

TP3:

GUILLAUME ARRUDA 1635805

École polytechnique de Montréal

Dans le cadre du cours
INF8225 - Techniques probabilistes et d'apprentissage

19 Février 2016

Partie 1: Pseudocode

Avec $\theta = W + b$ pw_linear(x) =

```

    si x >= 0
        retourne x
    sinon
        retourne a * x

```

pw_linear'(x) =

```

    si x >= 0
        retourne 1
    sinon
        retourne a

```

softmax(x) =

```

    retourne  $\frac{\exp(x)}{\sum(\exp(x_k))}$ 

```

Feedforward(X) =

```

    activation0 = X + Colonne de 1
    Pour chaque couche cachée c
        pré-activationc =  $\theta_c * activation_{(c-1)}$ 
        activationc = pw_linear(pré-activationc) + Colonne de 1
    activationFinale = softmax( $\theta_f * activation_c$ )

```

Backpropagate(X,Y) =

```

    deltafinal = -(Y - activationFinale)
    pour chaque couche cachée c à partir de la fin
        D = pw_linear'(pr - activationc)
        deltac = D .*  $\theta_{(c+1)}$  * delta(c+1)

```

Update() =

```

    pour chaque couche c
         $\theta_c = \theta_c - \text{tauxDapprentissage} * (delta_c * entre_c)$ 

```

LossFunction(Y) =

```

    retourne - ( $\sum Y. * \log(activation_{Finale})$ )

```

TrainNetwork(X,Y, Xvalid, Yvalid) =

```

    Tant que !converged
        mbXs, mbYs = creer_mini_batch(X,Y)
        pour chaque mbX, mbY dans mbXs, mbYs
            Feedforward(mbX)
            BackPropagate(mbX,mbY)
            Update()
        Feedforward(Xvalid)
        loss = LossFunction(Yvalid)
        precision = calcul_precision(YValid)
        converged = loss > dernier_lost

```

Partie 2: Résultats

La première partie des expériences a permis de déterminer le nombre de neurone optimale dans le réseau à une couche et à deux couches. Toutefois, il est rapidement devenu apparent que le nombre de neurone passé un certain point a un effet marginale sur la performance du réseau, mais un effet significatif sur la performance.

Table 1: Performance du réseau de neurone à une couche en fonction du nombre de neurone

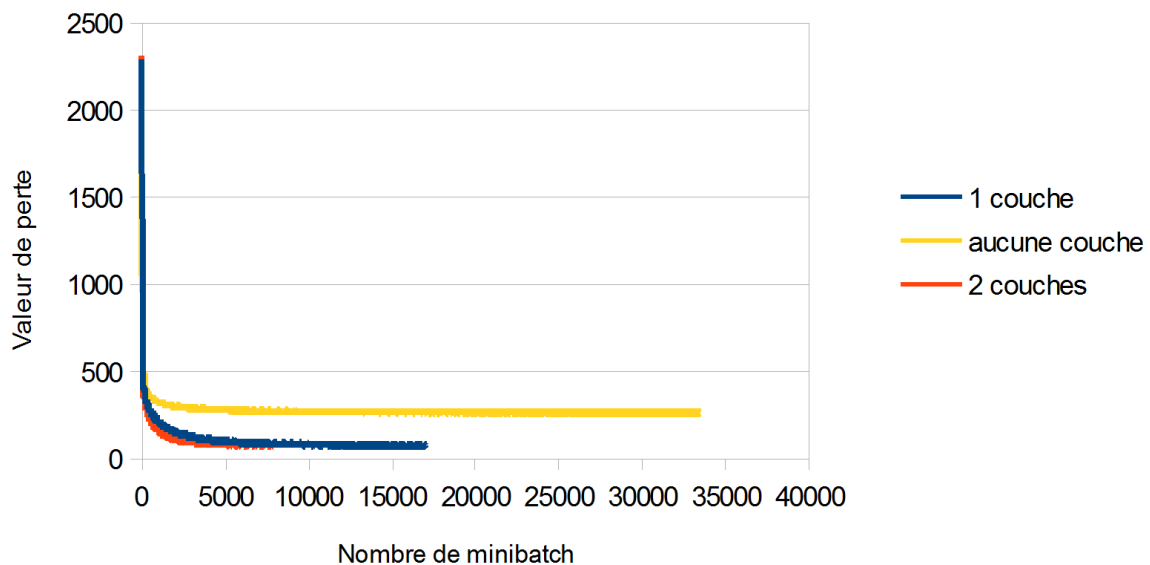
| nombre de neurones | précision(%) | perte moyenne | nombre d'itérations |
|--------------------|--------------|---------------|---------------------|
| 800 | 97.99 | 70.57 | 33 |
| 500 | 97.94 | 70.11 | 33 |
| 300 | 97.85 | 71.92 | 37 |
| 100 | 97.62 | 78.41 | 30 |

Table 2: Performance du réseau de neurone à deux couches en fonction du nombre de neurone

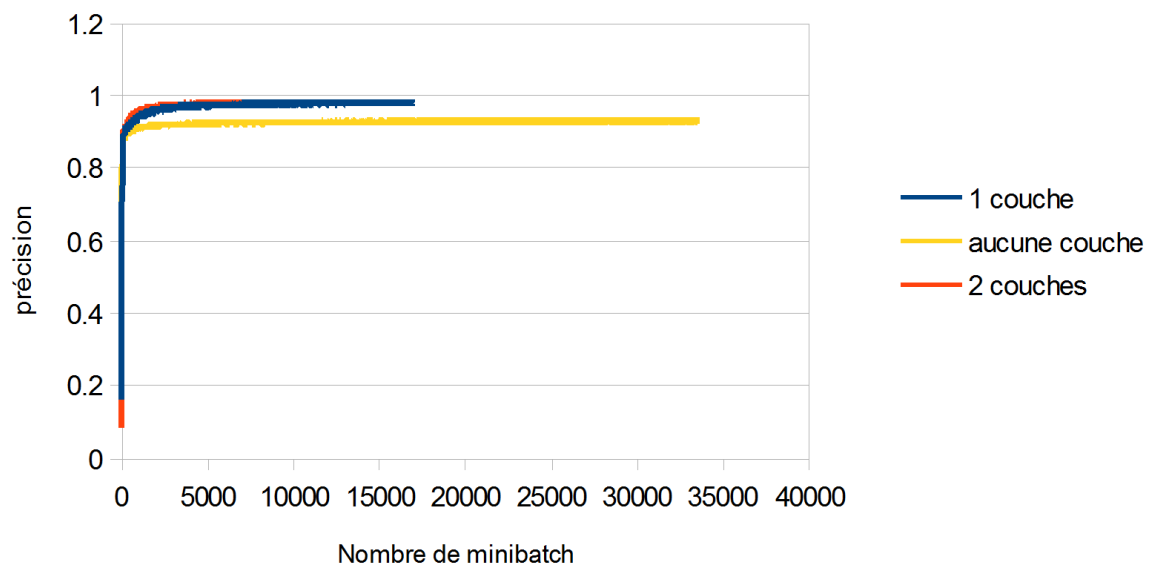
| nombre de neurones | précision(%) | perte moyenne | nombre d'itérations |
|--------------------|--------------|---------------|---------------------|
| 800,500 | 97.97 | 73.59 | 13 |
| 500,300 | 97.88 | 74.51 | 13 |
| 300,100 | 97.79 | 82.04 | 12 |
| 100,10 | 97.11 | 105.69 | 11 |

Pour la création des courbes d'apprentissages suivantes, un réseau à une couche de 500 neurones a été utilisé et un réseau de deux couches de 500, 300 neurones a été utilisé.

Courbe de la fonction de perte en fonction du nombre de minibatch



Courbe de la précision en fonction du nombre de minibatch



Il est possible de voir dans les figures que la performance du réseau à deux couches est très similaire à celle du réseau à une seule couche. Elle a toutefois nécessité beaucoup moins de mini-batch pour atteindre son optimum. Le réseau sans couche cachée est clairement moins performant les deux autres.