# Auto-encodeur variationnel: vers de nouvelles applications et une mise à jour de la théorie.

Cédric Beaulac

University of Toronto

5 novembre 2020

# Introduction

- Explorer l'apprentissage automatique: découvrir comment ces algorithmes peuvent contribuer à la statistique.
- Apprendre à connaître cette communauté de recherche.
- Visiter toutes les étapes de l'analyse de données.
- Travailler avec de vieux algorithmes bien établis (ex. forêt aléatoire) et nouveaux modèles en développement (ex. VAE).

# Auto-encodeur variationnel

# Auto-encodeur variationnel

La théorie

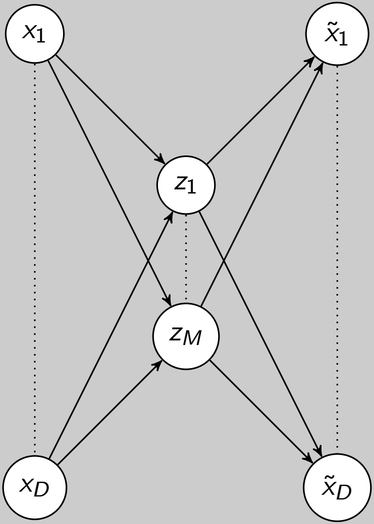
# Qu'est-ce qu'un auto-encodeur variationnel

- Modèle à variables latentes commes les chaînes de Markov cachées (HMM) ou les modèles de mélange Gaussien (GMM).
- Kingma 2013: utilisé avec succès en vision artificielle.
- Très peu connu en statistique.
- Des tonnes d'articles publiés de tout côté, plusieurs algorithmes fonctionnent mais j'aimerais m'assurer que la théorie soit mise-à-jour.

# Qu'est-ce qu'un auto-encodeur

- Un auto-encodeur est un modèle non-supervisé qui apprend à encoder ($q$) et décoder ($p$) des observations.
- Traditionnellement le code (variable latente) est de plus petite dimension $M << D$; ce modèle sert à compresser et décompresser des observations de grande dimension.
- *Notations* : $x$ est l'observation, $z$ est son code, $p$ est une fonction déterministe qui encode $x$ ($p(x) = z$) et $q$ décode ($q(z) = x$)).

# Auto-encodeur: Compression et décompression



# Auto-encodeur

- Plusieurs fonctions $p$ et $q$ et plusieurs méthodes d'optimisation possibles.
- Par exemple: si $p$ et $q$ sont des combinaisons linéaires
- et si nous voulons minimiser l'erreur de reconstruction quadratique : $\frac{1}{n}\sum_{i=1}^{n}||\mathbf{x_i} - \tilde{\mathbf{x}}_\mathbf{i}||^2$ en fonction des coefficients de $p$ et $q$,
- la solution à ce problème sont les composantes principales.

# Vers un auto-encodeur probabiliste

- Pouvons-nous en faire un modèle probabiliste ?
- Supposons des lois de probabilité:
  1. $p(z) = \mathcal{N}(0, I)$
  2. $p(x|z) = \mathcal{N}(Wz + \mu, \sigma^2 I)$
- C'est une *analyse en composantes principales probabiliste* (pPCA, Tipping & Bishop 1999).
- La distribution marginale de $x$ est normale, et les paramètres $W$, $\mu$ et $\sigma$ sont estimés par maximum vraisemblance.
- Nous pouvons aussi calculer analytiquement $p(z|x)$.

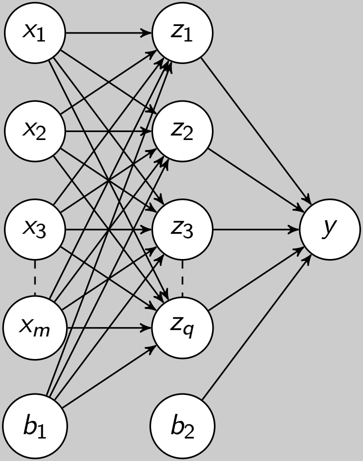
# Vers un auto-encodeur probabiliste

- Nous avons maintenant une *compression probabiliste* $p(z|x)$
- et une *décompression probabiliste* où $p(x|z)$.
- Cette formulation probabiliste offre plusieurs avantages :
  1. EM nous évite de calculer la matrice de covariance.
  2. Facilite la gestion des valeurs manquantes.
  3. Permet une formulation Bayesienne.
  4. Peut modéliser des distributions conditionnelles pour la classification.
  5. Permet la génération de nouvelles observations par échantillonnage ancestral.

# Vers un auto-encodeur variationnel

- Un auto-encodeur variationnel est une généralisation de pPCA.
- On veut permettre des fonctions $p$ et $q$ plus complexes que de simples combinaisons linéaires.
- La fonction flexible de choix est le réseau de neurones (neural network (NN)).
- Composition de transformations linéaires paramétriques et transformations non linéaires non paramétriques.
- Facile à optimiser par rétropropagation (back-propagation) du gradient.
- Considéré (et démontré) comme étant un *estimateur de fonctions universel*.

# Réseau de neurones simple: représentation graphique



# Réseau de neurones simple: représentation fonctionnelle

$$\mathbf{z} = f\left(\mathbf{B}_1\mathbf{x}\right) \tag{1}$$

où $\mathbf{B}_1$ est un matrice de coefficients et $f$ est une fonction non linéaire. Par exemple: $f(a) = \frac{1}{1+e^{-a}}$ . Si $y$ est une variable binaire, alors:

$$\tilde{y} = \text{logit}\left(\mathbf{B}_2 f\left(\mathbf{B}_1\mathbf{x}\right)\right) \tag{2}$$

On peut calculer le gradient d'une erreur par rapport au paramètres (coefficients de $\mathbf{B}_1$ et $\mathbf{B}_2$) par rétropropagation.

# Vers un auto-encodeur variationnel

- Supposons:
  1. $p_\theta(z) = \mathcal{N}(0, I)$
  2. $p_\theta(x|z) = \mathcal{N}(\mu_x, \sigma_x^2 I)$ où $\mu_x = NN_1(z)$ et $\sigma_x = NN_2(z)$.
- La moyenne et la variance observationnelle sont le résultat d'un réseau de neurones prenant $z$ en entrée.
- $\theta$ est l'ensemble des paramètres à estimer de la distribution $p$.
- *Notation* : $\theta = \{\mu_x(z), \sigma_x(z)\}$

# Vers un auto-encodeur variationnel

- Cela nous permet de représenter/capturer des distributions marginales pour $x$ bien plus complexes sans augmenter la dimension de $z$.
- La distribution a posteriori $p_\theta(z|x)$ ne se calcule pas analytiquement.
- Nous utilisons des idées bayésiennes variationnelles, nous approximons $p_\theta(z|x)$ par $q_\varphi(z|x)$.
- $\varphi$ sont les paramètres des distributions approximatives.
- Avec $q_\varphi(z|x) = N(\mu_z, \sigma_z^2 I)$ alors $\varphi = \{\mu_z(x), \sigma_z(x)\}$ sont aussi des réseaux de neurones.

# ELBO

- Il est impossible de maximiser directement $\log p(x)$ ou d'utiliser EM ($p_\theta(z|x)$ est insoluble).
- La solution est d'optimiser une borne inférieure de $\log p(x)$, le ELBO (*Evidence Lower BOund*).

# ELBO

$$
\begin{aligned}
\log p(x) &= \mathbf{E}_{q(z|x)}[\log p(x)] \\
&= \mathbf{E}_{q_\varphi(z|x)}\left[\log\left(\frac{p(x,z)}{p(z|x)}\right)\right] \\
&= \mathbf{E}_{q(z|x)}\left[\log\left(\frac{p(x,z)q(z|x)}{q(z|x)p(z|x)}\right)\right] \\
&= \mathbf{E}_{q(z|x)}\left[\log\left(\frac{p(x,z)}{q(z|x)}\right)\right] - \mathbf{E}_{q(z|x)}\left[\log\left(\frac{p(z|x)}{q(z|x)}\right)\right] \\
&= \mathcal{L}(q_\varphi, p_\theta) + KL(q_\varphi || p_\theta).
\end{aligned} \tag{3}
$$

# ELBO

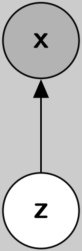
$$\mathcal{L}(q_\varphi, p_\theta) = \mathbf{E}_{q_\varphi(z|x)}\left[\log p_\theta(z) + \log p_\theta(x|z) - \log q_\varphi(z|x)\right] \quad (4)$$

- La différence entre $\log p(x)$ et $\mathcal{L}(q_\varphi, p_\theta)$ est $KL(q_\varphi || p_\theta)$
- Il est impossible de calculer cette intégrale, nous l'estimons par Monte-Carlo.

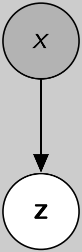
# VAE : Algorithme

| **Algorithme** : Entraîner VAE($x$) |
|---|
| 1) Entrer les observations $x$ dans le NN $\varphi$ |
| pour obtenir $\mu_z(x)$ et $\sigma_z(x)$ |
| 2) Échantilloner $z$ de $q_{\varphi(x)}(z\|x)$ |
| 3) Entrer l'échantillon $z$ dans le NN $\theta$ |
| pour obtenir $\mu_x(z)$ et $\sigma_x(z)$ |
| 4) Évaluer $\log p_\theta(z) + \log p_\theta(x\|z) - \log q_\varphi(z\|x)$ |
| 5) Maximiser l'estimation ELBO (algorithme du gradient) |
| par rapport aux paramètres de $\varphi$ et $\theta$ |
| Répéter 1-5 jusqu'à convergence. |

# VAE: Représentation graphique




(a) Composante générative $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p(\mathbf{x}|z)$.




(b) Composante d'inférence. Étant donné $x$ nous avons $q(\mathbf{z}|x)$.

Figure: Représentation graphique des deux composantes formant un VAE

# VAE: les utilisations

- Compression. Encodage, stockage et analyse de l'espace latent.
- Génération. Générer de nouvelles observations par échantillonnage ancestral: $z \sim p_\theta(z)$ puis $x \sim p_\theta(x|z)$.
- Classification et régression. Le modèle peut être adapté pour ces tâches.

# Auto-encodeur variationnel

Application à l'analyse de survie

# Introduction

- Nous avons obtenu des données du *Children's Oncology Group*.
- Pour les 1 712 patients, un ensemble de caractéristiques et symptômes sont collectés ainsi que le traitement et la réponse.
- La réponse est le temps avant un évènement et est censurée à droite pour la majorité des patients.
- Nous voulons créer un système qui peut recommander un traitement.
- Recherches publiées au *NeurIPS 2018 ML4H workshop* et dans *Applied Artificial Intelligence*

# Notre modèle: SAVAE (Survival Analysis VAE)


(a) Composante générative. Suppose $p(x, y, t, z) = p(z)p(x|z)p(t|x)p(y|t, z)$.

(b) Composante d'inférence. Étant donné $x$ et $y$ nous avons $q(z|x, y)$.

Figure: Représentation graphique du SAVAE. La réponse est identifiée par $y$, le traitement par $t$, les caractéristiques par $\mathbf{x}$ et la variable latente représente le réel état de santé $\mathbf{z}$.

# SAVAE

$$\begin{aligned}\text{ELBO} &= \mathbf{E}_{q_\varphi}\left[\log \frac{p_\theta(\mathbf{x}, t, y, \mathbf{z})}{q_\varphi(\mathbf{z}|\mathbf{x}, y)}\right] = \mathbf{E}_{q_\varphi}\left[\log p_\theta(\mathbf{x}, t, y, \mathbf{z}) - \log q_\varphi(\mathbf{z}|\mathbf{x}, y)\right] \\ &= \mathbf{E}_{q_\varphi}[\log p_\theta(\mathbf{z}) + \log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(t|\mathbf{x}) + \log p_\theta(y|t, \mathbf{z}) \\ &\quad - \log q_\varphi(\mathbf{z}|\mathbf{x}, y)].\end{aligned} \tag{5}$$

où

$$\log p_\theta(y|t, \mathbf{z}) = \delta \log f_\theta(y|t, \mathbf{z}) + (1-\delta) \log S_\theta(y|t, \mathbf{z}), \tag{6}$$

avec $\delta = 1$ si $y$ est observé et 0 si $y$ est censuré.

# SAVAE

Nous pouvons décider des distributions. Par exemple :

$$p_\theta(\mathbf{x}|z) = \prod_{j=1}^{D_x} p_\theta(x_j|z) \tag{7}$$

$$p(\mathbf{t}_i|x) = \text{Ber}(\hat{\pi}_i) \text{ pour } i \in \{1, 2\}. \tag{8}$$

$$p(\mathbf{y}|t, z) = \text{Weibull}(\lambda, K) \tag{9}$$

$$\theta = f_2(\mathbf{B}_2 f_1(\mathbf{B}_1 z)) \tag{10}$$

$$[\pi_1, \pi_2] = f_4(\mathbf{B}_4 f_3(\mathbf{B}_3 x)) \tag{11}$$

$$[\lambda, K] = f_6(\mathbf{B}_6 f_5(\mathbf{B}_5 [t, z])) \tag{12}$$

# SAVAE

$$q(\mathbf{z}|x, y) = \mathcal{N}(\mu, \sigma^2 I) \tag{13}$$

$$[\mu, \sigma] = f_8(\mathbf{B}_8 f_7(\mathbf{B}_7[x, y])). \tag{14}$$

# SAVAE

Finalement nous obtenons $p(y|t, x)$ par échantillonnage préférentiel:

$$p(y|t, x) \approx \sum_{l=1}^{L} w_l p_\theta(y|t, z_l) \tag{15}$$

où :

$$w_l = \frac{p_\theta(x|z_l)}{\sum_{k=1}^{L} p_\theta(x|z_k)} \tag{16}$$

# Résultats

- Performe mieux que la régression de Cox selon l'indice Brier, une erreur quadratique généralisée pour l'analyse de survie.
- Nous obtenons une distribution Weibull pour chaque patient et traitement.
- Cela permet aux médecins d'établir plusieurs manières de choisir le traitement.

# Résultats

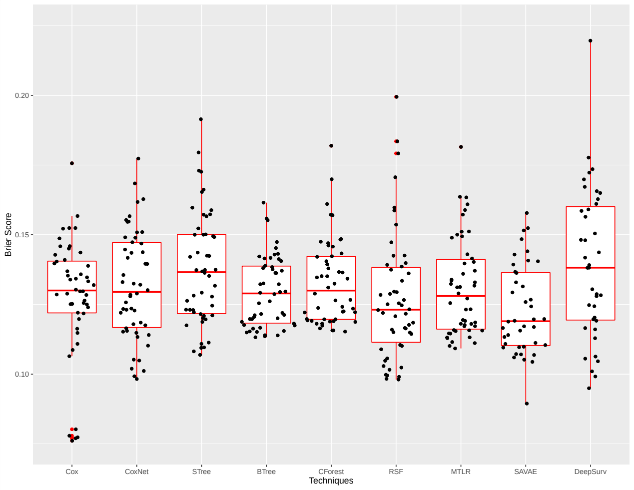



Figure: Comparaison de plusieurs algorithmes pour l'analyse de survie.

# Auto-encodeur variationnel

## Application en vision artificielle

# Introduction

- Nouveau projet tout juste soumis!
- Vision artificielle est un sujet qui me passionne.
- Reflète bien ma recherche actuelle.
- Contributions: un nouveau jeu de données ET une analyse.
- Article soumis au *International Journal of Computer Vision*

# Motivations

Inspiré par le célèbre *MNIST data set*.

Figure: Échantillon provenant du *MNIST data set*.

# Motivations

À l'aide VAE on observe que les chiffres de style similaire sont près les uns des autres.

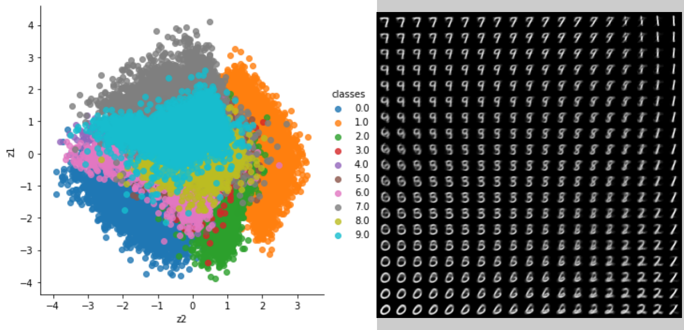



Figure: Représentation latente du *MNIST data set*.

# Motivations

- Les *styles* d'écriture dépendent de la personne qui l'écrit. Peut-on l'estimer ?
- MNIST est trop simple, contient des images de petite résolution et seulement le chiffre comme variable expliquée. La prédiction est trop facile.
- Nous voulons collecter notre propre base de données:
  1. Peut-on déterminer qui écrit les chiffres ?
  2. Peut-on faire de l'inférence sur d'autres caractéristiques ?
  3. Peut-on générer le chiffre de notre choix avec le *style* de notre choix ?

# Collecte de données

- Objectif: 200-300 étudiants de l'Université de Toronto.
- Salles de classe réservées.
- COVID contre-attaque!
- Envois postaux: Moins de données (97) et données moins diversifiées.

# Collecte de données

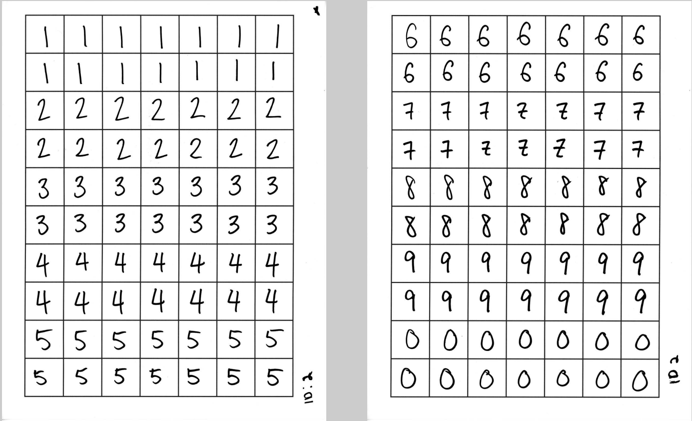


Figure: Exemple de feuilles de données.

# Données : détails

- 97 auteurs, 14 répliques de chacun des 10 chiffres pour un total de 13 580 images de haute résolution (500 x 500).
- Les variables attachées sont: chiffre, ID, âge, genre biologique, taille, langue première, main forte, niveau d'éducation et médium d'écriture principale.
- Disponible publiquement sur mon site web.
- Plusieurs formats.

# Données : un échantillon

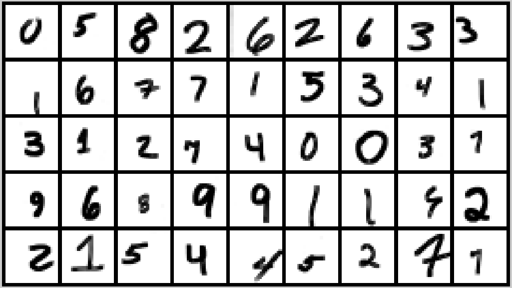


Figure: Échantillon de 45 images.

# Les questions spécifiques à notre base de données

1. Peut-on prédire le chiffre (tâche facile), l'ID (plus difficile) et les autres caractéristiques?
2. Quel est l'impact de la résolution?
3. Peut-on faire des prédictions semi-supervisées ?
4. Peut-on faire de la génération d'images contrôlée ?

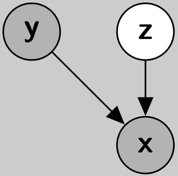
# Résultats

- Pour (1) et (2), notre base de données offre de nouvelles opportunités comparativement à MNIST.
- Une belle diversité de force de signal.
- Découverte intéressante par rapport au gain d'une haute résolution.
- Concentrons-nous sur les applications des VAEs.

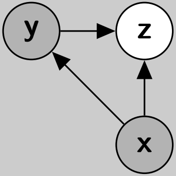
# Analyse semi-supervisée

- Peut-on intégrer des données sans réponse $(S_u)$ à une base de données existante avec réponses $(S_l)$ pour aider l'estimation.
- Notre base de données est différente du *MNIST data set*, mais suffisamment semblable pour ce type d'expérience.
- Nous pouvons vérifier si nos prédictions sont plus précises en ajoutant MNIST pendant l'apprentissage.
- Nous utilisons le VAE M2 (Kingma 2014)

# VAE: Modèle M2




(a) Composante générative. Suppose $p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{y}) = p_\theta(\mathbf{z}) p_\theta(\mathbf{y}) p_\theta(\mathbf{x}|z, y)$.




(b) Composante d'inférence. Étant donné $x$ et $y$ nous pouvons obtenir $q_\varphi(\mathbf{z}|x, y)$. Si $y$ est manquant nous pouvons l'estimer par $q_\varphi(\mathbf{y}|x)$.

Figure: Représentation graphique du modèle M2.

# VAE: Modèle M2

$$\begin{aligned}\log p_\theta(\mathbf{x},\mathbf{y}) &\geq \mathbf{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y})}\left[\log p_\theta(\mathbf{z}) + p_\theta(\mathbf{y}) + p_\theta(\mathbf{x}|z,y) - \log q_\varphi(\mathbf{z}|x,y)\right] \\ &= \mathcal{L}(x,y)\end{aligned} \tag{17}$$

$$\begin{aligned}\log p_\theta(\mathbf{x}) &\geq \mathbf{E}_{q(\mathbf{z},\mathbf{y}|x)}\left[\log p_\theta(\mathbf{z}) + p_\theta(\mathbf{y}) + p_\theta(\mathbf{x}|z,y) - \log q_\varphi(\mathbf{z},\mathbf{y}|x)\right] \\ &= \sum_y \left[q_\varphi(\mathbf{y}|x)(\mathcal{L}(x,y))\right] + \mathcal{H}(q_\varphi(\mathbf{y}|x)) \\ &= \mathcal{U}(x)\end{aligned} \tag{18}$$

$$\mathcal{J} = \sum_{S_l} \mathcal{L}(x,y) + \sum_{S_u} \mathcal{U}(x) \tag{19}$$

# VAE: Modèle M2

Par contre, de cette manière on entraîne la fonction de classification $q_\varphi(\mathbf{y}|x)$ (ici un CNN) seulement sur les données sans réponse. La solution proposée (Kingma 2014) est de modifier la fonction objectif :

$$\mathcal{J}^\alpha = \mathcal{J} + \alpha \mathbf{E}_{S_l}\left[-\log q_\varphi(\mathbf{y}|x)\right] \tag{20}$$

# Classification semi-supervisée: résultats

| | CNN | | M2 | |
|---|---|---|---|---|
| | Moyenne | É.-T. | Moyenne | É.-T. |
| Chiffre | 0.9399 | 0.0143 | **0.9542** | 0.0060 |
| ID | 0.3473 | 0.0136 | **0.4174** | 0.0099 |

Table: Précision de prédiction pour deux problèmes de classification.

# Projet en cours

$$\mathcal{J}^{\alpha} = \alpha\mathcal{J} + \mathbf{E}_{S_l}\left[-\log q_{\varphi}(\mathbf{y}|x)\right] \tag{21}$$

Lorsque $\alpha = 0$ nous retrouvons exactement l'ancien problème.

Notre hypothèse est que le modèle utilise la *machinerie* du VAE comme régularisation (pénalisation).

# Génération

- Les VAE ont une composante générative. Le modèle $p$ est complètement défini: nous permet de générer $x$.
- Dans ce cas-ci, il s'agit de générer de nouvelles images.
- Avec un VAE simple $z \sim p(z)$ puis $x \sim p(x|z)$.
- Ce processus génère toute sorte de chiffres aléatoires avec des styles tout aussi aléatoires.

# Génération contrôlée

- Peut-on décider du chiffre ou du style ? Oui grâce au VAE conçu pour la classification telle que le modèle M2 introduit précédemment.
- Dans ce cas-ci: nous fixons $y$ puis $z \sim p(z)$ et $x \sim p(x|z, y)$.
- *Hypothèse:* Si un signal existe entre une variable et son image, nous pouvons utiliser cette variable lors de la génération.
- Nous avons présenté des résultats dans notre plus récent article.

# Génération contrôlée : Résultats

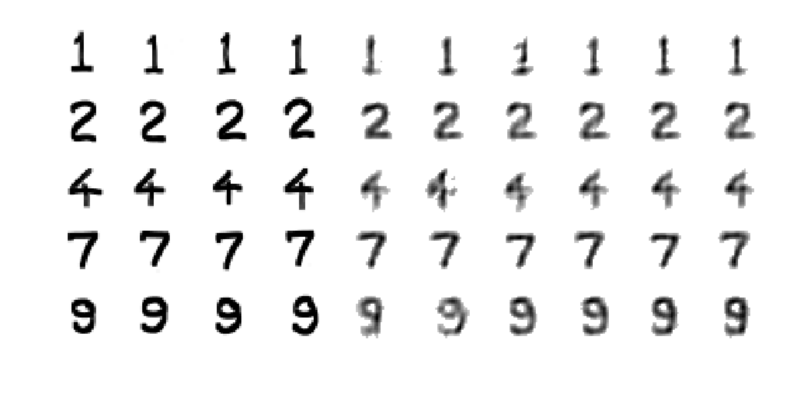


Figure: Exemple de génération d'images contrôlée.

# Génération contrôlée : Résultats

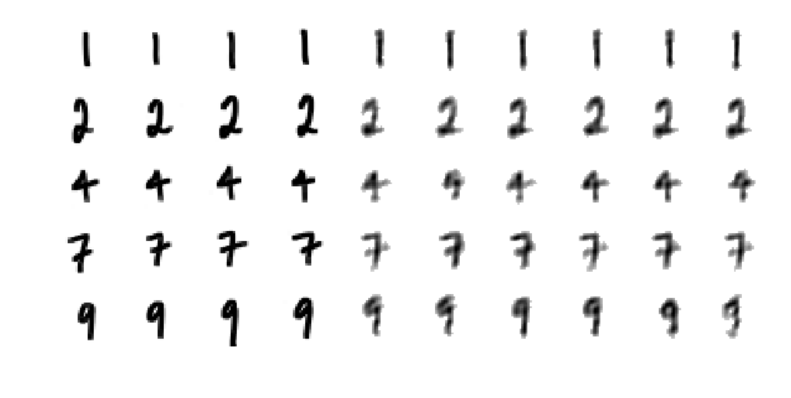


Figure: Exemple de génération d'images contrôlée.

# Génération contrôlée : Résultats

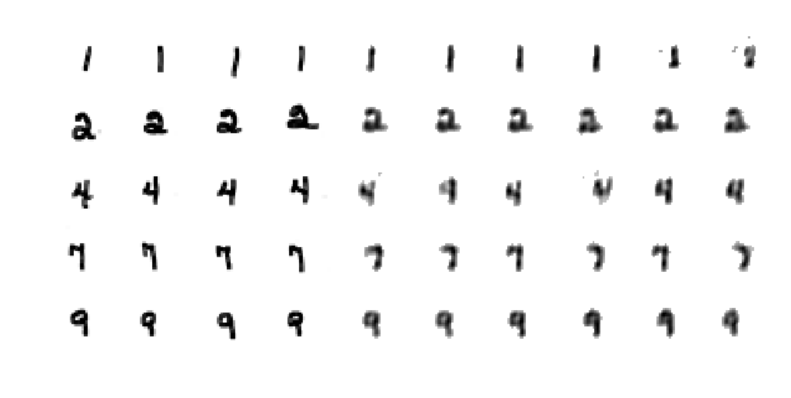


Figure: Exemple de génération d'images contrôlée.

# Génération contrôlée : Résultats

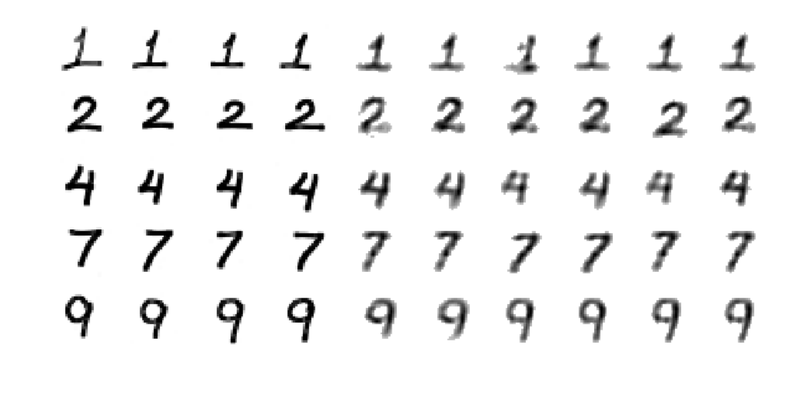
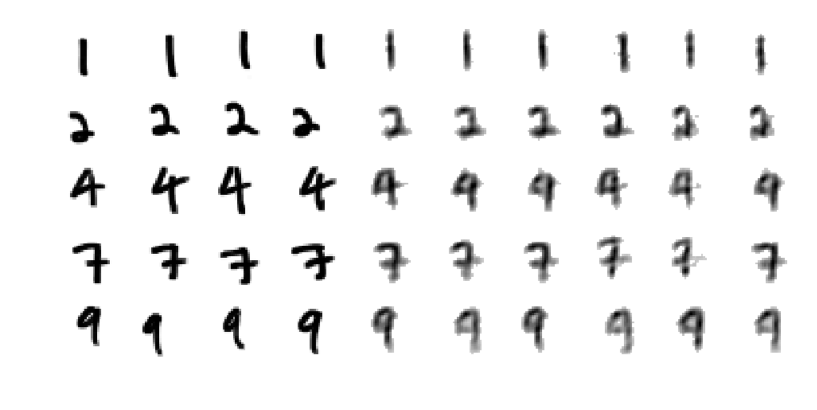


Figure: Exemple de génération d'images contrôlée.

# Génération contrôlée : Résultats



## Génération contrôlée : suite du projet

- Modèles conçus pour l'analyse semi-supervisée: une grande précision est atteinte avec peu d'observations annotées.
- *Hypothèse:* Si un signal existe entre une variable et son image, nous pouvons utiliser cette variable lors de la génération.
- *Notre idée*: Nous voulons utiliser ce principe pour décider des caractéristiques de l'image que l'on contrôle.
- *Exemple*: Télécharger des images de ciel sur internet et assigner une variable binaire pour caractériser *dégagé* ou *nuageux*.
- S'il existe un signal, nous pourrons générer une image d'un ciel et contrôler si celui-ci est dégagé ou nuageux.

# Génération contrôlée : futur projet

- Nous voulons *mathématiser* certains concepts . Comment évaluer notre *contrôle* ?
- Par exemple, *la force du contrôle* peut être évaluée à l'aide de l'information mutuelle :

$$\begin{aligned} I(Y,X) &= \int_y \int_x p(y,x) \log \frac{p(y,x)}{p(y)p(x)} dxdy \\ &= H(X) - H(X|Y) \end{aligned} \tag{22}$$

# Génération contrôlée : futur projet

- Nous sommes aussi intéressés par l'interpolation et l'extrapolation
- et par l'interprétation et la séparabilité des variables de contrôle.
- Nous voulons fournir une définition mathématique de ces composantes en parallèle d'une définition intuitive.

# Auto-encodeur variationnel

## Différence entre les implémentations courantes et la théorie

# Différences entre implémentations courantes et théorie

- De nombreuses implémentations actuelles fonctionnent
- mais ceux-ci ne respectent pas exactement la théorie.
- Nous voulons analyser la situation, nous voulons:
  1. démontrer en quoi la théorie n'est plus respectée,
  2. comprendre quels sont les problèmes résolus par ces implémentations
  3. et suggérer des modifications à la théorie pour s'y attaquer différemment.

# Modification en pratique courante

Nous allons discuter trois modifications majeures faites lors de l'implémentation :

- $\beta$-VAE: Ajuster la régularisation du modèle
- Modifier la distribution observationnelle.
- Modifier la procédure d'échantillonnage.

# $\beta$-VAE

Rappel :

$$\begin{aligned}\mathcal{L}(\varphi, \theta) &= \mathbf{E}_{q_\varphi(z|x)}\left[\log p_\theta(z) + \log p_\theta(x|z) - \log q_\varphi(z|x)\right] \\ &= \underbrace{\mathbf{E}_{q_\varphi(z|x)}\left[\log p_\theta(x|z)\right]}_{\text{Vraisemblance}} - \underbrace{KL\left(q_\varphi(z|x) | p_\theta(z)\right)}_{\text{Régularisation}}\end{aligned} \tag{23}$$

# $\beta$-VAE

L'objectif est de se donner le pouvoir de contrôler le ratio entre ces deux composantes :

$$\mathbf{E}_{q_\varphi(z|x)}\left[\log p_\theta(x|z)\right] - \beta KL\left(q_\varphi(z|x)|p_\theta(z)\right) \tag{24}$$

Pour améliorer la reconstruction, il est proposé de sélectionner un $\beta < 1$.

Cela diminue l'effet de la régularisation et permet $q_\varphi(z|x)$ d'être plus variable.

# $\beta$-VAE

Nous démontrons que cette solution est problématique pour trois raisons :

- La fonction objectif n'est plus une borne inférieure de $\log p(x)$.
- $\beta$ est un nouveau hyper paramètre (encore!) difficile à fixer.
- Le manque de régularisation nuit aux capacités génératrices du VAE.

# $\beta$-VAE: cas limite

Discutons le cas limite où $\beta = 0$.

- Produis les meilleures reconstructions.
- Élimine complètement la composante probabiliste sur $z$.
- Se rapproche d'un AE.

# Distribution observationnelle

Rappel :

$$\mathcal{L}(\varphi, \theta) = \underbrace{\mathbf{E}_{q_\varphi(z|x)}\left[\log p_\theta(x|z)\right]}_{\text{Vraisemblance}} - \underbrace{KL\left(q_\varphi(z|x) | p_\theta(z)\right)}_{\text{Régularisation}} \tag{25}$$

- Les implémentations en ligne (tutoriel officiel de PyTorch & TensorFlow) remplacent simplement $\mathbf{E}_{q_\varphi(z|x)}\left[\log p_\theta(x|z)\right]$ par l'erreur quadratique $(x - \mu_x(z))^2$.
- Cela fixe la variance observationnelle.

# Échantillonnage

Échantillonnage ancestral:

1. Échantillonner $z$ de $\mathcal{N}(0, I)$
2. Passer $z$ dans le NN $\theta$ pour obtenir $\mu_x(z)$ et $\sigma_x(z)$
3. Échantillonner $x$ de $\mathcal{N}(\mu_x(z), \sigma_x(z))$
4. Retourner $x$

Échantillonnage effectué en pratique:

1. Échantillonner $z$ de $\mathcal{N}(0, I)$
2. Passer $z$ dans le NN $\theta$ pour obtenir $\mu_x(z)$ et $\sigma_x(z)$
3. Retourner $\mu_x(z)$.

Ceci ignore la variance observationnelle lors de la génération de nouvelles observations.

# VAE: notre perspective

- Si nous retournons $\mu_x(z)$ lors de la génération
- et nous obtenons la fonction de reconstruction $\mu$ en minimisant $\frac{1}{n}\sum_{i=1}^{n}||\mathbf{x_i} - \mu_x(z_i)||^2$
- $x|z$ devient une fonction déterministe de $z$; $x|z$ n'est plus aléatoire
- Nous avons complètement éliminé la composante probabiliste de $x$ et nous nous rapprochons d'un AE traditionnel.

# VAE: notre perspective

- En combinant ces trois modifications nous éliminons la composante probabiliste sur le code $z$ et la variable observée $x$.
- Ces modifications transforment un VAE en AE.
- Notre constat est que les trois modifications discutées influencent tous comment la variance est distribué entre les deux composantes du modèle.

# VAE: pistes de solution

- *Hypothèse:* C'est un problème d'identifiabilité de la variance.
- Dans un modèle à variables latentes, la variance observationnelle est divisée en deux parties: $z$ et $x|z$.
- Nous aimerions faire face à ce problème d'une manière théorique.

# VAE: pistes de solution

1. Il faut trouver une manière d'optimiser naturellement comment diviser la variabilité. *On analyse comment pPCA optimise le ELBO.*
2. Permettre à la variabilité observationnelle de s'exprimer différemment: utiliser une matrice de covariance plutôt qu'une simple diagonale principale. *On s'inspire des modèles de statistique spatiale.*

# VAE: une solution théorique

Nous croyons qu'il est important de conserver les garanties théoriques et les composantes probabilistes du modèle pour:

1. mieux saisir la variabilité naturelle de certaines données
2. permettre au modèle de mieux se généraliser à de nouvelles applications
3. pouvoir s'appuyer sur la théorie si l'on rencontre des problèmes
4. et conserver la capacité générative du modèle.

# Merci!

Beaulac, C., Rosenthal, J. S., & Hodgson, D. (2018). A deep latent-variable model application to select treatment intensity in survival analysis. MI4H Workshop, NeurIPS 2018.

Beaulac, C., Rosenthal, J. S., Pei, Q., Friedman, D., Wolden, S., & Hodgson, D. (2020). An evaluation of machine learning techniques to predict the outcome of children treated for Hodgkin-Lymphoma on the AHOD0031 trial. Applied Artificial Intelligence, 1-15.

Beaulac, C. & Rosenthal (2020). Analysis of a high-resolution hand-written digits data set with writer characteristics, pre-print.

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. Proceedings of the 2nd International Conference on Learning Representations (ICLR)

Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In Advances in neural information processing systems (pp. 3581-3589).

Tipping, M. E., & Bishop, C. M. (1999). Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 61(3), 611-622.