

Apprentissage supervisé

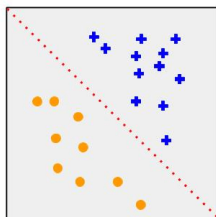
Partie 3

Analyse de données et Classification 2
ENSEEIH - 3ème année Sciences du Numérique

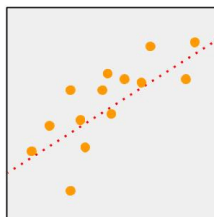
Contact :

Sandrine.Mouysset@irit.fr
sandrine.mouysset@toulouse-inp.fr

2 principaux types d'apprentissage supervisé :



Classification



Regression

Approches par modèles supervisés :

- Arbres de Décision
- Apprentissage d'ensemble :
Forêts aléatoires
- Réseaux de neurones
- Ouverture sur le deep learning

Méthodes d'évaluation :

- Validation croisée
- Matrice de confusion
- Precision, Rappel, F-mesure
- Courbe ROC

Validation croisée : méthode d'estimation de fiabilité d'un modèle fondé sur une technique d'échantillonnage.

- *Testset validation ou holdout method* : on divise l'échantillon de taille n en deux sous-échantillons, le premier dit d'*apprentissage* (communément supérieur à 60 % de l'échantillon) et le second dit de *test*. Le modèle est bâti sur l'échantillon d'apprentissage et validé sur l'échantillon de test. L'erreur est estimée en calculant un test, une mesure ou un score de performance du modèle sur l'échantillon de test;
- *k-fold cross-validation* :
- *Leave-one-out cross-validation (LOOCV)* :

Validation croisée : méthode d'estimation de fiabilité d'un modèle fondé sur une technique d'échantillonnage.

- *Testset validation ou holdout method :*
- *k-fold cross-validation :* on divise l'échantillon original en k échantillons, puis on sélectionne un des k échantillons comme ensemble de validation et les $k - 1$ autres échantillons constitueront l'ensemble d'apprentissage. On calcule comme dans la première méthode le score de performance, puis on répète l'opération en sélectionnant un autre échantillon de validation parmi les $k - 1$ échantillons qui n'ont pas encore été utilisés pour la validation du modèle. L'opération se répète ainsi k fois pour qu'en fin de compte chaque sous-échantillon ait été utilisé exactement une fois comme ensemble de validation. La moyenne des k erreurs quadratiques moyennes est enfin calculée pour estimer l'erreur de prédiction ;
- *Leave-one-out cross-validation (LOOCV) :* cas particulier de la deuxième méthode où $k = n$, c'est-à-dire que l'on apprend sur $n - 1$ observations puis on valide le modèle sur la i ème observation et l'on répète cette opération n fois.

Matrice de confusion : consiste à compter le nombre de fois où des observations de la classe A ont été rangées dans la classe B .

- Chaque ligne de la matrice de confusion représente la *classe réelle* tandis que chaque colonne représente une *classe prédite*.
- Les éléments hors diagonaux représenteront les erreurs de classification.
Exemple : si on souhaite connaître le nombre de fois où le classifieur a pris des 5 pour des 3, on regardera l'élément hors diagonal (5,3) dans la matrice de confusion.
- Les éléments diagonaux représenteront le nombre d'éléments bien classés.
Classification parfaite = matrice de confusion diagonale

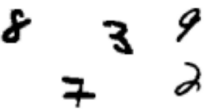



Précision, Rappel : à partir de la matrice de confusion, on peut évaluer d'autres métriques basées sur les notions de vrais positifs (TP), faux positifs (FP) et respectivement vrais et faux négatifs (TN, FN)

- **Précision :** évalue l'exactitude des prédictions positives

$$Precision = \frac{TP}{TP + FP}$$

- **Rappel :** évalue le taux d'observations positives ayant été correctement détectées par le classifieur

$$Rappel = \frac{TP}{TP + FN}$$

		Predicted		
		Negative	Positive	
Actual	Negative			TN FP
	Positive			FN TP

Precision (e.g., 3 out of 4)

Recall (e.g., 3 out of 5)

F-mesure : moyenne *harmonique* de la précision et du rappel donnant davantage de poids aux faibles valeurs.

$$F_1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Rappel}} = 2 \frac{Precision \times Rappel}{Precision + Rappel}$$

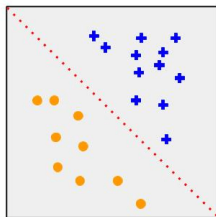
Interprétation : le classifieur n'obtiendra un bon score F_1 que si son rappel et sa précision sont élevés. Le score F_1 favorise donc les classifieurs ayant une précision et un rappel similaires (ce qui n'est pas forcément souhaitable suivant l'étude que vous menez).

Courbe ROC : courbe d'efficacité du récepteur (Receiver Operating Characteristic ou ROC) très semblable à la courbe précision/rappel mais elle croise les taux de vrais positifs (rappel) avec le taux de faux positifs (c-à-d le pourcentage d'observations négatives qui sont incorrectement classées comme positives ($FP=1-TP$)).

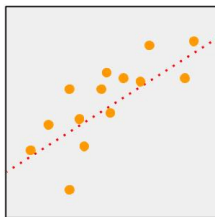
Différences entre courbes Précision/Rappel et ROC :

- préférer une courbe Précision/Rappel si la classe positive est rare ou si on attache plus importance aux faux positifs qu'aux faux négatifs;
- préférer la courbe ROC ou le score ROC AUC représentant l'aire sous la courbe ROC (égal à 1 si classifieur parfait) dans les autres cas.

2 principaux types d'apprentissage supervisé :



Classification



Regression

Approches par modèles supervisés :

- Arbres de Décision
- Apprentissage d'ensemble :
Forêts aléatoires
- Réseaux de neurones
- Ouverture sur le deep learning

Méthodes d'évaluation :

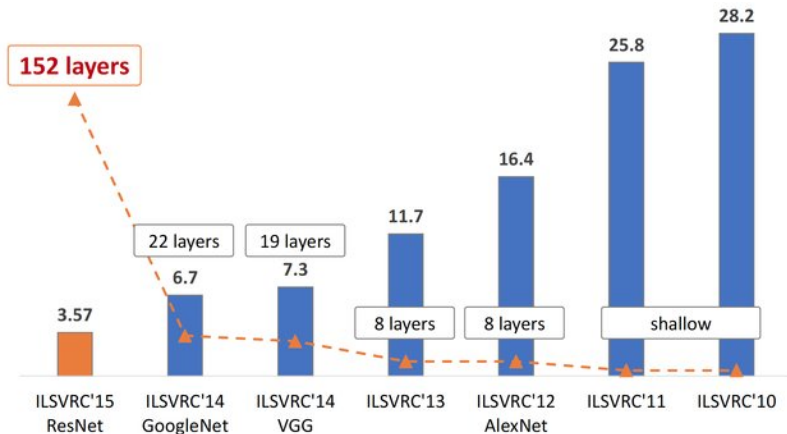
- Validation croisée
- Matrice de confusion
- Precision, Rappel, F-mesure
- Courbe ROC

Problème : classification d'images sur ImageNet

- 21841 classes : "boa constricteur", "papier toilette", "berger allemand", "noeud papillon", etc.
- > 14 millions d'images réparties sur les différentes classes



Classification d'image sur ImageNet : Évolution des performances



AlexNet (2012) : 8 couches

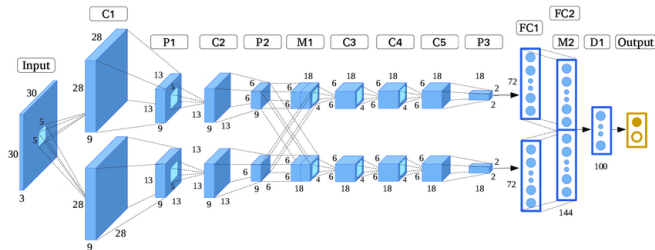


Image de [Krizhevsky et al. 2012] Imagenet classification with deep convolutional neural networks

ResNet (2015) : 152 couches

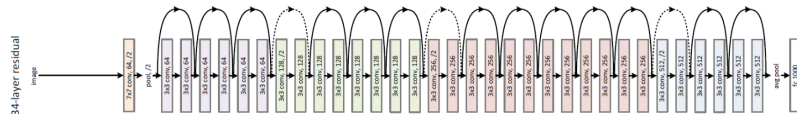


Image de [He et al. 2016] Deep Residual Learning for Image Recognition

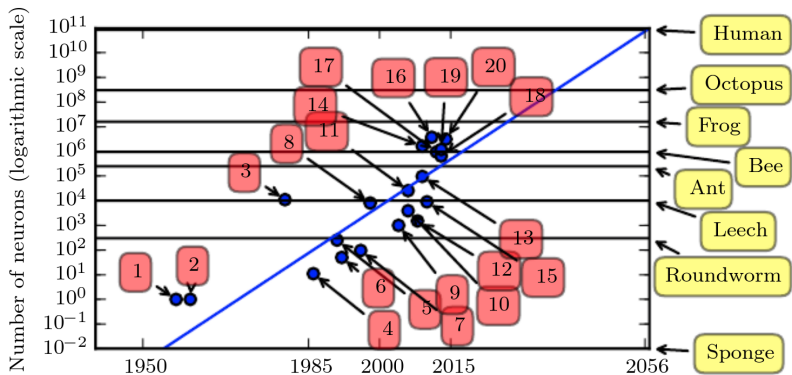


Image de [Goodfellow et al. 2015] Deep Learning

Les progrès récents de l'apprentissage profond se basent sur 2 avancées majeures :

- L'émergence de gigantesques bases de données d'apprentissage.
- L'apparition des cartes GPU, qui permettent de paralléliser efficacement le calcul matriciel au coeur des réseaux de neurones, à la fois pour la prédiction et pour l'entraînement.

La théorie était présente depuis des années...





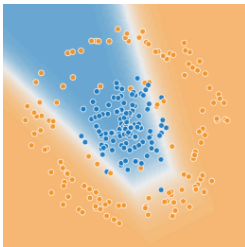
Avec l'augmentation de la profondeur des réseaux de neurones, un certain nombre de problèmes se posent (*vanishing* et *exploding gradients*; sur-apprentissage ou sous-apprentissage, etc.) qui nécessitent de mettre en application quelques bonnes pratiques.

- **Risque empirique** : erreur de prédiction moyenne sur l'ensemble d'apprentissage.
- **Risque espéré** (ou risque de généralisation) : erreur de prédiction moyenne sur l'ensemble des données... **Inconnu !**

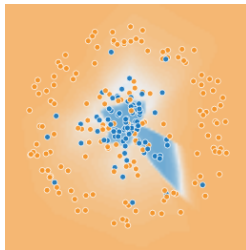
L'objectif principal d'un algorithme d'apprentissage est de minimiser le **risque espéré**, mais nous ne sommes capables d'évaluer que le **risque empirique**.

On parle de **sous-apprentissage** (*underfitting*) lorsque le modèle appris explique trop mal l'ensemble d'apprentissage.

On parle de **sur-apprentissage** (*overfitting*) lorsque le modèle appris explique à l'inverse trop bien l'ensemble d'apprentissage ; ce modèle se généralise alors mal à la population cible.



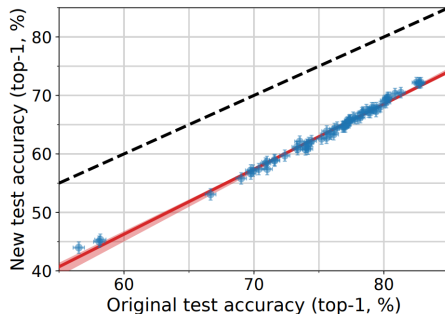
Sous-apprentissage



Sur-apprentissage

Solution : gérer 3 ensembles de données distincts

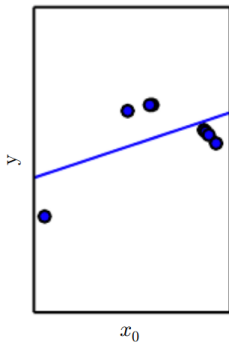
- l'ensemble **d'apprentissage**, sur lequel on va effectuer la descente de gradient.
- l'ensemble de **validation** : qui va nous fournir une estimation de l'erreur de généralisation, et nous permettre d'optimiser les hyperparamètres.
- l'ensemble de **test** : qui détermine la performance objective du réseau de neurones.



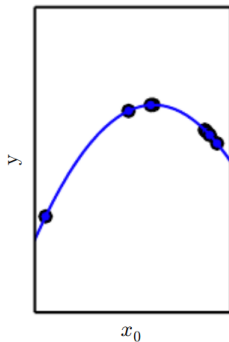
Les classifieurs sont en moyenne 10 à 15% plus performants sur l'ensemble de test original d'ImageNet que sur un nouvel ensemble généré par la même procédure.

[Recht et al. 2019] Do ImageNet Classifiers Generalize to ImageNet?

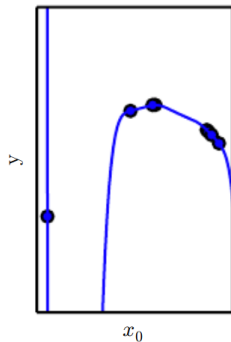
Underfitting



Appropriate capacity



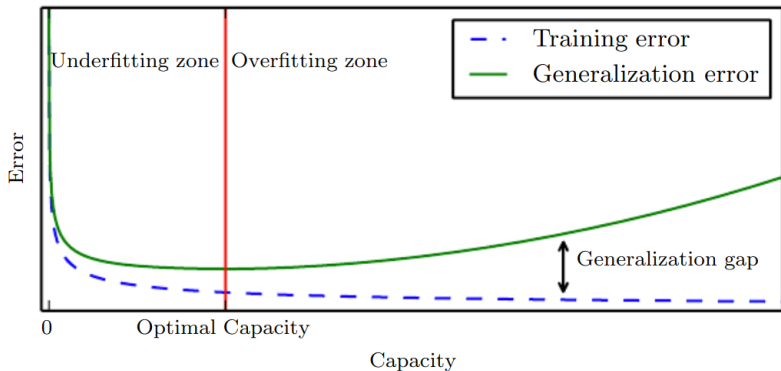
Overfitting



Il faut vérifier que l'erreur sur l'ensemble de validation est comparable à l'erreur sur l'ensemble d'apprentissage, pour détecter le sur-apprentissage.

Image de [Goodfellow et al. 2015] Deep Learning

Illustration graphique :



Un modèle de trop large capacité (profondeur, nombre de neurones) engendre du sur-apprentissage.

Idem pour un modèle entraîné trop longtemps !

Image de [Goodfellow et al. 2015] Deep Learning

L'arrêt anticipé est une stratégie utilisée pour éviter le surapprentissage, qui consiste à observer l'erreur commise sur l'ensemble de validation et mettre un terme à l'apprentissage quand cette erreur commence à remonter.

En pratique : l'erreur sur l'ensemble de validation est bruitée, il faut attendre un peu avant de s'arrêter pour de bon.

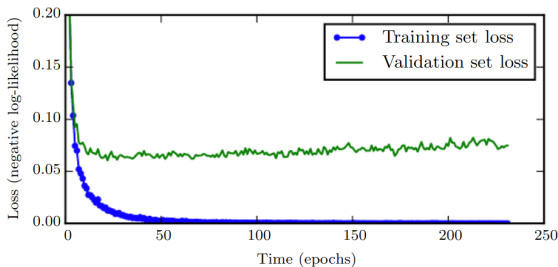


Image de [Goodfellow et al. 2015] Deep Learning

Eviter le surajustement grâce à la régularisation :

- Régularisation \mathcal{L}^2 ou **Ridge**: ajout d'un terme à la fonction de coût pour maintenir les coefficients du modèle aussi petits que possible

$$J(\theta) = \text{RisqueEmpirique}(\theta) + \alpha \frac{1}{2} \sum_{i=1}^m \theta_i^2$$

où α contrôle la qualité de régularisation souhaitée

- Régularisation \mathcal{L}^1 ou **Lasso** : ajout d'un terme à la fonction de coût

$$J(\theta) = \text{RisqueEmpirique}(\theta) + \alpha \sum_{i=1}^m |\theta_i|$$

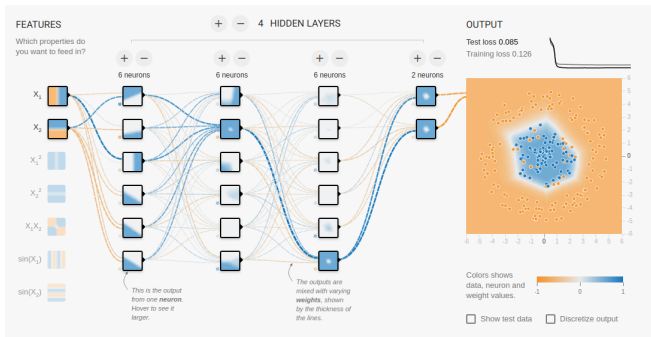
tend à éliminer complètement les poids des variables les moins importantes
(\Rightarrow produit un modèle creux)

- Régularisation **Elastic net** (filet élastique) : compromis entre Ridge et Lasso exprimé par le paramètre $r \in [0, 1]$:

$$J(\theta) = \text{RisqueEmpirique}(\theta) + r\alpha \frac{1}{2} \sum_{i=1}^m |\theta_i| + \frac{1-r}{2} \alpha \sum_{i=1}^m \theta_i^2$$

<https://playground.tensorflow.org/>

Une des causes de sur-apprentissage est l'émergence, pendant l'entraînement, de chemins préférentiels dans le réseau de neurones.



Pour éviter ce phénomène, on peut aléatoirement "éteindre" (déconnecter) des neurones du réseau pendant l'étape de prédiction. Cela permet de favoriser la redondance et l'exhaustivité des prédictions.

Si un réseau apprend à détecter un visage parce qu'un neurone particulier détecte très bien les nez, il faut enseigner au réseau à rechercher d'autres composants caractéristiques du visage au cas où l'information du nez serait moins présente dans certaines images.

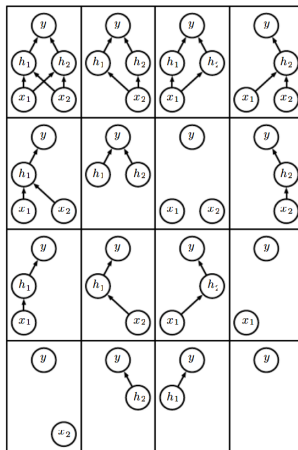
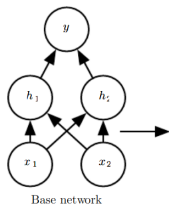


Image de [Goodfellow et al. 2015] Deep Learning

Pour mettre au point les valeurs optimales d'hyperparamètres (taux d'apprentissage, momentum, nombre de neurones par couche, etc.), une pratique commune est de déterminer un ensemble de valeurs possibles pour chaque hyperparamètre et de tester toutes les combinaisons. On conserve la combinaison qui minimise l'erreur sur l'ensemble de validation.

Pour réduire le risque espéré, on peut entraîner plusieurs modèles (formes de réseau) différents, et les faire voter pour dégager la prédiction la plus populaire. L'intuition derrière cette méthode est que les différents modèles se tromperont à différentes reprises...

Utiliser des ensembles de données de taille réduite peut induire un surapprentissage.

→ augmentation artificielle de la taille de la base de données en les altérant avec des transformations contrôlées.

Cette pratique est particulièrement utile en *traitement d'images* : un réseau de neurones ne sait pas reconnaître des formes dans une image qui sont transformées par translation, rotation ou changement d'échelle.

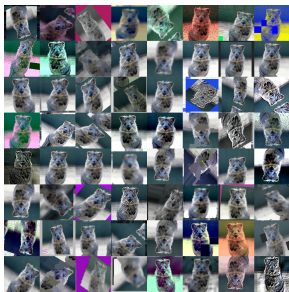
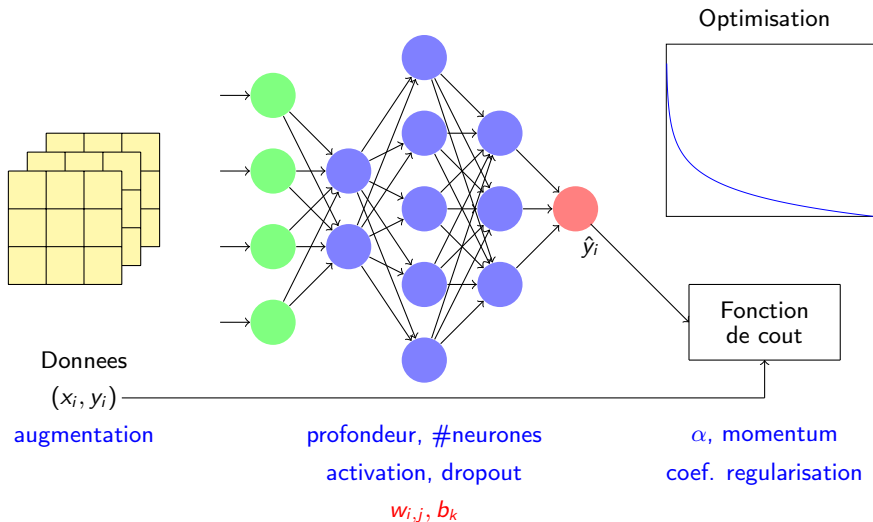


Image de <https://github.com/aleju/imgaug>



hyperparamètres et paramètres

Théorème d'approximation universelle (Cybenko 1989)

Toute fonction f , continue, de $[0, 1]^m$ dans \mathbb{R} , peut être approximée par un perceptron multi-couche à une couche cachée comportant suffisamment de neurones (avec une fonction d'activation sigmoïde).

Note: le théorème a été également prouvé avec la fonction reLU.

Le théorème **ne dit pas comment** déterminer ce réseau de neurones !

Principales architectures de deep learning

- **Réseaux convolutifs (CNN)** : c'est le réseau le plus utilisé pour le traitement et la reconnaissance d'images;
- **Réseaux autoencodeurs** fournit de bons "extracteurs" de caractéristiques propres au jeu de données;
- **Réseaux Génératifs Adversariaux (GAN)** permet de générer des ensembles de données;
- **Réseaux récurrents (RNN)** : adaptés pour des données d'entrée de tailles variables, utilisé en traitement de la parole, langage naturel.

Réseaux convolutifs (CNN)

L'idée centrale des réseaux convolutifs est basée sur "diviser pour régner" : c'est de concevoir une architecture comprenant des couches dédiées qui vont apprendre les prétraitements sur les images au lieu de les coder, afin d'extraire les caractéristiques de l'image. Celles-ci sont ensuite transmises à un réseau plus classique qui effectue la phase de reconnaissance.

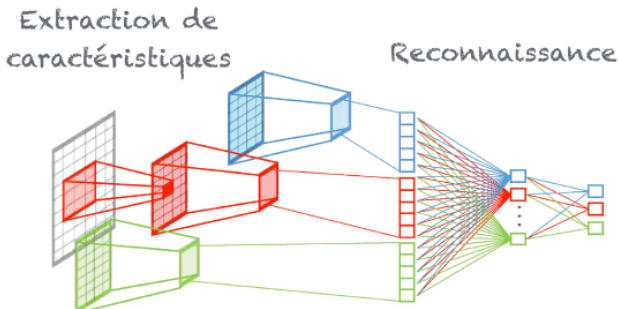


Figure: Principe de la convolution sur des images

Réseaux convolutifs (CNN)

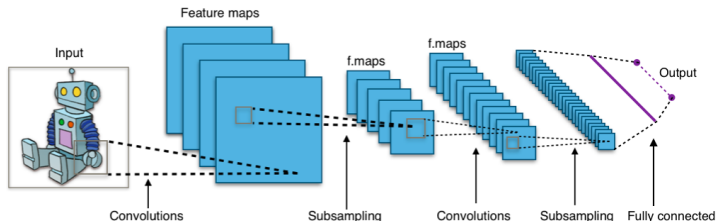


Figure: Exemple de CNN sur une image

- 1 les couches de *convolution* qui calculent la réponse à des filtres de convolution.
- 2 les couches de *pooling* (*max*, *mean* ou *sum pooling*) qui compressent l'information en réduisant la taille de l'image intermédiaire par sous-échantillonnage (*subsampling*) ;
- 3 les couches *fully connected* qui correspondent à des couches classiques de neurones formels totalement connectés ;
- 4 une couche de sortie, qui est la dernière du réseau. Elle porte notamment une fonction d'activation qui est spécifique au problème (classification ou régression).

Réseau autoencodeur

type de réseau particulier comportant une couche d'entrée et de sortie comprenant un même nombre de neurones mais avec une couche cachée plus restreinte.

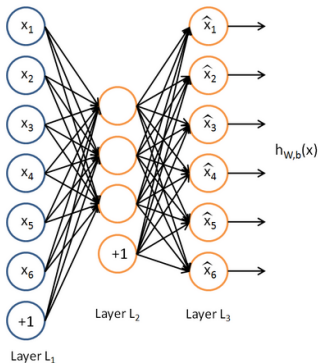


Figure: Architecture typique d'un réseau autoencodeur

Remarque : Autoencodeur avec fonction de coût MSE est équivalent à l'ACP.

Réseaux Génératifs Adversariaux (GAN)

comprend deux modèles (Génératif et Discriminatif) jouant l'un avec l'autre.

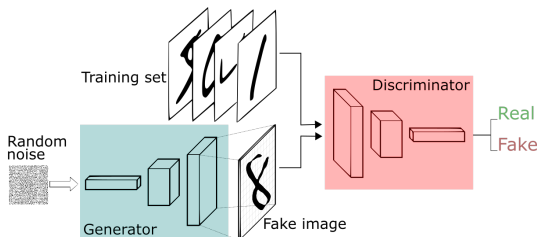


Figure: Architecture typique d'un réseau GAN

- 1 Le **Modèle Génératif** (*Le faussaire*), en prenant n'importe quelle entrée aléatoire essaiera de générer des choses réelles (courbes, images, sons, textes,...). Il ne connaît pas les données réelles, il essaiera seulement d'ajuster à partir de la rétroaction de l'autre modèle.
- 2 Le **Modèle Discriminatif** (*Le flic*), prendra comme base de données un ensemble de données générées à partir de l'autre modèle et des données réelles que nous voulons que l'autre modèle apprenne à simuler. Les résultats de sortie de ce modèle serviront à la rétropropagation de l'autre modèle.

Réseaux récurrents (RNN)

réseaux de neurones dans lesquels l'information peut se propager dans les deux sens, y compris des couches profondes aux premières couches. En cela, ils sont plus proches du vrai fonctionnement du système nerveux, qui n'est pas à sens unique.

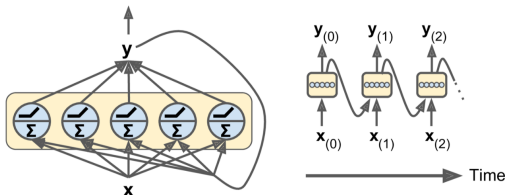


Figure: Couche de neurones récurrents, dépliée dans le temps

Ces réseaux possèdent des connexions récurrentes au sens où elles conservent des informations en mémoire : ils peuvent prendre en compte à un instant t un certain nombre d'états passés.

⇒ conviennent en particulier pour l'analyse de séries temporelles.

Cellule LSTM comprenant 2 vecteurs : $h_{(t)}$ l'état à court terme et $c_{(t)}$ l'état à long terme

⇒ **idée centrale** : le réseau peut apprendre ce qu'il faut stocker dans l'état à long terme, ce qui doit être oublié et ce qu'il faut y lire.

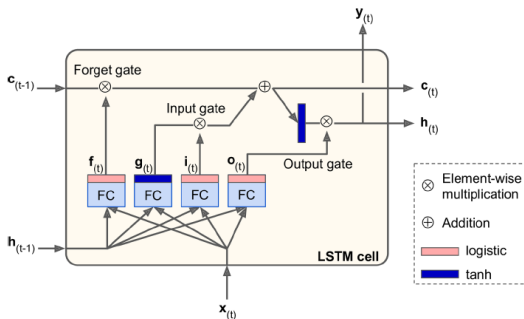


Figure: Cellule de longue mémoire à courte mémoire : architecture

⇒ Très utilisées en reconnaissance automatique de la parole et en traitement du langage naturel.

- L'apprentissage profond supervisé connaît aujourd'hui des réussites inégalées.
- Une des tendances actuelles est d'aller vers toujours plus de profondeur dans les réseaux, ce qui pose des problèmes de sous- et sur-apprentissage.
- Un autre enjeu actuel, transversal au précédent, est d'identifier des architectures à faible nombre de paramètres qui obtiennent des résultats aussi bons que des architectures profondes (problématiques d'embarquabilité, de consommation énergétique, etc.).

- *Pattern classification*, RO Duda, PE Hart, DG Stork - 2012
- *Pattern recognition and machine learning*, CM Bishop - 2006
- *Deep Learning*, I Goodfellow, Y Bengio, A Courville - 2016
- *Neural networks and deep learning*, A Géron - 2018

Les cours 7 et 8 ont été conçus avec Axel Carlier (IRIT-ENSEEIH) :
`axel.carlier@irit.fr`