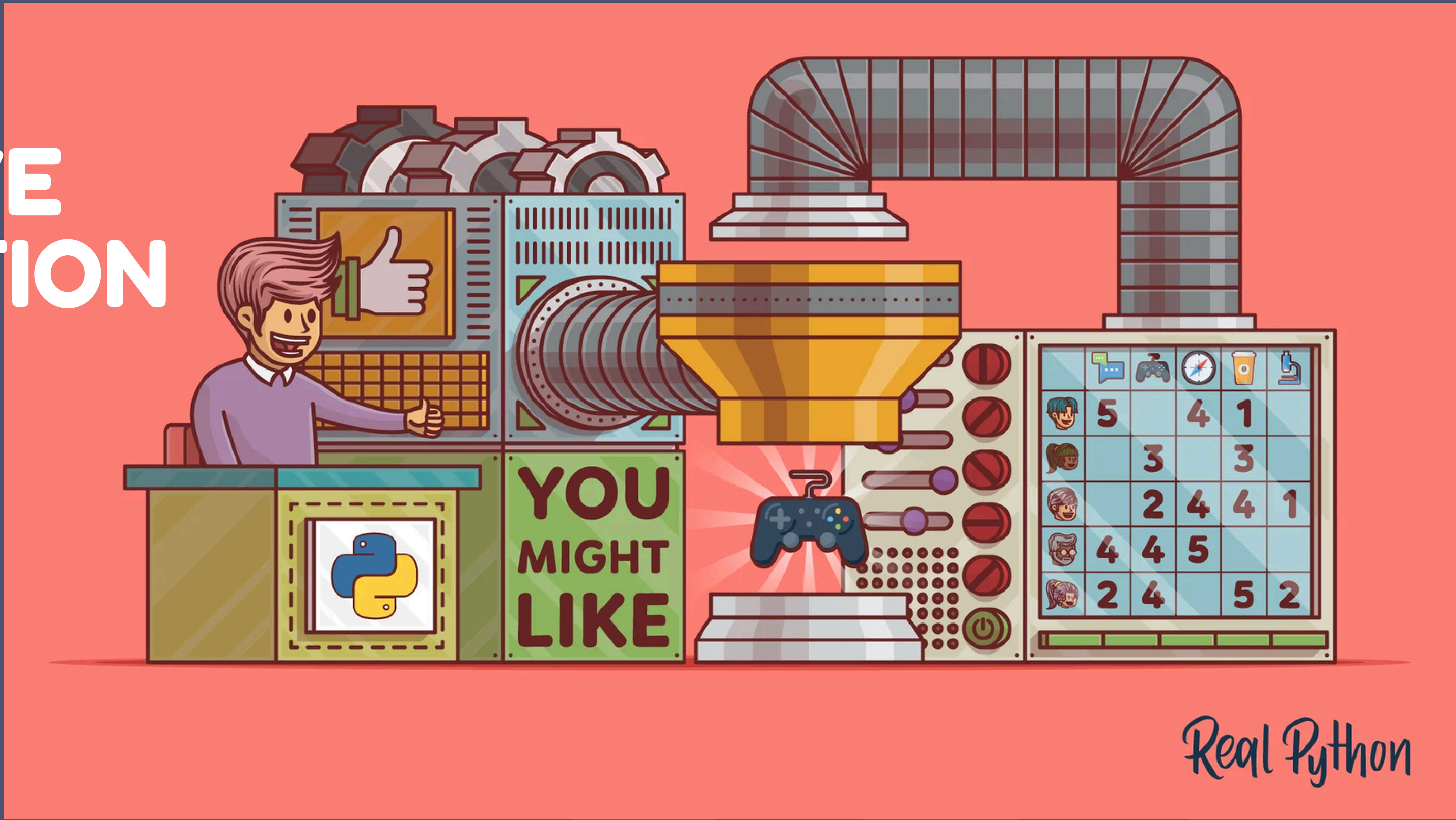
COLLABORATIVE RECOMMENDATION ENGINE WITH A RESTRICTED BOLTZMANN MACHINE



Group 8 : Hamza EL OTMANI, Adam LOZI, Reda BENKIRANE, Aymane EL FAHSI, Mohamed EL IDRISSI

GOAL OF THE PROJECT

Build a movie recommendation system

using **Restricted Boltzman Machines**

to :

- predict
- recommend

movies based on user preferences

CONTEXT

Recommendation systems personalize user experiences.

They are key to boosting **customer loyalty and satisfaction.**

Used by leaders like **Netflix** and **Amazon**.

OBJECTIVES

Predict missing ratings

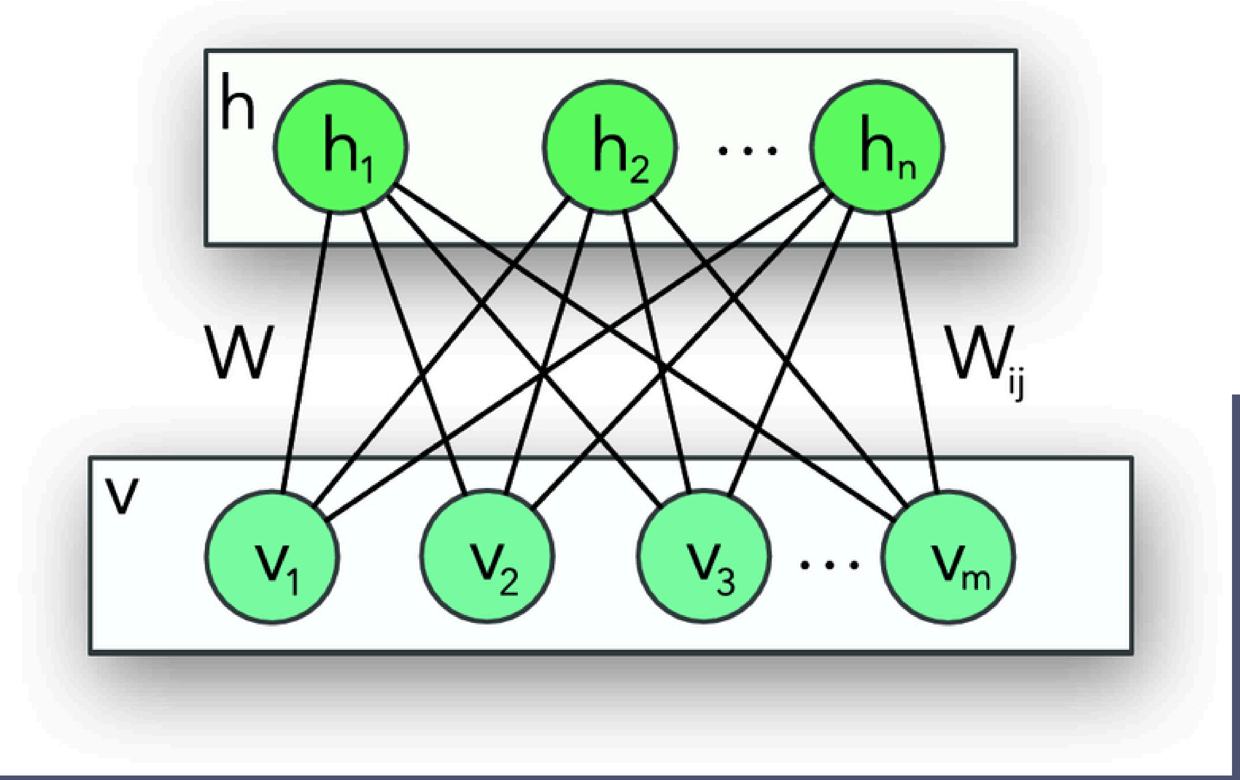
Generate Personalized Recommendations

Build a Scalable Model for Real-World Use

RBMs :

- Capture hidden patterns in user-item interactions
- Manage missing data effectively
- Use probabilities to estimate preferences

WHY RBMS?



BASIC RBM ARCHITECTURE

Visible layer

it represents the input data (a movie ratings matrix)

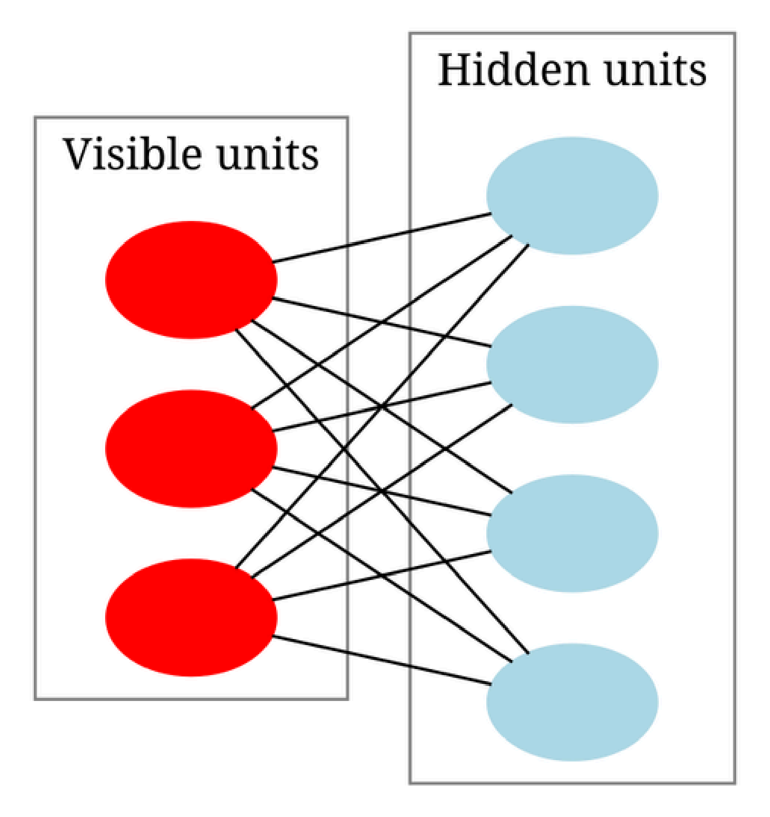
The visible units represent normalized movie ratings, making continuous units ideal for this type of data. The visible layer captures user preferences directly and identifies unrated items

Hidden layer

It captures latentfeatures

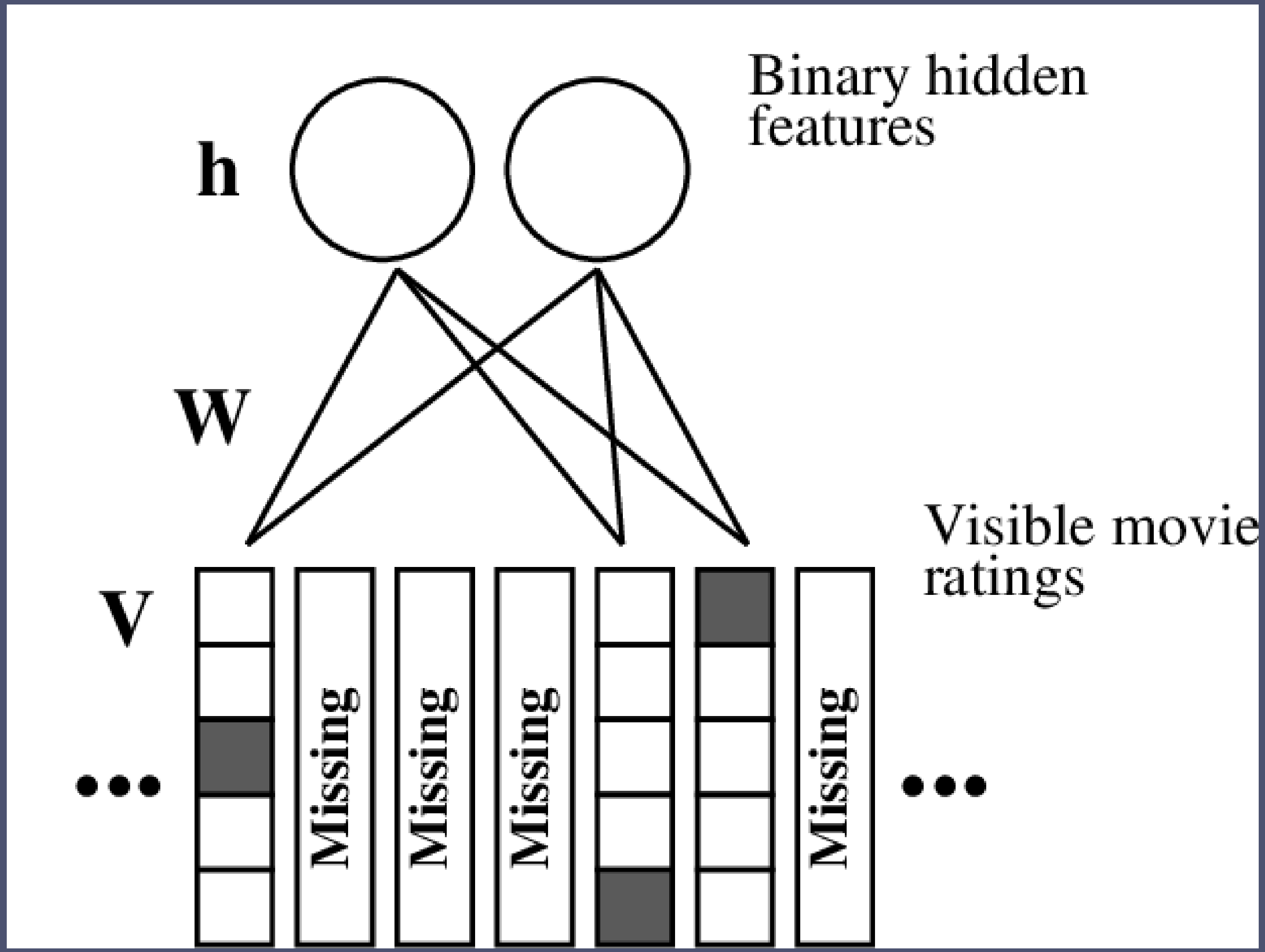
Each unit in the hidden layer represents an underlying, unobserved factor (or "latent feature") that influences user preferences for certain types of movies.

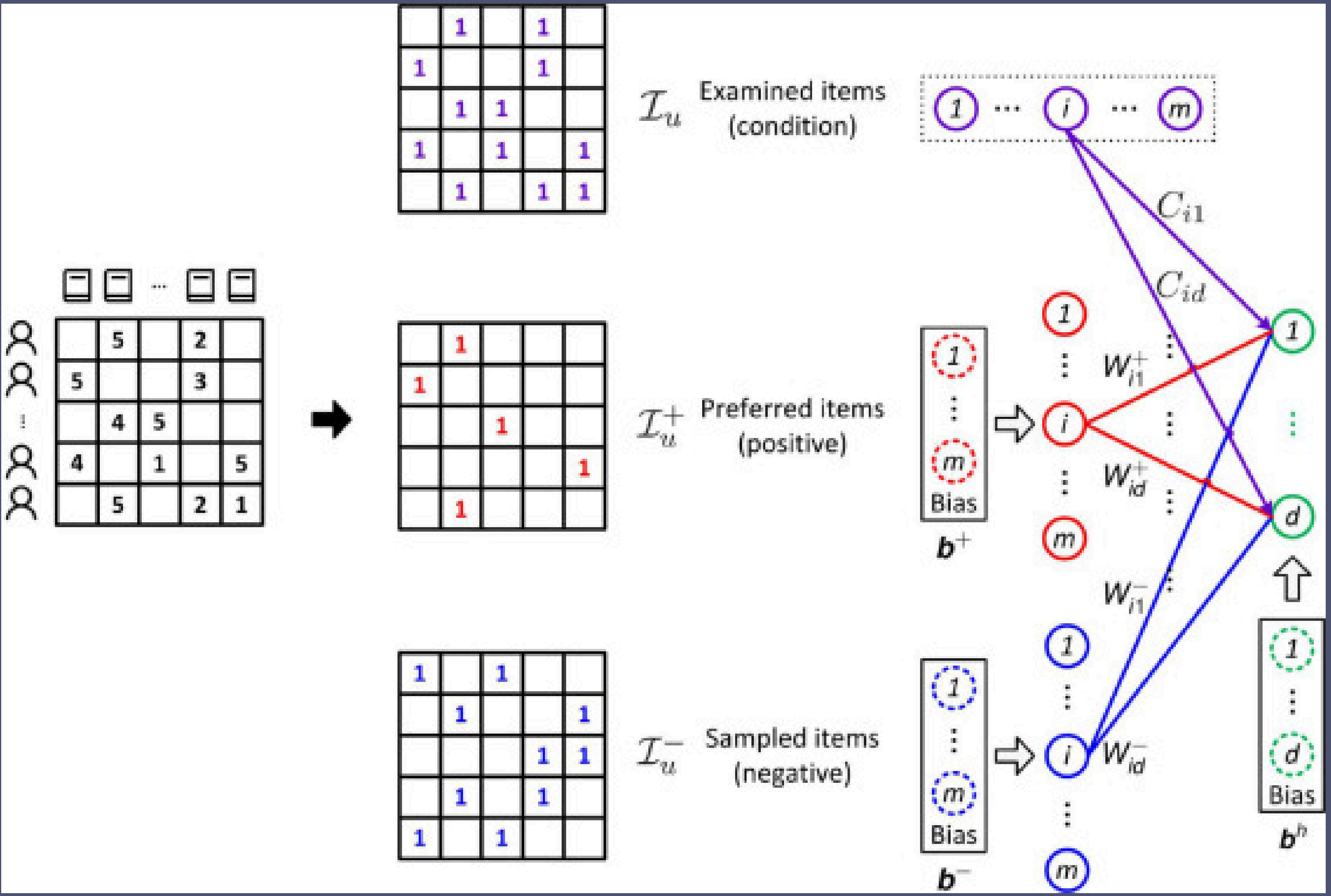
No connections within each layer



binary hidden and continuous visible units

APPLICATION TO COLLABORATIVE FILTERING





DATA PREPARATION

Dataset

Movielens dataset with user-movie ratings

Normalisation

Ratings are normalized to [0,1]

$$\text{Normalized Rating} = \frac{\text{Rating} - \text{Rmin}}{\text{Rmax} - \text{Rmin}}$$

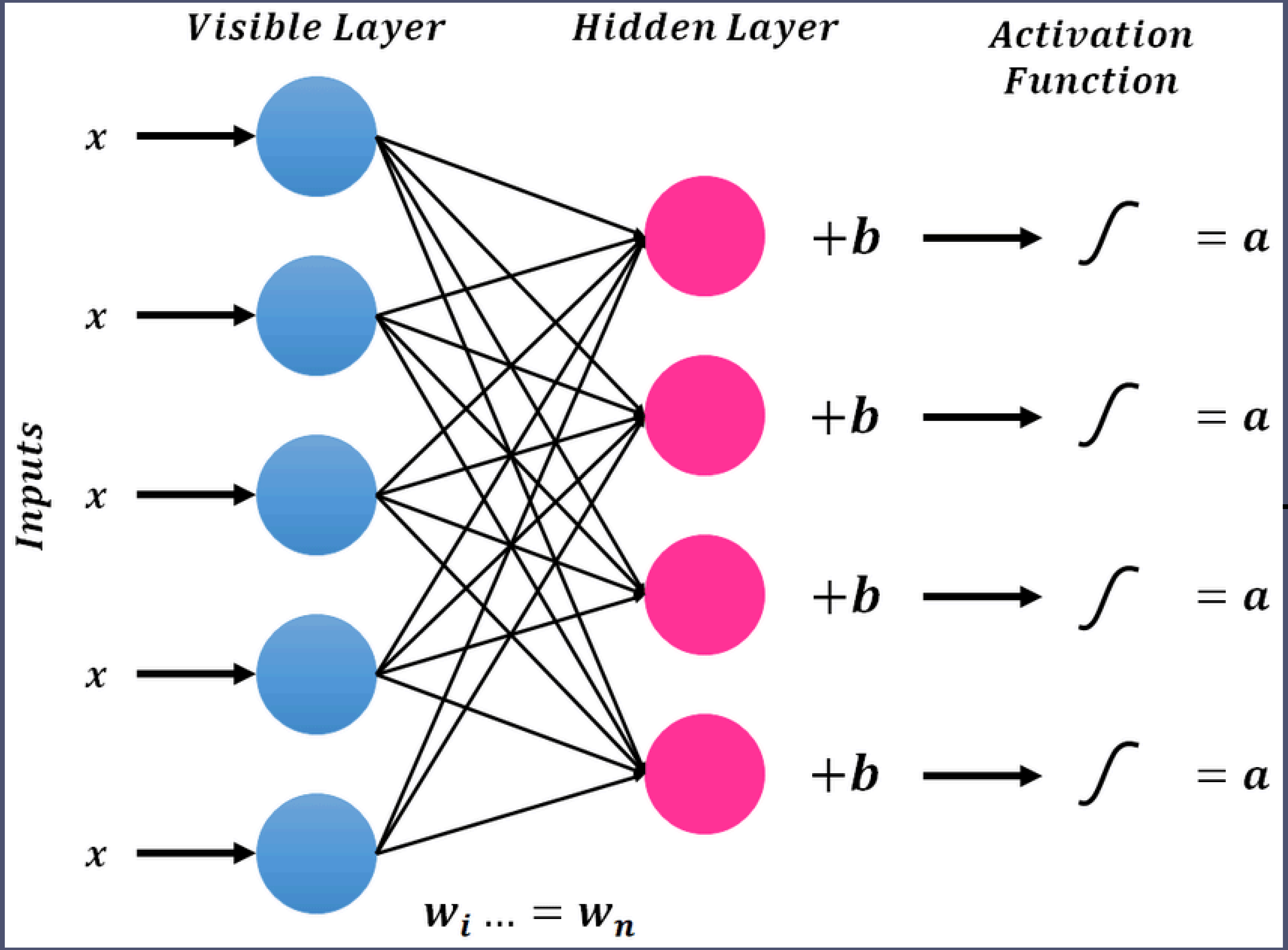
(with Rmin = 0.5 and Rmax = 5, based on the report).

Missing data

Unrated movies are represented by zeros

Model Overview

Ratings



Recommandation

Evaluation Metrics

Hit Rate@K :

$$\text{Hit Rate@K} = \frac{\text{Number of Users with Hits}}{\text{Total Number of Users}}$$

Precision@K:

$$\text{Precision@K} = \frac{\text{Relevant Items Recommended}}{K}$$

Evaluation Metrics

Recall@K:

$$\text{Recall@K} = \frac{\text{Relevant Items Recommended}}{\text{Total Relevant Items}}$$

Experimental results

- The best performance was achieved with 200 hidden units and a learning rate of 0.005.

The RBM achieved:

- Hit Rate@10: 0.9918
- Average Precision@10: 0.6016
- Average Recall@10: 0.0868

Challenges

- Choosing the Appropriate RBM Variant (Bernoulli-Bernoulli vs. Gaussian-Bernoulli
- Hyperparameter Tuning for Optimal Performance
- Balancing Model Complexity and Computational Efficiency
- Interpreting and Validating Model Outputs