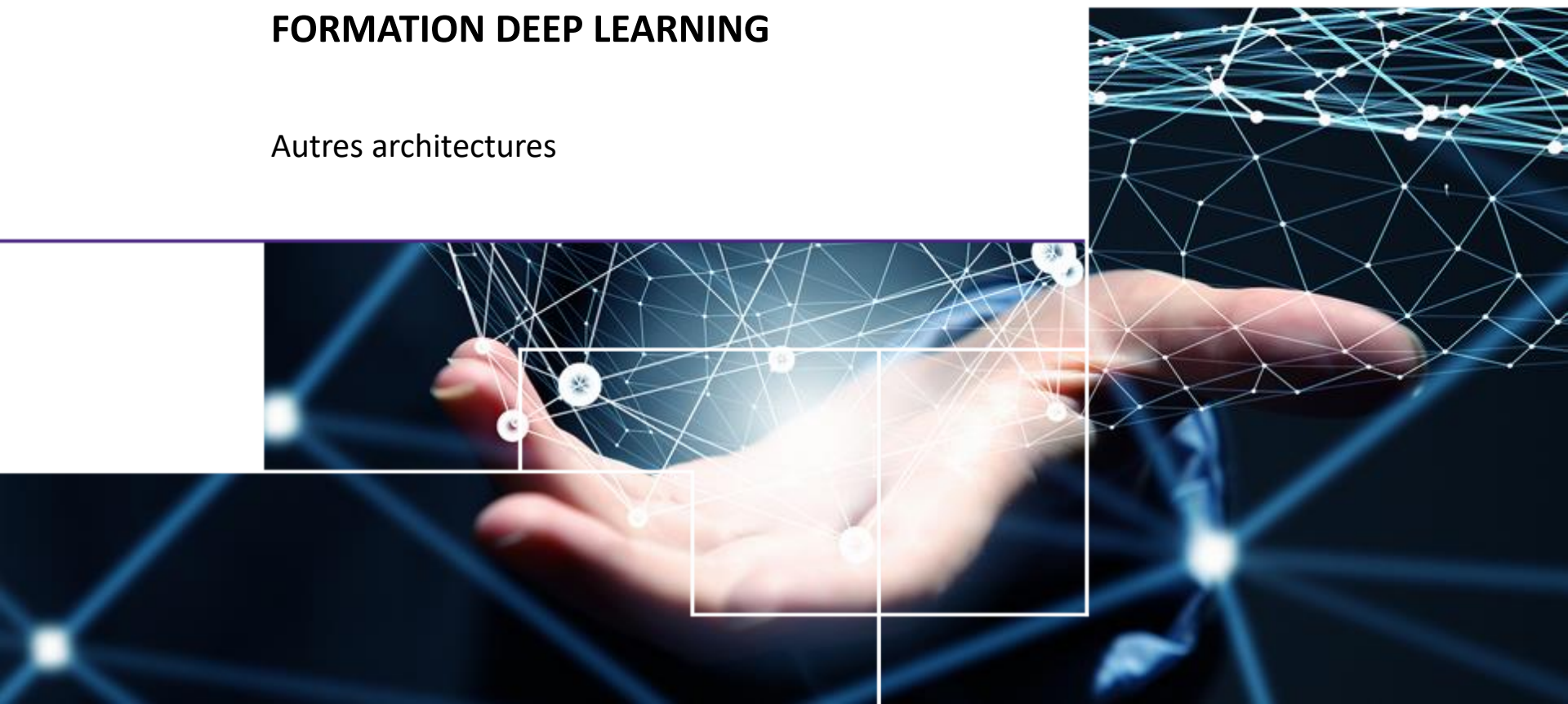
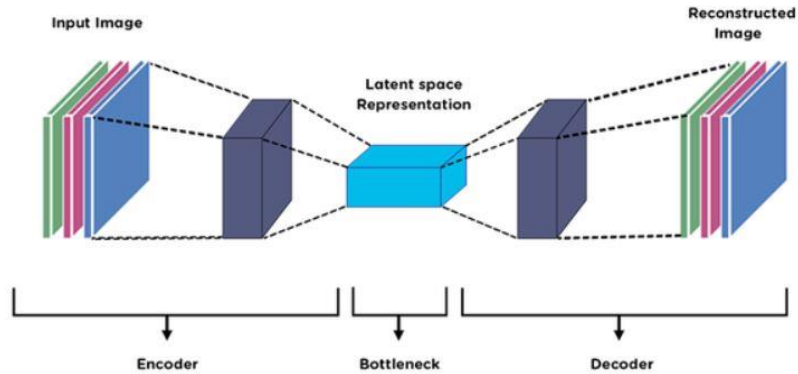


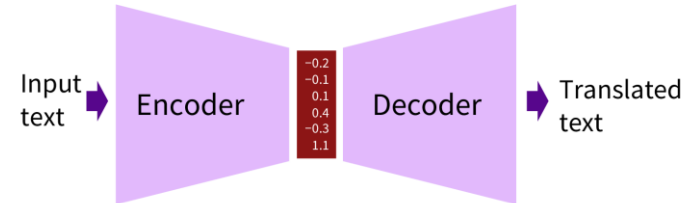
FORMATION DEEP LEARNING

Autres architectures





Neural encoder-decoder architectures



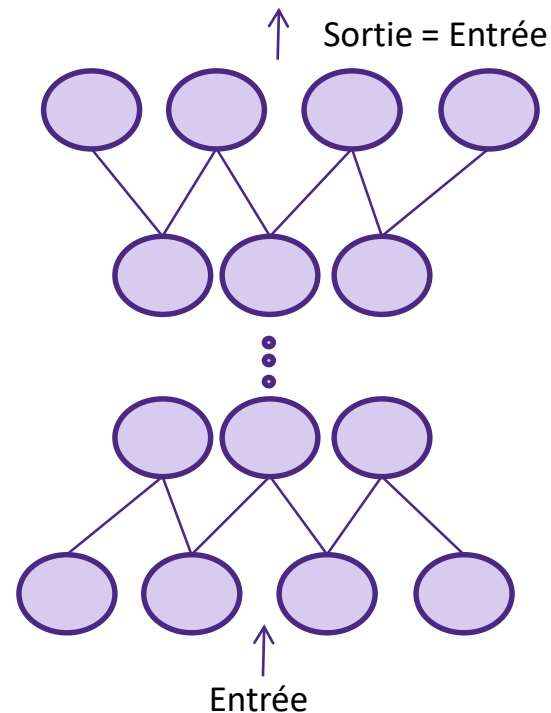
15



AUTOENCODER

Principe : Apprendre X à partir de X.

Mais quel est donc l'intérêt?



Principe : Apprendre X à partir de X .

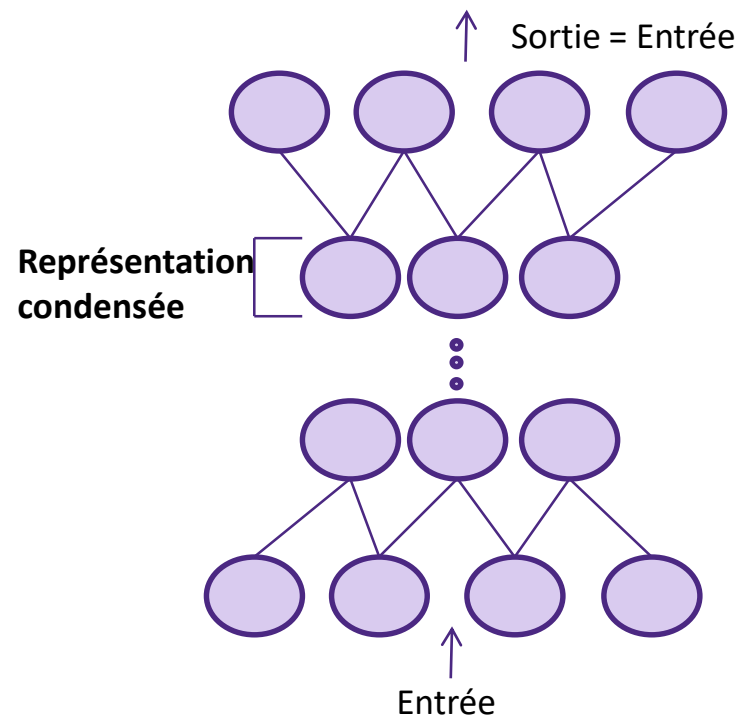
- **Apprend une représentation condensée de X .**
- **N'apprend que les caractéristiques les plus pertinentes de X .**

Applications :

- **Permet de trouver l'exemple appris le plus proche de l'entrée.**

Exemple : trouver des textes similaires à un texte en entrée.

- **La représentation condensée peut être utilisée comme entrée pour d'autres réseaux**



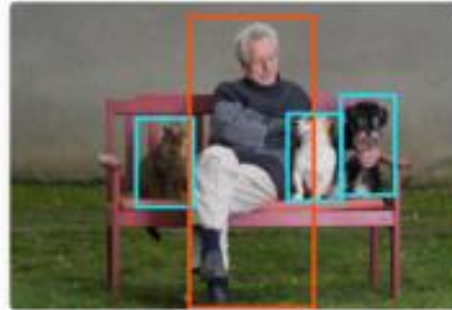


PIXEL SEGMENTATION

PERSON, CAT, DOG



(A) Classification



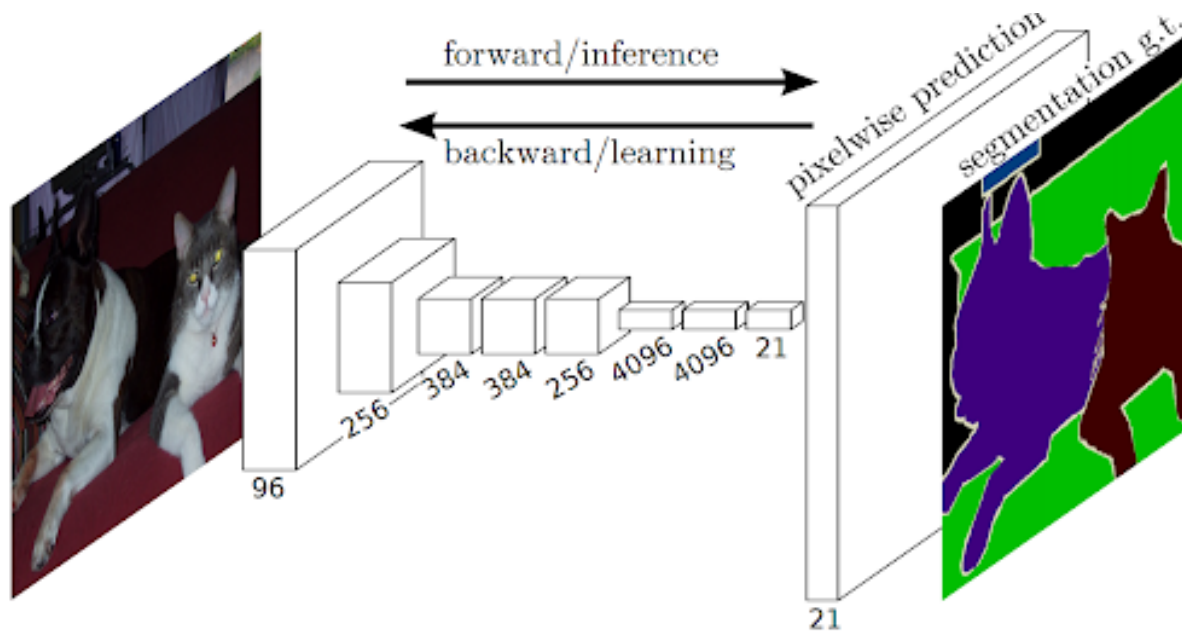
(B) Detection



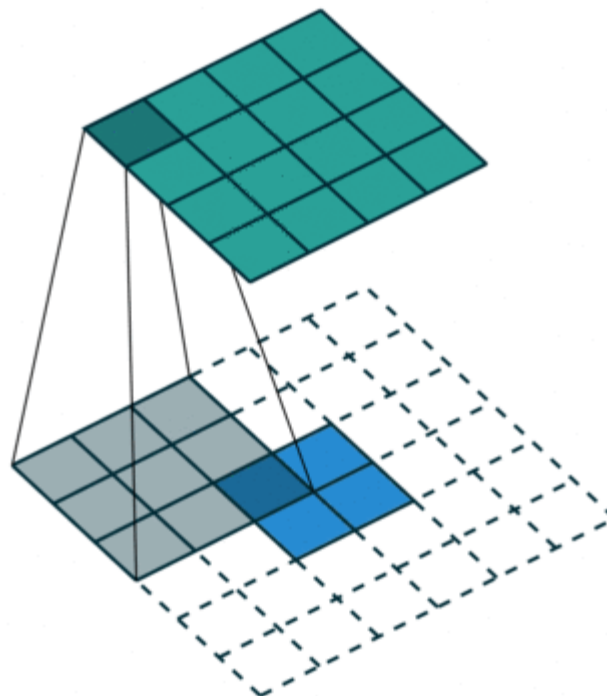
(C) Segmentation

THE FULLY-CONVOLUTIONAL NETWORK

UNE APPROCHE DIFFÉRENTE



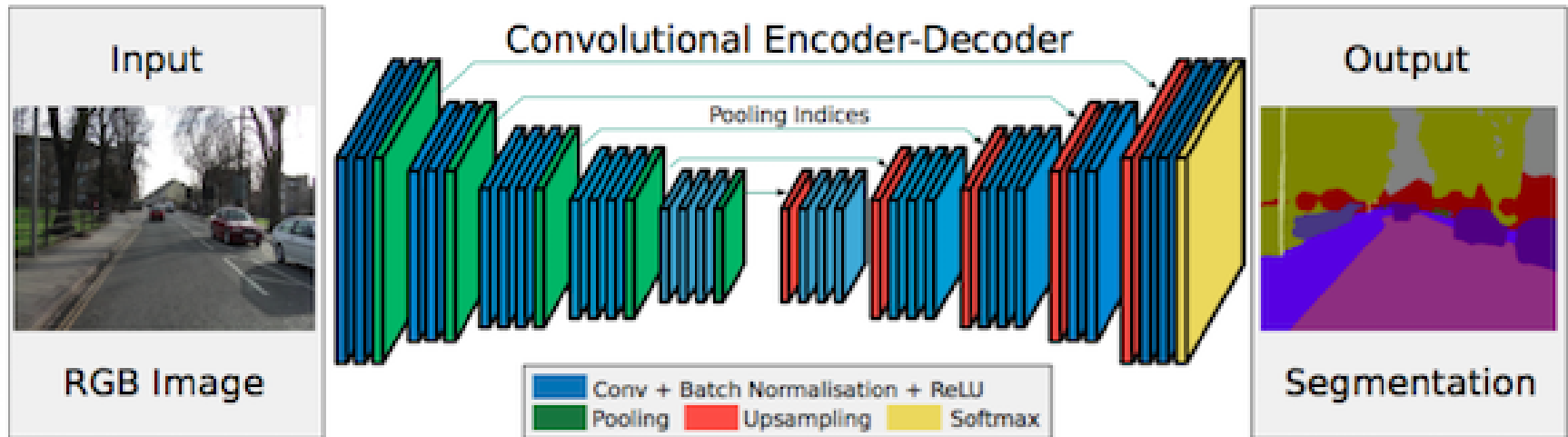
UPSAMPLING



Upsampling Via Deconvolution (Blue: Input, Green: Output)

SEGMENTATION D'IMAGES

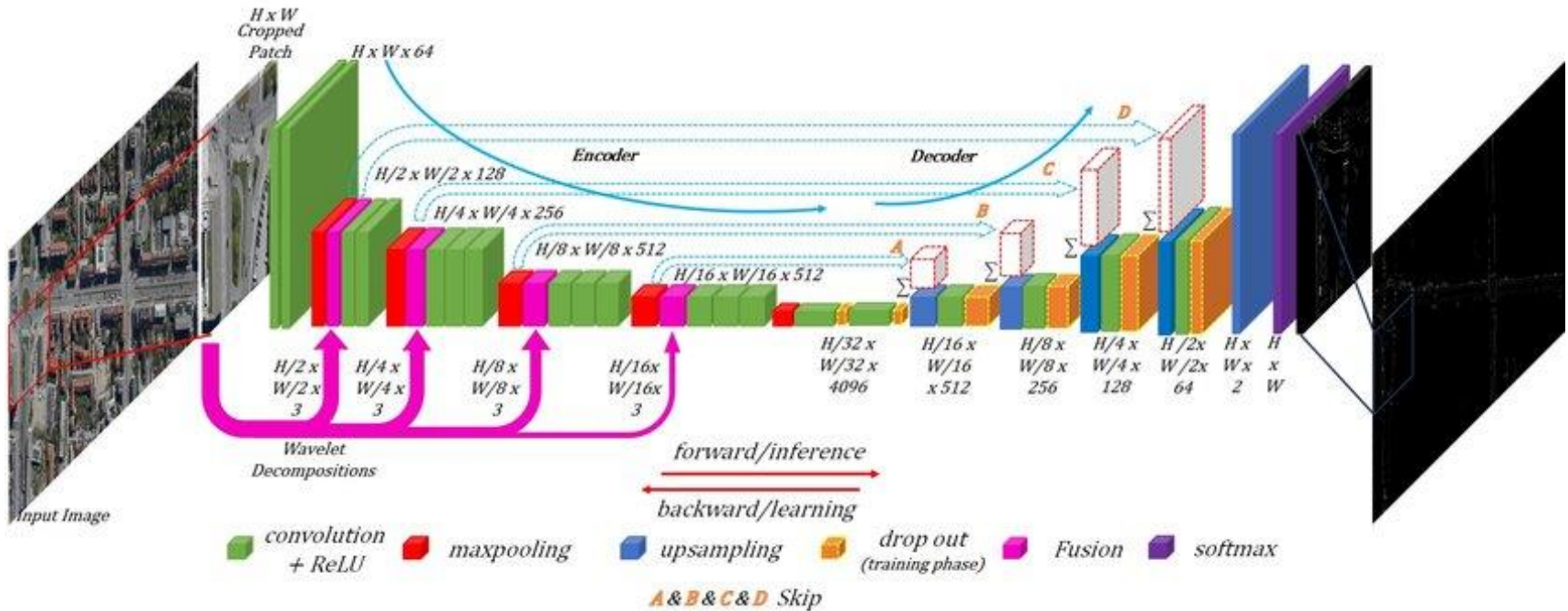
L'AJOUT DE SKIP CONENCTION



- On réduit la taille de l'information pour la reconstruire à l'aide d'Upsampling
- On réalise une prédiction par pixel

SEGMENTATION D'IMAGES

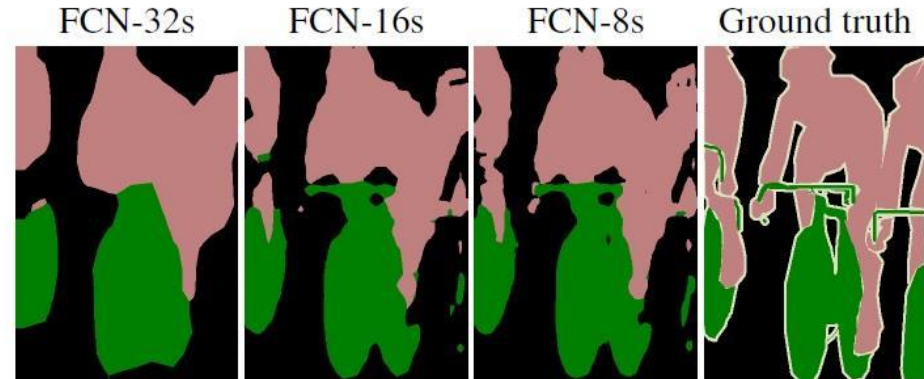
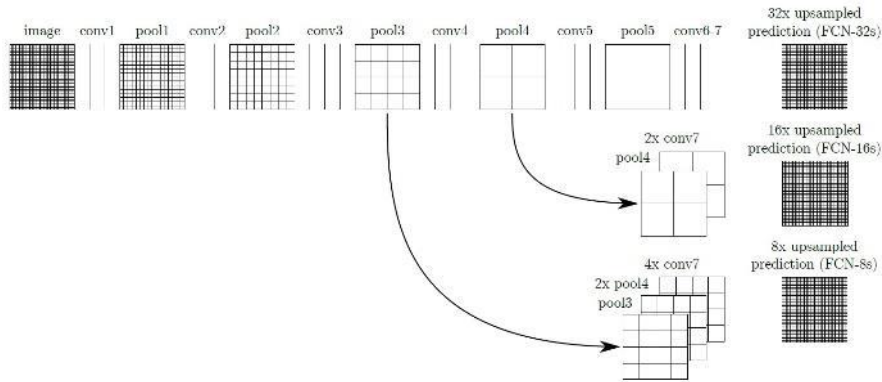
VUE DÉTAILLÉE



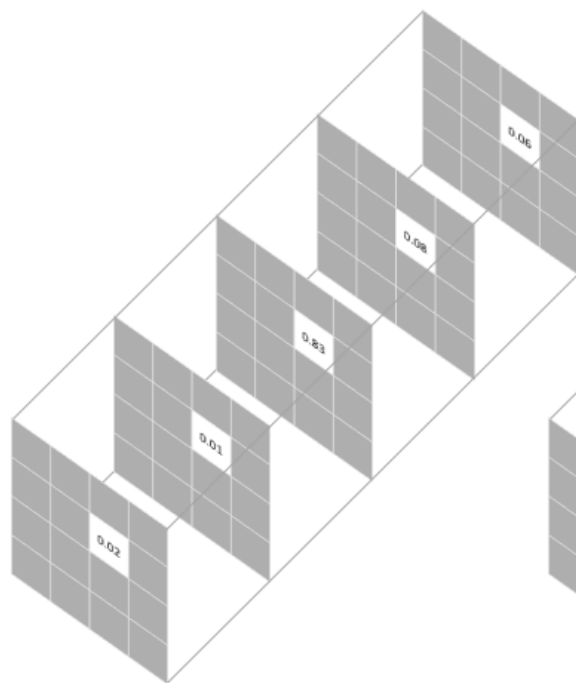
- On observe une décomposition en 2 parties : encoder, decoder
- On utilise des « skip connections »

SEGMENTATION D'IMAGES

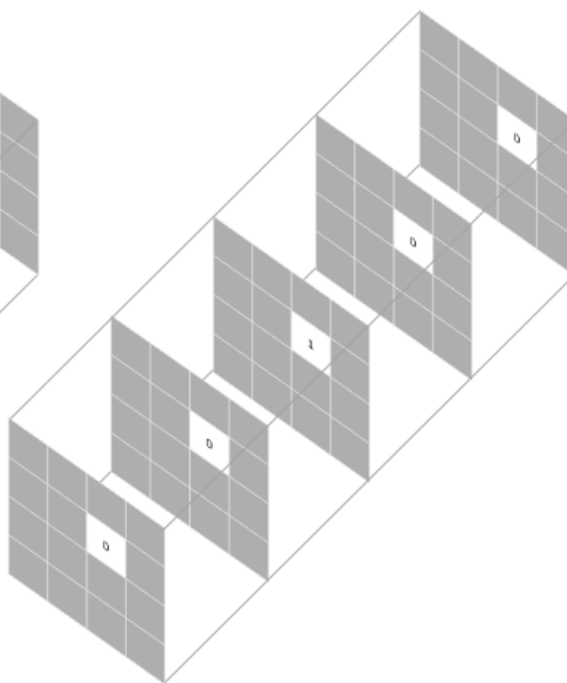
SKIP CONNECTIONS



LE CALCUL DE LA FONCTION DE LOSS



Prediction for a selected pixel



Target for the corresponding pixel

Pixel-wise loss is calculated as the log loss, summed over all possible classes

$$-\sum_{classes} y_{true} \log(y_{pred})$$

This scoring is repeated over all **pixels** and averaged

- **Fully Convolutional Networks**
- **U-Net**: Convolutional Networks for Biomedical Image Segmentation
- **The One Hundred Layers Tiramisu**: Fully Convolutional DenseNets for Semantic Segmentation
- **DeepLab**: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs
- **FastFCN**: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation
- ...

Pour en savoir plus: [source](#)

**Il existe plusieurs problèmes pour du traitement d'image :
classification, détection, segmentation.**

**L'usage du deep learning a révolutionné les performances pour ces
3 problématiques.**

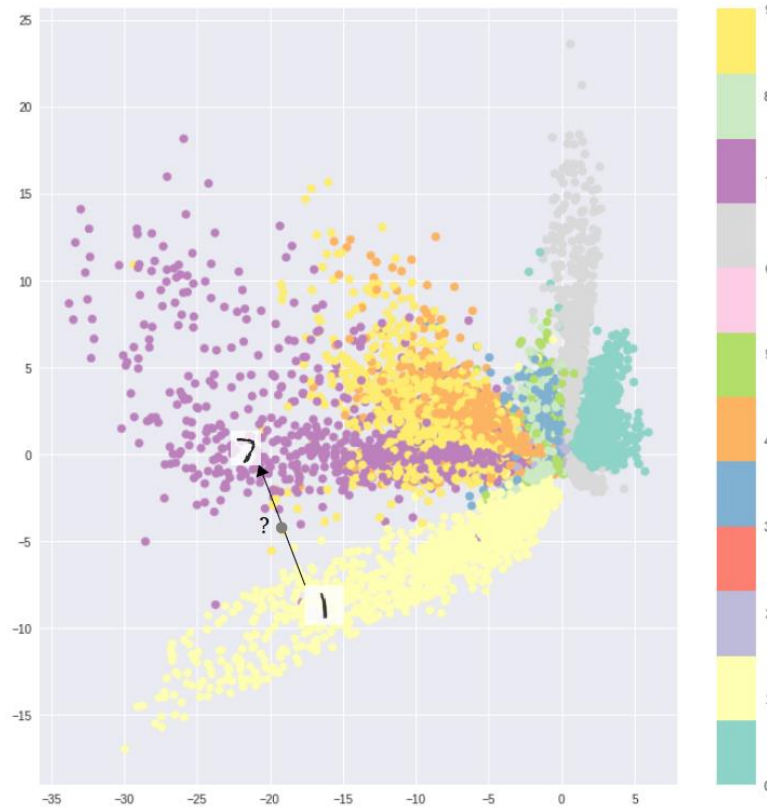
**Il existe plusieurs architectures différentes pour chaque problème.
Certains ont une meilleur performance de prédiction et d'autres un
temps d'inférence plus rapide.**

Là encore, tout est question de compromis.



GÉNÉRATION DE DONNÉES

L'ESPACE LATENT D'UN AUTOENCODER

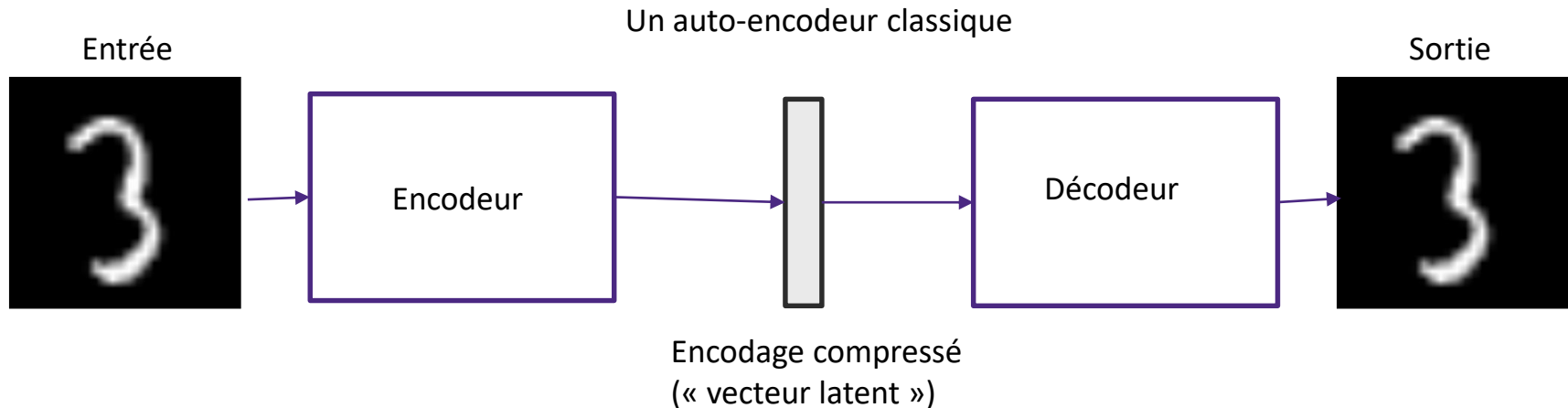


Peut-on, en modifiant l'espace latent de manière aléatoire générer de nouvelles images ?

Visualisation de l'espace latent d'un autoencoder entraîné sur MNIST

VAE : variationnel auto-encodeur

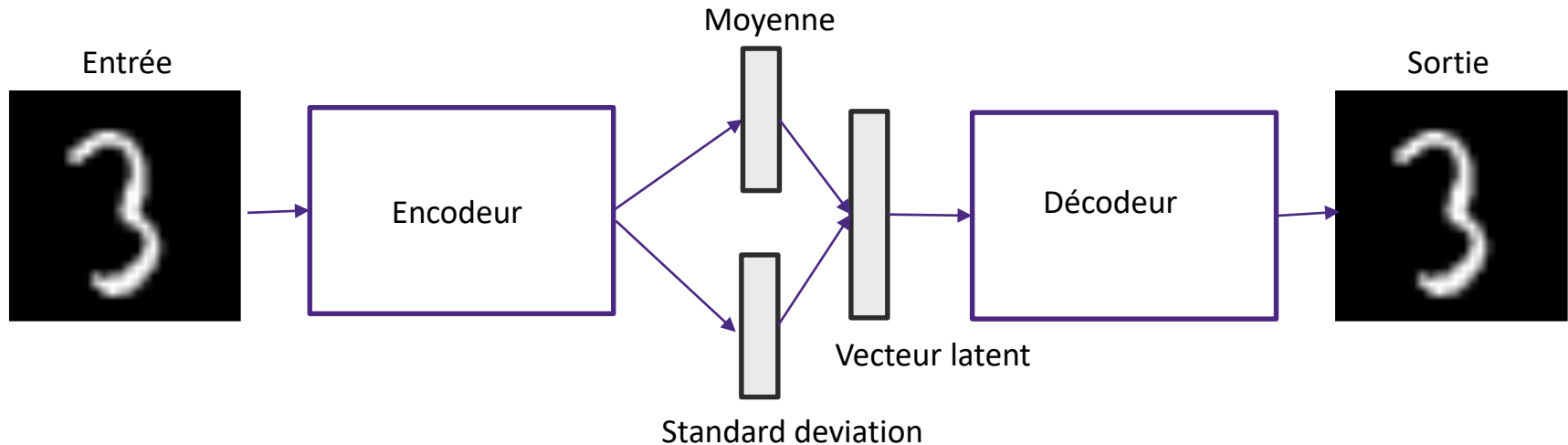
- Principe : utiliser l'encodage compressé appris par un auto-encodeur pour générer de nouvelles images



- Problème : si je génère un encodage aléatoire, comment garantir que celui-ci soit suffisamment proche d'un vrai encodage, pour générer une image ayant du sens ?

VAE : variationnal auto-encodeur

- **Solution : imposer que le vecteur latent suive une distribution gaussienne, ce qui permet de générer facilement des vecteurs aléatoires « crédibles ».**



Un VAE

- **Dernier problème : comment faire pour que les nouvelles couches apprennent bien une moyenne et une standard deviation ?**

- **Dernier problème : comment faire pour que les nouvelles couches apprennent bien une moyenne et une standard deviation ?**
- **Solution : inclure ces contraintes lors de l'apprentissage du réseau.**

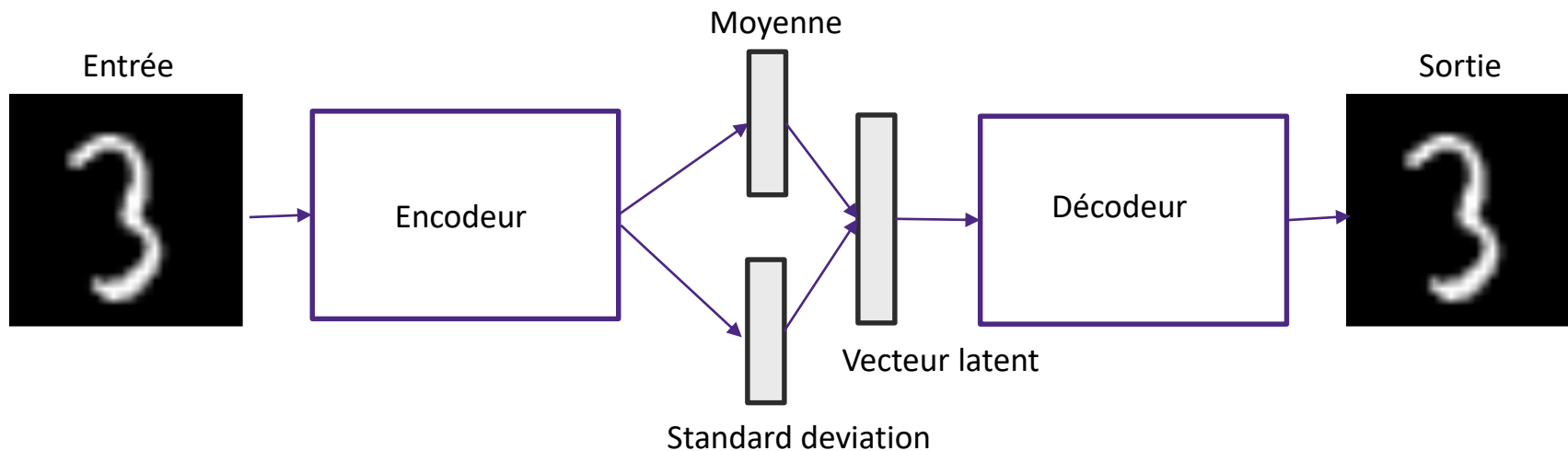
Erreur = erreur_reconstruction + erreur_latente

Et le réseau est entraîné à minimiser cette erreur comme un réseau classique.

- **Erreur_reconstruction mesure la différence entre l'entrée et la sortie reconstruite.**
- **Erreur_latente mesure si les valeurs de la couche latente suit bien une loi gaussienne avec pour paramètre les valeurs des couches mean et std.**

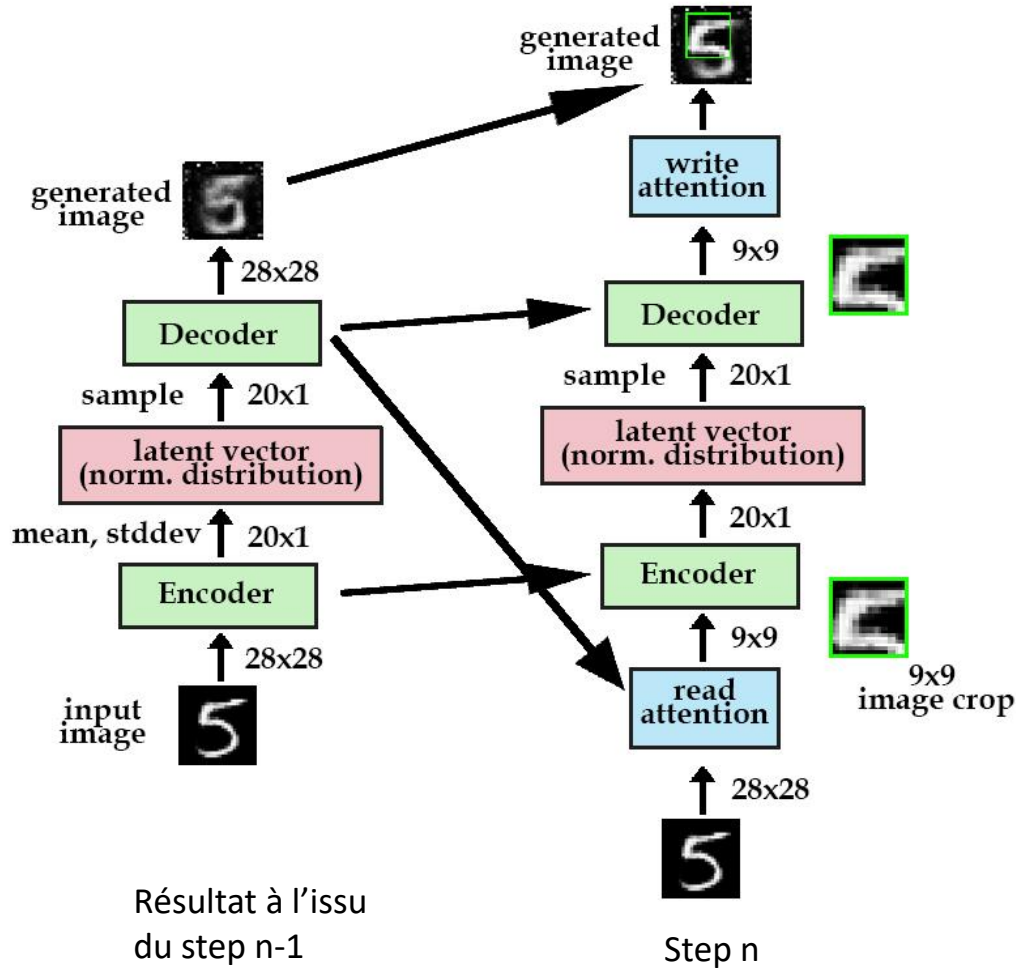
- Avantages : plus rapides que les gans.
- Très bon compromis entre temps d'entraînement et qualité de l'image générée.

Nous verrons ça sur une application !



Draw : Recurrent neural network for image generation

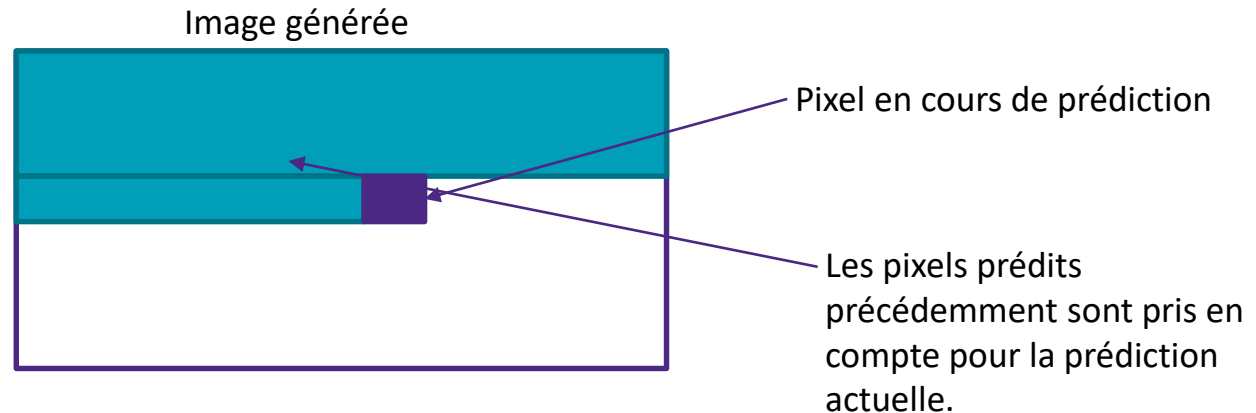
- Basé sur un VAE.
- Idée : inspiré des artistes, ils ne dessinent pas le dessin final d'un seul coup, mais améliore leur dessin au fur et à mesure et partie par partie.
- Utilisation de récurrence dans le réseau.
- Utilisation de mécanisme attentionnel :
 - A chaque étape de la récurrence le réseau se concentre sur une région partielle différente de l'image
- Meilleur qu'un VAE, mais plus long



Fonctionnement du
Modèle DRAW

Pixel RNN : Modèle autorégressif utilisé pour la génération ou reconstruction d'image.

- **Autorégressif car prédit les pixels au fur et à mesure, et prend en compte les derniers pixels prédits pour prédire le suivant.**
- **Utilisation de neurones récurrents de type LSTM.**
- **Apprend la probabilité de la couleur d'un pixel en fonction des précédents.**



- Utilisé pour la génération ou reconstruction d'entrée.



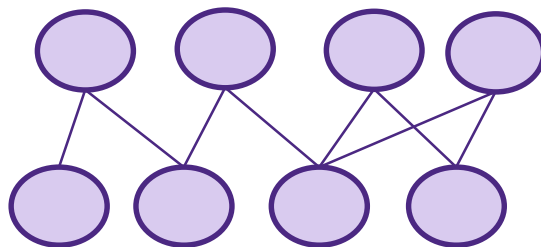
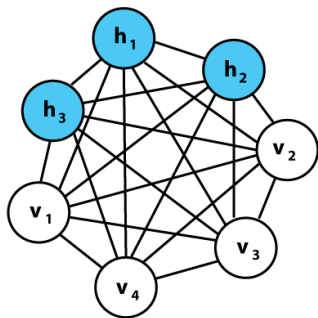
Figure 1. Image completions sampled from a PixelRNN.



AUTRES MODÈLES

Machine de boltzmann : réseau constitué d'une entrée et d'une couche cachée

- Restreinte (RBM) si toutes les connections sont entre l'entrée et la couche cachée.
- En pratique seule les RBM sont entraînables.
- « Auto-encodeur » replié qui apprend à estimer la distribution de probabilité des features en entrée.

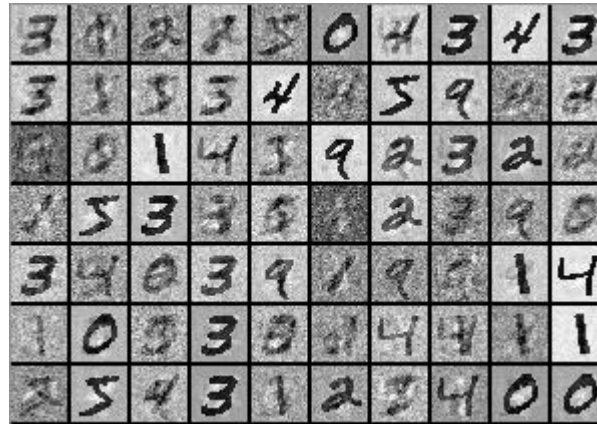


Couche cachée = features en entrées
souvent présentes ensembles.

Entrée

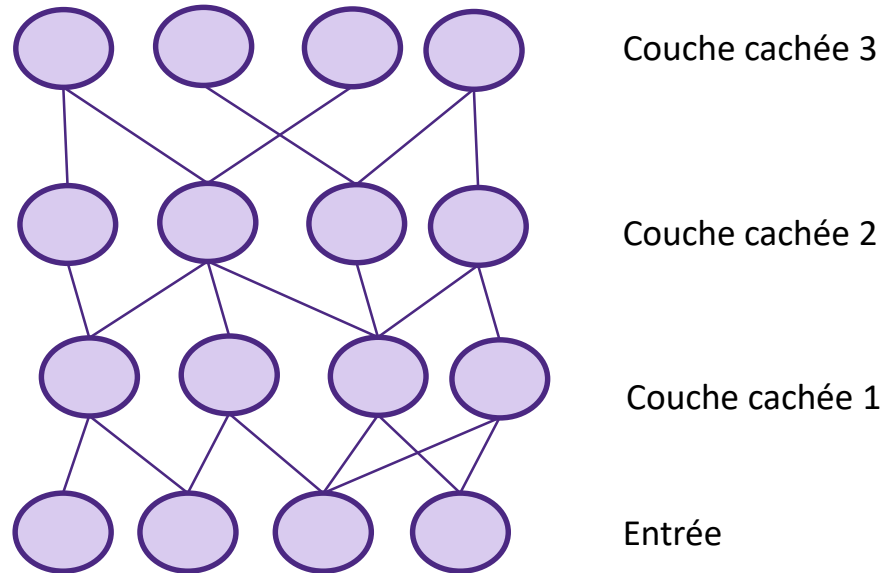
Machine de Boltzmann restreinte

- Les neurones cachés représentent les dépendances entre les features de l'entrée
- **Applicables à la génération de données**
 - Si l'on connaît la distribution de probabilité on peut générer de nouvelles images



- **Application à la reconstruction de donnée**
 - En passant une entrée partielle, reconstruit l'entrée.

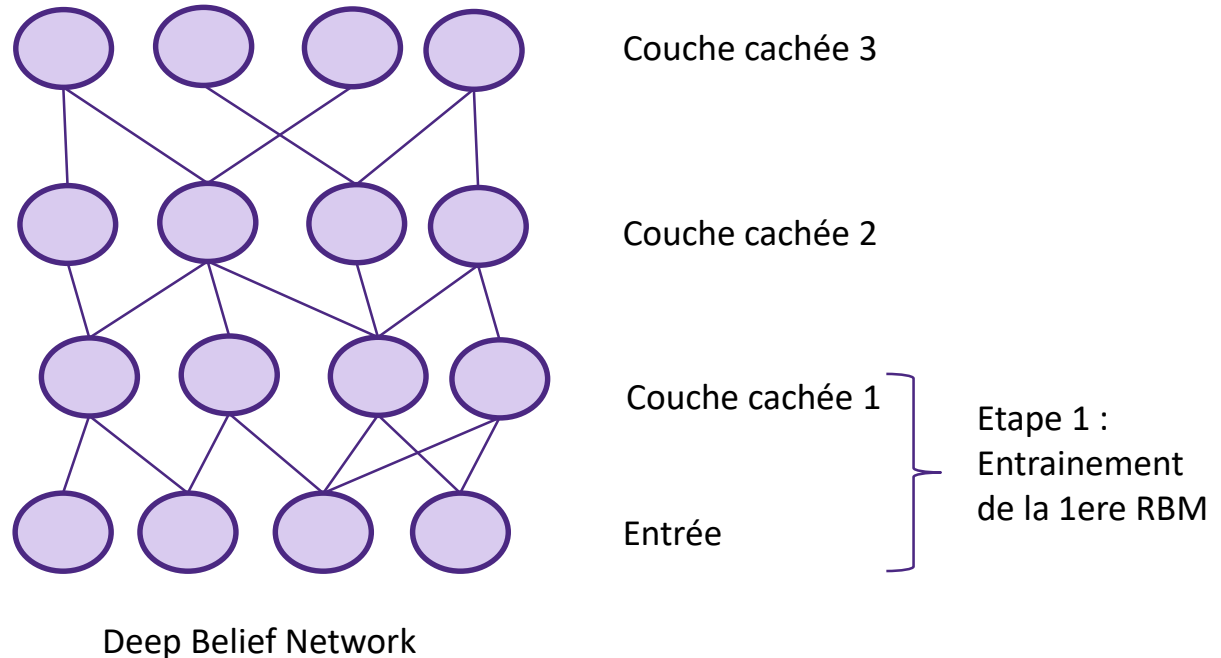
Deep belief network : réseau construit en « empilant » des RBMs



Deep Belief Network

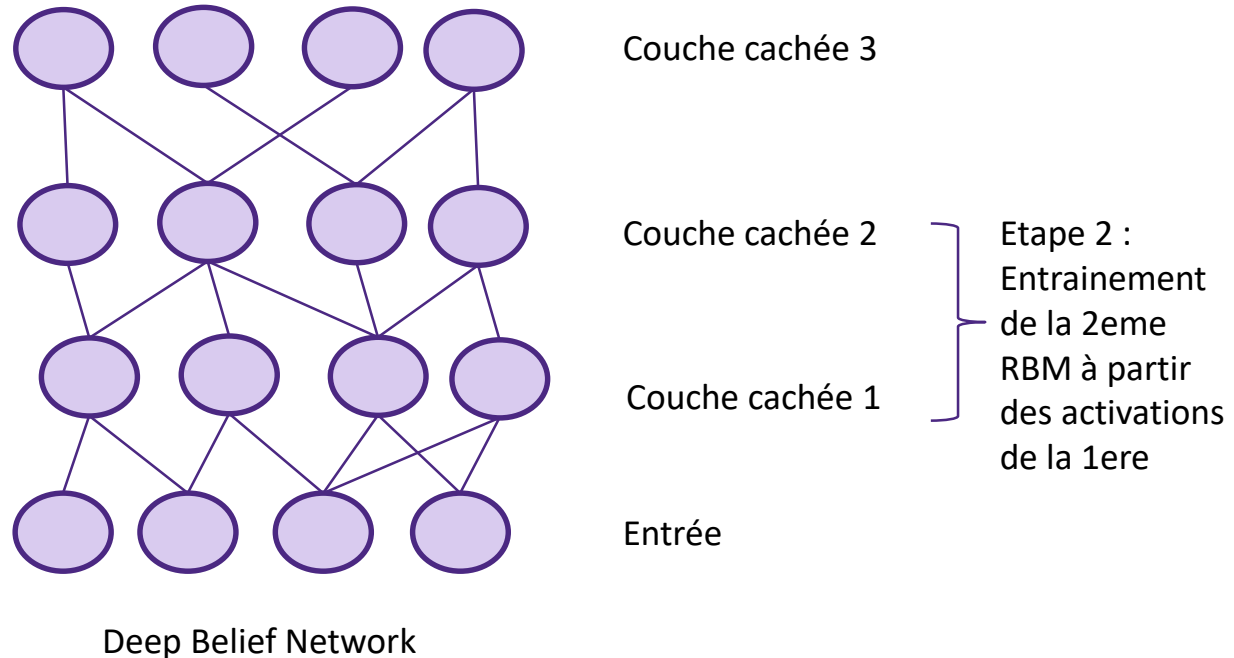
Deep belief network : réseau construit en « empilant » des RBMs

Entraînement « greedy » des RBMs une par une.



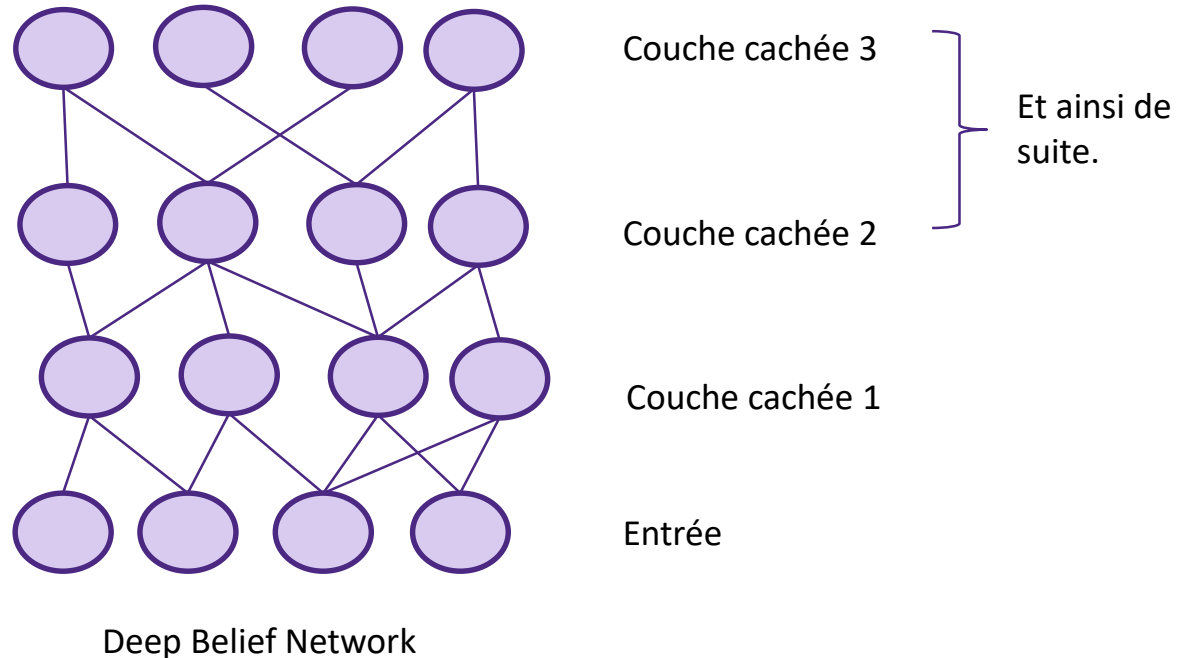
Deep belief network : réseau construit en « empilant » des RBMs

Entraînement « greedy » des RBMs une par une.



Deep belief network : réseau construit en « empilant » des RBMs

Entraînement « greedy » des RBMs une par une.

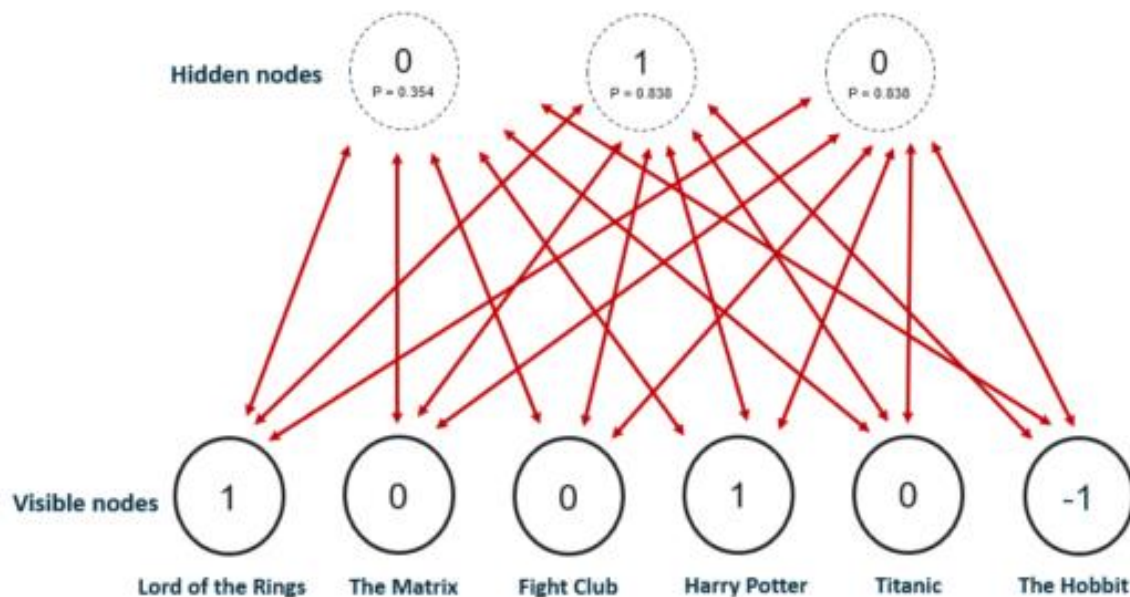


Application : reconstruction d'une entrée bruitée ou partielle

Application : extracteur de feature non supervisé

- **Pour les images, le transfert learning donne de meilleurs résultats.**
- **Applicable dans des domaines, ou sur des données, où le transfert learning n'est pas applicable.**
- **Ou si les modèles entraînés n'ont pas appris de features adaptées à la tâche.**

MOTEUR DE RECOMMANDATIONS



Permet, une fois entraîné,

À partir des films aimés ou non par l'utilisateur de générer une liste de features latentes qui décrivent les « goûts » pour un utilisateur.