

TP4

Lamjed Lounissi
INF4230
UAQM

May 8, 2020

1 Question 1: Simulation de la phase d'entraînement du réseau

Calcul de la fonction d'activation $a_k = g(\sum_j \omega_{j,k} a_j)$

avec $in_j = \sum_j \omega_{j,k} a_j$

Pour $x_4 = \{0, 1, 1, 0\}$ on obtient:

Calcul de la fonction d'activation a_5

$$a_5 = g(\sum_k \omega_{j,5} a_j)$$

$$in_5 = w_{1,5} * a_1 + w_{2,5} * a_2 + w_{3,5} * a_3 + w_{1,5} * a_4 = (0.02 * 0) + (0.05 * 1) + (0.05 * 1) + (-0.001 * 0) = 0.1$$

$$\text{d'où } a_5 = g(0.1) = 0.52$$

$$\text{avec } g(x) = (1 + e^{-in})^{-1}$$

Calcul de la fonction d'activation a_6

$$a_6 = g(\sum_k \omega_{j,6} a_j)$$

$$in_6 = w_{1,6} * a_1 + w_{2,6} * a_2 + w_{3,6} * a_3 + w_{1,5} * a_4 = 0.14$$

$$a_6 = g(0.14) = 0.53$$

Calcul de la fonction d'activation a_7

$$a_7 = g(\sum_k \omega_{j,7} a_j)$$

$$in_7 = w_{5,7} * a_5 + w_{6,7} * a_6 = 0,03$$

$$a_7 = g(0.03) = 0.51$$

Calcul de Δ_7 :

$$\Delta_7 = (1 - a_7) * a_7 * (1 - a_7) = 0,12$$

Calcul de Δ_5 :

$$\Delta_5 = (1 - a_5) * a_5 * w_{5,7} * \Delta_7 = 0,02$$

Calcul de Δ_6 :

$$\Delta_6 = (1 - a_6) * a_6 * w_{6,7} * w_{ij} = -0,02$$

1.1 Correction des poids

$$w_{ij} \leftarrow w_{ij} + \alpha * a_i * \Delta_j$$

on obtient alors les nouveaux poids:

$$w_{15} \leftarrow w_{15} + \alpha * a_1 * \Delta_5 = 0,0200$$

$$w_{16} \leftarrow w_{16} + \alpha * a_1 * \Delta_6 = -0,0200$$

$$w_{25} \leftarrow w_{25} + \alpha * a_2 * \Delta_5 = 0,0497$$

$$w_{26} \leftarrow w_{26} + \alpha * a_2 * \Delta_6 = -0,0094$$

$$w_{35} \leftarrow w_{35} + \alpha * a_3 * \Delta_5 = 0,0497$$

$$w_{36} \leftarrow w_{36} + \alpha * a_3 * \Delta_6 = 0,1506$$

$$w_{45} \leftarrow w_{45} + \alpha * a_4 * \Delta_5 = -0,0100$$

$$w_{46} \leftarrow w_{46} + \alpha * a_4 * \Delta_6 = 0,1000$$

$$w_{57} \leftarrow w_{57} + \alpha * a_5 * \Delta_7 = -0,0371$$

$$w_{67} \leftarrow w_{67} + \alpha * a_6 * \Delta_6 = 0,1132$$

De la même manière on fait le calcul pour les exemples x_5 et x_6 (voir fichier Excel).

2 Question 2: Simulation de la phase de test du réseau entraîné

L'objectif est de prédire la valeur d'attente dans un restaurant.

On exécute maintenant les 3 exemples du jeux de test x_1 , x_2 et x_3

On calcul après la fonction perte.

Après il faut faire une minimisation de l'erreur.

2.0.1 Phase de test

Pour $x_1 = \{0, 0, 1, 1\}$ on obtient :

On utilise ici les nouveaux poids initialisés dans la phase de rétro-propagation de la question 1.

Calcul de la fonction d'activation a_5

$$a_5 = g(\sum_k \omega_{j,5} a_j)$$

$$in_5 = w_{1,5} * a_1 + w_{2,5} * a_2 + w_{3,5} * a_3 + w_{4,5} * a_4 = 0,04$$

$$\text{d'où } a_5 = g(0.04) = 0,51$$

Calcul de la fonction d'activation a_6

$$a_6 = g(\sum_k \omega_{j,6} a_j)$$

$$in_6 = w_{1,6} * a_1 + w_{2,6} * a_2 + w_{3,6} * a_3 + w_{4,6} * a_4 = 0,25$$

$$a_6 = g(0.25) = 0,56$$

Calcul de la fonction d'activation a_7

$$a_7 = g(\sum_k \omega_{j,7} a_j)$$

$$in_7 = w_{5,7} * a_5 + w_{6,7} * a_6 = 0,0307$$

$$a_7 = g(0,0307) = 0.5$$

De la même manière on fait le calcul pour les exemples x_2 et x_3 (voir fichier Excel).

2.0.2 Calcul de l'erreur

Soit la fonction d'erreur suivant:

$$E(w) = \frac{1}{2} (\sum_{d \in D} \sum_{i \in \text{sorties}} (y_i - g_i(in))^2)$$

- y_i : sortie désirée
- a_i ou g_{in} : sortie obtenue
- D : ensemble des exemples d'entraînement

- sorties: ensemble des sorties du réseau

on obtient alors:

$$E = \frac{1}{2}((1 - 0.5)^2 + (0 - 0.5070)^2 + (1 - 0.51)^2) = 0.37$$

3 Question 3: Optimisation du réseau

Oui il est possible de modifier l'algorithme afin rétropropager la perte cumulées de plusieurs exemple.

Pour ce faire on peut cumuler les activations et les résultats de la phase de propagation en avant sur plusieurs exemple, et de passer la moyenne de ces résultats à la phase de rétropropagation.

La formule pour la modifications d'un poids avec N exemple, s'écrit donc:

$$w_{ij} = w_{ij} + \alpha * \frac{1}{N} * \sum (a_{ik} * \Delta_{jk})$$

ou k , allant de 1 à N , est l'indice de l'exemple qui a produit l'activation a_{ik} et le delta Δ_{jk} .

4 Question 4: Implémentation

4.1 Résultats numériques

4.1.1 Entraînement

Pour les résultats d'entraînement voir Fig 1, 2, 3

4.1.2 Test

Pour les résultats des tests voir Fig 4,5,6

Pour les résultats de calcul voir Fig.7

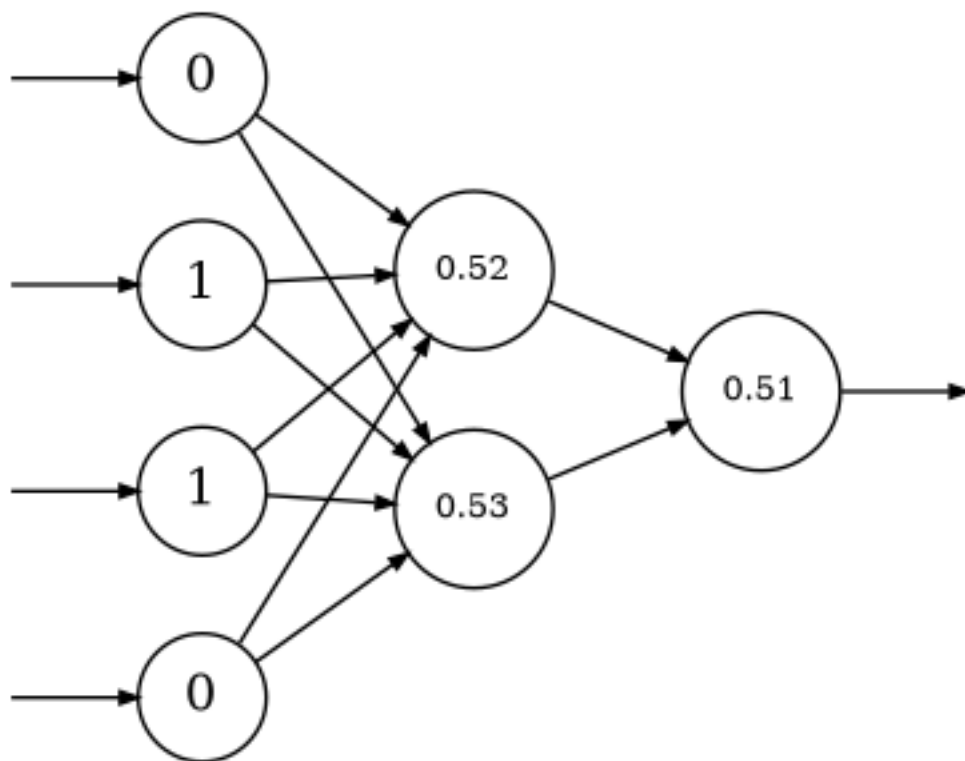


Figure 1: Training X_4

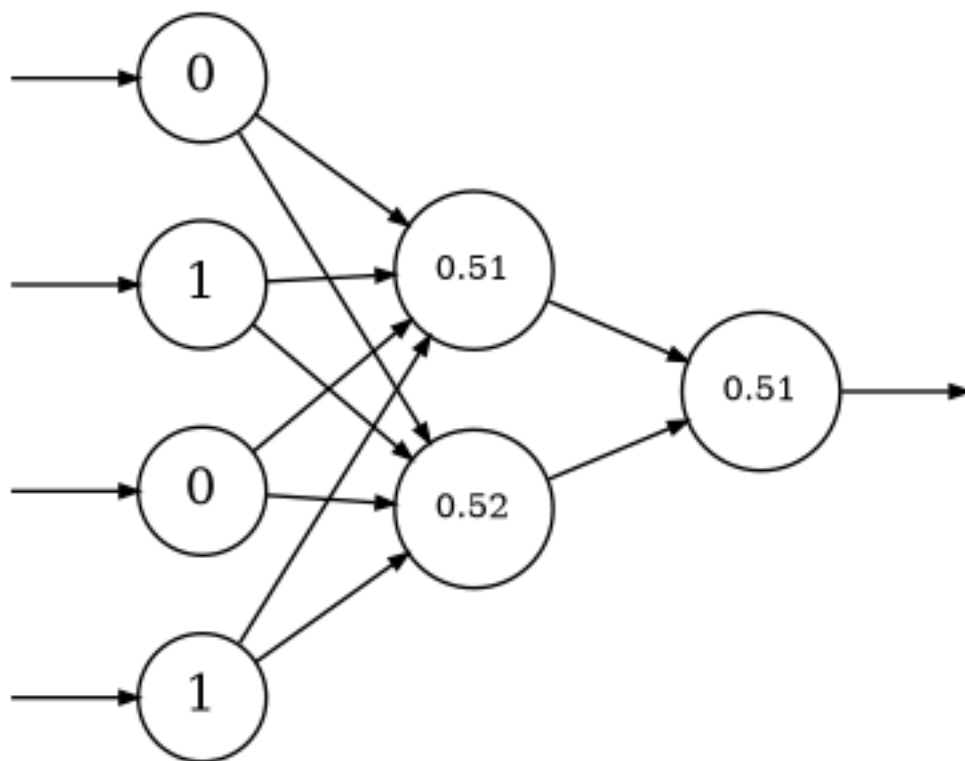


Figure 2: Training X_5

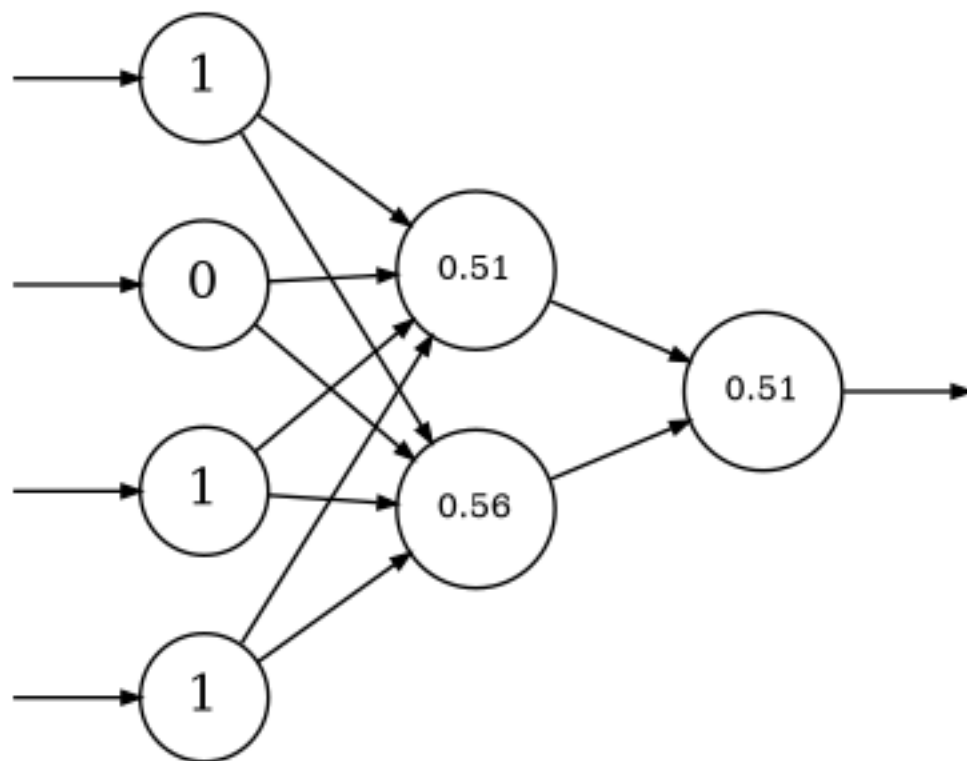


Figure 3: Training X_6

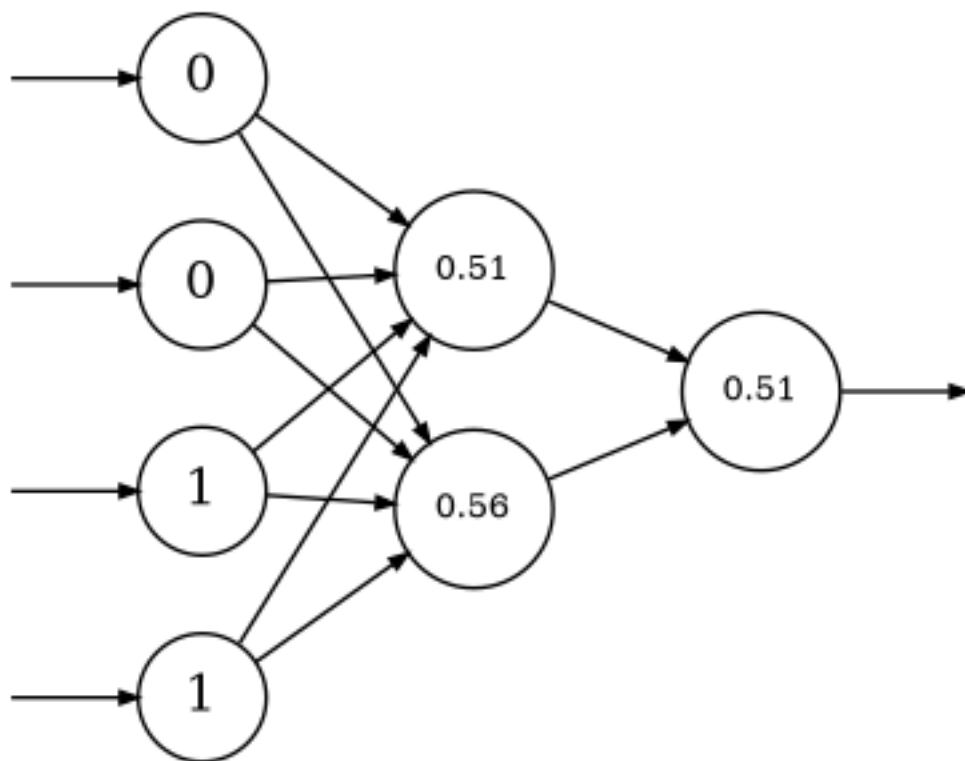


Figure 4: Test X_1

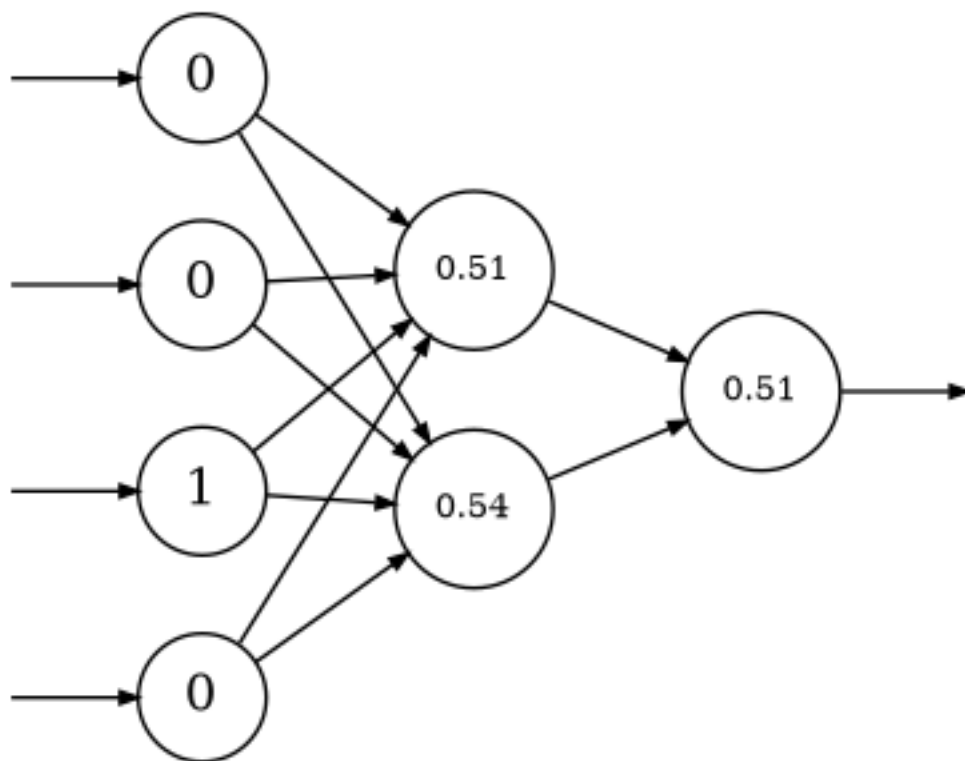


Figure 5: Test X_2

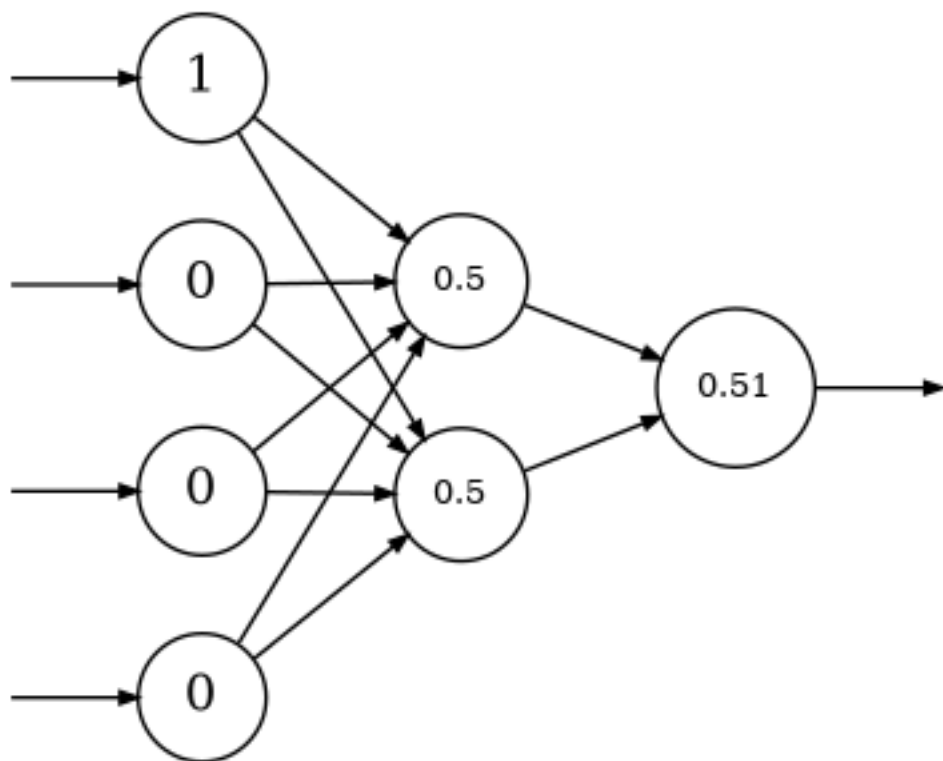


Figure 6: Test X_3

```

Training...
-----
Forward:
input : [0, 1, 1, 0]
layer : [0.52, 0.53]
output: 0.51

Backward:
w1: [[0.02, 0.0497, 0.0497, -0.01], [-0.02, -0.0094, 0.1506, 0.1]]
w2: [-0.037, 0.113]
-----
Forward:
input : [0, 1, 0, 1]
layer : [0.51, 0.52]
output: 0.51

Backward:
w1: [[0.02, 0.0499, 0.0497, -0.0098], [-0.02, -0.0101, 0.1506, 0.0993]]
w2: [-0.05, 0.1]
-----
Forward:
input : [1, 0, 1, 1]
layer : [0.51, 0.56]
output: 0.51

Backward:
w1: [[0.0197, 0.0499, 0.0494, -0.0101], [-0.0194, -0.0101, 0.1512, 0.0999]]
w2: [-0.037, 0.114]
=====

Testing...
-----
Forward:
input : [0, 0, 1, 1]
layer : [0.51, 0.56]
output: 0.51
-----
Forward:
input : [0, 0, 1, 0]
layer : [0.51, 0.54]
output: 0.51
-----
Forward:
input : [1, 0, 0, 0]
layer : [0.5, 0.5]
output: 0.51
Test score: 0.247

```

Figure 7: Résultats numériques