Équations

Alafate ABULIMITI

Février 2019

1 Introduction

Nous devons examiner de plus près ce modèle seq2seq, puis définissons l'équation. Le modèle seq2seq est divisé en deux parties distinctes, une pour le Encoder et une pour le Decoder.

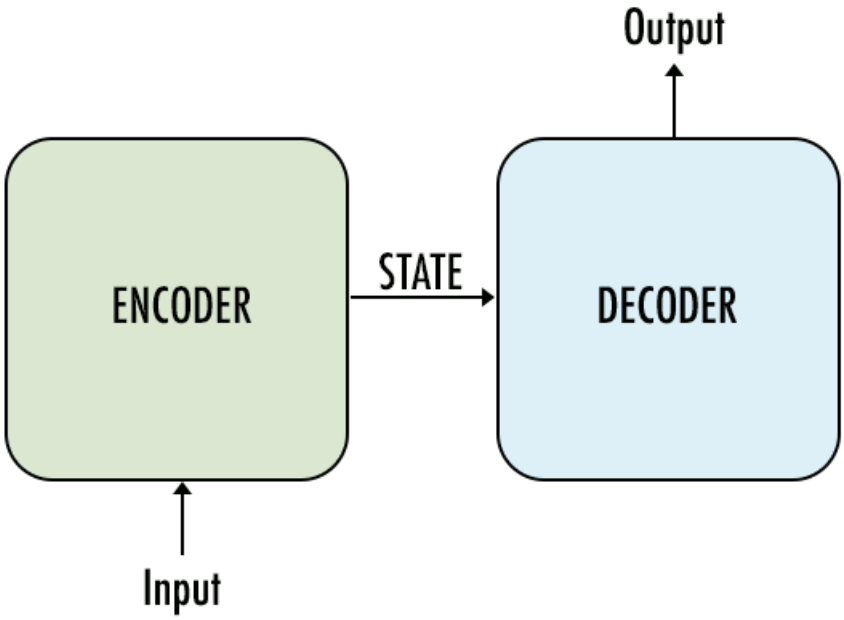

Figure 1: Seq2Seq Modèle

2 RNN

Avant d'introduire le modèle seq2seq, nous introduisons d'abord le modèle RNN classique.

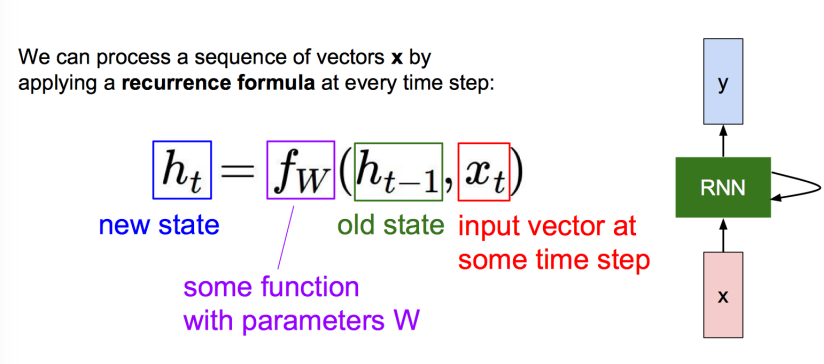

Figure 2: RNN classique

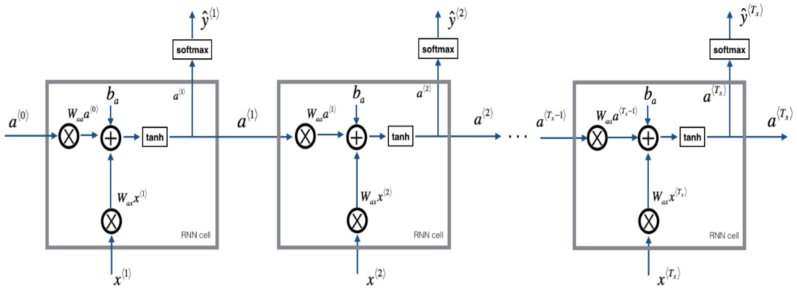

Figure 3: Structure de RNN classique

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}X^{<t>} + b_a)$$
$$\hat{y}^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

J'utiliserai ces conventions symboliques pour représenter ces indices matriciels, par exemple W_{ax}, le deuxième indice signifie que W_{ax} doit multiplier Avec un x, le premier indice a indique qu'il est utilisé pour calculer une variable a. Le même, vous pouvez voir W_{ya} ici est multiplié par a, utilisé pour calculer y.

Comme indiqué ci-dessus:

✓ n est la taille de séquence.

✓ $a^{<t>} \in \mathbb{R}^{1\times n}$ est la valeur d'activation au moment de t.

- ✓ $\hat{y}^{<t>}$ est le nouvel état au moment t , $\hat{y} \in \{0,1\}^{1\times n}$.
- ✓ $W_{aa} \in \mathbb{R}^{n-1\times n-1}$.
- ✓ $W_{ax} \in \mathbb{R}^{n\times n}$.
- ✓ $W_{ya} \in \mathbb{R}^{n\times n}$.
- ✓ $b_a \in \mathbb{R}^{n\times}$ est la valeur de déviation pour calculer $a^{<t>}$.
- ✓ $b_y \in \mathbb{R}^{n\times 1}$ est la valeur de déviation pour calculer $\hat{y}^{<t>}$.
- ✓ X_t est l'entrée au moment t , $X \in \mathbb{R}^{n\times 2}$.

3 Encoder

Ici, nous utilisons le bi-direction encoder le plus courant.Une séquence de vecteur $X = (x_1, ...x_{T_x})$. nous définissons :

- ✓ h_t est un état caché au moment t, $h \in \mathbb{R}^{n\times 1}$.
- ✓ $c \in \mathbb{R}^{n\times 1}$ est un vecteur généré à partir de la séquence des états caché.
- ✓ f et q sont des fonctions non linéaires.

Nous avons:

$$h_t = f(x_t, h_{t-1})$$
$$c = q(h_1, ...h_{T_x})$$

Cependant, étant donné que RNN standard traitent des séquences dans une série chronologique, ils ont tendance à ignorer les informations contextuelles futures. Une solution très évidente consiste à ajouter un délai entre l'entrée et la cible, ce qui donne au réseau le temps d'ajouter des informations de contexte futures, c'est-à-dire d'ajouter des informations futures sur la période M afin de prédire la sortie ensemble. L'idée de base du RNN bidirectionnel est de

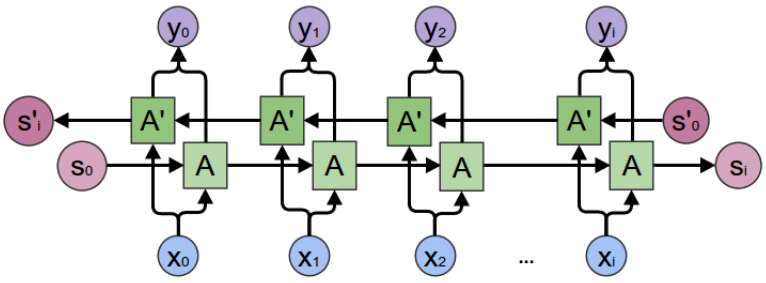

Figure 4: Bi-directional Encoder

proposer que chaque séquence d'entraînement constitue deux RNN aller-retour,

et que les deux soient connectés à une couche de sortie. Cette structure fournit des informations complètes sur le contexte passé et futur pour chaque point de la séquence d'entrée de la couche en sortie.

4 Decoder

Ici, nous utilisons le Decoder en fonction du contenu de cet article. nous

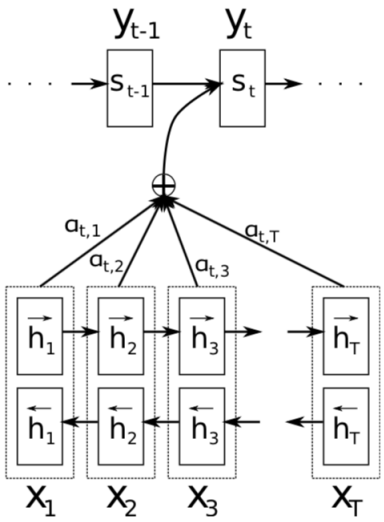

Figure 5: Decoder

définissons :

- ✓ s_t est un état caché RNN pour le moment t, $s \in \mathbb{R}^{1\times n}$.
- ✓ c_t est un vecteur généré à partir de la séquence des états caché. Le vecteur de contexte c_t dépend d'une séquence d'annotations $(h_1, ...h_{T_x})$ à laquelle un encoder mappe la phrase d'entrée,$c \in \mathbb{R}^{n\times 1}$.
- ✓ $\alpha_{tj} \in \mathbb{R}^{n\times n}$ est les poids de chaque h_j.
- ✓ $e_{tj} \in \mathbb{R}^{n\times n}$est un modèle d'alignement qui indique dans quelle mesure les entrées autour de la position$j$ et les sorties à la position i correspondent. Le score est basé sur l'état caché RNN s_{t-1} et la j-ème h_j de la phrase d'entrée.
- ✓ Nous paramétrons le modèle d'alignement a en tant que réseau de neurones à action directe formé conjointement avec tous les autres composants du système proposé.

✓ y_t est la sortie au moment de t $y \in \{0,1\}^{n\times 1}$.

Nous avons:

$$c_t = \sum_{j=1}^{T_x} \alpha_{tj} h_{tj}$$

$$e_{tk} = a(s_{t-1}, h_j)$$

$$\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{k=1}^{T_x} exp(e_{tk})}$$

$$s_t = f(s_{t-1}, y_{t-1}, c_t)$$

$$p(y_t|y_1, ...y_{t-1}, X) = g(y_{t_1}, s_t, c_t)$$

5 Loss Fonction

Dans notre projet, nous utilisons cette fonction de loss: categorical crossentropy.

$$L_i = -\sum_j \hat{y}_{i,j} \log(y_{i,j})$$

$\hat{y}$ sont les prédictions, y sont les varies valeurs, i désigne le point de données et j désigne la classe.

6 Référence

1. NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

2. Different types of RNNs