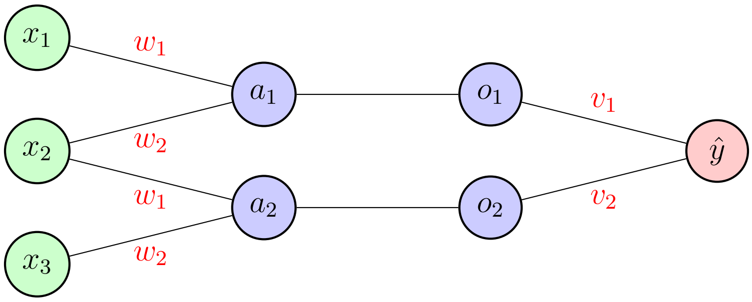
Réseau de neurones et rétropropagation
- L2 Math, IMIA -

Exercice 1

Objectif : mettre en oeuvre la rétropropagation du gradient pour un exemple simple de réseau de neurones.

On considère le mini réseau de convolution [1] suivant



Plus concrètement, le modèle s'écrit

$$
\begin{aligned}
a_1 &= x_1 w_1 + x_2 w_2 \\
a_2 &= x_2 w_1 + x_3 w_2 \\
o_1 &= max(0, a_1) \\
o_2 &= max(0, a_2) \\
\hat{y} &= o_1 v_1 + o_2 v_2
\end{aligned}
$$

Pour un exemple $((x), y) \in \mathbb{R}^3 \times \mathbb{R}$, la fonction de perte des moindre carrés est

$$\ell(W; \mathbf{x}, y) = \frac{1}{2}(y - \hat{y}(W, \mathbf{x}))^2$$

avec $W = (w_1, w_2, v_1, v_2)$.

1. Écrire les dérivées partielles $\frac{\partial \ell}{\partial v_1}$ et $\frac{\partial \ell}{\partial v_2}$. On pourra noter $e = y - \hat{y}$.
2. Écrire les dérivées partielles $\frac{\partial \ell}{\partial w_1}$ et $\frac{\partial \ell}{\partial w_2}$. Montrer les étapes intermédiaire de la dérivée en chaine.
   La dérivée de la fonction ReLU est $H(a) = \mathbb{I}(a > 0)$.
3. En utilisant les dérivations ci-dessus, complétez les détails manquants de la boucle de l'algorithme de rétropropagation ci-dessous qui est utilisée pour entrainer ce mini réseau de convolution.

1. Un réseau de convolution est une réseau de neurones dans lequel les neurones partagent des poids commun. Ces réseaux sont utilisée pour analyser des courbes, des images, etc.

Algorithme : Rétropropagation for le mini réseau de convolution

Entrées : Un ensemble d'apprentissage $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)$, taux d'apprentissage α

Initialisation : tirer w_1, w_2, v_1, v_2 aléatoirement

Répéter :

choisir aléatoirement une exemple $(\mathbf{x}_i, y_i)$

Passe avant :

Passe arrière :

4. Proposer au moins un critère d'arrêt pour la boucle de rétro-propagation.

Exercice 2

Objectif : illustrer l'impact de la taille des batch pour l'entrainement d'un réseau de neurones (très simple).

On considère un ensemble de données contenant 4 exemples. Pour chaque exemple on a une variable d'entrée x (variable explicative) et une variable de sortie y (variable à expliquer ou variable cible). Le réseau de neurone est réduit à un neurone, sans biais. La fonction d'activation est l'identité. On considère la fonction de perte des moindres carrés.

Ensemble des données

	x	y
1	1.0	2.0
2	2.0	4.0
3	3.0	6.0
4	4.0	8.0

La relation (théorique) entre x et y est $y = 2x$.

On rappelle que l'algorithme de descente de gradient stochastique (SGD) s'écrit comme suit

Algorithme de descente de gradient stochastique

Entrées :

ensemble d'apprentissage $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)$,
taux d'apprentissage α,
taille de batch n_b,
nombre maximum d'*epoch* N_{epoch}

Initialisation : $w = w_0$

Pour epoch = 1, $\cdots$,N_{epoch} :

Pour b in 0, $\cdots$,$[N/n_b] - 1$:

Calculer le gradient $G_b = 1/n_b \sum_{i=b*n_b+1}^{(b+1)*n_b} \nabla \ell(w, x_i, y_i)$

Mettre à jour les poids $w = w - \alpha G_b$

Mettre en oeuvre une epoch de l'algorithme SGD avec comme initialisations $w_0 = 0.5$, $\alpha = 0.1$, le modèle $\hat{y} = wx$, la fonction de perte $\ell = \frac{1}{2}(y - \hat{y})^2$ et le gradient $\nabla \ell = x(y - \hat{y})$. dans les cas suivants

1. $n_b = 1$
2. $n_b = 2$
3. $n_b = 4$

Et vérifier qu'on obtient les résultats suivants

Taille du batch	poids final	nombre de mises à jour
1	2.047	4
2	1.809	2
4	1.654	1

Que peut-on retenir de cet exercice concernant le choix de la taille des batchs ?