Entropie croisée et descente de gradient pour un réseau de neurones multi-classes linéaire

Softmax et entropie croisée

Soit un classifieur linéaire multi-classes constitué d'une couche d'entrée et d'une couche de sortie. Ce réseau prend en entrée un vecteur $\vec{x}$, le multiplie avec une matrice de paramètres W et ajoute un bias $\vec{b}$. Le résultat est un vecteur $\vec{f}$ à C dimensions, où C est le nombre de classes :

$$f = W\vec{x} + \vec{b}$$

C'est ce qu'on appelle le **score** du réseau. Ainsi, le score de la i-ème classe peut être représenté comment suit :

$$f_i = W_i^T \vec{x} + b_i$$

où W_i est la i-ème ligne de la matrice W. Par la suite, le score de chaque classe passe par la fonction **Softmax** :

$$S_i = \frac{e^{f_i}}{\sum_{j=0}^{C-1} e^{f_j}}$$

qui retourne une valeur entre 0 et 1. La sortie S_i peut être vue comme la probabilité conditionnelle de la i-ème classe car les sorties du Softmax somment à 1.

Supposons maintenant que t est l'étiquette cible du vecteur $\vec{x}$ (t est un entier entre 0 et $C - 1$). On peut ainsi calculer la fonction de perte (ici l'entropie croisée) comme suit :

$$L = -\ln(S_t)$$

où S_t est la t-ème sortie du Softmax, c'est-à-dire la probabilité prédite par le réseau pour la classe cible :

$$S_t = \frac{e^{f_t}}{\sum_{j=0}^{C-1} e^{f_j}}.$$

Si la probabilité prédite pour la classe cible est 1, alors la perte sera nulle : $L = -\ln(1) = 0$. À l'inverse, si la probabilité de la classe cible est 0, la perte sera infinie : $L = -\ln(0) = \infty$. Mentionnons également qu'il est fréquent d'ajouter un terme de régularisation L2 sur les poids. Ce faisant, on obtient pour la paire $(\vec{x}, t)$ la perte suivante :

$$L = -\ln(S_t) + \lambda \|W\|^2$$

où λ contrôle la force de la régularisation et aide à réduire le sur-apprentissage. À noter que puisque le biais n'est pas inclus dans W, on réécrira la perte comme :

$$L = -\ln(S_t) + \lambda\Big(\|W\|^2 + \|\vec{b}\|^2\Big).$$

Au final, puisque l'ensemble d'entraînemnt contient N données, la perte sera la moyenne du logarithme des probabilités prédites pour l'ensemble des cibles plus le terme de régularisation :

$$L_{tot} = -\frac{1}{N}\sum_{n=1}^{N} \ln(S_{t_n}) + \lambda\Big(\|W\|^2 + \|\vec{b}\|^2\Big).$$

Gradient de l'entropie croisée

Afin d'opérer une descente du gradient, il faut calculer le gradient de la perte par rapport aux paramètres (et biais) et ce, à l'aide d'une opération de **dérivée en chaîne** du type $\frac{\partial p}{\partial t} = \frac{\partial p}{\partial q}\frac{\partial q}{\partial t}$. Pour ce faire, on peut réécrire la fonction de perte comme suit : $L = L_{pred} + L_{reg}$, où L_{pred} est l'entropie croisée et L_{reg} est la norme L2 du réseau. Ainsi, nous obtenons les dérivées partielles suivantes :

$$\frac{\partial L}{\partial W_i} = \frac{\partial L_{pred}}{\partial W_i} + \frac{\partial L_{reg}}{\partial W_i} = \frac{\partial L_{pred}}{\partial S_t}\frac{\partial S_t}{\partial f_i}\frac{\partial f_i}{\partial W_i} + \frac{\partial L_{reg}}{\partial W_i},$$

où

$$\frac{\partial L_{reg}}{\partial W_i} = \frac{\lambda \partial\Big(\|W\|^2 + \|\vec{b}\|^2\Big)}{\partial W_i} = 2\lambda W_i$$

et

$$\frac{\partial L_{pred}}{\partial S_t} = \frac{\partial(-\ln S_t)}{\partial S_t} = -\frac{1}{S_t}$$

$$\frac{\partial f_i}{\partial W_i} = \frac{\partial(W_i^T \vec{x} + b_i)}{\partial W_i} = \vec{x}.$$

Pour la dérivée partielle $\frac{\partial S_t}{\partial f_i}$, nous utiliserons la règle en vertu de laquelle la dérivée par rapport à x d'une fonction $f(x) = \frac{g(x)}{h(x)}$ est égale à $f'(x) =$

$\frac{g'(x)h(x)-h'(x)g(x)}{[h(x)]^2}$. Nous devons également considérer deux cas de figure, soit $i = t$ et $i \neq t$.

- Pour $i = t$, on obtient:

$$\begin{aligned}
\frac{\partial S_t}{\partial f_i} &= \frac{\partial\left(\frac{e^{f_t}}{\sum_{j=1}^{C-1} e^{f_j}}\right)}{\partial f_i} \\
&= \frac{(e^{f_t})' \sum_{j=0}^{C-1} e^{f_j} - (\sum_{j=0}^{C-1} e^{f_j})' e^{f_t}}{\left[\sum_{j=0}^{C-1} e^{f_j}\right]^2} \\
&= \frac{e^{f_t} \sum_{j=0}^{C-1} e^{f_j} - e^{f_i} e^{f_t}}{\left[\sum_{j=0}^{C-1} e^{f_j}\right]^2} \\
&= \frac{e^{f_t}}{\sum_{j=0}^{C-1} e^{f_j}} \frac{\sum_{j=0}^{C-1} e^{f_j} - e^{f_i}}{\sum_{j=0}^{C-1} e^{f_j}} \\
&= S_t(1 - S_i).
\end{aligned}$$

- Pour $i \neq t$, on obtient:

$$\begin{aligned}
\frac{\partial S_t}{\partial f_i} &= \frac{\partial\left(\frac{e^{f_t}}{\sum_{j=1}^{C-1} e^{f_j}}\right)}{\partial f_i} \\
&= \frac{(e^{f_t})' \sum_{j=0}^{C-1} e^{f_j} - (\sum_{j=0}^{C-1} e^{f_j})' e^{f_t}}{\left[\sum_{j=0}^{C-1} e^{f_j}\right]^2} \\
&= \frac{0 \sum_{j=0}^{C-1} e^{f_i} - e^{f_i} e^{f_t}}{\left[\sum_{j=0}^{C-1} e^{f_i}\right]^2} \\
&= \frac{-e^{f_i} e^{f_t}}{\left[\sum_{j=0}^{C-1} e^{f_i}\right]^2} \\
&= -S_i S_t.
\end{aligned}$$

En combinant le tout, on obtient au final :

$$\begin{aligned}
\frac{\partial L}{\partial W_i} &= \begin{cases} -\frac{1}{S_t} S_t(1 - S_1)\vec{x} + 2\lambda W_i & \text{si } i = t \\ -\frac{1}{S_t}(-S_i S_t)\vec{x} + 2\lambda W_i & \text{si } i \neq t \end{cases} \\
&= \begin{cases} (S_i - 1)\vec{x} + 2\lambda W_i & \text{si } i = t \\ S_i \vec{x} + 2\lambda W_i & \text{si } i \neq t \end{cases}
\end{aligned}$$

De façon similaire, on peut calculer la dérivée de la perte par rapport au bias en changeant le dernier terme de la dérivée en chaîne. En utilisant

$$\frac{\partial L}{\partial b_i} = \frac{\partial L_{pred}}{\partial S_t}\frac{\partial S_t}{\partial f_i}\frac{\partial f_i}{\partial b_i} + \frac{\partial L_{reg}}{\partial b_i}$$

et

$$\frac{\partial f_i}{\partial b_i} = \frac{\partial (W_i^T \vec{x} + b_i)}{\partial b_i} = 1,$$

on obtient au final :

$$\frac{\partial L}{\partial b_i} = \begin{cases} S_i - 1 + 2\lambda b_i & \text{si } i = t \\ S_i + 2\lambda b_i & \text{si } i \neq t \end{cases}$$

Et finalement, on peut "vectoriser" le gradient en calculant la dérivée de la perte par rapport à la matrice W au complet et par rapport au vecteur $\vec{b}$ au complet. On obtient alors :

$$\frac{\partial L}{\partial W} = \nabla_W L = (\vec{S} - \vec{t})\vec{x}^T + 2\lambda W$$

$$\frac{\partial L}{\partial \vec{b}} = \nabla_{\vec{b}} L = \vec{S} - \vec{t} + 2\lambda \vec{b},$$

où $\vec{t}$ est un vecteur cible en format "one-hot".

Une fois les gradient calculés, on peut enfin lancer la descente du gradient :

$$W = W - \eta \nabla_W L$$

$$\vec{b} = \vec{b} - \eta \nabla_{\vec{b}} L.$$