In-Depth View of NNs with Worked Examples

Johar M. Ashfaque

March 27, 2025

1 Introduction

A **Neural Network** (NN) is a computational model inspired by the way biological neural networks in the human brain process information. Neural networks are a fundamental part of the field of machine learning and artificial intelligence (AI). They are used to identify patterns, classify data, and make predictions by processing information through layers of interconnected nodes or "neurons."

In this document, we will cover the following:

- The structure of a neural network
- The different types of neural networks
- How neural networks work with worked examples
- Applications of neural networks

2 Structure of a Neural Network

A neural network consists of three main components:

1. **Input Layer**: This layer takes in the input features or data and feeds them to the next layer.
2. **Hidden Layer(s)**: These layers process the input using weights, biases, and activation functions. Neural networks can have one or more hidden layers, and the complexity increases with more layers.
3. **Output Layer**: The output layer provides the final output, which could be a class label, a probability, or a continuous value, depending on the task.

Each layer consists of neurons that receive signals, process them, and pass the results to the next layer. Each neuron performs a weighted sum of its inputs, applies a bias term, and uses an activation function to produce its output.

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

Where:

- w_i are the weights
- x_i are the inputs
- b is the bias
- f is the activation function
- y is the output of the neuron

3 Worked Example: Feedforward Neural Network

Consider a simple feedforward neural network with one hidden layer. Suppose we are trying to predict whether a person will buy a product based on their age (x_1) and income (x_2).

Given the following parameters: - Input: $x_1 = 25$ (age), $x_2 = 50000$ (income) - Weights: $w_1 = 0.5$, $w_2 = -0.2$ (for the input layer to hidden layer) - Bias: $b = 1$ - Activation function: Sigmoid, $f(x) = \frac{1}{1+e^{-x}}$

The input to the neuron in the hidden layer is the weighted sum of inputs plus the bias:

$$z = w_1 x_1 + w_2 x_2 + b$$

Substituting the values:

$$z = (0.5 \times 25) + (-0.2 \times 50000) + 1 = 12.5 - 10000 + 1 = -9986.5$$

Now, apply the sigmoid activation function:

$$y = f(z) = \frac{1}{1 + e^{-(-9986.5)}} \approx 0$$

Since the sigmoid function outputs values between 0 and 1, the result is very close to 0, meaning the person is unlikely to buy the product.

4 Types of Neural Networks

There are several types of neural networks, each suited for different tasks. Below, we discuss the most commonly used types.

4.1 Feedforward Neural Networks (FNN)

Feedforward Neural Networks (FNN) are the simplest type of neural network. In a feedforward network, the data moves in one direction: from the input layer to the output layer through the hidden layers. There are no cycles or loops in the network.

Example: A simple image classification problem where the input is a vectorized representation of an image, and the output is a predicted label.

$$y = f(W \cdot x + b)$$

Where W is the weight matrix, x is the input vector, and b is the bias term.

4.2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) are specialized for processing grid-like data, such as images. CNNs consist of convolutional layers, pooling layers, and fully connected layers.

Convolution Layer: The convolution operation involves sliding a filter or kernel over the input data and computing a dot product to produce feature maps.

Example: Image Classification using CNN

Consider a simple CNN designed to classify images of handwritten digits (MNIST dataset). The input is a 28x28 pixel grayscale image, and the task is to classify the image into one of 10 classes (digits 0-9).

- The first convolution layer applies a 3x3 kernel to the image. The kernel performs a dot product with the 3x3 patch of the image and produces a feature map. - After applying the convolution operation, a max-pooling layer reduces the spatial size of the feature map. - The output from the pooling layer is flattened into a vector and passed through fully connected layers for classification.

$$y = \text{Softmax}(W \cdot \text{Flatten}(x) + b)$$

Where the *Softmax* function converts the output into probabilities, and *Flatten* converts the feature map into a 1D vector.

4.3 Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNN) are used to model sequential data, such as time series, speech, and natural language. Unlike feedforward networks, RNNs have loops that allow information to be passed from one step of the sequence to the next.

Example: Sentiment analysis of a sentence, where each word influences the prediction of the next word or sentiment.

$$h_t = f(W \cdot x_t + U \cdot h_{t-1} + b)$$

Where:

- h_t is the hidden state at time step t
- x_t is the input at time step t
- h_{t-1} is the hidden state at the previous time step
- W, U, and b are learnable parameters

4.4 Long Short-Term Memory (LSTM) Networks

LSTM networks are a type of RNN that is specifically designed to address the vanishing gradient problem. LSTMs are used for tasks that require learning long-term dependencies, such as speech recognition or machine translation.

LSTMs use three gates: input, forget, and output gates, to control the flow of information.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

Where f_t, i_t, and o_t represent the forget, input, and output gates, respectively.

4.5 Generative Adversarial Networks (GAN)

Generative Adversarial Networks (GANs) consist of two neural networks: a generator and a discriminator. The generator creates fake data, and the discriminator tries to distinguish between real and fake data. The two networks are trained together in a competitive setting, leading to the generation of realistic data.

Example: Generating realistic images of people that do not exist.

$$\text{Generator} : z \to G(z)$$
$$\text{Discriminator} : x \to D(x)$$

Where z is a random noise vector, $G(z)$ is the generated image, and $D(x)$ is the probability that x is a real image.

5 How Neural Networks Work

Neural networks work through a process known as *forward propagation*, where data is passed through the layers to make predictions, and *backpropagation*, where errors are calculated and used to update the network's parameters.

5.1 Forward Propagation

During forward propagation, the input data is passed through each layer. In each layer, the inputs are multiplied by the weights, summed, and passed through an activation function. The output of one layer becomes the input to the next layer until the final output is produced.

5.2 Backpropagation

Backpropagation is the process of adjusting the weights of the network to minimize the error. The error is computed by comparing the predicted output to the actual output using a loss function (e.g., Mean Squared Error or Cross-Entropy Loss). The gradients of the loss function with respect to the weights are calculated using the chain rule, and the weights are updated using an optimization algorithm, such as Gradient Descent.

$$\Delta w = -\eta \frac{\partial L}{\partial w}$$

Where η is the learning rate, L is the loss, and $\frac{\partial L}{\partial w}$ is the gradient of the loss with respect to the weight.

6 Examples of Neural Network Applications

Neural networks have a wide range of applications in various domains. Here are some examples:

- **Image Classification**: Identifying objects or scenes in images (e.g., classifying handwritten digits in the MNIST dataset using CNNs).
- **Speech Recognition**: Converting speech to text or understanding spoken commands (e.g., using RNNs or LSTMs).
- **Natural Language Processing (NLP)**: Tasks such as sentiment analysis, machine translation, and question answering (e.g., using RNNs or Transformer-based networks).
- **Game Playing**: Training AI agents to play games such as Chess or Go (e.g., using Deep Q-Networks or reinforcement learning).

7 Conclusion

Neural networks are a powerful tool in the field of artificial intelligence and machine learning. They are versatile and can be applied to a variety of tasks, from image recognition to natural language processing. The development of different types of neural networks, such as CNNs, RNNs, and GANs, has enabled the solving of complex problems that were once thought to be unsolvable.

Understanding the underlying architecture and functioning of neural networks is crucial for leveraging their potential in real-world applications.