

Machine Learning in Python

Artificial Neural Networks and Deep Learning

Cristian A. Marocico, A. Emin Tatar

Center for Information Technology
University of Groningen

Friday, July 11th 2025

Outline

- 1 Introduction to Neural Networks
- 2 Convolutional Neural Networks (CNNs)
- 3 Recurrent Neural Networks (RNNs)
- 4 Long Short-Term Memory (LSTM) Networks
- 5 Generative Adversarial Networks (GANs)
- 6 Autoencoders
- 7 Large Language Models (LLMs)

What are Neural Networks?

What are Neural Networks?

Definition

What are Neural Networks?

What are Neural Networks?

Definition

Neural networks are computational models inspired by the human brain, consisting of interconnected nodes (neurons) that process and transmit information. They are used to recognize patterns, classify data, and make predictions based on input features.

Artificial Neurons: A Biological Inspiration

Artificial Neurons: A Biological Inspiration

An **artificial neuron** is a simplified model of a biological neuron. It receives inputs, applies weights to them, sums them up, and passes the result through an activation function to produce an output. This process roughly mimics how biological neurons transmit signals in the brain.

Activation Functions

Activation Functions

Activation functions introduce non-linearity into the model, allowing it to learn complex patterns. Common activation functions include:

- **Sigmoid**: Maps input values to a range between 0 and 1, useful for binary classification.
- **ReLU (Rectified Linear Unit)**: Outputs the input directly if it is positive; otherwise, it outputs zero. It is widely used due to its simplicity and effectiveness in deep networks.
- **Tanh (Hyperbolic Tangent)**: Maps input values to a range between -1 and 1, often used in hidden layers.

Structure of a Neural Network

Structure of a Neural Network

A neural network consists of artificial neurons organized in layers:

- **Input Layer:** Receives the input features.
- **Hidden Layers:** Intermediate layers that process the inputs through weighted connections and activation functions.
- **Output Layer:** Produces the final output, which can be a classification, regression value, or other predictions.

Information flows from the input layer through the hidden layers to the output layer, with each neuron applying its activation function to the weighted sum of its inputs (**feedforward** process).

Training Neural Networks

Training Neural Networks

Training a neural network involves adjusting the weights of the connections between neurons to minimize the difference between the predicted output and the actual target values. This process typically involves:

- **Forward Propagation:** The input data is passed through the network, and the output is computed.
- **Backpropagation:** The error (difference between predicted and actual output) is propagated backward through the network to update the weights using gradient descent.

Forward Propagation

Forward Propagation

Forward propagation is the process of passing input data through the network to compute the output. Each neuron computes a weighted sum of its inputs, applies an activation function, and passes the result to the next layer. This process continues until the output layer is reached.

Backpropagation

Backpropagation

Backpropagation is the algorithm used to update the weights of the network based on the error calculated during forward propagation. It involves:

- Calculating the gradient of the loss function with respect to each weight.
- Propagating the error backward through the network to compute the gradients.
- Updating the weights using gradient descent, which adjusts the weights in the direction that reduces the error.

Loss Functions

Loss Functions

A loss function quantifies the difference between the predicted output and the actual target values. It guides the training process by providing a measure of how well the network is performing. Common loss functions include:

- **Mean Squared Error (MSE)**: Used for regression tasks, it calculates the average of the squared differences between predicted and actual values.
- **Cross-Entropy Loss**: Used for classification tasks, it measures the dissimilarity between the predicted probability distribution and the actual distribution of classes.

Gradient Descent

Gradient Descent

Gradient descent is an optimization algorithm used to minimize the loss function by iteratively updating the weights of the network. The weights are adjusted in the direction of the negative gradient of the loss function with respect to the weights. The learning rate determines the size of the step taken in each iteration.

Deep Learning

Deep Learning

Deep learning is a subset of machine learning that focuses on training deep neural networks with many layers. These networks can learn complex representations of data, enabling them to perform tasks such as image recognition, natural language processing, and speech recognition. Deep learning has achieved significant breakthroughs in various fields due to its ability to automatically learn features from raw data without extensive feature engineering.

Deep Neural Networks

Deep Neural Networks

Deep neural networks (DNNs) are neural networks with multiple hidden layers between the input and output layers. The depth of the network allows it to learn hierarchical representations of data, capturing complex patterns and relationships. DNNs have been successful in tasks such as image classification, speech recognition, and natural language processing.

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a specialized type of neural network designed for processing structured grid data, such as images. They use convolutional layers to automatically learn spatial hierarchies of features, making them particularly effective for image recognition tasks. CNNs consist of:

- **Convolutional Layers:** Apply convolution operations to extract features from the input data.
- **Pooling Layers:** Downsample the feature maps to reduce dimensionality and retain important features.
- **Fully Connected Layers:** Connect every neuron in one layer to every neuron in the next layer, typically used in the final layers for classification tasks.

Applications of CNNs

Applications of CNNs

CNNs are widely used in various applications, including:

- **Image Classification:** Identifying objects or scenes in images.
- **Object Detection:** Locating and classifying multiple objects within an image.
- **Image Segmentation:** Dividing an image into meaningful segments for detailed analysis.
- **Facial Recognition:** Identifying and verifying individuals based on facial features.

Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a type of neural network designed for processing sequential data, such as time series or natural language. They have connections that loop back on themselves, allowing them to maintain a hidden state that captures information from previous time steps. This makes RNNs suitable for tasks where context and order matter.

Applications of RNNs

Applications of RNNs

RNNs are commonly used in applications involving sequential data, including:

- **Natural Language Processing (NLP)**: Tasks such as language modeling, text generation, and sentiment analysis.
- **Speech Recognition**: Converting spoken language into text.
- **Time Series Prediction**: Forecasting future values based on historical data.

Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks are a type of RNN designed to overcome the limitations of traditional RNNs, such as vanishing gradients. LSTMs use special gating mechanisms to control the flow of information, allowing them to capture long-term dependencies in sequential data. They are particularly effective for tasks that require remembering information over extended periods.

Applications of LSTMs

Applications of LSTMs

LSTMs are widely used in applications that involve long sequences of data, including:

- **Machine Translation:** Translating text from one language to another.
- **Speech Synthesis:** Generating human-like speech from text.
- **Video Analysis:** Analyzing sequences of frames in videos for tasks such as action recognition or event detection.

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are a class of neural networks used for generating new data samples that resemble a given training dataset. GANs consist of two components:

- **Generator:** A neural network that generates new data samples from random noise.
- **Discriminator:** A neural network that distinguishes between real data samples from the training set and fake samples generated by the generator.

The generator and discriminator are trained together in a competitive process, where the generator aims to produce realistic samples while the discriminator tries to correctly classify real and fake samples.

Applications of GANs

Applications of GANs

GANs have been successfully applied in various domains, including:

- **Image Generation**: Creating realistic images from random noise or low-resolution images.
- **Image-to-Image Translation**: Converting images from one domain to another, such as turning sketches into realistic images.
- **Data Augmentation**: Generating additional training samples to improve model performance, especially in scenarios with limited data.

Autoencoders

Autoencoders

Autoencoders are a type of neural network used for unsupervised learning tasks, such as dimensionality reduction and feature extraction. They consist of two main components:

- **Encoder:** A neural network that compresses the input data into a lower-dimensional representation (latent space).
- **Decoder:** A neural network that reconstructs the original input data from the compressed representation.

The goal of an autoencoder is to learn a compact representation of the input data while minimizing the reconstruction error between the original input and the reconstructed output.

Applications of Autoencoders

Applications of Autoencoders

Autoencoders are used in various applications, including:

- **Dimensionality Reduction**: Reducing the number of features in high-dimensional data while preserving important information.
- **Anomaly Detection**: Identifying unusual patterns in data by comparing the reconstruction error of input samples.
- **Data Denoising**: Removing noise from data by training the autoencoder to reconstruct clean data from noisy inputs.

Large Language Models (LLMs)

Large Language Models (LLMs)

Large Language Models (LLMs) are advanced neural networks designed to understand and generate human language. They are trained on vast amounts of text data and can perform a wide range of natural language processing tasks, such as text generation, translation, summarization, and question answering. LLMs leverage deep learning techniques, including transformers, to capture complex language patterns and relationships.

Applications of LLMs

Applications of LLMs

LLMs have numerous applications in various domains, including:

- **Chatbots and Virtual Assistants:** Providing conversational interfaces for customer support and information retrieval.
- **Content Generation:** Automatically generating articles, stories, or other written content based on prompts.
- **Sentiment Analysis:** Analyzing text to determine the sentiment or emotion expressed.
- **Language Translation:** Translating text from one language to another with high accuracy.