Neural Speed Reading via Skim-RNN

Pierre Stefani & Dan Mimouni
AS 2018

Avantages du Modèle

- Facile à mettre en place sur un RNN classique
- Même entrée/sortie qu'un RNN classique
- Inférence plus rapide
- Pas ,ou peu, de baisse de précision

Présentation du Modèle

Composition

- 2 RNN :
  - un grand de taille d (Big RNN)
  - un plus petit de taille d' (Small RNN)
- Un module linéaire W pour choisir le RNN

Infèrence

A chaque mot le module linéaire W choisit s'il faut :

- lire complètement le mot, en passant dans le RNN principal et en mettant à jour la totalité de l'état caché
- le "skimmer", en ne mettant à jour qu'une petite partie de l'état caché via le Small RNN. Le reste de l'Ãľtat cachÃĺ est simplement copié.

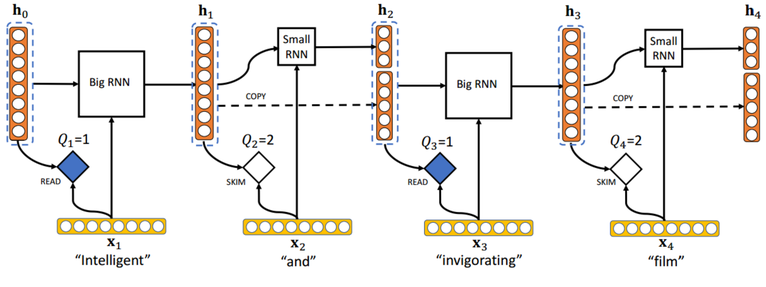

Figure 1: Skim-RNN

Modèles similaire

Il existe le modèle LSTM-JUMP qui ressemble au SKIM-RNN à la différence que LSTM-JUMP saute totalement contrairement au SKIM-RNN qui lit tout les mots.

Inférence

A chaque temps t, Skim-RNN prend l'entrée x et l'état caché précédent h_{t-1}.

Calcul de $p_t = softmax(W[x_t; h_{t-1}] + b)$

Tirage multinomial Q sur p_t : $Q \sim Multinomial(p_t)$

Si $Q = 1$ on choisit big RNN, small RNN sinon.

Apprentissage

Apprentissage des RNN

L'apprentissage des RNN se fait de la même façon qu'un RNN classique.

Apprentissage de W

Calcul de la loss :

$$\mathbb{E}Q \sim Multinomial(p_t)[L(\theta)] = \sum_Q L(\theta; Q)P(\theta) = \sum_Q L(\theta; Q) \prod p_t^{Q_j} \tag{1}$$

Le tirage multinomial de Q implique que la loss est non différentiable. Pour palier à ce problème, on approxime le tirage en utilisant une distribution gumbel-softmax.

On introduit la variable $r_t \in R^k$ différentiable :

$$r_t^i = \frac{\exp(\log(p_t^i) + g_t^i/\tau)}{\sum_j \exp(\log(p_t^j) + g_t^j/\tau)} \tag{2}$$

avec $g_t^j = Gumbel(0,1) = -\log(-\log(Uniform(0,1)))$

et τ la température.

Dans le calcul de la loss, on remplace les p_t par r_t .

Enfin, pour encourager le modèle a skimmer, on additionne à la loss la moyenne arithmétique de la log probabilité de skimmer , $\frac{1}{T}\sum_t \log(p_t^2)$, selon un hyperparamÃĺtre γ

On obtient donc :

$$L'(\theta) = L(\theta) + \gamma \frac{1}{T} \sum_t \log(p_t^2) \tag{3}$$

Tests

Benchmark : analyse de sentiments (Rotten Tomatoes)

Modèle	Dimensions	Accuracy	Tps train	Tps test	%skimmés
RNN	200	70.6%	1	1	-
SKIM-RNN	(200,10)	69.8%	1.04x	1.27x	47%
SKIM-RNN	(200,5)	**71.6%**	1.05x	1.3x	49%
SKIM-RNN	(100,10)	71.1%	1.09x	1.31x	47%
SKIM-RNN	(100,5)	70.7%	1.12x	1.33x	47%
LSTM-JUMP	(200,0)	69.2%	**1.25x**	**1.35x**	57%

Table 1: Comparaison RNN, Skim-RNN et LSTM-JUMP avec $\gamma = 0.01$

Gamma: régularisation pour encourager le skimrate

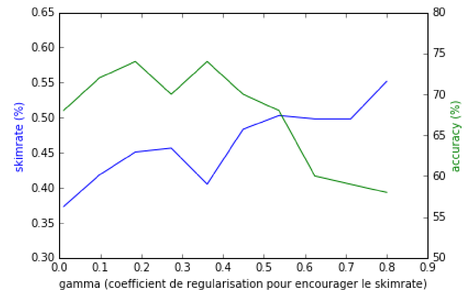

Figure 2: Pourcentage de mots skimmés et accuracy selon γ

Visualisation des mots lus et skimmés

Figure 3: Découpage d'une phrase avec mots skimmés