

TECHNIQUES D'APPRENTISSAGE

MAÎTRISE EN INFORMATIQUE



IFT 712 : Devoir 4

JOUFFROY Emma (19157145) || ADOLPHE Maxime
(19156782) || MOUGIN Cyril (19108350)

M. JODOIN Pierre-Marc & M. THEBERGE Antoine

Décembre 2019

1 Question 1

[1.5 points] :Nous avons vu dans le cours quod l'opération Softmax est représentée par l'équation mathématiques suivante :

$$y_{\vec{w}_i}(\vec{x}) = \frac{e^{a_i}}{\sum_c e^{a_i}}$$

où a_i est la sortie du i-ème neurone de sortie et que la fonction de perte entropie croisée est :

$$E_D(W) = - \sum_{n=1}^N \sum_{k=1}^K t_{kn} \ln(y_{\vec{w}_i}(\vec{x}))$$

où t_{kn} est la cible du n-ième élément de la base de données d'entraînement. Donner l'équation du gradient de la fonction de perte par rapport à a_i ainsi que toutes les étapes mathématiques pour arriver à ce résultat.

Solution :

Soit la cross-entropy définie par :

$$E_D(\vec{w}) = - \sum_{n=1}^N \sum_{k=1}^K t_{kn} \ln\left(\frac{e^{a_i}}{\sum_c e^c}\right)$$

On cherche $\frac{\partial E_D}{\partial a_i}$, soit la valeur de la dérivée partielle de la fonction de perte par rapport à la sortie du ième neurone de la dernière couche (avant passage dans la couche "softmax"). Le calcul de cette opération se fait par dérivée. Pour simplifier les opérations, on pose $y = y_{\vec{w}_i}(\vec{x})$

Soit :

$$\begin{aligned} E_D &= - \sum_{n=1}^N \sum_{k=1}^K t_{kn} \ln(y) \\ y &= \frac{e^{a_i}}{\sum_c e^c} \end{aligned}$$

Alors par dérivation en chaîne :

$$\frac{\partial E_D}{\partial a_i} = \frac{\partial E_D}{\partial y} * \frac{\partial y}{\partial a_i}$$

Dans un premier temps, il est nécessaire de calculer $\frac{\partial E_D}{\partial y}$, le gradient de la fonction de perte par rapport à notre sortie y . Dans notre cas, la loss est une entropie-croisée. On considère donc \vec{t} un "one-hot vector", ce qui signifie que t_k est toujours à 0 sauf pour la bonne classe que l'on souhaite prédire ($t_c = 1$ pour la vérité terrain).

Soit,

$$\frac{\partial E_D}{\partial y} = \frac{\partial [-\ln(y)]}{\partial y} = \frac{-1}{y}$$

Dans un second temps, calculons $\frac{\partial y}{\partial a_i}$:

Pour cette opération, deux cas sont à distinguer :

- $\frac{\partial y}{\partial a_i}$ qui correspond à la dérivée de la sortie i au neurone i où la vérité terrain indique $t_i = 1$.
- $\frac{\partial y}{\partial a_k}$ qui correspond à la dérivée de la sortie i au neurone k où k différent de i (et où la vérité terrain indique $t_k = 0$).

On sait que $\frac{\partial}{\partial x} \left(\frac{g(x)}{f(x)} \right) = \frac{f'(x)f(x) - g(x)f'(x)}{f(x)^2}$.

Ainsi :

$$\frac{\partial y_i}{\partial a_i} = \frac{\partial}{\partial a_i} \left(\frac{e^{a_i}}{\sum_c e^{a_c}} \right)$$

$$\frac{\partial y_i}{\partial a_i} = \frac{e^{a_i} \sum_c e^{a_c} - e^{a_i} e^{a_i}}{(\sum_c e^{a_c})^2}$$

$$\frac{\partial y_i}{\partial a_i} = \frac{e^{a_i} (\sum_c e^{a_c} - e^{a_i})}{(\sum_c e^{a_c})^2}$$

$$\frac{\partial y_i}{\partial a_i} = \frac{e^{a_i}}{(\sum_c e^{a_c})} * (1 - \frac{e^{a_i}}{\sum_c e^{a_c}})$$

$$\frac{\partial y_i}{\partial a_i} = y_i * (1 - y_i)$$

Et

$$\frac{\partial y_i}{\partial a_k} = \frac{\partial}{\partial a_k} \left(\frac{e^{a_i}}{\sum_c e^{a_c}} \right)$$

$$\frac{\partial y_i}{\partial a_k} = \frac{0 * \sum_c e^{a_c} - e^{a_i} e^{a_k}}{(\sum_c e^{a_c})^2}$$

$$\frac{\partial y_i}{\partial a_k} = \frac{-e^{a_i}}{\sum_c e^{a_c}} * \frac{e^{a_k}}{\sum_c e^{a_c}}$$

$$\frac{\partial y_i}{\partial a_j} = y_i * y_k$$

Ainsi, nous obtenons de façon plus générale :

$$\frac{\partial y_i}{\partial a_k} = \begin{cases} \text{Si } i = k, & y_i * (1 - y_i) \\ \text{Sinon} & y_i * y_k \end{cases}$$

Finalement, par la dérivée en chaîne, on obtient :

$$\frac{\partial E_D}{\partial a_i} = \frac{\partial E_D}{\partial y} * \frac{\partial y}{\partial a_i}$$

$$\frac{\partial E_D}{\partial a_i} = \begin{cases} \text{Si } k = i, & -(1 - y_i) \\ \text{Sinon} & -y_k \end{cases}$$

2 Question 2

[1.5 points] : On vous demande de classifier les quatre (4) modèles de véhicules suivantes : (1) voitures de sport, (2) voitures familiales, (3) camionnettes, (4) camions lourds. Pour faire vous disposez pour chaque véhicule des caractéristiques suivantes : (i) longueur, (ii) poids, (iii) accélération 0-100 km/h, (iv) prix, (v) consommation d'essence. Dans ce contexte, et considérant qu'on vous fournit un ensemble d'entraînement, expliquez (a) ce qu'est une distribution de vraisemblance et comment vous pourriez la calculer, (b) ce qu'est la distribution à priori et comment vous pourriez la calculer et (c) dites s'il est raisonnable ou non d'émettre l'hypothèse que la distribution :

$$p(\text{consommation d'essence} | \text{voitures de sport})$$

est gaussienne. Justifiez bien vos réponses.

Solution :

Dans le cas de l'exercice, nous avons à faire à un problème de classification , pour lequel nous avons un ensemble d'entraînement :

$$D = (\vec{x}_1, t_1), \dots, (\vec{x}_n, t_n)$$

où $\vec{x}_n \in R^5$ vecteur de données du n-ème élément, et $t_n \in \{\text{voiture de sport, voiture familiale, camionnette, camion lourd}\}$ étiquette de classe du n-ème élément.

Avec D, on souhaite apprendre une fonction de classification :

$$y_{\vec{w}}(\vec{x}) : R^5 \rightarrow \{\text{voiture de sport, voiture familiale, camionnette, camion lourd}\}$$

capable d'inférer la classe à laquelle appartient (\vec{x}).

La distribution de vraisemblance s'exprime comme étant la probabilité d'avoir une certaine étiquette de classe t sachant un vecteur de données \vec{x} , soit $p(t|\vec{x})$. (Dans le cadre de l'exercice, la probabilité d'avoir un certain type de voiture sachant ses caractéristiques). Afin de calculer cette distribution, on peut émettre l'hypothèse qu'il existe un modèle tel qu'une étiquette de classe $t_n = y_{\vec{w}}(\vec{x}_n) + \varepsilon$, avec ε un bruit supposé gaussien de paramètres μ et σ^2 . Maximiser cette vraisemblance s'exprime alors comme suit :

$$\vec{w} = \underset{\vec{w}}{\operatorname{argmax}} P(T|X, \vec{w}, \sigma^2)$$

En émettant l'hypothèse que X, T et \vec{w} sont iid de distribution gaussienne, on peut démontrer que :

$$\vec{w} = \underset{\vec{w}}{\operatorname{argmin}} \sum_{n=1}^N (\vec{w}^T \vec{x}_n - t_n)^2$$

Ainsi, la meilleure solution de \vec{w} est celle qui va faire en sorte que le gradient de l'erreur E_D soit nul. D'où :

$$\nabla_{\vec{w}} E_D(\vec{w}) = \sum_{n=1}^N (\vec{w}^T \vec{x}_n - t_n) \vec{w}_n^T = 0$$

En isolant \vec{w} , on obtient :

$$\vec{w}_{MV} = (X^T X)^{-1} X^T T$$

avec : $X = \begin{pmatrix} 1 & x_{1,1} & \dots & x_{1,d} \\ 1 & x_{2,1} & \dots & x_{2,d} \\ \dots & \dots & \dots & \dots \\ 1 & x_{N,1} & \dots & x_{N,d} \end{pmatrix}$, $T = \begin{pmatrix} t_1 \\ t_2 \\ \dots \\ t_N \end{pmatrix}$ et MV : Maximum de Vraisemblance.

La distribution à priori s'exprime comme étant la probabilité d'avoir un certain vecteur de données \vec{x} sachant une certaine étiquette de classe t , soit $p(\vec{x}|t)$ (dans le cadre de l'exercice, la probabilité d'avoir certaines caractéristiques sachant le type de voiture considéré). Afin de calculer cette distribution, il est nécessaire d'émettre les mêmes hypothèses que précédemment, à savoir qu'il existe un modèle tel qu'une étiquette de classe $t_n = y_{\vec{w}}(\vec{x}_n) + \varepsilon$, avec ε un bruit supposé gaussien de paramètres μ et σ^2 . Maximiser cet à priori s'exprime alors comme suit :

$$\vec{w} = \underset{\vec{w}}{\operatorname{argmax}} p(\vec{w}, T, \sigma^2)$$

En utilisant la formule de Bayes :

$$\vec{w} = \underset{\vec{w}}{\operatorname{argmax}} \frac{p(T|X, \vec{w}, \sigma^2)p(\vec{w})}{p(X, T, \sigma^2)}$$

Comme $p(X, T, \sigma^2)$ ne dépend pas de \vec{w} :

$$\vec{w} = \underset{\vec{w}}{\operatorname{argmax}} p(T|X, \vec{w}, \sigma^2)p(\vec{w})$$

En émettant l'hypothèse que X, T et \vec{w} sont iid de distribution gaussienne, on peut démontrer que :

$$\vec{w} = \underset{\vec{w}}{\operatorname{argmin}} \sum_{n=1}^N (\vec{w}^T \vec{x}_n - t_n)^2 + \lambda \vec{w}^T \vec{w}$$

avec $\lambda = \frac{\sigma^2}{\alpha^2}$. De même que précédemment, trouver le meilleure \vec{w} revient à forcer le gradient de E_D à zéro. Il est également possible de démontrer qu'ainsi, :

$$W_{MAP} = (X^T X + \lambda I)^{-1} X^T T$$

Finalement, émettre l'hypothèse que $p(\text{consommation d'essence} \mid \text{voiture de sport})$ soit gaussienne paraît pertinent. Effectivement, pour une classe de voiture, on pourrait obtenir une valeur moyenne de consommation d'essence. Pour autant, cette dernière dépendra forcément de différents facteurs, comme de l'endroit où a roulé la voiture ou encore du type de conduite du conducteur. Il est donc pertinent d'émettre l'hypothèse que cette distribution suive une loi normale de moyenne μ et d'écart-type σ^2 .

3 Question 3

[1 point] : Démontrer le lien qu'il y a entre la descente de gradient de type momentum, et les formules pour calculer la position, la vitesse et l'accélération d'un objet en mouvement.

Solution :

Pour rappel, le calcul d'un nouveau paramètre avec l'algorithme du momentum est effectué de la manière suivante :

$$\begin{cases} v_{t+1} = \rho \cdot v_t + \nabla E(w_t) \\ w_{t+1} = w_t - \eta v_{t+1} \end{cases}$$

Prenons une analogie physique pour mieux comprendre ce que cela signifie. Soit une particule de masse m en déplacement, localisée par sa position $x(t)$, sa vitesse $v(t)$, son accélération $a(t)$ soumis à une force $F(x(t))$. Le principe fondamental de la dynamique ainsi que la relation position/vitesse donne :

$$\begin{cases} \frac{\partial x(t)}{\partial t} = v(t) \\ m.a(t) = m \cdot \frac{\partial v(t)}{\partial t} = F(x(t)) \end{cases}$$

D'après la discréétisation semi-implicite d'Euler :

$$\begin{cases} x_{n+1} = x_n + \Delta t \cdot v_n \\ m \cdot v_{n+1} = m \cdot v_n + \Delta t F(x_n) \end{cases}$$

Notons que nous ne démontrerons pas cette discréétisation ici. Cependant, physiquement, elle peut se comprendre aisément. Après une durée Δt , la position de notre système se déplaçant à une vitesse v_{n+1} se trouve dans la même direction et à une distance proportionnelle à la vitesse (cette approximation tient seulement pour des petits intervalles de temps ! Rq : on considère que la vitesse pendant tout l'intervalle est constante égale à la vitesse d'arrivée). De la même manière, la variation de la quantité de mouvement ($m \cdot v(t)$) pendant une durée Δt entre un instant n et un instant $n + 1$ va correspondre approximativement à la somme des forces multipliée par la durée écoulé (mathématiquement c'est une approximation à l'ordre 1).

On ré-écrit le système :

$$\begin{cases} x_{n+1} = x_n + \Delta t \cdot v_{n+1} \\ v_{n+1} = v_n + \frac{\Delta t}{m} F(x_n) \end{cases}$$

On peut maintenant faire le lien avec l'algorithme du momentum ! En effet si l'on considère que la position de notre particule est en fait la "position" de notre modèle dans l'espace des paramètres : on peut appeler $x_n = \omega_n$. La force qui "déplace" notre modèle (normalement vers notre minimum) correspond à l'opposée de la dérivée de la fonction d'énergie, on peut donc renommer $F(x_n) = \nabla E(\omega_n)$. Le temps écoulé pendant le passage d'une position à une autre correspond au coefficient d'apprentissage (i.e taille des sauts entre deux positions d'énergie), on peut donc renommer $\Delta t = \eta$. On tombe finalement sur le système d'équations :

$$\begin{cases} \omega_{n+1} = \omega_n + \eta \cdot v_{n+1} \\ v_{n+1} = v_n + \frac{\eta}{m} \nabla E(\omega_n) \end{cases}$$

Pour finir, on voit qu'en renommant $\rho = \frac{\eta}{m}$, on retombe sur l'équation du momentum définie plus haut (à une constante près) :

$$\begin{cases} \omega_{n+1} = \omega_n + \eta \cdot v_{n+1} \\ v_{n+1} = \rho v_n + \nabla E(\omega_n) \end{cases}$$

Comme pour le momentum on voit que si ρ tend vers 0 alors on converge vers une "stochastic gradient descent". Cela se comprend physiquement par le fait que si ρ tend vers 0 cela signifie que la masse de l'objet tend elle aussi vers 0. Or si l'objet est de masse quasi-nulle, on peut estimer qu'il serait bien plus aisé pour la force F de modifier la direction de la trajectoire de notre système. Au contraire, quand ρ grandit, cela signifie dans notre modélisation que la masse de l'objet augmente. Or, on comprend aisément qu'il est plus difficile d'altérer la trajectoire d'un objet super massif.

4 Question 4

[6 points] :Programmez des réseaux de neurones à 1 et 2 couches afin d'avoir un classifieur linéaire et un classifieur non linéaire telle qu'illusté au chapitre 7 du cours. Pour ce faire, vous devez planter un entropie croisée (cross-entropy) avec une couche Softmax à la fin du réseau. Le classifieur non linéaire exige l'implantation d'un réseau avec une couche d'entrée, une couche cachée et une couche de sortie. Le code est contenu dans le fichier `devoir4.zip` via le site web du cours.