# Introduction to Neural Networks

Lecture Notes

Introduction to Machine Learning 2024 - 2025

# 1 Neural Networks

## 1.1 Fundamentals of Neural Networks

### 1.1.1 Biological Inspiration

**Neural Networks in Nature**

Neural networks are inspired by the structure and functioning of biological neurons. The human brain processes information using a network of approximately $10^{11}$ neurons, with each neuron connected to an average of $10^4$ other neurons. Below is a breakdown of biological and artificial neural networks:

- **Structure of Biological Neurons:** Biological neurons consist of key components such as dendrites (input), a cell body (processing), and axons (outputs). These components are interconnected by synapses, where chemical signals allow the transfer of information.
- **Comparison with Artificial Neurons:** Artificial neurons mimic the structure of biological neurons mathematically. Instead of chemical signals, artificial neurons use numerical inputs and outputs through mathematical models like weights for processing.
- **Basic Information Processing in Biological Systems:** Biological systems process information through parallel and distributed networks, allowing complex decision-making. Artificial neural networks (ANNs) attempt to replicate such capabilities using algorithms and computational frameworks.

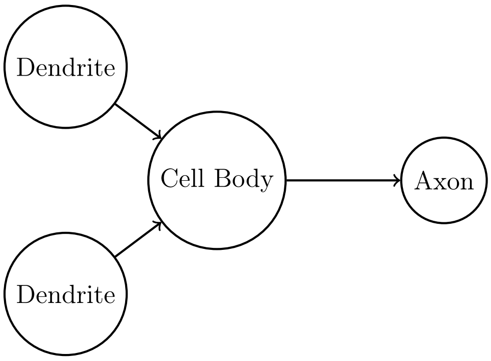


**Figure 1: Structure of a Biological Neuron**

### 1.1.2 Artificial Neural Network Architecture

**Basic Components**

Artificial neural networks are mathematical models composed of interconnected artificial neurons. These components enable supervised and unsupervised learning. Below are the primary components of a neural network:

- **Input Layer:** This layer comprises nodes that receive raw data as input. Each input node corresponds to a feature or variable of the dataset.
- **Hidden Layers:** Hidden layers process input data to extract features using activation functions. They form the computational core of the network, enabling hierarchical feature representation.
- **Output Layer:** The output layer produces the final result of the network, such as classification, prediction, or regression outputs, based on the task.
- **Neurons (Nodes):** Nodes represent the fundamental processing unit of a neural network. Each node receives inputs, applies weights, sums them, and processes them using an activation function.
- **Weights and Biases:** Weights define the strength of connections between two nodes, while biases ensure flexibility in decision boundaries, guiding the learning process based on input relationships.
- **Connections (Edges):** Connections transmit signals between nodes. These edges can be adjusted during network training through algorithms like backpropagation to minimize errors.

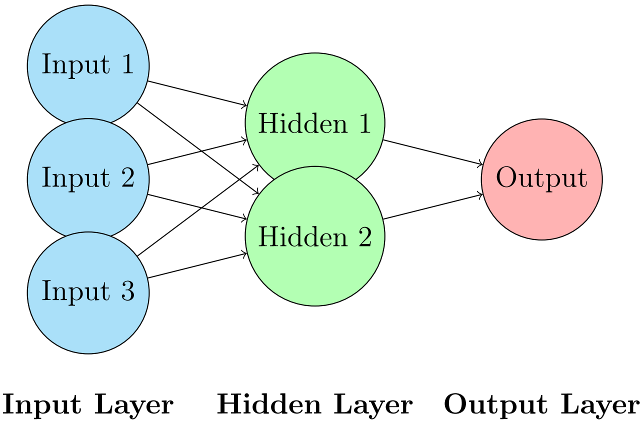


Figure 1: Diagram of an Artificial Neural Network

Artificial neural networks provide the foundation for deep learning by extending the complexity of architectures and leveraging large datasets for precise predictions.

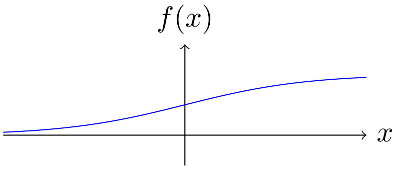
## 1.2 Neural Network Components

### 1.2.1 Activation Functions

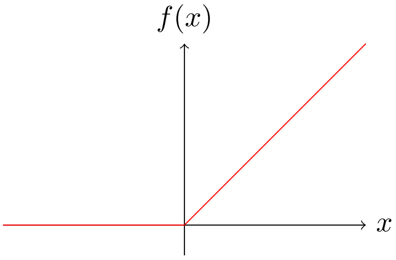
**Common Functions**

Activation functions are mathematical equations applied to the output of neurons in a neural network. They determine whether a neuron should be activated or not, introducing non-linearities that enable neural networks to learn complex patterns.

- **Sigmoid:** Defined as $f(x) = \frac{1}{1+e^{-x}}$. Its output lies between 0 and 1, making it ideal for binary classification tasks.
- **ReLU (Rectified Linear Unit):** Defined as $f(x) = \max(0, x)$. It is computationally efficient and helps mitigate the vanishing gradient problem in deep networks.
- **Tanh:** Defined as $f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$. It outputs values between -1 and 1, making it centered around zero.
- **Softmax:** Converts logits into probabilities. It is defined as $f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$ and is often used in multi-class classification.



**Sigmoid Function**



**ReLU Function**

**Topics Explored**

- **When to use each function:** For example, use Sigmoid for binary classification problems, ReLU for deep networks due to its efficiency, Tanh for zero-centered data, and Softmax for multi-class classification.
- **Practical examples:** Apply Softmax to the final layer of an image classification network or Tanh in RNNs for tasks like sentiment analysis.

### 1.2.2 Network Topology

**Types of Neural Networks**

Network topology refers to the structure of a neural network, including its layers and connections.

- **Feed-forward Networks:** Information flows in one direction, from input to output. There is no looping or feedback in the network.
- **Fully Connected Layers:** Each neuron in one layer is connected to every neuron in the subsequent layer, enabling dense feature representations.
- **Network Depth vs. Width:** Increasing depth (number of layers) enables learning of hierarchical features, while increasing width (number of neurons per layer) allows richer representations at each layer.

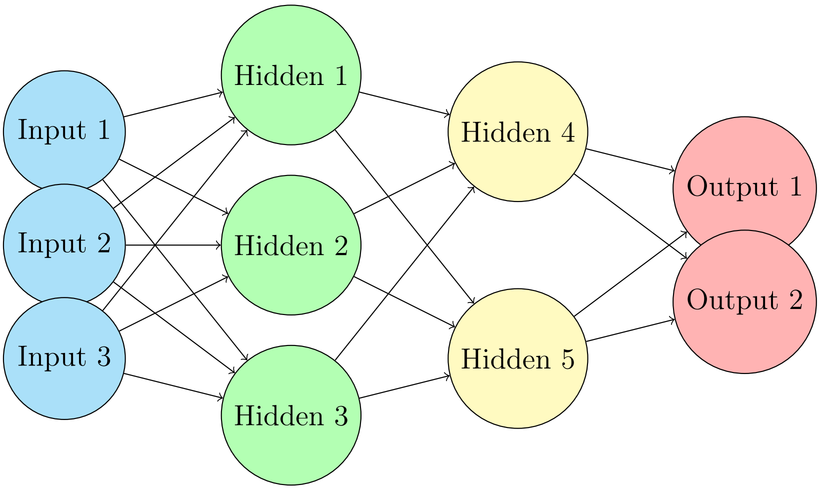


Figure 2: Example of Network Depth and Fully Connected Layers

Neural networks vary in topology depending on the complexity of the task and the data being used. Selecting an appropriate architecture ensures optimal performance for the problem at hand.

## 1.3 Training Neural Networks

### 1.3.1 Forward Propagation

Forward propagation is the process of passing input data through the network to produce an output. This process involves computing the values of the neurons layer-by-layer from the input layer to the output layer.

**Topics Covered**

- **Step-by-step process:**
  1. Input values are fed into the input layer of the neural network.
  2. Weighted sums are computed at each neuron in the hidden and output layers.
  3. An activation function is applied to the weighted sum to produce an output for the neuron.
  4. The process is repeated layer by layer until the final output is generated in the output layer.
- **Mathematical foundations:** If $x_i$ are the inputs and $w_{ij}$ are weights connecting input neuron $i$ to a neuron $j$ in the next layer, the output of neuron $j$ can be expressed as: $z_j = \sum_i w_{ij} x_i + b_j$ followed by applying an activation function $f$: $a_j = f(z_j)$
- **Practical examples:** Forward propagation is used in neural networks for tasks such as computing predicted probabilities for classification or regression outputs in forecasting.

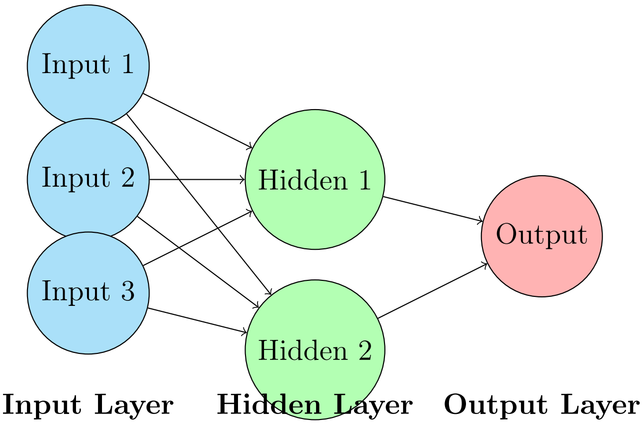


Figure 3: Illustration of Forward Propagation in a Neural Network

### 1.3.2 Backward Propagation

Backward propagation is a key algorithm for training neural networks. It updates the weights of the network by minimizing the error between predicted and true outputs.

**Key Concepts**

- **Gradient Descent:** It is an optimization algorithm used to minimize the cost function by updating weights in the direction of the steepest descent. $w_{new} = w - \eta \cdot \frac{\partial J}{\partial w}$ where $w$ is the weight, $\eta$ is the learning rate, and $J$ is the cost function.
- **Learning Rate:** The step size at which weights are updated. A smaller learning rate takes longer to converge, while a larger rate may overshoot the optimal solution.
- **Cost Functions:** Measures the error of the model. Common examples include Mean Squared Error (MSE) for regression tasks and Cross-Entropy Loss for classification tasks.
- **Weight Updates:** Adjusts weights using computed gradients to minimize the cost function.

Backward propagation combines forward propagation with the gradient of the cost function to iteratively improve the performance of the network.

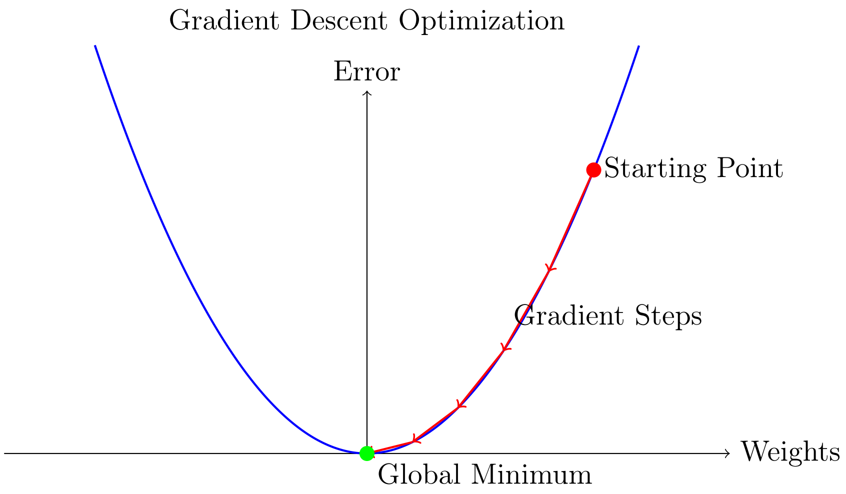


Figure 4: Visualization of Gradient Descent showing how weights are updated to minimize error. The red arrows show the path taken by the optimization algorithm from a starting point (red dot) to the global minimum (green dot). Each step moves down the error surface in the direction of steepest descent.

# 2 Deep Learning Introduction

## 2.1 From Shallow to Deep Networks

Deep learning represents a significant evolution from traditional neural networks, characterized by multiple layers of processing that enable the learning of hierarchical representations from data.

### 2.1.1 Definition of Deep Learning

Deep learning refers to artificial neural networks with multiple hidden layers (typically more than 2-3) that can automatically learn representations of data with multiple levels of abstraction.

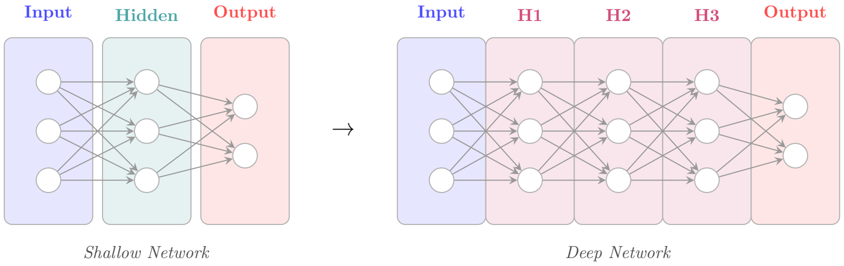


Figure 5: Comparison of neural network architectures showing the evolution from shallow to deep networks. The deep network shows full connectivity between adjacent layers, demonstrating the increased complexity and capacity for feature learning.

### 2.1.2 Advantages and Challenges

**Advantages:**

- Hierarchical Feature Learning
- Better Generalization
- Automatic Feature Engineering
- Scale with Data and Compute

**Challenges:**

- Requires Large Amounts of Data
- Computational Intensity
- Vanishing/Exploding Gradients
- Model Interpretability

### 2.1.3 Popular Architectures

**1. Convolutional Neural Networks (CNNs)**

CNNs are specialized for processing grid-like data, such as images. They use convolution operations to detect spatial patterns.

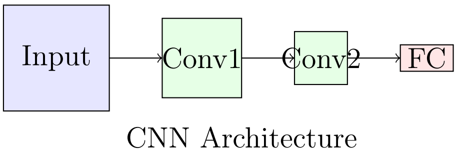


Figure 6: Basic CNN Architecture

**2. Recurrent Neural Networks (RNNs)**

RNNs are designed for sequential data processing, maintaining an internal state (memory) that allows them to process sequences of inputs.

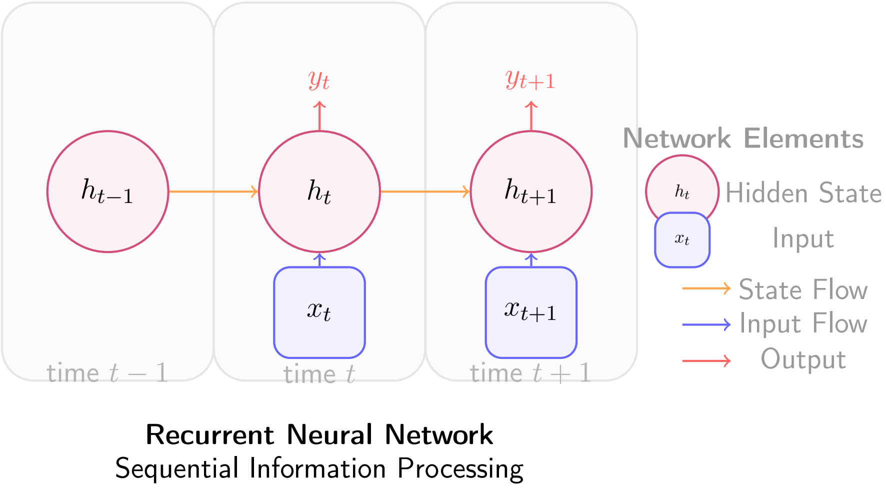


Figure 7: Modern visualization of an RNN showing how information flows through time. Hidden states maintain temporal context while processing sequential inputs and generating outputs at each time step.

**3. Transformers**

Transformers represent a revolutionary architecture that uses self-attention mechanisms to process sequential data in parallel, making them highly effective for natural language processing tasks.

## 2.2 Applications and Impact

Deep learning has revolutionized numerous fields:

- **Computer Vision:** Object detection, image classification, facial recognition
- **Natural Language Processing:** Machine translation, text generation, sentiment analysis
- **Speech Recognition:** Voice assistants, real-time translation

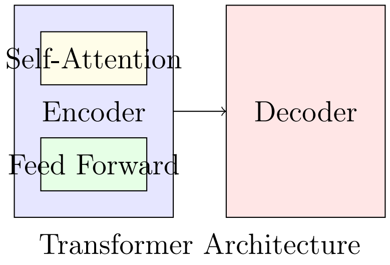


Figure 8: Simplified Transformer Architecture

- **Healthcare:** Disease diagnosis, drug discovery
- **Autonomous Systems:** Self-driving cars, robotics

The field continues to evolve rapidly, with new architectures and applications emerging regularly. Understanding these fundamental concepts provides a foundation for exploring more advanced topics in deep learning.

# 3 Discussion and Future Trends

## 3.1 Ethical Considerations

The rapid advancement of machine learning technologies brings forth critical ethical considerations that demand careful attention from practitioners, researchers, and society at large.

### 3.1.1 Bias in AI

- **Sources of Bias**
  - Data bias: Training data may underrepresent certain demographics
  - Algorithm bias: Model architectures may favor certain patterns
  - Historical bias: Past prejudices encoded in historical data
  - Selection bias: Skewed sampling in data collection
- **Mitigation Strategies**
  - Regular bias audits
  - Diverse training data collection
  - Implementation of fairness metrics
  - Transparent model documentation

### 3.1.2 Privacy Concerns

1. **Data Collection**
   - Informed consent procedures
   - Minimal data collection principle
   - Clear usage policies and documentation
2. **Data Processing & Storage**
   - Standard encryption protocols
   - Strict access controls
   - Regular security audits

3. **Model Development**
    - Implementation of differential privacy
    - Federated learning approaches
    - Privacy-preserving ML techniques

### 3.1.3 Responsible AI Development

| Principle | Description |
|---|---|
| Transparency | Models and decisions must be explainable and interpretable |
| Accountability | Clear responsibility chain for AI decisions |
| Fairness | Equal treatment across all demographic groups |
| Safety | Robust testing and validation procedures |
| Human Oversight | Maintaining human control over AI systems |

## 3.2 Future Trends

### 3.2.1 Current Challenges

- **Technical Challenges**
    - Model interpretability limitations
    - Computational efficiency barriers
    - Data quality and quantity issues
    - Model robustness concerns
- **Operational Challenges**
    - Infrastructure scaling requirements
    - Cost optimization needs
    - Talent acquisition difficulties
    - Integration complexity management

### 3.2.2 Emerging Architectures

**Transformer Evolution**
- Sparse attention mechanisms
- Efficient transformer variants
- Domain-specific architectures

**Neural Architecture Search**
- Automated architecture optimization
- Hardware-aware design principles
- Multi-objective optimization strategies

**Hybrid Models**
- Symbolic-neural integration approaches
- Physics-informed neural networks
- Neuro-symbolic AI systems

### 3.2.3 Industry Applications

| Industry | Current Impact | Future Potential |
|---|---|---|
| Healthcare | High | Very High |
| Climate | Medium | Very High |
| Finance | High | High |
| Manufacturing | Medium | High |

### 3.3 Conclusion

Key takeaways from this discussion:

- Ethics and privacy are fundamental considerations
- Technical innovation continues at rapid pace
- Industry applications are expanding significantly
- Responsible development remains crucial
- Future impact will be transformative

## 4 Learning Outcomes

By the end of this lecture, students should be able to:

- Understand the basic architecture of neural networks
- Explain the role of different activation functions
- Implement a simple neural network
- Analyze the training process
- Evaluate ethical considerations in AI applications