

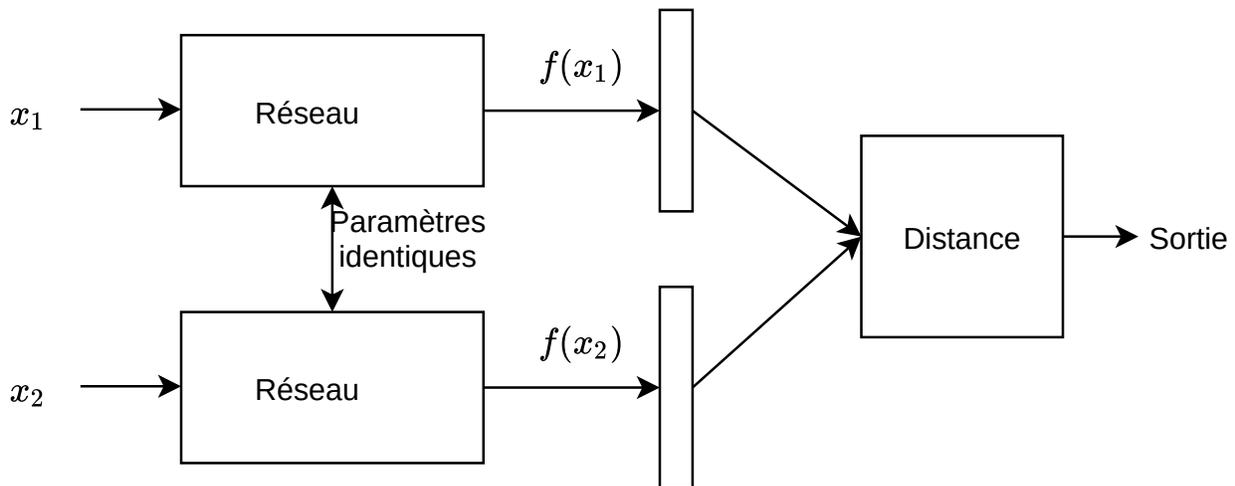
AS - TP 12

Réseaux Siamois

Nicolas Baskiotis - Benjamin Piwowarski

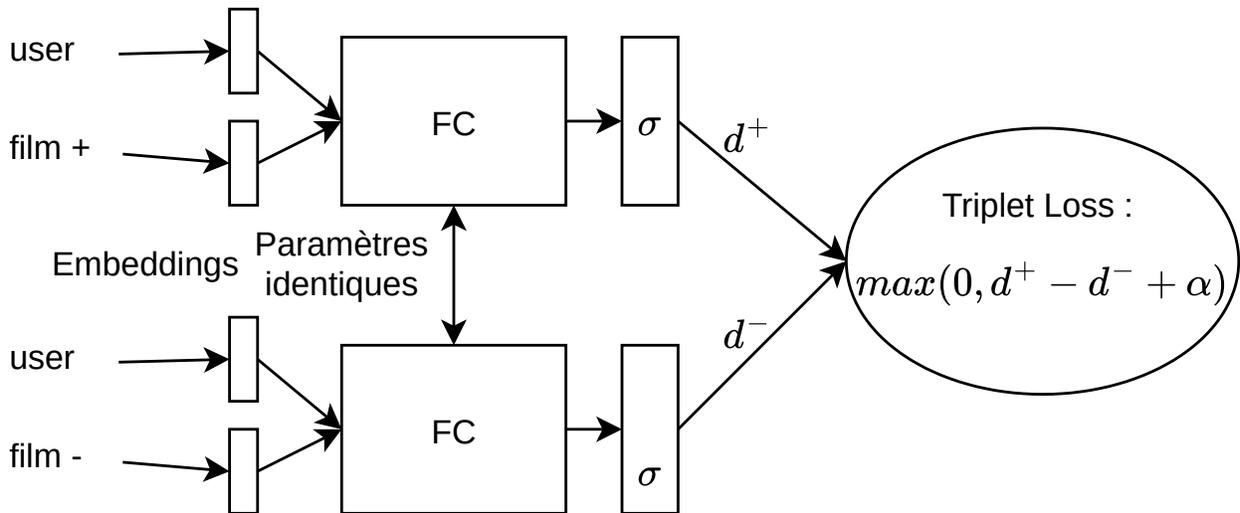
2019-2020

Un réseau siamois (ou *siamese network*) est un réseau qui utilise deux sous-modules identiques (même poids, même architecture). Un exemple est passé dans chaque sous-module puis la sortie est comparée (à l'aide d'une fonction simple comme une distance euclidienne ou possiblement un nouveau réseau de neurones). L'idée sous-jacente est que le réseau permet d'extraire les descripteurs adéquats pour comparer les deux entrées des sous-modules. Ce type d'architecture est très utilisé dans les tâches de détection de similarité/dissimilarité (est-ce que deux photos représentent le même objet par exemple) et également pour le one-shot learning.



Dans ce TP, nous allons utiliser l'architecture pour le problème de la recommandation de films (base movielens). La base contient une liste de notes attribuées par des utilisateurs sur des films et la tâche est de prédire pour chaque utilisateur la liste des films triée du film qu'il a le plus de chance d'aimer vers le film qu'il a le moins de chance d'aimer. Dans notre contexte, le sous-module du réseau siamois est utilisé pour prédire un scalaire représentant un score de dissimilarité (une distance) entre un utilisateur et un film. Au début du sous-module, deux couches de représentation sont utilisées, l'une pour encoder les utilisateurs, l'autre pour encoder les films. Les deux représentations sont ensuite concaténées et la représentation résultante est passée dans un réseau fully-connected (par exemple une succession de couches linéaires avec des ReLU pour l'activation). La dernière activation est une sigmoïde pour obtenir un score entre 0 et 1, 0 indiquant une grande préférence de l'utilisateur pour le film et 1 un film qu'il n'a pas aimé du tout.

Pour entraîner le réseau, nous allons utiliser un *triplet loss* : pour un utilisateur u donné, un film m^+ qu'il a aimé est tiré au hasard ainsi qu'un film m^- qu'il n'a pas aimé. Soit d^+ la sortie du sous-module pour le couple (u, m^+) et d^- la sortie pour le couple (u, m^-) ; l'objectif est que $d^+ \ll d^-$. Le triplet loss est défini selon la formule : $\max(0, d^+ - d^- + \alpha)$, avec α une constante pour définir la marge.



Souvent en recommandation les données sont très déséquilibrées (les gens ont tendances à noter essentiellement les items qu'ils apprécient). Plutôt que de tirer m^- parmi les films mal notés, une autre technique - le negative sampling - consiste à le tirer parmi tous les films - bien, mal ou pas notés.

Expérimentations Implémenter un **Dataset** qui permet de renvoyer un triplet utilisateur, film positif, film négatif. Vous devrez binariser le jeu de données (tous les films au-dessus d'une note sont positifs, les autres négatifs). Implémenter les deux versions : négative sampling et tirage uniquement parmi les films notés.

Implémenter une baseline simple, qui utilise la popularité d'un item (score = nombre de notes positives parmi l'ensemble des utilisateur).

Implémenter le réseau et le triplet loss. Comparer les deux versions, et la baseline "popularité".

Vous pouvez pré-traiter les données afin de supprimer les films et les utilisateurs qui ont trop peu de notes. L'ensemble de test sera constitué d'une partie des notes (non fournies en entraînement bien sûr) : vous ordonnerez tous les couples de l'ensemble de test par score croissant et vous utiliserez une mesure d'AUC et/ou de NDCG pour l'évaluation (vous pouvez également tracer les courbes ROC ou précision/rappel avec les outils de `sklearn`).

Exercice complémentaire : Bayesian Personalized Ranking (BPR) Un autre modèle courant en recommandation est BPR ; au lieu d'utiliser une Hinge loss, ce modèle optimise un maximum a posteriori :

$$\begin{aligned} \log p(\theta|\mathcal{D}) &= cste + \log p(\theta) + \log p(\mathcal{D}|\theta) \\ &= cste + \log p(\theta) + \log \prod_{(u, m^+, m^-)} p(u \text{ préfère } m^+ \text{ à } m^-) \end{aligned}$$

et où on suppose que

- Chaque paramètre a un a priori normal avec une variance fixe (hyperparamètre)
- La probabilité de préférer est donnée par la sigmoïde de la différence des scores, i.e.

$$p(u \text{ préfère } m^+ \text{ à } m^-) \stackrel{def}{=} \text{sigmoid}(s^+ - s^-)$$