HW1 of Neural Networks

Yang Guang 1140339072

1. Derive the BP algorithm for MLQPs in both online learning and batch learning ways.

k: the layer of the neural networks.(from 1 to 3)
$v_{kj} = \sum_{i=1}^{N_{k-1}} (u_{kji}x_{k-1,i}^2 + v_{kji}x_{k-1,i}) + b_{kj}$, it is the input of neuron j in the k layer.
$x_{kj} = f(v_{kj})$, it is the output of neuron j in the k layer.
$e_{kj} = d_{kj} - x_{kj}$, it is the error of the output compared to label of neuron in the k layer.

1.1. Online Learning

For online learning, each iteration use a training example, and for the n_{th} iteration, the total instantaneous error energy is $\epsilon(n) = \frac{1}{2}\sum_{j\in C} e_j^2(n)$, here C stands for all the neurons in the output layer.

So the local gradient for output neuron is $\delta_j(n) = e_j(n)f_j'(v_j(n))$.

And for hidden neurons is $\delta_j(n) = f_j'(v_j(n))\sum_K \delta_k(2u_kx_k + v_k)$, here k stands for single successor neuron connected to hidden neuron j and K stands for all successor neurons connected to hidden neuron j.

So we can get the change of weights as the following formula:
$\Delta u_{ji}(n) = a_1\Delta u_{ji}(n-1) + \eta_1\delta_j(n)x_i^2(n)$
$\Delta v_{ji}(n) = a_2\Delta v_{ji}(n-1) + \eta_2\delta_j(n)x_i(n)$.

Thus, we can derive the algorithm as follows:
Step1, Initialize the weights of u and v randomly.
Step2, For each training example n, calculate its x_{kj} of the output neuron as the forward pass, and update u and v using given formula below:
$u(n+1) = u(n) + \Delta u$
$v(n+1) = v(n) + \Delta v$
Step3, Repeat Step2 until convergence.

1.2. Batch Learning

For batch back propagation, the adjustment delta values are accumulated over all training items, to give an aggregate set of deltas, and then the aggregated deltas are applied to each weight and bias.

So we can derive the algorithm for batch as follows:
Step1, Initialize the weights of u and v randomly.
Step2, For each training example n, calculate its x_{kj} of the output neuron as the forward pass, and calculate Δu and Δv for each iteration, then add them to ΔU and ΔV
Step3, modify u and v by:
$u = u + \Delta U$
$v = v + \Delta V$.

2. Program Illustration

There are two Matlab programs, *BP_batch.m* and *BP_online.m*, respectively implements the training and testing procedure of Batch Learning and Online Learning of back propagation algorithm .

3. Performance of the two ways

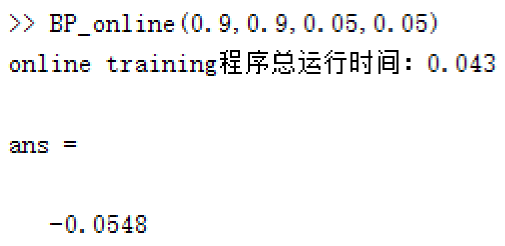
The performance can be viewed at Figure 1 below. The ans parameter is the average error of the forecasted data and the given label.

3.1. Processing Time

The process time of the two methods of back propagation runs similar in this test case.

I think this is because the testing file is not large sufficient to discriminate the difference of running time. Since batch method needs to load all training examples, in theory it will take a longer time to process with these data compared with online method. .

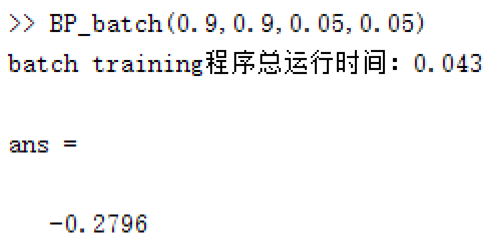
3.2. **Precision**



```
>> BP_online(0.9,0.9,0.05,0.05)
online training程序总运行时间: 0.043

ans =

   -0.0548
```



```
>> BP_batch(0.9,0.9,0.05,0.05)
batch training程序总运行时间: 0.043

ans =

   -0.2796
```

Figure 1: Performance for BP

By compute the average error between the forecasted result and given label, we can see the performance of precision of batch and online BP algorithm.

In the experiment displayed in the figure, We can find the precision of online training fits within 5 percentage and 27 percentage of batch method. .