Cours IFT6266,
Pseudo-code pour MLPs

On met en pratique ici les conseils mis de l'avant dans le feuillet "mlp_tricks".

On définit trainMLP(D_{train},D_{valid},λ,h,G,L,t_{max},opt,ω):

1. $D_{train} = (\mathbf{D_{train}}.\mathbf{X}, \mathbf{D_{train}}.\mathbf{Y})$.
2. $(n, d) = dim(\mathbf{X})$, $(n, m) = dim(\mathbf{Y})$
3. On prend D_{train} les 70% premiers exemples de D et D_{valid} le reste.
4. On définit $f(x) = G(b + W \tanh(c + Vx))$ et $\theta = (b, W, c, V)$.
5. On définit $err(x, y) = L(f, (x_i, y_i)) + \lambda||\theta||^2$.
6. On définit $toterr(D) = \frac{1}{|D|} \sum_{(x_i,y_i)\in D} err(x, y)$
7. On définit $g(x, y, \theta) = \frac{\partial err(x,y)}{\partial \theta}$
8. On initialise $b = 0$, $c = 0$, $W = U[-1/\sqrt{h}, 1/\sqrt{h}]$, $V = U[-1/\sqrt{d}, 1/\sqrt{d}]$.
9. $\theta_0 = \theta$
10. $t = 0$. On répète
    (a) faire une itération de l'optimisation: θ = opt(θ,err,g,D_{train},1,t)
    (b) $validerr$ = toterr(D_{valid})
    (c) $t \leftarrow t + 1$
    (d) Si $validerr < besterr$: $besterr = validerr$, $\theta^* = \theta$, $t^* = t$.

    tant que $t < T_{max}$ ou $validerr$ est 5% pire que la meilleure vue ou ne s'est pas amelioree depuis 0.1 T_{max} itérations.
11. si n est petit, ré-optimiser avec toutes les données: θ^* = opt(θ_0, err, g, $D_{train} \cap D_{valid}$, t^*, 0)
12. retourner (θ^*,$besterr$).

L'algorithme d'optimisation le plus simple est la descente de gradient stochastique:

On définit StochGradOpt(θ,err,g,D,T,t,ω)
avec paramètres internes $\omega = (c, \epsilon_0)$:

1. $\tau = t \times |D|$
2. répeter T fois
   (a) boucler sur les n exemples (x_i, y_i) de D
      i. $\epsilon = \epsilon_0/(c\tau + 1)$
      ii. $\theta \leftarrow \theta - \epsilon g(x_i, y_i, \theta)$
      iii. $\tau \leftarrow \tau + 1$
3. retourner θ

Pour sélectionner les hyper-paramètres λ, h, ω on appelle trainMLP avec différentes valeurs de ces hyper-paramètres et on choisit des valeurs pour essayer de minimiser l'erreur de validation $besterr$.

Une fois que cela est fait on peut estimer l'erreur de généralisation sur un ensemble gardé à part = toterr(D_{test}).

Comme heuristique d'optimisation de ces hyper-paramètres, on commence avec $\lambda = 1e{-}6$, $h = \sqrt{n}$, $c = 0$, et on optimise ϵ_0 avec T_{max} petit ($T_{max} = max(1, 10000/n)$). Puis on varie c avec T_{max} plus grand. Finalement on garde c et ϵ_0 fixes et on optimise λ et h.