

Apprentissage semi-supervisé

A. Carlier

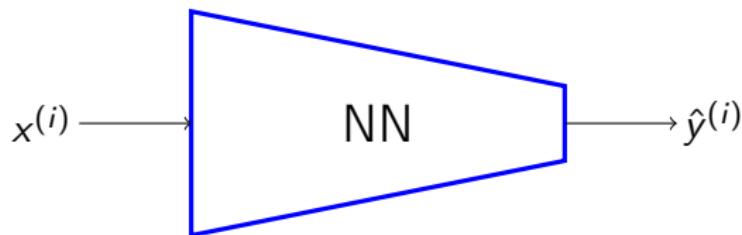
2022

Apprentissage supervisé

Dans le cadre de l'**apprentissage supervisé**, on dispose d'observations et de leurs étiquettes (appelées encore cibles (*target*), catégories ou *labels*) qui constituent un ensemble d'apprentissage. On le note :

$$\mathcal{D} = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}.$$

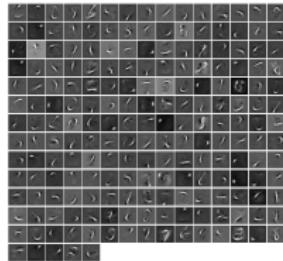
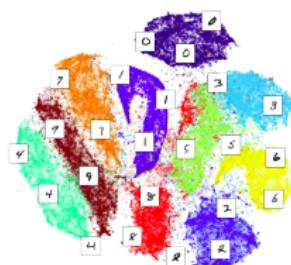
Les labels permettent d'enseigner à l'algorithme à établir des correspondances entre les observations et les labels.



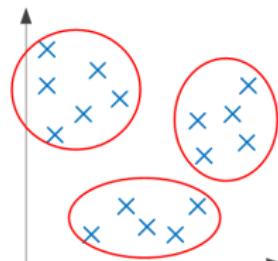
Apprentissage non-supervisé

Dans le cadre de l'**apprentissage non supervisé**, on dispose uniquement d'observations

$$\mathcal{D} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}.$$

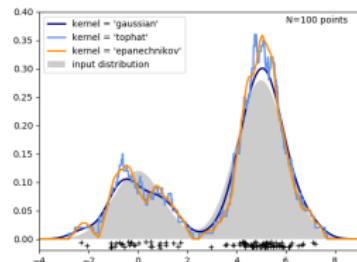


Réduction de dimension



Clustering

Extraction de caractéristiques



Estimation de densité

Apprentissage semi-supervisé

Dans le cadre de l'**apprentissage semi-supervisé**, on dispose de deux ensembles d'observations. Un ensemble labellisé :

$$\mathcal{L} = \{(x_l^{(1)}, y^{(1)}), \dots, (x_l^{(m)}, y^{(m)})\}.$$

et un ensemble constitué uniquement d'observations non labellisées :

$$\mathcal{U} = \{x_u^{(1)}, \dots, x_u^{(n)}\}.$$

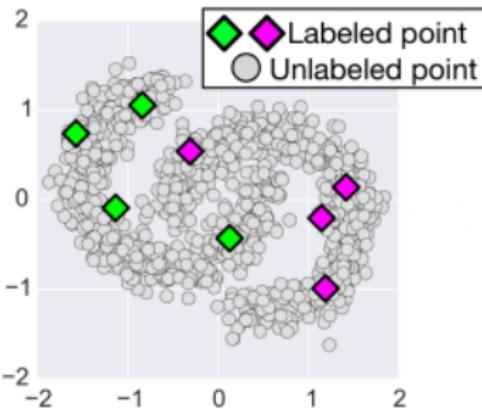
Apprentissage semi-supervisé

On peut différencier deux grandes familles de cas où l'apprentissage semi-supervisé est pertinent :

- $|\mathcal{L}|$ est petit : on dispose d'un ensemble limité d'annotations (car trop cher/rare/complexe) mais d'un grand nombre d'observations. C'est un **cas courant dans l'industrie**.
- $|\mathcal{L}|$ est grand, et $|\mathcal{U}|$ est encore plus grand : les données non labellisées amènent une information supplémentaire qui peut par exemple permettre d'entraîner des modèles de plus grande capacité. Ce type de méthodes permet d'obtenir **des performances quasiment à l'état de l'art** sur les bases de données de *benchmarks* (cf. Meta Pseudo Labels, présenté un peu plus tard).

Apprentissage semi-supervisé

Quel est l'intérêt de disposer de données non labellisées ?



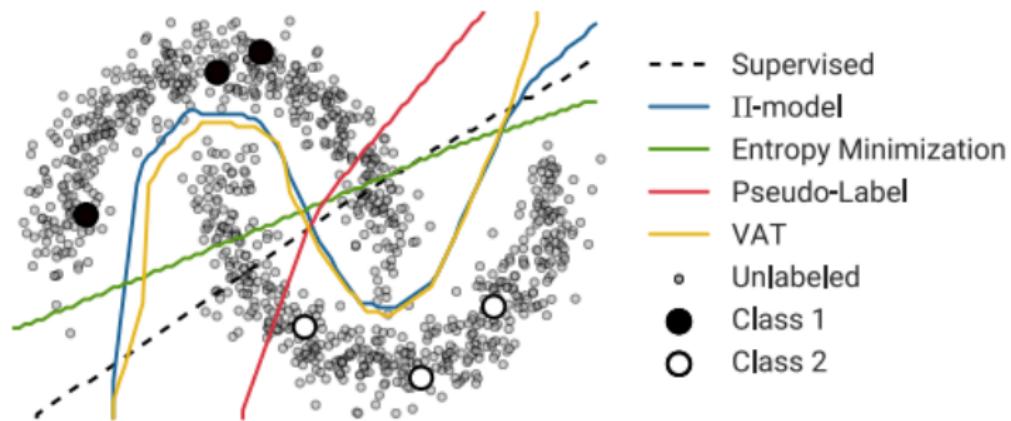
"L'hypothèse cluster" (*cluster assumption*) postule que les frontières de décision (inter-classes) doivent se situer dans des régions où la densité de données est faible.

Image de [Miyati et al.] Virtual Adversarial Training : A Regularization Method for Supervised and Semi-Supervised Learning

[Chapelle et al.] Semi-Supervised Classification by Low Density Separation.

Apprentissage semi-supervisé

Quel est l'intérêt de disposer de données non labellisées ?



Les données non labellisées fournissent des informations sur la densité des échantillons.

Image de [Ovital et al.] Realistic Evaluation of Deep Semi-Supervised Learning Algorithms

Quelques idées clés (en guise d'aperçu)

- Minimisation d'entropie et pseudo-labels
- Algorithmes *Teacher-Student*
- Coût de cohérence
- Augmentation de données

Une première idée : entropie et confiance

L'entropie H de la distribution de probabilité prédite par le réseau (on suppose ici que l'on s'intéresse à un problème de classification) pour une donnée x est une mesure de la confiance du modèle dans sa prédiction \hat{y} .

$$H(\hat{y}) = - \sum_{i=1}^C \hat{y}_i \log(\hat{y}_i)$$

Par exemple, pour un problème de classification à 4 classes :

- Si $\hat{y} = [\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}]$, $H(\hat{y}) \approx 1.38$ et le modèle n'a aucune confiance dans sa prédiction.
- Si $\hat{y} = [0.997, 0.001, 0.001, 0.001]$, $H(\hat{y}) \approx 0.02$ et le modèle est très confiant dans sa prédiction.

Une première idée : entropie et confiance

On peut donc construire une fonction objectif semi-supervisée de la façon suivante :

$$J = \sum_{(x,y) \in \mathcal{L}} CE(y, \hat{y}) + \lambda \sum_{x \in \mathcal{U}} H(\hat{y})$$

où \hat{y} est la prédiction du réseau associée à la donnée d'entrée x , CE désigne l'entropie croisée, H l'entropie et λ est un hyperparamètre contrôlant la régularisation.

Le processus d'optimisation minimise ainsi conjointement l'entropie croisée, qui utilise les données supervisées dont on dispose, et l'entropie des données non labellisées ce qui déplace les frontières de décision dans des zones de plus faible densité.

[Grandvalet et al.] Semi-supervised Learning by Entropy Minimization .

Minimisation d'entropie : Illustration

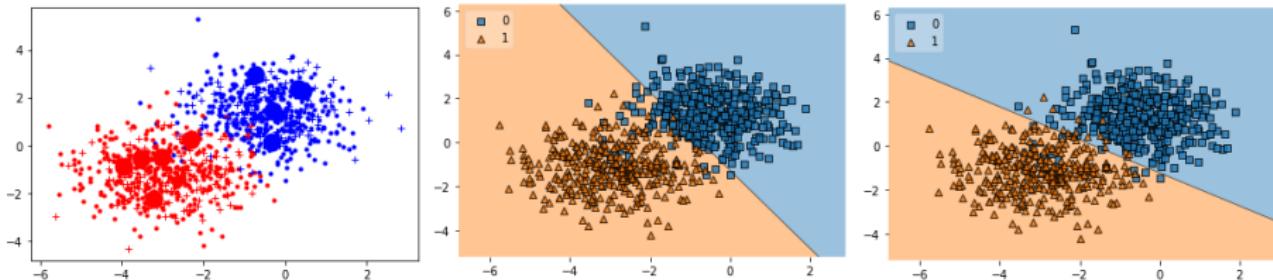


Illustration vue en TP :

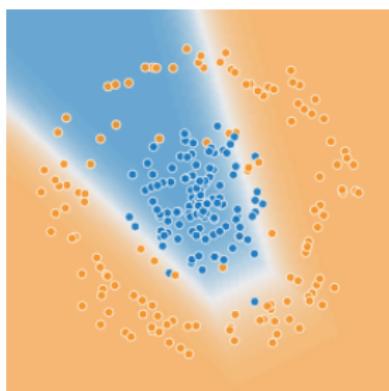
- à gauche : données. Les gros points sont les données labellisées, les petits points les données non labellisées, et les croix sont les données de test.
- au milieu : frontière de décision à l'issue d'un apprentissage supervisé uniquement par les données labellisées.
- à droite : frontière de décision à l'issue d'un apprentissage semi-supervisé par minimisation d'entropie.

[Grandvalet et al.] Semi-supervised Learning by Entropy Minimization .

Rappel : sur-apprentissage et régularisation

On parle de **sous-apprentissage** (*underfitting*) lorsque le modèle appris explique trop mal l'ensemble d'apprentissage.

On parle de **sur-apprentissage** (*overfitting*) lorsque le modèle appris explique à l'inverse trop bien l'ensemble d'apprentissage ; ce modèle se généralise alors mal à la population cible.

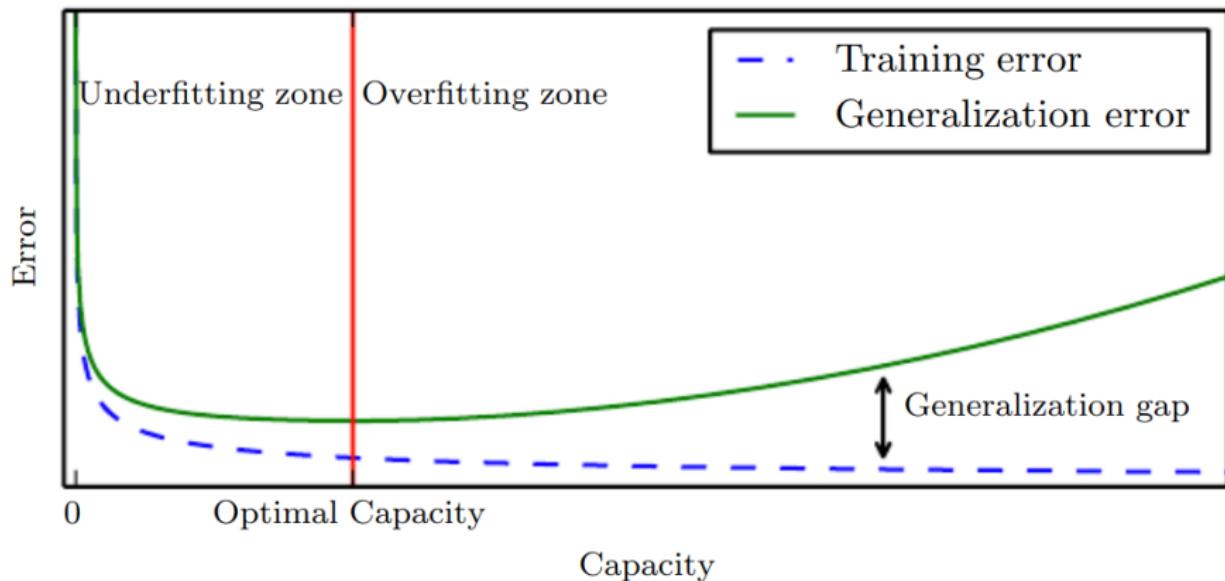


Sous-apprentissage



Sur-apprentissage

Rappel : sur-apprentissage et régularisation



Un modèle de trop grande capacité (profondeur, nombre de neurones) engendre du sur-apprentissage.

Image de [Goodfellow et al. 2015] Deep Learning

Rappel : sur-apprentissage et régularisation

Ajout d'une contrainte sur les paramètres du réseau :

- Régularisation \mathcal{L}^2 ou **Ridge** maintient les coefficients du modèle aussi petits que possible :

$$J(\theta) = \text{RisqueEmpirique}(\theta) + \lambda \frac{1}{2} \sum_{i=1}^m \theta_i^2$$

où λ contrôle la qualité de régularisation souhaitée

- Régularisation \mathcal{L}^1 ou **Lasso** : tend à éliminer complètement les poids des variables les moins importantes (\Rightarrow produit un modèle creux) :

$$J(\theta) = \text{RisqueEmpirique}(\theta) + \lambda \sum_{i=1}^m |\theta_i|$$

<https://playground.tensorflow.org/>

[Krogh, Hertz 1992] A simple weight decay can improve generalization

Apprentissage semi-supervisé et régularisation

En apprentissage semi-supervisé, on dispose d'un ensemble labellisé \mathcal{L} de taille souvent trop limitée : on observe donc un sur-apprentissage de notre modèle.

L'idée-clé est d'utiliser les données non labellisées via un terme de régularisation bien choisi (souvent basé sur la *cluster assumption*), qui aide à limiter le sur-apprentissage.

$$J = \sum_{(x,y) \in \mathcal{L}} CE(y, \hat{y}) + \lambda \sum_{x \in \mathcal{U}} H(\hat{y})$$

Pseudo-labelisation

Pendant l'entraînement, on fait prédire au réseau les données non supervisées $x \in \mathcal{U}$ et on crée à partir de ces prédictions \hat{y} des pseudo-labels y' :

$$y'_i = \begin{cases} 1 & \text{si } i = \underset{j}{\operatorname{argmax}} \hat{y}_j \\ 0 & \text{sinon.} \end{cases}$$

On minimise ensuite la fonction objectif suivante :

$$J = \sum_{(x,y) \in \mathcal{L}} CE(y, \hat{y}) + \lambda \sum_{x \in \mathcal{U}} CE(y', \hat{y})$$

[Lee] Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks

Pseudo-labellisation

La pseudo-labellisation est en fait équivalente à la minimisation de l'entropie des données non supervisées !

En effet, minimiser l'entropie croisée entre prédiction et pseudo-labels revient à renforcer la confiance de la prédiction du réseau, ce qui est le même objectif que la minimisation de l'entropie de prédiction.

[Lee] Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks

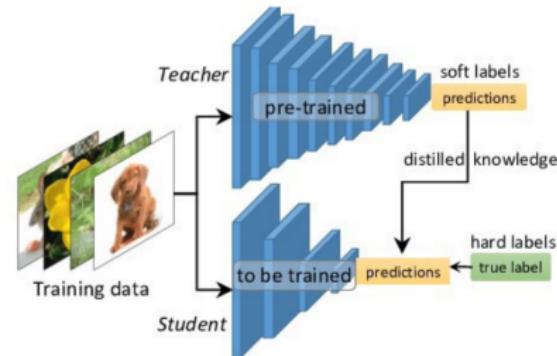
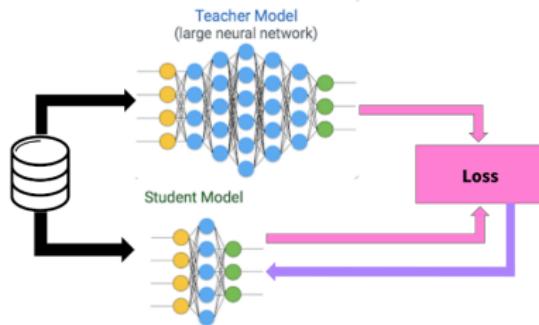
Pseudo-labelisation en pratique

$$J = \sum_{(x,y) \in \mathcal{L}} CE(y, \hat{y}) + \lambda(t) \sum_{x \in \mathcal{U}} CE(y', \hat{y})$$

- Le taux de régularisation varie au cours du temps : il est initialement nul et augmente ensuite, renforçant progressivement la prise en compte des données non labellisées.
- On utilise un nombre d'échantillons différent pour les batches supervisés (ici, 32) et non-supervisés (256).

[Lee] Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks

La distillation

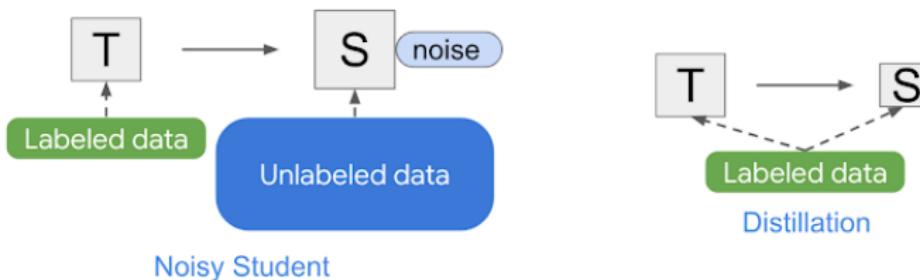


La distillation est un processus utile pour la mise en production de réseaux de neurones qui consiste à entraîner un réseau étudiant (*Student*), de capacité faible, à partir d'un réseau enseignant (*Teacher*) de capacité plus forte.

Image de <https://towardsdatascience.com/knowledge-distillation-simplified-dd4973dbc764>

Le modèle Teacher-Student pour l'apprentissage semi-supervisé

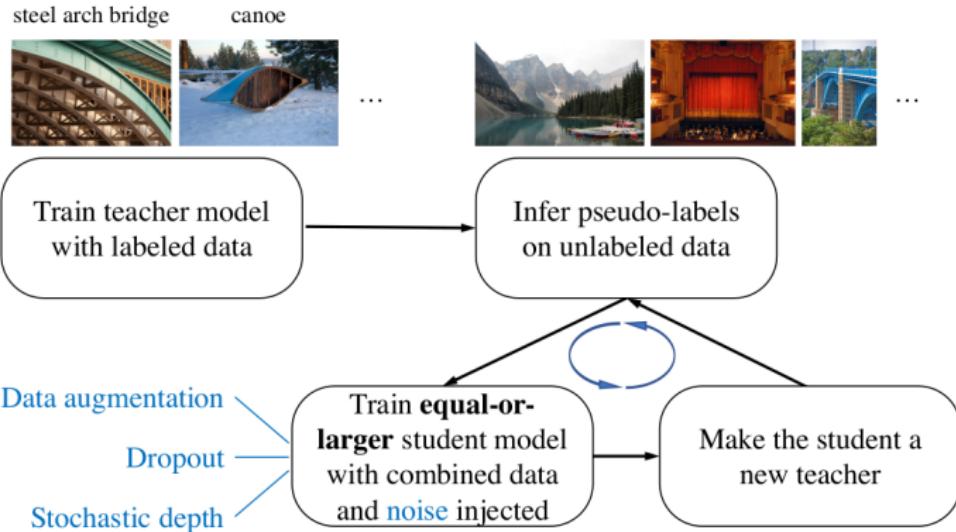
Une variante de la distillation peut être mise en place pour l'apprentissage semi-supervisé :



Deux différences fondamentales :

- La capacité de l'étudiant doit être égale, voire supérieure, à la capacité de l'enseignant.
- Pour l'entraînement de l'étudiant, on bruite les données (augmentation) et le réseau (*dropout, stochastic depth*).

Noisy Student



Après plusieurs itérations, on obtient une précision top-1 de 88.4 % sur ImageNet avec un réseau EfficientNet-L2 (480M de paramètres), l'état de l'art lors de la sortie de cet article.

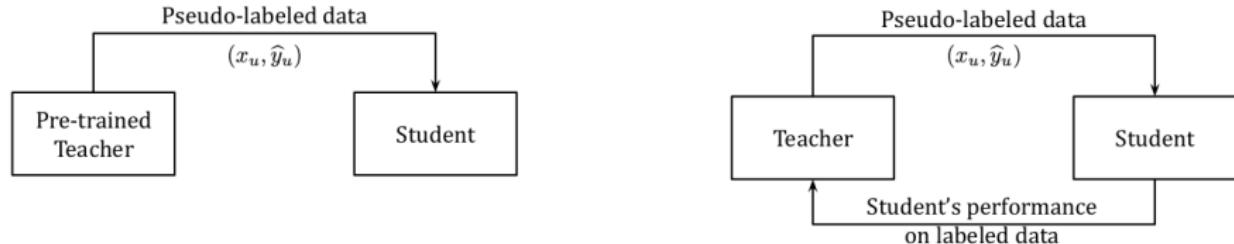
[Xie et al.] Self-training with Noisy Student improves ImageNet classification

Pourquoi bruiter les données ?

Dans *Noisy Student*, les pseudo-labels sont inférés sur les données non-labellisées, et on entraîne le *Student* à reproduire ce pseudo-label sur ces mêmes données altérées (augmentées). Ceci encourage le réseau à produire la même sortie pour des entrées similaires légèrement perturbées, ce qui accroît sa stabilité.

Cette idée est également utilisée dans de nombreux autres travaux et souvent formalisée sous la forme d'une fonction de perte de "cohérence" (*consistency loss*).

Meta Pseudo-Labels



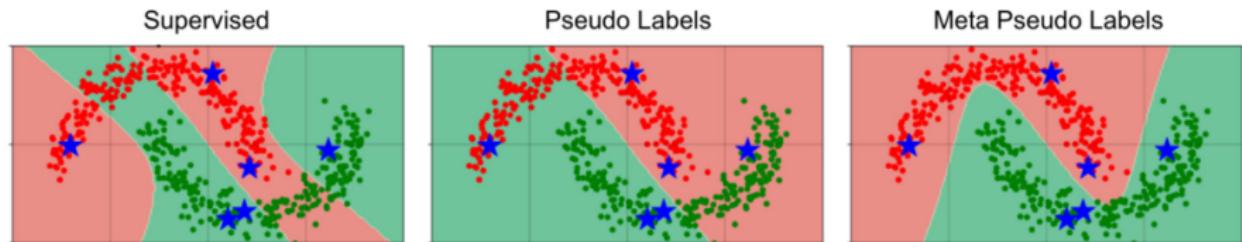
On peut co-entraîner les réseaux *Teacher* (T) et *Student* (S) :

- S minimise l'erreur entre ses prédictions et les pseudo-labels générés par T, sur les données non labellisées.
- T cherche à générer des pseudo-labels qui maximisent la performance de S sur les données labellisées.

Cette procédure, inspirée du *Meta-learning*, permet d'éviter le biais de renforcement des erreurs du *Teacher*.

[Pham et al.] Meta Pseudo Labels

Meta Pseudo-Labels

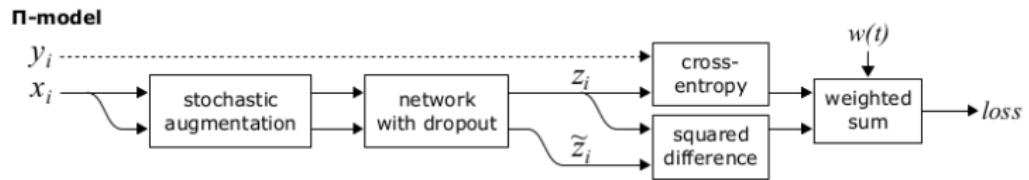


Cette procédure, inspirée du *Meta-learning*, permet d'éviter le biais de renforcement des erreurs du *Teacher*.

Gra  e   cette technique, le cap des 90% de pr  ision Top-1 sur ImageNet a  t    p  s   pour la premi  re fois en 2021.

[Pham et al.] Meta Pseudo Labels

Π -modèle



Pour des données $(x, y) \in B$, où B désigne le mini-batch courant :

$$J = \sum_{(x,y) \in \mathcal{L} \cap B} CE(y, \hat{y}) + \lambda(t) \sum_{x \in B} \|\tilde{\hat{y}} - \hat{y}\|^2$$

La fonction de coût combine un terme supervisé (une entropie classique) et un terme non-supervisé qui assure la robustesse au bruit (e.g. augmentation de données) de la prédiction du modèle.

[Laine et al.] Temporal Ensembling for Semi-Supervised Learning

Π -modèle

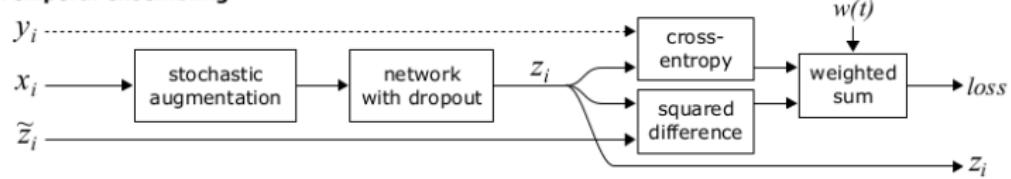
Algorithm 1 Π -model pseudocode.

```
Require:  $x_i$  = training stimuli
Require:  $L$  = set of training input indices with known labels
Require:  $y_i$  = labels for labeled inputs  $i \in L$ 
Require:  $w(t)$  = unsupervised weight ramp-up function
Require:  $f_\theta(x)$  = stochastic neural network with trainable parameters  $\theta$ 
Require:  $g(x)$  = stochastic input augmentation function
for  $t$  in  $[1, num\_epochs]$  do
    for each minibatch  $B$  do
         $z_{i \in B} \leftarrow f_\theta(g(x_{i \in B}))$                                 ▷ evaluate network outputs for augmented inputs
         $\tilde{z}_{i \in B} \leftarrow f_\theta(g(x_{i \in B}))$                             ▷ again, with different dropout and augmentation
         $loss \leftarrow -\frac{1}{|B|} \sum_{i \in (B \cap L)} \log z_i[y_i]$           ▷ supervised loss component
         $+ w(t) \frac{1}{C|B|} \sum_{i \in B} \|z_i - \tilde{z}_i\|^2$                   ▷ unsupervised loss component
        update  $\theta$  using, e.g., ADAM                                         ▷ update network parameters
    end for
end for
return  $\theta$ 
```

[Laine et al.] Temporal Ensembling for Semi-Supervised Learning

Temporal ensembling

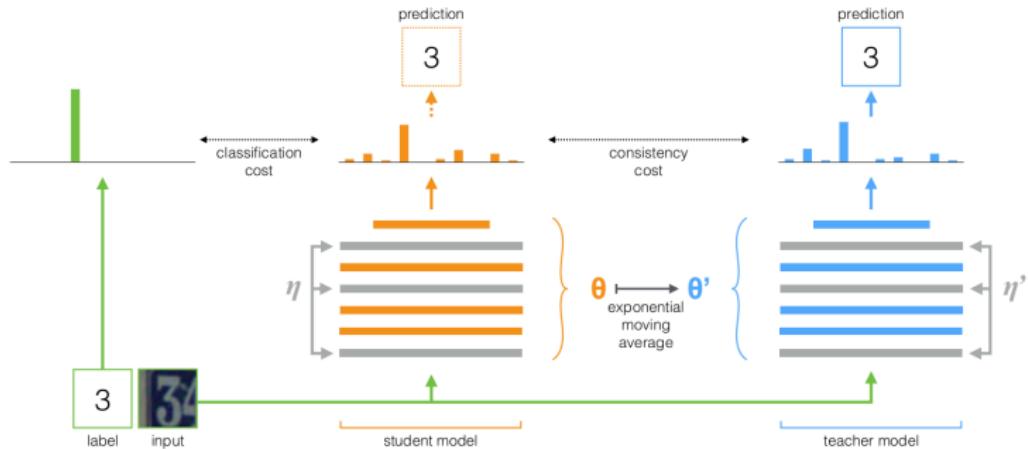
Temporal ensembling



On peut stabiliser les prédictions en les moyennant sur plusieurs *epochs* successives. (*temporal ensembling*)

[Laine et al.] Temporal Ensembling for Semi-Supervised Learning

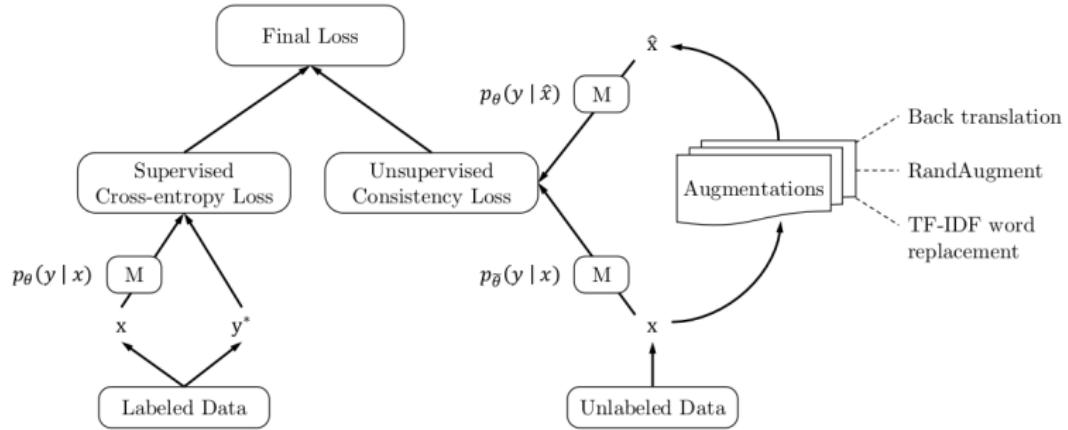
Mean-Teachers



On peut aussi, plutôt que de moyenner les prédictions des réseaux enseignants, directement moyenner les réseaux enseignants !
(*Mean-Teacher*)

[Tarvainen et al.] Mean teachers are better role models : Weight-averaged consistency targets improve semi-supervised deep learning results

Unsupervised Data Augmentation (UDA)



Idée similaire au *Pi*-modèle, mais étude systématique d'augmentations plus à l'état de l'art (cf. fin du cours).

[Xie et al.] Unsupervised Data Augmentation for Consistency Training

Unsupervised Data Augmentation (UDA)

Augmentation (# Sup examples)	Sup (50k)	Semi-Sup (4k)
Crop & flip	5.36	10.94
Cutout	4.42	5.43
RandAugment	4.23	4.32

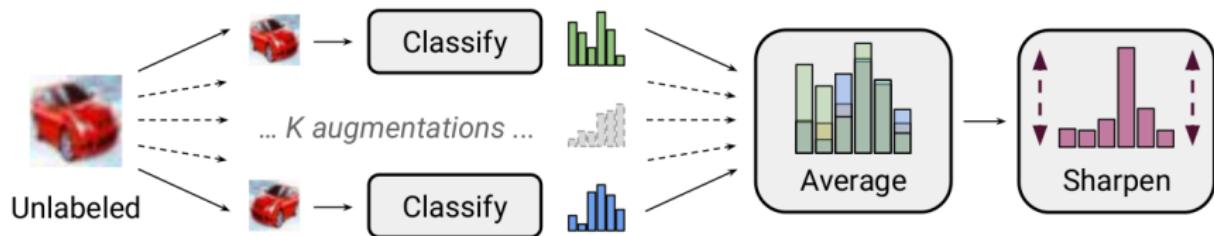
Table 1: Error rates on CIFAR-10.

Une conclusion importante de l'étude est que les résultats de l'approche semi-supervisée dépendent énormément de l'augmentation de données qui a été utilisée.

La qualité de l'augmentation dans un contexte supervisé est directement corrélée avec les performances obtenues avec cette même augmentation dans un contexte semi-supervisé.

[Xie et al.] Unsupervised Data Augmentation for Consistency Training

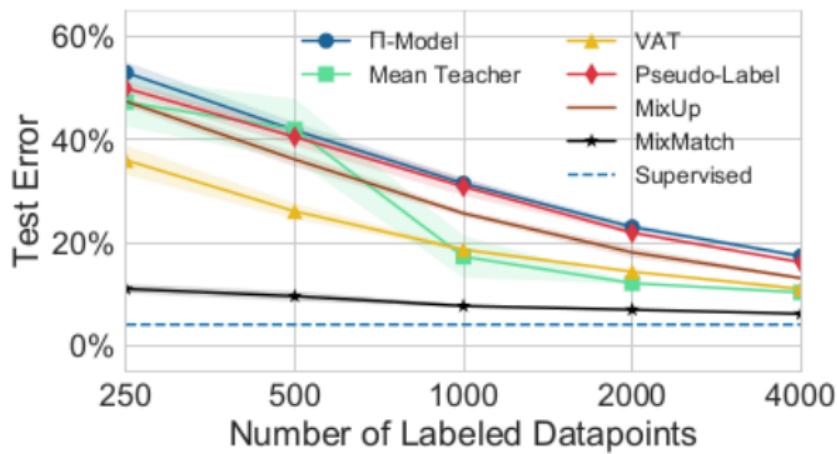
MixMatch



Dans MixMatch, on élabore un pseudo-label pour les données non-supervisées (cf. schéma) et on considère ensuite toutes les données comme labellisées. Lors de l'entraînement, on utilise la technique d'augmentation de données MixUp pour mélanger indifféremment données labellisées et non labellisées.

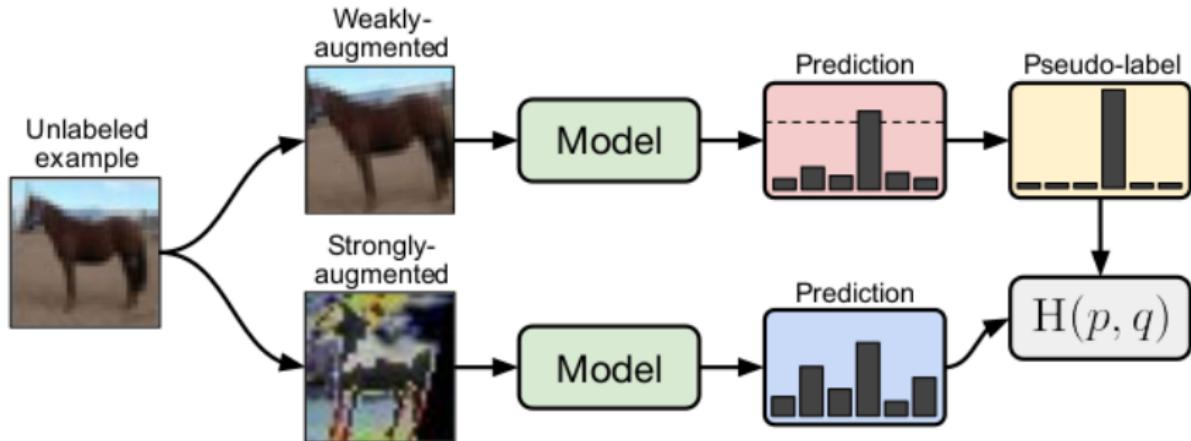
[Berthelot et al.] MixMatch : A Holistic Approach to Semi-Supervised Learning

MixMatch



[Berthelot et al.] MixMatch : A Holistic Approach to Semi-Supervised Learning

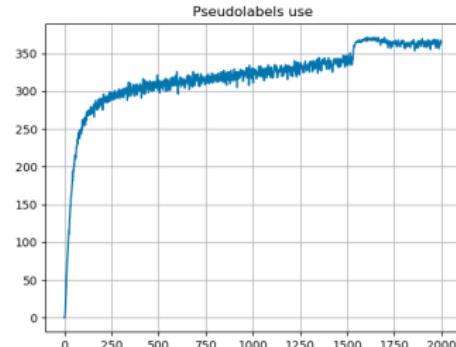
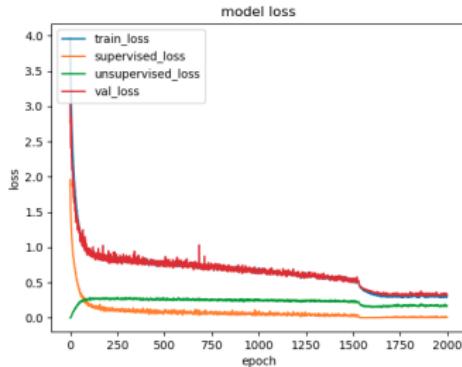
FixMatch



Dans FixMatch, les pseudo-labels sont uniquement les prédictions qui atteignent une certaine valeur de confiance (au-dessus d'un seuil de 0.95). Une différence est faite entre augmentation "faible" et "forte".

[Sohn et al.] FixMatch : Simplifying Semi-Supervised Learning with Consistency and Confidence

FixMatch



Le taux de régularisation n'est pas évolutif comme pour la majorité des méthodes vues précédemment ; la perte non-supervisée (de cohérence) augmente progressivement à mesure que la confiance du modèle augmente.

[Sohn et al.] FixMatch : Simplifying Semi-Supervised Learning with Consistency and Confidence

Method	CIFAR-10			CIFAR-100			SVHN		
	40 labels	250 labels	4000 labels	400 labels	2500 labels	10000 labels	40 labels	250 labels	1000 labels
IL-Model	-	54.26±3.97	14.01±0.38	-	57.25±0.48	37.88±0.11	-	18.96±1.92	7.54±0.36
Pseudo-Labeling	-	49.78±0.43	16.09±0.28	-	57.38±0.46	36.21±0.19	-	20.21±1.09	9.94±0.61
Mean Teacher	-	32.32±2.30	9.19±0.19	-	53.91±0.57	35.83±0.24	-	3.57±0.11	3.42±0.07
MixMatch	47.54±11.50	11.05±0.86	6.42±0.10	67.61±1.32	39.94±0.37	28.31±0.33	42.55±14.53	3.98±0.23	3.50±0.28
UDA	29.05±5.93	8.82±1.08	4.88±0.18	59.28±0.88	33.13±0.22	24.50±0.25	52.63±20.51	5.69±2.76	2.46±0.24
ReMixMatch	19.10±9.64	5.44±0.05	4.72±0.13	44.28±2.06	27.43±0.31	23.03±0.56	3.34±0.20	2.92±0.48	2.65±0.08
FixMatch (RA)	13.81±3.37	5.07±0.65	4.26±0.05	48.85±1.75	28.29±0.11	22.60±0.12	3.96±2.17	2.48±0.38	2.28±0.11
FixMatch (CTA)	11.39±3.35	5.07±0.33	4.31±0.15	49.95±3.01	28.64±0.24	23.18±0.11	7.65±7.65	2.64±0.64	2.36±0.19

FixMatch obtient d'excellentes performances même avec peu de données supervisées.

[Sohn et al.] FixMatch : Simplifying Semi-Supervised Learning with Consistency and Confidence

FixMatch



FixMatch atteint les 78 % de bonne classification en utilisant uniquement les labels associés à ces 10 images.

Le choix des données labellisées reste un point peu discuté dans ces articles.

[Sohn et al.] FixMatch : Simplifying Semi-Supervised Learning with Consistency and Confidence

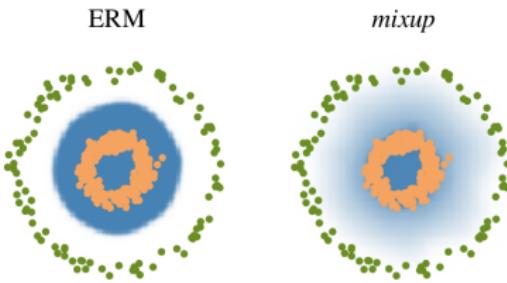
Quelques mots sur l'augmentation de données



...en plus des classiques translations, rotations, crop, etc.

MixUp

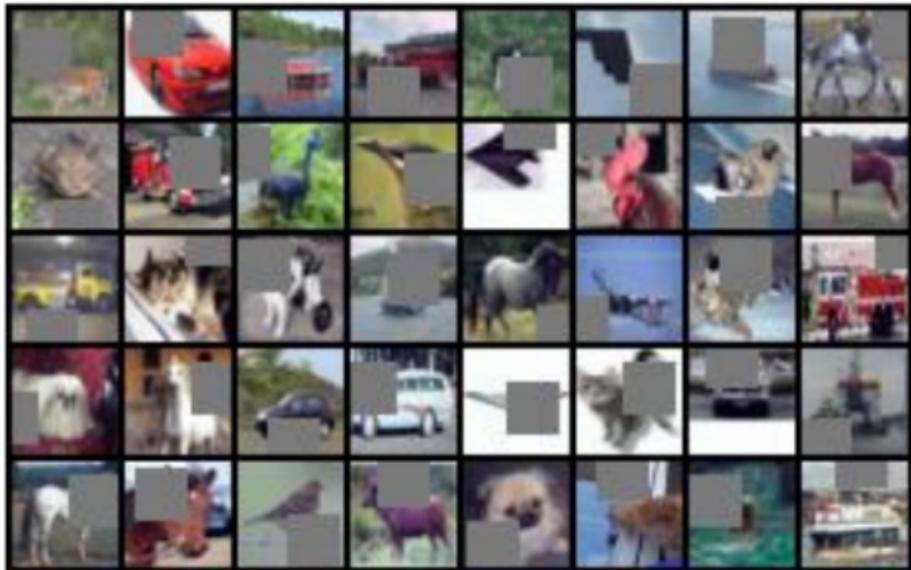
$$\begin{aligned}\tilde{x} &= \lambda x_i + (1 - \lambda)x_j, \\ \tilde{y} &= \lambda y_i + (1 - \lambda)y_j,\end{aligned}$$



Création de données artificielles en mélangeant des données labellisées.

[Zhang et al.] MixUp : Beyond Empirical Risk Minimization

CutOut



Généralisation du *dropout* à des images d'entrée.

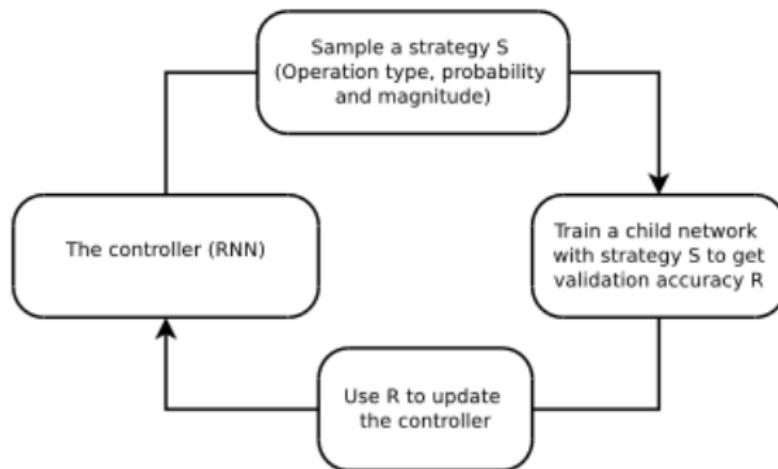
[DeVries et al.] Improved Regularization of Convolutional Neural Networks with Cutout

CutMix

	ResNet-50	Mixup [47]	Cutout [3]	CutMix
Image				
Label	Dog 1.0	Dog 0.5 Cat 0.5	Dog 1.0	Dog 0.6 Cat 0.4
ImageNet	76.3	77.4	77.1	78.6
Cls (%)	(+0.0)	(+1.1)	(+0.8)	(+2.3)
ImageNet	46.3	45.8	46.7	47.3
Loc (%)	(+0.0)	(-0.5)	(+0.4)	(+1.0)
Pascal VOC	75.6	73.9	75.1	76.7
Det (mAP)	(+0.0)	(-1.7)	(-0.5)	(+1.1)

[Yun et al.] CutMix : Regularization Strategy to Train Strong Classifiers with Localizable Features

RandAugment, AutoAugment, CTAugment

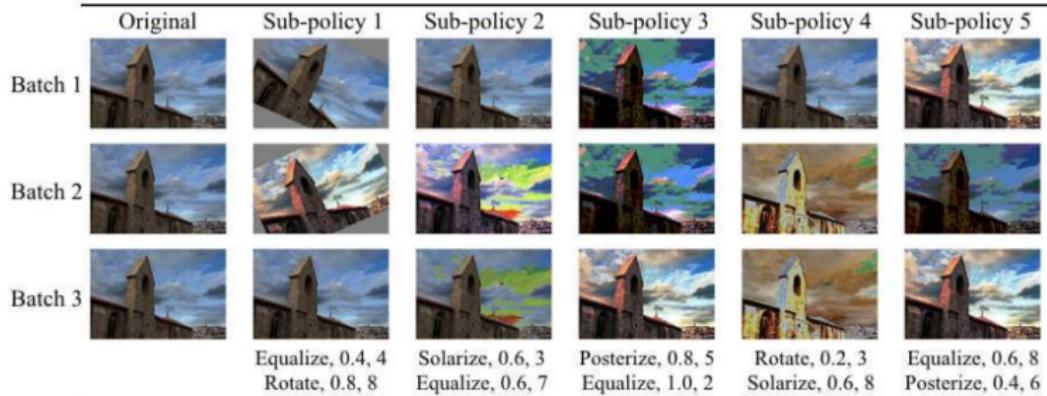


Recherche automatique (par renforcement) des meilleures augmentations.

[Cubuk et al.] AutoAugment : Learning Augmentation Strategies from Data

[Cubuk et al.] RandAugment : Practical automated data augmentation with a reduced search space

RandAugment, AutoAugment, CTAugment



Exemple d'une politique apprise par renforcement.

[Cubuk et al.] AutoAugment : Learning Augmentation Strategies from Data

[Cubuk et al.] RandAugment : Practical automated data augmentation with a reduced search space