Impact of reducing the moving target problem: a experimental investigation

Pieraut Francis, Caporossi Gilles, Bengio Yoshua

Algorithm (stochastic back-propagation)

Initialisation n (learning rate), Θ, e=0, m=0
Faire e=e+1 (epoche)
  do $m \leftarrow m+1$ (all examples)
    x^m =randomly chosen pattern
    propagate inputs values
    compute δ_k related to cost function
    compute $\nabla w_{jk} = \delta_k \cdot y_j$
    compute $\nabla \delta_j ; \delta_j = [\sum_{k=1}^{c} w_{kj} \cdot \delta_k] \cdot g'_y(y_j)$
    compute $\nabla w_{ij} = \delta_j \cdot x_i$
    $w_{ij} \leftarrow w_{ij} - n \cdot \nabla w_{ij}$
    $w_{jk} \leftarrow w_{jk} - n \cdot \nabla w_{jk}$
  until $\|\nabla J\| < \Theta$

Fig. 1. Stochastic back-propagation algorithm

Algorithm (stochastic back-propagation modified)

Initialisation n (learning rate), Θ, e=0, m=0
Faire e=e+1 (epochs)
  do $m \leftarrow m+1$ (all examples)
    x^m =randomly chosen pattern
    for $j = 1$ to h (nb of hidden units)
      propagate inputs values
      compute δ_k related to cost function
      compute $\nabla w_{jk} = \delta_k \cdot y_j$
      compute $\nabla \delta_j ; \delta_j = [\sum_{k=1}^{c} w_{kj} \cdot \delta_k] \cdot g'_y(y_j)$
      compute $\nabla w_{ij} = \delta_j \cdot x_i$
      $w_{ij} \leftarrow w_{ij} - n \cdot \nabla w_{ij}$
      $w_{jk} \leftarrow w_{jk} - n \cdot \nabla w_{jk}$
  until $\|\nabla J\| < \Theta$

Fig. 2. modified stochastic back-propagation algorithm

Abstract— **Ours investigations to understand the reasons why huge neural networks seems to not be able to take advantage of their capacity to get better learning result compare to smaller one bring use to look at optimisation problems related to back-propagation algorithm. One of the solutions we try was to reduce or remove the moving target problem. Experimental results showed us that this approach allows an important speedup of learning.**

Index Terms— **Neural Networks, Moving target problem.**

I. INTRODUCTION

OPtimization process of huge neural networks is extremely inefficient. We believe that the moving target problem, an optimisation problem identified, is a significant cause of this behavior.

A. Moving target problem

In stochastic gradient algorithm(figure 1), gradients for each parameters are compute independently of each other. Cost is not recompute each time we apply a single gradient. So, it is the same as if the target value is changing for each gradient computing. This problem limit of the optimisation process and become more important as the number of parameters increase. One common manifestation of *the moving target problem* is what we call the *herd effect*. Suppose we have 2 separate computational sub-tasks and knowing that units cannot communicate with one another, each unit must decide independently which of the two problem it will tackle. This problem increase as the number of parameters is hight.

B. Reducing the moving target problem

One way to reduce the moving target problem is to optimise some parameters at a time. We try to optimize parameters related to a hidden neuron one by one (figure 2). Instead of doing one forward propagation for each example, we do h (number of hidden units) forward propagation. It is important to notice that in both algorithms (1 and 2 optimise only once each parameters for each example.

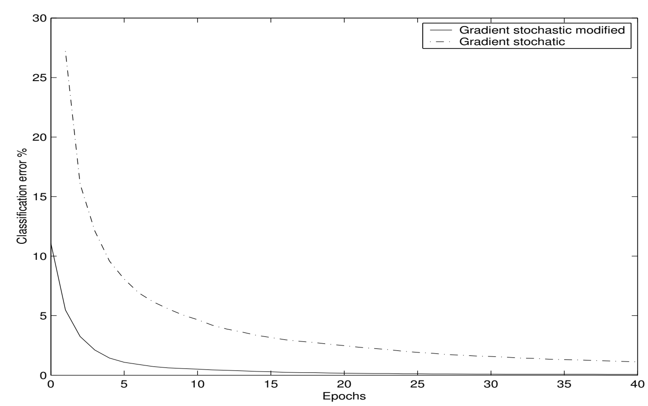

Fig. 3. Compare gradient stochastic vc gradient stochastic modified (Error)

II. EXPERIMENTAL RESULTS

We compare the training process on ”Letters”[1] database with the stochastic back-propagation and the stochastic back-propagation modified. Even is the cost in nearly the same, the error decrease much faster. We use the same number of hidden units (100) and the same learning rate (O.01). The LogSoftMax cost function has been use to train those networks.

[1]Letters is a UCI commun database of characters recognition

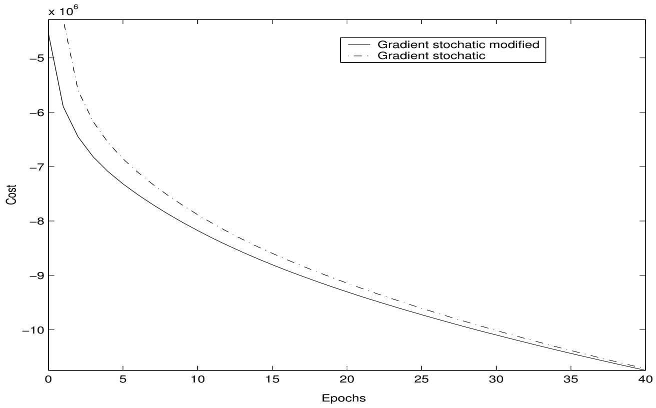

Fig. 4. Compare gradient stochastic vc gradient stochastic modified (Cost)

III. Conclusion

Our experimental investigation show that using a stochastic algorithm that reduce moving target problem speedup learning process of neural network. Ours results show that the moving target problem is a significant optimisation problem.