Project Title: Gene Classification using Sparse RBM

Introduction

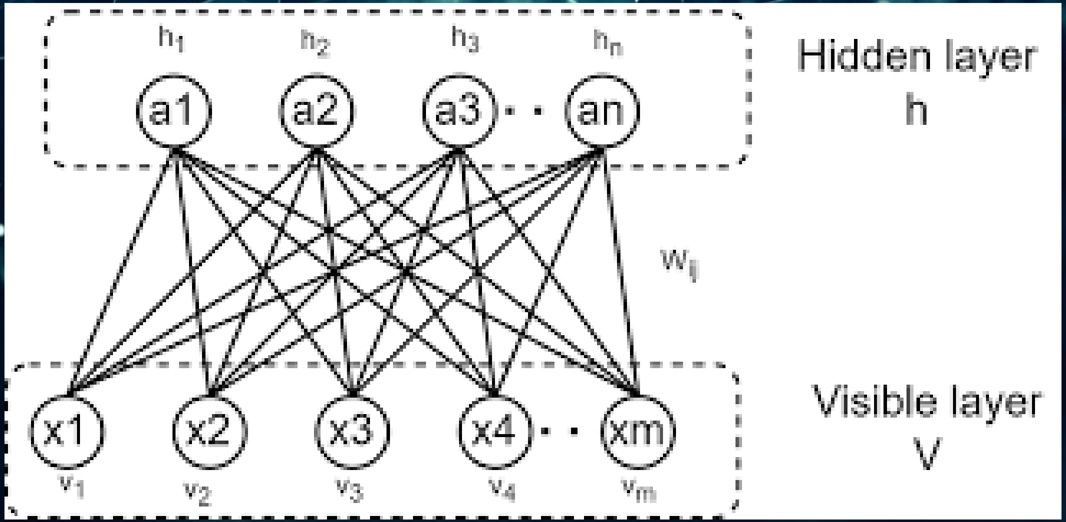
- Gene classification is crucial in bioinformatics for disease diagnosis.
- Deep learning techniques, including RBMs, can extract key patterns from gene expression data.
- Sparse Restricted Boltzmann Machines (RBMs) help in feature selection and dimensionality reduction.

What is a Restricted Boltzmann Machine (RBM)?

- A two-layer neural network used for unsupervised learning.
- Consists of a visible layer and a hidden layer with symmetric connections.
- Learns probabilistic representations of input data.

RBM Architecture

- Visible layer: Represents input features (gene expression values).
- Hidden layer: Learns latent features through weight adjustments.
- No intra-layer connections, only between visible and hidden layers.



Sparse RBM

- Adds sparsity constraint to limit active hidden units.
- Encourages a small subset of neurons to activate for each input.
- Improves feature selection, model generalization, and performance.

Why Use Sparse RBM for Gene Classification?

- Gene expression data is high-dimensional and noisy.
- Sparse RBM helps in reducing irrelevant features while retaining key patterns.
- Provides meaningful feature representations for improved classification.

Dataset Preprocessing

- Data normalization to ensure consistent scale.
- Feature selection to remove redundant or irrelevant genes.
- Splitting into training and testing sets.

Training the Sparse RBM

- Initialize weights and biases.
- Apply the Contrastive Divergence (CD) algorithm for training.

- Enforce sparsity by adding a regularization term.
- Contrastive Divergence (CD), an approximation method for the gradient of the log-likelihood.

RBM Structure Recap

- Visible Layer (v): Represents input data (e.g., pixels in an image).
- Hidden Layer (h): Extracts useful features from input data.
- Weights (W): Connects visible and hidden layers (no intra-layer connections).
- Bias Terms (b,c): For visible and hidden layers.
- The energy func

$$E(v,h) = -\sum_i b_i v_i - \sum_j c_j h_j - \sum_{i,j} v_i W_{ij} h_j$$

Steps to train using Contrastive Divergence

1. Initialize Weights Randomly

   Wij~N(0,0.01)

   Bias terms b and c are initialized to small values

2. Forward pass

   Compute hidden activations using:

   $$P(h_j = 1|v) = \sigma(W_j v + c_j)$$

   Sample hj using a Bernoulli distribution.

3. Forward Pass

   Reconstruct visible units:

   $$P(v_i = 1|h) = \sigma(W_i^T h + b_i)$$

   Sample vi using a Bernoulli distribution.

   Recompute hidden activation using reconstructed v'

   $$P(h'_j = 1|v') = \sigma(W_j v' + c_j)$$

4. Weight Update Using Gradient Approximation

Compute the weight update using the difference between original and reconstructed data

$$\Delta W_{ij} = \eta \left(v_i h_j - v_i' h_j' \right)$$

Similarly, update biases:

$$\Delta b_i = \eta(v_i - v_i')$$

$$\Delta c_j = \eta(h_j - h_j')$$

η is the learning rate.

Feature Extraction & Classification

- Extract compressed features from hidden layer activations.
- Use classifiers like SVM, Random Forest, or Neural Networks.
- Evaluate using metrics such as accuracy, precision, recall, and F1-score.

Performance Evaluation

- Compare classification accuracy with and without Sparse RBM.
- Analyze feature importance and model interpretability.
- Use cross-validation for robustness.

Challenges & Limitations

- Training RBMs requires careful tuning of hyperparameters.
- Computational complexity can be high for large datasets.
- Interpretation of learned features remains a challenge.

Future Directions

- Combining RBMs with deep learning architectures like CNNs or RNNs.
- Exploring alternative sparsity constraints for better feature selection.
- Applying to multi-omics data for broader biological insights.

Conclusion

- Sparse RBMs effectively reduce dimensionality and improve gene classification.
- They enhance feature selection by focusing on key genes.
- Further research is needed to refine their application in bioinformatics.

Thanks!