PROJET D'APPRENTISSAGE STATISTIQUE

Generative Adversial Network

Balsan Thibault, Carvaillo Thomas, L'archevêque Valentin

Résumé

Intro ici

Partie ? - Modélisation mathématique

Dans cette partie, nous allons apporter les divers éléments mathématiques de la méthode ; en modélisant les perceptrons multicouches comme des distributions de probabilité, et en explicitant une méthode de construction optimale des perceptrons.

Le perceptron multicouche générateur G sera ici modélisé par la fonction différentiable

$$G_{\theta_g} : \left| \begin{array}{ccc} \mathbb{R}^n & \longrightarrow & E \\ z & \longmapsto & x = G_{\theta_g}(z) \end{array} \right.$$

où $z \sim p(z)$ est le bruit donné en entrée du générateur et E l'espace de l'échantillon d'apprentissage.
Le discriminateur $D_{\theta_d}(x)$ sera représenté par la fonction toute aussi différentiable

$$D_{\theta_D} : \left| \begin{array}{ccc} E & \longrightarrow & [\![0,1]\!] \\ x & \longmapsto & \mathbb{P}\left(x = G_{\theta_g}(z)\right) \end{array} \right.$$

Nous dénoterons par p_{data} la distribution de probabilité de l'échantillon originel, et p_g la distribution de probabilité de l'échantillon généré par G. Le but de ce GAN étant ainsi la convergence de p_{data} vers p_g. Ici, $D_{\theta_d}(x)$ ne retournera pas une valeur binaire, mais un scalaire compris entre 0 et 1, représentant la probabilité que x soit ($D_{\theta_d}(x) = 0$) ou non ($D_{\theta_d}(x) = 1$) généré par le générateur.
Avant de présenter la formalisation du problème, nous allons évoquer un cas élémentaire, en émettant l'hypothèse forte que la distribution p_{data} est connue.

Proposition 1 (Optimalité de D) *Soit G_{θ_g} un générateur fixe, alors le discriminateur optimal $D^*_{\theta_d}(x)$ est définit par*

$$D^*_{\theta_d}(x) = \frac{p_{data(x)}}{p_{data(x)} + p_g(x)}$$

Une démonstration est présentée en annexe.
Comme nous l'avons vu, il s'agit de maximiser la probabilité que D ait raison, i.e. maximiser $D_{\theta_d}(x)$, et, dans un même temps, que G trompe D, i.e. de minimiser $D_{\theta_d}(G_{\theta_g}(z))$. Ceci peut se réécrire sous le problème d'optimisation suivant :

$$\min_{G_{\theta_g}} \left(\max_{D_{\theta_d}} \left(\mathbb{E}_{x \sim p_{data}} \left[log \left(D_{\theta_d}(x) \right) \right] + \mathbb{E}_{z \sim p_z} \left[log \left(1 - D_{\theta_d}(G_{\theta_g}(z)) \right) \right] \right) \right)$$

Ce problème ce résout de manière computationnelle, à l'aide d'une descente de gradient pour le problème de minimisation et d'une ascension de gradient pour celui de maximisation.

Algorithm 1 - Goodfellow et al.

Entrée(s) : $T \in \mathbb{N}$ le nombre d'itérations pour l'apprentissage, K un hyperparamètre

Pour $t \in \{1, \ldots, T\}$ **faire**

  Pour $k \in \{1, \ldots, K\}$ **faire**

    Générer aléatoirement un échantillon de bruit $z^{(1)}, \ldots, z^{(m)}$

    Extraire aléatoirement un échantillon $x^{(1)}, \ldots, x^{(m)}$ de l'échantillon originel

    Mettre à jour le discriminateur par ascension de gradient :

$$\theta_d = \theta_d + \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left(log \left[D_{\theta_d}(x^{(i)}) \right] + log \left[1 - D_{\theta_d}(G_{\theta_g}(z^{(i)})) \right] \right)$$

  Fin du pour

  Générer aléatoirement un échantillon de bruit $z^{(1)}, \ldots, z^{(m)}$

  Mettre à jour le générateur par descente de gradient :

$$\theta_g = \theta_g - \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left(log \left[1 - D_{\theta_d}(G_{\theta_g}(z^{(i)})) \right] \right)$$

Fin du pour