Semi-supervised learning

Nazeef Hamid

MSS Math Talk

August 2025

Outline

1. Background
2. Machine learning in practice
3. Semi-supervised learning theory
4. Semi-supervised learning algorithms

Inputs and outputs

- **Features** (inputs) are measurable properties of data .
    - Numerical or categorical
    - $\mathbf{x} \in \mathbb{R}^d$ - d-dimensional feature vector
- **Labels** (outputs) are denoted y, possibly multi-dimensional
    - Continuous for regression problems
    - Finitely many values for classification problems
- Encode categorical labels with a "one-hot" vector

$$\text{“red”} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \text{“green”} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \text{“blue”} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

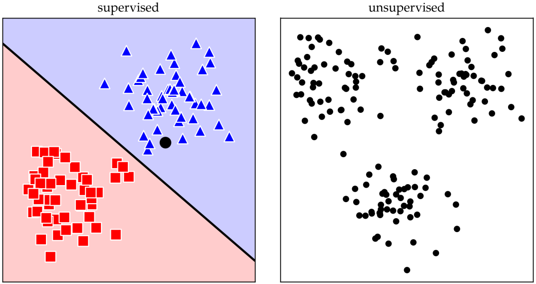
Supervised and unsupervised learning

Supervised learning: $\mathcal{D}_{\text{sl}} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$

- "learn" how to choose the correct y for a given $\mathbf{x}$

Unsupervised learning: $\mathcal{D}_{\text{usl}} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$

- Find patterns or trends in the $\mathbf{x}_i$



Semi-supervised learning is somewhere in the middle!

- $\mathcal{D}_{\text{ssl}} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_r, y_r), \mathbf{x}_{r+1}, \ldots, \mathbf{x}_n\}$

Why do semi-supervised learning?

Semi-supervised learning dataset - r is small

- $\mathcal{D}_{\text{ssl}} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_r, y_r), \mathbf{x}_{r+1}, \ldots, \mathbf{x}_n\}$

Labels can be very expensive, while data "in the wild" is usually cheap

- Medical imaging
- Fraud detection

In this talk we focus on classification problems

Empirical risk minimisation (ERM)

- Choose a function h that maps $\mathbf{x}$ to the correct y
- Loss function measures "correctness"
  - **Squared error:** $\mathcal{L}(h(\mathbf{x}), \mathbf{y}) = ||h(\mathbf{x}) - \mathbf{y}||_2^2$
  - **Absolute error:** $\mathcal{L}(h(\mathbf{x}), \mathbf{y}) = ||h(\mathbf{x}) - \mathbf{y}||_1$
  - **0-1 loss:** $\mathcal{L}(h(\mathbf{x}), \mathbf{y}) = \mathbb{1}_{\{h(\mathbf{x}) \neq \mathbf{y}\}}$
  - **Cross entropy:** $\mathcal{L}(h(\mathbf{x}), \mathbf{y}) = -\sum_{c=1}^{C} (\mathbf{y})_c \log((h(\mathbf{x}))_c)$
- Assume $h(\cdot; \boldsymbol{\theta})$ is a parametric function
  - Linear function $\mathbf{y} = W\mathbf{x} + \mathbf{b}$
  - Polynomial $a_n x^n + \cdots + a_1 x + a_0$
  - Neural network
- We wish to minimise the loss over **all** possible $(\mathbf{x}, \mathbf{y})$ pairs.

Empirical risk minimisation (ERM)

- In general we do not have all possible $(\mathbf{x}, y)$ pairs.
- We do our best using a sample

Definition (Empirical risk minimisation)

Given a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ define

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta} \in \Omega} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(h(\mathbf{x}_i; \boldsymbol{\theta}), y_i), \tag{1}$$

where $\boldsymbol{\theta}$ is a vector of parameters. The resulting model $h(\cdot; \boldsymbol{\theta}^*)$ is called the empirical risk minimiser for $\mathcal{D}$, $\mathcal{L}$.

- "Learning" or "training" = solve the optimisation problem

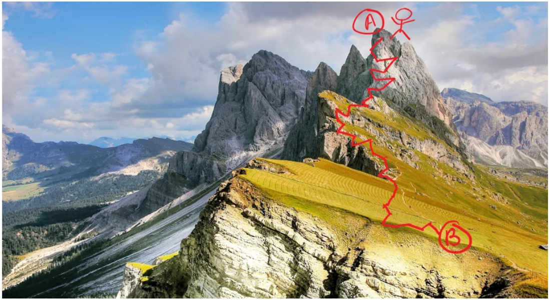
Gradient-based learning

- Train the model using gradient descent - iterative algorithm

$$f(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(h(\mathbf{x}_i; \boldsymbol{\theta}), y_i)$$

- Gradient descent update rule

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} + \eta^{(t)} \nabla f(\boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(t)}}$$

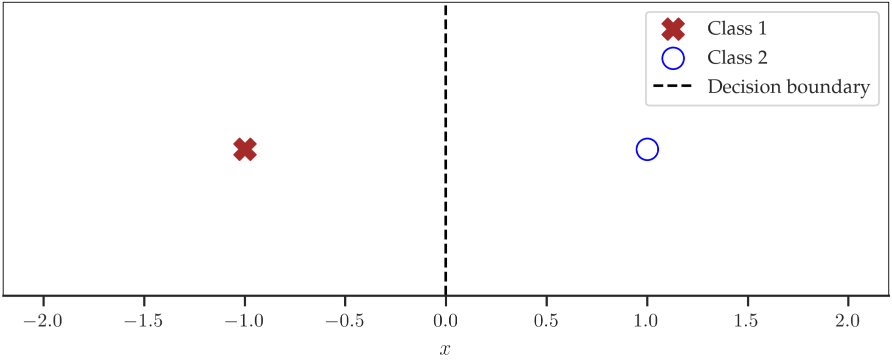


https://www.nucleusbox.com/an-intuition-behind-gradient-descent-using-python/

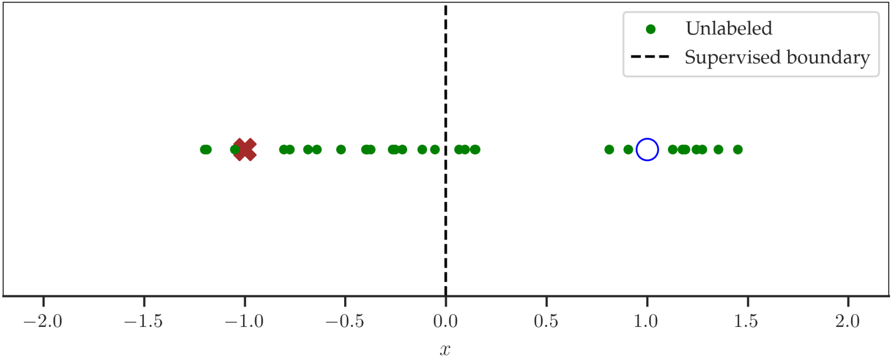
How is SSL possible?

$$f(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(h(\mathbf{x}_i; \boldsymbol{\theta}), y_i)$$

- How can we use unlabelled data? The ERM needs labels

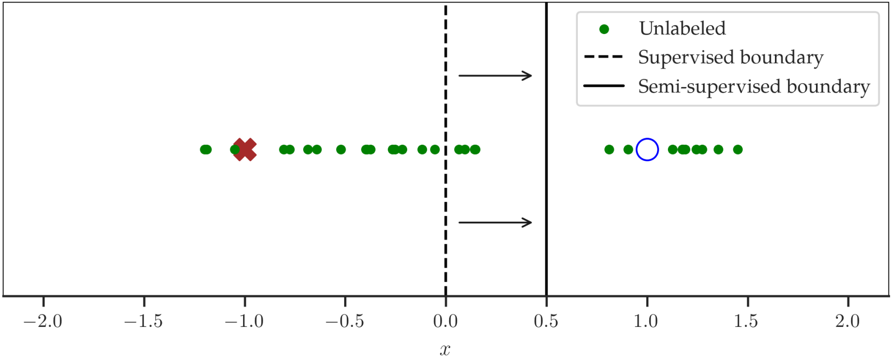


How is SSL possible?



- Where should we move the decision boundary?

How is SSL possible?



- We need to assume a link between **y** and **x**

SSL assumptions[1]

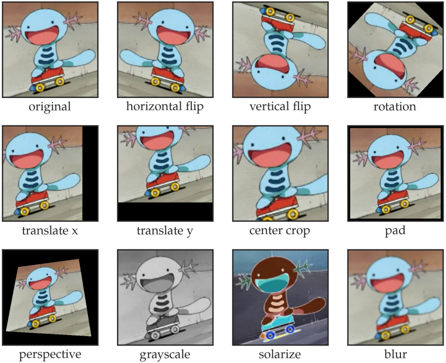
- **Smoothness**:
  - Examples with similar features should have similar labels
- **Low density separation**:
  - Decision boundaries should lie in low density regions
- Strategy: introduce a loss that penalises failing the assumptions
  - Consistency regularisation
  - Entropy minimisation
- New ERM problem encourages model output to match assumptions

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sup}} + \lambda_u \mathcal{L}_{\text{unsup}}$$

[1] Chapelle, Schölkopf, and Zien 2006.

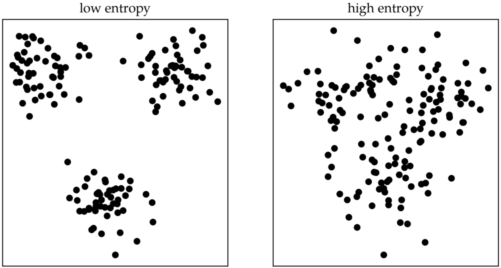
Consistency regularisation

- Predictions made on similar examples should have the same label
- In practice, we generate similar examples with data augmentation
- Let $\mathbf{x}$ be unlabelled, α, β are augmentations. $h(\alpha(\mathbf{x})) \approx h(\beta(\mathbf{x}))$



Entropy minimisation[2]

- The class conditional entropy is a measure of class overlap



- Introduce a loss that penalises weak separation of classes

[2] Grandvalet and Bengio 2004.

FixMatch algorithm

- An simple algorithm that leverages pseudo-labelling and data augmentation[3]
- Supervised loss - standard cross entropy on labelled examples

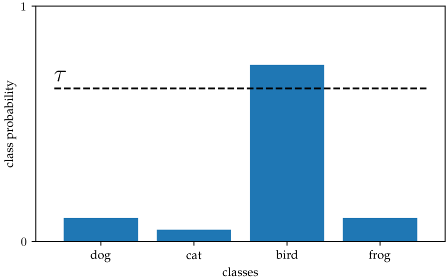
$$\mathcal{L}_{\text{sup}} \leftarrow \frac{1}{r} \sum_{i=1}^{r} \text{CE}((h(\mathbf{x}_i); \theta), y_i)$$

- Unsupervised loss
    - Consistency regularisation by data augmentation
    - Entropy minimisation with pseudo-labelling

[3] Sohn et al. 2020.

Pseudo-labelling[4]

- If we don't have the label - make an artificial one
- Use a model that gives a probability distribution over the classes.
- Select high confidence predictions ($> \tau$) as pseudo labels

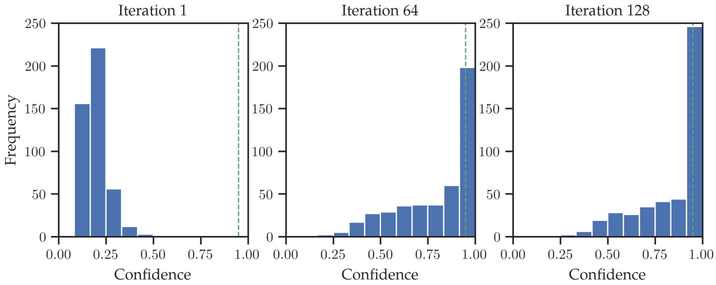


$$\mathbf{y}_{\text{pseudo}} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

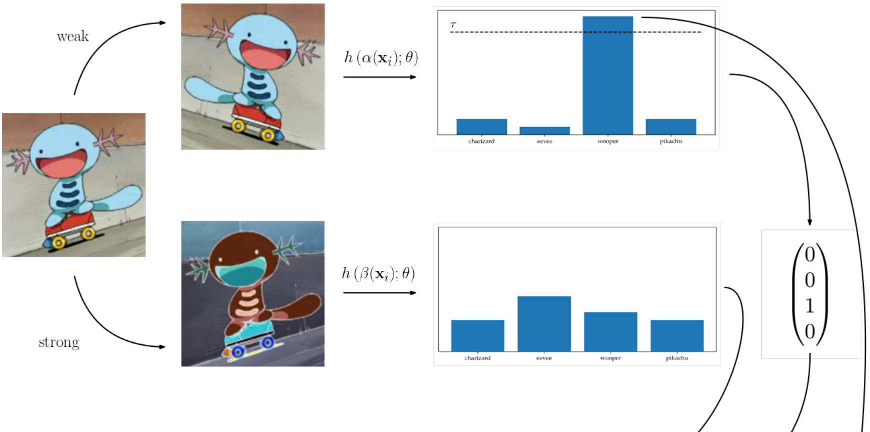
[4] Lee 2025.

Pseudo-labelling

- Equivalent to entropy minimisation
- The confidence threshold is important to reduce confirmation bias



FixMatch: Consistency regularisation



$$\mathcal{L}_{\text{unsup}} = \frac{1}{n-r} \sum_{i=r+1}^{n-r} \text{CE}((h(\ \beta(\mathbf{x}_i)\ ;\ \theta),\ y_{\text{pseudo}}) \cdot \mathbb{I}\left[p_i > \tau\right]$$

FixMatch

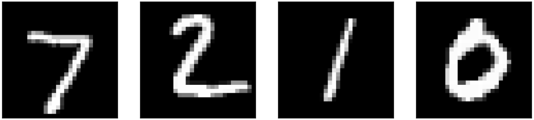
Algorithm 5 FixMatch forward step [18]

Require: Model $h(\cdot;\theta)$
Labelled minibatch $\mathcal{B}_{\mathrm{s}} = \{(\mathbf{x}_i, \mathbf{y}_i) : i = 1, 1, \ldots, B_s\}$
Unlabelled minibatch $\mathcal{B}_{\mathrm{u}} = \{\mathbf{u}_i : i = 1, \ldots, B_u\}$
Confidence threshold τ
Weak augmentation $\alpha(\cdot)$, strong augmentation $\beta(\cdot)$

$\mathcal{L}_{\text{sup}} \leftarrow \frac{1}{B_s}\sum_{i=1}^{B_s} \mathrm{CE}(h(\mathbf{x}_i);\theta), y_i)$ ▷ *supervised loss on labelled examples*
for all $i \in \{1, \ldots, B_u\}$ **do**
  $z_i \leftarrow h(\alpha(\mathbf{u}_i);\theta)$ ▷ *model output on weakly augmented input*
  $\hat{y}_i \leftarrow \arg\max\{\sigma(z_i)\}$ ▷ *pseudo-label for weakly augmented input*
  $p_i \leftarrow \max\{\sigma(z_i)\}$ ▷ *confidence in pseudo-label*
▷ *Supervised loss on strongly augmented unlabelled examples with high confidence pseudo-labels* ◁
$\mathcal{L}_{\text{unsup}} \leftarrow \frac{1}{B_u}\sum_{i=1}^{B_u} \mathbb{1}\left[p_i > \tau\right] \cdot \mathrm{CE}\left(h(\beta(\mathbf{u}_i);\theta), \hat{y}_i\right)$
$\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{sup}} + \lambda_u \mathcal{L}_{\text{unsup}}$
return $\mathcal{L}_{\text{total}}$

Experiments

- MNIST handwritten digits dataset
- 60000 training examples, 10000 test examples
- SSL dataset simulated by withholding most labels



Experiments

Labels per class	1	4	10
Supervised	0.201±0.010	0.418±0.039	0.731±0.075
FixMatch	0.620±0.078	0.934±0.049	0.969±0.001
FullySupervised		0.993±0.001	

Table: Average prediction accuracy on test set (mean±s.d over 3 seeds)

- `https://github.com/nhamid289/sslpack`

References I

Chapelle, Olivier, Bernhard Schölkopf, and Alexander Zien, eds. (2006). *Semi-supervised learning*. en. Cambridge, Mass.

Grandvalet, Yves and Yoshua Bengio (2004). *Semi-supervised Learning by Entropy Minimization*. (Visited on 06/02/2025).

Lee, Dong-Hyun (2025). *Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks*. en. (Visited on 06/02/2025).

Sohn, Kihyuk et al. (2020). "FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence". In: *Advances in Neural Information Processing Systems*. (Visited on 08/10/2025).