Broadcasting Rules

This is a clarification with more worked out examples. Pulled out from https://github.com/dougc333/Colab-Notebooks/blob/main/broadcast_1.ipynb

Numpy Broadcasting; vector scalar right aligned, matrix batch aligned

- Elements are broadcast or replicated with stride 0 trick. Use stride 0 to make it match the broadcast dimension. When computing the loop iteration pytorc and GPU replace the stride element with stride[0] which means keep on reading the current element. It doesn't allocate new memory for broadcast.
- **Numpy warning. np.dot() doesn't support broadcasting, *, @ do.**
- both pytorch and numpy broadcasting don't allocate memory
- They use a stride trick; instead of a stride of 1 they keep on reading the same element which is a stride of 0.

Element wise Broadcast Rules

1. first left pad the smaller shape then compare each dim
2. either dim ==1 or both equal, (3,1), (1,1) then dimension passes
3. Broadcast fails if neither dim ==1 and they aren't equal to each other.

Example

- a=(2,5),b=(3,). Can we broadcast these?
- (2,5)

> Process the template iterate through the 1-3 rules above

- 1) (1,3) left pad the smaller one to right align
- Iterate through the dims
- 2) Compare 2<->1, this ok bc one of the dims is a 1
- 3) Compare 5<->3, this fails because the dims not equal and neither one is 1.

Example

- a=(2,10),b=(10,).

Can we broadcast these?

- 1) left pad means adding 1 to the second tuple! (2,10), (1,10)
- Iterate through the dims of the padded tuples, (2,10), (1,10)
- 2) compare dim 1: 2<->1, ok one dim ==1
- 3) compare dim2: 10<->10 ok, both equal

Example

- a=(4,3,2), b=(2,)
- left pad (4,3,2), (1,1,2)
- Iterate through the dims
- 4<->1; ok
- 3<->1; ok
- 2<->2; ok
- all tests pass

Example

- a=(6,1,5), b=(3,5)

Use algo first the dims aren't equal so we need left pad a 1 into the smaller none first.

- left pad b;a=(6,1,5), b=(1,3,5)
- Iterate through the dims
- 6<->1; ok, second dim is 1
- 1<->3; ok, first dim is 1
- 5<->5; ok, equal dims

Example

- a = (4,3,2), b= (4,2)

Use the algo, the len(dims) not equal, left pad b since it is smaller.

- left pad b: a = (4,3,2), b= (1, 4,2)
- Iterate through the dims
- 4<->1; ok
- 3<->4; fails
- 2<->2; ok

Rules modification for matmul

1. pytorch and numpy same. Do not reverse the order. A=(2,3,4,5), B=(5,6), this is not the same as A=(5,6) and B=(2,3,4,5)
2. last 2 dims do not broadcast. Left pad smaller first and write 2 dimensions on top of each other.
3. Verify matmul rules (a,b)x(b,c); the b has to match.
   (1,1,3,5)
   (2,3,5,3) match (3,5)=(a,b) and (b,c)=(5,3)
   (1,1,3,4)
   (2,3,5,3). (a,b)=(3,4)_ and (b,c)=(5,3). NO match!
4. Follow earlier element broacasting

Matmul broadcast Example

- A: (3, 4), B: (2, 5, 4, 6)
- Left pad smaller A first, then match matrix dims B:(2, 5, 4, 6) and A(1,1,3,4) alignment; Write 2 matrix aligned vertically
- (1, 1, 3, 4)
- (2, 5, 4, 6) (a,b)x(b,c) where b is 4.
- Iterate through the dims
- 2<->1; ok
- 5<->1; ok
- 3<->4; dont align, verify matrix rules; ok
- 4<->6; dont align, verify matrix rules; ok, repeat matrix rules not necessary
- OK

Matmul broadcast Example

- A: (4, 3, 5, 3, 8) B: (8, 6)
- match matrix dims B:(2, 5, 4, 6) and left pad for right alignment. Write dims on top of each other.
- (4, 3, 5, 3, 8)
- (1, 1, 1, 8, 6)
- Iterate through the dims
- 4<->1; ok

- 3<->1; ok
- 5<->1; ok
- 3<->8; dont align, verify matrix rules; ok
- 8<->6; dont align, verify matrix rules; ok

Matmul broadcast Example

- A: (4,2,3,5) B: (3,2,5,6)
- match matrix dims, no left pad needed
- (4, 2, 3, 5)
- (3, 2, 5, 6)
- Iterate through the dims
- 4<->3; fails
- 2<->2; ok
- 3<->5; dont align, verify matrix rules; ok
- 5<->6; dont align, verify matrix rules; ok