

Self-Supervised Learning (SSL) কী?

Self-Supervised Learning (SSL) হলো Machine Learning-এর এমন একটি পদ্ধতি যেখানে ডেটার সাথে আগে থেকে কোনো লেবেল (**Label**) বা সঠিক উত্তর দেওয়া থাকে না, কিন্তু তারপরও মডেল নিজে নিজেই শিখতে পারে।

এখানে মূল আইডিয়া হলো:

- সাধারণ Supervised Learning-এ মানুষের দেওয়া **Label** লাগে (যেমন: এটা বিড়াল, এটা কুকুর)
- কিন্তু SSL-এ মডেল নিজেই ডেটা থেকে “প্রশ্ন” বানায় এবং নিজেই “উত্তর” বানায় তারপর সেই বানানো প্রশ্ন-উত্তর দিয়েই নিজের ট্রেনিং চালায়

এজন্য একে বলা হয় **Self-Supervised**

মানে: নিজেই নিজের শিক্ষক

SSL কেন দরকার?

বাস্তব জীবনে ডেটা অনেক থাকে, কিন্তু ডেটা Label করা খুব কঠিন, সময়সাপেক্ষ এবং ব্যয়বহুল।

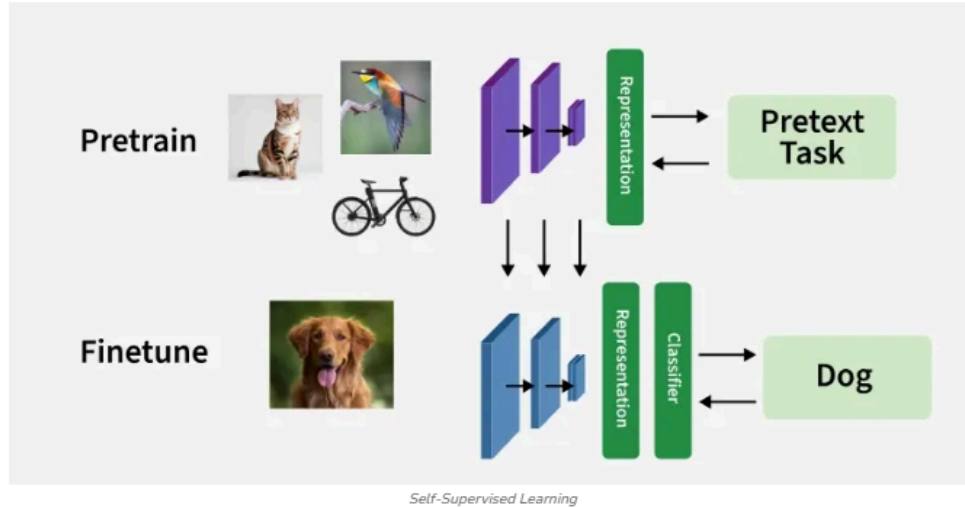
উদাহরণ:

- YouTube-এ কোটি কোটি ভিডিও আছে, কিন্তু প্রতিটা ভিডিও Label করা অসম্ভব
- ইন্টারনেটে কোটি কোটি লেখা আছে, কিন্তু প্রতিটা লাইনের Label বানানো কষ্টকর
- হাসপাতালের অনেক মেডিকেল ইমেজ আছে, কিন্তু ডাক্তার দিয়ে সব Label করানো খুব ব্যয়বহুল

তাই SSL খুব গুরুত্বপূর্ণ, কারণ:

অনেক **Unlabeled Data** থাকলেও মডেল শিখতে পারে

SSL কীভাবে কাজ করে?



উদাহরণ ১: বাক্যের কিছু শব্দ লুকিয়ে শেখা

ধরুন বাক্য:

"আমি আজ স্কুলে যাব"

SSL-এ মডেলকে দেওয়া হবে:

"আমি আজ ____ যাব"

এখন মডেলকে অনুমান করতে হবে:

শূন্যস্থানে কী হবে?

সঠিক উত্তর: "স্কুলে"

এখানে লক্ষ্য করুন:

- মানুষ Label দেয়নি
- মডেল নিজেই বাক্য থেকে শব্দ লুকিয়ে প্রশ্ন বানিয়েছে
- আবার নিজেই উত্তর জানে, কারণ মূল বাক্য তার কাছেই আছে

এভাবেই মডেল ভাষার গঠন শিখে ফেলে।

উদাহরণ ২: ছবির অংশ কেটে দিয়ে শেখা

একটা বিড়ালের ছবি থেকে যদি মডেলকে দেখানো হয়:

- ছবির অর্ধেক অংশ

এবং বলা হয়:

- বাকি অংশটা কেমন হতে পারে অনুমান কর

এভাবে মডেল:

- বিড়ালের গঠন
- চোখ-কান-লোমের প্যাটার্ন
- ছবির ভেতরের structure

এসব শিখে নেয়।

SSL-এর Key Characteristics (মূল বৈশিষ্ট্য) ডিটেইলসে

1) Uses Unlabeled Data (লেবেল ছাড়াই শেখে)

SSL-এর সবচেয়ে বড় বৈশিষ্ট্য:

এটা **raw** ডেটা (অপরিষ্কার/লেবেলবিহীন ডেটা) থেকেই শিখে

এটা খুব শক্তিশালী, কারণ:

- লেবেল বানাতে মানুষ লাগে না
- বিশাল ডেটা ব্যবহার করা যায়

2) Dynamic Label Generation (নিজেই Label তৈরি করে)

SSL-এর সবচেয়ে মজার বিষয়:

মডেল ডেটা থেকেই Label বানায়।

যেমন:

- বাক্য থেকে শব্দ লুকিয়ে, লুকানো শব্দটাই Label
- ছবি থেকে অংশ লুকিয়ে, লুকানো অংশটাই Label
- অডিও থেকে কিছু অংশ কেটে, সেই অংশটাই Label

এই Label মানুষ বানায় না, ডেটা থেকেই বের হয়।

3) Mix of Learning Methods (Supervised আর Unsupervised-এর মাঝামাঝি)

SSL আসলে পুরোপুরি Supervised না, আবার পুরোপুরি Unsupervised-ও না।

কারণ:

- Supervised Learning: মানুষের দেওয়া Label লাগে
- Unsupervised Learning: কোনো Label নেই, শুধু pattern বের করে
- SSL: Label নেই, কিন্তু মডেল নিজেই Label বানায়

তাই SSL হলো মাঝামাঝি একটা পদ্ধতি।

4) Learns Useful Features

Machine Learning-এ “Feature” মানে হলো:

ডেটার ভেতরের গুরুত্বপূর্ণ বৈশিষ্ট্য

যেমন:

- ছবিতে: edges, shapes, texture
- লেখায়: grammar, meaning, context
- অডিওতে: pitch, tone, phoneme

SSL-এর বড় সুবিধা:

মডেল যখন নিজে নিজে শেখে, তখন ডেটার গভীর pattern ধরতে পারে।
ফলে পরের কাজগুলোতে সে দ্রুত ভালো করতে পারে।

5) Wide Applications

SSL সবচেয়ে বেশি ব্যবহার হয়:

(ক) Image Recognition

যেমন:

- face detection
- object detection
- medical image analysis

(খ) Natural Language Processing (NLP)

যেমন:

- ChatGPT টাইপ মডেল
- translation
- summarization
- question answering

(গ) Speech Recognition

যেমন:

- voice to text
- speaker identification

কারণ এই ক্ষেত্রগুলোতে:

Label বানানো কঠিন এবং ব্যয়বহুল।

6) Helps Transfer Learning (Transfer Learning সহজ করে)

এটা SSL-এর সবচেয়ে বড় বাস্তব সুবিধা।

Transfer Learning মানে:

একটা কাজ শিখে অন্য কাজেও ব্যবহার করা।

যেমন:

- মডেলকে আগে কোটি কোটি unlabeled ছবি দিয়ে শেখানো হলো
- পরে তাকে ছোট dataset দিয়ে “cat vs dog” classification শেখানো হলো

এখন সে খুব দ্রুত শিখবে, কারণ:

সে আগেই শিখে ফেলেছে:

- কীভাবে ছবি বুঝতে হয়
- shape, texture, pattern কেমন হয়

এভাবে SSL pre-training করে পরে fine-tuning করা হয়।

SSL-এর সবচেয়ে বড় সুবিধাগুলো

SSL ব্যবহার করলে:

1. Label ছাড়াই বড় ডেটা থেকে শেখা যায়
2. কম Label দিয়েও ভালো performance পাওয়া যায়
3. Model generalization ভালো হয় (নতুন ডেটাতেও ভালো করে)
4. Transfer learning সহজ হয়
5. বাস্তব সমস্যা সমাধানে খরচ কমে যায়

Applications of SSL (SSL কোথায় কোথায় ব্যবহার হয়)

Self-Supervised Learning (SSL) সবচেয়ে বেশি কাজে লাগে সেই জায়গাগুলোতে যেখানে:

- ডেটা অনেক আছে
- কিন্তু লেবেল করা ডেটা কম
- বা লেবেল করা খুব খরচের/সময় লাগার কাজ

এখন এক এক করে দেখি।

1) Computer Vision (ছবি ও ভিডিও সম্পর্কিত কাজ)

Computer Vision মানে হলো:

কম্পিউটারকে ছবি বা ভিডিও “দেখ” বুঝতে শেখানো।

SSL এখানে অনেক কাজে লাগে, কারণ:

- ইন্টারনেটে কোটি কোটি ছবি আছে
- কিন্তু প্রতিটা ছবির label বানানো অসম্ভব

SSL দিয়ে মডেল unlabeled ছবি থেকে শিখে নেয়:

- আকৃতি (shape)
- প্রান্ত (edges)
- texture
- আলো-ছায়া
- বস্তু কোথায় আছে

এতে উপকার হয়:

- Image Recognition (ছবি চিনতে পারা)
- Video Recognition (ভিডিওতে কী হচ্ছে বোঝা)
- Object Detection (ছবিতে বস্তু কোথায় আছে বের করা)
- Medical Image Analysis (X-ray, MRI, CT scan থেকে রোগ ধরতে সাহায্য)

2) Natural Language Processing (NLP)

NLP মানে হলো:

কম্পিউটারকে মানুষের ভাষা বুঝতে শেখানো।

এখানে SSL খুব শক্তিশালী, কারণ:

- ইন্টারনেটে অনেক লেখা আছে (Wikipedia, news, books, blogs)
- কিন্তু এগুলোর label নেই

SSL দিয়ে মডেল শেখে:

- শব্দের অর্থ
- বাক্যের গঠন
- context (পরিপ্রেক্ষিত)
- semantics (গভীর অর্থ)

এভাবে BERT, GPT টাইপ বড় বড় ভাষা মডেল তৈরি হয়।

এতে সাহায্য হয়:

- Translation (অনুবাদ)
- Sentiment Analysis (লেখা positive না negative বোঝা)
- Text Classification (লেখা কোন ক্যাটাগরির)
- Question Answering
- Summarization

3) Speech Recognition (কথা থেকে লেখা বানানো)

Speech Recognition মানে:

মানুষের কথা শুনে সেটা লেখা বানানো বা বোঝা।

SSL এখানে কাজে লাগে কারণ:

- অনেক অডিও ডেটা আছে
- কিন্তু অডিওর সাথে সঠিক transcript (লিখিত কথা) বানানো খুব কঠিন

SSL দিয়ে মডেল unlabeled audio থেকে শিখে নেয়:

- শব্দের pattern
- pronunciation
- ভাষার rhythm
- phoneme structure

ফলে পরে কম labeled ডেটা দিয়েও ভালো কাজ করে।

4) Healthcare (স্বাস্থ্যখাত)

Healthcare-এ SSL খুব গুরুত্বপূর্ণ কারণ:

মেডিকেল ডেটা label করা মানে:

- ডাক্তার/বিশেষজ্ঞ দিয়ে annotate করানো
- যা খুব ব্যয়বহুল এবং সময়সাপেক্ষ

উদাহরণ:

- X-ray-এ কোনটা pneumonia
- MRI-তে tumor কোথায়
- CT scan-এ abnormality আছে কিনা

SSL দিয়ে মডেল অনেক unlabeled scan দেখে আগে শিখে নেয়, পরে অল্প label দিয়েই রোগ ধরতে সাহায্য করতে পারে।

5) Autonomous Systems and Robotics

এখানে পড়ে:

- Self-driving car
- Drone
- Robot navigation

এগুলোতে sensor data থাকে:

- camera
- lidar
- radar
- depth sensor

SSL মডেলকে raw sensor data থেকে শেখায়:

- রাস্তা কোথায়
- বাধা কোথায়
- মানুষ বা গাড়ি কোথায়
- সামনে কী হতে পারে

এর ফলে:

- navigation ভালো হয়
- perception উন্নত হয়
- decision making উন্নত হয়

Advantages of Self-Supervised Learning (SSL-এর সুবিধা)

এখন SSL কেন এত জনপ্রিয়, তার মূল কারণগুলো দেখি।

1) Less Dependence on Labeled Data (Label কম লাগবে)

SSL-এর সবচেয়ে বড় সুবিধা:

মানুষের হাতে label করানোর উপর নির্ভরতা কমে যায়।

ফলে:

- সময় বাঁচে
- খরচ কমে
- বড় ডেটা ব্যবহার করা যায়

2) Better Generalization (নতুন ডেটাতেও ভালো করে)

SSL মডেল ডেটার ভেতরের structure শিখে।

এ কারণে মডেল শুধু মুখস্থ করে না, বরং বোঝে।

ফলে:

- নতুন ছবি
- নতুন লেখা
- নতুন অডিও

এসবেও performance ভালো হয়।

3) Supports Transfer Learning (অন্য কাজে সহজে ব্যবহার করা যায়)

SSL মডেলকে আগে unlabeled ডেটা দিয়ে pre-train করা হয়।

তারপর ছোট dataset দিয়ে fine-tune করা হয়।

ফলে:

- training দ্রুত হয়
- accuracy বেশি হয়
- কম data দিয়েও ভালো ফল পাওয়া যায়

4) Scalable (বড় ডেটায় সহজে কাজ করে)

SSL বিশাল dataset handle করতে পারে, কারণ:

- label বানানোর bottleneck নেই

Big data scenario-তে SSL খুব উপকারী।

Limitations of Self-Supervised Learning (SSL-এর সীমাবদ্ধতা)

SSL খুব শক্তিশালী হলেও কিছু সমস্যা আছে। এগুলো বোঝা জরুরি।

1) Quality of Supervision Signal (Pseudo-label ভুল হতে পারে)

SSL যেহেতু নিজে label বানায়, তাই label সবসময় perfect হয় না।

এগুলোকে বলা হয়:

Pseudo-label

সমস্যা হলো:

- pseudo-label noisy হতে পারে
- incomplete হতে পারে

ফলে supervised learning-এর মতো সবসময় maximum accuracy পাওয়া যায় না।

2) Task Restrictions (সব ডেটায় SSL সহজ না)

SSL কাজ করে যখন:

ডেটা থেকে একটা meaningful “pretext task” বানানো যায়।

কিন্তু কিছু ক্ষেত্রে:

- ডেটা খুব unstructured
- ডেটার pattern খুব জটিল
- task design করা কঠিন

তখন SSL কম কার্যকর হয়।

3) Training Complexity (Training সেটআপ কঠিন)

SSL-এর অনেক method যেমন:

- Contrastive Learning

এগুলোর জন্য দরকার হয়:

- careful design
- hyperparameter tuning
- ভালো batch strategy

এটা beginner-এর জন্য কঠিন হতে পারে।

4) High Computational Cost (কম্পিউটেশন খরচ বেশি)

SSL মডেলগুলো সাধারণত বড় হয় এবং dataset-ও বিশাল হয়।

ফলে দরকার হয়:

- powerful GPU/TPU
- অনেক training time
- বড় batch size

এজন্য SSL training অনেক সময় expensive হয়ে যায়।

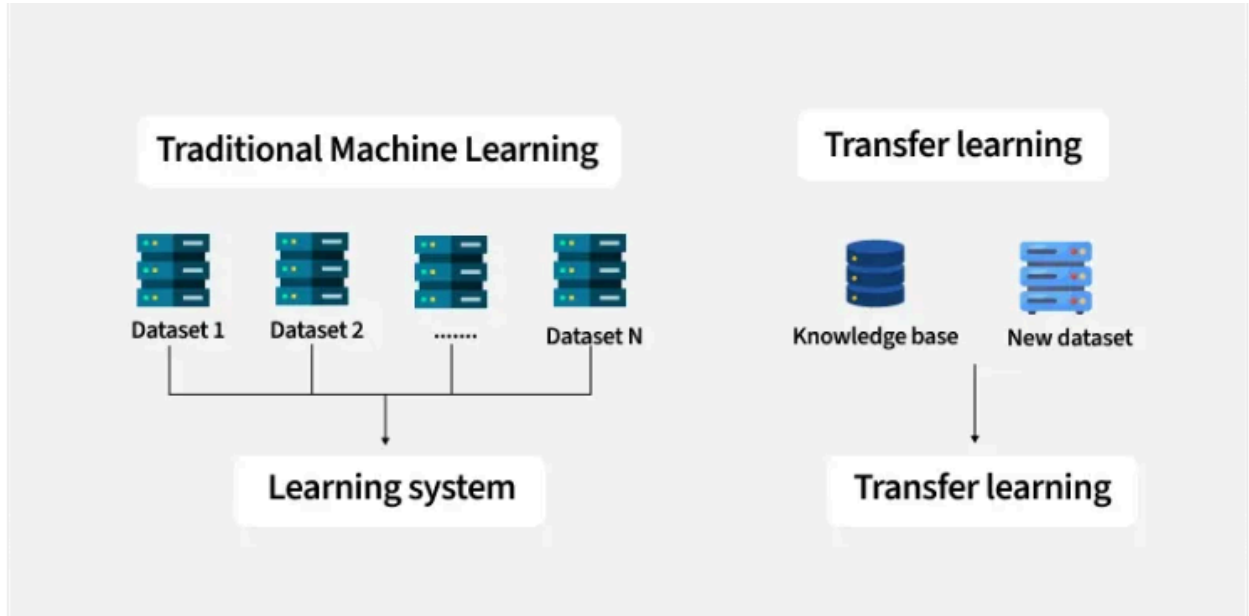
Transfer Learning কী?

Transfer Learning হলো Machine Learning-এর এমন একটি কৌশল যেখানে:

একটা কাজের জন্য আগে থেকে শেখানো (trained) একটি মডেলকে আরেকটা নতুন, কিন্তু সম্পর্কিত কাজের জন্য ব্যবহার করা হয়।

সহজ কথায়:

এক জায়গায় শেখা জ্ঞান অন্য জায়গায় কাজে লাগানোই **Transfer Learning**।



সহজ বাস্তব উদাহরণ

ধরো:

- একজন মানুষ আগে ইংরেজি শিখেছে
- পরে সে জার্মান শিখতে গেলে তার কাজ সহজ হয়

কারণ:

- ব্যাকরণ শেখার ধারণা আছে
- শব্দ শেখার অভ্যাস আছে

Machine Learning-এও ঠিক একই ব্যাপার।

একটা মডেল যদি আগে অনেক ডেটা দেখে “শেখে” থাকে, তাহলে নতুন কাজ শিখতে তার সময় কম লাগে।

Transfer Learning কেন দরকার?

সব সময় নতুন কাজের জন্য:

- প্রচুর labelled data পাওয়া যায় না
- শুরু থেকে model train করা ব্যয়বহুল

Transfer Learning এই সমস্যা সমাধান করে।

Transfer Learning-এর মূল ধারণা

Transfer Learning ধরে নেয় যে:

একটি কাজের জন্য শেখা **low-level** ও **general feature** আরেকটি সম্পর্কিত কাজেও কাজে লাগবে।

যেমন:

- ছবিতে edge, shape, texture
- লেখায় grammar, word relationship
- অডিওতে sound pattern

এই feature গুলো এক কাজ থেকে আরেক কাজে transfer করা যায়।

Importance of Transfer Learning

এখন এক এক করে কারণগুলো বুঝি।

1) Limited Data সমস্যার সমাধান

বাস্তবে বড় সমস্যা হলো:

- পর্যাপ্ত labelled data পাওয়া যায় না
- Label বানানো সময় ও খরচ সাপেক্ষ

Transfer Learning-এ:

- আগে থেকেই trained model ব্যবহার করা হয়
- নতুন কাজের জন্য অল্প ডেটাই যথেষ্ট হয়

ফলে কাজ অনেক সহজ হয়ে যায়।

2) Enhanced Performance (ভালো পারফরম্যান্স)

যেহেতু মডেল আগে থেকেই অনেক ডেটা দেখে শিখেছে:

- সে data structure বুঝে
- useful pattern চিনতে পারে

ফলে:

- দ্রুত converge করে
- accuracy বেশি হয়
- stable result পাওয়া যায়

বিশেষ করে যেখানে high accuracy দরকার, সেখানে Transfer Learning খুব কার্যকর।

3) Time and Cost Efficiency

শুরু থেকে model train করলে দরকার হয়:

- অনেক সময়
- শক্তিশালী GPU
- বড় খরচ

Transfer Learning-এ:

- training time কমে যায়
- computational cost কম হয়
- resource efficiently ব্যবহার হয়

এ কারণেই industry-তে Transfer Learning বেশি ব্যবহার করা হয়।

4) Adaptability (বহুমুখী ব্যবহার)

একটা model বিভিন্ন কাজে ব্যবহার করা যায়।

যেমন:

- ImageNet-এ train করা model
 - medical image analysis
 - face recognition
 - object detection
- ভাষা model
 - translation

- sentiment analysis
- text classification

এজন্য Transfer Learning খুব versatile।

Working of Transfer Learning (কীভাবে কাজ করে)

এখন ধাপে ধাপে পুরো process টা বুঝি।

Step 1: Pre-trained Model

প্রথমে এমন একটি model নেওয়া হয় যেটা:

- বড় dataset দিয়ে train করা
- সাধারণ feature শিখেছে

এটা হতে পারে:

- ImageNet-trained CNN
- BERT, GPT টাইপ language model

এই model-টাই হলো foundation।

Step 2: Base Model কী?

Pre-trained model-টাকেই যখন আমরা নতুন কাজে ব্যবহার করি, তখন তাকে বলা হয় **Base Model**।

Base model-এর ভিতরে থাকে:

- একাধিক layer
- প্রতিটি layer ভিন্ন level-এর feature শেখে

যেমন:

- নিচের layer → simple feature (edge, color)
- মাঝের layer → pattern, shape
- উপরের layer → complex concept

Step 3: Transfer Layers নির্বাচন

সব layer নতুন কাজের জন্য দরকার হয় না।

তাই আমরা:

- এমন layer নির্বাচন করি যেগুলো **general knowledge** ধরে রেখেছে
- সাধারণত এগুলো early বা middle layer হয়

এই layer গুলোই এক কাজ থেকে আরেক কাজে transfer করা হয়।

Step 4: Fine-tuning

এখন নতুন কাজের ডেটা দিয়ে:

- কিছু layer freeze রাখা হয়
- কিছু layer retrain করা হয়

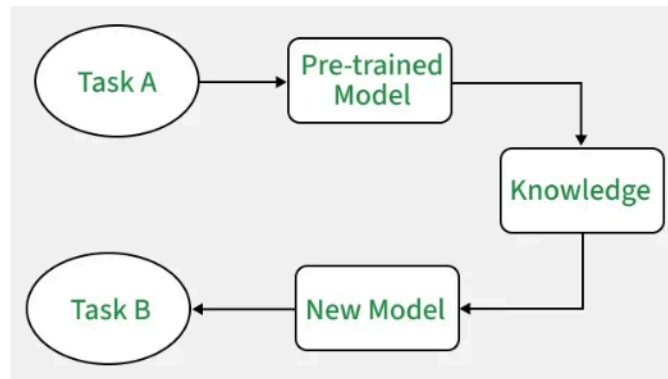
এই প্রক্রিয়াকে বলা হয় **Fine-tuning**।

এর ফলে:

- পুরোনো শেখা জ্ঞান নষ্ট হয় না
- নতুন কাজের সাথে model adjust হয়ে যায়

গুরুত্বপূর্ণ ধারণা (Task A → Task B)

Transfer Learning-এর মূল বিশ্বাস হলো:



Low-level features learned for task A should be beneficial for learning of model for task B.

Task A-তে শেখা **low-level feature**

Task B-তেও কাজে লাগবে।

যেমন:

- বিড়াল চিনতে শেখা model
→ কুকুর চিনতেও সাহায্য করে
- সাধারণ ইংরেজি শেখা model
→ news বা review বুঝতেও সাহায্য করে

Transfer Learning-এর সংক্ষিপ্ত সারাংশ

Transfer Learning মানে হলো:

- আগে শেখা model ব্যবহার করে
- নতুন কাজ দ্রুত ও ভালোভাবে শেখানো

এর সুবিধা:

- কম ডেটা লাগে
- সময় ও খরচ কম
- accuracy বেশি হয়

এ কারণেই modern AI system-এ Transfer Learning অত্যন্ত গুরুত্বপূর্ণ।

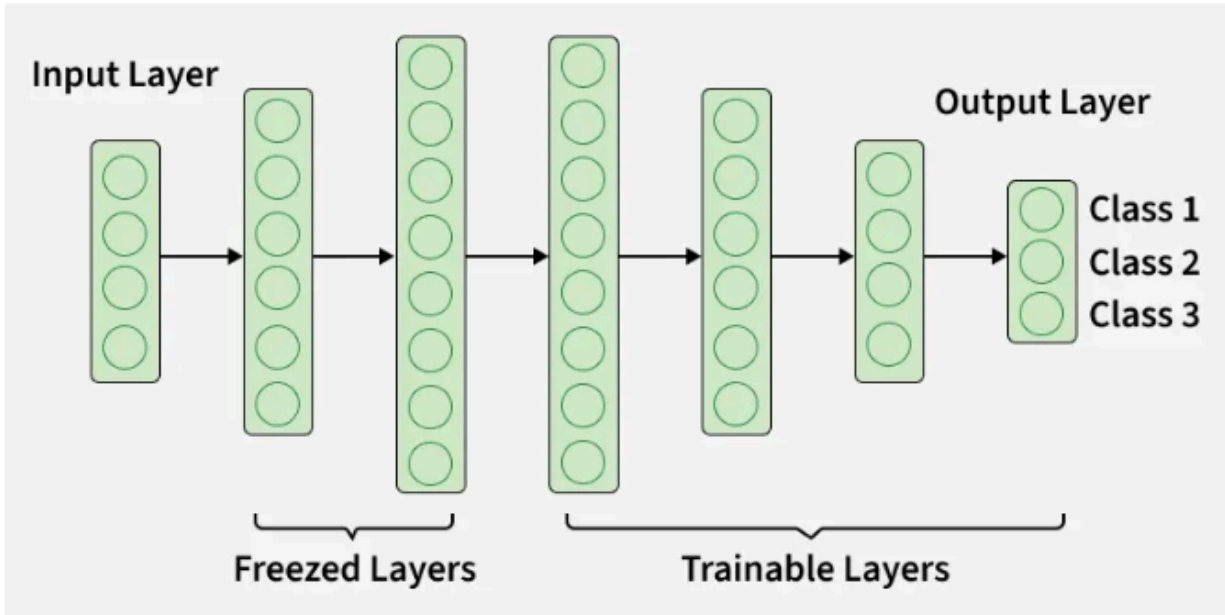
Frozen Layers vs Trainable Layers কী?

Transfer Learning-এ আমরা সাধারণত একটি **pre-trained model** ব্যবহার করি।
এই model-এর ভিতরে অনেকগুলো layer থাকে।

এই layer গুলোকে আমরা দুইভাবে ব্যবহার করতে পারি:

1. কিছু layer **Freeze** করে রাখা
2. কিছু layer **Trainable** করে রাখা

এই দুইটার পার্থক্য বোঝা খুব গুরুত্বপূর্ণ।



Frozen Layers কী?

Frozen Layers হলো সেই layer গুলো:

- যাদের weight আমরা অপরিবর্তিত রাখি
- Training-এর সময় update করি না

অর্থাৎ:

এই layer গুলো আগে যেটা শিখেছিল, সেটাই ধরে রাখে।

Frozen Layers-এর কাজ কী?

Frozen layer-এর মূল উদ্দেশ্য:

বড় **dataset** থেকে শেখা **general feature** সংরক্ষণ করা

যেমন:

- ছবির ক্ষেত্রে: edge, texture, basic shape
- লেখার ক্ষেত্রে: grammar, word relationship
- অডিওর ক্ষেত্রে: basic sound pattern

এই feature গুলো প্রায় সব task-এই কাজে লাগে।

Frozen Layers কীভাবে শেখে?

Frozen layer-এ:

- Backpropagation হয় না
- Weight update হয় না
- Layer গুলো constant থাকে

ফলে training দ্রুত হয়।

Frozen Layers কখন ব্যবহার করা হয়?

- যখন নতুন dataset ছোট
- অথবা নতুন task পুরোনো task-এর মতোই

এতে overfitting কম হয়।

Trainable Layers কী?

Trainable Layers হলো সেই layer গুলো:

- যাদের weight training-এর সময় update করা হয়
- নতুন dataset অনুযায়ী adjust করা হয়

এগুলো model-কে নতুন কাজের সাথে মানিয়ে নিতে সাহায্য করে।

Trainable Layers-এর কাজ কী?

Trainable layer-এর উদ্দেশ্য:

নতুন **task**-এর জন্য **specific feature** শেখা

যেমন:

- medical image-এ tumor pattern
- sentiment analysis-এ emotion-related feature
- fraud detection-এ unusual transaction pattern

Trainable Layers কীভাবে শেখে?

Trainable layer-এ:

- Backpropagation হয়
- Loss অনুযায়ী weight update হয়
- Model ধীরে ধীরে নতুন কাজ শিখে

Trainable Layers কখন ব্যবহার করা হয়?

- যখন dataset বড়
- অথবা নতুন task পুরোনো task থেকে অনেক আলাদা

Frozen vs Trainable Layers

Definition (সংজ্ঞা)

- **Frozen Layers:** যাদের weight স্থির থাকে
- **Trainable Layers:** যাদের weight পরিবর্তন হয়

Purpose (উদ্দেশ্য)

- Frozen: সাধারণ knowledge ধরে রাখা
- Trainable: নতুন কাজের সাথে মানিয়ে নেওয়া

Learning Process

- Frozen: শেখা বন্ধ, শুধু ব্যবহার
- Trainable: শেখা চালু থাকে

Use Case

- Frozen: ছোট বা similar dataset

- Trainable: বড় বা ভিন্ন dataset

Computation Cost

- Frozen: কম খরচ
- Trainable: বেশি খরচ

CNN-এর উদাহরণ

- Frozen: শুরু দিকের convolution layer (edge, texture, shape ধরে)
- Trainable: শেষের layer (task-specific feature ধরে)

Input Layer-এর সাথে সম্পর্ক

Input layer-এর কাছের layer গুলো সাধারণত:

- basic feature শেখে
- সব task-এই কাজে লাগে

তাই এগুলোকে অনেক সময় freeze করা হয়।

আর output-এর কাছের layer গুলো:

- task-specific decision নেয়
- তাই এগুলো trainable রাখা হয়।

কোন Layer Freeze করবো, কোনটা Train করবো?

এটা নির্ভর করে দুইটা জিনিসের উপর:

1. Dataset-এর আকার
2. Dataset কতটা আলাদা

এখন চারটা case দেখি।

Case 1: Small + Similar Dataset

যেমন:

- ImageNet → cat vs dog
- English text → product review sentiment

এই ক্ষেত্রে:

- বেশিরভাগ layer freeze
- শুধু শেষ ১-২টা layer train

কারণ:

- dataset ছোট
 - overfitting-এর ঝুঁকি বেশি
-

Case 2: Large + Similar Dataset

যেমন:

- ImageNet → object detection dataset

এই ক্ষেত্রে:

- কিছু extra layer unfreeze করা যায়
 - model একটু বেশি adapt করতে পারে
-

Case 3: Small + Different Dataset

যেমন:

- natural image → medical image (ছোট dataset)

এই ক্ষেত্রে:

- input-এর কাছের কিছু layer train করা দরকার
- কারণ feature আলাদা

তবে সাবধানে করতে হয়, না হলে overfitting হবে।

Case 4: Large + Different Dataset

যেমন:

- general image → satellite image (বড় dataset)

এই ক্ষেত্রে:

- পুরো model fine-tune করা যায়
- কারণ data যথেষ্ট আছে

Applications of Frozen & Trainable Layers

Computer Vision

- medical imaging
- face recognition
- object detection

Natural Language Processing

- BERT / GPT fine-tuning
- sentiment analysis
- translation
- question answering

Healthcare

- X-ray, MRI analysis
- disease detection

Finance

- fraud detection
- credit scoring
- risk analysis

Advantages (সুবিধা)

1) Training দ্রুত হয়

- শুরু থেকে শেখা লাগে না
- pre-trained knowledge ব্যবহার হয়

2) Performance ভালো হয়

- model আগে থেকেই smart
- কম data দিয়েও ভালো ফল দেয়

3) Small Dataset-এ কাজ করে

- overfitting কম হয়
- general feature আগেই শেখা থাকে

Disadvantages (অসুবিধা)

1) Domain Mismatch

- যদি পুরোনো task আর নতুন task খুব আলাদা হয়
- তাহলে pre-trained knowledge কাজে নাও লাগতে পারে

2) Overfitting

- বেশি layer train করলে
- ছোট dataset-এ model মুখস্থ করে ফেলতে পারে

3) Complexity ও Cost

- fine-tuning computationally expensive

- GPU দরকার হতে পারে

Leveraging Unlabeled Data

১. Leveraging Unlabeled Data মানে কী?

Leveraging unlabeled data মানে হলো —

মানুষের দেওয়া লেবেল ছাড়া (label ছাড়া) অনেক ডেটা ব্যবহার করে মডেলকে ট্রেন করানো এবং ডেটার ভেতরের প্যাটার্ন নিজে নিজে শেখানো।

সহজভাবে

Supervised Learning এ:

- Input + Label দরকার (মানুষ দিয়ে বানাতে হয়, সময় লাগে, খরচ বেশি)

Self-Supervised Learning এ:

- Model নিজেই **supervision signal** তৈরি করে
- মানুষের label লাগে না

অর্থাৎ → **Unlabeled data = Learning resource**

২. Unlabeled Data কেন গুরুত্বপূর্ণ?

বাস্তবে:

- Labeled data → কম + expensive
- Unlabeled data → অনেক + ফ্রি

উদাহরণ:

- ইন্টারনেটে কোটি কোটি ছবি (label নেই)
- অনেক text data
- Medical, sensor, cybersecurity log data

Self-Supervised Learning প্রথমে unlabeled data থেকে **representation** শেখে, পরে ছোট labeled data দিয়ে final task শেখে।

এতে:

- খরচ কমে
- কম data তেও ভালো কাজ করে
- Overfitting কম হয়

৩. Self-Supervised Learning এর মূল আইডিয়া

Model unlabeled data থেকে একটি **pretext task** বানায় (fake supervised task)।

উদাহরণ:

Data	Pretext Task
Image	ছবির missing অংশ predict করা
Text	Masked word predict করা
Audio	Next signal predict
Time series	Future value predict

এতে model **structure + pattern + meaning** শেখে।

৪. Unlabeled Data ব্যবহার করার প্রধান পদ্ধতি

৪.১ Pseudo-Labeling

ধাপ:

1. ছোট labeled data দিয়ে model train
2. Unlabeled data এর label model নিজে predict করে
3. সেই label কে pseudo label হিসেবে ব্যবহার

4. আবার retrain

লক্ষ্য → dataset বড় করা

সমস্যা → ভুল pseudo label হলে noise ঢুকে যায়

8.২ Contrastive Learning

Idea:

Model শেখে **similar vs different data compare** করে

উদাহরণ (Image):

- একই ছবির দুই augmentation → Positive pair
- আলাদা ছবি → Negative pair

Objective:

Similar → কাছে আনবে

Different → দূরে পাঠাবে

Result → label ছাড়া শক্তিশালী feature learning

8.৩ Autoencoder

Structure:

Encoder → data compress

Decoder → original reconstruct

Model শেখে **latent representation**

Loss = Reconstruction error

ব্যবহার:

- Feature extraction
- Dimensionality reduction
- Anomaly detection

8.8 Masked Modeling

Text এ \rightarrow BERT
Image এ \rightarrow MAE

Process:

Input এর কিছু অংশ mask \rightarrow model missing অংশ predict

উদাহরণ:

“The cat is on the ____” \rightarrow model = “table”

Model context + meaning শেখে।

৪.৫ Consistency Regularization

Idea:

Input একটু বদলালেও prediction same থাকতে হবে।

উদাহরণ:

- Original image
- Noisy image

Prediction same \rightarrow model robust হয়

৫. Training Pipeline

Step 1: Pretraining (Unlabeled Data)

Model general feature শেখে

Step 2: Fine-tuning (Small Labeled Data)

Specific task শেখে:

- Classification
- NLP
- Medical diagnosis
- Cybersecurity

৬. Mathematical Idea

আমরা চাই:

$$z = f(x)$$

x = input

z = learned feature

Loss হতে পারে:

- Reconstruction loss
- Contrastive loss
- Cross entropy (masked prediction)

Goal → label ছাড়া useful feature শেখা

৭. সুবিধা

1. প্রচুর data ব্যবহার করা যায়
2. Labeling cost কম
3. Better generalization
4. Strong representation
5. Small labeled data তেও কাজ করে
6. Large AI model (LLM, ViT) train করা যায়

৮. সমস্যা

1. ভালো pretext task বানানো কঠিন
2. Training cost বেশি হতে পারে
3. Learned feature সবসময় final task এ perfect না
4. Contrastive method এ collapse হতে পারে
5. Large compute লাগতে পারে

৯. Real-World ব্যবহার

- NLP → BERT, GPT
- Computer Vision → SimCLR, MAE
- Medical imaging → Disease detection
- Cybersecurity → Anomaly detection

- Speech recognition
- Recommender system
- Autonomous driving

১০. Supervised vs Self-Supervised

বিষয়	Supervised	Self-Supervised
Label লাগে	হ্যাঁ	না
Data scale	কম	অনেক
Cost	বেশি	কম
Feature	Task specific	General
Pretraining	না	হ্যাঁ

Self-Prediction vs Contrastive Learning

Self-Supervised Learning-এ মডেল **label** ছাড়া **data** থেকেই শেখে। এই শেখার দুটি প্রধান পদ্ধতি হলো:

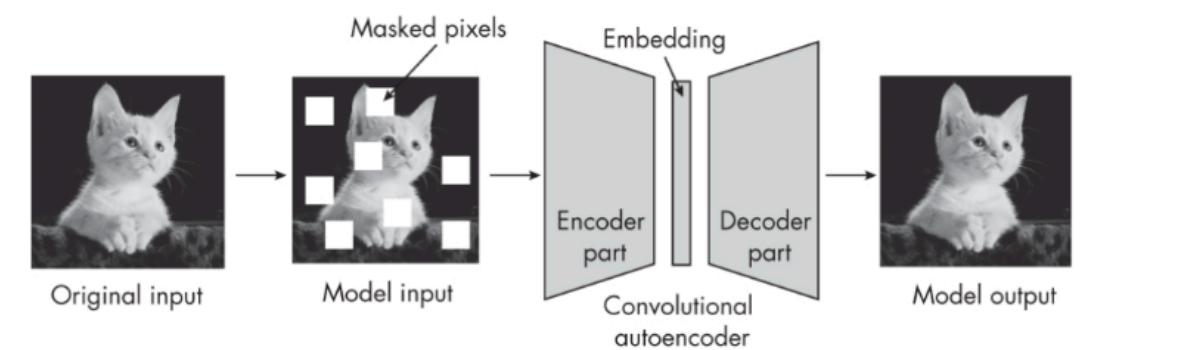
1. **Self-Prediction Method**
2. **Contrastive Learning Method**

এই দুই পদ্ধতির উদ্দেশ্য একই — **Unlabeled data** থেকে ভালো **representation** শেখা, কিন্তু শেখার পদ্ধতি ভিন্ন।

১. Self-Prediction Method কী?

Self-Prediction এ input data-এর কিছু অংশ লুকানো বা পরিবর্তন করা হয়, তারপর মডেলকে বলা হয় **original data** পুনরুদ্ধার (**reconstruct**) করতে।

অর্থাৎ → Model নিজের data থেকেই prediction শিখে।



Self-Prediction কীভাবে কাজ করে (Step by Step)

Step 1: Raw Data

ধরো, তোমার কাছে অনেক image আছে, কিন্তু কোনো label নেই।

Step 2: Artificial Task তৈরি

আমরা data-এর কিছু অংশ mask বা hide করি।

উদাহরণ:

- ছবির কিছু অংশ ঢেকে দেওয়া
- Text-এর কিছু শব্দ mask করা

Step 3: Prediction

Model কে বলা হয় → Missing অংশ predict করো।

Step 4: Learning

Missing অংশ predict করতে গিয়ে model শিখে:

- Context
- Shape
- Texture
- Object structure
- Data pattern

এগুলোই **useful features**।

কেন এটাকে **Self-Supervised** বলে?

এখানে label বাইরে থেকে আসে না।

Input-এর hidden অংশই label হিসেবে কাজ করে।

Model নিজেই supervision তৈরি করে।

Classic Example: Masked Autoencoder

Masked Autoencoder:

- Image-এর কিছু অংশ mask করা হয়
- Model original image reconstruct করে
- এতে model context, structure, pattern শেখে

২. Contrastive Learning কী?

Contrastive Learning-এ model শেখে **data compare** করে।

Similar data → কাছাকাছি

Different data → দূরে

Model একটি **embedding space** শেখে যেখানে similar object গুলো cluster হয়ে থাকে।



The supervision comes from the data itself—specifically, the relationships between different

সহজ উদাহরণ

ধরো:

- একই মানুষের দুইটি ছবি → Similar (Positive pair)
- দুইজন আলাদা মানুষ → Different (Negative pair)

Model শেখে:

- Similar → Distance কম
- Different → Distance বেশি

এভাবেই representation শেখে।

Supervision কোথা থেকে আসে?

Label নেই, কিন্তু **Similarity relation** আছে।

Model শেখে:

- Positive pair vs Negative pair

এই self-generated signal-ই supervision।

Contrastive Learning-এর ধরন

Contrastive Learning-এ মডেল শেখে **similar data** কাছাকাছি এবং **dissimilar data** দূরে রাখতে।
এই শেখার দুইটি প্রধান স্টাইল আছে:

1. **Sample Contrastive Learning**
2. **Dimension Contrastive Learning**

১. Sample Contrastive Learning

কী?

এখানে একটি **sample (data point)**-কে অন্য sample-এর সাথে compare করে শেখানো হয়।

অর্থাৎ → **Sample vs Sample comparison**

কীভাবে কাজ করে

ধরো একটি image আছে।

আমরা একই image-এর দুইটি augmented version বানালাম:

- Crop
- Rotate
- Blur
- Color change

এখন:

- Same image (augmented) → **Positive pair**
- Different image → **Negative pair**

Model শেখে:

- Positive pair → embedding distance কমাও
- Negative pair → embedding distance বাড়াও

Example (Image)

Dog image → augmentation → Dog image again
Cat image → different

Model শেখে:

- Dog vs Dog → Similar
- Dog vs Cat → Different

Objective

Embedding space-এ:

- Similar sample → কাছাকাছি
- Different sample → দূরে

Used in

- SimCLR
- MoCo
- CLIP
- BYOL (modified)

Representation শেখে কেমন?

- 👉 Global semantic feature
- 👉 Object identity
- 👉 General representation

কোথায় বেশি **useful**?

- Image classification pretraining
- Face recognition
- Clustering
- Representation learning

২. Dimension Contrastive Learning

কী?

এখানে **sample vs sample** নয়, বরং একই **sample**-এর **different feature dimension compare** করা হয়।

অর্থাৎ → **Feature vs Feature comparison**

Simple Idea

ধরো একটি sample-এর embedding vector:

$z = [z_1, z_2, z_3, z_4, z_5]$

এখানে প্রতিটি dimension একটি feature।

Dimension Contrastive Learning শেখে:

- Same feature dimension across views → Similar
- Different dimension → Independent / informative

কীভাবে কাজ করে

ধরো:

একই image-এর দুইটি view:

View 1 \rightarrow embedding $\rightarrow z^1$

View 2 \rightarrow embedding $\rightarrow z^2$

এখন:

- z^1_1 vs $z^2_1 \rightarrow$ Positive (same dimension)
- z^1_1 vs $z^2_2 \rightarrow$ Negative (different dimension)

Model শেখে:

- Same dimension \rightarrow correlation বাড়াও
- Different dimension \rightarrow redundancy কমাও

Objective

- 👉 Feature dimension-গুলোকে informative করা
- 👉 Redundant feature কমানো
- 👉 Representation decorrelate করা

Representation শেখে কেমন?

- Disentangled feature
- Independent representation
- Informative embedding

কোথায় useful?

- Feature learning improvement
- Redundancy reduction
- Self-supervised representation refinement

Main Difference (Simple Table)

বিষয়	Sample Contrastive	Dimension Contrastive
Comparison	Sample vs Sample	Feature vs Feature
Positive pair	Same data (different view)	Same feature dimension
Negative pair	Different sample	Different dimension
Goal	Similar object group করা	Feature informative করা
Representation	Semantic / Global	Disentangled / Independent
Popularity	খুব বেশি	তুলনামূলক কম

৩. Self-Prediction vs Contrastive Learning

Objective (লক্ষ্য)

Self-Prediction

- Corrupted input থেকে original reconstruct
- Missing অংশ predict করা

Contrastive

- Similar vs Different distinguish করা
- Embedding space শেখা

Data Manipulation

Self-Prediction

- Data-এর ভেতরের অংশ hide / modify
- Single data point নিয়ে কাজ

Contrastive

- Different data compare
- Pair বা group তৈরি
- Data modify করা হয় না

Learning Mechanism

Self-Prediction

- Data-এর internal structure শেখে
- Context + continuity বুঝে

Contrastive

- Data-এর relationship শেখে
- Similarity / dissimilarity বুঝে

Representation Learning

Self-Prediction

- Local feature শেখে
- Reconstruction-focused

Contrastive

- Global feature শেখে
- More generalized representation

Application Suitability

Self-Prediction ভালো যেখানে:

- Image restoration
- Denoising
- Inpainting
- Language masked prediction

Contrastive ভালো যেখানে:

- Clustering
- Anomaly detection
- Classification
- Representation learning

Example Use Cases

Self-Prediction

- Autoencoder
- Image completion
- Next word prediction (Language model)

Contrastive

- Face recognition
- Unsupervised pretraining
- Feature learning
- Fine-grained classification