Polar Transformer Networks Review

Thibault **Gorassini** & Umit **Ince**
Sorbonne Université, Paris VI

Contexte

Lorsque l'on travaille sur des tâches de classification, notamment d'images, il peut être souhaitable d'avoir un algorithme qui soit robuste aux variations : malgré les perturbations présentent sur l'image, le modèle doit être capable d'estimer correctement la classe de celle-ci. Les perturbations pouvant être rencontrées sont diverses : **rotations, changements d'échelle ou de points de vue, bruits**. Le but de l'article est la présentation d'un modèle **Polar Transform Network** invariant à la translation et équivariant aux rotations et aux dilatations.

Problématique

Les tâches de classifications d'images ayant subies des rotations peuvent sembler triviales pour un humain, mais elles peuvent poser problèmes aux algorithmes. En effet ceux-ci travaillent avec une représentation matricielle de l'image. Ainsi une simple transformation, comme une rotation, peut changer complètement la matrice de pixels et ainsi la représentation de l'image pour l'algorithme. Pour pouvoir classifier correctement, il faut donc pouvoir gérer ces transformations

Figure: Rotation du chiffre 1 - MNIST

Invariance et Equivariance

On définit les équivariances et invariances de groupe, avec ϕ une fonction de transformation et L la représentation d'une image x :

$$Invariance : L(x) = L(\phi(x)) \quad (1)$$

$$Equivariance : \phi(L(x)) = L(\phi(x)) \quad (2)$$

Etat de l'art

Le **Spatial Transformer Network** est un modèle de classification d'images déformées (i.e rotation, translation) qui fonctionne en 3 étapes:

STN[1]

- **(a)** Localisation network: Un réseau de neurones qui extrait et centre la partie de l'image sur laquelle on souhaite travailler, par exemple des chiffres.
- **(b)** Grid generator: Consiste à trouver une matrice de transformation qui annule la rotation subit par l'image, pour revenir à une image sans déformation.
- **(c)** CNN: un convolutional neural network qui reconnait l'image finale.

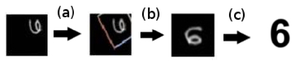

Figure: Exemple de reconnaissance d'un 6 par un STN

Modèle

Le modèle **Polar Transformer Network**[2] combine des idées du **STN**, mais au lieu d'utiliser une transformation apprise pour annuler les déformations, on utilise un module Polar Transformer qui utilise des **coordonnées polaires** afin de changer la représentation de l'image. Voici les 3 modules du modèle:

Polar Origin Predictor

Le Polar Origin Predictor est un module qui prédit le centre de l'information sur l'image et l'extrait. Il est similaire au module **Localisation Network** du STN. Il utilise une **heatmap**[3] de l'image qui représente une pondération de l'information sur celle-ci. Le **centroide de la heatmap** est utilisé comme **centre polaire**.
Input : l'image dont on doit prédire la classe.
Output : Single channel map : centroide de l'image.

Polar transformer Module

Ce module contient l'idée centrale du PTN. Plutôt que d'apprendre et d'appliquer une matrice de correction de la déformation comme dans le STN, le module applique une transformation appelé **log polaire**. Cette transformation fait appel aux notions de coordonnées polaires d'un point, où un point est représenté par son **angle** à l'origine, et le **logarithme** de sa **distance** à celle-ci. La transformation log-polaire s'exprime ainsi, en termes de points source et cible:

$$x_i = x_0 + r^{x_i/W} cos\frac{2\pi y_i}{H} \quad (3)$$

$$y_i = y_0 + r^{x_i/W} sin\frac{2\pi y_i}{H} \quad (4)$$

(x_0, y_0) est l'origine, W, H est la taille de l'image et r la distance maximale à l'origine.
Input : l'image dont on doit prédire la classe et son centroide.
Output : La représentation log-polaire de l'image.

Dans une représentation log-polaire, les rotations autour de l'origine deviennent des décalages verticaux. Les dilatations deviennent des décalages horizontaux. La distance entre les deux lignes de la figure est proportionnelle à la rotation/dilatation.

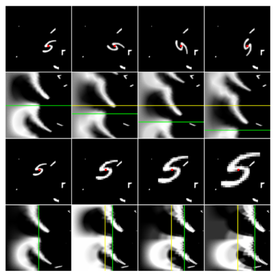

Figure: Partie sup : Séquence de rotation. Partie inf : Séquence de dilatation

Classifieur

Le dernier module est un classifieur, qui est entrainé et prédit sur les représentations log-polaires des images. Il s'agit d'un Convolutional Neural Network. Les propriétés particulières des représentations log-polaires permettent de rendre le classifieur equivariant aux rotations et dilatations, et invariant à la translation. L'architecture finale du PTN est présentée à la figure 4.
Input : La représentation log-polaire de l'image.
Output : La classe de l'image.

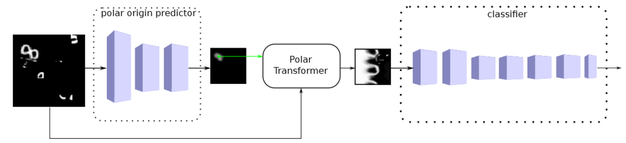

Figure: Architecture du PTN

Expérience

Dans cette partie nous présentons les différents tests qui ont été menés, les jeux de données utilisés, les différents résultats.

Données utilisées

Nous exploitons principalement des variations du jeux de données **MNIST**, dont il existe plein de datasets différents permettant de tester la robustesse de nouveaux algorithmes de machine learning. Nous avons par exemple utilisé la base **MNIST-rot** (a), qui contient les chiffres de **MNIST** mais avec un angle de rotation aléatoire uniforme entre 0 et 2π. Nous avons aussi utilisé la base **MNIST-rot-back image** (b) qui combine les chiffres pivotés de MNIST-rot avec une image d'arrière plan perturbatrice en noir et blanc.

(a)

(b)

Figure: (a) MNIST-rot (b) MNIST-rot-back-image

Résultats

Nous avons lancé le Polar Transformer Network en apprentissage sur 50 époques. Pour avoir une baseline nous avons aussi comparé à deux autres implémentations : un Spatial Transformer Net et un CNN eux aussi entrainés sur 50 époques, afin d'avoir une baseline pour valider les résultats.

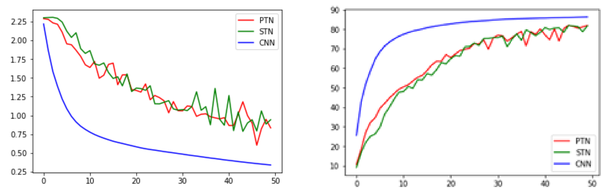

Figure: A gauche : Erreur en apprentissage. A droite : Accuracy en test.

References

[1] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al.
Spatial transformer networks.
In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[2] Xiaowei Zhou Kostas Daniilidis Carlos Esteves, Christine Allen-Blanchette.
Polar transformer networks.
International Conference on Learning Representations, 2018.

[3] Alejandro Newell, Kaiyu Yang, and Jia Deng.
Stacked hourglass networks for human pose estimation.
In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.