IIT ROORKEE

NPTEL ONLINE CERTIFICATION COURSE

# ARTIFICIAL NEURAL NETWORK PART-4

## LECTURE 56

**DR. GAURAV DIXIT**

DEPARTMENT OF MANAGEMENT STUDIES

# ARTIFICIAL NEURAL NETWORKS

- Normalization
  - Scale of [0,1] is typically recommended for neural network models for performance purposes
  - For numeric variables,

$$V_{norm} = \frac{V - \min(V)}{\max(V) - \min(V)}$$

# ARTIFICIAL NEURAL NETWORKS

- Normalization
  - Binary variables (categorical variables with two classes)
    - Create dummy variables: set of values {0, 1}
  - Nominal variables with m (>2) classes
    - Create m-1 dummy variables: set of values {0, 1}
  - Ordinal variables with m (>2) classes
    - Map the values to the set {0, 1/(m-1), 2/(m-1), …, (m-2)/(m-1), 1}

# ARTIFICIAL NEURAL NETWORKS

- Other transformations
  - Transformations which could spread the values more symmetrically can be done for performance purposes
    - Log transform of a right-skewed variable

- Estimation method
  - Least squares and maximum likelihood methods use a global metric of errors (e.g., SSE) to estimate the parameters

# ARTIFICIAL NEURAL NETWORKS

- Estimation method
  - Neural networks use error values of each observation to update the parameters in an iterative fashion (referred as learning)
    - Error for the output node (prediction error) is distributed across all the hidden layer nodes
    - All hidden layer nodes share responsibility for part of the error (referred as node-specific error)
    - Node-specific errors are used to update the connection weights and bias values

# ARTIFICIAL NEURAL NETWORKS

- Back Propagation
  - An algorithm to update weights and bias values of a neural network
  - Error values are computed from output layer back to hidden layers
    - All hidden layer and output layer nodes and all connection weights become part of learning process
  - Node-specific error for output node,

$$err = correction\ factor \times (actual\ value - predicted\ value)$$

$$\theta_{new} = \theta_{old} + learning\ rate \times err$$

$$w_{new} = wold + learning\ rate \times err$$

  - Learning rate controls the rate of change from previous iteration
    - Value is typically a constant in the range [0,1]

# ARTIFICIAL NEURAL NETWORKS

- Back Propagation
  - Node-specific error for hidden nodes
    - Based on *err* value of output node instead of prediction error
    - Steps are same as those used for output node
- Methods for updating weight and bias values
  - Case updating
    - Updating is done after each case or record is run through the network (referred as a trial)
    - When all the records are run through the network, it is referred as ***one epoch, or sweep through the data***
    - Many epochs could be used to train the network

# ARTIFICIAL NEURAL NETWORKS

- Methods for updating weight and bias values
  - Batch updating
    - Updating is done after all the records are run through the network
    - In place of prediction error of the record, sum of prediction errors for all records is used
    - Many epochs could be used to train the network
- Case updating vs. batch updating
  - Case updating yields more accurate results
    - With a longer runtime

# ARTIFICIAL NEURAL NETWORKS

- Stopping Criteria for updating
  - Small incremental change in bias and weight values from previous iteration
  - Rate of change of error function values reaches a required threshold
  - Limit on no. of runs is reached

- Open RStudio

# Key References


- Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data by EMC Education Services (2015)
- Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner by Shmueli, G., Patel, N. R., & Bruce, P. C. (2010)

Thanks…