TP reconnaissance des formes

TP 3 : réseau de neurones

Guillaume VAUDAUX RUTH
Adrien CHAN-HON-TONG

définitions

input : $(xin_i)_i$
variables : $(w_{t,i,j})_{t,i,j}$
convention : $x_{0,i} = Xin_i, x_{t,0} = 1$
règles du forward :

- $x_{t+1,i} = relu(\alpha_{t+1,i})$
- $\alpha_{t+1,i} = \sum_j x_{t,j} w_{t,i,j}$
- $l(w) = loss(w) = cout(w) = relu(1 - x_{10,1})$

Attention, le coût et la fonction de non linéarité sont juste des exemples

réseau de neurones

forward

```
A[t][i] = 0
for t
    A[t][0] = 1
for i
    A[0][i] = Xin[i]
for t
    for i
        for j
            A[t][i] += relu(A[t-1][j])*w[t-1][i][j]
```

réseau de neurones

forward

$$x_{t+1,i} = relu\left(\alpha_{t+1,i}\right)$$
$$\alpha_{t+1,i} = \sum_j x_{t,j} w_{t,i,j}$$
$$l\left(w\right) = relu\left(1 - x_{10,1}\right)$$

objectif

On chercher à calculer $\frac{\partial l}{\partial w_{t,i,j}}$

Pas trivial

réseau de neurones

objectif

$$x_{t+1,i} = relu\left(\alpha_{t+1,i}\right)$$
$$\alpha_{t+1,i} = \sum_j x_{t,j} w_{t,i,j}$$
$$l\left(w\right) = relu\left(1 - x_{10,1}\right)$$

On chercher à calculer $\frac{\partial l}{\partial w_{t,i,j}}$

Réduction w - α

$$\frac{\partial l}{\partial w_{t,i,j}} = \frac{\partial l}{\partial \alpha_{t,i,j}} \frac{\partial \alpha_{t,i,j}}{\partial w_{t,i,j}} = \frac{\partial l}{\partial \alpha_{t,i,j}} x_{t,j}$$

Réduction w - α

$\frac{\partial l}{\partial w_{t,i,j}} = \frac{\partial l}{\partial \alpha_{t,i,j}} \frac{\partial \alpha_{t,i,j}}{\partial w_{t,i,j}} = \frac{\partial l}{\partial \alpha_{t,i,j}} x_{t,j}$

Ce n'est vrai **que parce que**, si on note f fonction qui a $\alpha_{t,i,j}$ associe l, g la fonction qui a $w_{t,i,j}$ associe $\alpha_{t,i,j}$, et, h la fonction qui a $w_{t,i,j}$ associe l, on a bien $h = fog$.

réseau de neurones

objectif

$x_{t+1,i} = relu\left(\alpha_{t+1,i}\right)$
$\alpha_{t+1,i} = \sum_j x_{t,j} w_{t,i,j}$
$l\left(w\right) = relu\left(1 - x_{10,1}\right)$
On chercher à calculer $\frac{\partial l}{\partial w_{t,i,j}}$

Réduction α - α

$$\frac{\partial e}{\partial \alpha_{t,j}} = \sum_i \frac{\partial e}{\partial \alpha_{t+1,i}} \frac{\partial \alpha_{t+1,i}}{\partial \alpha_{t,j}} = \sum_i \frac{\partial e}{\partial \alpha_{t+1,i}} w_{t,i,j} relu'\left(\alpha_{t,j}\right)$$

Attention

La somme dans $\frac{\partial e}{\partial \alpha_{t,j}} = \sum_i \frac{\partial e}{\partial \alpha_{t+1,i}} \frac{\partial \alpha_{t+1,i}}{\partial \alpha_{t,j}}$ ne vient **pas** de la somme dans $\alpha_{t+1,i} = \sum_j x_{t,j} w_{t,i,j}$.
Elle vient de $f(u) = a(b(u), c(u))$ implique $\frac{\partial f}{\partial u} = \frac{\partial a}{\partial b}\frac{\partial b}{\partial u} + \frac{\partial a}{\partial c}\frac{\partial c}{\partial u}$.
Lui même vient de $f(u+h) = f(u) + f'(u)h$

réseau de neurones

forward

$$x_{t+1,i} = relu\left(\alpha_{t+1,i}\right)$$
$$\alpha_{t+1,i} = \sum_j x_{t,j} w_{t,i,j}$$
$$l\left(w\right) = s\left(x_{10,1}\right)$$

backward

$$\frac{\partial e}{\partial w_{t,i,j}} = \frac{\partial e}{\partial \alpha_{t+1,i}} \frac{\partial \alpha_{t+1,i}}{\partial w_{t,i,j}} = \frac{\partial e}{\partial \alpha_{t+1,i}} x_{t,j}$$
$$\frac{\partial e}{\partial \alpha_{t,j}} = \sum_i \frac{\partial e}{\partial \alpha_{t+1,i}} \frac{\partial \alpha_{t+1,i}}{\partial \alpha_{t,j}} = \sum_i \frac{\partial e}{\partial \alpha_{t+1,i}} w_{t,i,j} h'\left(\alpha_{t,j}\right)$$

réseau de neurones

forward backward

```
A[t][i] = 0
for t
    A[t][0] = 1
for i
    A[0][i] = xin[i]
for t
    for i
        for j
            A[t][i] += relu(A[t-1][j])*w[t-1][i][j]
DA[t][i] = 0
```

DA[z][1] = $\frac{\partial e}{\alpha_{\mathbf{10},\mathbf{1}}}$

```
for t from 9 to 1
    for j
        for i
            DA[t][j] += DA[t+1][i]*w[t][i][j]*relu'(A[t][j])
```

Attention

Les pseudo codes de ce document ne sont que des pseudo codes : il dispose de tableaux allouables à la volé, suppose un certain nombre de conventions non triviales et ne seront pas directement adapté au TP.

Il convient de comprendre puis adapter et non de recopier puis débugger.