Introduction to Artificial Neural Networks

BEGINNER CLASSIFICATION DEEP LEARNING PYTHON PYTHON STRUCTURED DATA SUPERVISED

Researchers use **Artificial Neural Networks** (ANN) algorithms based on brain function to model complicated patterns and forecast issues. The Artificial Neural Network (ANN) is a deep learning method that arose from the concept of the human brain Biological Neural Networks. They are among the most powerful **machine learning algorithms** used today. The development of ANN was the result of an attempt to replicate the workings of the human brain. The workings of ANN are extremely similar to those of biological neural networks, although they are not identical. ANN algorithm accepts only numeric and structured data.

This article explores Artificial Neural Networks (ANN) in machine learning, focusing on how CNNs and RNNs process unstructured data like images, text, and speech. You'll learn about neural networks in AI, their types, and their role in machine learning.

Also, you will discover the fundamentals of artificial neural networks (ANN) in machine learning. We'll explore what an artificial neural network is, delve into neural network architecture, and discuss the ANN algorithm. Additionally, we'll highlight various applications of artificial neural networks and provide an introduction to neural networks in artificial intelligence.

Learning Objectives:

- Understand the concept and architecture of Artificial Neural Networks (ANNs)
- Learn about the different types of ANNs like feedforward, convolutional, recurrent, etc.
- Gain hands-on experience in building a simple ANN model using Python and Keras

This article was published as a part of the Data Science Blogathon.

Table of contents

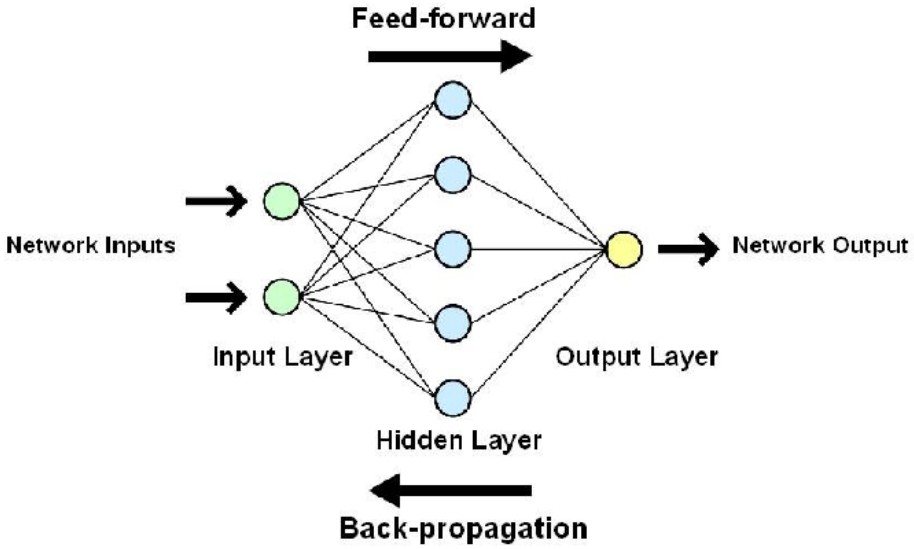
What is Artificial Neural Network(ANN)?

Artificial neural **networks** (ANNs) are created to replicate how the human brain processes data in computer systems. Neurons within interconnected units collaborate to identify patterns, acquire knowledge from data, and generate predictions. Artificial neural networks (ANNs) are commonly employed in activities such as identifying images, processing language, and making decisions.

Like human brains, artificial neural networks are made up of neurons that are connected like brain cells. These neurons process and receive information from nearby neurons before sending it to other neurons.

Artificial Neural Networks Architecture

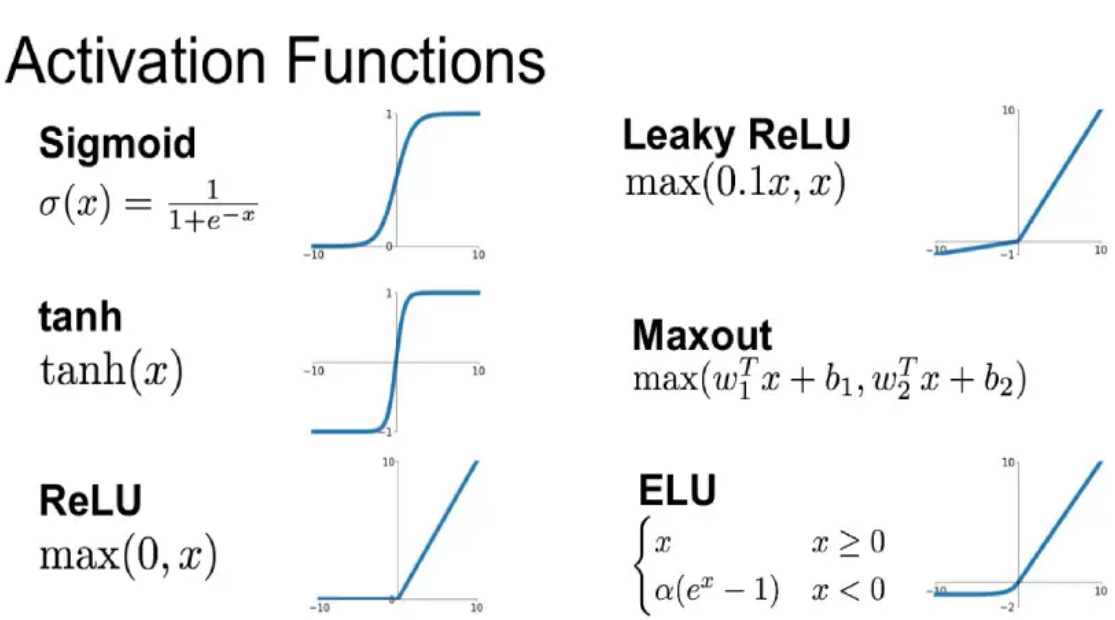
- There are three layers in the network architecture: the input layer, the hidden layer (more than one), and the output layer. A typical **feedforward network** processes information in one direction, from input to output. Because of the numerous layers are sometimes referred to as the MLP (Multi-Layer Perceptron).



- It is possible to think of the hidden layer as a “distillation layer,” which extracts some of the most relevant patterns from the inputs and sends them on to the next layer for further analysis. It accelerates and improves the efficiency of the network by recognizing just the most important information from the inputs and discarding the redundant information.

The activation function is important for two reasons: first, it allows you to turn on your computer. It contributes to the conversion of the input into a more usable **final output**.

- This model captures the presence of nonlinear relationships between the inputs.
- It contributes to the conversion of the input into a more usable output.



- Finding the “optimal values of W — weights” that minimize prediction error is critical to building a successful model. The “backpropagation algorithm” is a method by which **neural networks work**, converting ANN into a learning algorithm by learning from mistakes.

- The optimization approach uses a “gradient descent” technique to quantify prediction errors. This technique is a cornerstone of **supervised learning**, as it iteratively adjusts weights to minimize errors. In order to find the optimum value for W, we try small adjustments in W and examine the impact on prediction errors. Ultimately, we choose those W values as ideal because further changes in W do not reduce mistakes.

Benefits of Artificial Neural Networks

ANNs offers many key **benefits** that make them particularly well-suited to specific issues and situations:

- ANNs can learn and model non-linear and complicated interactions, which is critical since many of the relationships between inputs and outputs in real life are non-linear and complex.
- Artificial Neural Network in machine learning can generalize – After learning from the original inputs and their associations, the model may infer unknown relationships from anonymous data, allowing it to generalize and predict unknown data.
- ANN does not impose any constraints on the input variables, unlike many other prediction approaches (like how they should be distributed). Furthermore, numerous studies have demonstrated that ANN algorithms can better simulate heteroskedasticity, or data with high volatility and non-constant variance, because of their capacity to discover latent correlations in the data without imposing any preset associations. This is particularly helpful in financial time series forecasting (for example, stock prices) when significant data volatility.

Types of Artificial Neural Networks

Five Types of Artifical Neural Networks:

- **Feedforward Neural Networks (FNNs)**: These are straightforward networks where information flows in one direction, like from the input to the output. They’re used for tasks like identifying patterns in data or making predictions, making them ideal for pattern recognition.

- **Convolutional Neural Networks (CNNs):** Think of these as networks designed specifically for understanding images. They're great at recognizing patterns in pictures, making them perfect for tasks like identifying objects in photos or videos.
- **Recurrent Neural Networks (RNNs)**: These networks are good with sequences, like predicting the next word in a sentence or understanding the context of words. They remember previous information, which helps them understand the current data better.
- **Long Short-Term Memory Networks (LSTMs)**: LSTMs are a type of RNN that are really good at remembering long sequences of data. They're often used in tasks where understanding context over time is important, like translating languages or analyzing time-series data.
- **Generative Adversarial Networks (GANs)**: These networks are like artists. One part of the network generates new data, like images or music, while the other part critiques it to make sure it looks or sounds realistic. GANs are a key technology in **generative AI**. GANs are used for creating new content, enhancing images, or even generating deepfakes.

How do Artificial Neural Networks Learn?

Here is the Steps to learn AI neural Network:

1. **Starting Point**: Imagine you're building a robot brain, but initially, it knows nothing. So, you randomly assign some strengths to the connections between its "neurons" (like how our brain's neurons are connected).
2. **Seeing Data**: Now, show the robot some examples of what you want it to learn. For instance, if you're teaching it to recognize cats, show it lots of pictures of cats.
3. **Guessing and Checking**: The robot tries to imagine what it's seeing based on the strengths of its connections. At first, it'll make lots of mistakes because it's just guessing randomly.
4. **Getting Feedback**: You tell the robot how wrong its guesses are. For example, you say, "No, that's not a cat; it's a dog." This helps the robot understand where it went wrong and adjust through feedback loops.
5. **Adjusting Strengths**: The robot tweaks the strengths of its connections based on the feedback. If it guessed wrong, it changes the connections to be a bit stronger or weaker so that next time it might make a better guess. This **learning process** helps the robot improve its accuracy over time.
6. **Practice Makes Perfect**: The robot keeps looking at more examples, guessing, getting feedback, and adjusting until it gets better and better at recognizing cats.
7. **Testing Skills**: Once the robot has seen lots of examples and adjusted its connections a lot, you give it a new picture it hasn't seen before to see if it can correctly identify whether it's a cat or not.

Also, Checkout this Article [Artifical Neural Networks with Implementation]

Application of Artificial Neural Networks

ANNs have a wide range of applications because of their unique properties. A few of the important applications of ANNs include:

1. Image Processing and Character recognition

ANN algorithms play a significant part in picture and character recognition because of their capacity to take in many inputs, process them, and infer hidden and complicated, non-linear correlations. Character

recognition, such as handwriting recognition, has many applications in fraud detection (for example, bank fraud) and even national security assessments.

Image recognition is a rapidly evolving discipline with several applications ranging from social media facial recognition to cancer detection in medicine to satellite image processing for agricultural and defense purposes.

Deep neural networks, which form the core of “deep learning,” have now opened up all of the new and transformative advances in computer science, speech recognition, and natural language processing – notable examples being self-driving vehicles, and other applications powered by **neural nets**.

2. Forecasting

Everyday company decisions (sales, the financial allocation between goods, and capacity utilization), economic and monetary policy, finance, and the stock market widely use it. Forecasting issues are frequently complex; for example, predicting stock prices is complicated with many underlying variables (some known, some unseen).

Traditional forecasting models have flaws when it comes to accounting for these complicated, non-linear interactions. Given its capacity to model and extract previously unknown characteristics and correlations, ANNs can provide a reliable alternative when used correctly even in **unsupervised learning** scenarios. ANN also has no restrictions on the input and residual distributions, unlike conventional models.So, this ai neural network applications.

Advantages of Artificial Neural Networks

- ANN **algorithms** use attribute-value pairs to represent problems.
- The output of ANN algorithms can be discrete-valued, real-valued, or a vector of multiple real or discrete-valued characteristics, while the target function can be discrete-valued, real-valued, or a vector of numerous real or discrete-valued attributes.
- Noise in the training data is not a problem for ANN learning techniques. There may be mistakes in the training samples, but they will not affect the final result.
- It’s utilized when a quick assessment of the taught target function is necessary.
- The number of weights in the network, the number of training instances evaluated, and the settings of different learning algorithm parameters can all contribute to extended training periods for ANNs.

Disadvantages of Artificial Neural Networks

1. Hardware Dependence

- The construction of Artificial Neural Networks necessitates the use of parallel processors.
- As a result, the equipment's realization is contingent.

2. Understanding the network's operation

- This is the most serious issue with ANN.
- When ANN provides a probing answer, it does not explain why or how it was chosen.
- As a result, the network's confidence is eroded.

3. Assured network structure:

- Any precise rule does not determine the structure of artificial neural network in machine learning
- Experience and trial and error are used to develop a suitable network structure.

4. Difficulty in presenting the issue to the network

- ANNs are capable of working with numerical data.
- Before being introduced to ANN, problems must be converted into numerical values.
- The display method that is chosen will have a direct impact on the network's performance.
- The user's skill is a factor here.

5. The network's lifetime is unknown

- When the network's error on the sample is decreased to a specific amount, the training is complete.
- The value does not produce the best outcomes.

Create a Simple ANN for the famous Titanic Dataset

Now that we have discussed the architecture, advantages, and disadvantages it's time to create an ANN model so that we would know how it works. This **tutorial** will guide you through creating an **ANN model** for the famous Titanic dataset.

For understanding ANN algorithms we would be using world-famous titanic survival prediction. you can find the dataset here https://www.kaggle.com/jamesleslie/titanic-neural-network-for-beginners/data?

select=train_clean.csv. This **classifier** will help us predict which passengers survived the disaster based on various features.

Let's start with importing the dependencies.

```
## import dependencies import numpy as np import pandas as pd import seaborn as sns import matplotlib.pyplot as plt from matplotlib.pyplot import rcParams %matplotlib inline rcParams['figure.figsize'] = 10,8 sns.set(style='whitegrid', palette='muted', rc={'figure.figsize': (15,10)}) import os from sklearn.preprocessing import StandardScaler from sklearn.model_selection import train_test_split from sklearn.model_selection import GridSearchCV from keras.wrappers.scikit_learn import KerasClassifier from keras.models import Sequential from keras.layers import Dense, Activation, Dropout from numpy.random import seed from tensorflow import set_random_seed
```

Once you have all the preprocessing and modeling libraries imported, we will read the training and testing data.

```
# Load data as Pandas dataframe train = pd.read_csv('./train_clean.csv', ) test = pd.read_csv('./test_clean.csv') df = pd.concat([train, test], axis=0, sort=True) df.head()
```

We have concatenated both training and testing CSV in order to apply the same preprocessing method on both of them. once created the dataset we would start preprocessing the dataset since it has multiple columns that are non-numbers. Starting with the column name 'sex' in the dataset, we would be converting it to binary variables.

```
# convert to cateogry dtype df['Sex'] = df['Sex'].astype('category') # convert to category codes df['Sex'] = df['Sex'].cat.codes
```

After this, we need to convert the rest of the variables:

```
# subset all categorical variables which need to be encoded categorical = ['Embarked', 'Title'] for var in categorical: df = pd.concat([df, pd.get_dummies(df[var], prefix=var)], axis=1) del df[var] # drop the variables we won't be using df.drop(['Cabin', 'Name', 'Ticket', 'PassengerId'], axis=1, inplace=True) df.head()
```

```
## scale continuous variable continuous = ['Age', 'Fare', 'Parch', 'Pclass', 'SibSp', 'Family_Size'] scaler = StandardScaler() for var in continuous: df[var] = df[var].astype('float64') df[var] = scaler.fit_transform(df[var].values.reshape(-1, 1))
```

Once preprocessing is done we need to split the train and test the dataset again, for that you can use the following code.

```
X_train = df[pd.notnull(df['Survived'])].drop(['Survived'], axis=1) y_train = df[pd.notnull(df['Survived'])]['Survived'] X_test = df[pd.isnull(df['Survived'])].drop(['Survived'], axis=1)
```

Now is the time to define the hyperparameters and define the architecture of the ANN model.

```
lyrs=[8] act='linear' opt='Adam' dr=0.0 # set random seed for reproducibility seed(42) set_random_seed(42) model = Sequential() # create first hidden layer model.add(Dense(lyrs[0], input_dim=X_train.shape[1], activation=act)) # create additional hidden layers for i in range(1,len(lyrs)): model.add(Dense(lyrs[i], activation=act)) # add dropout, default is none model.add(Dropout(dr)) # create output layer model.add(Dense(1, activation='sigmoid')) # output layer model.compile(loss='binary_crossentropy', optimizer=opt, metrics=['accuracy']) model = create_model() print(model.summary())
```

after model definition, we will fit the model on our training data and would get the model insight.

```
# train model on full train set, with 80/20 CV split training = model.fit(X_train, y_train, epochs=100, batch_size=32, validation_split=0.2, verbose=0) val_acc = np.mean(training.history['val_acc']) print("n%s: %.2f%%" % ('val_acc', val_acc*100)) # summarize history for accuracy plt.plot(training.history['acc']) plt.plot(training.history['val_acc']) plt.title('model accuracy') plt.ylabel('accuracy') plt.xlabel('epoch') plt.legend(['train', 'validation'], loc='upper left') plt.show()
```

Now you can use the model for predictions on test data, using the following code chunk:

```
# calculate predictions test['Survived'] = model.predict(X_test) test['Survived'] = test['Survived'].apply(lambda x: round(x,0)).astype('int') solution = test[['PassengerId', 'Survived']]


print(solution)
```

Conclusion

Artificial neural networks (ANNs) have many **applications** in various industries, including medical, security/finance, government, agricultural, and defense. Researchers have mentioned several noteworthy uses of ANNs, making them powerful models that can be applied in many scenarios in artificial intelligence. ANN algorithms are particularly effective in tasks such as image recognition, natural language processing, and predictive analytics. They have the ability to learn complex patterns and relationships from data, making them invaluable tools for solving a wide range of problems in different domains.

Hope you liked the article and now have a better understanding of the ANN full form in machine learning. The ANN full form, or artificial neural network, is a powerful tool in the world of AI. If you're wondering what is ANN in machine learning, it's a type of AI neural network that excels at pattern recognition and data analysis, enabling intelligent systems to learn and adapt from vast amounts of information."

Key Takeaways

- ANNs are computational models inspired by biological neural networks, capable of learning complex patterns from data
- They have diverse applications in areas like image recognition, natural language processing, and predictive analytics
- ANNs offer advantages like modeling non-linear relationships and generalizing to unseen data, but require significant computational resources

Frequently Asked Questions

Q1. What is the artificial neural network?

A. An artificial neural network (ANN) is a computing system inspired by the biological neural networks of animal brains, designed to recognize patterns and solve complex problems.

Q2. Where is ANN used?

A. ANNs are used in various fields such as image and speech recognition, medical diagnosis, financial forecasting, and autonomous driving, thanks to their ability to learn from data.

Q3. What is the difference between CNN and ANN?

A. While ANNs are general-purpose neural networks, Convolutional Neural Networks (CNNs) are specialized for processing grid-like data structures, particularly images, through convolutional layers.

Q4. What are the basics of ANN?

A. The basics of ANN include neurons (nodes), layers (input, hidden, output), weights, biases, activation functions, and the process of learning through backpropagation and optimization algorithms.

Article Url - https://www.analyticsvidhya.com/blog/2021/09/introduction-to-artificial-neural-networks/

Gourav

Applied Machine Learning Engineer skilled in Computer Vision/Deep Learning Pipeline Development, creating machine learning models, retraining systems, and transforming data science prototypes to production-grade solutions. Consistently optimizes and improves real-time systems by evaluating strategies and testing real-world scenarios.