Attention Mechanisms

PSTALN

Gabriel Marzinotto

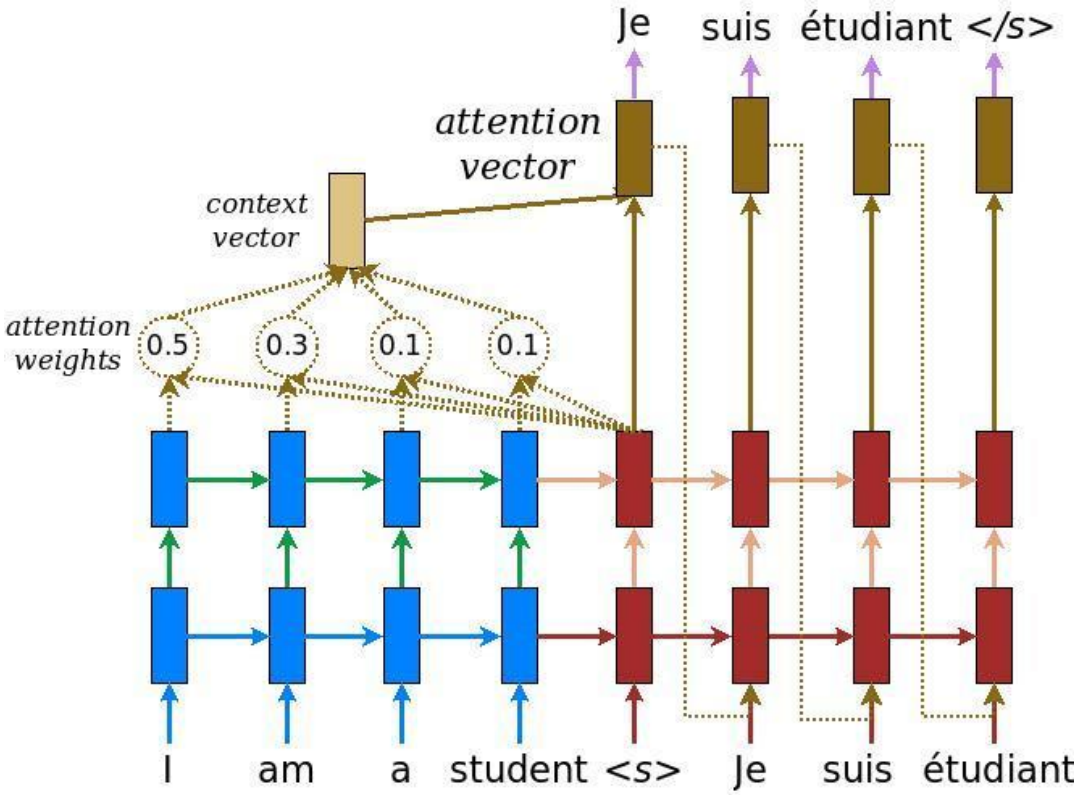
Mécanismes d'attention

- **Considérer certains éléments de la phrase lors de la prédiction**

- **Attention Weights**
  - Quel état caché contient le plus d'information utile
  - Dépend de l'état caché du décodeur

- **Vecteur de contexte**
  - Pondération des états cachés par les poids du modèle d'attention

- **Décision**
  - Contexte + état actuel

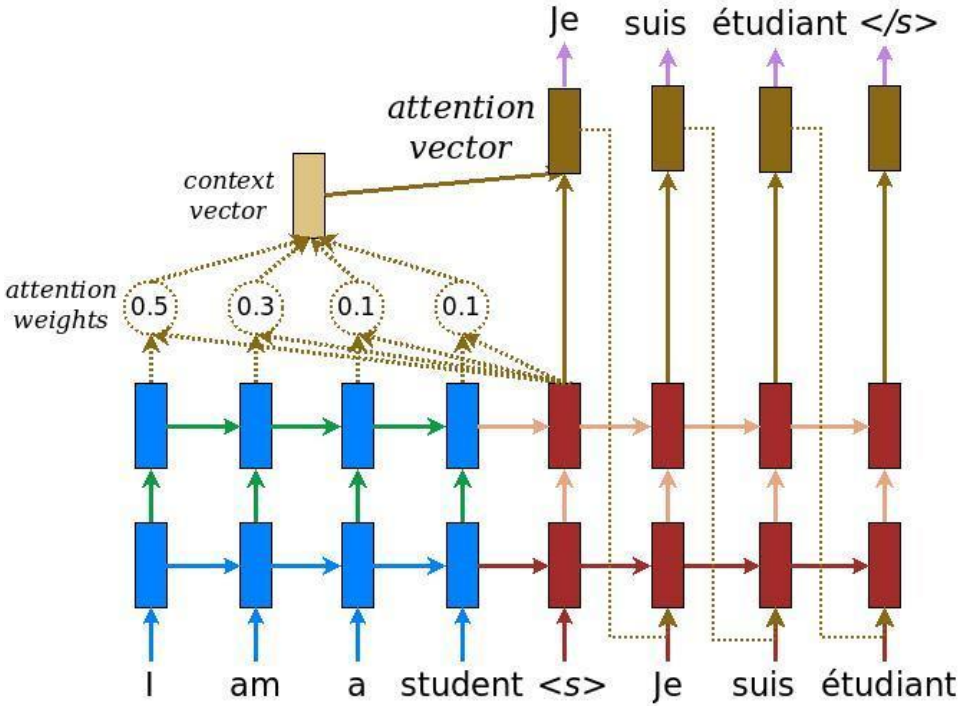


Mécanismes d'attention

$$\alpha_{ts} = \frac{\exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\text{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \quad \text{[Attention weights]}$$

$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s \quad \text{[Context vector]}$$

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W}_c[\boldsymbol{c}_t; \boldsymbol{h}_t]) \quad \text{[Attention vector]}$$



Score de pertinence entre les états de l'encodeur et du décodeur

Softmax pour générer une distribution de probabilité sur les mots de l'encodeur

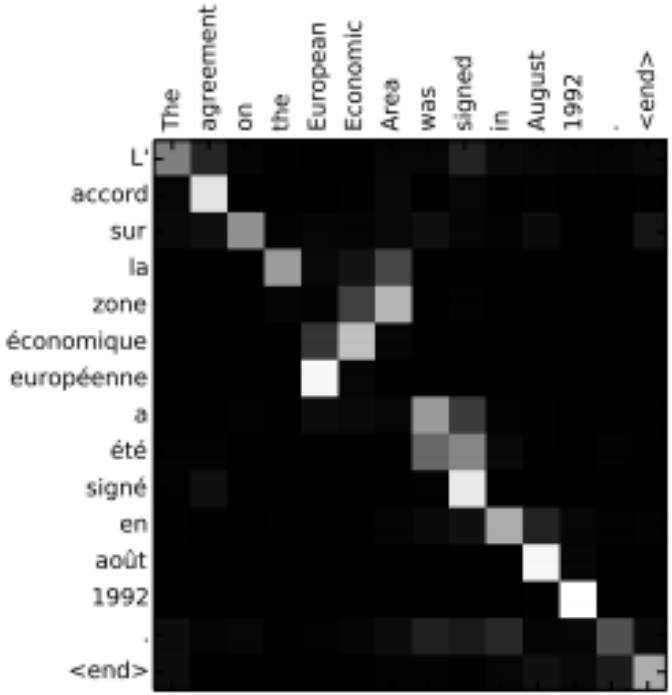
Ajoute d'une matrice Wc des poids d'attention

Mécanismes d’attention → Interprétation des modèles

Permet d’étudier quel partie de l’entrée est utilisé dans la prise de décision

Facilite l’étude et l’interprétation des modèles

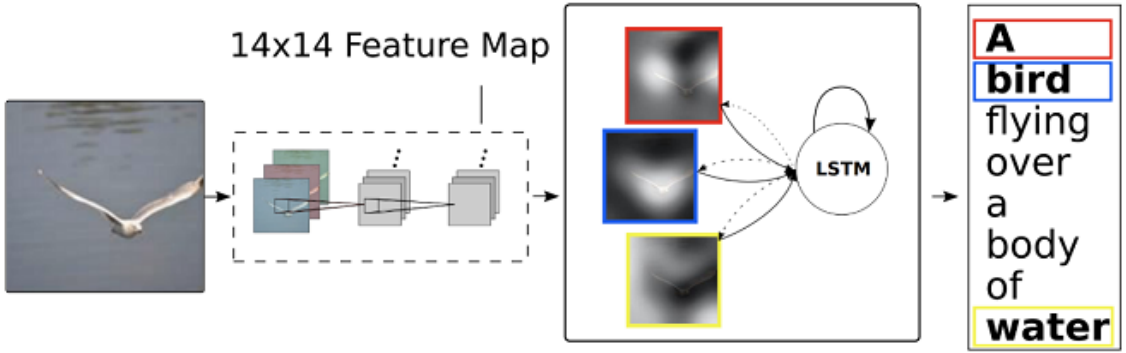
C’est moins une blackbox



Mécanismes d'attention → Synthèse et Résumés

Input: Article 1st sentence	Model-written headline
metro-goldwyn-mayer reported a third-quarter net loss of dlrs 16 million due mainly to the effect of accounting rules adopted this year	mgm reports 16 million net loss on higher revenue
starting from july 1, the island province of hainan in southern china will implement strict market access control on all incoming livestock and animal products to prevent the possible spread of epidemic diseases	hainan to curb spread of diseases
australian wine exports hit a record 52.1 million liters worth 260 million dollars (143 million us) in september, the government statistics office reported on monday	australian wine exports hit record high in september

Multimodal Attention Mechanisms



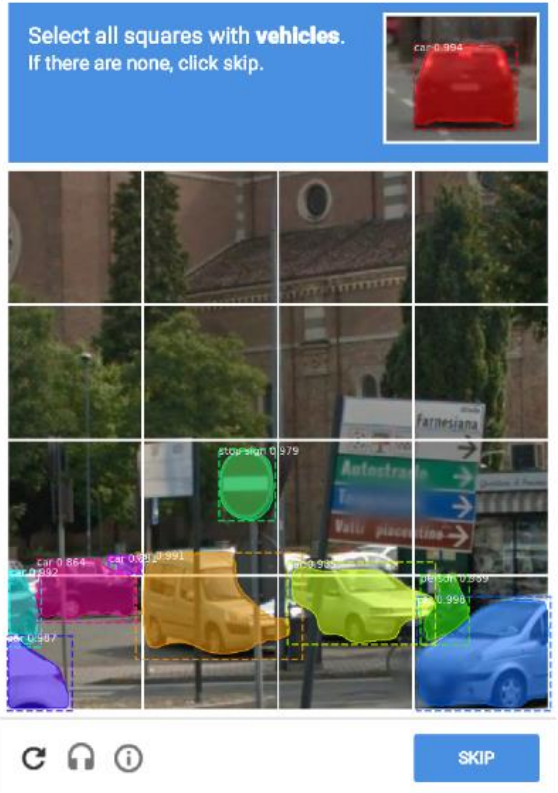
. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)

A woman is throwing a frisbee in a park.

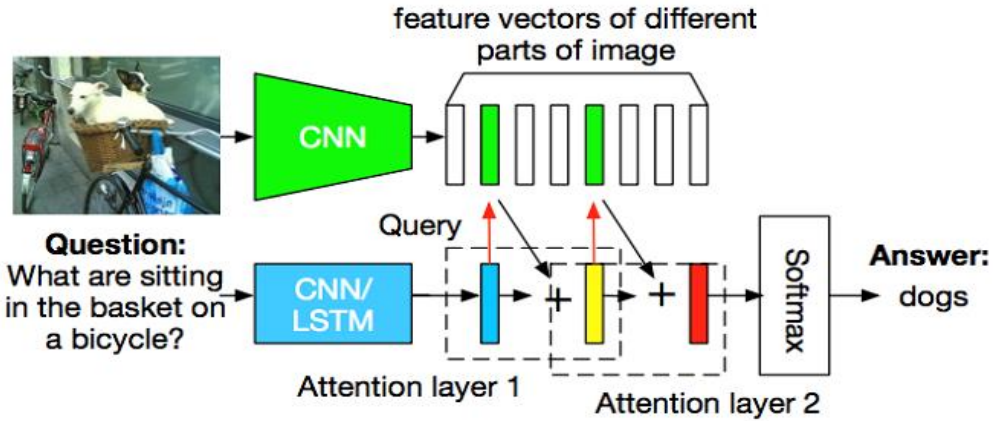
A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

Multimodal Attention Mechanisms → Captchas



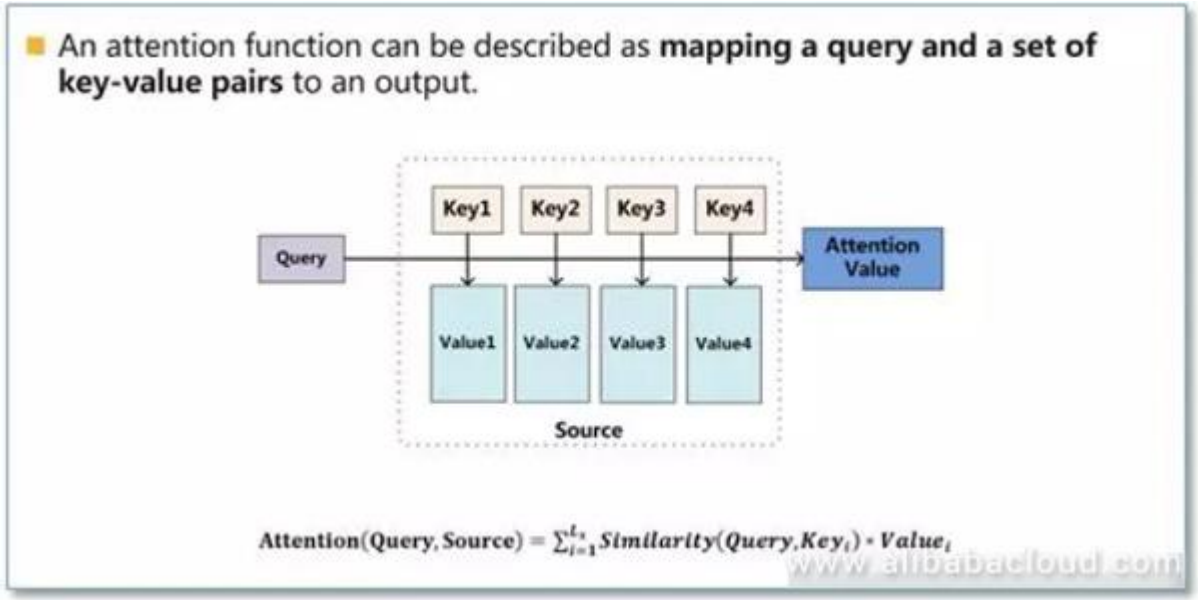
Multimodal Attention Mechanisms → Q&A



(a) Stacked Attention Network for Image QA

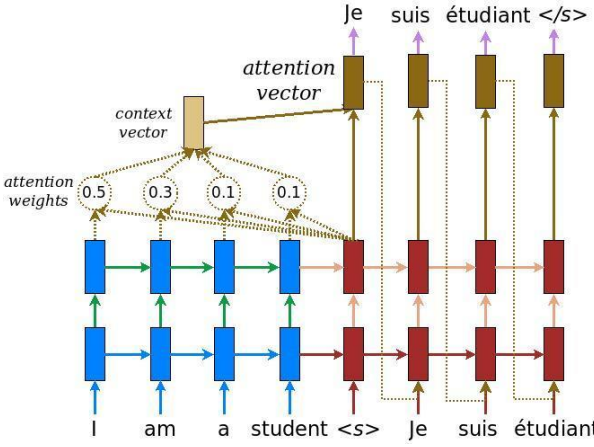
Original Image First Attention Layer Second Attention Layer

Mécanismes d'attention



$$\mathbf{c}_i = \sum_j a_{ij}\mathbf{s}_j$$

$$\mathbf{a}_i = \mathrm{softmax}(f_{att}(\mathbf{h}_i, \mathbf{s}_j))$$



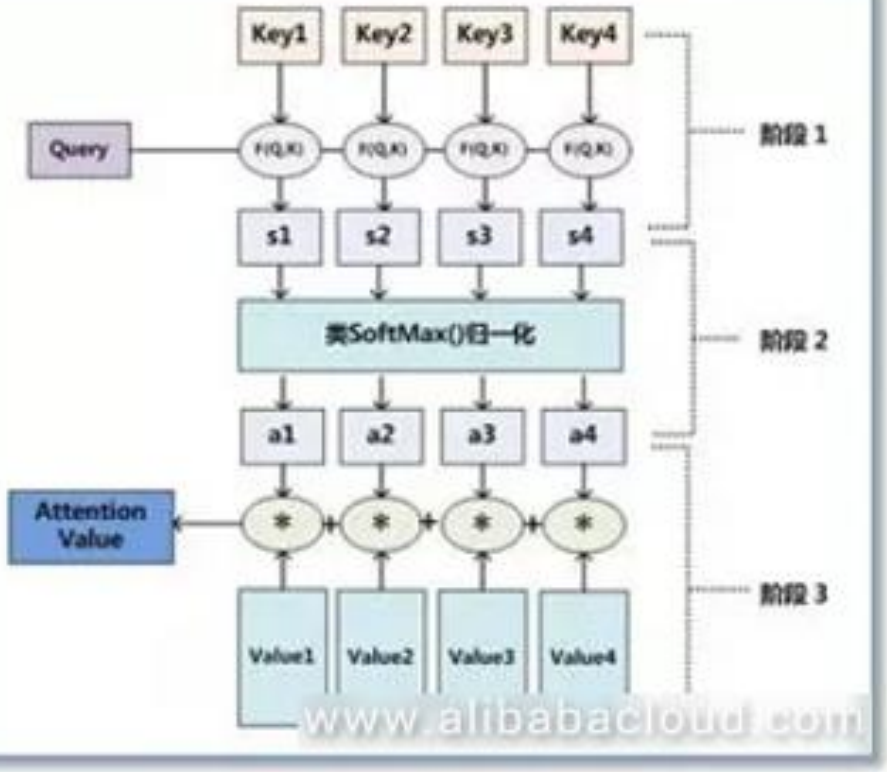
Nous avons des Requêtes, des Clés et des Valeurs
Les requêtes sont générés par le décodeur
Souvent Clés = Valeur

Mécanismes d'attention

$$f(Q,K_i)=\begin{cases} Q^T K_i & dot \\ Q^T W_a K_i & general \\ W_a[Q;K_i] & concat \\ v_a^T \tanh(W_a Q + U_a K_i) & perceptron \end{cases}$$

$$a_i = soft\max(f(Q,K_i)) = \frac{\exp(f(Q,K_i))}{\sum_j \exp(f(Q,K_j))}$$

$$\text{Attention}(Q,K,V) = \sum_i a_i V_i$$



Mécanismes d'attention

Mécanisme d'attention → trouver des poids pour faire une somme pondérée des états cachées

$$\mathbf{c}_i = \sum_j a_{ij}\mathbf{s}_j$$

$$\mathbf{a}_i = \text{softmax}(f_{att}(\mathbf{h}_i, \mathbf{s}_j))$$

Plusieurs types
Les plus importants sont:

- **Additive Attention :**
  - Original Version

$$f_{att}(\mathbf{h}_i, \mathbf{s}_j) = \mathbf{v}_a{}^\top \tanh(\mathbf{W}_1\mathbf{h}_i + \mathbf{W}_2\mathbf{s}_j)$$

- **Multiplicative Attention :**
  - Plus simple mais tout aussi performant
  - Plus rapide et moins gourmande en termes de mémoire

$$f_{att}(h_i, s_j) = h_i^\top \mathbf{W}_a s_j$$

- **Dot Product Attention**
  - Encore plus simple

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Mécanismes d'attention

- **Self Attention :**
  - Additive attention calculé sur soit même (pas de encodeur - décodeur)

$$f_{att}(\mathbf{h}_i) = \mathbf{v}_a{}^\top \tanh(\mathbf{W}_a \mathbf{h}_i)$$

- **Key Value Attention :**
  - Séparation des Clés et des Valeurs explicité

$$[\mathbf{k}_i; \mathbf{v}_i] = \mathbf{h}_i$$

$$\mathbf{a}_i = \mathrm{softmax}(\mathbf{v}_a{}^\top \tanh(\mathbf{W}_1 [\mathbf{k}_{i-L}; \ldots; \mathbf{k}_{i-1}] + (\mathbf{W}_2 \mathbf{k}_i)\mathbf{1}^\top))$$

$$\mathbf{c}_i = [\mathbf{v}_{i-L}; \ldots; \mathbf{v}_{i-1}]\mathbf{a}^\top$$

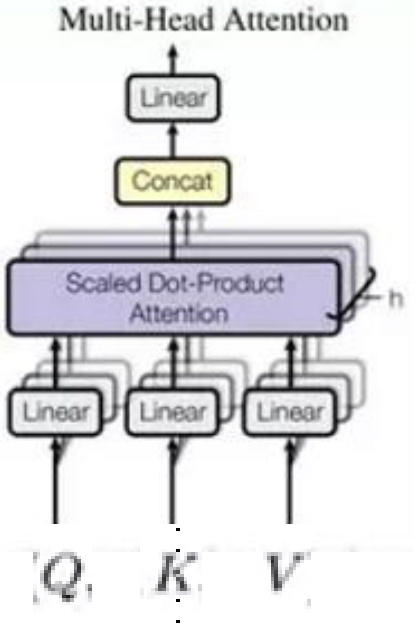
Mécanismes d'attention

Multi-Head Attention

$$\text{head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$

- Multi-head attention allows the model to jointly attend to **information from different representation subspaces** at different positions.



Adversarial Learning

PSTALN

Gabriel Marzinotto

Adversarial Learning

Selon Yann LeCun la découverte la plus importante du Deep Learning

Le terme Adversarial Learning est utilisé pour:

- **Adversarial Examples : Technique qui permet de piéger les réseaux de neurones**
- **Adversarial Training : Apprentissage avec une fonction de cout spéciale qui tient compte les exemples adversaires**
- **Generative Adversarial Networks : Génération d'images et synthèse de parole réalistes**
- **Adversarial Learning : Apprentissage sous contraintes grâce à la mise en compétition de plusieurs modèles de IA.**

Adversarial Examples

Technique qui permet de « piéger » les réseaux de neurones

- Bruiter les entrés pour induire le modèle à l'erreur

$$w^\top \tilde{x} = w^\top x + w^\top \eta$$

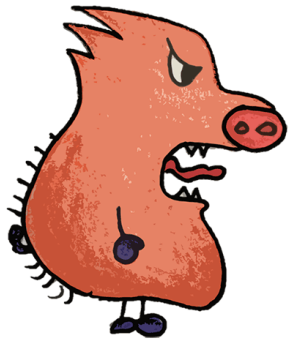
- Un humain ne pourrais pas faire la distinction

- Ajouter un bruit adversaire de magnitude epsilon

$$\eta = \epsilon \text{sign}\left(\nabla_x J(\theta, x, y)\right)$$

  - Par exemple, pour les images ε ≈ 1/256
  - Bruit vers la direction d'erreur maximal (gradient)

  - Comprendre les modèles pour augmenter la robustesse
  - Optimiser les annotations de données



https://adversarial-attacks.net

Adversarial Training

- **Adversarial Training :** **Apprentissage avec une fonction de cout spéciale qui tient compte les exemples adversaires**

$$\tilde{J}(\boldsymbol{\theta}, \boldsymbol{x}, y) = \alpha J(\boldsymbol{\theta}, \boldsymbol{x}, y) + (1 - \alpha) J(\boldsymbol{\theta}, \boldsymbol{x} + \epsilon \text{sign}\left(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y)\right)$$

$$L_{\text{adv}}(x_l, \theta) := D\left[q(y|x_l), p(y|x_l + r_{\text{adv}}, \theta)\right]$$
$$\text{where } r_{\text{adv}} := \underset{r;\|r\| \leq \epsilon}{\arg\max}\, D\left[q(y|x_l), p(y|x_l + r, \theta)\right],$$

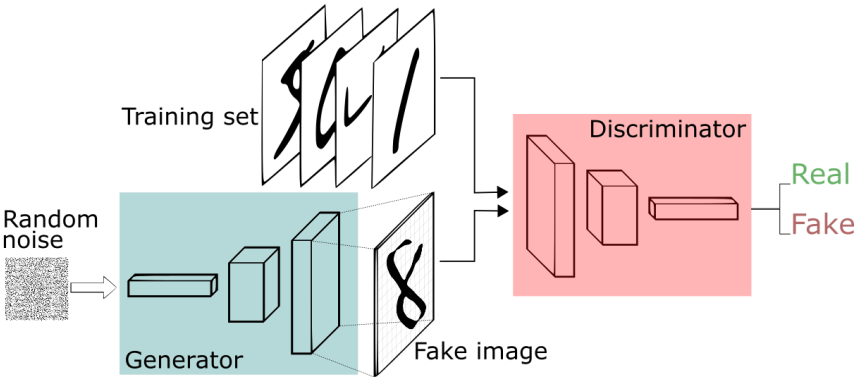
- **Virtual Adversarial Training :**
  technique pour les exemples non annotés

$$\text{LDS}(x_*, \theta) := D\left[p(y|x_*, \hat{\theta}), p(y|x_* + r_{\text{vadv}}, \theta)\right]$$
$$r_{\text{vadv}} := \underset{r;\|r\|_2 \leq \epsilon}{\arg\max}\, D\left[p(y|x_*, \hat{\theta}), p(y|x_* + r)\right],$$

Generative Adversarial Networks (GAN)

Génération d'images et synthèse de parole réalistes

- Mettre en compétition deux IA
- Une IA doit générer un contenu, l'autre doit juger sa qualité (authentique ou fausse)
- Certains informations sont partagés entre les deux IA à travers le gradient.



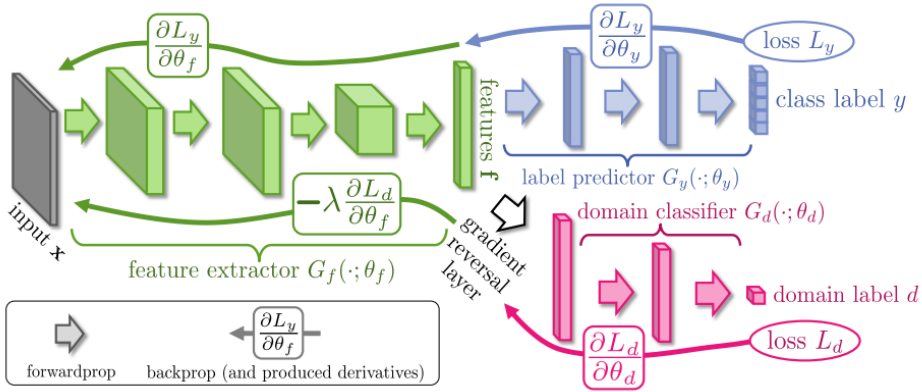
Adversarial Learning

Apprentissage sous contraintes grâce à la mise en compétition de plusieurs modèles

- C'est un cas plus général des GAN
- Une tâche principale que nous intéresse et des tâches contraintes
- Inversion du gradient des tâches adversaires

- Pour les GAN : Tâche Principale → Générer des images
  Contraintes → qu'une IA ne puisse pas savoir si elles sont vraies ou pas

- Pour l'adaptation de domaine: Tâche Principale → Analyse en Frame en domaine ouvert
  Contraintes → qu'une IA ne puisse pas reconnaitre le domaine



Difficultés de l'Adversarial Learning

- **Instabilité lors de l'apprentissage si les modèles sont mal initialisés**
  - Application progressive des contraintes

- **Problème de bias**
  - Quelle distribution suivent les donnés de la tâche adversaire ?

- **Relation entre la tâche principale et la tâche adversaire**
  - Tâches orthogonales → Bruit
  - Tâches trop corrélés → Confusion et dégradation 
  - Tâches corrélés mais associés aux sources de bias → Amélioration 

- **Complexité de la tâche adversaire**
  - Elle doit être modélisable
  - Cout de calcul raisonnable

Difficultés de l'Adversarial Learning

Le modèle est composé de 3 sous parties:

- Les couches partagées
- La tache principale
- La tache adversaire

$$\theta_f \longleftarrow \theta_f - \mu\left(\frac{\partial L_y^i}{\partial \theta_f} - \lambda\frac{\partial L_d^i}{\partial \theta_f}\right)$$

$$\theta_y \longleftarrow \theta_y - \mu\frac{\partial L_y^i}{\partial \theta_y}$$

$$\theta_d \longleftarrow \theta_d - \mu\frac{\partial L_d^i}{\partial \theta_d}$$

Utilisation d'un paramètre lambda qui module la pénalisation de la tâche adversaire

$$\lambda_p = \frac{2}{1+\exp(-\gamma \cdot p)} - 1$$

Lambda commence en 0 et puis incrémentera jusqu'à 1

Merci