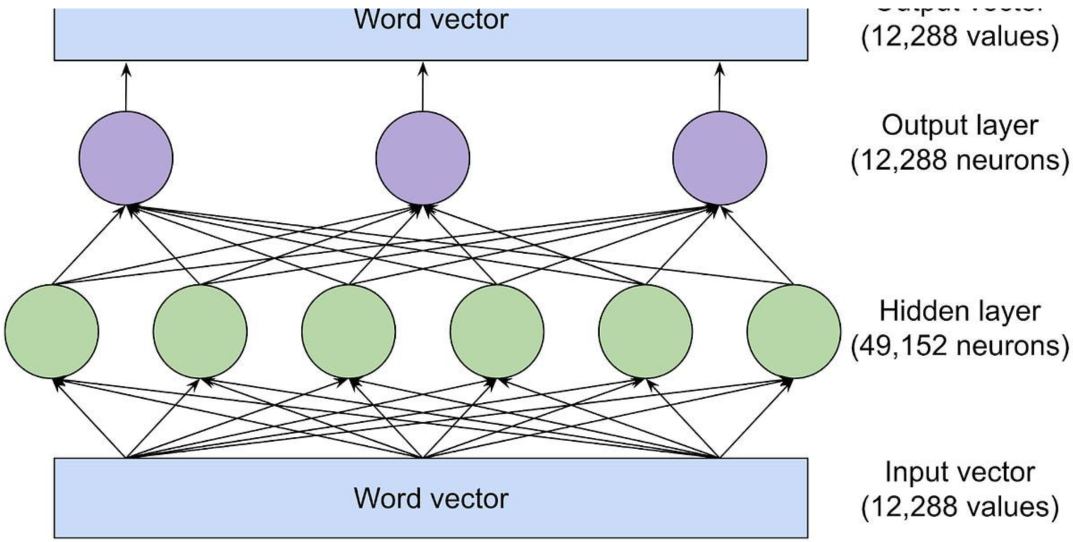
- Machine Learning / Statistical Learning:
  - Machine learning: A broad discipline that focuses on how computers can learn from data
  - Statistical learning: A branch of artificial intelligence that focuses on turning raw data into actionable information. Statistical learning theory is a framework that uses statistical and functional analysis to build models that can make predictions and draw conclusions from data
- Neuron:
  - A collection of a set of inputs, a set of weights, and an activation function.
  - It translates these inputs into a single output.
- Deep Learning:
  - Type of machine learning based on artificial neural networks in which multiple layers of processing are used to extract progressively higher level features from data

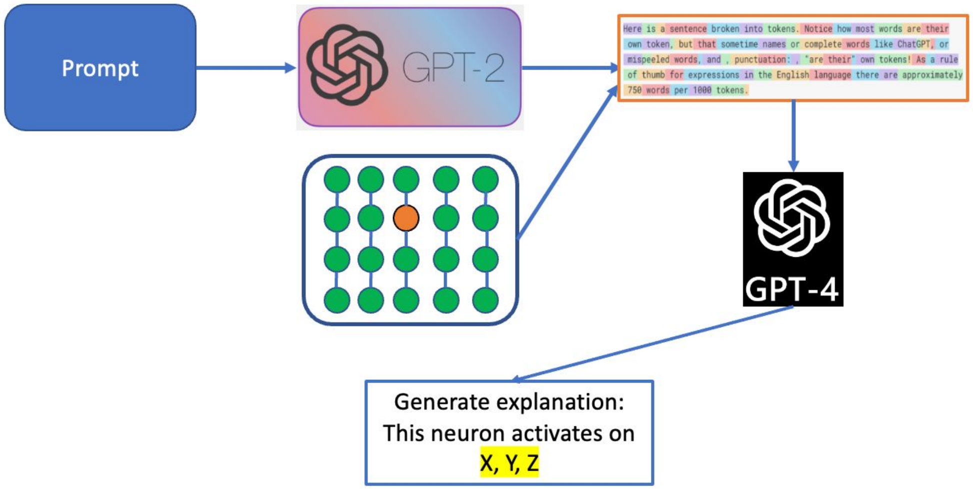


How a Single Artificial Neuron Works



- LLM:
  - Large Language Model
  - Form of Deep learning on NLP (Natural Language Processing)
- Generative AI:
  - A type of AI that uses generative models to create new content, such as text, images, videos, music, and audio
  - Based off an LLM (example: ChatGPT3.5)
- Neuron Activation
  - Non-linear function that we apply over the input data coming to a particular neuron and the output from the function will be sent to the neurons present in the next layer as input
  - A Path of Neurons





Here is a sentence broken into tokens. Notice how most words are their own token, but that sometime names or complete words like ChatGPT, or mispeeled words, and , punctuation: , "are their" own tokens! As a rule of thumb for expressions in the English language there are approximately 750 words per 1000 tokens.

Generate explanation:
This neuron activates on
X, Y, Z

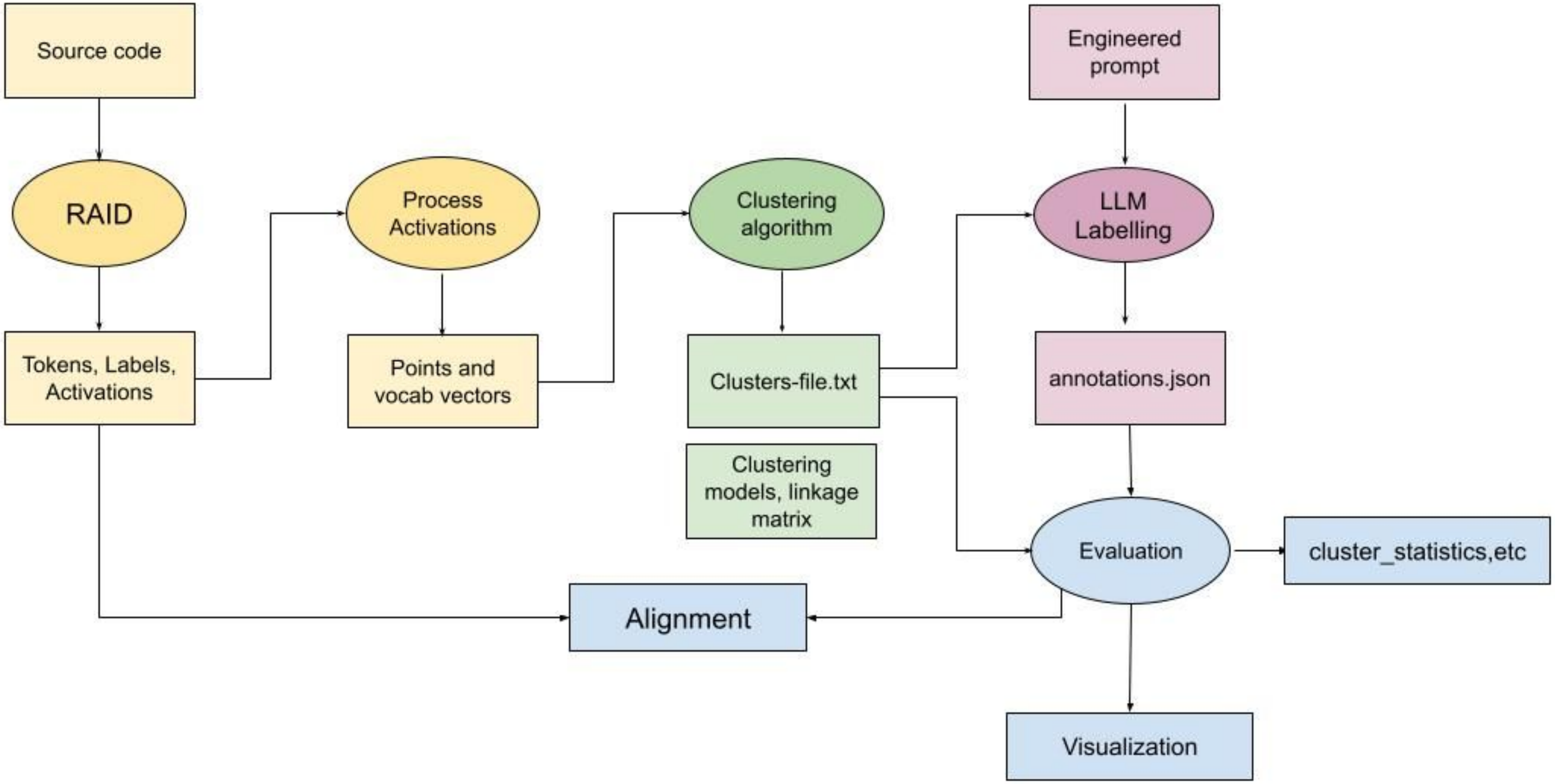
Project Overview

- Client
  - Dr. Ali Jannesari/ISU SwAPP Lab
- Abstract
  - Focus on auto-labeling code datasets using AST tools, regular expressions, and LLM-generated labels.
- Goal
  - Deepen the understanding of LLMs and generative AI by analyzing neuron activations and applying metrics and heuristics. The project will also unify two existing code bases into a flexible and scalable framework that can be deployed seamlessly across Colab, local environments, and HPC clusters.

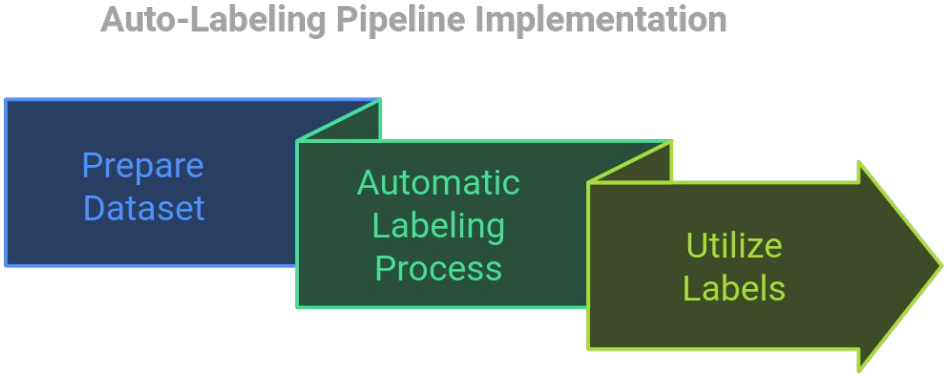
Project Overview - Continued

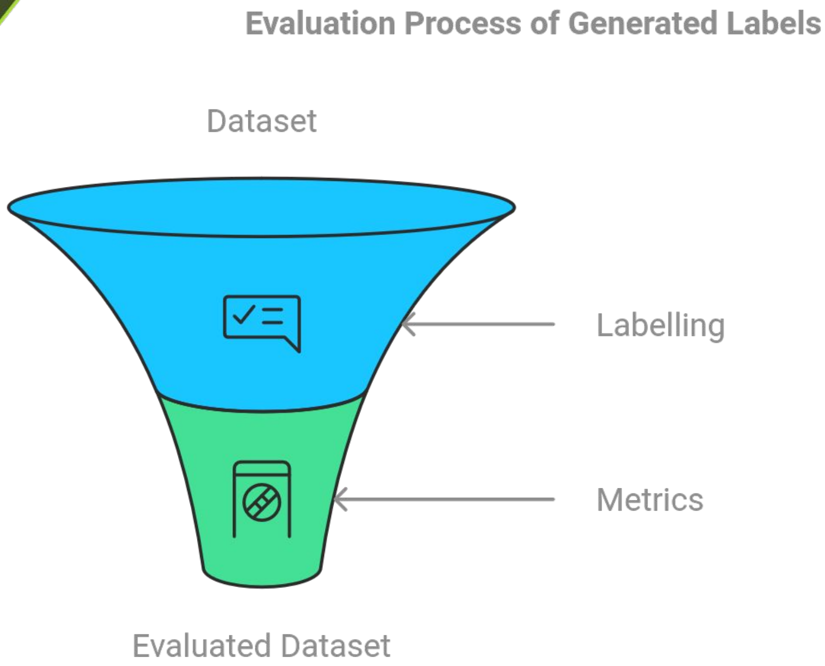
- Users
    - Researchers
        - ML Engineers / Researchers
        - Prompt Engineers / Researchers
        - Computer Scientists
    - Students
        - Graduate Students
        - Undergraduate Students
    - Industry Professionals
- We aim to include students, researchers, and industry professionals as key users. Our documentation will prioritize accessibility and clarity for all experience levels, while the codebase will remain flexible and scalable to meet both academic and industry needs.

Detailed Design Visuals



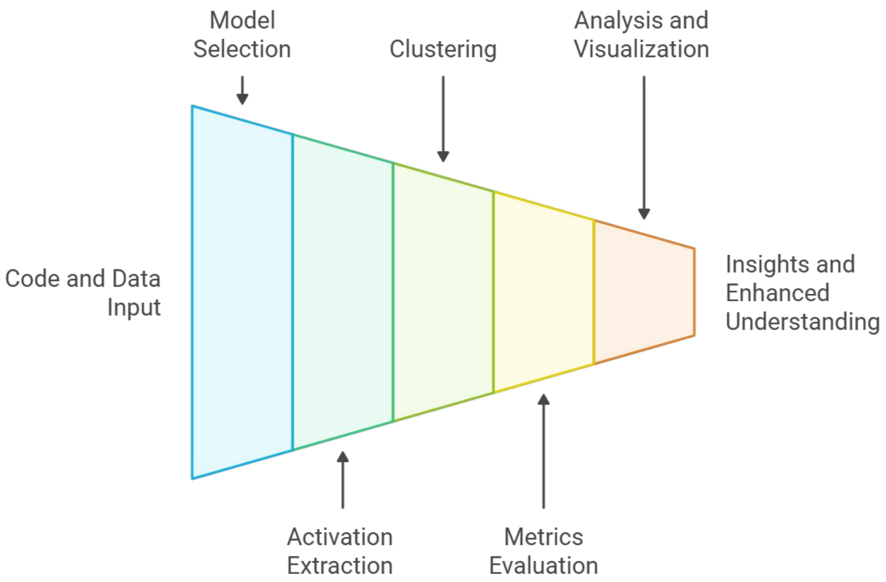
Detailed Design Visuals


Auto-Labeling Pipeline Implementation
Prepare Dataset
Automatic Labeling Process
Utilize Labels



Functionality

- Data Input:
  - User Action: Load datasets, which can be source code files (for languages supported by Tree-sitter), OpenMP code, or data in CSV and JSON formats.
  - System Response: Efficiently processes and imports the data, handling any errors related to unsupported formats.
- Model Selection and Activation Extraction:
  - User Action: Specify the neural network model to use—either loading from sources like Hugging Face or importing custom models—and select specific layers or components for activation extraction.
  - System Response: Extracts activation data from the specified model layers, preparing it for clustering.



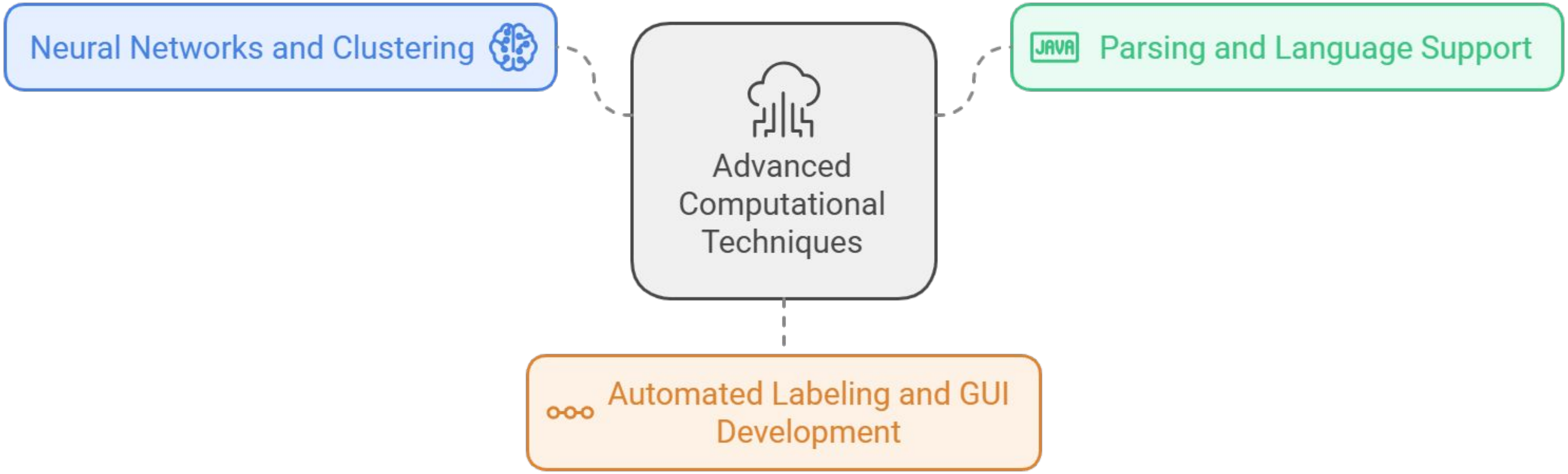
Functionality

- Clustering Algorithms:
  - User Action: Choose from a variety of clustering algorithms (Agglomerative, K-means, Leaders, BIRCH, Hyperbolic Clustering) to process the activation data.
  - System Response: Applies the selected algorithm(s) to cluster the data efficiently, even with large datasets.
- Alignment and Metrics Evaluation:
  - User Action: Define which alignment metrics and criteria (lexical, contextual, or enhanced metrics) to use for evaluating clusters.
  - System Response: Calculates and provides metrics that assess how well clusters align with known concepts or ground truths.

Functionality

- Analysis and Visualization:
  - User Action: Perform in-depth analysis of clustering results and visualize relationships between clusters. Users can also generate comprehensive reports or conduct custom analyses.
  - System Response: Offers tools and visualizations that present insights clearly, aiding in the interpretation of results.
- Automated Labeling:
  - User Action: Utilize automated labeling features, possibly adjusting strategies or editing labels through a graphical user interface (GUI).
  - System Response: Implements LLM labeling with DSPY2 for improved prompts, allowing for both automated and manual label refinement.
- Probing and Reverse Feature Engineering:
  - User Action: Extract meaningful features from latent representations to understand and enhance model performance.
  - System Response: Provides functionalities that facilitate probing into the model's features and relationships.

Technology Considerations



Neural Networks and Clustering

Advanced Computational Techniques

Parsing and Language Support

Automated Labeling and GUI Development

Technology Considerations

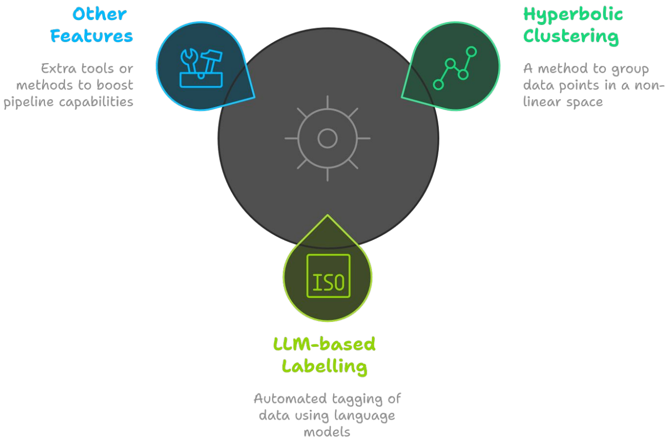
- Flexible neural network models and multiple clustering algorithms optimized for large datasets typical in High-Performance Computing (HPC) environments.
- Tree-sitter for parsing various programming languages but requires extensions or alternatives for OpenMP code support.
- Automated labeling with LLMs and DSPY2, enhancing efficiency but necessitating bias mitigation per ISO/IEC TR 24027:2021; development of a GUI enhances user interaction but adds complexity.

Areas of Concern and Development

- Optimizing performance for large-scale HPC data processing and extending support for OpenMP code.
- Providing comprehensive documentation, usability enhancements, and achieving rigorous testing with at least 60% code coverage.
- Ensuring seamless integration of all system components to meet user and client needs effectively.

Conclusion

- Aims to deliver a robust, user-friendly library fulfilling requirements for latent concept analysis on source code.
- Success depends on overcoming challenges in performance optimization, OpenMP support, documentation quality, and testing standards.
- Committed to refining the product by adhering to engineering standards and user feedback, becoming an invaluable tool for researchers and practitioners.



Other Features

Extra tools or methods to boost pipeline capabilities

Hyperbolic Clustering

A method to group data points in a non-linear space

LLM-based Labelling

Automated tagging of data using language models