

১. টেন্সর কী এবং কেন গুরুত্বপূর্ণ?

- গুরুত্ব: মেশিন লার্নিং (ML) কীভাবে কাজ করে (the 'How' part), সেই ব্যবহারিক দিকটিতে ফোকাস করার জন্য টেন্সর বোঝা অপরিহার্য।
- মৌলিক সংজ্ঞা: টেন্সর আর কিছুই নয়, কেবল একটি ডেটা স্ট্রাকচার। এটি ডেটা সংরক্ষণ করার একটি পদ্ধতি।
- **ML** লাইব্রেরিতে টেন্সরের ভূমিকা: বর্তমান সময়ে যত লিডিং মেশিন লার্নিং লাইব্রেরি রয়েছে, যেমন Scikit-learn, TensorFlow, তারা প্রত্যেকেই তাদের সবচেয়ে মৌলিক ডেটা স্ট্রাকচার হিসেবে টেন্সর ব্যবহার করে। গুগল-এর এক নম্বর ডিপ লার্নিং লাইব্রেরিটির নামও টেন্সরফ্লো (TensorFlow)।
- সফটওয়্যার কনসেপ্টের সাথে সম্পর্ক: কম্পিউটার সায়েন্স ব্যাকগ্রাউন্ড থেকে যারা এসেছেন, তাদের জন্য টেন্সর হলো **N-ডাইমেনশনাল** অ্যারে (**N-dimensional Array**)-এর সমার্থক। গণিত ও পদার্থবিদ্যা থেকে এই শব্দটি মেশিন লার্নিং কমিউনিটিতে গৃহীত হয়েছে।

২. টেন্সরের সংজ্ঞা ও পরিচিত রূপ

টেন্সরকে সাধারণত সংখ্যা সংরক্ষণের একটি ধারক (**Container for Numbers**) হিসেবে বর্ণনা করা হয়। যদিও খুব কম ক্ষেত্রেই এতে অক্ষর বা স্ট্রিং স্টোর করা যায়, তবে ৯৯.৯৯% ক্ষেত্রে এটি সংখ্যার জন্যই ব্যবহৃত হয়।

আপনি টেন্সর ইতোমধ্যেই ব্যবহার করেছেন, যদি আপনি কখনও ভেক্টর বা ম্যাট্রিক্স নিয়ে পড়াশোনা করে থাকেন; এগুলো আসলে টেন্সরই।

টেন্সরের প্রকারভেদ (মাত্রা বা **Dimension** অনুযায়ী):

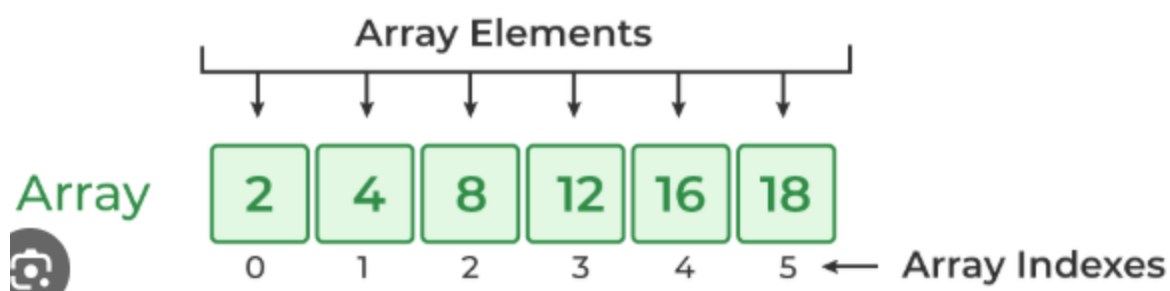
1. জিরো-ডি টেন্সর (**0D Tensor**) বা স্কেলার (**Scalar**):

- এটি হলো একটি সিঙ্গেল নাম্বার।
- এর মাত্রা (Dimension) শূন্য (0)।
- NumPy ব্যবহার করে যখন একটি স্কেলার তৈরি করা হয়, তখন তার মাত্রা পরীক্ষা করলে **0** পাওয়া যায় (`.ndim`)।

2. ওয়ান-ডি টেন্সর (**1D Tensor**) বা ভেক্টর (**Vector**):

- এটি হলো একাধিক সংখ্যার একটি তালিকা (List of Numbers)।
- এর মাত্রা এক (**1**)।
- গুরুত্বপূর্ণ পার্থক্য: যদিও এটি 1D টেন্সর, এর ভেতরে কতগুলি সংখ্যা রয়েছে তার ভিত্তিতে এটিকে সেই সংখ্যক মাত্রার ভেক্টর বলা হয়। উদাহরণস্বরূপ, যদি একটি 1D টেন্সরে চারটি সংখ্যা থাকে, তবে এটি একটি 4D ভেক্টর। এই দুটি ধারণা (টেনসর মাত্রা এবং ভেক্টর মাত্রা) গুলিয়ে ফেলা উচিত নয়।

One Dimensional Array in C



3. টু-ডি টেন্সর (2D Tensor) বা ম্যাট্রিক্স (Matrix):

- এটি হলো একাধিক ভেক্টর (1D টেন্সর)-এর একটি সংগ্রহ বা কলাম্বদ্ধতা।
- এতে দুটি অ্যাক্সেস বা অক্ষ থাকে (একটি রো অ্যাক্সেস এবং একটি কলাম অ্যাক্সেস)।

Introduction to Matrix

Data Structure

		Columns →		
Rows ↓		0	1	2
	0	10	8	3
	1	5	20	6
	2	7	9	30

4. হাই-ডি টেন্সর (Higher Dimensional Tensors):

- টেন্সরের মাত্রা সবসময় আগের মাত্রার টেন্সরকে যোগ করে বা সংগ্রহ করে বা কলাম্বদ্ধ করে বৃদ্ধি পায়।
- যেমন, একটি 3D টেন্সর হলো একাধিক ম্যাট্রিক্স (2D টেন্সর)-এর সংগ্রহ।
- একটি 4D টেন্সর হলো একাধিক 3D টেন্সর-এর সংগ্রহ।
- একটি 5D টেন্সর হলো 4D টেন্সরগুলির একটি ম্যাট্রিক্স।

৩. টেন্সর সম্পর্কিত মূল ধারণাগুলি

টেন্সর নিয়ে কাজ করার সময় কিছু পরিভাষা বোঝা আবশ্যিক:

- **র্যাঙ্ক (Rank), অ্যাক্সেস (Axis), ও ডাইমেনশন (Dimension):**
 - র্যাঙ্ক হলো টেন্সরের মাত্রার সংখ্যা।
 - অ্যাক্সেস হলো টেন্সরের মধ্যে কতগুলি দিক রয়েছে।
 - এই তিনটি পদ প্রায়শই একই অর্থে ব্যবহৃত হয়: নম্বর অফ অ্যাক্সেস = র্যাঙ্ক = নম্বর অফ ডাইমেনশন।
- **শেপ (Shape):**
 - শেপ বোঝায়, একটি নির্দিষ্ট অ্যাক্সেসে বা অক্ষে কতগুলি আইটেম বা উপাদান রয়েছে।
 - উদাহরণস্বরূপ, একটি ম্যাট্রিক্সের শেপ যদি (2, 3) হয়, তবে রো অ্যাক্সেসে দুটি আইটেম এবং কলাম অ্যাক্সেসে তিনটি আইটেম রয়েছে।
- **সাইজ (Size):**
 - সাইজ হলো টেন্সরের মধ্যে মোট আইটেমের সংখ্যা।
 - এটি শেপের সবগুলি সংখ্যাকে গুণ করে পাওয়া যায়।
 - একটি স্কেলারের সাইজ সবসময় 1 হয়।

8. ব্যবহারিক মেশিন লার্নিংয়ে টেন্সরের উদাহরণ (Practical Examples)

বাস্তব জগতে মেশিন লার্নিং বা ডিপ লার্নিং করার সময় কোন মাত্রার টেন্সরগুলি কীভাবে দেখা যায়, তার উদাহরণগুলি নিচে দেওয়া হলো:

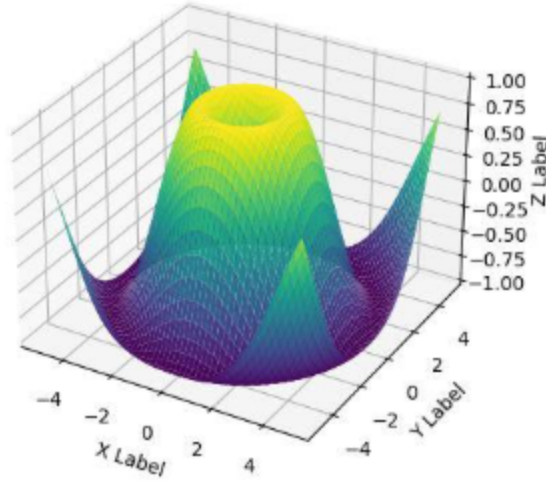
ক. 1D এবং 2D টেন্সর (টেবুলার ডেটা)

যখন আপনি কোনো টেবুলার ডেটা নিয়ে কাজ করেন (যেমন ছাত্র-ছাত্রীদের রেজাল্ট ডেটা), তখন:

- **1D টেন্সর (ভেক্টর):** ডেটাসেটের প্রতিটি রো (**Row**), যা একজন একক শিক্ষার্থীর ডেটা (যেমন CGPA, IQ, State) উপস্থাপন করে, সেটি হলো একটি 1D টেন্সর। ইনপুট কলামগুলি যতগুলি হবে, সেটি তত ডাইমেনশনাল ভেক্টর হবে (যেমন ৩টি কলাম থাকলে 3D ভেক্টর, কিন্তু টেন্সর 1D)। টার্গেট কলাম বা আউটপুট (যেমন প্লসমেন্ট হয়েছে কিনা) সেটিও একটি 1D টেন্সর।
- **2D টেন্সর (ম্যাট্রিক্স):** ইনপুট ডেটার সম্পূর্ণ সংগ্রহ (যেমন ১০,০০০ শিক্ষার্থীর ডেটা) হলো একটি ম্যাট্রিক্স বা 2D টেন্সর।

খ. 3D টেন্সর (NLP এবং টাইম সিরিজ ডেটা)

- **ন্যাচারাল ল্যাঙ্গুয়েজ প্রসেসিং (NLP):**
 - টেক্সট ডেটাকে ML-এর জন্য সংখ্যায় রূপান্তর করতে হয়।
 - একটি শব্দকে একটি ভেক্টর (1D টেন্সর) দ্বারা উপস্থাপন করা হয় (যেমন: One-hot Encoding)।
 - একটি বাক্য হলো সেই শব্দ-ভেক্টরগুলির একটি সংগ্রহ, যা একটি 2D টেন্সর তৈরি করে।
 - একাধিক বাক্যের সংগ্রহ (যেমন, তিনটি বাক্য) একটি 3D টেন্সর গঠন করে।

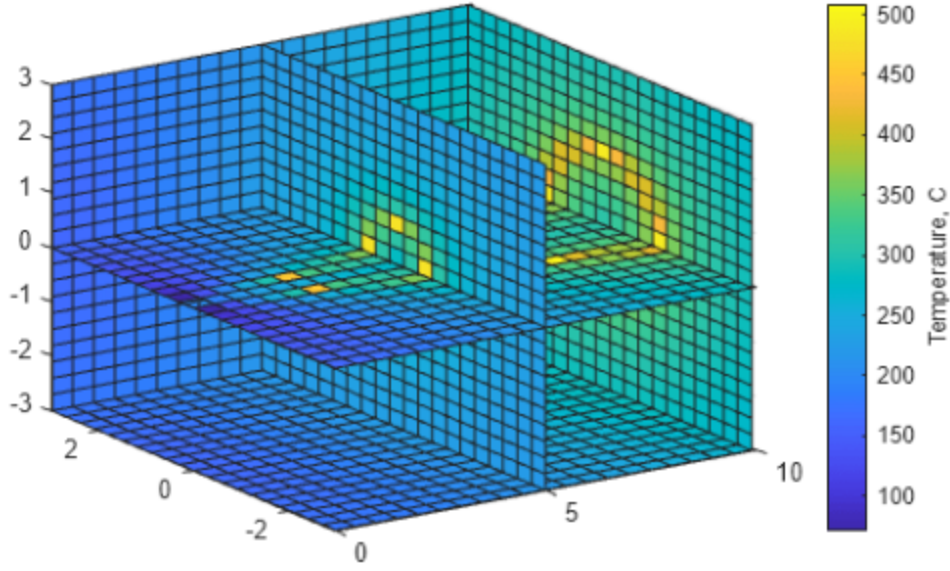


- টাইম সিরিজ ডেটা:
 - এক বছরের স্টক মার্কেটের দৈনিক হাই প্রাইস ও লো প্রাইসের ডেটা একটি 2D টেন্সর।
 - কিন্তু, যদি আপনি একই স্টকের ১০ বছরের ডেটা দেখেন, তবে এটি আসলে ১০টি 2D টেন্সরের সংগ্রহ, যা একটি **3D** টেন্সর তৈরি করে। এই ক্ষেত্রে মধ্যবর্তী অ্যাক্সেসটিকে টাইম অ্যাক্সেস বলা হয়।

গ. **4D** টেন্সর (ইমেজ ডেটা)

যখন কম্পিউটার ভিশন বা ইমেজ নিয়ে কাজ করা হয়:

- একটি কালার ইমেজ: প্রতিটি কালার ইমেজ আসলে তিনটি চ্যানেল (Red, Green, Blue বা RGB)-এর সমন্বয়ে তৈরি। প্রতিটি চ্যানেল নিজেই একটি 2D ম্যাট্রিক্স, তাই একটি কালার ইমেজ হলো একটি **3D** টেন্সর।
- একাধিক কালার ইমেজের সংগ্রহ: ডিপ লার্নিংয়ে যখন ৫০টি ইমেজ নিয়ে কাজ করা হয়, তখন তা একটি **4D** টেন্সর তৈরি করে (ইমেজের সংখ্যা x চ্যানেল x উচ্চতা x প্রস্থ)।



ঘ. 5D টেম্পর (ভিডিও ডেটা)

- ভিডিওর সংজ্ঞা: ভিডিও হলো আসলে খুব দ্রুত চোখের সামনে দিয়ে চলে যাওয়া ফ্রেম বা ইমেজগুলির সংগ্রহ।
- **5D** টেম্পরের গঠন:
 - একটি সিঙ্গেল ফ্রেম হলো একটি 3D টেম্পর।
 - একটি ভিডিও হলো সময়ের সাথে ফ্রেমগুলির সংগ্রহ, যা একটি **4D** টেম্পর তৈরি করে।
 - একাধিক ভিডিওর সংগ্রহ (যেমন চারটি ভিডিও) একটি **5D** টেম্পর তৈরি করে। এই টেম্পরের প্রথম উপাদানটি হলো ভিডিওর সংখ্যা।

পরামর্শ যদিও সাধারণত 0D থেকে 5D টেম্পর নিয়েই কাজ হয়, তবে ইমেজ বা ভিডিও ডেটা প্রক্রিয়াকরণের জন্য অনেক বেশি স্টোরেজ প্রয়োজন (যেমন চারটি ৬০ সেকেন্ডের ভিডিওর জন্য প্রায় ২৭ GB স্পেস)। এই কারণেই ভিডিওগুলি স্টোরেজ অপটিমাইজ করার জন্য MP4 বা MKV-এর মতো ভিডিও এনকোডিং ফরম্যাটে সংরক্ষণ করা হয়।

সব মিলিয়ে টেম্পর হলো ডেটা উপস্থাপনের একটি সাধারণ ধারণা, যা মেশিন লার্নিং অ্যালগরিদমগুলিকে সংখ্যা নিয়ে কার্যকরভাবে কাজ করতে সাহায্য করে। এই ধারণাটি হলো অনেকটা একটি জেনেরিক কনসেপ্টের মতো, যা স্কেলার, ভেক্টর বা ম্যাট্রিক্সের মতো নির্দিষ্ট ধারণাগুলির সীমাবদ্ধতা থেকে মুক্ত করে।