UNIVERSITÉ LAVAL

GLO-4030/7030 APPRENTISSAGE PAR RÉSEAUX DE NEURONES PROFONDS

Auto-encodeurs

Hiver 2020

Philippe Giguère

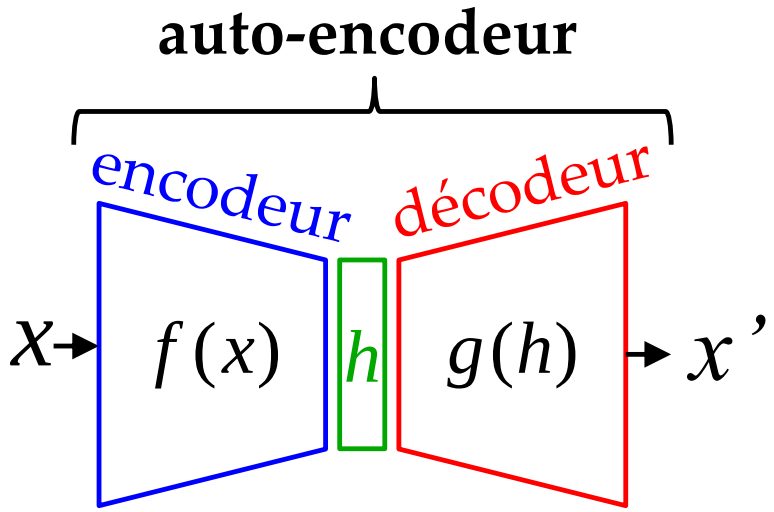
Problèmes des données étiquetées

- Jusqu'à maintenant, toujours supervisé
- Nécessite beaucoup de données étiquetées
- **Que faire si beaucoup de données, mais ne sont pas étiquetées ?**
- Apprentissage supervisé → non-supervisé
  - Supervisé + non-supervisé = semi-supervisé
- Pas juste d'ordre pratique
  - théorie de l'apprentissage en général

Pertes

- **Supervisé** : perte basée sur l'erreur entre prédiction et vérité-terrain (+régularisation)
- **Non-supervisé** : erreur basée sur la reconstruction de l'entrée *x*

Bottleneck : Conceptuellement similaire à PCA, conv. 1x1

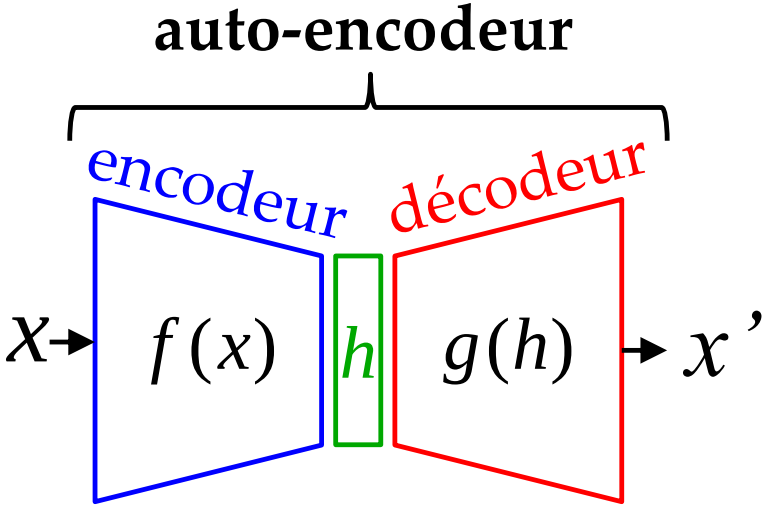


Pour éviter des solutions inintéressantes

Perte : $L = \left| g(f(x)) - x \right|^2$ + régularisation

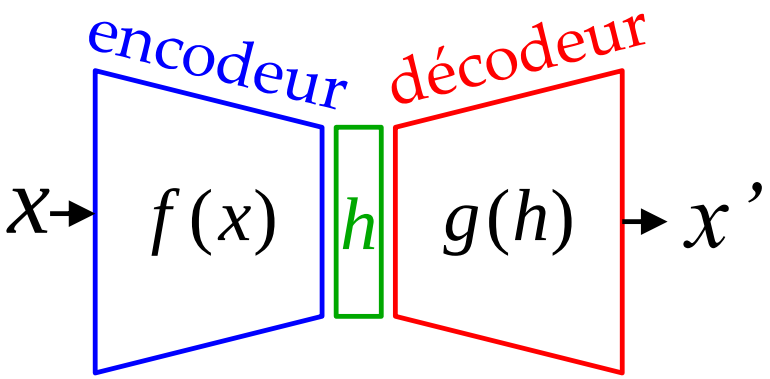
Architectures

- Autrefois : linéaire ou non-linéaire (sigmoïde)
- Puis : profond + pleinement connecté
- Maintenant : ReLU, Convolution, Déconvolution (upconv)



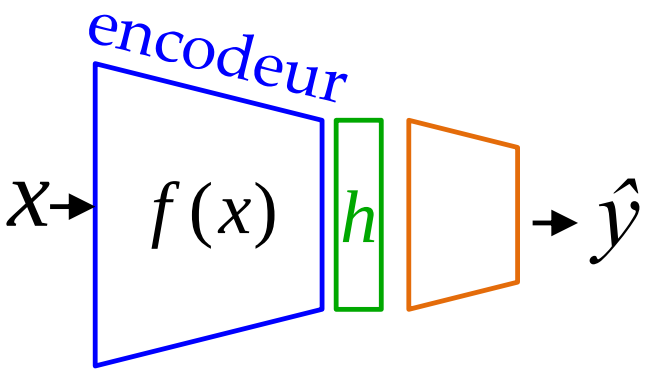
Comme pré-entraînement

- Utilise la grande quantité de données non-étiquetées pour entraîner l'encodeur-décodeur
- Retire le décodeur, remplace par classificateur (fully-connected, softmax, etc)



Comme pré-entraînement

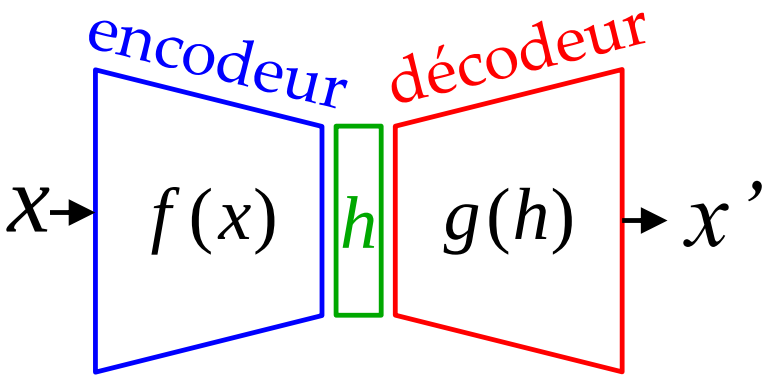
- Utilise la grande quantité de données non-étiquetées pour entraîner l'encodeur-décodeur
- Retire le décodeur, remplace par classificateur (fully-connected, softmax, etc)



Faire un fine-tuning avec les données étiquetées

Taxonomie

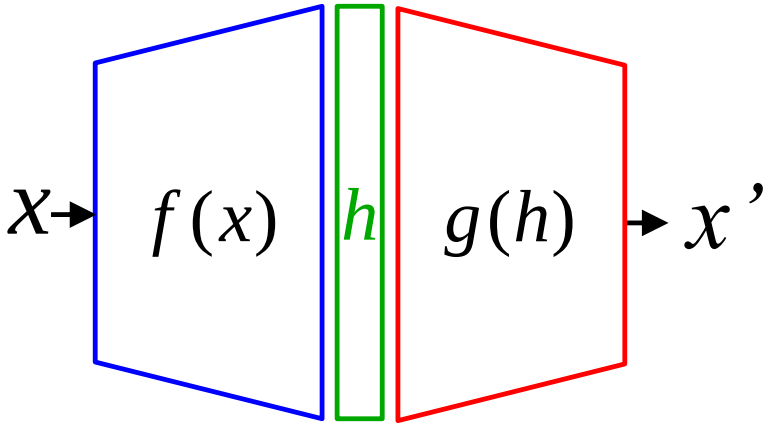
Undercomplete



taille x > taille h

- encodeur doit trouver une projection vers un espace de plus petite dimension
- si f, g sont linéaires : proche de PCA (exactement si $f=U^T$, $g = U$, $U^TU=I$)
- si f, g sont non-linéaires, projections plus puissantes

Overcomplete



taille x < taille h

- sera inutile sans régularisation
  - copie de x dans h
- exemple : x bruité

Importance de la régularisation

- Sans régularisation, l'encodage h pourrait être inutile
  - Perte de reconstruction n'influence pas directement l'**utilité** de l'encodage h
  - Cas pathologique théorique : encodeur-décodeur très puissant, taille $h = 1$

$$x \rightarrow \text{indice } i \rightarrow x$$

(rappel : réseaux profonds peuvent apprendre par cœur des jeux de données)

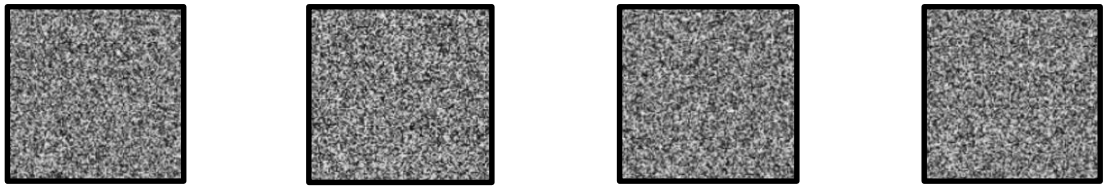
- Priorisation de certains aspects pertinents de x sera souvent **utile à d'autres tâches**
- Régularisation explicite préférable à diminuer la capacité des réseaux f, g

(note : les animations sont légèrement décalées par rapport à la narration)

Variété (*manifold*)

- Principe d'apprentissage machine
- La plupart des données réelles vont résider dans des sous-régions particulières de l'espace de *x*

pixels pigés au hasard : occupe tout l'espace de x



Grande

vs.

visage (structuré): n'occupe qu'un petit espace de x

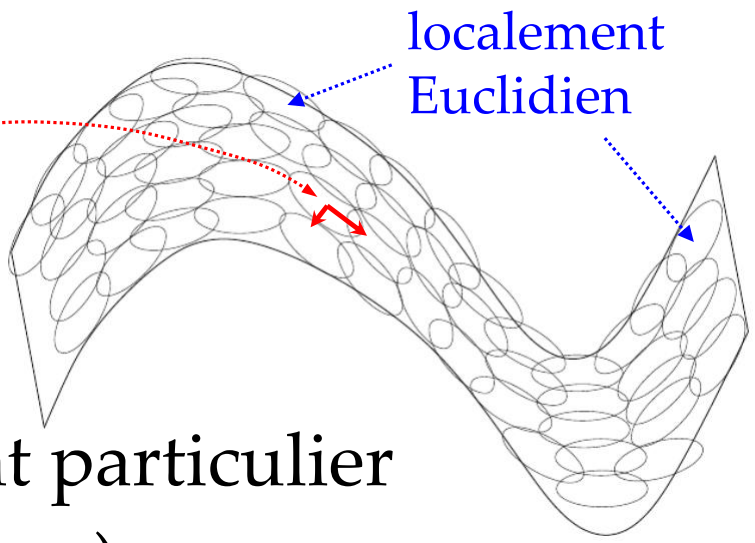
Faible

Information (entropie)

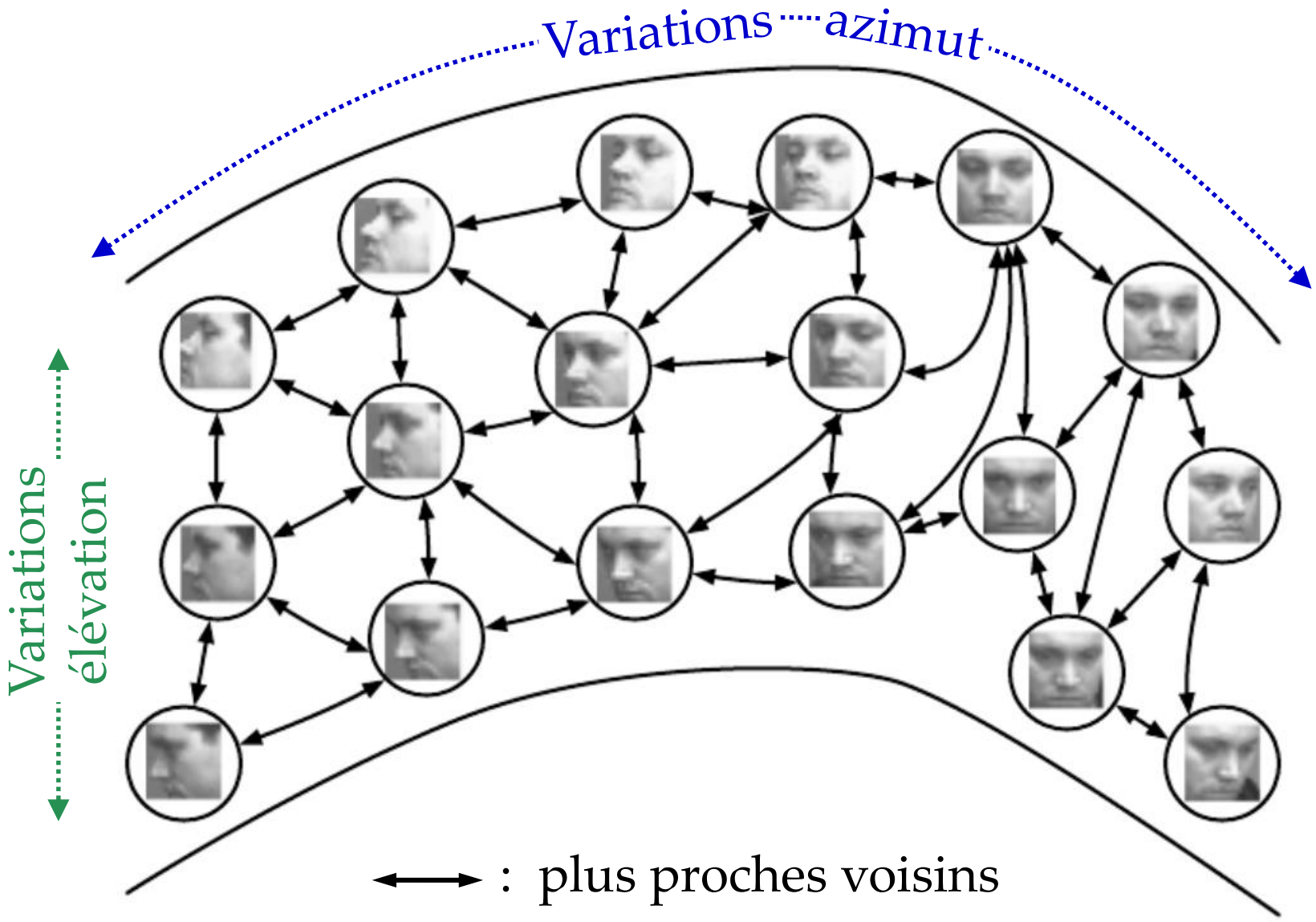
- Compression de *x* possible car réseau n'a pas à gérer les cas en dehors du manifold

Régularisation vs. manifold

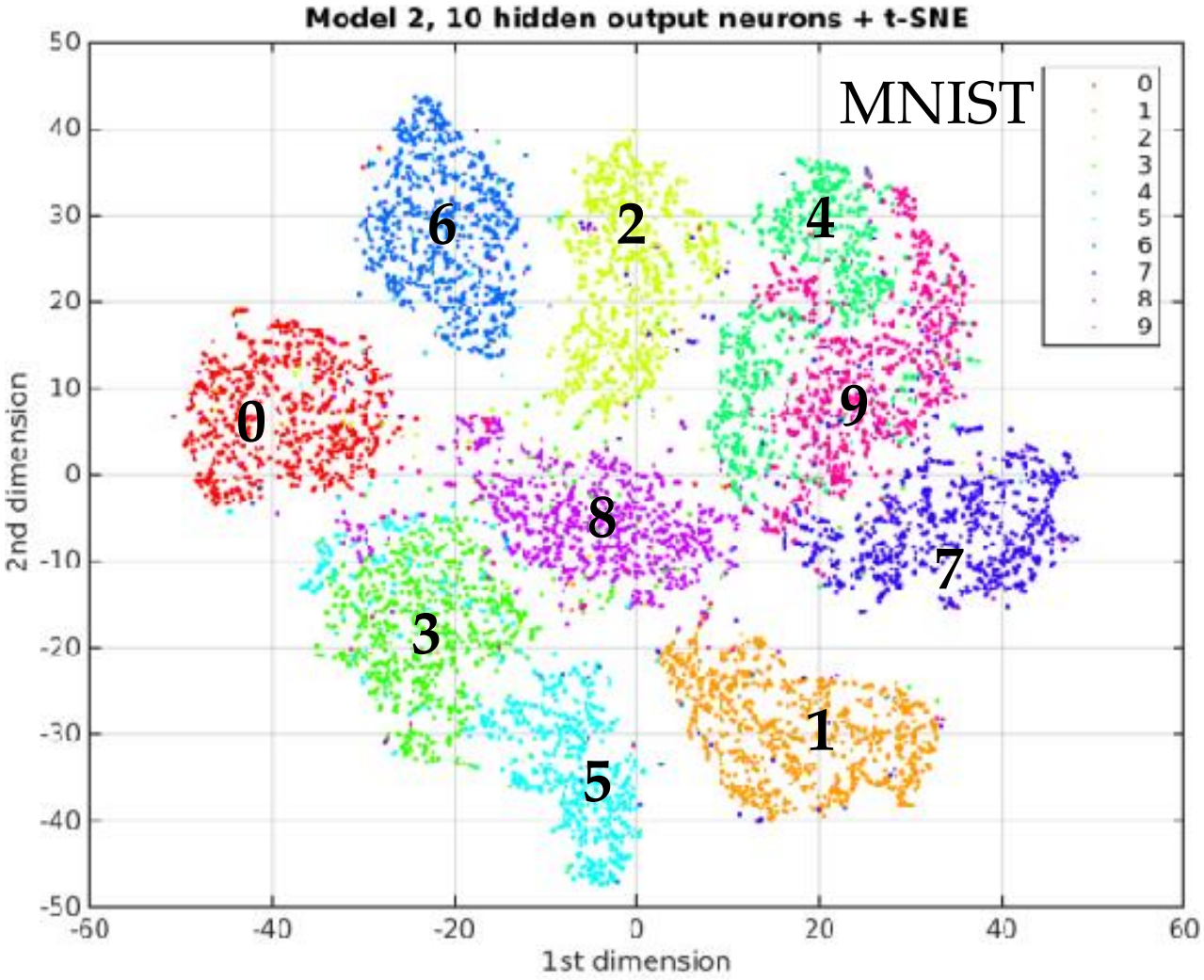
- Doit régulariser la perte de reconstruction pour **espérer apprendre ce manifold**
- Idéalement, l'encodeur trouvera les variations pertinentes dans ce manifold
  - apprendre la « surface » du manifold (tangente)
- Formuler l'**entrainement** ou l'**architecture** pour encourager un comportement particulier (générer (VAE), débruiter, etc…)
- On ne s'assure pas des performances du réseau en dehors du manifold



Exemple de manifold



t-SNE sur vecteur *h* taille 10



L.J.P. van der Maaten and G.E. Hinton. **Visualizing High-Dimensional Data Using t-SNE**. *Journal of Machine Learning Research* 9(Nov):2579-2605, 2008.

Importance de la visualisation

Faites du code de visualisation des données pour chaque étape de votre pipeline de traitement

- Comme si vous vouliez expliquer votre algorithme à un néophyte
- Permettra de :
  - l'expliquer!
  - faire des schémas explicatifs (articles, rapport)
  - mais surtout, débugger

Familles auto-encodeurs (AE)

- Sparse
- Denoising
- Contractive
- Variational (VAE)

Auto-encodeur sparse

- Perte supplémentaire sur le code *h* :

$$J(\theta) = L\left(x, g(f(x))\right) + \Omega(h)$$

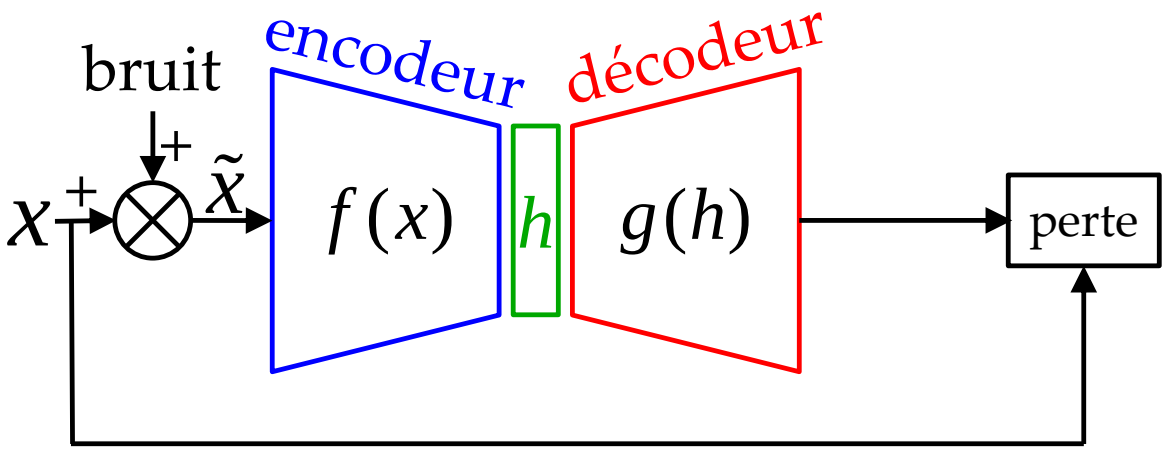
- Exemple : perte *L1*

$$\Omega(h) = \lambda \sum_i |h_i|$$

- Cousin du *sparse coding*
- Constitue un *prior* sur les modèles de *h* pour la génération des données *x*
- Semble aider pour des tâches connexes de classification

AE denoising

- Ajouter du **bruit aléatoire** à l'entrée x



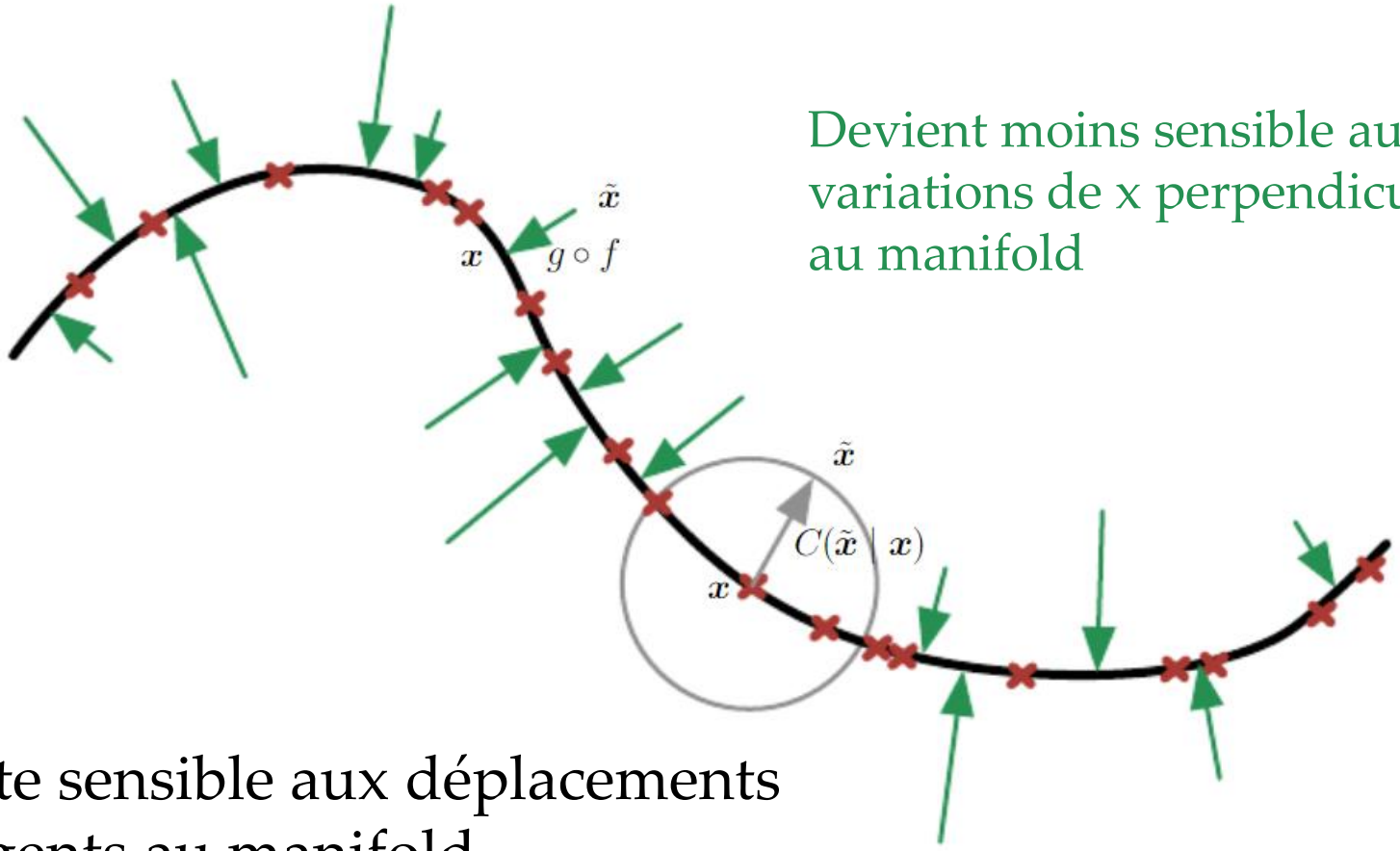
- Cherche quand même à reconstruire x

$$L\left(x, g\left(f\left(\tilde{x}\right)\right)\right)$$

- Fonctionne avec AE overcomplete/réseaux très puissants

AE denoising

- Apprend à déplacer des entrées corrompues $\tilde{x}$ vers le manifold



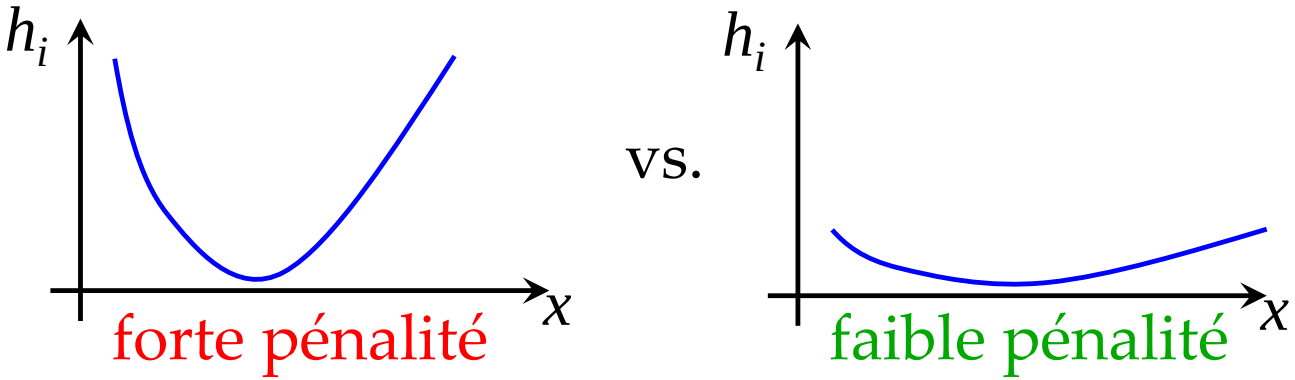
Devient moins sensible aux variations de x perpendiculaires au manifold

Reste sensible aux déplacements tangents au manifold

AE contractive

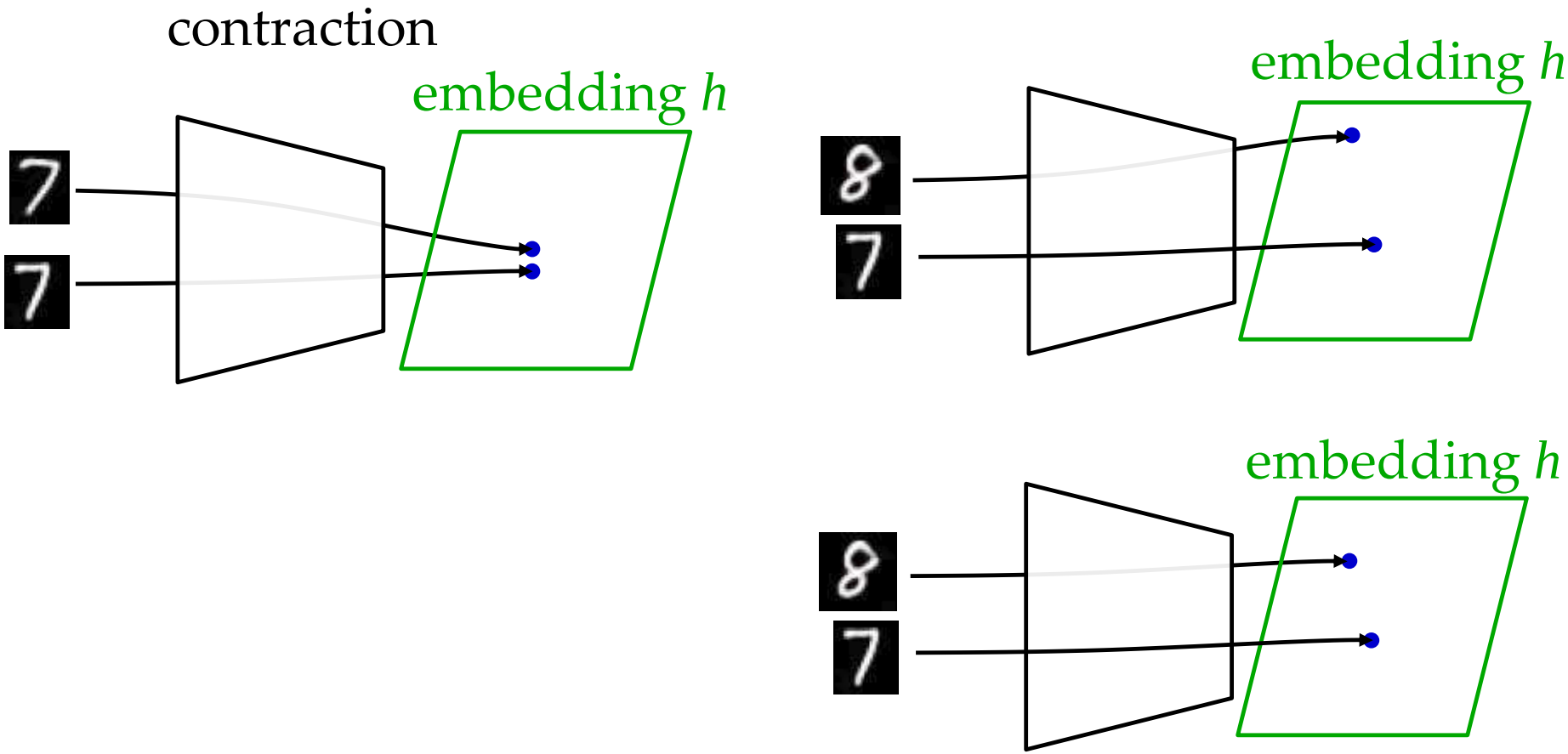
- Ajout d'une pénalité sur les gradients de l'encodeur f sur les données x

$$J(\theta) = L\left(x, g(f(x))\right) + \lambda \sum_i \left\| \nabla_x h_i \right\|^2$$



- Désensibiliser l'encodeur à certaines directions (perpendiculaires au manifold)

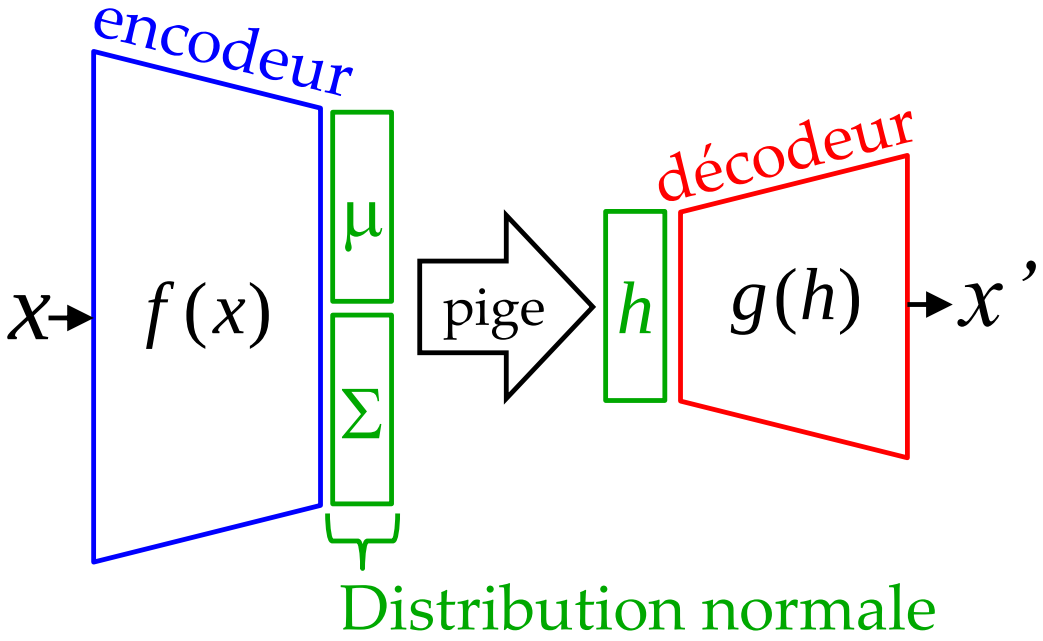
AE contractive



- Pour très petits bruits, denoising AE et contractive coïncident *(pensez différentiation numérique vs analytique)*

Variational AE (VAE)

- Processus stochastique



- Perte : Reconstruction + KL divergence *(pour forcer la distribution d'être proche d'une normale)*
- L'encodeur en charge d'estimer les paramètres de génération
- Entraînement plus complexe (reparameterization trick) car gradient ne passe pas l'opération de sampling

Autoencodeur : application

- Réduction de dimensionnalité pour classification (généralisation)
- Permet de combiner non-supervisé avec supervisé : semi-supervisé

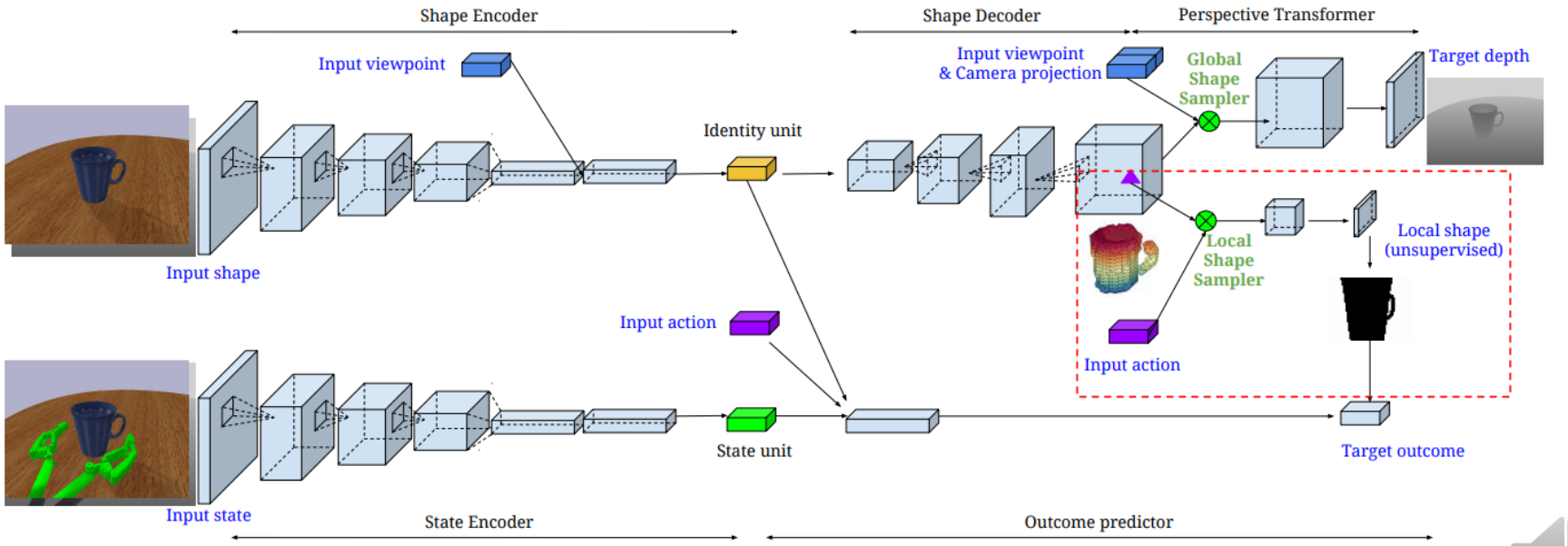
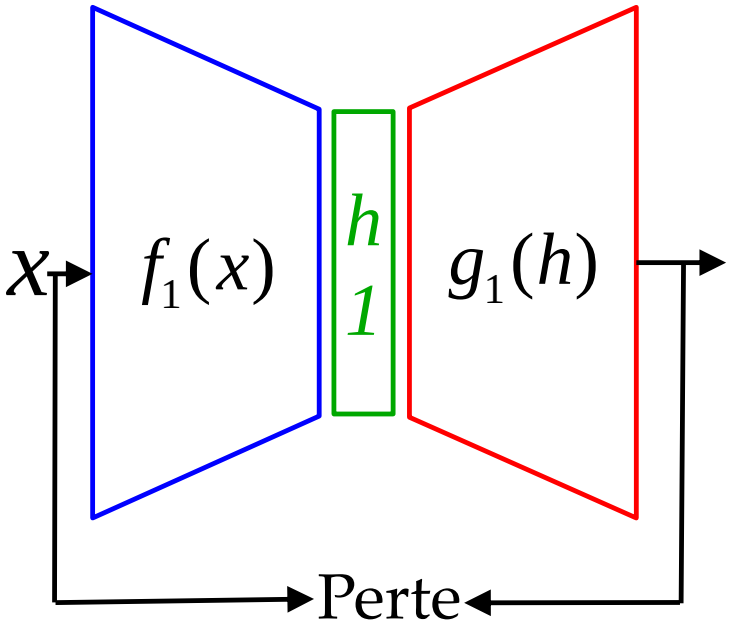


Fig. 2. Illustration of deep geometry-aware grasping network.

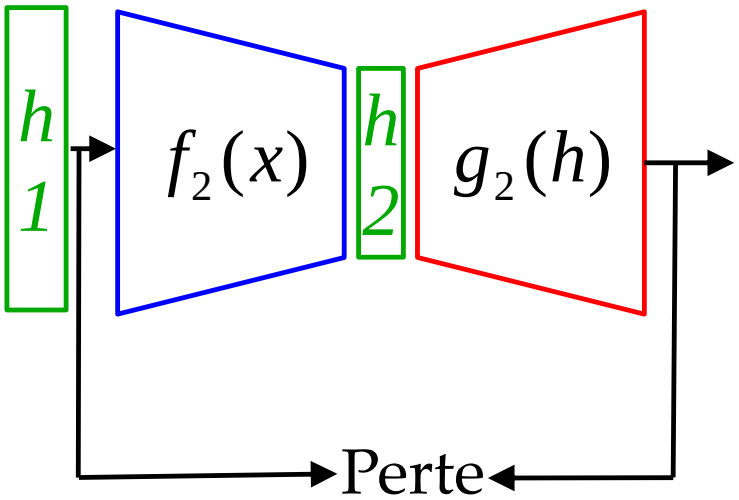
Deep AE : entraînement par couche

- Si difficulté d'entraîner un AE profond, possibilité d'y aller de manière vorace, couche par couche



Deep AE : entraînement par couche

- Si difficulté d'entraîner un AE profond, possibilité d'y aller de manière vorace, couche par couche



Conclusion

- Manière d'exploiter des données non-étiquetées (nombreuses, peu coûteuses)
- Basé sur encodeur-décodeur, embedding *h*
- Recherche du *manifold* des données
- Perte de base : reconstruction
- Quatre approches principales
  - Sparse
  - Denoising
  - Contractive
  - Variational (VAE) ← (Note : servira dans les Generative Adverserial Network)