Word2vec: সহজ ভাষায় সম্পূর্ণ ধারণা

এই আলোচনায় ন্যাচারাল ল্যাঙ্গুয়েজ প্রসেসিং (NLP)-এর একটি অত্যন্ত গুরুত্বপূর্ণ ও বহুল ব্যবহৃত টেকনিক **Word2vec** নিয়ে বিস্তারিত ব্যাখ্যা করা হয়েছে। Word2vec মূলত এমন একটি পদ্ধতি যা কম্পিউটারকে মানুষের ভাষার অর্থ বুঝতে সাহায্য করে।

১. ওয়ার্ড এমবেডিং **(Word Embedding)** কী?

কম্পিউটার সরাসরি মানুষের লেখা শব্দ বা বাক্য বুঝতে পারে না। তাই শব্দকে প্রথমে সংখ্যায় রূপান্তর করতে হয়। এই সংখ্যায় রূপান্তর করার প্রক্রিয়াকেই বলা হয় ওয়ার্ড এমবেডিং।

সহজভাবে বললে,
ওয়ার্ড এমবেডিং = শব্দ → সংখ্যার ভেক্টর

আগের পদ্ধতিগুলো যেমন:

- Bag of Words
- TF-IDF

এইগুলো শুধু গুনে দেখত কোন শব্দ কয়বার এসেছে। কিন্তু তারা বুঝতে পারত না:

- কোন শব্দের অর্থ কাছাকাছি
- কোন শব্দ বিপরীত অর্থ বহন করে
- কোন শব্দ কোন প্রেক্ষাপটে ব্যবহৃত হয়েছে

কিন্তু Word2vec এখানে আলাদা। এটি শব্দের অর্থগত সম্পর্ক ধরতে পারে। যেমন:

- “Happy” এবং “Joy” কাছাকাছি অর্থের শব্দ
- “King” এবং “Queen” এর মধ্যে সম্পর্ক আছে

এই ধরনের সম্পর্ক Word2vec ভেক্টরের মাধ্যমে ধরে রাখতে পারে।

২. **Word2vec** কেন এত জনপ্রিয়**? (**বৈশিষ্ট্য ও সুবিধা**)**

ক**)** ডেন্স ভেক্টর ব্যবহার

Word2vec খুব বড় ও ফাঁকা ভেক্টরের বদলে ছোট কিন্তু তথ্যসমৃদ্ধ ভেক্টর তৈরি করে।
সাধারণত প্রতিটি শব্দকে ১০০ থেকে ৩০০ ডাইমেনশনের ভেক্টরে রূপান্তর করা হয়।

এর সুবিধা:

- কম মেমোরি লাগে
- দ্রুত ক্যালকুলেশন হয়
- বড় ডেটাসেটে ভালো কাজ করে

খ) ওভারফিটিং কম হয়

এখানে অপ্রয়োজনীয় শূন্য মান (zero value) খুব কম থাকে।
এর ফলে মডেল শুধু ট্রেনিং ডেটা মুখস্থ না করে জেনারাল প্যাটার্ন শিখতে পারে।

গ) ভেক্টর অ্যারিথমেটিক **(Vector Arithmetic)**

Word2vec এর সবচেয়ে মজার দিক হলো, এটি শব্দের মধ্যে গাণিতিক সম্পর্ক তৈরি করতে পারে।

উদাহরণ:

- King − Man + Woman ≈ Queen

এর মানে:
মডেল বুঝতে পারে,

- King এবং Queen একই ধরনের
- Man ও Woman জেন্ডারের পার্থক্য বোঝায়

এই সম্পর্কগুলো সম্পূর্ণভাবে সংখ্যার ভেতর লুকানো থাকে।

৩. **Word2vec** কীভাবে কাজ করে? **(Intuition** বা মূল ধারণা**)**

Word2vec কাজ করে একটি খুব শক্তিশালী ধারণার উপর:

যেসব শব্দ একই ধরনের প্রেক্ষাপটে ব্যবহৃত হয়**,** তাদের অর্থও কাছাকাছি হয়।

উদাহরণ:

- “আমি ভাত খাই”
- “আমি রুটি খাই”

এখানে “ভাত” ও “রুটি” একই জায়গায় বসেছে, তাই তাদের অর্থ কাছাকাছি।

Word2vec একটি ছোট নিউরাল নেটওয়ার্ক ব্যবহার করে:

- নিজে নিজে শব্দের বৈশিষ্ট্য শিখে নেয়
- যেমন জেন্ডার, ক্ষমতা, সম্পর্ক, ভূমিকা ইত্যাদি

এই বৈশিষ্ট্যগুলো আমরা আলাদা করে বলে দিই না।
মডেল নিজেই শিখে নেয়।

৪. **Word2vec** এর দুটি প্রধান মডেল **(Architecture)**

Word2vec সাধারণত দুইভাবে তৈরি করা হয়।

ক) **CBOW (Continuous Bag of Words)**

CBOW পদ্ধতিতে:

- চারপাশের শব্দ দেওয়া হয়
- মাঝখানের শব্দ অনুমান করা হয়

উদাহরণ:
“আমি ___ ভাত খাই”

এখানে “আমি”, “ভাত”, “খাই” দেওয়া থাকলে
মডেল অনুমান করবে ফাঁকা জায়গায় কোন শব্দ বসবে।

CBOW এর বৈশিষ্ট্য:

- দ্রুত কাজ করে
- ছোট ডেটাসেটের জন্য ভালো
- সাধারণ ভাষার ক্ষেত্রে কার্যকর

খ) **Skip-gram**

Skip-gram পদ্ধতিতে:

- মাঝখানের একটি শব্দ দেওয়া হয়
- তার চারপাশের শব্দগুলো অনুমান করা হয়

উদাহরণ:
যদি “ভাত” দেওয়া হয়,
মডেল অনুমান করবে তার আশেপাশে “আমি”, “খাই” ইত্যাদি শব্দ আসতে পারে।

Skip-gram এর বৈশিষ্ট্য:

- বড় ডেটাসেটে ভালো ফল দেয়
- কম ব্যবহৃত শব্দ ভালোভাবে শিখতে পারে
- তুলনামূলক ধীর কিন্তু বেশি শক্তিশালী

৫. প্র্যাকটিক্যাল উদাহরণ: **Game of Thrones**

বাস্তব একটি উদাহরণ দেখানো হয়েছে যেখানে:

- Game of Thrones সিরিজের বইয়ের টেক্সট ব্যবহার করা হয়েছে
- Python এর Gensim লাইব্রেরি দিয়ে Word2vec মডেল ট্রেন করা হয়েছে

এই মডেল দিয়ে দেখা গেছে:

- কোন চরিত্র কার সাথে বেশি সম্পর্কিত
- কোন দুইটি চরিত্রের নাম অর্থগতভাবে কাছাকাছি
- জন স্নো কেন অন্য চরিত্রদের থেকে আলাদা অবস্থানে আছে

এছাড়াও:

- PCA বা t-SNE ব্যবহার করে
- ৩০০ ডাইমেনশনের ভেক্টরকে ২ডি বা ৩ডি গ্রাফে দেখানো হয়েছে
- এতে চোখে দেখা যায় কোন শব্দ বা চরিত্র কার কাছে অবস্থান করছে