



UNIVERSITÉ
LAVAL

Auto-apprentissage

Stage en apprentissage automatique
Été 2020

Présenté à
M. Brahim Chaib-draa & M. Mathieu Pagé Fortin

Par
Jérémy Gince, 111 224 046, jérémy.gince.1@ulaval.ca

Université Laval
May 15, 2020

Contents

1	Auto-apprentissage	4
1.1	Exemple, auto-apprentissage	5
1.1.1	Stratégie	5
1.1.2	Résultats	5

1 Auto-apprentissage

L'apprentissage non supervisé est un type d'apprentissage où les données d'entraînement ne sont pas étiquetées. En effet, les données d'apprentissage sont fournies au modèle sans que celui-ci ait la capacité de directement savoir s'il a tort ou raison. Il doit apprendre par lui-même la corrélation entre les données sans support d'un expert. Un bon exemple de ce type d'apprentissage est le partitionnement de données ou plus communément appelé «clustering». En effet, le partitionnement de données se fait de façon non supervisée. On demande au modèle de trouver lui-même la structure cachée des données afin de regrouper les données dans des sous-groupes possédant les mêmes caractéristiques.

L'apprentissage non-supervisé est divisé en plusieurs domaines. Un de ceux-ci est l'auto-apprentissage ou *self-learning*. En fait, la particularité de ce sous-domaine de l'apprentissage non-supervisé est que ce type d'entraînement apprend lui-même à générer des étiquettes aux données d'entraînement. Il va de soit que l'absence de d'étiquettes générés par des expert rend la tâche d'entraînement plus complexe, alors il est normale de se demander qu'elles sont les avantages de ce type d'entraînement.

Dans un premier temps, l'apprentissage supervisé possède le grand désavantage d'être excessivement long à préparer. En effet, afin d'apprendre des tâches complexe, un modèle doit avoir accès à des milliers voir même des millions de données d'entraînement et il ne manque pas d'argument pour mettre en évidence la tâche colossal que représente l'étiquetage d'un aussi grand nombre de données. L'auto-apprentissage ne requiert aucun expert et donc va dans le même de ce pourquoi l'intelligence est inventé, soit l'accomplissement d'une tâche fait par un ordinateur sans intervention humaine.

Dans un deuxième temps, il existe une technique en apprentissage profond servant à transférer l'apprentissage d'un modèle à un autre modèle. Cette technique se nomme trivialement le transfert d'apprentissage. Cette dernière est beaucoup utiliser puisqu'elle permet à un modèle de converger excessivement plus rapidement vers une solution stable que si ce modèle avait commencer son apprentissage de zéro. En effet, cela est possible puisqu'il existe certaine caractéristiques récurrente à plusieurs problèmes. Prenons comme exemple la vision par ordinateur, un modèle entraîné à reconnaître des chatons aura appris à reconnaître certaine caractéristique d'un visage, d'un mammifère, de textures, d'une image etc. Ces caractéristiques peuvent logiquement s'appliquer à la reconnaissance faciale, ou la classification de types de tigre. L'auto-apprentissage sera donc très utile pour faire du transfert d'apprentissage. En effet, il est possible d'entraîner un modèle afin qu'il exécute une tâche x de façon non supervisé afin de transférer cette apprentissage à des modèles superviser entraîné à exécuter les tâches y_k . Afin de concrétiser le tout, un petit exemple très populaire est fait à la section 1.1.

L'auto apprentissage est actuellement utilisé pour le traitement du langage naturelle ainsi que pour la classification et obtient de bons résultats dans ces domaines. Il est moins utilisé en vision numérique actuellement puisque certaines tâches sont définitivement très complexes et requiert d'avantage de recherche dans de domaine afin d'obtenir des résultats aussi bon qu'en sémantique. Toutefois, l'auto-apprentissage promet d'être le future de l'apprentissage machine et d'être l'avenir de domaines tels que la conduite autonome, la prédiction de vidéo, la résolution d'équations différentielles complexes et bien d'autres.

1.1 Exemple, auto-apprentissage

1.1.1 Stratégie

L'objectif final sera d'entraîner un classifieur d'espèces de chat avec un nombre très limité de données étiquetées par un expert. Toutefois, un grand nombre de données non étiquetées sont disponibles sur un certain serveur. La stratégie sera donc de concevoir un modèle d'auto-apprentissage exécutant une certaine tâche reliée à l'ensemble de données non étiquetées afin de préparer les caractéristiques associées aux espèces de chat. Ensuite, ces caractéristiques seront fournies au classifieur supervisé qui sera entraîné sur les données étiquetées.

Il faut maintenant déterminer la tâche pour le modèle non-supervisé ainsi que son architecture. Sa tâche sera de prédire la coloration des données d'entraînement transformées en images nuancées de gris. Dû à une telle tâche, l'architecture utilisée pourrait bel et bien être une *Fully Convolutional Network (FCN)* qui est un ensemble de couches de convolution avec une déconvolution en sortie afin de prédire un tenseur représentant l'image en couleur. Afin d'imager cette stratégie, on peut observer un schéma simpliste du modèle actuel à la figure 1.

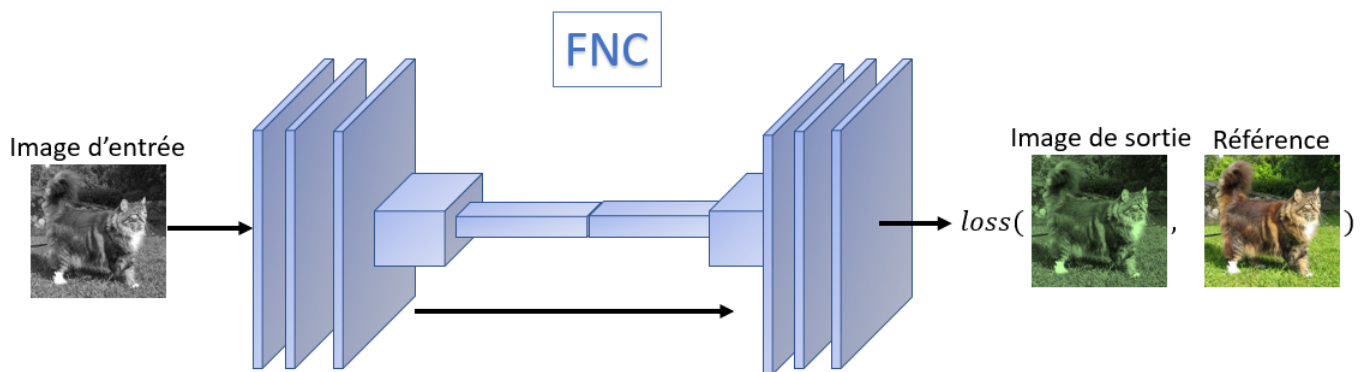


Figure 1: Schéma simpliste de l'exemple de chat

Finalement une fois le modèle d'auto-apprentissage entraîné, on peut utiliser une architecture telle qu'*AlexNet* pour le classifieur et ainsi transférer les paramètres des couches de convolution de la *FCN* dans les couches de convolution d'*AlexNet*.

1.1.2 Résultats

Pour commencer, la définition du modèle ainsi que son apprentissage a été exécutée à l'aide du module python *tensorflow*. On peut observer le schéma de l'architecture utilisé généré par ce dernier module à la figure 2. Les données utilisées afin d'entraîner le présent modèle est un sous-ensemble de 51 classes sur 61 comptant un nombre d'environ 63 000 images de l'ensemble de données disponible au lien suivant: <https://www.kaggle.com/ma7555/cat-breeds-dataset>.

Afin de diversifier les données, celles-ci ont été augmentées avec des techniques utilisées par *tensorflow* telles que le flip horizontal, le zoom, la rotation, etc. De plus, afin de faciliter l'entraînement du réseau de neurones profond, les images ont été transformées de RGB à Lab. De cette façon, le modèle doit seulement apprendre à prédire un tenseur composé de 2 canaux (ab) au lieu de 3 canaux (RGB) afin d'appliquer la coloration sur l'image en nuance de gris donnée en entrée du modèle.

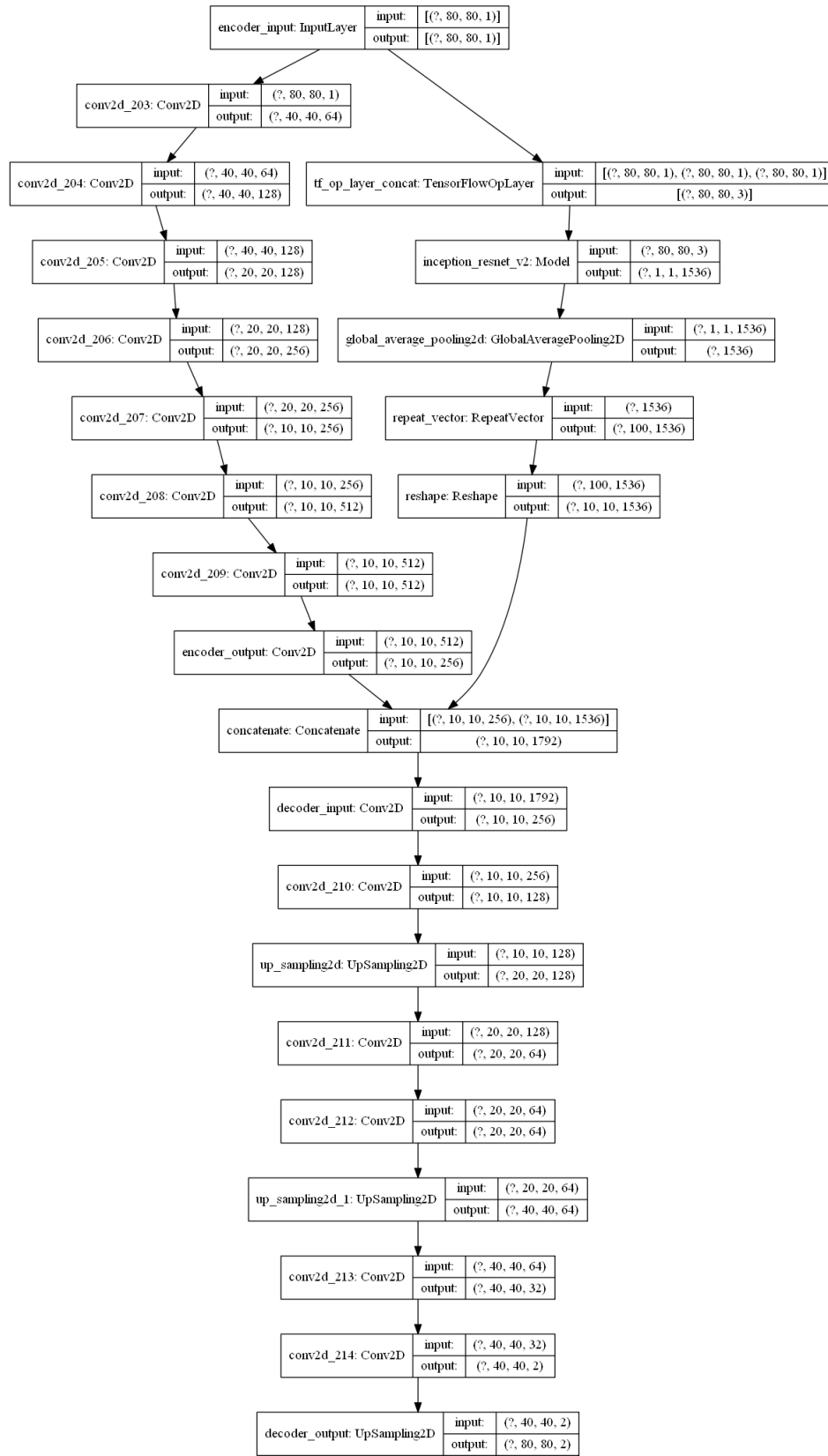


Figure 2: Architecture

Le modèle colorieur de chats a été entraîné avec les hyperparamètres affichés au tableau 1.

Largeur d'images [<i>pix</i>]	Hauteur d'images [<i>pix</i>]	mini-batch [—]	taux d'apprentissage [—]	nombre d'epoch [—]	Fonction de coût [—]
80	80	100	1e-3	31	mae (L1)

Table 1: hyperparamètres pour la coloration de chats

On peut observer à l'aide de la courbe d'apprentissage à la figure 3 que le modèle ne semble pas apprendre durant son apprentissage. Cela peut être dû à plusieurs raisons, mais la plus susceptible d'avoir un grand impact est le fait que le coût associé à chacun des epoch est très petit ce qui aura comme effet de minimiser grandement la mise à jour des poids du modèle. Dans ce cas, une solution serait peut-être d'essayer de trouver une métrique efficace afin de bien exprimer la différence entre deux tenseurs ab dans le but de fournir un coût plus représentatif de la réalité au modèle.

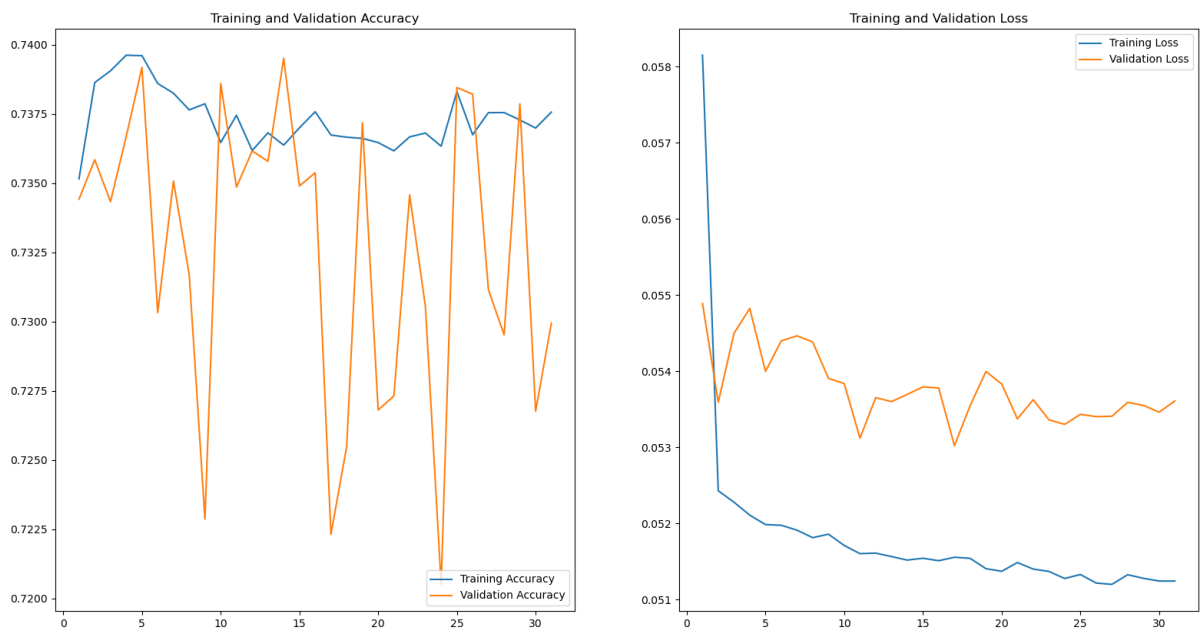


Figure 3: Courbe d'apprentissage

À la figure 4 on peut observer un exemple de prédiction de modèle après son entraînement. Il est facile de remarquer que la prédiction est loin de ressembler à ce qui était attendu pour un observateur humain.

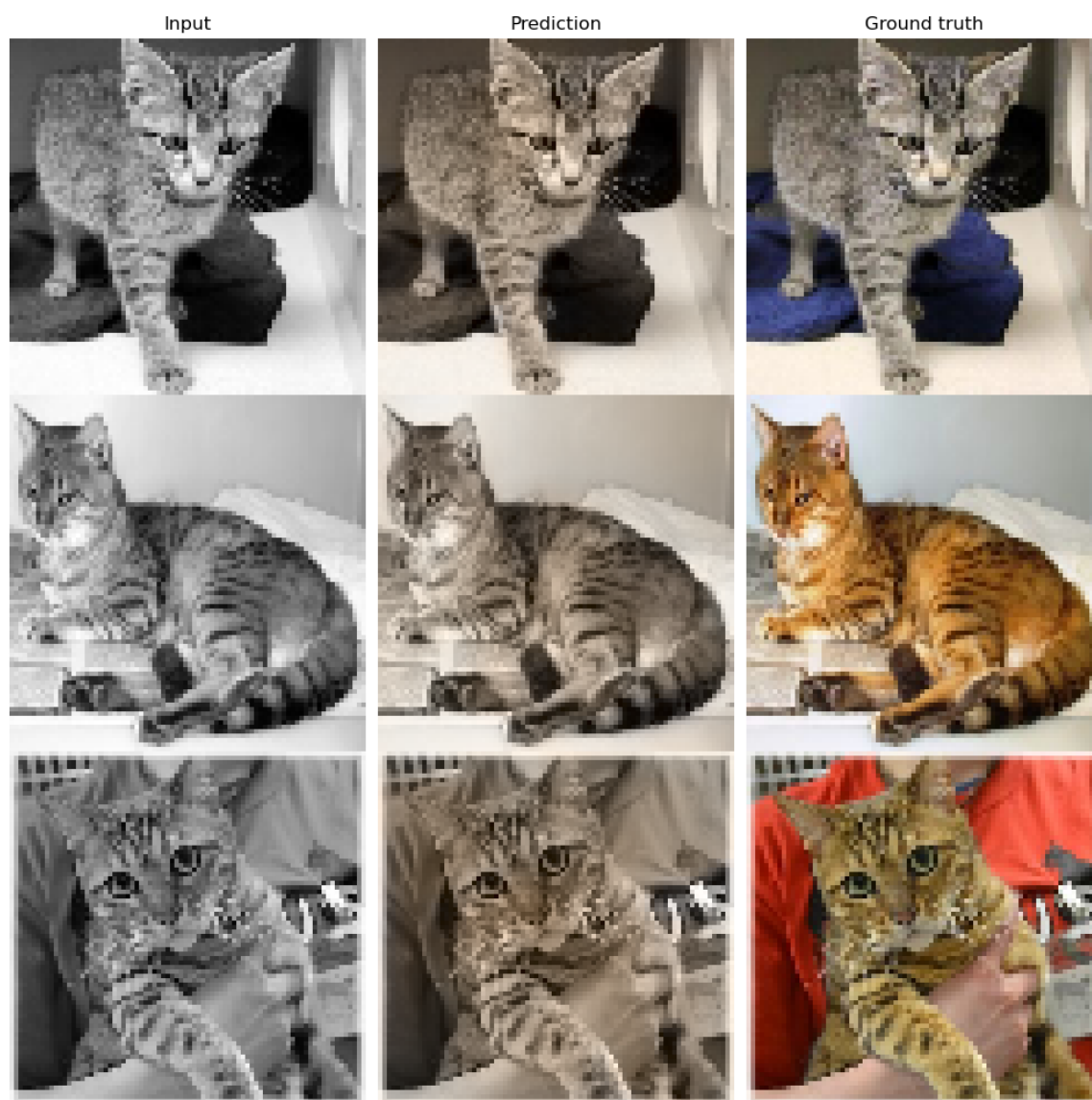


Figure 4: Exemple de prédiction