

# AS - TP5 - Graphes de calcul

## Module Softmax

Le module softmax permet de calculer une distribution de probabilités sur un ensemble discret d'éléments. Soit le vecteur  $x_1, \dots, x_n$  de scores réels (par exemple produits par d'autres modules), la définition du softmax est :

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_k e^{x_k}} \quad (1)$$

Le module softmax permet de faire du sampling sur des données discrètes.

## Réseaux de neurones récurrents pour la génération de séquences

Soit une séquence  $w_1, \dots, w_T$  d'éléments. Nous allons modéliser la probabilité  $P(w_1, \dots, w_T)$  à l'aide d'un réseau de neurones récurrent. Soit  $h_t$  un état cachés à l'instant  $t$ , nous avons  $P(w_1, \dots, w_T) = \prod P(w_t | w_{<t})$  et nous allons supposer que  $g_\tau(h_t) = P(w_t | w_{<t})$ . La fonction  $g$  mapper un état caché vers une distribution de probabilité (en utilisant une fonction softmax).

La récurrence du réseau sera définie par  $h_t = f_\theta(h_{t-1}, w_{t-1})$  qui correspond au modèle dynamique du réseau de neurone.

### 1 : Définition de la fonction $g$

Pour commencer, en considérant un espace latent de dimension  $N$ , et un espace d'entrée de dimension  $n$ , définissez sous forme de module (en utilisant `nngraph`) la fonction  $g_\tau$  sous la forme d'une fonction linéaire suivie d'un softmax.

### 2 : Définition de la fonction $h$

Définissez sous forme de module la fonction  $h_\theta$  de la forme

$$h_t = \tanh(\Theta^d h_{t-1} + \Theta^i w_{t-1}) \quad (2)$$

où  $w_{t-1}$  est le vecteur zéro, avec une unique valeur 1 correspondant à l'élément lu par le réseau.

### 3 : Définition du réseau

Nous allons définir un module correspondant au réseau de neurone "déplié" sur  $T$  timesteps. Ce réseau prendra en entrée une séquence de vecteur  $w_1, \dots, w_{T-1}$  et un état initial  $h_0 = 0$  et produira la séquence  $w_2, \dots, w_T$ .

**N.B :** Pour cela, les modules  $g_r$  et  $h_\theta$  qui apparaissent plusieurs fois doivent être répliqués à l'aide de la fonction *clone\_many\_times* fournie.

### 4 : Apprentissage

L'apprentissage d'un tel module sera effectué à l'aide d'un critère de type *Negative LogLikelihood* : *ClassNLLCriterion* dans torch

### 5 : Inférence

Dans le cadre de l'utilisation de ce réseau en inférence, la procédure consiste à sampler les entrées à l'aide de la fonction  $g_r$ . Ecrire la fonction qui permet de générer une nouvelle séquence à l'aide d'un modèle

### 6 : Chargement des données

- Les données seront préparées à l'aide du script *data-news-20Xx-to-character-l-data.sh*
- Les tenseurs seront générés à l'aide du script *prepare\_data.lua*
- Les données seront par la suite chargées comme dans le script *load\_exemple.lua*

**Suite... au prochain numéro**