Artificial Intelligence: From Neurons to Networks

Introduction

* AI is becoming a part of our everyday experience.
* Neural networks are at the heart of AI.
* This article explores how neural networks function and mimic the human brain.

What Is a Neural Network?

* A neural network is a web of interconnected digital cells.
* These cells communicate and learn from one another, like neurons in the human brain.
* They detect relationships, identify trends, and make predictions.
* Each neuron (or node/unit) acts as a small processor.
* **Example:** A child recognizing a dog after seeing many dogs.

How Neural Networks Work Digitally

* Each neuron receives input numbers (e.g., pixel values).
* Every connection has a weight, like the strength of a synapse.
* Neurons multiply inputs by weights, sum them, and use an activation function.
* The output is sent to the next layer.
* During training, the network adjusts weights until it learns the right patterns.

Perceptron

* The Perceptron is the simplest type of Artificial neural network.

* Invented by Frank Rosenblatt in 1958, inspired by neurons in the brain.

Perceptron Components

1. **Inputs (x?, x?, x?, ?)**

   * Features or data points provided to the model.

   * **Example:** Email spam filter inputs like hyperlinks, keywords, or sender patterns.

2. **Weights (w?, w?, w?, ?)**

   * Numerical values indicating the importance of each input.

   * The perceptron multiplies each input by its respective weight.

3. **Summation (?)**

   * Calculates the total weighted sum:

     * `z = w1x1 + w2x2 + ... + wnxn + b`

   * `b` is the bias, which adjusts the output threshold.

4. **Activation Function (Step Function)**

   * Decides whether the neuron should "activate."

     * If the sum exceeds a threshold ? Output = 1

     * If the sum is below the threshold ? Output = 0

   * Enables the perceptron to classify data.

5. **Output (y)**

   * The perceptron's final decision based on inputs and weights.

    *   **Example:** Identifying whether an email is spam or not.

Artificial Neural Networks (ANNs)

*   Connecting multiple neurons together forms an ANN.
*   Inspired by the structure of the human brain.
*   ANNs solve complex problems like image/voice recognition, weather prediction, and medical diagnosis.
*   Data is analyzed through multiple layers of connected nodes.

ANN Layers

1. **Input Layer**
    *   Receives raw information (e.g., pixel values, numerical data, sound waveforms).
    *   Captures and passes information for analysis.

2. **Hidden Layers**
    *   Handle internal computation.
    *   Each neuron receives inputs, applies weights, performs summation, and uses an activation function.
    *   Learns meaningful patterns and relationships.

3. **Output Layer**
    *   Generates the final prediction or classification.
    *   **Example:** Determining if an image shows a cat or a dog.

How Learning Happens in an ANN

1. **Forward Propagation**
   * Information travels from the input layer through hidden layers to the output layer.
   * Each neuron processes and forwards the data.
   * The final layer produces a prediction.

2. **Backward Propagation (Backpropagation)**
   * The model compares its prediction with the actual result and calculates the error.
   * The error is sent backward through the network.
   * The system adjusts neuron weights to minimize the error.
   * This cycle repeats until the network achieves high accuracy.

Convolutional Neural Networks (CNNs)

* Specialized for processing visual data (images/videos).
* Foundation of modern computer vision systems.
* Used in facial recognition, autonomous driving, medical image analysis, and object detection.
* Focuses on local patterns (shapes, edges, textures).

CNN Layers

1. **Convolutional Layer**
   * Applies filters (kernels) that slide over the input image.
   * Performs convolution to detect specific features.
   * Early layers detect lines/edges; deeper layers recognize shapes/objects.

2. **Activation Function (ReLU)**

   * Introduces non-linearity after convolution.

   * Allows the network to understand complex relationships.

3. **Pooling Layer**

   * Reduces the size of feature maps while keeping important information.

   * Max Pooling selects the highest value from a region.

   * Makes the model faster, reduces memory, and prevents overfitting.

4. **Fully Connected Layer**

   * Flattened features are passed into fully connected (dense) layers.

   * Combines detected features to make the final prediction.

5. **Output Layer**

   * Produces the result, often using a Softmax activation function for probabilities.

   * Chooses the class with the highest probability as its prediction.

CNN Learning

1. **Forward propagation:** Image passes through all layers, producing a prediction.

2. **Loss calculation:** Compares the prediction to the correct label.

3. **Backward propagation:** Adjusts filters and weights to reduce future errors.

Recurrent Neural Networks (RNNs)

* Designed to recognize sequential data (where order/context matters).
* Has a "memory" to retain information from previous inputs.
* Effective for language translation, speech recognition, time-series forecasting, and text generation.

How RNNs Work

1. Takes an input (e.g., a word or a number).
2. Produces an output based on the current input and previous information.
3. Uses a hidden state (short-term memory).
4. This process is repeated for every element in the sequence.

* **Example:** Predicting the next word in a sentence ("The cat is on the ___.").

RNN Unit Details

* Receives:
    * Input at the current time step (x?)
    * Hidden state from the previous step (h???)
* Produces:
    * Output (y?)
    * New hidden state (h?)
* The hidden state is passed along, creating a chain-like structure.

Limitations of Basic RNNs

* Struggles with long sequences.

* Suffers from the vanishing gradient problem.

Types of Recurrent Neural Networks

1. **Vanilla RNN (Simple RNN)**
    * Basic RNN where outputs are passed from one step to the next.
    * Best suited for small/short sequential data.
    * **Example:** Predicting the next character in a short text.
    * **Limitation:** Struggles to remember information over long sequences.

2. **Long Short-Term Memory Network (LSTM)**
    * Designed to overcome the forgetting problem.
    * Uses gates (input, forget, and output) to control information flow.
    * **Input Gate:** Decides what new information should enter memory.
    * **Forget Gate:** Determines what old information should be removed.
    * **Output Gate:** Controls what part of memory is used to produce the output.
    * **Example:** Text generation models remembering context from sentences earlier.
    * **Applications:** Text and speech recognition, sentiment analysis, music composition, and text generation.

3. **Gated Recurrent Unit (GRU)**
    * Simplified version of LSTM (combines forget and input gates).
    * Faster computations while maintaining similar performance.
    * **Example:** Used in predictive text keyboards or time-series forecasting.
    * **Advantages:** Faster training, fewer parameters, performs well with moderate sequence lengths.

4. **Bidirectional RNN (Bi-RNN)**

* Processes data in two directions (forward and backward).

* Considers both past and future context.

* **Example:** Sentiment analysis (understanding "not" later can change earlier words).

* **Applications:** Speech and handwriting recognition, language modeling, and machine translation.

Conclusion

* Understanding the human mind and computational intelligence deepens as AI advances.
* The journey from biological neurons to artificial networks reflects humanity?s pursuit of systems that think, learn, and understand.