On peut essayer de détecter un drift de deux façons, en amont du modèle par analyse des distributions des données, ou en aval en regardant la stabilité des performances du modèle. Afin de créer un cadre pour étudier les changements de distribution on va se placer dans le cadre ou l’on regarde l’évolution de la qualité de prédiction des modèles soumis à un changement de distribution.

On cherche à vérifier si l’on peut mesurer un drift par un une baisse de performance d’un modèle sur un jeu de donné où le drift est évident. Pour matérialiser ce changement de distribution, on choisit le jeu de données MNIST. On va entraîner un modèle sur certaines modalités de chiffres et scorer sur l’ensemble des modalité afin de mesurer l’impact du drift.

Pour cela on va entraîner un GAN constitué d’un générateur et d’un discriminateur. Le but du générateur est d’apprendre la distribution sous-jacente des images afin d’en générer les plus vraisemblables possible. Le but du discriminateur est de prédire si les images sont des vraies ou si elles proviennent du générateur. On se place dans le cadre d’un changement de distribution des données à priori.

Avantages d’un GAN sur cet exemple:

* Permettre d’utiliser les images sans aucun traitement préalable pouvant potentiellement amoindrir le drift.
* De nombreuses architectures déjà éprouvées pour ce problème existent et rendent ainsi possible de ne pas se préoccuper des hyperparamètres où de l’architecture.

Inconvénients d’un GAN :

* Interprétabilité : Il est difficile de savoir comment le modèle se fait une représentation des données.
* \* Le temps d’entraînement long. Après analyse, une augmentation du nombre d’itération augmente la probabilité moyenne des prédictions, le discriminateur attribue une probabilité forte aux vrais chiffres mais le générateur devient meilleur et le discriminateur attribue aux nombres généré une probabilité également plus forte.
* \* Un manque de précision, on ne peut pas analyser l’appartenance à une classe. Une autre architecture de GAN permets de palier à ce problème.

Pour j allant de 1 à 9 on entraine le modèle sur j des 10 classes du jeu de données. On entraine le modèle jusqu’à ce que les performances stagnent, c’est-à-dire 70 000 itérations. L’entrainement des dix instances du modèle va mettre 24h sur un ordinateur classique.

Toutes les 500 itérations on donne au discriminateur des images des 10 classes jamais vu lors de la phase d’entrainement, puis une séries d’images générés. On va ensuite comparer les scores attribués aux images dont les chiffres figuraient dans la base d’entrainement à ceux qui n’y figuraient pas et aux images générées au fil des itérations.

Résultats

On constate qu’augmenter le nombre d’itérations fait augmenter les probabilités d’appartenance à la base d’image réelle données a toutes les images générés :   
- réelles dont les classes ont été utilisés comme base d’apprentissage.  
- réelles dont des observations des classes n’ont pas été vue par le générateur.  
- générées par le générateur.

Les probabilités données par le discriminateur cessent néanmoins d’augmenter après 70000 itérations avec des valeurs autours de 0.8.