**2.1 Step forward**

since this is batch processing, x is placed before Wx.

**2.2 Step backward**

db is obtained by summing all the instances of the batch of da.

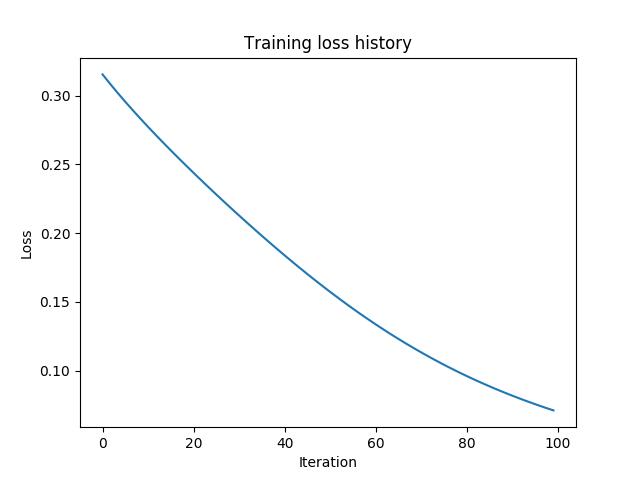
**3.1 Rnn forward**

Rnn forward is equivalent to run step forward function T times. For each time t, the values are calculated as following:

**3.2 Rnn backward**

Here represents the cumulative gradients of h at timestamp t+1. is dh (input of the function) at timestamp t (it is equivalent to ).

dx, dWx, etc., can be calculated by applying which is same as calling step backward function twice. One with and another time with

**5 Rnn loss**

This is equivalent to stack multiple layers together.

The first layer is the RNN layer, the output rnn\_out can be calculated using formula in 3.1.

Rnn\_out is fed into temporal affine. W is of shape [D, A], data is converted to dimension [N, T, A]. (x\*W +b) with some reshaping. Output is temp\_affine\_out.

temp\_affine\_out is then fed into average forward layer. Assume stochastic gradient, mask of size (T), temp\_affine\_out of size (T \* A). The output is calculated as mask \* temp\_affine\_out / (sum(mask)). With batch processing, it can be calculated with a for loop.

The output is then used to calculate softmax loss. ; ;