BMEG3105_Lec09_1155186269

Date: 3 October 2024

Lecturer: Yu Li

Topic: Classification, Logistic Regression, Neural Networks

1. Recap from Last Lecture

- **Hierarchical Clustering**: Basic idea is to keep merging closest clusters until you get one big cluster. Starts with each data point as its own cluster.
- **K-Nearest Neighbor (KNN)**: A simple method for classification. We look at the 'K' closest neighbors and assign the most common class among them.
- **Clustering vs Classification**: Clustering groups similar items; classification assigns labels to items. Clustering doesn't know the number of groups in advance, while classification does.

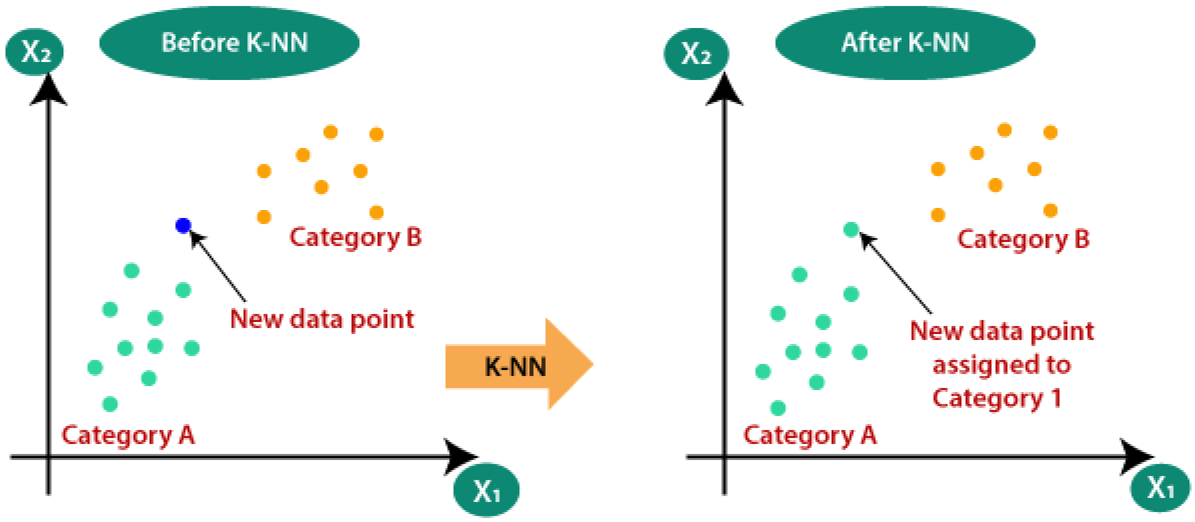
2. Hierarchical Clustering – Recap

- **Steps**:
    1. Compute a **similarity/distance matrix** for all data points.
    2. Merge the **two closest clusters**.
    3. **Update the distance matrix**.
    4. Repeat until only **one cluster remains**.

Example: Genes clustering based on similarities.

3. K-Nearest Neighbor (KNN) Classification

- **How KNN Works**:
    5. **Normalize** data to ensure all features are on the same scale.
    6. Compute **distances** between data points.
    7. Identify the **K nearest neighbors**.
    8. Assign the **most frequent class** among them to the new point.



Example: Given height, weight, and gender data for some individuals, use KNN to predict the gender of a new person.

- **Problems with KNN**:
  - **Storage**: Need to store all the data.
  - **Computation**: Calculating the distance matrix for each new data point is slow, especially for large datasets.

4. Formula-Based Classification

- Instead of storing and comparing every data point, **use a formula** to predict class directly.

Example: If height + weight ≥ 0.5 → male. This is faster than recalculating distances.

- **Adjust the formula**: Different features (height, weight) may have different importance, so we introduce **weights** (w1, w2) and a **bias** (w0). Now, we predict based on w1H + w2W + w0.

5. Logistic Regression

- **What is it?** A more sophisticated version of classification. It uses weights for each feature and calculates probabilities. We use the **logistic (sigmoid) function** to squash predictions between 0 and 1.

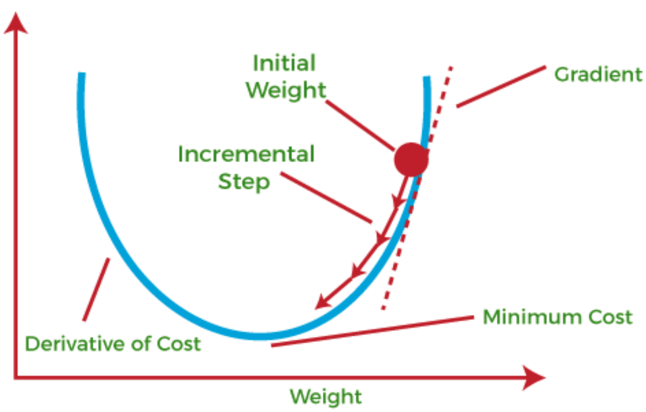
Formula: 1 / (1 + e^(-w1H - w2W - w0))

- **Training the Model**:
   9. Fit the model to **training data** by finding the weights that best separate the classes.
   10. **Loss Function**: Measures how far off our predictions are from the actual labels (goal: minimize this).

6. Gradient Descent Algorithm

The best way to define the local minimum or local maximum of a function using gradient descent is as follows:

- If we move towards a negative gradient or away from the gradient of the function at the current point, it will give the **local minimum** of that function.
- Whenever we move towards a positive gradient or towards the gradient of the function at the current point, we will get the **local maximum** of that function.



- **How to minimize the loss function?**
   - Use **gradient descent**: Start with random weights, calculate the loss, and then update the weights in the direction that reduces the loss. Repeat until the loss is minimized.
- **Steps**:
   11. Initialize weights randomly.
   12. Calculate predictions.
   13. Update weights using w = w + Δw, where Δw is proportional to how far off the predictions are.
   14. Keep doing this until the model converges (i.e., no more big changes in weights).

7. From Logistic Regression to Neural Networks

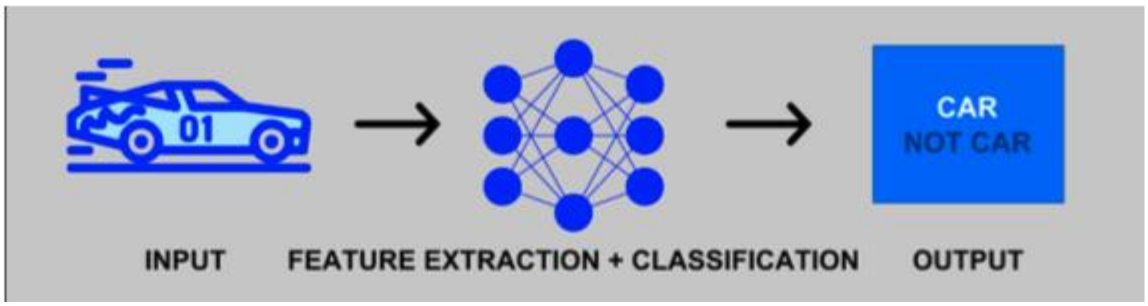
- **Why Neural Networks?** They extend logistic regression by stacking multiple layers of calculations. Each neuron in a neural network computes a weighted sum of inputs and applies an activation function (like the logistic function) before passing it on to the next layer.

Advantages:

- **Fast Prediction**: Once trained, predictions are almost instant.
- **Good with Noisy Data**: NN can handle noisy, complex data better than simpler methods.

Disadvantages:

- **Long Training Time**: Takes a while to train on large datasets.
- **Hard to Interpret**: Neural networks are often considered "black boxes" because it’s difficult to see exactly how they arrive at their predictions.



8. Deep Learning Example: AlphaFold

- **AlphaFold**: A breakthrough application of deep learning in protein folding. This demonstrates how powerful deep learning can be for real-world, complex problems.