1 Démonstration des relations de rétropropagation

1.1 Définitions

Réseau. Un réseau de neurones artificiels est une application de $\mathbb{R}^p$ dans $\mathbb{R}^q$. Il comporte plusieurs **couches de neurones**. On notera l_c le nombre de neurones que compte la couche c, et dans la suite, n désignera l'indice de la dernière couche du réseau.

Neurone. Un neurone est une application $\varphi : \mathbb{R}^m \to \mathbb{R}$. L'image d'un vecteur $x = (x_1, \ldots, x_m)$ par un neurone est donné par $\varphi(x) = f(\sum_{i=1}^{m} p_i x_i - b)$, où les réels $p_1, \ldots, p_m$ sont appelés **poids**, où f est appelée la **fonction d'activation**, et où b est le **biais**. Ce dernier paramètre n'est pas pris en compte dans la suite : on peut, sans perte de généralité, ajouter un poids $p_{m+1} = -b$ supplémentaire au neurone et faire en sorte que la coordonnée correspondante de x, x_{m+1}, soit toujours égale à 1.

1.2 Structure du réseau

Dans notre cas, un neurone d'une couche donnée est "connecté" à *tous* les neurones de la couche précédente, c'est-à-dire que le vecteur formé des valeurs scalaires de chacun des neurones de la couche précédente constitue le vecteur d'entrée du neurone considéré. On notera $n_j^{(c)}$ la sortie du neurone j de la couche c et on notera $p_{i,j}^{(c)}$ le poids de la connexion entre $n_j^{(c)}$ et $n_i^{(c-1)}$.

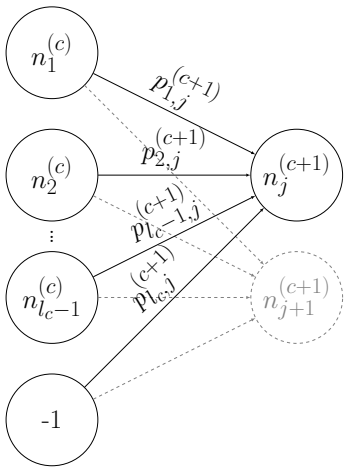
On a donc, pour $c \geq 2$,

$$n_j^{(c)} = f(\sum_{i=1}^{l_{c-1}} p_{i,j}^{(c)} n_i^{(c-1)}) = f(\sigma_j^{(c)}) \tag{1}$$

en posant $\sigma_j^{(c)} = \sum_{i=1}^{l_{c-1}} p_{i,j}^{(c)} n_i^{(c-1)}$.

Les neurones de la première couche du réseau sont quand à eux l'image du vecteur d'entrée du réseau, noté $x = (x_1, \ldots, x_p)$. On note de même $y = (y_1, \ldots, y_q)$ la sortie du réseau, qui par définition est donnée par

$$y = (n_1^{(n)}, \ldots, n_{l_n}^{(n)}).$$



Un neurone est connecté à chacun des neurones de la couche précédente.

En outre, on voit que le nombre de neurones par couche est arbitraire, mais que la taille d'une couche détermine le nombre d'entrées des neurones de la couche suivante.

1.3 Entraînement et rétropropagation

On souhaite **entraîner** le réseau de neurones à classer des vecteurs, c'est-à-dire à associer la même sortie vectorielle (à un certain seuil de tolérance près) à des vecteurs représentants le même objet, et on dit alors que de tels vecteurs appartiennent à la même **classe**. Pour atteindre cet objectif, on entraîne le réseau au cours de plusieurs **cycles** de correction. Un cycle correspond au calcul, pour quelques **échantillons** $(x, t) \in \mathbb{R}^p \times \mathbb{R}^q$, de l'image de x

par le réseau (notée y) que l'on compare avec $t = (t_1, \ldots, t_q)$: l'erreur obtenue permettra d'apporter une correction adéquate au réseau. L'**erreur quadratique** associée aux vecteurs y et t est donnée par

$$E = \frac{1}{2}\sum_{i=1}^{q}(t_i - y_i)^2. \tag{2}$$

Les seuls paramètres que l'on peut modifier une fois la structure du réseau fixée sont les poids, initialisés à de petites valeurs aléatoires. La correction consiste donc à calculer l'erreur dûe à chaque poids $p_{j,k}^{(c)}$, c'est-à-dire la valeur $\frac{\partial E}{\partial p_{j,k}^{(c)}}$. On utilise ensuite un algorithme "à direction de descente" pour trouver un minimum de la fonction d'erreur à partir de la donnée du gradient ainsi calculé. Cette opération, répétée pour chacun des poids, fait converger l'erreur globale vers zéro.

Pour un poids $p_{j,k}^{(c)}$ donnée, on appliquera donc la correction

$$\Delta p_{j,k}^{(c)} = -\alpha \frac{\partial E}{\partial p_{j,k}^{(c)}} \tag{3}$$

où α, le **taux d'apprentissage**, module l'importance de la modification apportée.

On pose dans la suite

$$e_{jk}^{(c)} = -\frac{\partial E}{\partial p_{j,k}^{(c)}} \frac{1}{n_j^{(c-1)}} \tag{4}$$

ce qui permet d'exprimer la correction par

$$\Delta p_{j,k}^{(c)} = \alpha e_{jk}^{(c)} n_j^{(c-1)}. \tag{5}$$

On souhaite démontrer les relations suivantes, qui permettent de calculer récursivement à partir de la dernière couche la correction à apporter à chacun des poids pour un échantillon donné, d'où le nom de **rétropropagation** :

$$\boxed{e_k^{(n)} = f'(\sigma_k^{(n)})(t_k - y_k)} \tag{6}$$

$$\text{et } \boxed{e_k^{(c)} = f'(\sigma_k^{(c)}) \sum_{i=1}^{l_{c+1}} p_{k,i}^{(c+1)} e_i^{(c+1)}}. \tag{7}$$

Cette notation à deux indices est licite : la démonstration montre qu'en fait, ces scalaires ne dépendent pas du premier indice, ce qui simplifie le travail algorithmique.

1.4 Démonstration

La fonction E est différentiable, on peut donc appliquer la règle de la chaîne. Comme le terme $p_{j,k}^{(c)}$ n'apparaît que dans l'expression de $n_k^{(c)}$,

$$\frac{\partial E}{\partial p_{j,k}^{(c)}} = \sum_{i=1}^{l_c} \frac{\partial E}{\partial n_i^{(c)}} \frac{\partial n_i^{(c)}}{\partial p_{j,k}^{(c)}} = \frac{\partial E}{\partial n_k^{(c)}} \frac{\partial n_k^{(c)}}{\partial p_{j,k}^{(c)}}. \tag{8}$$

On peut facilement calculer le membre de droite quelle que soit la couche c considérée. En effet, si on applique à nouveau la règle de la chaîne en remarquant que $p_{j,k}^{(c)}$ n'apparaît que pour $\sigma_k^{(c)}$,

$$\frac{\partial n_k^{(c)}}{\partial p_{j,k}^{(c)}} = \frac{\partial n_k^{(c)}}{\partial \sigma_k^{(c)}} \frac{\partial \sigma_k^{(c)}}{\partial p_{j,k}^{(c)}} = f'(\sigma_k^{(c)}) \frac{\partial}{\partial p_{j,k}^{(c)}} \left(\sum_{i=1}^{l_{c-1}} p_{i,k}^{(c)} n_i^{(c-1)} \right) = f'(\sigma_k^{(c)}) n_j^{(c-1)}. \tag{9}$$

Couche de sortie. Dans le cas de la couche de sortie, indexée n, la sortie $n_k^{(n)}$ d'un neurone est égale au terme y_k du vecteur de sortie. On en déduit que

$$\frac{\partial E}{\partial n_k^{(n)}} = \frac{\partial E}{\partial y_k} = \frac{1}{2} \frac{\partial}{\partial y_k} \left(\sum_{i=1}^{q} (t_i - y_i)^2 \right) = y_k - t_k.$$

On obtient donc la première relation.

$$e_{jk}^{(n)} = -\frac{\partial E}{\partial p_{j,k}^{(n)}} \frac{1}{n_j^{(n-1)}} = f'(\sigma_k^{(n)})(t_k - y_k).$$

Couche quelconque. Pour une couche quelconque c, le calcul de $\frac{\partial E}{\partial n_k^{(n)}}$ n'est pas aussi simple et c'est pourquoi on aboutit à une relation de récurrence. Supposons que l'on connaisse l'erreur sur la couche $c+1$. Les variations du neurone $n_k^{(c)}$ n'entraînent des variations de l'erreur qu'au travers des neurones de la couche suivante $c+1$ auxquels il est connecté, c'est-à-dire les neurones $n_i^{(c+1)}, 1 \leq i \leq l_{c+1}$. La règle de la chaîne permet alors d'écrire que

$$\frac{\partial E}{\partial n_k^{(c)}} = \sum_{i=1}^{l_{c+1}} \frac{\partial E}{\partial n_i^{(c+1)}} \frac{\partial n_i^{(c+1)}}{\partial n_k^{(c)}}.$$

En outre, par un calcul similaire à 9,

$$\frac{\partial n_i^{(c+1)}}{\partial n_k^{(c)}} = \frac{\partial n_i^{(c+1)}}{\partial \sigma_i^{(c+1)}} \frac{\partial \sigma_i^{(c+1)}}{\partial n_k^{(c)}} = f'(\sigma_i^{(c+1)}) \frac{\partial}{\partial n_k^{(c)}} \left(\sum_{j=1}^{l_c} p_{j,i}^{(c+1)} n_j^{(c)} \right) = f'(\sigma_i^{(c+1)}) p_{k,i}^{(c+1)}.$$

D'où la formule attendue :

$$\begin{aligned}
e_{jk}^{(c)} &= -\frac{\partial E}{\partial p_{j,k}^{(c)}} \frac{1}{n_j^{(c-1)}} \\
&= -\frac{\partial E}{\partial n_k^{(c)}} \frac{\partial n_k^{(c)}}{\partial p_{j,k}^{(c)}} \frac{1}{n_j^{(c-1)}} \text{ d'après 8} \\
&= \frac{\partial n_k^{(c)}}{\partial p_{j,k}^{(c)}} \frac{1}{n_j^{(c-1)}} \sum_{i=1}^{l_{c+1}} -\frac{\partial E}{\partial n_i^{(c+1)}} \frac{\partial n_i^{(c+1)}}{\partial n_k^{(c)}} \\
&= f'(\sigma_k^{(c)}) n_j^{(c-1)} \frac{1}{n_j^{(c-1)}} \sum_{i=1}^{l_{c+1}} -\frac{\partial E}{\partial n_i^{(c+1)}} f'(\sigma_i^{(c+1)}) p_{k,i}^{(c+1)}
\end{aligned}$$

$$
\begin{aligned}
&= f'(\sigma_k^{(c)}) \sum_{i=1}^{l_{c+1}} -\frac{\partial E}{\partial n_i^{(c+1)}} f'(\sigma_i^{(c+1)}) n_k^{(c)} \frac{1}{n_k^{(c)}} p_{k,i}^{(c+1)} \\
&= f'(\sigma_k^{(c)}) \sum_{i=1}^{l_{c+1}} \left(-\frac{\partial E}{\partial n_i^{(c+1)}} \frac{\partial n_i^{(c+1)}}{\partial p_{k,i}^{(c+1)}} \frac{1}{n_k^{(c)}} \right) p_{k,i}^{(c+1)} \text{ d'après 9} \\
&= f'(\sigma_k^{(c)}) \sum_{i=1}^{l_{c+1}} e_{ki}^{(c+1)} p_{k,i}^{(c+1)}.
\end{aligned}
$$