ENSA AL HOCEIMA

المدرسة الوطنية للعلوم التطبيقية بالحسيمة

Filière : Génie civil
A.U. : 2020/2021
Semestre : S2

TD : Optimisation non linéaire avec et sans contraintes

EXERCICE 1

On considère la fonction $f\,(x, y) = 5x^2 + y^2 + 4xy - 2\,x$ et le problème d'optimisation suivant :

$$(P) : Min\, f\,(x, y) \qquad (x, y) \in R^2$$

1. trouver une solution optimale locale de (P).
2. cette solution est-elle globale ?
3. Décrire la méthode de la plus forte pente pour f
4. Etudier la convergence de cette méthode.

EXERCICE 2

On considère le problème de programmation non linéaire suivant :

$$(p)\begin{cases} Minf(x,y) = x^2 + y^2 = 4x - 2y \\ 2x + y \leq 2 \\ -x + 2y \leq 1 \\ x - 2y \geq 0 \\ x \geq 0 \\ -y \leq 0 \end{cases}$$

1. Montrer que le problème (P) admet une solution optimale. Cette solution-est-elle unique ?
2. **i.** Utiliser les conditions de Kuhn-Tucker pour caractériser la solution optimale de (P).

   ii. Calculer la solution de (P).
3. **i.** Donner le problème dual de (P).

   ii. Donner l'expression de la fonction duale (P).

   iii. Résoudre le problème dual.

ENSA AL HOCEIMA
المدرسة الوطنية للعلوم التطبيقية بالحسيمة

Filière : Génie civil
A.U. : 2020/2021
Semestre : S2

TD : Apprentissage du réseau de neurone Artificielle / la rétro propagation de gradient

EXERCICE

Soit un réseau de neurones artificiels de type perceptron multicouche, utilisant la fonction d'activation sigmoïde pour les réseaux de la couche cachée ainsi que pour la couche de sortie, ce réseau est définit par les matrices suivantes :

$w^1 = \begin{pmatrix} 0 & 0 & 0 \\ -0.5 & -0.2 & 0.1 \end{pmatrix}$; $b^1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$; $w^2 = [-0.4, 0.3]$; $b^2 = [-2]$

1. Donner la topologie du réseau correspondant,
2. Pour la fonction sigmoïde $\mathbf{f}(x) = \frac{1}{1+e^{-x}}$, Montrer que :

$$\frac{df(x)}{dx} = f(x)(1 - f(x))$$

3. Calculer les sorties de réseau (couche cachée, couche de sortie) Avec :

$$e = \begin{pmatrix} -1 \\ 2 \\ 5 \end{pmatrix} \text{ Vecteur d'entrée.}$$

$X_3 = -1$

4. En utilisant l'algorithme de rétro propagation de gradient
   i. Pour toutes les unités de sortie, calculer l'erreur : → sortie calculée

   $f'(x) = f(x)(1 - f(x))$ ← $\Delta_k \leftarrow f'(S_k) \times (d_k - a_k)$ → sortie désirée

   ii. Pour toutes les unités cachées j, calculer l'erreur

   $$\Delta_j \leftarrow f'(S_j) \times \sum W_{jk} \Delta_k$$

   iii. Calculer les nouvelles valeurs de poids après une Passe complète de propagation directe.
   La sortie désirée est **d=2** associe à l'entrée **e** le pas d'apprentissage est **$\tau = 0.05$.**