

# Projet : Développez une preuve de concept

Antoine Maby - 28/12/2021

# Problématique

Le domaine du machine learning, et plus généralement de la data science, évolue très rapidement

Il est important de développer l'habitude de se tenir au courant des avancées dans ce domaine en effectuant une veille thématique

Il est important de monter rapidement en compétence sur une nouvelle thématique, en sachant effectuer une recherche et mettre en pratique un nouvel algorithme de façon autonome

# Problématique

Déroulement du problème :

- Trouver un algorithme
- Choisir un Dataset
- Trouver une bibliographie
- Choisir une Baseline
- Implémenter la nouvelle méthode
- Comparer les résultats

# Choix du problème

- Le projet précédent portait sur la classification d'images
- Les algorithmes nécessitent une grande quantité d'images étiquetées
- Existe-t'il une solution à ce problème ?

# Self-supervised Learning

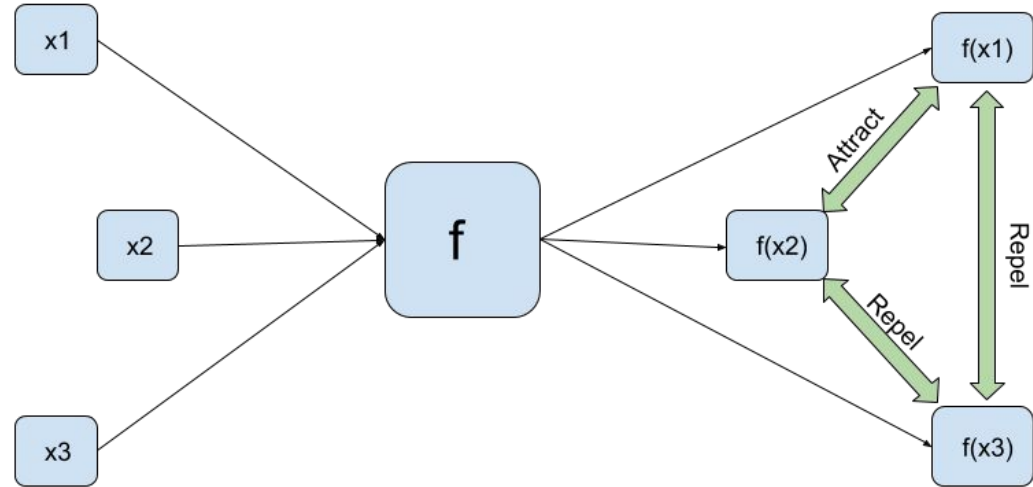
- Rechercher dans des données non étiquetées des groupes d'exemples partageant des caractéristiques communes
- Chacune d'entre elles est fournie, en entrée, dans sa forme originale et une version transformée
- Les plus performantes sont : les Transformer et les méthodes Contrastives

# Self-supervised Learning

Méthode Contrastive

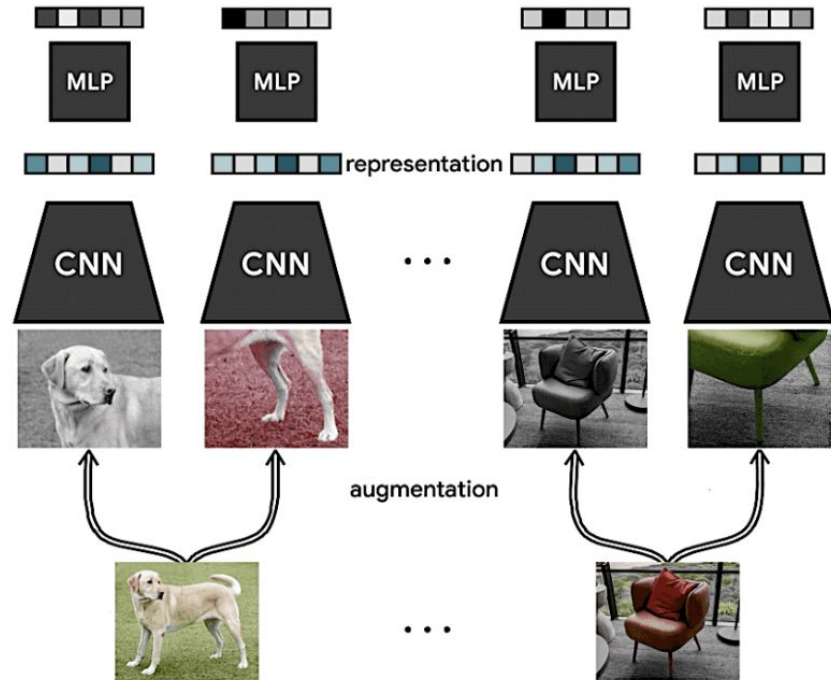
Un algorithme  
semi-supervisé

Mon choix s'est porté sur  
SimCLR



# SimCLR

- Augmentation
- Création de représentation
- MLP



# SimCLR

---

**Algorithm 1** SimCLR’s main learning algorithm.

---

**input:** batch size  $N$ , constant  $\tau$ , structure of  $f, g, \mathcal{T}$ .

**for** sampled minibatch  $\{x_k\}_{k=1}^N$  **do**  
     **for all**  $k \in \{1, \dots, N\}$  **do**

draw two augmentation functions  $t \sim \mathcal{T}, t' \sim \mathcal{T}$

# the first augmentation

$$\tilde{\mathbf{x}}_{2k-1} = t(\mathbf{x}_k)$$
$$h_{2k-1} = f(\tilde{x}_{2k-1}) \quad \# \text{ representation}$$
$$z_{2k-1} = g(h_{2k-1}) \quad \# \text{ projection}$$

# the second augmentation

$$\tilde{\mathbf{x}}_{2k} = t'(\mathbf{x}_k)$$
$$h_{2k} = f(\tilde{x}_{2k}) \quad \# \text{ representation}$$
$$\mathbf{z}_{2k} = g(\mathbf{h}_{2k}) \quad \# \text{ projection}$$

end for

**for all**  $i \in \{1, \dots, 2N\}$  **and**  $j \in \{1, \dots, 2N\}$  **do**
$$s_{i,j} = \mathbf{z}_i^\top \mathbf{z}_j / (\|\mathbf{z}_i\| \|\mathbf{z}_j\|) \quad \# \text{ pairwise similarity}$$

end for

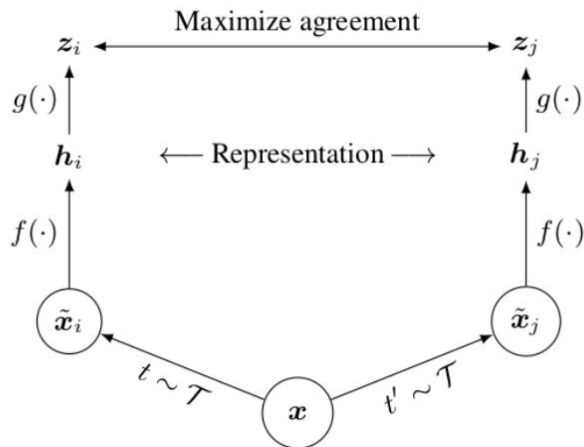
define  $\ell(i, j)$  as  $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^N \exp(s_{i,k}/\tau)}$

$$c = \frac{1}{N} \sum_{k=1}^N [\ell(\mathbf{y}_k - 1, \mathbf{y}_k) + \ell(\mathbf{y}_k, \mathbf{y}_k - 1)]$$
$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^N [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$$

update networks  $f$  and  $g$  to minimize  $\mathcal{L}$

update networks  $f$  and  $g$  to minimize  $\mathcal{L}$   
and for

end for

return encoder network  $f(\cdot)$ , and throw away  $g(\cdot)$ 



# Les données

- Base de données : Stanford Dogs Dataset
- Plus de 20 000 images de chiens
- 120 races différentes



# Les données

	Pretrain	Train	Test
Nombre de photos	12000	1716	6864



# Baseline

- Inceptionv3
- Entraînement seulement sur le Classifieur
- Classifieur composé de Trois couches denses et d'un Drop Out

# Baseline

- Résultats obtenus après entraînement sur le Train, performance sur le Test

	Inceptionv3
Accuracy Top 1	4.1

# Mise en place de SimCLR

- Pré-entraînement grâce au routine fourni sur <https://github.com/google-research/simclr>
- Finetune sur les données de Train
- Tests des performances sur le Test

# Mise en place de SimCLR

- Résultats obtenus

	InceptionV3	SimCLRv1
Accuracy Top 1	4.1	6,8

# Mise en place de SimCLR

- SimCLR est déjà plus performant que notre Baseline
- Les performances restent limitées
- Testons le modèle fourni par les auteurs en finetune sur les données

# Mise en place de SimCLR

- Pré-entraînement des auteurs sur ImageNet
- Finetune sur les données de Train
- Tests des performances sur le Test



# Mise en place de SimCLR

- Résultats obtenus

	Inceptionv3	SimCLRv1	SimCLRv2
Accuracy Top 1	4.1	6.8	62,5

# Mise en place de SimCLR

- SimCLR est bien plus performant que notre Baseline et notre SimCLR
- Les performances s'approchent des résultats obtenus sur le projet 6
- Ceci nous donne une borne inférieure des performances possibles

# Conclusion

- SimCLR répond à notre problématique en ayant de bonnes performances
- Des améliorations sont possibles :
  - pré-entraînement sur plus grand nombre données
  - Meilleurs paramétrages du pré-entraînement
- Il est peut être possible de dépasser les performances obtenu lors du projet 6



Merci pour votre  
attention