# Algorithme semi-supervisé : Fixmatch

Fares Ernez
Baptiste Aussel
Albin Cintas

# Introduction

Plan

1) Présentation du jeu de données et de la méthode

2) Fonction pertes et optimiseurs

3) Comparaison des résultats

# Présentation du jeu de données

1. Dataset de 9975 images 64*64.

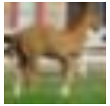
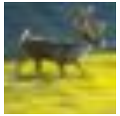
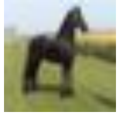
2. 475 images sont associées à un label, 9500 ne le sont pas.

3. Le dataset possède 95 type d’animaux différents
    a. ex : marmotte, macaque, fourmi , girafe, tortue, araigné, hippopotame, corcodile, ours brun, ours noir ...

4. 2850 images pour le test

# Présentation du jeu de données CIFAR-10

1. Dataset de 50000 images 32*32.

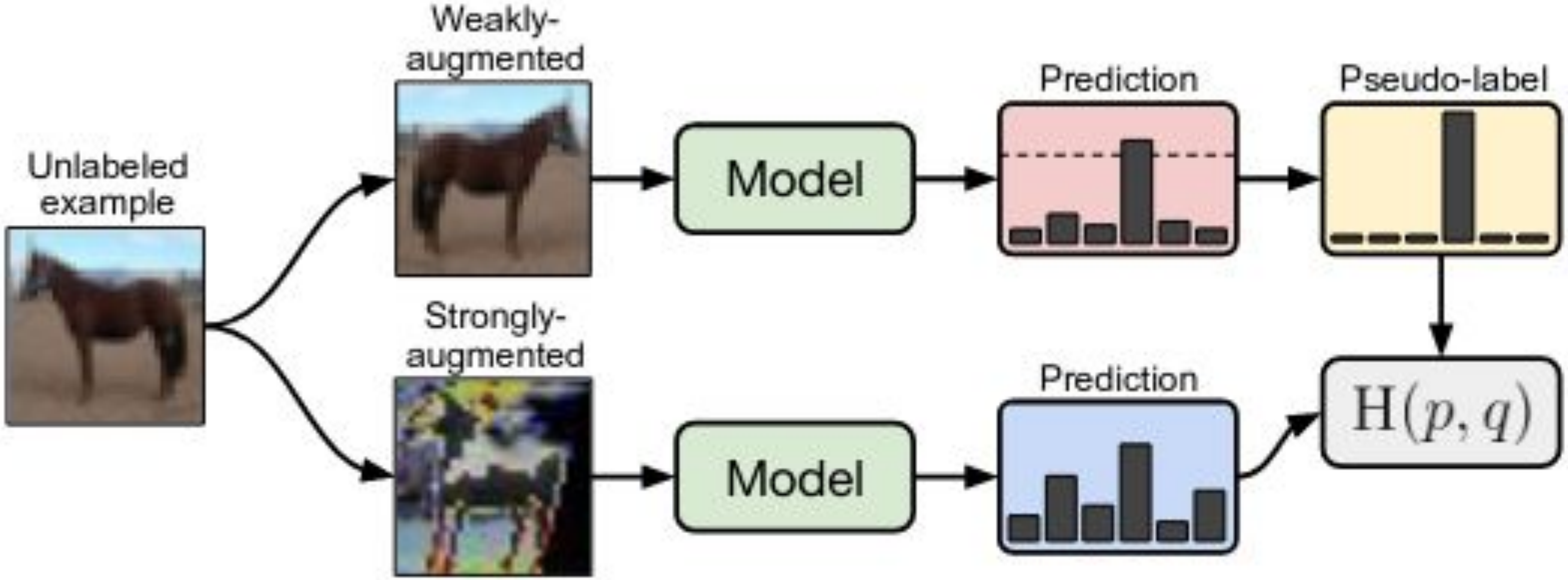
   

2. 2500 images sont associées à un label, 47500 ne le sont pas.

3. Le jeu CIFAR-10 possède 10 classes différentes
   a. avion, cerf, chien, chat, oiseau, cheval, camion, grenouille, voiture, bateau

4. 10000 images pour le test

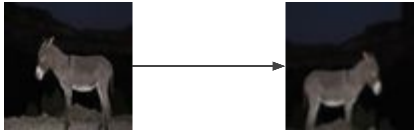
# Présentation de l’algorithme

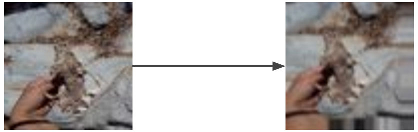


# Les augmentations de données : faible

Simple stratégie d’augmentation flip and shift .

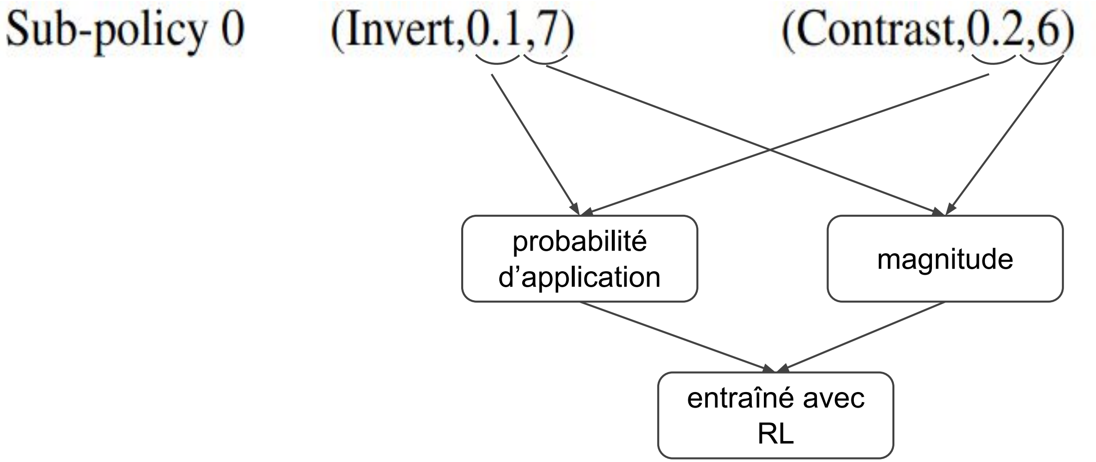
On retourne l’image avec une probabilité 0.5 et on décale l’image de 12.5 % verticalement et horizontalement





# Les augmentations de données : forte (AutoAugment) *

En amont : Définition de policies et sub-policies (paires)




| | Operation 1 | Operation 2 |
|---|---|---|
| Sub-policy 0 | (Invert,0.1,7) | (Contrast,0.2,6) |
| Sub-policy 1 | (Rotate,0.7,2) | (TranslateX,0.3,9) |
| Sub-policy 2 | (Sharpness,0.8,1) | (Sharpness,0.9,3) |
| Sub-policy 3 | (ShearY,0.5,8) | (TranslateY,0.7,9) |
| Sub-policy 4 | (AutoContrast,0.5,8) | (Equalize,0.9,2) |
| Sub-policy 5 | (ShearY,0.2,7) | (Posterize,0.3,7) |
| Sub-policy 6 | (Color,0.4,3) | (Brightness,0.6,7) |
| Sub-policy 7 | (Sharpness,0.3,9) | (Brightness,0.7,9) |
| Sub-policy 8 | (Equalize,0.6,5) | (Equalize,0.5,1) |
| Sub-policy 9 | (Contrast,0.6,7) | (Sharpness,0.6,5) |
| Sub-policy 10 | (Color,0.7,7) | (TranslateX,0.5,8) |
| Sub-policy 11 | (Equalize,0.3,7) | (AutoContrast,0.4,8) |
| Sub-policy 12 | (TranslateY,0.4,3) | (Sharpness,0.2,6) |
| Sub-policy 13 | (Brightness,0.9,6) | (Color,0.2,8) |
| Sub-policy 14 | (Solarize,0.5,2) | (Invert,0.0,3) |
| Sub-policy 15 | (Equalize,0.2,0) | (AutoContrast,0.6,0) |
| Sub-policy 16 | (Equalize,0.2,8) | (Equalize,0.6,4) |
| Sub-policy 17 | (Color,0.9,9) | (Equalize,0.6,6) |
| Sub-policy 18 | (AutoContrast,0.8,4) | (Solarize,0.2,8) |
| Sub-policy 19 | (Brightness,0.1,3) | (Color,0.7,0) |
| Sub-policy 20 | (Solarize,0.4,5) | (AutoContrast,0.9,3) |
| Sub-policy 21 | (TranslateY,0.9,9) | (TranslateY,0.7,9) |
| Sub-policy 22 | (AutoContrast,0.9,2) | (Solarize,0.8,3) |
| Sub-policy 23 | (Equalize,0.8,8) | (Invert,0.1,3) |
| Sub-policy 24 | (TranslateY,0.7,9) | (AutoContrast,0.9,1) |

Table 7. AutoAugment policy found on reduced CIFAR-10.

*:"AutoAugment : Learning Augmentation Strategies from Data". Ekin D. Cubuk , Barret Zoph, Dandelion Man é, Vijay Vasudevan

# Les augmentations de données : forte (AutoAugment) *

Résultats :

| | Operation 1 | Operation 2 |
|---|---|---|
| Sub-policy 0 | (Invert,0.1,7) | (Contrast,0.2,6) |
| Sub-policy 1 | (Rotate,0.7,2) | (TranslateX,0.3,9) |
| Sub-policy 2 | (Sharpness,0.8,1) | (Sharpness,0.9,3) |
| Sub-policy 3 | (ShearY,0.5,8) | (TranslateY,0.7,9) |
| Sub-policy 4 | (AutoContrast,0.5,8) | (Equalize,0.9,2) |
| Sub-policy 5 | (ShearY,0.2,7) | (Posterize,0.3,7) |
| Sub-policy 6 | (Color,0.4,3) | (Brightness,0.6,7) |
| Sub-policy 7 | (Sharpness,0.3,9) | (Brightness,0.7,9) |
| Sub-policy 8 | (Equalize,0.6,5) | (Equalize,0.5,1) |
| Sub-policy 9 | (Contrast,0.6,7) | (Sharpness,0.6,5) |
| Sub-policy 10 | (Color,0.7,7) | (TranslateX,0.5,8) |
| Sub-policy 11 | (Equalize,0.3,7) | (AutoContrast,0.4,8) |
| Sub-policy 12 | (TranslateY,0.4,3) | (Sharpness,0.2,6) |
| Sub-policy 13 | (Brightness,0.9,6) | (Color,0.2,8) |
| Sub-policy 14 | (Solarize,0.5,2) | (Invert,0.0,3) |
| Sub-policy 15 | (Equalize,0.2,0) | (AutoContrast,0.6,0) |
| Sub-policy 16 | (Equalize,0.2,8) | (Equalize,0.6,4) |
| Sub-policy 17 | (Color,0.9,9) | (Equalize,0.6,6) |
| Sub-policy 18 | (AutoContrast,0.8,4) | (Solarize,0.2,8) |
| Sub-policy 19 | (Brightness,0.1,3) | (Color,0.7,0) |
| Sub-policy 20 | (Solarize,0.4,5) | (AutoContrast,0.9,3) |
| Sub-policy 21 | (TranslateY,0.9,9) | (TranslateY,0.7,9) |
| Sub-policy 22 | (AutoContrast,0.9,2) | (Solarize,0.8,3) |
| Sub-policy 23 | (Equalize,0.8,8) | (Invert,0.1,3) |
| Sub-policy 24 | (TranslateY,0.7,9) | (AutoContrast,0.9,1) |

Table 7. AutoAugment policy found on reduced CIFAR-10.

Une sub-policy choisie aléatoirement

*:"AutoAugment : Learning Augmentation Strategies from Data". Ekin D. Cubuk , Barret Zoph, Dandelion Man é, Vijay Vasudevan

# Fonction perte

$$\mathbf{L} = \ell_s + \lambda_u \ell_u$$

Avec:

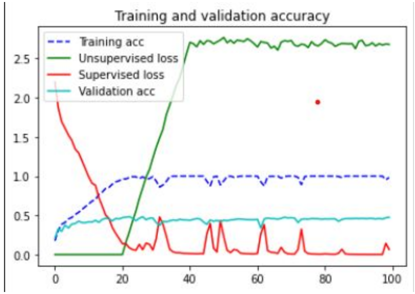
- $\ell_s = \frac{1}{B} \sum_{b=1}^{B} \mathrm{H}(p_b, p_{\mathrm{m}}(y \mid \alpha(x_b)))$ l'entropie croisée sur les données labellisées
- $\ell_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) \geq \tau)\, \mathrm{H}(\hat{q}_b, p_{\mathrm{m}}(y \mid \mathcal{A}(u_b)))$ l'entropie croisée entre les données non labellisées fortement augmentées et les pseudo-labels
- $q_b = p_{\mathrm{m}}(y \mid \alpha(u_b))$ la prédiction du modèle sur une donnée non labellisées faiblement augmentées qu'on retient si l'arg max depasse le threshold $\tau$
- $\lambda_u$ un hyperparamètre à étudier et $\mu$ le ratio de données non labellisées

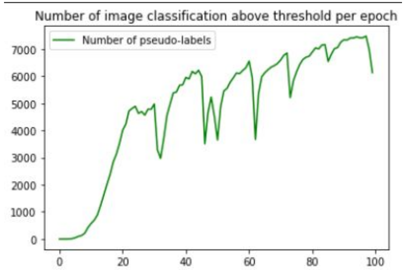
# Optimiseurs

Essais des optimiseurs Adam et SGD avec threshold = 0.9 et $\lambda = \begin{cases} 0 \text{ si } t < 20 \\ (t-20) \times \frac{1}{40} \text{ si } t < 60 \\ 1 \text{ sinon.} \end{cases}$

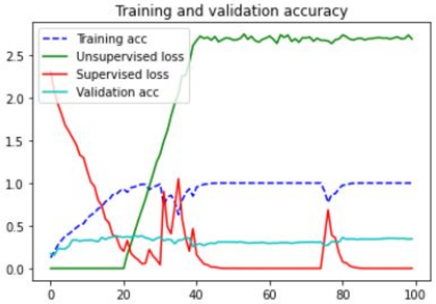
- Optimiseur Adam

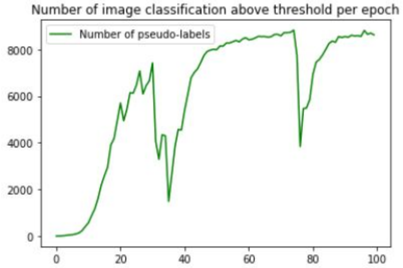
learning rate = 3e-3



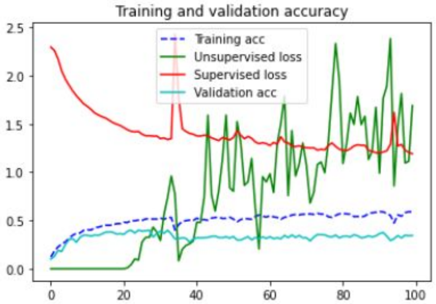



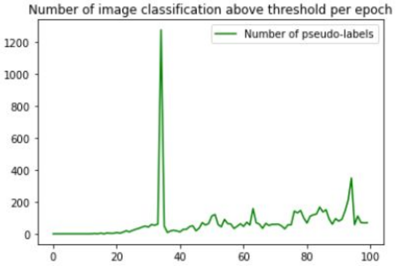

learning rate = 3e-4




learning rate = 3e-5



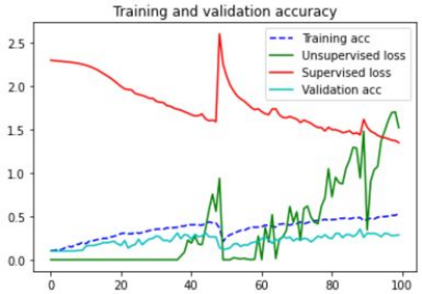



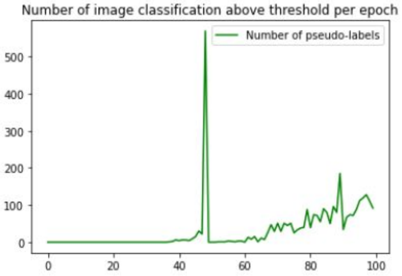
# Optimiseurs

Essais des optimiseurs Adam et SGD avec threshold = 0.9 et $\lambda = \begin{cases} 0 \text{ si } t < 20 \\ (t-20) \times \frac{1}{40} \text{ si } t < 60 \\ 1 \text{ sinon.} \end{cases}$

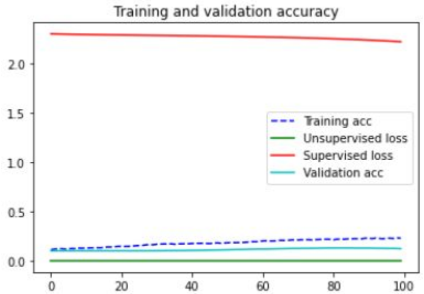
- Optimiseur SGD

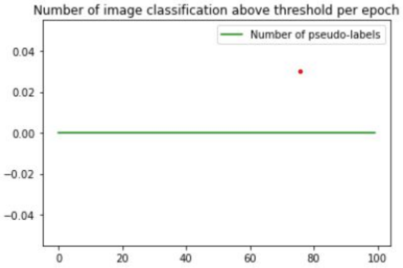
learning rate = 3e-3 et momentum = 0.5



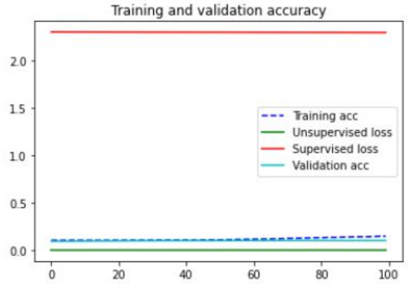




learning rate = 3e-4 et momentum = 0.5



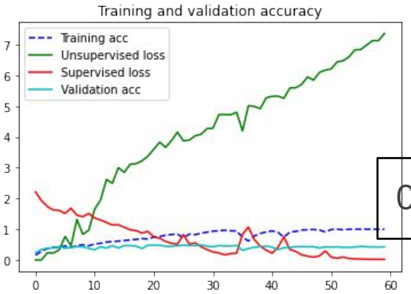

learning rate = 3e-5 et momentum = 0.5



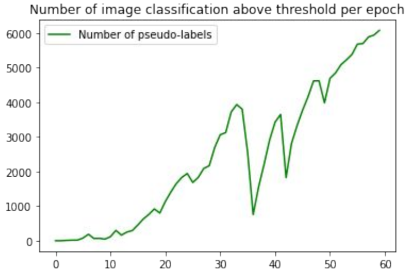

# Comparaison des résultats

1) variation des lambdas

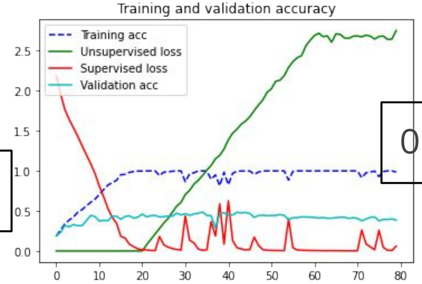
$$\lambda(t) = e^{t/Nbepochs}$$



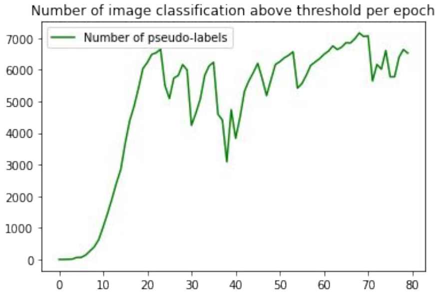

0.47



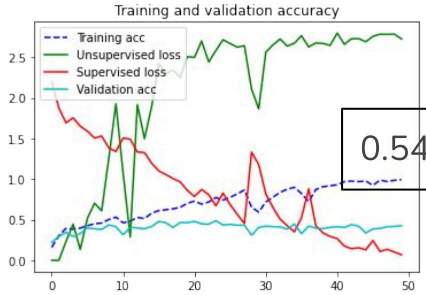

$$\lambda = \begin{cases} 0 \text{ si } t < 20 \\ (t-20) \times \frac{1}{40} \text{ si } t < 60 \\ 1 \text{ sinon.} \end{cases}$$



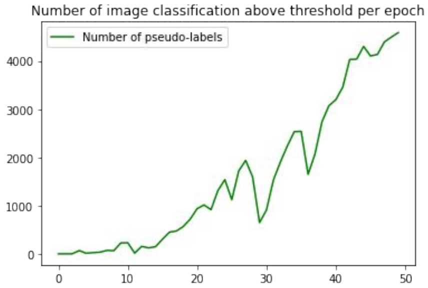

0.51



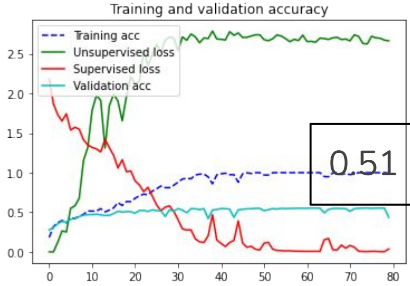

$$\lambda(t) = e^{t/2*Nbepochs}$$



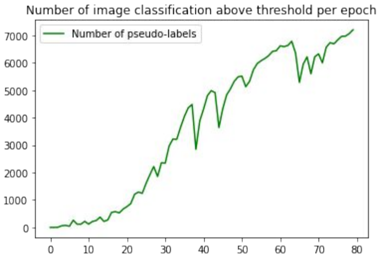

0.54




$$\lambda = cste$$




0.51

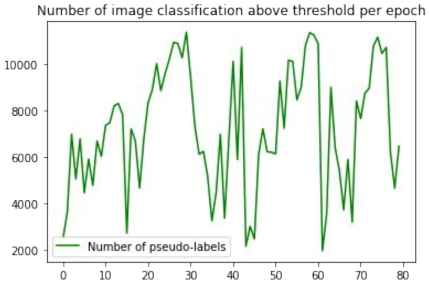


# Comparaison des résultats

2) variation du threshold

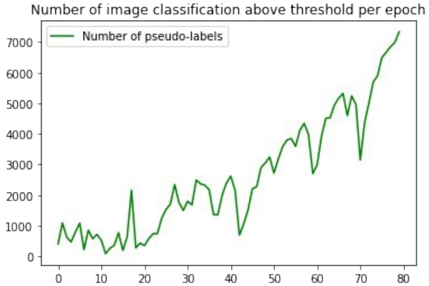
thre = 0.2

acc = 0.23



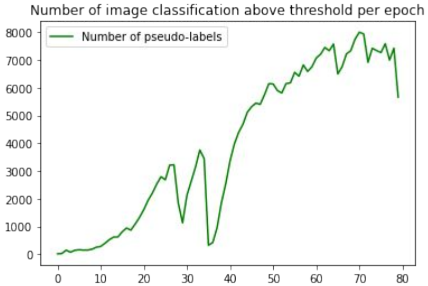

thre = 0.5

acc = 0.38



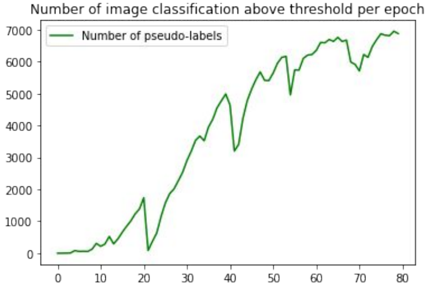

thre = 0.8

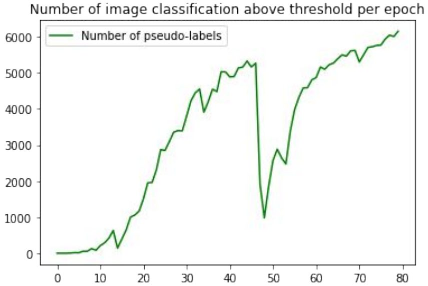
acc = 0.43




thre = 0.9

acc = 0.51




thre = 0.95

acc = 0.53

# Conclusion

- Nécessité d’avoir un jeu de données important
- Meilleur résultat avec
  - Adam
  - LR = 3e-4
  - lambda = exp(t/(2*Nbepochs))
  - threshold = 0.95
- Amélioration par rapport au modèle simplement supervisé de ~+3%