Résultats concernant l'arithmétique d'intervalle appliquée aux réseaux de neurones

Capitaine de corvette Guillaume Berthelot

Aout 2024

On considère l'arithmétique d'intervalle appliquée aux réseaux de neurones.

Théorème 1. *Soit un réseau de neurones quelconque. Soit A_ϵ un tenseur de symboles de bruits pour l'arithmétique d'intervalle. Soit f la fonction telle que l'image du tenseur à travers le réseau est donnée par $f(a_{\epsilon_i})$ pour chacun des éléments de A_ϵ.*

S'il existe une base de dimension n sur laquelle la projection de A_ϵ est injective et s'il existe une matrice W représentant f sur cette base, c'est-à-dire pour tout i appartenant à $\{1, \dots, n\}$, $f(a_{\epsilon_i}) = W a_{\epsilon_i} e_i$,

Alors,

$$Z^+ = |W||A_\epsilon|$$

est un vecteur bornant pour l'arithmétique d'intervalle. Si C est le vecteur centre, alors les bornes sont données par $C \pm Z^+$.

Proof. Soit p le nombre d'éléments de A_ϵ, on a

$$Z^+ = \sum_{i=1}^{p} |f(a_{\epsilon_i})| = \sum_{i=1}^{p} |W a_{\epsilon_i} e_i| = |W| \sum_{i=1}^{p} |a_{\epsilon_i} e_i|$$

d'où le résultat. □

Lemme 1. *Il existe une base canonique dans laquelle les opérations dites linéaires des réseaux de neurones sont représentables par une matrice. Il s'agit de la base générée par l'opération d'aplatissement.*

Lemme 2. *Si p est le tenseur des coefficients d'approximation d'une fonction d'activation, alors l'image de p dans la base canonique est l'aplatissement de p.*

Corollaire 1. *Si $L_1, R_1, L_2, R_2, \dots, L_n, R_n$ est une succession de couches linéaires et d'activations, et si A_ϵ est un tenseur de symboles de bruits pour l'arithmétique d'intervalle se projetant injectivement sur la base canonique de L_1,*

Si $p_1, p_2, \dots, p_n$ sont les tenseurs d'approximation projetés dans leurs bases canoniques respectives, alors l'image A_s de A_ϵ pour l'arithmétique d'intervalle est donnée par

$$Z_s = |W_r||A_\epsilon| = |((p_n W_n) \otimes (p_{n-1} W_{n-1}) \otimes \cdots \otimes (p_1 W_1))||A_\epsilon|$$

Autrement dit, il est possible de réduire à un produit matriciel le calcul de l'arithmétique d'intervalle appliquée à l'ensemble du réseau.

Proof. Soit L_n, R_n les deux dernières couches linéaire et d'activation,

Soit A_{n-1} le tenseur de symboles d'entrée, A_s l'image du tenseur d'entrée par l'approximation linéaire du doublet L_n, R_n. Alors

$$A_s = f_n(p_n A_{n_1}) = p_n W_n A_{n-1}$$

puis récursivement

$$= p_n W_n p_{n-1} W_{n-1} A_{n-2} =$$

$$\ldots = ((p_n W_n) \otimes (p_{n-1} W_{n-1}) \otimes \cdots \otimes (p_1 W_1)) A_\epsilon$$

Or A_ϵ se projecte injectivement sur son espace canonique.

D'où le résultat.

□

Théorème 2. *Soit d le tenseur de bruit généré par l'approximation d'une couche d'activation. Alors d se projette injectivement sur la base canonique.*

Proof. L'opération d'approximation est définie. □

En appliquant les résultats précédents, il est possible de réduire l'évaluation de l'arithmétique d'intervalle à un produit matriciel.

Théorème 3. *Soit $L_1, R_1, L_2, R_2, \ldots, L_n, R_n$ une succession de couches linéaires et d'activations, soit j une couche intermédiaire L_j. Alors les bornes pour l'arithmétique d'intervalle pour la couche d'activation R_j sont données par*

$$C_j \pm \sum_{l=1}^{j} |W_{r_i}||A_{d_i}|$$

où

$$W_{r_i} = (W_j) \otimes (p_{j-1} W_{j-1}) \otimes \cdots \otimes (p_i W_i) \quad \forall i \in \{1, \ldots, j\}$$

Proof. Ce résultat est immédiat. □

Corollaire 2. *Soit un réseau constitué de couches linéaires et de fonctions d'activations. L'algorithme 1 est un algorithme d'approximation affine pour ce réseau en temps polynomial*

Algorithm 1 Affine Approximation Algorithm for the Network

Step 1:
for each linear layer **do**
  Calculate W_l
  Create an empty list L_W
end for
Step 2:
Choose an input x
for each dimension of x **do**
  Establish a noise level δ
end for
Create the vector A_δ
Create a list A with the first element A_δ
Initialize a unit approximation vector p
for each layer of the network, in increasing order **do**
  if linear layer **then**
    Calculate the center
    for each element of the list L_W **do**
      Multiply it by $p \times W_l$ on left side
    end for
    Add W_l to the list L_W
    Create a copy $|L_W|$
    for each pair of elements $(|W_L|, |A|)$ **do**
      Stack the sum of the products: result Z^+
    end for
    Overwrite p with a unit vector of the output dimension
  else if activation layer **then**
    Calculate the bounds and store the result
    Define p, q, d
    Shift the center
    Add A_d to the list A
  end if
end for