Introduction to Semi-Supervised Learning

What is Semi-Supervised Learning (SSL)?

- SSL is a machine learning paradigm that uses both:
  - A small amount of labeled data
  - A large amount of unlabeled data
- It lies between supervised and unsupervised learning.
- Goal: Improve model performance by leveraging unlabeled data.

Motivation: Real-Life Example

- Imagine you want to classify emails as "Spam" or "Not Spam".
- You label 100 emails manually (costly and time-consuming).
- You have 10,000 more unlabeled emails.
- SSL helps use the 100 labeled and 10,000 unlabeled emails to train a better classifier than using 100 alone.

Another Example: Medical Diagnosis

- Labeled Data: 500 X-ray images diagnosed by radiologists.
- Unlabeled Data: 50,000 raw X-rays without diagnosis.
- SSL can help predict disease labels by learning patterns from both labeled and unlabeled data.

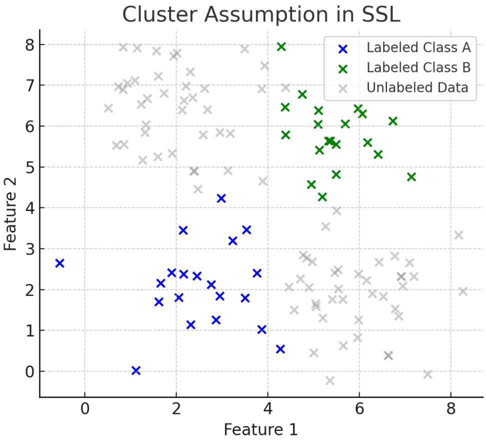
Mathematical Formulation

- Let labeled data be: $\mathcal{D}_L = \{(x_i, y_i)\}_{i=1}^{l}$
- Unlabeled data: $\mathcal{D}_U = \{x_i\}_{i=l+1}^{l+u}$
- The objective is to learn a function $f(x)$ that performs well on both labeled and unlabeled inputs.

Why is SSL Important?

- In many domains, labels are expensive, but unlabeled data is cheap.
- Examples:
  - Medical imaging
  - Speech recognition
  - Text classification
  - Autonomous driving
- SSL can significantly reduce annotation costs.

Visualizing SSL



- Labeled points guide the boundary.
- Unlabeled points help refine the decision surface.

SSL vs Other Learning Types

Learning Type	**Data Used**	**Example**
Supervised	Only labeled data	Image classification with labels
Unsupervised	Only unlabeled data	Clustering customer data
Semi-Supervised	Few labeled + many unlabeled	Spam detection with few labeled emails

How SSL Works: High-Level Intuition

- SSL assumes structure in data.
- Example Assumptions:
  - **Cluster assumption:** Same class points form clusters.
  - **Smoothness assumption:** Nearby points have similar labels.
  - **Manifold assumption:** Data lies on a low-dimensional manifold.

Summary

- SSL is powerful when labeled data is scarce.
- It bridges the gap between supervised and unsupervised learning.
- Useful in many real-world scenarios.
- We now move on to key SSL methods like Ladder Networks and -Models.

Assumptions in Semi-Supervised Learning

What is Semi-Supervised Learning?

- Combines a small amount of labeled data with a large amount of unlabeled data.
- Goal: Improve learning accuracy with less labeling effort.
- Bridge between Supervised and Unsupervised learning.

Self-Training Assumption

- **Assumption:** If a model is confident about its prediction on an unlabeled point, it is likely correct.
- **Explanation:** Train on labeled data, predict on unlabeled data. Add confident predictions to training data and retrain.
- **Example:** Suppose we have:
  - Labeled data: Apple (red, round), Banana (yellow, long)
  - Unlabeled image looks red and round. Model predicts "Apple" with 98% confidence.
  - Add it as "Apple" to labeled set, retrain.
- **Works well when:** Model is reliable and consistent in confidence.

Co-Training Assumption

- **Assumption:** Two independent and sufficient views of data can teach each other.
- **Explanation:** Train two classifiers on different feature sets (views). Each helps improve the other by labeling unlabeled data.
- **Example:** Classifying web pages as "Sports" or "Politics":
  - View 1: Words in the page ("goal", "election")
  - View 2: Links to/from the page (ESPN, CNN)
  - Classifier A (text) labels a page as "Sports". Classifier B (links) uses it as training.
  - Each teaches the other.
- **Works well when:** Views are conditionally independent and each is sufficient.

Generative Model Assumption

- **Assumption:** Data comes from known probabilistic distributions.
- **Explanation:** Fit distributions using both labeled and unlabeled data. Classify based on likelihood under each class distribution.
- **Example:**
  - Labeled: (1,1) = Class A, (5,5) = Class B
  - Unlabeled data form two blobs around these points.
  - Fit Gaussians to each cluster. Classify new points based on proximity to each distribution.
- **Works well when:** Class distributions match assumed probabilistic models.

Cluster Assumption

- **Assumption:** Points in the same cluster likely share the same label.
- **Explanation:** Use clustering to infer labels from a few labeled points.
- **Example:**
  - You have a cluster of images mostly labeled “Dog”.
  - Another cluster is mostly “Cat”.
  - Unlabeled images within the “Dog” cluster are assumed to be “Dog”.
- **Works well when:** Clusters are well-separated and meaningful.

Low-Density Separation Assumption

- **Assumption:** Decision boundary should pass through low-density regions.
- **Explanation:** Avoid placing the boundary where there are many data points (high density).
- **Example:**
  - Two moons dataset with few points in the middle gap.
  - A good classifier finds a decision boundary through the sparse middle, not through the dense moons.
- **Works well when:** Classes are naturally separated by sparse regions.

Manifold Assumption

- **Assumption:** Data lies on a low-dimensional manifold; labels vary smoothly along it.
- **Explanation:** Even in high-dimensional space, data has lower-dimensional structure. Label propagation can follow this structure.
- **Example:**
  - Handwritten digits vary smoothly by slant, stroke, thickness.
  - "3" written in different ways form a smooth curve on a manifold.
  - Label a few digits, then propagate labels across nearby points on the manifold.
- **Works well when:** Data has smooth, continuous variations.

Summary Table

Assumption	**Key Idea**	**Example**
Self-Training	Confident predictions are correct	Red fruit labeled as Apple
Co-Training	Two independent views teach each other	Webpage text + links
Generative Model	Known distributions generate data	Gaussian blobs
Cluster	Same cluster implies same label	Cat and dog blobs
Low-Density Separation	Boundary through sparse regions	Two moons dataset
Manifold	Smooth label variation on low-D manifold	Handwritten digits

Related Learning Paradigms to Semi-Supervised Learning

Semi-Supervised Learning (SSL) – Recap

- **Data:** Small labeled + large unlabeled data
- **Goal:** Use unlabeled data to boost performance of supervised models
- **Assumptions:** Cluster, low-density separation, smoothness, manifold
- **Example:** 100 labeled cat/dog images + 10,000 unlabeled

Transfer Learning

- **Definition:** Learn in one domain (source task) and transfer to another (target task)
- **Example:** Pretrained ImageNet model transferred to X-ray classification

SSL vs Transfer Learning

Aspect	**SSL**	**Transfer Learning**
Labeled data	Small amount in target task	Abundant in source, few in target
Unlabeled data	Used in target task	Not typically used
Goal	Leverage unlabeled data	Transfer knowledge

Weakly-Supervised Learning

- **Definition:** Learning from labels that are noisy, coarse, or incomplete
- **Example:** Video classification with video-level but not frame-level labels

SSL vs Weakly-Supervised

Aspect	**SSL**	**Weak Supervision**
Label quality	Few clean labels	Noisy or incomplete labels
Unlabeled data	Crucial	Optional
Goal	Improve using unlabeled data	Handle imperfect labels

Positive and Unlabeled Learning (PU Learning)

- **Definition:** Learn from only positive and unlabeled data
- **Example:** Spam detection with only spam labeled, rest unlabeled

SSL vs PU Learning

Aspect	**SSL**	**PU Learning**
Label types	Positive + Negative + Unlabeled	Only Positive + Unlabeled
Goal	Support full classification	Infer negatives from unlabeled
Class imbalance	Balanced	Positive class only

Meta-Learning

- **Definition:** Learn to adapt quickly to new tasks with few labels
- **Example:** 5-way 1-shot classification on Omniglot

SSL vs Meta-Learning

Aspect	**SSL**	**Meta-Learning**
Unlabeled data	In same task	Not necessarily used
Task structure	One task	Many small tasks
Label quantity	Few + unlabeled	Very few per class

Self-Supervised Learning

- **Definition:** Create supervision from data via pretext tasks
- **Example:** SimCLR, BERT, predicting missing image patches

SSL vs Self-Supervised

Aspect	**SSL**	**Self-Supervised**
Use of labels	Needs some	Needs none
Use of unlabeled data	Supporting role	Full training
Learning objective	Predict labels	Solve pretext task

Summary Table

Paradigm	**Label Setup**	**Typical Use**
SSL	Few labeled + many unlabeled	Classification with cheap unla beled data
Transfer Learning	Pretrained + few labels	Domain adaptation
Weak Supervision	Noisy or incomplete labels	Large-scale noisy datasets
PU Learning	Only positive + unlabeled	Web, bio, spam detection
Meta-Learning	Many few-shot tasks	Few-shot classification
Self-Supervised	No labels, pretext tasks	Representation learning

Detailed Examples

- **SSL:** 200 labeled medical images + 20K unlabeled
- **Transfer:** Pretrained ImageNet model adapted to X-ray images
- **Weakly-Supervised:** Topic-labeled videos without frame-level tags
- **PU:** Only fraud (positive) transactions labeled
- **Meta-Learning:** 1-shot image classification per class
- **Self-Supervised:** Learn features from image patches, fine-tune

Inductive vs Transductive Learning in Semi-Supervised Learning

Core Difference

- **Inductive Learning:** Learns a general function $f(x)$ to apply on any new input.
- **Transductive Learning:** Only predicts labels for the current unlabeled dataset.

Inductive Learning in SSL

- **Goal:** Learn a predictive model that generalizes.
- **Example:**
  - Train on 500 labeled + 5000 unlabeled cat/dog images.
  - Use pseudo-labeling to train a model.
  - This model can classify any new image later.
- **Applications:** Classification, object detection, medical imaging

Transductive Learning in SSL

- **Goal:** Label only the provided unlabeled data.
- **Example:**
  - 100 graded essays + 900 ungraded.
  - Use graph-based label propagation.
  - Predict grades for the 900, no general model is created.
- **Applications:** Document classification, node labeling in graphs

Summary Table (Part 1)

Feature	**Inductive**	**Transductive**
Goal	Learn general classifier $f(x)$	Predict current unlabeled labels
Output	General-purpose model	No reusable model
Generalization	Works on unseen data	Cannot handle new data

Summary Table (Part 2)

Feature	**Inductive**	**Transductive**
Examples	Pseudo-labeling, MixMatch	Label propagation, TSVM
Advantage	Deployable in real-world	High accuracy on fixed data
Limitation	May be less accurate than transductive	No generalization

Real-World Analogy

You are a tutor grading essays.

- **Transductive:** Estimate grades just for current papers by comparing them to known graded ones.
- **Inductive:** Derive a grading rubric and use it to evaluate current and future essays.

Which One Should You Use?

Use Case	**Best Choice**
Want to deploy model in production	Inductive
Labeling a fixed unlabeled batch	Transductive
High accuracy on known unlabeled samples	Transductive
Need to classify future unseen inputs	Inductive

Ladder Networks and -Models in Semi-Supervised Learning

Introduction

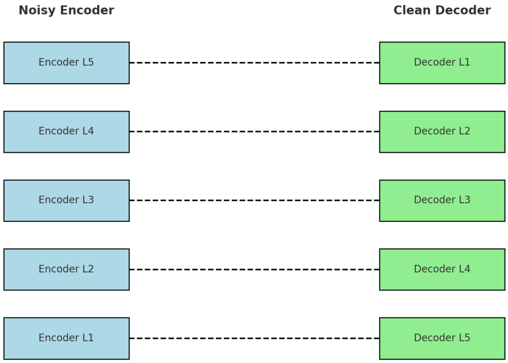
- Semi-Supervised Learning (SSL) combines small labeled and large unlabeled data.
- Ladder Networks and -Models are two classic SSL deep learning methods.
- They leverage unlabeled data via denoising and consistency regularization respectively.

What is a Ladder Network?

- A Ladder Network is a deep neural network architecture that combines:
    - A supervised classifier at the top
    - A denoising autoencoder for every intermediate layer
- Named for its "ladder"-like structure:
    - Encoder (bottom-up noisy path)
    - Decoder (top-down clean reconstruction path)
    - Skip connections resemble rungs of a ladder

Ladder Network Structure

- Each encoder layer outputs a noisy activation.
- Decoder tries to reconstruct clean version of each layer.
- Supervised loss is applied at the top layer using labeled data.



Skip connections resemble ladder rungs

Ladder Network Loss Function

- Total loss:

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \sum_{l=1}^{L} \lambda_l \cdot \mathcal{L}_{\text{recon}}^{l}$$

- $\mathcal{L}_{\text{sup}}$: Cross-entropy loss at top layer for labeled data.
- $\mathcal{L}_{\text{recon}}^{l}$: Mean squared error between clean and reconstructed activations at layer l.
- λ_l: Weight for reconstruction loss at each layer.

Example: MNIST with Ladder Network

- Labeled: 100 MNIST digits
- Unlabeled: 59,900 digits
- Encoder: Adds Gaussian noise
- Decoder: Reconstructs clean hidden states
- Learns robust representations + classifier simultaneously

What is a (Pi) Model?

- The Model enforces consistency between predictions under different noise.
- Named because it compares predictions *in parallel* (like the two legs of the letter Pi).
- It uses the same model twice with different dropout/noise/augmentations.

Model Mechanism

- Given input x, pass it twice through the model:

$$f_1(x + \epsilon_1), \quad f_2(x + \epsilon_2)$$

- Encourage both predictions to be consistent:

$$\mathcal{L}_{\text{unsup}} = \|f_1(x) - f_2(x)\|^2$$

- Add supervised loss for labeled examples:

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \alpha \cdot \mathcal{L}_{\text{unsup}}$$

Example: MNIST with Model

- Use 100 labeled and rest as unlabeled MNIST digits.
- Apply Gaussian noise to unlabeled images.
- Pass each noisy version through the model and match their outputs.
- Classifier becomes stable to input perturbations.

Comparison: Ladder vs Model

Aspect	**Ladder Network**	**Model**
Core Idea	Denoise hidden activations	Consistency of outputs under noise
Architecture	Encoder + Decoder	Single model (twice)
Loss	Supervised + reconstruction loss	Supervised + consistency loss
Noise Type	Gaussian at each layer	Dropout / Gaussian at input

Summary

- Both methods are foundational SSL models using deep networks.
- Ladder Network focuses on denoising internal states.
- Model enforces stable predictions under noise.
- Each has inspired newer methods like Mean Teacher, VAT, FixMatch.