Word Embedding : Backprop

Let **X** be a mini-batch of sequences of dimensions (**N** ,**T)**

where N : the no. of examples,

T : a T-sized vector of sequences

Note : Every element of T represents an index idx of the Vocabulary V where 0<idx<V

Let **W** be the **Weight matrix** of shape **(V, D)** giving word vectors for all words.

V : to represent vocabulary of V words

D : each word is represented by a D-dimensional vector

---

dout represents the **upstream gradients** of shape (**N,T,D**)

Useful function :

numpy.add.at(a,i,b) : *This function adds **b** into the specified index **idx** of **a** .*

Code snippet:

```
import numpy as np
np.add.at( dW, X, dout)
```

Naive implementation :

```
for i in range(N):
   for j in range(T):
      dW[ X[ i, j ] ] += dout[ i, j, : ]
```