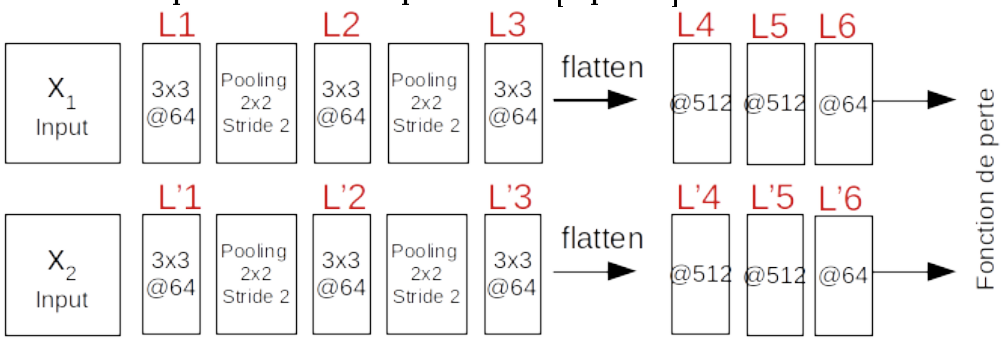
Partiel 13 decembre 2019 : Deep Learning

Jérôme Pasquet

Exercice 1 - Interprétation & Compréhension [7 points]



A chaque itération t ce réseau prend en entrée deux images $x_1^{(t)}$ et $x_2^{(t)}$ de mêmes dimensions (32x32x3) et un label $l^{(t)}$. $x_1^{(t)}$ et $x_2^{(t)}$ sont tirées aléatoirement à partir d'une même base. Les couches L1, L2, L3, L4 et L'1, L'2, L'3, L'4 s'activent via une fonction ReLU, les couches L5 et L'5 via une fonction PReLU. Les couches L6 et L'6 n'ont pas de fonction d'activation (c'est à dire que leur fonction d'activation est telle que $f(x) = x$). Toutes les couches ont un biais. Sauf contre indication explicite sur le schéma, toutes les couches ont un stride de 1 et un padding 'SAME'.
Nous notons $s_{(i)}^{(t)}$ la sortie de la couche i après avoir appliqué une fonction d'activation à l'itération t, $s_{(i)}^{(t)}$ est donc un vecteur ou un tenseur ! Par exemple la sortie de la couche $L'6$ est $s_{6'}^{(t)}$.

La fonction de perte utilisée est la suivante :

$$\mathcal{L} = l^{(t)} max(1 - D^{(t)}, 0) + (1 - l^{(t)}) D^{(t)}$$

Avec :

- $D^{(t)} = 0.5 \times ||s_6^{(t)} - s_{6'}^{(t)}||_2^2$
- Si $x_1^{(t)}$ et $x_2^{(t)}$ ont le même label alors $l^{(t)} = 0$
- Si $x_1^{(t)}$ et $x_2^{(t)}$ n'ont pas le même label alors $l^{(t)} = 1$

Autres notations à utiliser si nécessaire:

- $o_{(i)}^{(t)}$ la sortie de la couche i à l'itération t avant d'appliquer sa fonction d'activation.
- $w_{(ij)}^{(t)}$ le poids reliant le neurone i au neurone j

1. Nous rappelons la régle de mise à jour suivante : $w^{(t+1)} = w^{(t)} - \alpha \frac{\delta \mathcal{L}}{\delta w^{(t)}}$. Développez la mise à jour $\frac{\delta \mathcal{L}}{\delta w_{(56)}^{(t)}}$ pour la couche L6.
2. Développez la mise à jour $\frac{\delta \mathcal{L}}{\delta w_{(45)}^{(t)}}$ pour la couche L5.
3. Calculez le nombre de paramètres dans ce réseau.
4. A terme comment vont évoluer les poids du réseau ? Par exemple les poids de la couche L2 et L'2. Justifiez votre réponse.
5. Expliquez chaque terme de la fonction de perte. Expliquer son utilité.
6. Afin d'éviter le surapprentissage, nous proposons d'utiliser une méthode de dropout sur les couches L5, L'5, L6 et L'6. Déduisez l'effet de cette méthode dans notre scénario.

7. Nous ajoutons une régularisation KL divergente sur la couche L6. Le coefficient de spartité est fixé par l'utilisateur à 0.1. Calculez la mise à jour associée à cette fonction de régularisation : $\frac{\delta \mathcal{L}_{reg}}{\delta w^{(t)}_{(56)}}$.

   Nous rappelons la formule de la KL divergence $D_{KL}(P||Q) = \sum_i P(i)\ log(\frac{P(i)}{Q(i)})$

Exercice 2 - Questions cours & articles [7 points]

1. Expliquer la différence entre l'apprentissage profond et l'apprentissage 'classique' basé sur des algorithmes comme le SVM, Random Forest, etc...
   Un apprentissage via un réseau de neurones avec n couches est-il toujours considéré comme profond ?
2. Expliquez le théorème d'approximation universelle. D'après vous, pourquoi faire plusieurs couches ?
3. Expliquez l'opération de convolution effectuée dans les CNN. Utilisez un schéma détaillé mais ne donnez pas de valeur numérique.
   Donnez la différence avec une convolution classique utilisée en traitement du signal [bonus +0.5].
4. Enumérez et expliquez en deux phrases maximum quatre méthodes utilisées dans les réseaux modernes pour limiter le problème du vanishing gradient.

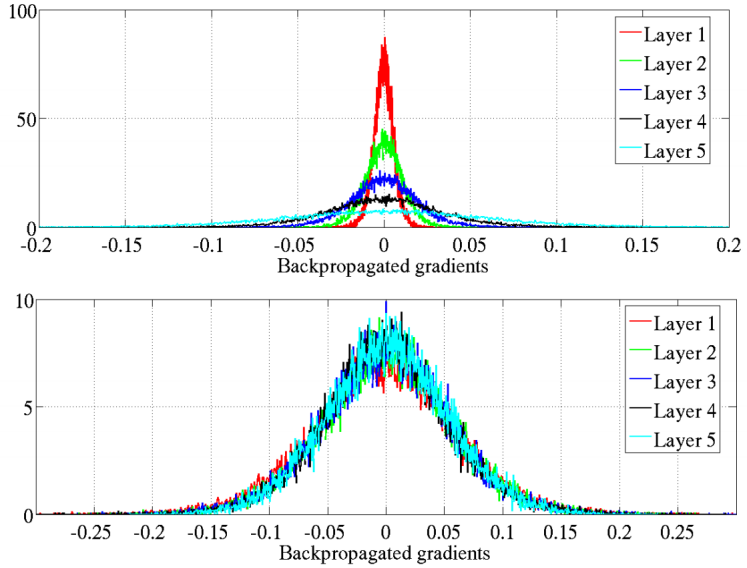


Figure 1: Back-propagated gradients normalized histograms with hyperbolic tangent activation, with standard (top) vs normalized (bottom) initialization.

5. La figure 1 présente deux initialisations différentes. Que peut-on en déduire ? Expliquez en quoi une des deux normalisations est un avantage durant l'apprentissage?

$$\begin{aligned} v_{i+1} &:= 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i} \\ w_{i+1} &:= w_i + v_{i+1} \end{aligned}$$

Figure 2: Considérons cette régle de mise à jour définie dans l'article l'Alexnet.

6. Analysons l'équation présente en figure 2.
   Comment appelle-t-on le coefficient 0.0005? De quelle règle de normalisation s'agit-il ? Pourquoi?
   La formule diffère du cours par l'ajout du facteur : $0.9v_i$. D'après vous quel est le rôle de ce facteur lors de l'apprentissage.

Bonus : Expliquez la différence entre une Variable TensorFlow, un Placeholder, un tenseur (sous Tensorflow) et une variable python. [bonus +1 si tout juste !]