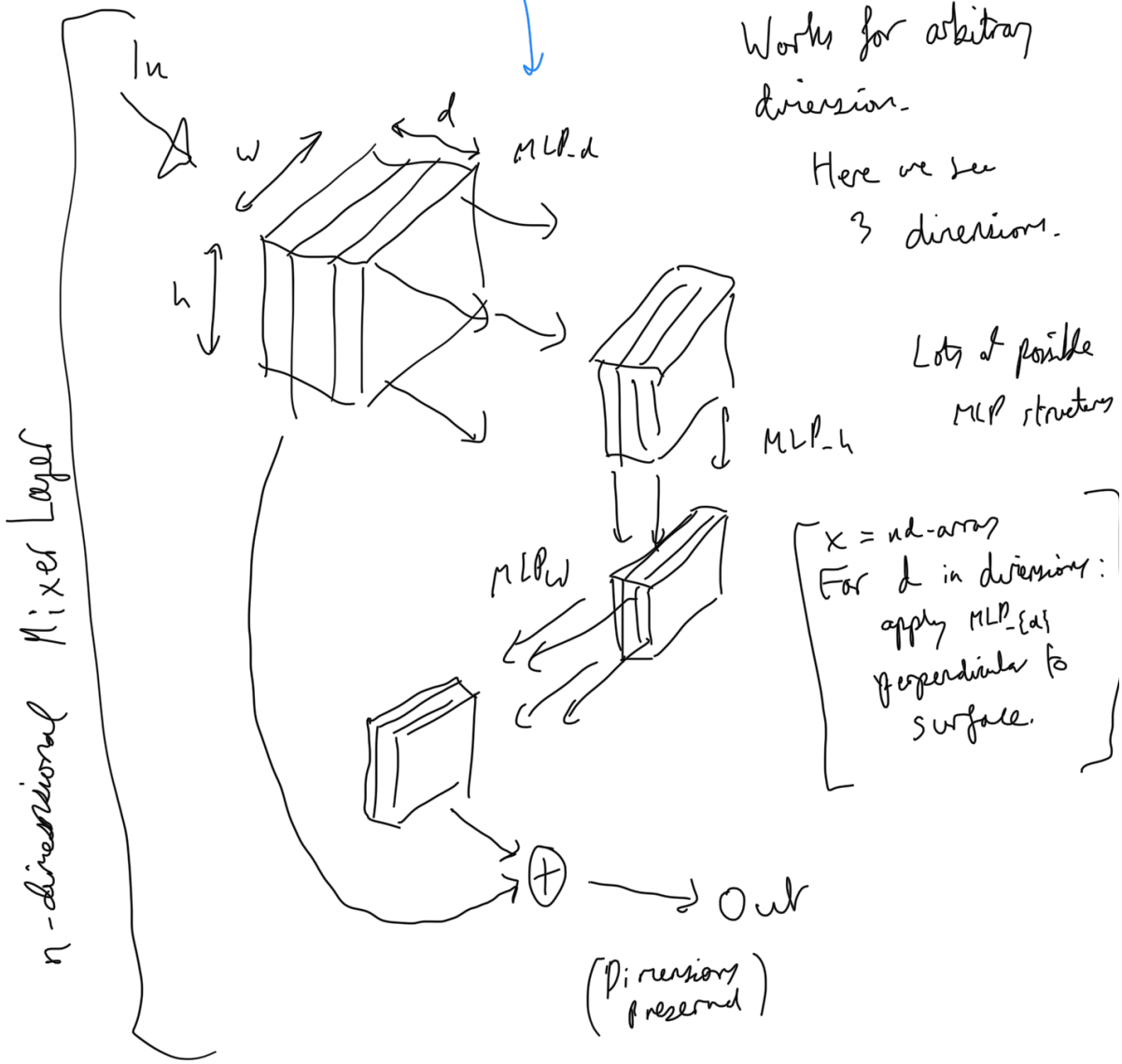
Note, same MLP used for each d slice. Output size = d.

and different MLP for each direction, in each Mixer layer.

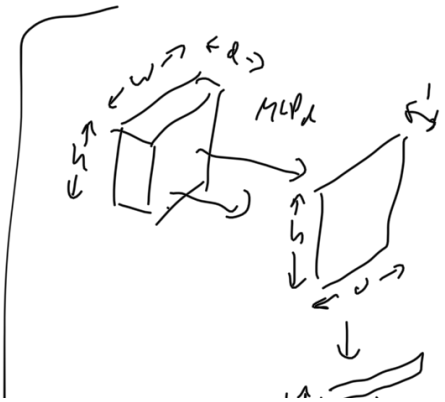


Works for arbitrary dimension.

Here we see 3 dimensions.

Lots of possible MLP structures

[x = nd-array
For d in dimensions:
apply MLP_{d} perpendicular to surface.]



nd Reduction Classifier

(here we see 3 dims)

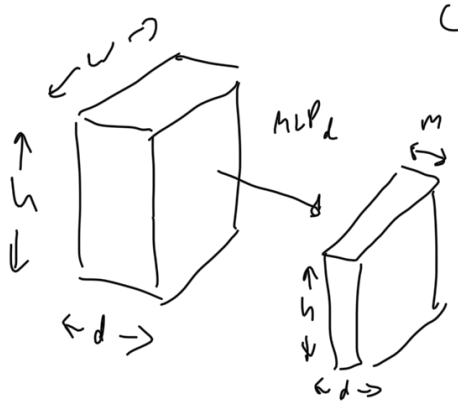
[for d in dimensions:
each MLP_{d}

→ Fully connected → class

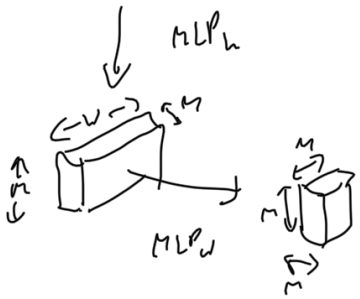
that has an
out size of k.

D-dimensional array

↳ Convert to M-size nd array.



For d in dimensions:
apply MLP{d}
that has out
size of M.



downside is
that size is
M^d where
d is # of
dimensions.

Can then flatten

Fully connected

class

Could also
do this proportional
to size, i.e.
For dimension d,
$\frac{M}{\text{size}(d)} \times k$ would
be the new size.