Générer ou reconnaître des images : les réseaux de neurones à la rescousse

Gabriel Peyré

CNRS & DMA
PSL, École normale supérieure

On sait désormais comment entraîner de façon supervisée des réseaux de neurones (voir l'article précédent). Ceci permet de résoudre efficacement des problèmes de classification, par exemple de reconnaissance d'images. Ce qui est peut-être encore plus surprenant, c'est que ces mêmes réseaux de neurones sont également utilisés de façon non supervisée afin de générer automatiquement des textes ou des images « virtuels », ce que l'on appelle souvent des *deep fake*s. Savez-vous qu'il existe un lien entre l'apprentissage de réseaux de neurones génératifs et la théorie du transport optimal ? Cette dernière a été proposée par Gaspard Monge (1746 – 1818), conte de Péluse, au XVIII^{e} siècle et a été reformulée par Leonid Vitalievitch Kantorovitch (1912 – 1986) au milieu du XX^{e} siècle. Elle est maintenant devenue un outil de choix pour aborder l'explosion récente de la science des données.

Une utilisation « à l'envers » des réseaux de neurones

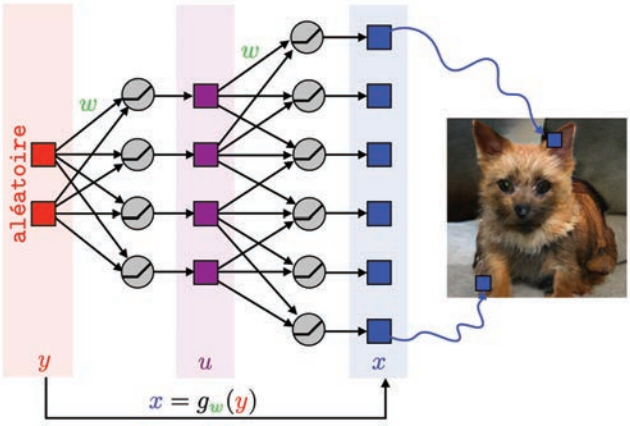
Au lieu d'utiliser des réseaux de neurones pour analyser des images, on peut les utiliser « à l'envers » afin de générer des images. Ces *réseaux de neurones génératifs* trouvent des applications pour les effets spéciaux, les jeux vidéo ou encore la création artistique. On retrouve des approches similaires dans l'apprentissage des voitures autonomes et la résolution de jeux de stratégie. La figure suivante montre la structure d'un tel réseau g_w, qui dépend de poids w. Les couches jouent en quelque sorte des rôles miroirs par rapport à l'architecture des réseaux de neurones discriminatifs de l'article précédent. À partir d'une entrée y composée d'un « petit » nombre de valeurs, qui sont typiquement tirées aléatoirement, on génère une image $x = g_w(y)$.

Le problème de l'apprentissage de tels réseaux est *non supervisé* : on dispose uniquement d'un grand nombre d'images d'apprentissage, sans indication sur ce qu'elles contiennent. Il n'y a plus besoin d'intervention humaine pour indiquer au réseau le contenu des images qu'il doit reconnaître !

* L'auteur remercie Gwenn Guichaoua pour sa relecture attentive, ainsi que Sébastien Racanière et Vincent Barra pour leurs corrections.

Un réseau de neurones génératif simplifié (un réseau permettant de générer des images aussi complexes possède en réalité plus de couches).

© G. Peyré



La collecte de données est plus facile que pour l'entraînement de réseaux discriminatifs. De plus, ce principe d'apprentissage non supervisé est « proche » de la façon dont les enfants apprennent, principalement en observant et manipulant le monde qui les entoure. Le but est alors de sélectionner les poids w des neurones du réseau g_w de sorte que les images aléatoires générées (les images fausses, *fakes* en anglais) ressemblent « le plus possible » aux images d'apprentissage.

L'apprentissage non supervisé : un problème de poids !

Le but des réseaux de neurones génératifs n'est pas de résoudre une tâche telle que la reconnaissance d'objets dans les images. Afin d'entraîner les poids w du réseau, il faut formaliser mathématiquement le problème. Il s'agit de générer un ensemble d'images « virtuelles » (les fausses) qui « ressemblent » aux images réelles d'une base de données. Il ne s'agit pas simplement qu'une image générée « ressemble » à une image réelle, il faut mettre en correspondance les deux ensembles d'images. Par exemple, si dans la base de données il y a la moitié d'images de chiens et la moitié d'images de chats, il faut que le réseau génère également pour moitié des chiens et pour moitié des chats.

On va noter $\{z_1, z_2 \dots z_n\}$ l'ensemble des n images de la base de données. Le nombre n d'images est très grand, de l'ordre de plusieurs milliers ou millions. Étant donné un réseau de neurones génératif g_w, qui est paramétré par ses poids w, on note $\{x_1, x_2 \dots x_n\}$ un ensemble de n images « fausses » générées aléatoirement par le réseau. Pour générer la première image fausse x_1, ceci signifie que l'on tire aléatoirement les valeurs d'entrée y_1 et que l'on applique

le réseau à ces entrées, pour obtenir l'image virtuelle $x_1 = g_w(y_1)$. On a ensuite fait la même chose avec $x_2 = g_w(y_2)$, et ainsi de suite.

Le but de l'apprentissage non supervisé est donc de trouver des poids w de sorte que l'ensemble des fausses images $\{x_1, x_2 \dots x_n\}$ soit « le plus proche » de l'ensemble des images $\{z_1, z_2 \dots z_n\}$ de la base de données. Le problème d'optimisation s'écrit ainsi :

$$\min_w \text{Distance}\left(\{x_1, x_2 \dots x_n\}, \{z_1, z_2 \dots z_n\}\right).$$

Les images générées, $\{z_1, z_2 \dots z_n\}$, dépendent du réseau g_w et donc des poids w. On peut reformuler le problème précédent comme suit :

$$\min_w \text{Distance}\left(\{g_w(y_1), g_w(y_2) \dots g_w(y_n)\}, \{z_1, z_2 \dots z_n\}\right).$$

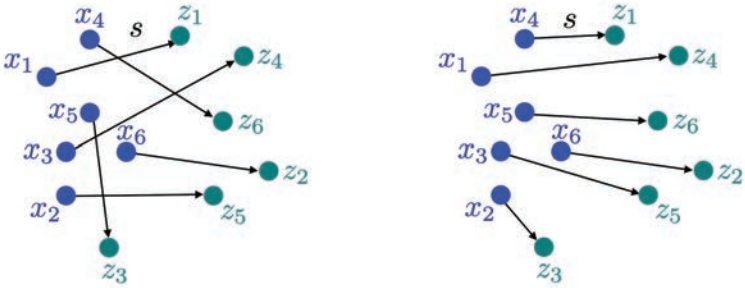
La question mathématique qui se pose est donc de définir une notion de distance entre deux ensembles de points. Il existe bien des façons de le faire ! L'une d'entre elles est particulièrement adaptée à ce problème d'apprentissage. Elle exploite la théorie du transport optimal.

Monge et le transport optimal, des tas de sable aux images

Le *problème du transport optimal* est formulé par Gaspard Monge en 1781, pour des applications militaires. Il s'agit de déterminer la façon la plus économe de transférer des objets depuis un ensemble de sources $\{x_1, x_2 \dots x_n\}$ vers un ensemble de destinations $\{z_1, z_2 \dots z_n\}$.

Pour Monge, il s'agit de transférer de la terre depuis des déblais pour créer des remblais. Mais cette question en fait trouve une multitude d'applications. Pour le problème d'entraînement de réseaux génératifs, les sources sont les images fausses générées par le réseau et les destinations sont les images de la base de données. Il s'agit alors de relier chaque source, par exemple x_1, vers un unique point de destination, que l'on va noter z_{s_1}, où s_1 est un entier entre 1 et n. De manière similaire, x_2 est relié à z_{s_2}, et ainsi de suite.

Sur la figure suivante, on a relié x_2 à z_5, ce qui signifie que $s_2 = 5$. Il faut également que chacune des n destinations soit approvisionnée par une source. Ceci signifie par exemple que x_1 et x_2 ne peuvent pas être reliés à la même destination : il faut relier toutes les sources à des destinations différentes. En d'autres termes, $\{s_1, s_2 \dots s_n\}$ doit être une permutation des n premiers entiers. Sur notre exemple simple (avec $n = 6$ éléments), on a choisi, sur la gauche, la permutation ($s_1 = 1$, $s_2 = 5$, $s_3 = 4$, $s_4 = 6$, $s_5 = 3$, $s_6 = 2$).



Exemple à gauche d'une permutation s non optimale
et à droite de la permutation optimale, dans le cas de six points en dimension 2.

Le problème de Monge consiste alors à trouver la permutation qui minimise la somme des coûts de transport. Monge a posé, pour ses besoins, que le coût de transport entre une source x et une destination z est égal à la distance euclidienne $\|x-z\|$ entre les deux points, mais on peut choisir un autre coût (temps de trajet, prix nécessaire en essence si on utilise des camions…). On doit ainsi résoudre le problème :

$$\min_{w} \text{Distance}\left(\{x_1, x_2 \dots x_n\}, \{z_1, z_2 \dots z_n\}\right).$$

Une fois que l'on a calculé une permutation $s^* = (s_1^*, s_2^* \dots s_n^*)$ optimale (donc solution du problème précédent), on définit la distance entre les ensembles de points comme la valeur du coût total de transport :

$$\text{Distance}\left(\{x_1, x_2 \dots x_n\}, \{z_1, z_2 \dots z_n\}\right) = \left\|x_1 - z_{s_1^*}\right\| + \left\|x_2 - z_{s_2^*}\right\| + \dots + \left\|x_n - z_{s_n^*}\right\|.$$

La difficulté pour calculer cette distance est que le nombre total de permutations à tester est très grand ! En effet, pour le choix de s_1 on a n possibilités, pour celui de s_2 il en reste $n-1$ (puisque la valeur de s_1 est prise), pour s_2 il y en a $n-2$… Donc le nombre total de permutations est égal à $n!$, la factorielle du nombre n, égale au produit $n(n-1)(n-2)\dots\times 2\times 1$.

Pour $n=6$, il existe donc $6!=720$ permutations possibles (faites le calcul !). Dans ce cas simple, on peut toutes les tester et choisir la meilleure, à savoir ($s_1=4$, $s_2=3$, $s_3=5$, $s_4=1$, $s_5=6$, $s_6=2$). Mais déjà pour $n=70$, on dénombre plus de 10^{100} possibilités, ce qui est à comparer aux 10^{79} atomes estimés dans l'univers connu… Et pour entraîner des réseaux de neurones, l'entier n est encore beaucoup plus grand ! Il a donc fallu attendre plusieurs révolutions mathématiques et algorithmiques pour pouvoir obtenir une méthode permettant de résoudre ce problème.

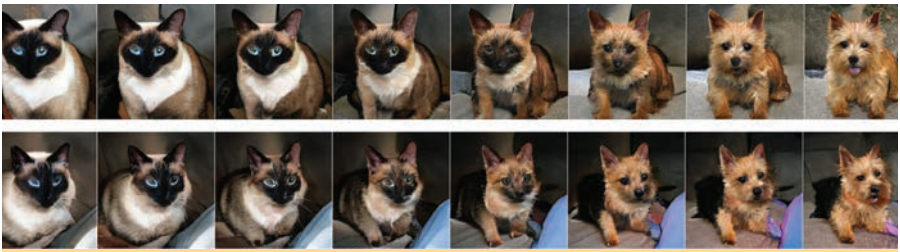
Théorie revisitée, division de la production et prix Nobel

Monge a remarqué que les solutions de son problème ont des structures très particulières. Par exemple, sur notre figure, à droite, les trajets optimaux ne se croisent pas, ce que le conte de Péluse avait prouvé dans son remarquable article. Mais cette remarque pertinente n'est pas suffisante pour résoudre le problème, car il existe encore énormément de trajectoires sans croisement.

Il a fallu plus de deux cents ans pour comprendre comment obtenir plus d'information sur les solutions afin de les calculer efficacement. C'est Leonid Kantorovitch qui a trouvé, en 1942, une nouvelle formulation du problème de transport optimal calculable rapidement. Il a autorisé chaque source à se « diviser » en plusieurs parties, par exemple deux parties égales avec une pondération de 1/2 chacune. Cette « division de la production » est intéressante car elle simplifie le problème d'optimisation. Elle est également naturelle pour les préoccupations de Kantorovitch, qui étaient de modéliser et de planifier la production en économie. Il a d'ailleurs obtenu en 1975 le prix Nobel pour cette idée.

Conjointement aux travaux pionniers de Kantorovitch, George Bernard Dantzig (1914 – 2005) a trouvé en 1947 *l'algorithme du simplexe*, qui permet de résoudre efficacement des problèmes de transport de grande taille. Sa complexité numérique pour résoudre un problème de transport optimal entre n points est de l'ordre de n^3, ce qui est beaucoup plus faible que $n!$. Cet algorithme est au cœur d'un nombre impressionnant de systèmes industriels qui doivent optimiser l'adéquation entre des moyens de production et de consommation.

On peut du coup l'utiliser également pour entraîner des réseaux de neurones génératifs ! Un court article de John Forbes Nash Jr (1929 – 2015), lui-même prix Nobel en 1994, introduit dès 1950 des algorithmes efficaces et des applications à la science des données. Depuis, l'exploration et la recherche continuent !



Deux exemples de *deep fakes* qui sont des images virtuelles interpolant entre chats et chiens.

Prendre en compte la géométrie des objets dans les images

Une difficulté pour appliquer le transport optimal pour entraîner des réseaux génératifs est qu'il faut choisir un « coût de transport entre deux images ». On pourrait calculer la distance euclidienne entre les pixels des images, mais ceci ne produit pas des résultats satisfaisants, car une telle distance ne peut pas prendre en compte la géométrie des objets présents dans les images.

Une idée très fructueuse a été introduite en 2014 par Ian Goodfellow et ses collaborateurs. Elle consiste à … utiliser un second réseau de neurones f pour déterminer ce coût de transport ! Ce nouveau réseau, nommé *réseau adversaire*, joue un rôle de discriminateur.

Alors que le but du générateur g est de générer des images fausses très ressemblantes, le but de f est au contraire de faire de son mieux pour reconnaître les vraies et les fausses images. Ces deux réseaux sont entraînés conjointement ; on parle de fait de *réseaux antagonistes*. L'entraînement de g et f correspond à un jeu à somme nulle, concept introduit en 1944 par John von Neumann (1903 – 1957) et généralisé ensuite par Nash en 1950.

Ces avancées récentes ont permis d'obtenir des résultats excellents pour la génération d'images. La figure en page précédente montre des résultats obtenus avec une méthode introduite par Andrew Brock, Jeff Donahue et Karen Simonyan, utilisée pour calculer des « chemins » d'images entre chiens et chats. Après le transport optimal et la géométrie, voici que la topologie se mobilise, ouvrant un nouveau champ à explorer !

G. P.

Pour en savoir (un peu) plus :

Mémoire sur la théorie des déblais et des remblais. Gaspard Monge, *Histoire de l'Académie royale des sciences*, 1781.

On the transfer of masses. Leonid Kantorovich, *Doklady Akademii Nauk* 37 (2), 1942.

Equilibrium points in n-person games. John Nash, *Proceedings of the National Academy of sciences* 36 (1), 1950.

Theory of games and economic behavior. Oskar Morgenstern et John von Neumann, Princeton University Press, 1953.

Origins of the simplex method. George Dantzig, dans *A history of scientific computing*, Addison-Wesley Publishing Company, 1990.

Generative adversarial nets. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville et Yoshua Bengio, dans *Advances in neural information processing systems*, 2014.

Large scale GAN training for high fidelity natural image synthesis. Andrew Brock, Jeff Donahue et Karen Simonyan, *Proceedings of the 7th international conference on learning representations*, 2019.