

Réunion Projet CLIPS

FALK Anthonin, HAMIE Bachar

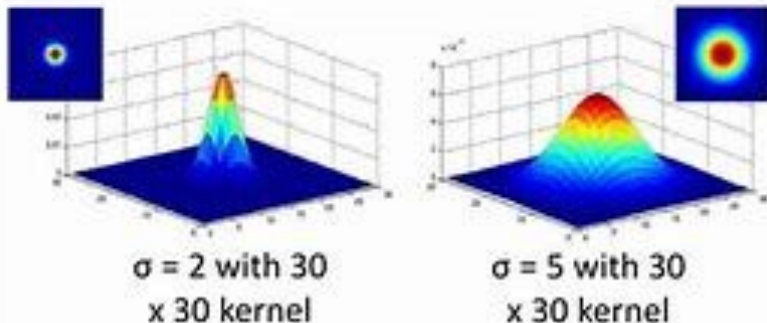




1) CL by smoothing

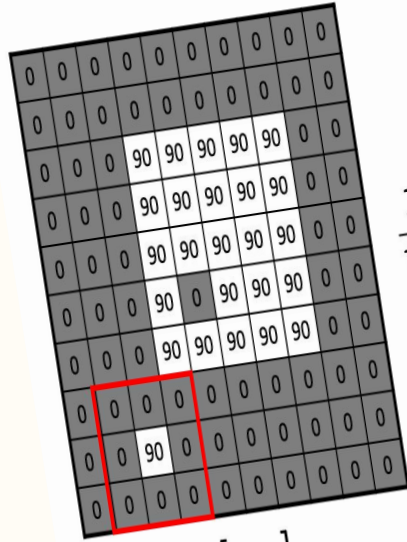
Gaussian filters

Variance of Gaussian: determines extent of smoothing



Curriculum by smoothing

Le **curriculum by smoothing** consiste à introduire un filtre gaussien dans les couches de convolution (convolution layers) au début de l'entraînement. Ce filtre agit comme un lissage sur les données en atténuant les détails fins et le bruit.



$F[x, y]$

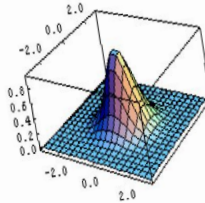
$\frac{1}{16}$

1	2	1
2	4	2
1	2	1

$H[u, v]$

This kernel is an approximation of a 2d Gaussian function:

$$h(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}$$



Etapes:

Filtre gaussien initial : Applique un flou pour simplifier les données.

Progression : Réduit progressivement l'intensité du filtre gaussien pour permettre au modèle d'apprendre à partir des détails complets.

Faiblesse

Il travaille sur la black box qui est le model et ne s'intéresse pas à la complexité différentes du data

-présente une faiblesse lorsqu'il est appliqué avec un lissage, car il ne prend pas en compte la difficulté intrinsèque des données.

-Au lieu d'adapter l'apprentissage en fonction de la complexité réelle des données, il suit une progression fixe de simplicité vers complexité.

-Cela peut amener le modèle à ne pas se concentrer sur les données particulièrement difficiles ou bruitées,



2) Predefined CL

2) Predefined CL

Méthodes avec information **a priori** uniquement

Il y a besoin de deux choses

- Mesureur de difficulté (Difficulty Measurer)
- Planificateur d'étude (Training Scheduler)

2 types de planificateurs :

- Changement des données d'entraînement après un nombre fixe d'époch
- Changement des données d'entraînement à chaque epoch

2) Predefined CL

Points forts :

- Permet de mettre l'information a priori
- Très facile à implémenter
- Peu couteux en mémoire et en temps

Points faibles :

- Comment déterminer la difficulté hormis les cas déjà connus ?
- La mesure de difficulté est-elle vraiment représentative ?
- Pas d'adaptation au réseau



3) Self-Paced Learning

3) Self-Paced Learning (SPL)

w: les paramètres du réseau

v_i : le poids de chaque exemple

lambda : paramètre d'âge

Méthode classique :

Entraînement du réseau en associant un poids à chaque exemple. On cherche à minimiser la fonction suivante (objectif d'apprentissage) :

$$\min_{w, v \in [0,1]^N} \mathbb{E}(w, v; \lambda) \sum_{i=1}^N v_i l_i + g(v; \lambda)$$

$$g(v; \lambda) = -\lambda \sum_{i=1}^N v_i$$

Méthode d'optimisation (à chaque epoch) :

- Etape 1 : on détermine le poids de chaque entrée
- Etape 2 : on continue l'apprentissage du réseau
- Etape 3 : modifier lambda (suite arithmétique/géométrique)

$$v_i^* = \arg \min_{v_i \in [0,1]} v_i l_i + g(v_i; \lambda) \text{ pour } i = 1, 2, \dots, n$$

$$w^* = \arg \min_w \sum_{i=1}^N v_i^* l_i$$

3) Self-Paced Learning (SPL)

Problème de la méthode classique :

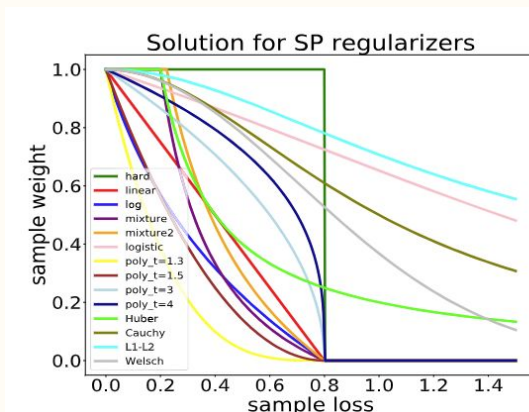
Les poids obtenus sont soit 0, soit 1.

Améliorations possibles :

- Reformuler la fonction objectif d'apprentissage à optimiser
- Changer la fonction de régularisation

TABLE 5
Common Types of SP-Regularizers $g(v; \lambda)$ and the Corresponding Close-Formed Solutions $v^*(l; \lambda)$

Regularizers	$g(v; \lambda)$	$v^*(l; \lambda)$
Hard [54]	$-\lambda \sum_{i=1}^N v_i$	$\begin{cases} 1, & l_i < \lambda \\ 0, & \text{otherwise} \end{cases}$
Linear [40]	$\frac{1}{2} \lambda \sum_{i=1}^N (v_i^2 - 2v_i)$	$\begin{cases} 1 - l_i/\lambda, & l_i < \lambda \\ 0, & \text{otherwise} \end{cases}$
Logarithmic [40]	$\sum_{i=1}^N \left(\zeta v_i - \frac{v_i}{\log \zeta} \right)$ $\zeta = 1 - \lambda, 0 < \lambda < 1$	$\begin{cases} \frac{\log(l_i + \zeta)}{\log \zeta}, & l_i < \lambda \\ 0, & \text{otherwise} \end{cases}$
Mixture [40]	$-\zeta \sum_{i=1}^N \log \left(v_i + \frac{\zeta}{\lambda_1} \right)$ $\zeta = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}, \lambda_1 > \lambda_2 > 0$	$\begin{cases} 1, & l_i \leq \lambda_2 \\ 0, & l_i \geq \lambda_1 \\ \zeta \left(\frac{1}{l_i} - \frac{1}{\lambda_1} \right), & \text{otherwise} \end{cases}$
Mixture2 [141]	$\frac{\gamma^2}{v_i + \frac{\gamma}{\lambda}}, \gamma > 0$	$\begin{cases} 1, & l_i \leq \left(\frac{\lambda \gamma}{\lambda + \gamma} \right)^2 \\ 0, & l_i \geq \lambda^2 \\ \gamma \left(\frac{1}{\sqrt{l_i}} - \frac{1}{\lambda} \right), & \text{otherwise} \end{cases}$
Logistic [127]	$\sum_{i=1}^N \ln(\mu_i)^{\mu_i}$ $+\ln(v_i)^{1-\lambda} - \lambda v_i, \lambda > 0,$ $\mu_i = 1 + e^{-\lambda} - v_i$	$\frac{1+e^{-\lambda}}{1+e^{l_i-\lambda}}$
Polynomial [26]	$\lambda \left(\frac{1}{2} \sum_{i=1}^N v_i \right)$ $\lambda > 0, t \in \mathbb{N}^+$	$\begin{cases} \left(1 - \frac{l_i}{\lambda} \right)^{\frac{1}{t-1}}, & l_i < \lambda \\ 0, & \text{otherwise} \end{cases}$



3) Self-Paced Learning (SPL)

Autres améliorations possibles :

1) Inclure de l'information à priori pour le choix des paramètres v_i :

- A priori sur les valeurs abberantes
- A priori sur la régularité temporelle/spatiale
- A priori sur l'importance de certains exemples
- A priorir de diversité

Les auteurs fournissent des exemples de modification du problème pour inclure ces types d'informations (voir ducoement rédigé sur GitHub)

3) Self-Paced Learning (SPL)

Autres améliorations possibles :

1) Inclure de l'information à priori pour le choix des paramètres v_i :

- A priori sur les valeurs aberrantes
- A priori sur la régularité temporelle/spatiale
- A priori sur l'importance de certains exemples
- A priorir de diversité

Les auteurs fournissent des exemples de modification du problème pour inclure ces types d'informations (voir ducoement rédigé sur GitHub)

2) Mettre à jour λ d'une autre façon

3) Apprendre les hyperparamètres d'arrêt et d'initialisation

2) SPL

Points forts :

- Méthode semi-automatique (inclusion d'information a priori)
- Adapté au réseau
- Méthode connue, beaucoup de références

Points faibles :

- Une solution de bonne qualité demande beaucoup de travail (optimisation de plusieurs hyperparamètres)
- Nécessite une deuxième optimisation (même si elle est plus rapide que l'optimisation des paramètres du réseau)
- Un réseau peu expérimenté peut avoir du mal à estimer la difficulté des exemples



4) Transfer Teacher

4) Transfer Teacher

Un réseau professeur + un réseau étudiant

Entraînement du réseau professeur puis utilisation des coûts de chaque exemple pour définir la difficulté de chacun d'eux.

Plus la perte est faible, plus l'exemple est facile.

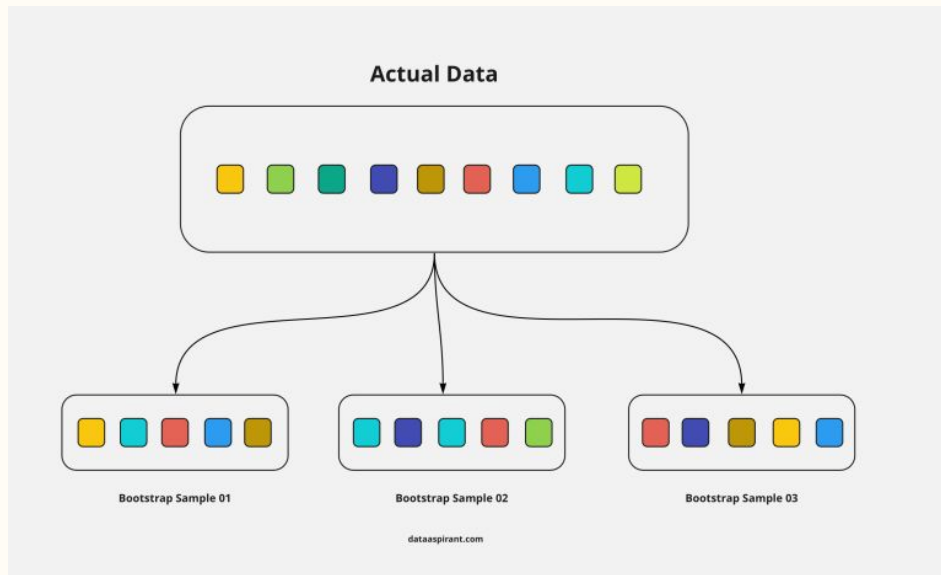
Le professeur peut être un réseau différent de l'élève.

Exemple de stratégies possibles :

- Bootstrapping
- Révision croisée
- Incertitude

Le Bootstrapping score?

Le bootstrap scoring est une méthode pour évaluer la difficulté ou la pertinence d'un exemple donné, afin de construire un curriculum. Il consiste à Mesurer la difficulté d'un exemple en utilisant des estimations issues de modèles préexistants ou en cours d'entraînement.



Critère pour le Bootstrapping score?

Erreur de segmentation

Calculer l'erreur entre la prédiction et le ground truth pour chaque image ou région

Plusieurs possibilités de combinaisons des scores :

- Dice score seul
- IoU score seul
- Dice et IoU

Critère pour le Bootstrapping score?



Erreur Historique

Si le modèle a systématiquement des erreurs sur une région ou une image donnée, ces exemples sont marqués comme difficiles.

Critere pour le Bootstraping score?



**Variabilite locale ou
structurelle**

Les exemples avec des formes complexes ou des bordures floues peuvent être considérés comme plus difficiles. Mesurer la courbure ou les gradients dans les masques.

Les points forts ?

Adaptabilité : Le score peut être recalculé périodiquement pour s'adapter aux améliorations du modèle, créant un curriculum dynamique

Méthode automatique : Pas de détermination de l'importance de chaque exemple manuellement

Fonctionne sur des exemples similaires

Facile à implémenter

Les points faibles avec solution?

Les scores bootstrap initiaux sont souvent calculés à partir d'un modèle peu ou pas entraîné. Ces scores peuvent être biaisés ou non représentatifs des vraies difficultés des exemples,

Solution potentielle : Recalculer périodiquement les scores pendant l'entraînement pour corriger les biais initiaux.

Nécessité d'un deuxième entraînement (mais déjà réalisé ici)

Difficile d'ajouter de l'information a priori

Définition de la "difficulté" La difficulté d'un exemple dépend du critère choisi (IoU, confiance, erreur, etc.). Si ce critère est mal défini ou trop simpliste : Des exemples importants (mais mal classés) peuvent être négligés.

Solution potentielle : Utiliser des critères de difficulté combinés (ex. : IoU + confiance + Dice score).



5) RL Teacher

5) RL Teacher

Interactions entre professeur et étudiant

Utilisation de méthodes par renforcements

Points forts :

- Fonctionne bien avec des réseaux profonds
- Adapté au multitache

Points faibles :

- Difficile à entraîner
- Prends du temps à entraîner
- Beaucoup de programmation, pas du plug and play
- Aucune info à priori, tout est automatique



Conclusion

Conclusion

On va privilégier la méthode du Transfer Teacher (bootstrapping)

Raisons :

- Facile et rapide à implémenter
- Un réseau déjà pré-entraîné (donc information a priori)
- Des solutions partielles aux problèmes que cette méthode peut poser