MODULE 1: Introduction

Structure

1.1 Introduction: Big Data,

1.2 Future Power Systems.

1.3 Big Data Application and Analytics in a Large - Scale Power System: Introduction

1.4 General Applications of Big Data

1.5 Algorithms for Processing Big Data

1.6 Application of Big Data in Power Systems.

Objectives of Course

Learning Objectives

1. To define big data and to explain big data application and analytics to power systems.
2. To explain the role of big data in smart grid communications and optimization of big data in electric power systems.
3. To explain security methods for the infrastructure communication and data mining methods for theft detection in power systems.
4. To explain the application of unit commitment method in the control of smart grid.
5. To explain protection algorithm for transformer based on data pattern recognition

1.1 Introduction: Big Data

- **Definition** – **Big Data** refers to extremely **large datasets** that cannot be processed efficiently using traditional data processing techniques.
- **Characteristics (5 Vs)** – Volume (size), Velocity (speed), Variety (different types), Veracity (quality/reliability), and Value (usefulness).
- **Sources** – Generated from social media, IoT devices, sensors, business transactions, healthcare, and industrial machines.
- **Challenges** – Data storage, processing, real-time analysis, and maintaining data security.
- **Technologies Used** – Hadoop, Spark, NoSQL databases, cloud computing platforms.
- **Applications** – Power systems, healthcare, finance, e-commerce, weather forecasting, and smart cities.

1.2 Future Power Systems.

Characteristics of Future Power Systems

- Increased **decentralization** with microgrids and nanogrids.
- **Expanded communication and monitoring** capabilities.
- **Wider variety of energy sources**, including more renewables.

Research Thrusts in Future Power Systems

- Expansion of the **Smart Grid** to improve monitoring and control.
- **Internet of Things (IoT)** integration for device-level communication.
- Increased **renewable energy penetration**.
- **Microgrid and nanogrid deployment** for local resiliency.

Smart Grid Implications

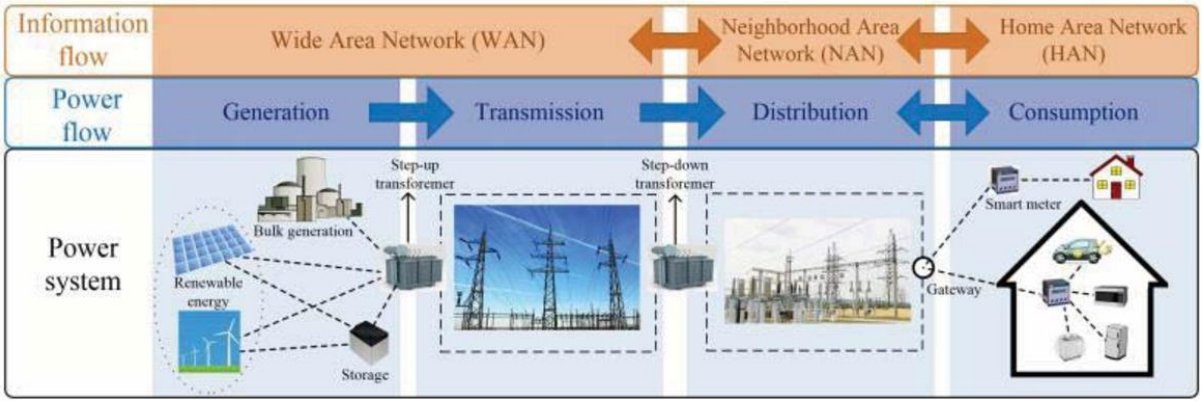
- Enables **data collection** for theft detection and grid optimization.
- Generates **big data challenges** due to the volume and velocity of collected data

IoT-enabled Power Grid

- Allows **communication with all devices** in the network.
- Provides **real-time monitoring of critical infrastructure (CI)**.
- Introduces **big data and cybersecurity challenges**.

Microgrids and Renewables

- Offer **local resiliency** but create **uncertainty for larger grid planning**.
- Renewable energy sources add **availability and reliability challenges**.



1.3 Big Data Application and Analytics in a Large - Scale Power System: Introduction

Big Data Analysis Methods

- Relies heavily on **machine learning techniques**.
- Machine learning is associated with **pattern recognition, statistics, and data mining**.

Emergence of Advanced Techniques

- **Deep learning** (large-scale neural networks) is increasingly used.
- Capable of handling the **size and complexity** of big data.

Applications in Power Systems

- Initially successful in **image recognition** tasks.
- Now being **applied to big data analysis in power systems** for enhanced insights and decision-making.

Big Data Analysis Methods in Power Systems

1. Role of Machine Learning (ML)

1. ML algorithms are essential for handling large-scale, complex datasets generated by modern power grids.
2. ML is associated with **pattern recognition, statistics, and data mining**, enabling automatic detection of patterns and anomalies in grid operations.

2. Applications in Power Systems

- **Load Forecasting:** Predicting electricity demand for better grid management.
- **Fault Detection and Diagnosis:** Recognizing abnormal patterns in sensor data to prevent outages.
- **Power Theft Detection:** Identifying irregular consumption patterns from smart meter data.
- **Renewable Energy Integration:** Forecasting solar and wind power output to optimize generation and storage.

3. Advanced Methods

1. **Deep Learning (DL):** Large-scale neural networks analyze high-dimensional, real-time data such as PMU (Phasor Measurement Unit) streams.
2. **Predictive Analytics:** Uses **historical and live data** to make proactive grid management decisions.

3. **Data Mining:** Extracts **hidden trends from vast power system databases** for planning and optimization.

1.4 General Applications of Big Data

Beyond Syllabus

Sector	Key Use Cases
Healthcare	Personalized treatment, predictive risk, diagnostics
Marketing	Targeting, personalization, campaign optimization
Transportation	Smart traffic, route optimization, congestion control
Finance	Fraud detection, risk analytics, compliance
Retail	Recommendations, dynamic pricing, inventory management
Government	Public safety, fraud control, social services
Manufacturing	Predictive maintenance, efficiency in operations
Education	Student performance analysis, learning insights
Agriculture	Crop monitoring, satellite insights, yield boost
Cybersecurity	Behavioral threat detection, anomaly identification

Key Applications of Big Data

- **Healthcare**: Enables personalized medicine, predictive analytics for **clinical risk, automated reporting, and improved diagnostics** using vast and varied medical data.
- **Marketing & Advertising**: Supports precise customer targeting and **personalized messaging** by analyzing behavior and preferences at scale.
- **Transportation & Smart Cities**: Powers intelligent traffic systems, real-time route planning, congestion prediction, and urban computing solutions like GPS-based path estimation
- **Finance & Banking:** Used for fraud detection, credit scoring, risk management, portfolio optimization, and regulatory compliance
- **Retail & E-commerce**: Enhances customer experience through recommendation systems, inventory optimization, dynamic pricing, and demand forecasting.
- **Government & Public Sector**: Facilitates fraud detection, public health tracking, environmental protection, and efficient public service delivery.
- **Manufacturing & Industrial**: Enables predictive maintenance, smart factory operations, real-time analytics, and supply chain visibility.
- **Education & Learning Analytics**: Helps track student engagement, performance trends, and improve teaching effectiveness
- **Climate, Agriculture & Earth Science**: Supports satellite data analysis for precision agriculture, yield forecasting, and climate trend monitoring.
- **Cybersecurity & Behavioral Analytics**: Detects anomalies, predicts potential fraud or threats, and analyzes user behavior to secure systems.

As per Text Book:

Introduction

- Big Data grew rapidly due to the widespread use of the Internet.
- The Internet acts as:
  - A broadcasting medium
  - A tool for information sharing
  - A platform for collaboration regardless of location
- Massive amounts of data are generated daily.
  - Example: Google processes 20+ petabytes/day of user data.
- Data sources:
  - Web browsing
  - E-commerce purchases
  - Bank/credit card transactions
  - Social networks
- The Big Data explosion impacts all businesses and industries.
- Healthcare and social networking are key examples of Big Data's wide-scale influence.

Applications

1. Health Care – Sources of Big Data

Eight major sources (Gartner 2016):

1. Physicians' free-text notes
2. Patient-generated health data (PGHD)
3. Genomics
4. Physiological monitoring data
5. Publicly available data
6. Credit card & purchasing data
7. Social media data
8. Medical imaging data

Data Volume & Structure:

- Genomics: ~100 GB per patient
- Physicians' notes: Terabytes (unstructured text)
- Medical imaging: Petabytes
- Data ranges from unstructured (notes) → structured formats (genomics).

Health Care Analytics – Importance & Challenges

- Healthcare analytics is vital due to:
  - Growing demand for information
  - Large, diverse datasets
  - Increased competition
  - Complex regulations
- **Innovations:**
  - Precision medicine
  - Value-based care
  - Population health management
- **Value-based care:**

    - Relies on strong data & analytics
    - Requires heavy investment in analytic infrastructure
    - Stakeholders: providers, systems, insurers, pharma, life sciences
- **Key Needs:**
    - Improve quality of information
    - Establish analytics programs
    - Data governance & IT platforms
    - Better information management
- **Sources**: electronic health records, claims, wearable devices, social media, patients.
- **Analytics can:**
    - Detect patterns in information
    - Deliver actionable insights
    - Enable predictive/self-learning systems (predict, infer, suggest alternatives).
- **Benefits:**
    - Reduce costs & improve quality
    - Identify and treat at-risk populations
    - Connect better with patients/consumers
    - Evaluate health interventions' impact
- **Challenges in healthcare business:**
    - Measure & improve clinical performance
    - Ensure clinical quality of care
    - Assess outcomes (mortality, infections, survival, treatment paths)
- **Raw data problem:**
    - Data are siloed → must be transformed into patient-centered structures
    - Data should be accessible to patients & stakeholders
    - Real-time predictive models help solve clinical & operational issues
- **Key goal of Big Data in healthcare:**
    - Identify which data sources improve analytics → support clinical decision-making & healthcare improvements.

2. Social Networking

1. Social networking depends heavily on the Internet.
2. Growth in **size, complexity, and unstructured data** has been explosive.
3. Research uses big data in social networking through **observations, experiments, and simulations**.
4. Companies like **Facebook, Google, Microsoft** collect vast data from calls, texts, clicks, etc.
5. This data is often kept private for **competitive reasons** and **user privacy protection**.

Handling Big Data

- Big data management relies on two key aspects:
    1. **Hardware capability** (storage, bandwidth, processing power).
    2. **Applications/algorithms** (social networking, machine learning, data mining, cloud computing).
- Hardware improvements allow **higher capacity at lower cost**.
- Algorithms enhance scalability, analytics, and predictive capabilities.

- Example: **Cost of 1 GB storage** dropped from **$300,000 (1981) → $1,000 (1994)** → just a few cents today.

1.5 Algorithms for Processing Big Data

Algorithms for Processing Big Data

- Algorithms for big data analysis include **machine learning** and **deep learning.**
- Deep learning is a subset of machine learning but uses advanced methods and architectures.

Machine Learning and Deep Learning Generalities

- **Machine Learning (ML): Uses training data to classify unseen data.**
- **Algorithms define weights for models and update them until convergence.**
- **Two types:**
    1. **Supervised Learning:** Uses labeled data (input + output known). Learns mapping rules.
    2. **Unsupervised Learning:** Uses unlabeled data. Aims to find structure/patterns in data.

Machine Learning

- ML handles large and complex datasets effectively.
- Provides classification and prediction algorithms for quick results.
- Useful in **real-time decision making**, e.g.:
    1. Power market management.
    2. System stability analysis.
    3. Electricity market predictions.
    4. Customer load forecasting.
    5. Power system state estimation.

1.5.1 Artificial Neural Network (ANN) Model

- **Neural networks** are effective at understanding complex data patterns.
- They can detect patterns too complex for humans or traditional computer methods.
- ANN consists of layers of interconnected processing nodes (neurons).
- The Multi-layer Perceptron (MLP) is the most commonly used ANN architecture.
- **A typical ANN has three layers:**
    1. **Input Layer – receives data from the environment.**
    2. **Hidden Layer – processes data and connects input to output.**
    3. **Output Layer – delivers the final response or classification.**
- Neurons are represented as circular nodes; connections (with weights) represent data flow between them.
- Weighted connections link the output of one neuron to the input of another.
- Hidden layers help extract complex information for classification.

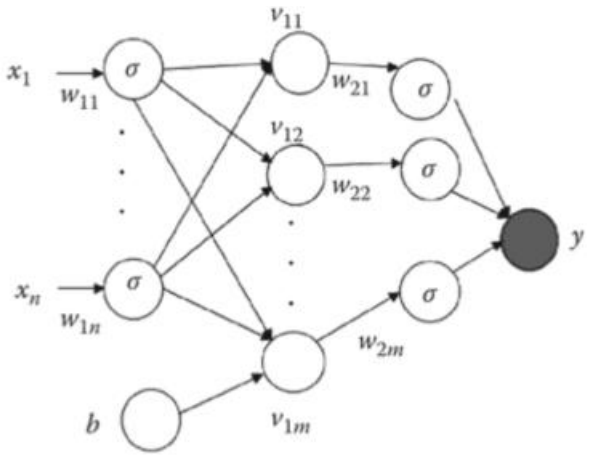


Fig: Feedforward Neural Network

Steps Describing ANN Neuron and Learning Process:

Step-1: Input and Output Definition

1. Input and Output Definition

- The **input layer** is a vector:
  $X = \{x_1, x_2, \ldots, x_n\}$
- The **output** is denoted as:
  y

Step-2:Neuron Computation in Hidden Layer

- A neuron k in the j^{th} hidden layer is calculated as:

$$v_{jk} = \sigma\left(\sum_{i=0}^{n}(w_{ji}x_i + b)\right)$$

- Where:
  - $\sigma(.)$: Activation function
  - $w_j = \{w_{j1}, w_{j2}, ..., w_{jn}\}$: Weight vector
  - b: Bias value

Step-3: Goal of ANN Learning

- The objective is to find a weight set $W = \{w_1, w_2\}$ that **minimizes the error**.
- Error is typically calculated using the **sum of squared errors**.

Step-4: Error (Cost) Function

- The cost of the error is given by:

$$E = [y - f(W, X)]^2$$

- Where:
  - $f(W, X)$: Output predicted by the ANN
  - y: Actual (desired) output

1.5.2 Support Vector Machine (SVM)

- **SVM** is an algorithm like ANN used for classification and regression tasks.
- It identifies a **function mapping** that splits input data into separate classes.
- The goal is to:
  - Map data into a **new space** where it becomes **linearly separable**.
  - Perform **linear classification** in this new space.

Step-1: Regression SVM Problem Statement

- **Training dataset**:
  $\{(X_i, y_i)\}$ for $i = 1, 2, ..., n$
  Where:
  - X_i: input vector
  - y_i: output vector
- **Objective**:
  Find a function $y(X)$ that relates input features to outputs and predicts the output for new inputs.

Step-2: Feature Mapping

- Input X is mapped to **m-dimensional feature space** using:
  $g_j(x), \quad j = 1, 2, ..., m$

Step-3: SVM Output Function

- The SVM function is given by:

$$y(x, \omega) = \sum_{j=1}^{m} \omega_j g_j(x) + b$$

Where:

- ω_j: weights
- $g_j(x)$: feature mapping functions
- b: bias term

Step-4: ε-Insensitive Loss Function

- The loss function ignores errors less than a threshold ε:

$$e_\varepsilon(r, y(x,\omega)) = \begin{cases} 0 & \text{if } |r - y(x,\omega)| \le \varepsilon \\ |r - y(x,\omega)| - \varepsilon & \text{otherwise} \end{cases}$$

Step-5: Optimization Objective

- The goal of SVM regression is to **minimize** both:
  - The model complexity (via weight norm $\|\omega\|^2$)
  - The training error (via slack variables ξ_i, ξ_i^*)

$$\min \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

Step-6: Subject to Constraints

$$\begin{cases} r_i - y(x_i,\omega) \le \varepsilon + \xi_i^* \\ y(x_i,\omega) - r_i \le \varepsilon + \xi_i \\ \xi_i, \xi_i^* \ge 0, \quad i = 1,\ldots,n \end{cases}$$

- ξ_i, ξ_i^*: Slack variables representing **lower and upper training errors**

Step-7: Dual Formulation of SVM Regression

1. Dual Form Equation

- The dual form of the SVM regression function is:

$$y(x) = -\frac{1}{2}\sum_t\sum_s(\alpha_t^+ - \alpha_t^-)(\alpha_s^+ - \alpha_s^-)K(x_t, x_s) - \varepsilon\sum_s(\alpha_s^+ + \alpha_s^-) + \sum_s r_s(\alpha_s^+ - \alpha_s^-) \quad (2.6)$$

2. Constraints

$$\begin{cases} 0 \le \alpha_t^+ \le C \\ 0 \le \alpha_t^- \le C \\ \sum_t(\alpha_t^+ - \alpha_t^-) = 0 \end{cases}$$

3. Kernel Function

- The kernel function $K(x, x_i)$ is used to map inputs to a higher-dimensional space:

$$K(x, x_i) = \sum_{j=1}^{m} g_j(x) g_j(x_i)$$

4. Role of Parameter C

- C controls the **trade-off** between:
  - Model complexity (large C → complex model)
  - Tolerance to error (small C → more error allowed)
- Higher C reduces training error but may increase overfitting.

5. Use of Kernel Functions

- The function $K(x, x_i)$ can be estimated using **kernel functions**.
- A commonly used kernel in SVMs is the **Radial Basis Function (RBF)** kernel.

Step-8: RBF

RBF (Radial Basis Function) Kernel

- The RBF kernel is defined as:

$$K(x_i, x) = \exp\left(-\frac{\|x - x_i\|^2}{2p^2}\right)$$

- This kernel measures similarity between inputs, where p controls the kernel width (spread of the function).

Step-9: Final SVM Regression Function

- After solving the dual problem (Eq. 2.6), the coefficients α_i^+, α_i^- are obtained.
- The final prediction function becomes:

$$y(x) = \sum_{i=1}^{n_{SV}} (\alpha_i - \alpha_i^*) K(x_i, x) + b$$

- This is a **weighted sum of support vectors**.

Example-1:

Imagine you want to predict the price of a house based on its size.

- **Step 1: Data** — You have house sizes (input) and their prices (output).
- **Step 2: Map Features** — Sometimes, the relationship isn't simple, so you transform the size data into a new form (feature mapping) that makes it easier to predict price linearly.
- **Step 3: Find the best function** — SVM tries to find a function (a line or curve) that predicts house prices as closely as possible.
- **Step 4: ε-Insensitive Loss** — SVM ignores small errors within a margin (ε) — small differences between predicted and actual prices are acceptable.
- **Step 5: Optimization** — It balances between fitting the data well and keeping the function smooth to avoid overfitting.
- **Step 6: Kernel Trick (e.g., RBF)** — If data isn't linearly predictable, it uses a kernel (like RBF) to map data into a higher-dimensional space where linear prediction works.
- **Step 7: Final Model** — The model predicts house prices as a combination of important "support vectors" (key training points) weighted properly.

In short:

SVM regression finds a function that **predicts outputs from inputs**, allowing some small errors, while **balancing accuracy and simplicity** — often transforming data to make predictions easier.

Example-2:

Imagine you want to classify emails as Spam or Not Spam based on the number of certain keywords.

- **Step 1: Data** — You have emails with features like the count of suspicious words, and labels (Spam or Not Spam).
- **Step 2: Map Features** — Sometimes the data can't be separated by a simple line, so features are transformed to a new space.
- **Step 3: Find the best boundary** — SVM finds the line (or hyperplane) that best separates Spam emails from Not Spam emails.
- **Step 4: Margin Maximization** — It chooses the boundary that keeps the biggest gap (margin) between the two groups.
- **Step 5: Handle errors** — Some emails might be misclassified to keep the margin large enough (soft margin).
- **Step 6: Kernel Trick** — If emails aren't separable in original features, SVM uses a kernel (like RBF) to separate them in a higher-dimensional space.
- **Step 7: Final Model** — The decision boundary is formed mainly by a few important emails called "support vectors."

1.5.3 Decision-Tree Classifier

1. Overview

- A Decision Tree is used for multi-stage decision-making.
- It follows a recursive top-down approach.

2. Structure

- Root node: The starting point of the tree (no incoming edges).
- Internal nodes: Perform recursive partitioning of input space using decision rules.
- Leaf nodes: Terminal nodes representing the final class labels (no outgoing edges).

3. Working

- The input space is partitioned into **two or more subclasses** at each internal node.
- **Recursion continues** until all data points are classified into appropriate leaf nodes.
- Each **leaf node** is assigned one class that best fits the data reaching that point.

CART Algorithm

- **CART**: Binary decision tree algorithm for **classification** and **regression**.
- **Splitting**: Chooses features that maximize **purity** (e.g., Gini Index, MSE).
- **Classification**: Uses **Gini Impurity** or **Entropy**.
- **Regression**: Uses **variance reduction** or **Mean Squared Error (MSE)**.
- **Tree growth**: Built **recursively**.
- **Pruning**: Applied to reduce **overfitting**.

Step-1: Gini Index

1. **Purpose**:
   - Measures the **impurity** of a node in the **CART algorithm**.
2. **Definition**:
   - Applied to node i, which processes dataset S_i from its parent node.
3. **Formula**:

$$\text{Gini}(S_i) = 1 - \sum_{k=1}^{K} (F_{k,i})^2$$

4. **Where**:
   - $F_{k,i}$: Fraction of data in S_i belonging to class k
   - K: Total number of classes
5. **Interpretation**:
   - **Minimum Gini (0)**: All data belongs to **one class** (pure node).
   - **Maximum Gini**: Data is **evenly distributed** among all classes (highest impurity).

Step-2: Gini Gain

1. **Attribute Selection**:
   - For each attribute $j \in \{1, 2, ..., N\}$, its values A^j are split into two disjoint subsets:
     - A_L^j (left)
     - A_R^j (right)
2. **Subset Properties**:
   - $A_L^j \cup A_R^j = A^j$ (they are **complementary** subsets)
3. **Partitioning**:
   - All possible partitions from set P_i divide dataset S_i at node i into two:
     - Left subset: $L_i(A_L^j, A_R^j)$
     - Right subset: $R_i(A_L^j, A_R^j)$
4. **Fraction of Data**:
   - Let:
     - $F_L\left(A_L^j, A_R^j\right)$: Fraction in left subset
     - $F_R\left(A_L^j, A_R^j\right)$: Fraction in right subset
5. **Constraint**:
   - The sum of fractions equals 1:

$$F_L\left(A_L^j, A_R^j\right) + F_R\left(A_L^j, A_R^j\right) = 1$$

Step-3: Gini Gain Calculation

1. **Fraction of Data per Class**:
   - For class $k \in \{1, 2, ..., K\}$, fractions of data in subsets are:
     - $F_{L,i,k}(A_L^j, A_R^j)$ for left subset
     - $F_{R,i,k}(A_L^j, A_R^j)$ for right subset
2. **Weighted Gini Index** for partition sets A_L^j and A_R^j:

$$\text{Weighted\_Gini}(S_i, A_L^j, A_R^j) = F_{L,i}(A_L^j, A_R^j) \cdot \text{Gini}(L_i(A_L^j, A_R^j)) + F_{R,i}(A_L^j, A_R^j) \cdot \text{Gini}(R_i(A_L^j, A_R^j))$$

3. **Gini for Left and Right subsets**:

$$\text{Gini}(L_i(A_L^j, A_R^j)) = 1 - \sum_{k=1}^{K} \left(F_{L,i,k}(A_L^j, A_R^j)\right)^2$$

$$\text{Gini}(R_i(A_L^j, A_R^j)) = 1 - \sum_{k=1}^{K} \left(F_{R,i,k}(A_L^j, A_R^j)\right)^2$$

4. **Gini Gain** calculation:

$$G(S_i, A_L^j, A_R^j) = \text{Gini}(S_i) - \text{Weighted\_Gini}(S_i, A_L^j, A_R^j)$$

Example: Classify Animals as Cat or Dog based on Tail Length

Animal	Tail Length	Label
1	Short	Cat
2	Short	Cat
3	Long	Dog

Animal	Tail Length	Label
4	Long	Dog
5	Short	Dog

Step 1: Calculate Gini Index for whole data

- Total animals = 5
- Cats = 2, Dogs = 3

$$p_{Cat} = \frac{2}{5} = 0.4, \quad p_{Dog} = \frac{3}{5} = 0.6$$

$$Gini = 1 - (0.4^2 + 0.6^2) = 1 - (0.16 + 0.36) = 0.48$$

Step 2: Split by Tail Length (Short vs Long)

- **Short Tail:** Animals 1, 2, 5

  Cats = 2, Dogs = 1

$$p_{Cat} = \frac{2}{3} \approx 0.67, \quad p_{Dog} = \frac{1}{3} \approx 0.33$$

$$Gini = 1 - (0.67^2 + 0.33^2) = 1 - (0.45 + 0.11) = 0.44$$

- **Long Tail:** Animals 3, 4

  Cats = 0, Dogs = 2

$$p_{Dog} = 1, \quad Gini = 0 \text{ (pure node)}$$

Step 3: Calculate weighted Gini after split

$$\text{Weighted Gini} = \frac{3}{5} \times 0.44 + \frac{2}{5} \times 0 = 0.264$$

Step 4: Calculate Gini Gain

$$GiniGain = \text{Original Gini} - \text{Weighted Gini} = 0.48 - 0.264 = 0.216$$

Conclusion:

- Splitting by tail length reduces impurity by 0.216 (improves classification).
- Decision Tree will split first on **Tail Length**.

Simple takeaway:

Gini Index measures impurity, Gini Gain shows how well a split separates classes, and the tree picks splits with highest gain.

1.5.4 Deep Learning Models

1. Deep learning analyses big data by using complex data representations and abstractions.
2. It is effective for decision-making, information retrieval, and classification, especially with unsupervised data.

3. Deep learning models use stacked layers where each layer performs nonlinear transformations.
4. Data flows hierarchically through these layers, automating feature extraction at each step.
5. More layers enable deeper, more complex feature extraction and richer data representation.
6. The final model output is a highly nonlinear function of the original input data.
7. Deep learning works with both supervised (labelled) and unsupervised (unlabelled) data.
8. **Convolutional Neural Networks (CNNs**) are a type of deep learning model with convolutional and pooling layers.
9. Convolutional layers apply learnable filters to extract features; pooling layers reduce data size via averaging or max-pooling.
10. Lower CNN layers capture simple features, while higher layers capture more abstract, complex features, with layer depth depending on problem complexity.

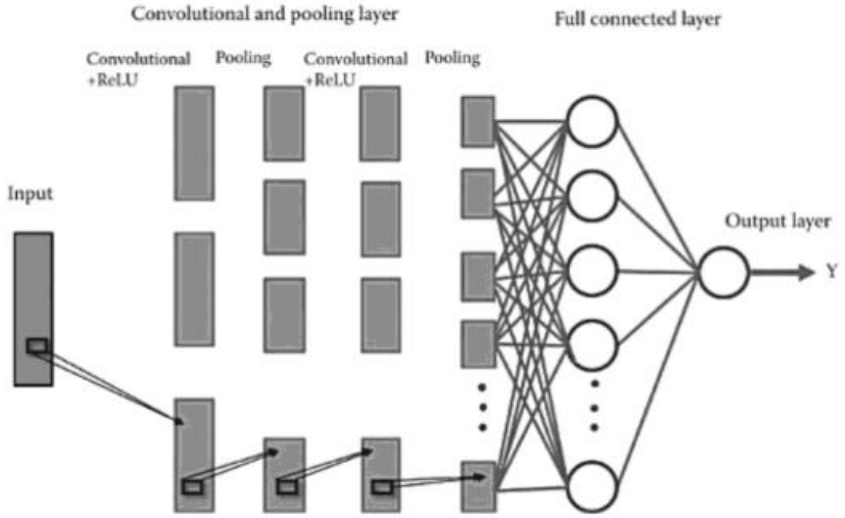


Fig: Scheme of Feedforward CNN

Example: Recognizing Handwritten Digits (0-9)

1. **Input:** An image of a handwritten digit (like '7').
2. **Convolutional Layers:** The CNN applies filters to detect simple features such as edges and curves in the image.
3. **Pooling Layers:** These reduce the image size while keeping important features, making the model faster and more efficient.
4. **Deeper Layers:** Later layers combine simple features to detect more complex patterns like loops or intersections.
5. **Output Layer:** Finally, the model predicts which digit (0-9) the image represents.

Summary:

CNNs automatically learn to extract features from images layer by layer, improving accuracy in recognizing patterns like handwritten digits.

Example: Diagnosing Pneumonia from Chest X-Ray Images

1. **Input:** Chest X-ray images of patients' lungs.
2. **Convolutional Layers:** The CNN applies filters to detect edges, textures, and patterns indicating lung abnormalities.
3. **Pooling Layers:** These reduce the size of the image data while preserving important features like spots or shadows.

4. **Deeper Layers:** Later layers combine these basic features to identify complex signs of pneumonia, such as specific shapes or textures in lung tissue.
5. **Output Layer:** The model outputs a prediction—whether the patient has pneumonia or not.

Why CNN works well here:

- Automatically extracts important features from raw images.
- Handles the complexity and variability in medical images.
- Reduces the need for manual feature engineering by experts.

1.6 Application of Big Data in Power Systems.

1. Big Data in Smart Grid Networks

- **Smart Grid** = Digital upgrade of traditional electricity grids.
- Equipped with **smart meters, IoT sensors, communication networks, and control systems**.
- Generates real-time data on **voltage, current, frequency, consumption, and system health**.

Applications:

1. **Demand Forecasting** – Predicting electricity demand patterns (hourly/daily/seasonal).
2. **Fault Detection** – Identifying outages or abnormal conditions quickly.
3. **Energy Theft Detection** – Data analytics can flag unusual consumption.
4. **Grid Optimization** – Deciding which lines/generators to use for minimum loss.

Example: In India, smart meters in Delhi DISCOMs provide data every 15 minutes, enabling quick detection of theft and billing errors.

2. Phasor Measurement Units (PMU)

- **PMUs** measure voltage and current phasors (magnitude + phase angle) at very high speed (30–60 samples/second).
- Data is time-synchronized using **GPS signals**, enabling **wide-area monitoring**.

Applications of Big Data in PMU:

1. **Early Warning Systems** – Detects oscillations that may cause blackouts.
2. **Grid Stability Analysis** – Helps prevent cascading failures.
3. **State Estimation** – Improves accuracy of real-time grid modeling.

Example: After the 2003 US–Canada blackout, North America deployed thousands of PMUs to provide wide-area situational awareness.

3. Renewable Energy

- Solar and wind are **intermittent** → variability creates challenges for grid operators.
- Big Data analytics uses **weather data, satellite images, and historical generation data** for forecasting.

Applications:

1. **Forecasting Solar/Wind Output** – Improves scheduling of backup power.
2. **Integration into Grid** – Avoids instability by predicting fluctuations.
3. **Market Participation** – Renewable producers can sell power more accurately in day-ahead markets.

Example: Tamil Nadu (India) has one of the largest wind farms. Forecasting tools help balance renewable energy with thermal generation.

4. CIM as Information Standard for Big Data Analytics

- **CIM (Common Information Model)** is an international standard (IEC 61970/61968).
- It provides a **common data model** for exchanging power system information between different devices, utilities, and vendors.

Why Important?

- Power system data comes from many sources → different formats.
- CIM ensures **interoperability** and allows Big Data platforms to integrate datasets smoothly.

Example: European utilities use CIM to integrate data from multiple renewable sources and grid operators.

Big Data Problem in Power System Modeling

Power system optimization problems are very **large-scale** and complex. Big Data creates both **opportunities** and **challenges**.

1. Security-Constrained Unit Commitment (SCUC)

- **Unit Commitment** = deciding which power plants to turn ON/OFF and at what output level.
- SCUC adds **security constraints** → ensures grid remains stable even if a generator/line fails.

Big Data Role:

- Faster simulations with real-time data.
- Uses machine learning to predict demand, prices, and failures.

Example: In Indian Energy Exchange (IEX), SCUC ensures backup power plants are available during peak load.

2. Decomposition Methods to Handle Big Data

- Large power system optimization problems are too complex for one computer.
- **Decomposition** splits them into smaller problems (e.g., regional subproblems).
- Each subproblem is solved separately, then results are combined.

Techniques:

- Benders Decomposition
- Lagrangian Relaxation

Benefit: Faster solutions for large systems (like all-India grid).

3. **Firm Transmission Right (FTR) Problems**

- **FTRs** = financial contracts that protect utilities from congestion charges.
- When transmission lines are congested, FTR holders get compensation.

Big Data Role:

- Predicts congestion patterns using historical + real-time data.
- Optimizes allocation of FTRs in electricity markets.

Example: In PJM (US electricity market), billions of dollars in FTRs are traded annually, supported by Big Data analytics.

4. **Time-Constrained Economic Dispatch**

- **Economic Dispatch** = determining generation levels to meet demand at minimum cost.
- Must be solved **within seconds** in real-time markets.

Big Data Role:

- Uses high-performance computing + data analytics.

- Improves accuracy of cost minimization under tight time limits.

Example: Indian real-time market (RTM) introduced in 2020 requires quick dispatch decisions every 15 minutes, enabled by advanced data platforms.

Outcomes

At the end of the module, students will be able to:
CO-1: Interpret the role of big data and machine-learning methods applicable to power systems and in particular to Smart Grid communications. [L2]

TEXT BOOKS:

Big Data Analytics in Future Power Systems, Ahmed F. Zobaa and Trevor J. Bihl, CRC Press 2019. 2019.

Reference Books/ Link

1. **Big Data Analytics for Power Systems – Big Data Analytics in Power Systems**
2. **Application of Big-Data Analytics in Power System Protection-Lec-37: Application of Big-Data Analytics in Power System Protection**