
Course: Artificial Intelligence and Machine Learning Code: 20CS51I
WEEK- 7 Big Data**❖ What is Big Data?****❖ Vs of Big Data****❖ Sources of data****❖ Role of Big Data in AI&ML****❖ Python Packages for Machine Learning and Deep Learning**

✓ - **Scientifics computing libraries**

✓ - **Visualization Libraries**

✓ - **Algorithmic libraries**

❖ Environment setup: install required packages**Session No. 7****What is Big Data?**

Big Data is a collection of data that is huge in volume, yet growing exponentially with time. It is a data with so large size and complexity that none of traditional data management tools can store it or process it efficiently. Big data is also a data but with huge size.

The **New York Stock Exchange** is an example of Big Data that generates about *one terabyte* of new trade data per day.

Vs of Big Data

Big data is a collection of data from many different sources and is often describe by five characteristics: volume, value, variety, velocity, and veracity.

- **Volume:** the size and amounts of big data that companies manage and analyze
- **Value:** the most important “V” from the perspective of the business, the value of big data usually comes from insight discovery and pattern recognition that lead to more effective operations, stronger customer relationships and other clear and quantifiable business benefits
- **Variety:** the diversity and range of different data types, including unstructured data, semi-structured data and raw data
- **Velocity:** the speed at which companies receive, store and manage data – e.g., the specific number of social media posts or search queries received within a day, hour or other unit of time
- **Veracity:** the “truth” or accuracy of data and information assets, which often determines executive-level confidence

Sources of Big Data

A significant part of big data is generated from three primary resources:

- Machine data
- Social data, and
- Transactional data.

In addition to this, companies also generate data internally through direct customer engagement. This data is usually stored in the company's firewall. It is then imported externally into the management and analytics system.

Machine Data

Machine data is automatically generated, either as a response to a specific event or a fixed schedule. It means all the information is developed from multiple sources such as smart sensors, SIEM logs, medical devices and wearables, road cameras, IoT devices, satellites, desktops, mobile phones, industrial machinery, etc. These sources enable companies to track consumer behaviour. Data extracted from machine sources grow exponentially along with the changing external environment of the market.

Social Data

It is derived from social media platforms through tweets, retweets, likes, video uploads, and comments shared on Facebook, Instagram, Twitter, YouTube, Linked In etc. The extensive data generated through social media platforms and online channels offer qualitative and quantitative insights on each crucial facet of brand-customer interaction

Transactional Data

As the name suggests, transactional data is information gathered via online and offline transactions during different points of sale. The data includes vital details like transaction time, location, products purchased, product prices, payment methods, discounts/coupons used, and other relevant quantifiable information related to transactions.

The sources of transactional data include:

- Payment orders
- Invoices
- Storage records and
- E-receipts

Transactional data is a key source of business intelligence. The unique characteristic of transactional data is its time print. Since all transactional data include a time print, it is time-sensitive and highly volatile. In plain words, transactional data will lose its credibility and importance if not used in due time. Thus, companies using transactional data promptly can gain the upper hand in the market.

Some additional steps ideal to the analysis of big data are:

- ✓ Deep learning offshoot of data
- ✓ Data mining
- ✓ Streaming analytics
- ✓ Predictive modelling
- ✓ Statistical analysis
- ✓ Text mining

Role of Big Data in AI&ML

How can big data be used in AI and machine learning?

Machine learning systems use data-driven algorithms and statistical models to analyze and find patterns in data. This is different from traditional rules-based approaches that follow explicit instructions. Big data provides the raw material by which machine learning systems can derive insights

What is the difference between artificial intelligence and big data?

Big data refers to large volumes of diverse and dynamic data that can be mined for information. AI is a set of technologies that enables machines to simulate human intelligence. AI requires the volumes of big data to effectively learn and evolve. Big data relies on AI to more intelligently mine for information.

Why big data influence the rise artificial intelligence?

AI is developed by using machine learning which inputs data for the algorithm to learn responses. Big data is a valuable source of data for machine learning, but the extent of their potential contribution is uncertain, given that big data often involves unstructured free text

Deep learning

“Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction”.

The notion of deep learning is built on the core of classical machine learning approaches (like regression analysis, classification algorithms, the use of optimization algorithms, etc.) and proves worthy over them in the following situations:

- when enormous amounts of data are available
- when enormous computational power is available
- a non-interpretable model is not detrimental

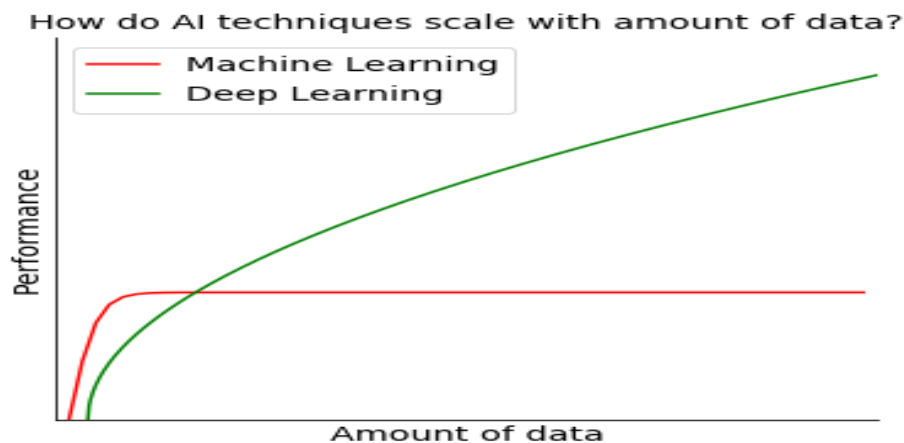
But how exactly does deep learning improve upon classical machine learning? To answer this question let us consider the cat scenario again.

If a classical machine learning algorithm has to identify a cat's face, then it would rely on feature engineering to build its model. The quality of features decides the accuracy of identification. On the contrary, a deep learning algorithm doesn't need feature engineering from the user end! Instead, it asks for a sequence of input images and creates features (whisker, eyes, nose, hair pattern, etc.) out of them, all by itself! In this sense, we can say that a deep learning algorithm learns just as a child does.

The higher the number of cats a child observes, the more accurate his/her ability to identify a cat becomes. Similarly, the more the number of training data samples a deep learning algorithm gets, the higher its accuracy becomes in identifying cats becomes.

Of course, similar to a child, a deep learning algorithm does perform mistakes while starting its learning journey but due to the availability of enormous data and computational resources in the present times it has proven to be far better than classical machine learning architectures in solving certain complicated problems (as seen before) and in some cases competitive to the human experts.

The following graph sums up the story. Higher the amount of data, higher is the deep learning architecture performance as compared to classical machine learning architectures.



Big data

Big data and artificial intelligence have a synergistic relationship. AI requires a massive scale of data to learn and improve decision-making processes and big data analytics leverages AI for better data analysis.

By bringing together big data and AI technology, companies can improve business performance and efficiency by:

- Anticipating and capitalizing on emerging industry and market trends.
- Analyzing consumer behavior and automating customer segmentation
- Personalizing and optimizing the performance of digital marketing campaigns
- Using intelligent decision support systems fueled by big data, AI, and predictive analytics

Python Packages for Machine Learning and Deep Learning

AI Big Data Analytics.

AI can assist users in all phases of the big data cycle, or the processes involved in the aggregation, storage, and retrieval of diverse types of data from various sources. These include data management, pattern management, context management, decision management, action management, goal management, and risk management.

AI can identify data types, find possible connections among datasets, and recognize knowledge using natural language processing. It can be used to automate and accelerate data preparation tasks, including the generation of data models, and assist in data exploration. It can learn common human error patterns, detecting and resolving potential flaws in information. And it can learn by watching how the user interacts with an analytics program, surfacing unexpected insights from massive datasets fast. AI can also learn subtle differences in meaning, or context-specific nuances, in order to help users better understand numeric data sources. And it can alert users to anomalies or unexpected patterns in data, actively monitoring events and identifying potential threats from system logs or social networking data, for example.

Machine Learning, as the name suggests, is the science of programming a computer by which they are able to learn from different kinds of data. A more general definition given by Arthur Samuel is – “Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed.” They are typically used to solve various types of life problems.

In the older days, people used to perform Machine Learning tasks by manually coding all the algorithms and mathematical and statistical formulas. This made the processing time-consuming, tedious, and inefficient. But in the modern days, it is become very much easy and more efficient compared to the olden days with various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reasons is its vast collection of libraries. Python libraries that are used in Machine Learning are:

- Numpy
- Scipy
- Scikit-learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Pandas
- Matplotlib

Numpy

NumPy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical functions. It is very useful for fundamental scientific computations in Machine Learning. It is particularly useful for linear algebra, Fourier transform, and random number capabilities. High-end libraries like TensorFlow uses NumPy internally for manipulation of Tensors.

```
# for some basic mathematical
# operations

import numpy as np

# Creating two arrays of rank 2
x = np.array([[1, 2], [3, 4]])
y = np.array([[5, 6], [7, 8]])

# Creating two arrays of rank 1
v = np.array([9, 10])
w = np.array([11, 12])

# Inner product of vectors
print(np.dot(v, w), "\n")

# Matrix and Vector product
print(np.dot(x, v), "\n")

# Matrix and matrix product
print(np.dot(x, y))
```

Output:

```
219

[29 67]

[[19 22]]
```

SciPy



SciPy is a very popular library among Machine Learning enthusiasts as it contains different modules for optimization, linear algebra, integration and statistics. There is a difference between the SciPy library and the SciPy stack. The SciPy is one of the core packages that make up the SciPy stack. SciPy is also very useful for image manipulation.

Scikit-learn

Scikit-learn is one of the most popular ML libraries for classical ML algorithms. It is built on top of two basic Python libraries, viz., NumPy and SciPy. Scikit-learn supports most of the supervised and unsupervised learning algorithms. Scikit-learn can also be used for data-mining and data-analysis, which makes it a great tool who is starting out with ML.

TensorFlow

It is a very popular open-source library for high performance numerical computation developed by the Google Brain team in Google. As the name suggests, Tensorflow is a framework that involves defining and running computations involving tensors. It can train and run deep neural networks that can be used to develop several AI applications. TensorFlow is widely used in the field of deep learning research and application.



```
# Python program using TensorFlow
# for multiplying two arrays

# import `tensorflow`
import tensorflow as tf

# Initialize two constants
x1 = tf.constant([1, 2, 3, 4])
x2 = tf.constant([5, 6, 7, 8])

# Multiply
result = tf.multiply(x1, x2)

# Initialize the Session
sess = tf.Session()

# Print the result
print(sess.run(result))

# Close the session
sess.close()
```

Output:

```
[ 5 12 21 32]
```

Keras

It provides many inbuilt methods for grouping, combining and filtering data.

Keras is a very popular Machine Learning library for Python. It is a high-level neural networks API capable of running on top of TensorFlow, CNTK, or Theano. It can run seamlessly on both CPU and GPU. Keras makes it really for ML beginners to build and design a Neural Network. One of the best thing about Keras is that it allows for easy and fast prototyping.

PyTorch

It is a popular open-source Machine Learning library for Python based on Torch, which is an open-source Machine Learning library that is implemented in C with a wrapper in Lua. It has an extensive choice of tools and libraries that support Computer Vision, Natural Language Processing(NLP), and many more ML programs. It allows developers to perform computations on Tensors with GPU acceleration and also helps in creating computational graphs.

```

import torch

dtype = torch.float
device = torch.device("cpu")
# device = torch.device("cuda:0") Uncomment this to run on GPU

# N is batch size; D_in is input dimension;
# H is hidden dimension; D_out is output dimension.
N, D_in, H, D_out = 64, 1000, 100, 10

# Create random input and output data
x = torch.randn(N, D_in, device=device, dtype=dtype)
y = torch.randn(N, D_out, device=device, dtype=dtype)

# Randomly initialize weights
w1 = torch.randn(D_in, H, device=device, dtype=dtype)
w2 = torch.randn(H, D_out, device=device, dtype=dtype)

learning_rate = 1e-6
for t in range(500):
    # Forward pass: compute predicted y
    h = x.mm(w1)
    h_relu = h.clamp(min=0)
    y_pred = h_relu.mm(w2)

    # Compute and print loss
    loss = (y_pred - y).pow(2).sum().item()
    print(t, loss)

    # Backprop to compute gradients of w1 and w2 with respect to loss
    grad_y_pred = 2.0 * (y_pred - y)
    grad_w2 = h_relu.t().mm(grad_y_pred)
    grad_h_relu = grad_y_pred.mm(w2.t())
    grad_h = grad_h_relu.clone()
    grad_h[h < 0] = 0
    grad_w1 = x.t().mm(grad_h)

    # Update weights using gradient descent
    w1 -= learning_rate * grad_w1
    w2 -= learning_rate * grad_w2

```

Output:

```

0 47168344.0
1 46385584.0
2 43153576.0
...
...
...
497 3.987660602433607e-05
498 3.945609932998195e-05
499 3.897604619851336e-05

```

Pandas

Pandas is a popular Python library for data analysis. It is not directly related to Machine Learning. As we know that the dataset must be prepared before training. In this case, Pandas comes handy as it was developed specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for grouping, combining and filtering data.

```

# Python program using Pandas for
# arranging a given set of data
# into a table

# importing pandas as pd
import pandas as pd

data = {"country": ["Brazil", "Russia", "India", "China", "South Africa"],
        "capital": ["Brasilia", "Moscow", "New Delhi", "Beijing", "Pretoria"],
        "area": [8.516, 17.10, 3.286, 9.597, 1.221],
        "population": [200.4, 143.5, 1252, 1357, 52.98] }

data_table = pd.DataFrame(data)
print(data_table)

```


Output:

| | country | capital | area | population |
|---|--------------|-----------|--------|------------|
| 0 | Brazil | Brasilia | 8.516 | 200.40 |
| 1 | Russia | Moscow | 17.100 | 143.50 |
| 2 | India | New Dehli | 3.286 | 1252.00 |
| 3 | China | Beijing | 9.597 | 1357.00 |
| 4 | South Africa | Pretoria | 1.221 | 52.98 |

Matplotlib

It is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots.

A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histogram, error charts, bar charts, etc,

Python3

```
# Python program using Matplotlib
# for forming a linear plot

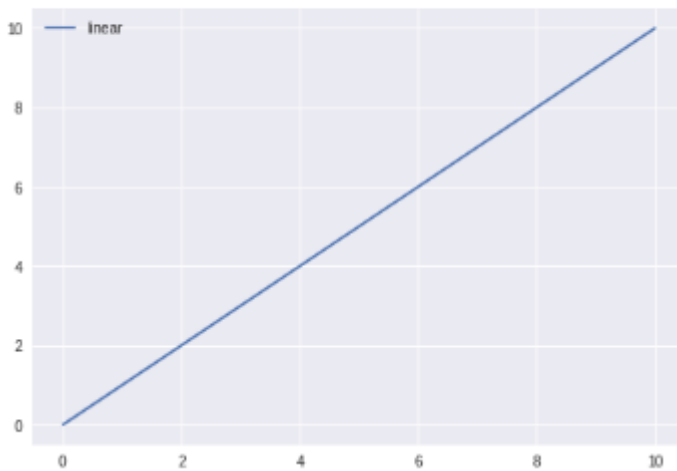
# importing the necessary packages and modules
import matplotlib.pyplot as plt
import numpy as np

# Prepare the data
x = np.linspace(0, 10, 100)

# Plot the data
plt.plot(x, x, label='linear')

# Add a legend
plt.legend()

# Show the plot
plt.show()
```

Output:

Visualization Library (VL) is an open source C++ middleware for 2D/3D graphics applications based on [OpenGL](#) 4, designed to develop portable applications for the [Microsoft Windows](#), [Linux](#) and [Mac OS X](#) operating systems.