Rapport Python3

Ewald Janin

Dernière mise à jour : 24 Avril 2020

Abstract

Même si je n'ai pas été très productif cet après-midi, je crois que je suis assez à l'aise avec les notions que l'on aborde dans ce cours, et j'ai bien compris le principe de ce réseau de neurones.

1 Étude théorique

1.1 Influence du taux d'apprentissage η

La formule de calcul du delta à appliquer sur le vecteur poids pour le mettre à jour pour chacun des neurones est $\Delta W_j = \eta e^{-\frac{||j-j^*||_c^2}{2\sigma^2}}(X - W_j)$.

Dans le cas présent, nous étudions le neurone gagnant, donc $||j - j^*|| = 0$, ce qui nous donne $e^{-\frac{||j-j^*||_c^2}{2\sigma^2}} = 1$ (car $\forall x \in \mathbb{R}, x^0 = 1$). On peut donc ramener la formule de calcul de la mise à jour du poids à $\Delta W_j = \eta(X - W_j)$.

1.1.1 Cas taux d'apprentissage $\eta = 0$

Dans le cas où $\eta = 0$, nous identifions immédiatement que $\Delta W_j = 0$. Il n'y a donc pas de modification appliquée sur le vecteur poids de chacun des neurones, donc leur poids restera celui qui a été déterminé à l'initialisation.

1.1.2 Cas taux d'apprentissage $\eta = 1$

Dans le cas où $\eta = 1$, $\Delta W_j = (X - W_j)$, c'est à dire que le delta de mise à jour pour le poids est égal à la distance euclidienne entre le neurone et l'entrée, donc son nouveau poids est la valeur de l'entrée : $W = W^* + (X - W^*) = X$.

1.1.3 Cas taux d'apprentissage $\eta \in]0, 1[$

Dans ce cas là, la formule permettant de calculer le nouveau poids du noeud gagnant est $W_j = \eta X + (1-\eta)W^*$. Cette fonction a pour représentation mathématique un plan. Plus η se rapproche de 1, plus le nouveau poids du noeud sera proche de la valeur de l'entrée X. À l'inverse, plus η tend vers 0, plus le poids va rester similaire à son ancien poids.
En résumé, si je note W_j le nouveau poids du noeud gagnant :

- $\lim_{\eta \to 0} W_j = W^*$
- $\lim_{\eta \to 1} W_j = X$

Pour les noeuds voisins, nous obtenons le même résultat, qui est à multiplier par la valeur de la fonction de voisinage, dont les valeurs ici appartiennent à $]0, 1[$ pour les noeuds voisins du gagnant. L'influence du taux d'apprentissage η est donc la même concernant la composante qui va avoir le plus d'importance dans le calcul du nouveau poids, à ceci près que le noeud apprend moins que le noeud gagnant.

1.2 Influence de la largeur du voisinage gaussien σ

Nous nous replaçons dans le cas général et non plus seulement sur le noeud gagnant, il nous faut donc bien prendre en compte la fonction de voisinage $V(j, j^*) = e^{-\frac{||j-j^*||_c^2}{2\sigma^2}}$.

1.2.1 Influence de σ sur l'apprentissage des noeuds voisins

Ici, la fonction de voisinage renvoie une valeur comprise dans $]0, 1]$ (car $\lim_{x \to -\infty} e^x = 0$), même si la valeur de 1 n'est retournée que pour le noeud gagnant. Plus les noeuds sont situés loin du noeud gagnant, plus la valeur de cette fonction de voisinage est faible et moins le noeud apprend. L'augmentation du coefficient de voisinage σ va permettre aux noeuds voisins d'apprendre plus, car l'exposant de l'exponentiel aura une valeur plus proche de zéro, et donc la fonction de voisinage sera plus élevée.

1.2.2 Influence de σ sur la densité de l'auto-organisation

Plus le coefficient de voisinage σ est élevé, plus des noeuds plus éloignés du noeud gagnant sont quand même influencés par la valeur de l'entrée du pas courant. Les noeuds vont donc avoir tendance à rester 'proches' les uns des autres avec l'augmentation de σ.

1.2.3 Mesure de l'influence de σ

Pour mesurer l'influence du coefficient de voisinage σ, je propose de faire deux simulations avec le même jeu de données, avec un σ très élevé dans une des simulations, et un σ très bas dans l'autre. Pour mettre ce phénomène le plus en avant possible, nous pourrions utiliser un petit réseau de neurones par rapport aux données, ce qui permettrait de mettre encore plus en évidence la différence de densité du réseau de neurones à convergence.

2 Conclusion