**Adam Algorithm:**

Adam is an awesome adaptive learning rate optimiser to improve gradient descent

efficiency for our multilayer neural network, with low memory consumption. It is

like a combination of momentum and RMSProp, which uses both first order momentum and second order momentum to help update parameters. The learning rate is no longer fixed as normal SGD, which will become larger when customising infrequently update dimension parameters and smaller when customising frequently update parameters.

There are three main super-parameters for Adam:

: learning rate, which controls update proportion of weights.

: first order moment estimation decay rate ( set 0.9 in our assignment)

: second order moment estimation decay rate ( set 0.999 in our assignment)

: a very small number, just in order to prevent zero denominator.

The implementation of Adam algorithm is as follow:

Set as the th iteration, as the parameters value in t th iteration:

Get original SGD gradients at last iteration:

Update biased first moment estimate:

Update biased second moment estimate:

Then we need to get the bias-corrected:

Now we can update the parameters:

The result turns out to be that using Adam is much faster in convergence.