Parallélisation des Réseaux de Neurones Récurrents: Pourquoi et Comment Procéder

HEINTCHEU TCHOKOUATA Ange Landryne

Departement d'informatique,
Facultés des sciences,
Université de Yaoundé I.

Plan

Idée générale

Définition

- Exécution d'un programme ou algorithme sur plusieurs processeurs simultanément;
- Réduction du temps d'exécution des programmes;

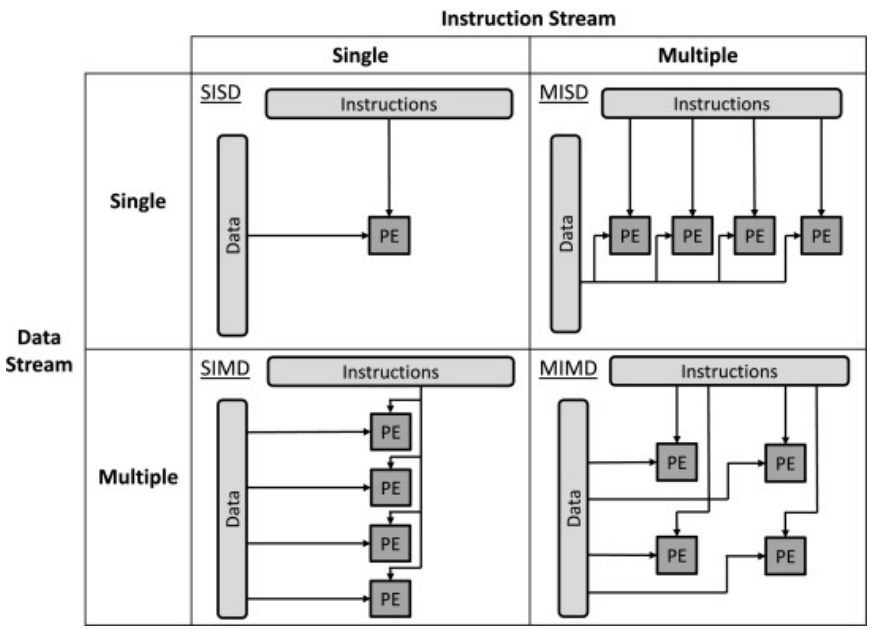
Disposition

- Prise en compte de l'architecture matérielle;
- Définition du modèle de parallélisation.

Architecture matérielle 1

Selon la taxonomie de Flynn on a 4 classes d'architectures [2]

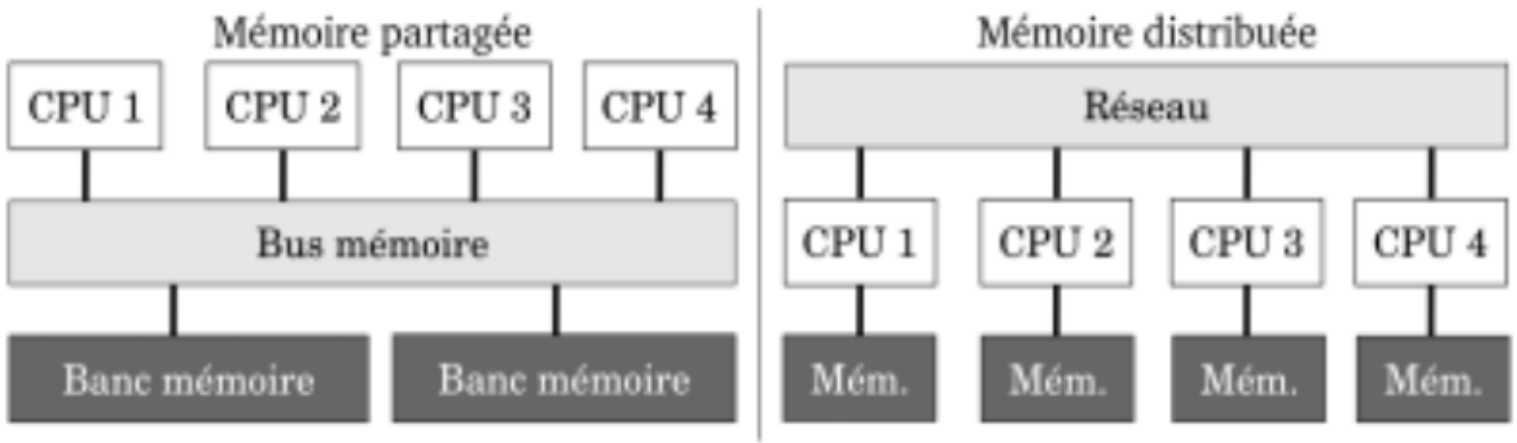
- Single Instruction Single Data (SISD);
- Single Instruction Multiple Data (SIMD) ;
- Multiple Instruction Single Data (MISD);
- Multiple Instruction Multiple Data (MIMD) .



Architecture matérielle 2

Dépendant de comment les processeur disposent des données en mémoire, on note 2 catégories d'architecture [4]:

- Architecture à mémoire partagé;
- Architecture à mémoire distribué.



Modèle de parallélisation

Il en existe 3 familles:

- **Parallélisation des tâches**: algorithme découpé en taches pour une exécution simultané;
- **Parallélisation des données**: données découpées en blocs pour un traitement simultané;
- **Parallélisation des instructions**: exécution simultané d'instructions d'une tache.

API de parallélisation

Définition

Interface de programmation d'applications(API): ensemble de fonctions et de types de données mis à la disposition d'un développeur pour écrire des logiciels

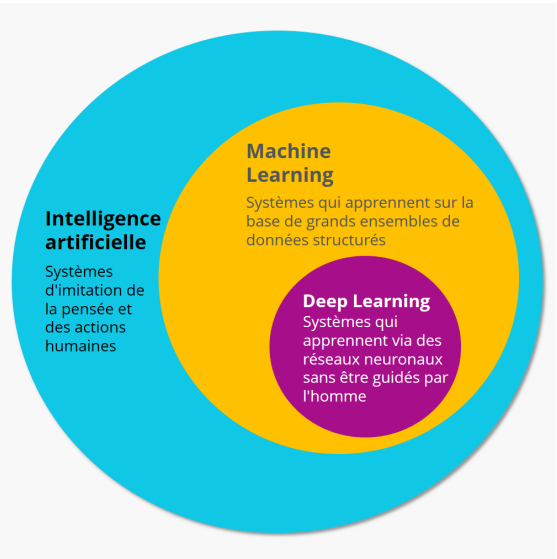
Quelques APIs de parallélisation

- POSIX thread (Pthread);
- Open Multi-Processing(Open MP);
- Message Passing Interfaces (MPI).

Les classes d'algorithmes d'IA

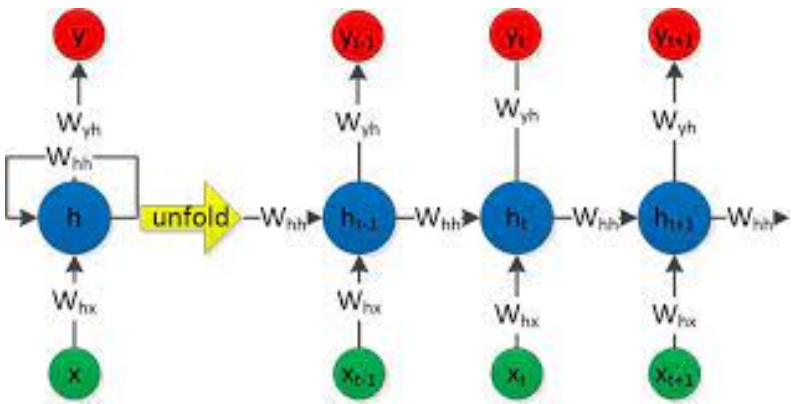
Résoudre les problèmes d'intelligence artificielle nécessite l'utilisation de différents types d'algorithmes

- Algorithmes qui s'adapte en fonction des données reçu : *Machine Learning* (ML).
- Catégories d'algorithmes de ML qui simulent le fonctionnent du cerveau humain pour s'adapter: *Deep Learning* (DL)



Les RNNs

- Algorithme de DL
- Adapté aux les données séquentielles
- Donnée séquentielles: suite ordonné de symboles (texte, vidéos...)
- Mémoire pour sauvegarder les dépendances qui existent entre les éléments de la séquence. [5]

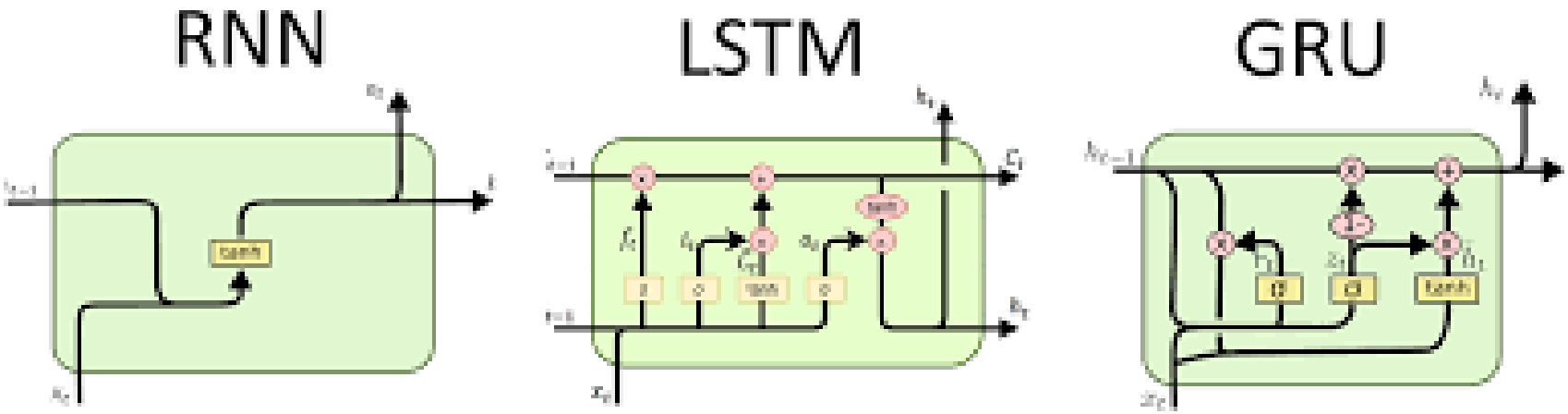


Les types RNNs 1

Dépendant de la taille des séquences à traiter il existe 3 variantes de RNN adéquates.

- les RNN standard [5]:
  - Utilise peu de paramètres;
  - Adapté aux courtes séquences.
- les GRU (*Gated Recurrent Units*) [1]:
  - Utilise un nombre moyen de paramètres;
  - Adapté aux moyen séquences.
- les LSTM (*Long Short-Term Memory*) [3]:
  - Utilise un plus grand nombre de paramètres;
  - Adapté aux longues séquences.

Les types RNNs 2



But de la parallélisation des modèles de DL

- Des grandes quantités de données produisent des modèles plus performants;
- Le temps d'apprentissage des modèles augmente avec la quantité de données.
- Entraînement rapide des modèles et conservation des performances.

Modèle de parallélisation pour les RNNS

Avec les RNNs on note:

- Utilisation des larges jeux de données pour produire des modèles performants;
- Interdépendance entre les instructions et les taches;
- Réalisation difficile d'une parallélisation d'instructions ou de taches.

Modèle adapté: **parallélisation des données**. Ce qui renvoie à une architecture SIMD.

Parallélisation des données pour les RNNS

L'exécution séquentielle d'un RNN s'effectue en 3 étapes:

- Création et initialisation du modèle;
- Entraînement du modèle;
- Mise à jour du modèle.

Pour une exécution parallèle on a:

- Création et initialisation du modèles principale;
- Création des copies du modèles principale;
- Subdivision des données en blocs;
- Entraînement simultané des blocs de données;
- Agrégation des copies de modèles.
- Mise à jour du modèle principale.

Méthodes d'agrégations

Difficultés

- Meilleur méthode d'agrégation;
- Limitation des pertes.

On note deux types de méthodes d'agrégations:

- Agrégation explicite: utilisation d'une fonction d'agrégation (la moyenne);
- Agrégation implicite: mise à jour systématique du modèle principale à la fin de l'exécution de chaque bloc de données ;

Récapitulatif

Pour réaliser la parallélisation des RNN:

- Architecture matérielle : SIMD;
- Modèle de parallélisation : parallélisation des données;
- Choix de la méthode d'agrégation.


Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio.
Learning phrase representations using rnn encoder-decoder for statistical machine translation.
arXiv preprint arXiv:1406.1078, 2014.


Michael J Flynn.
Some computer organizations and their effectiveness.
IEEE transactions on computers, 100(9):948–960, 1972.


Sepp Hochreiter and Jürgen Schmidhuber.
Long short-term memory.
Neural computation, 9(8):1735–1780, 1997.


T Rauber and G Rünger.
Parallel programming: For multicore and cluster systems.[sl]: Springer science & business media, 2013.


Merci!