

Tensors

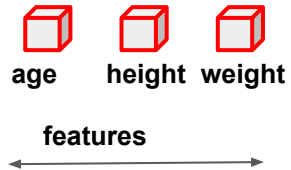
a single value - order zero tensor



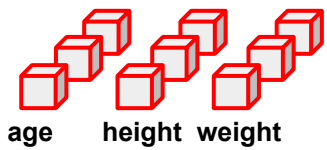


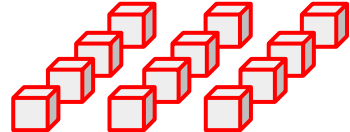


a vector of features - order one tensor



age height weight

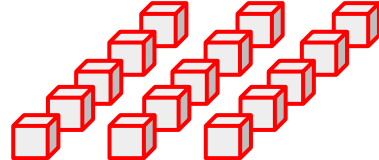




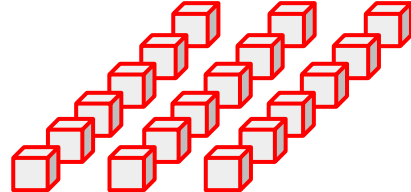
age

height

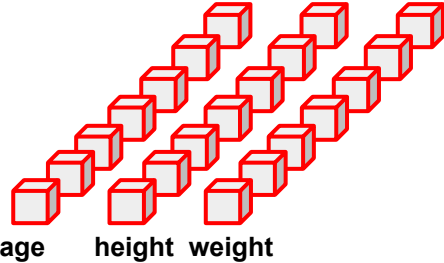
weight

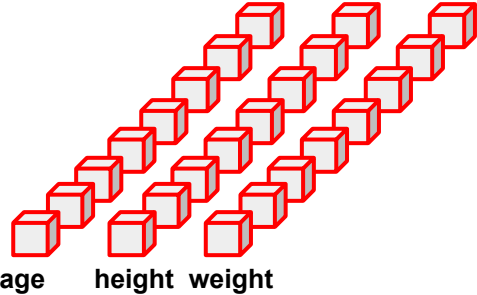


age height weight

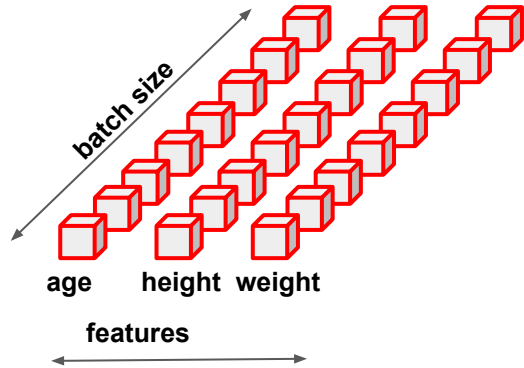


age height weight

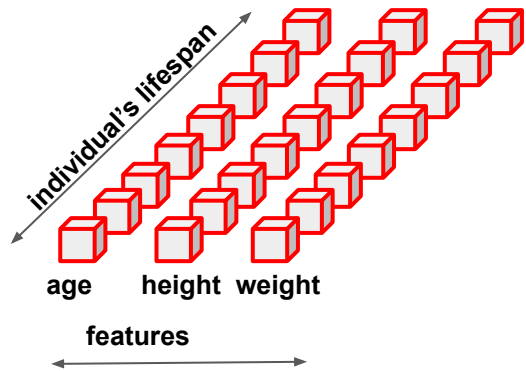




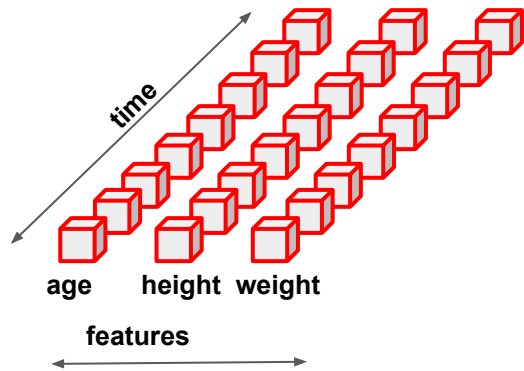
batched vectors - order two tensor



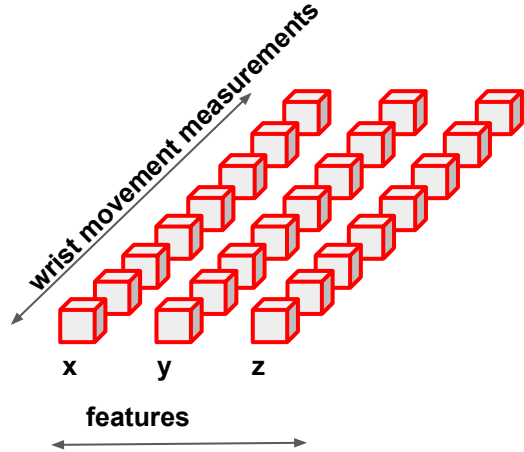
a time series - order two tensor

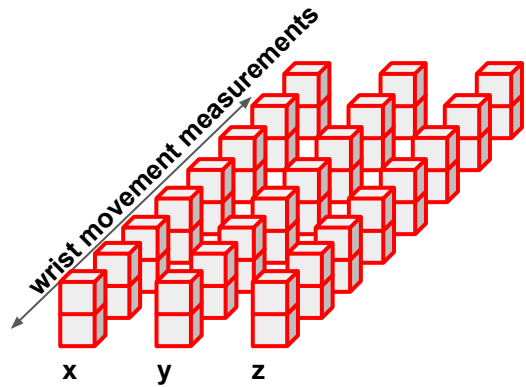


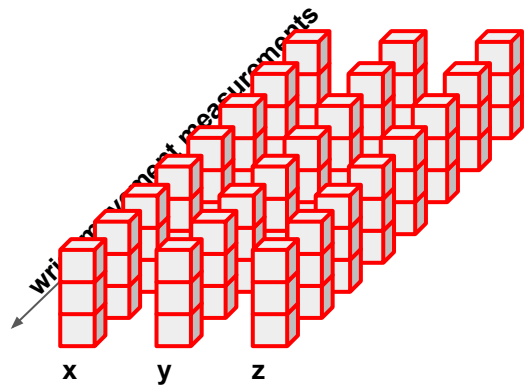
a time series - order two tensor

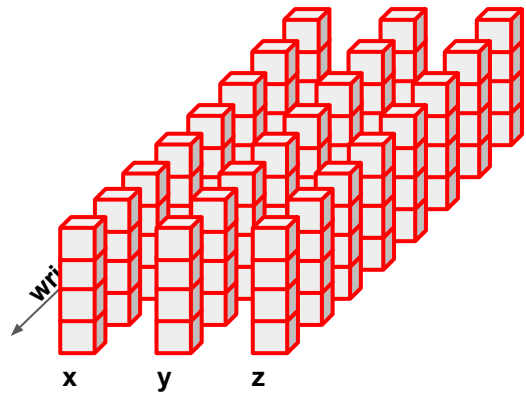


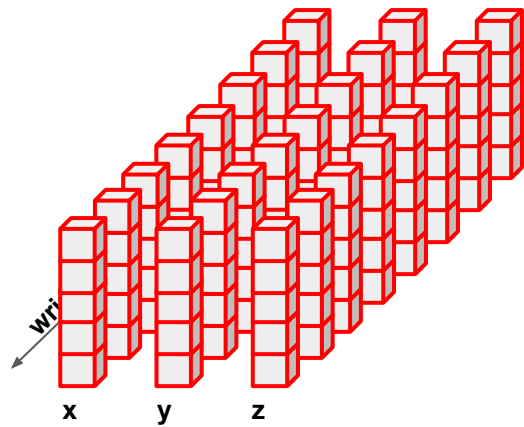
a time series - order two tensor

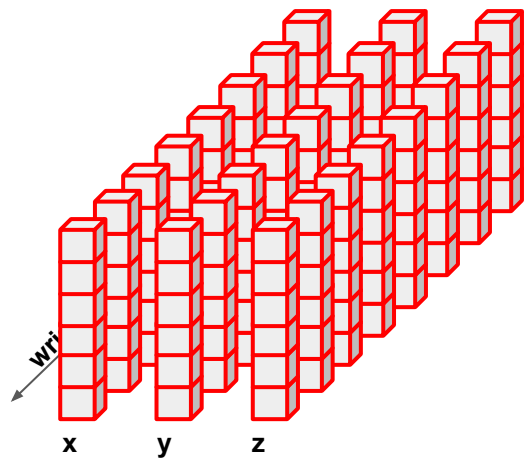




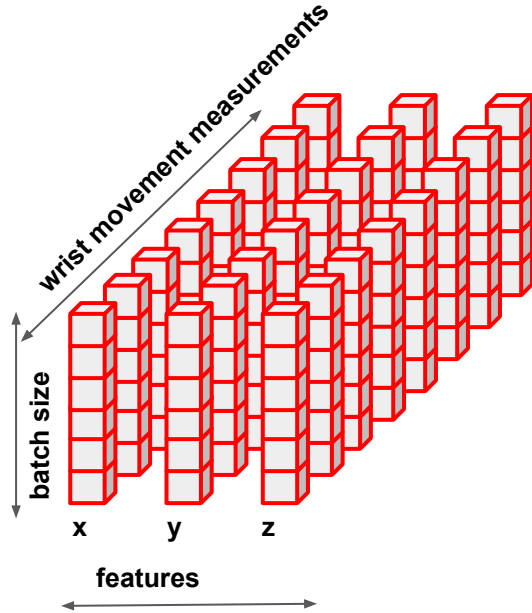




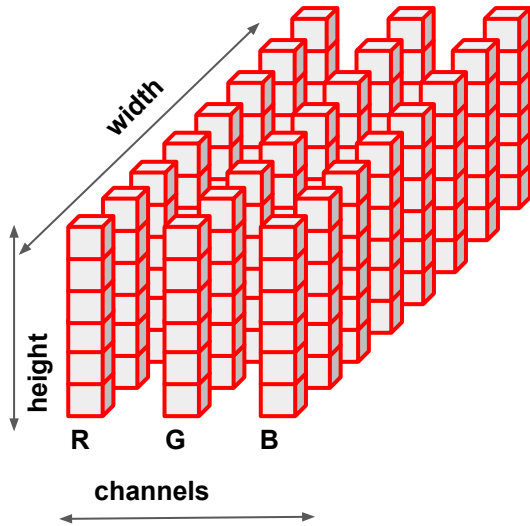


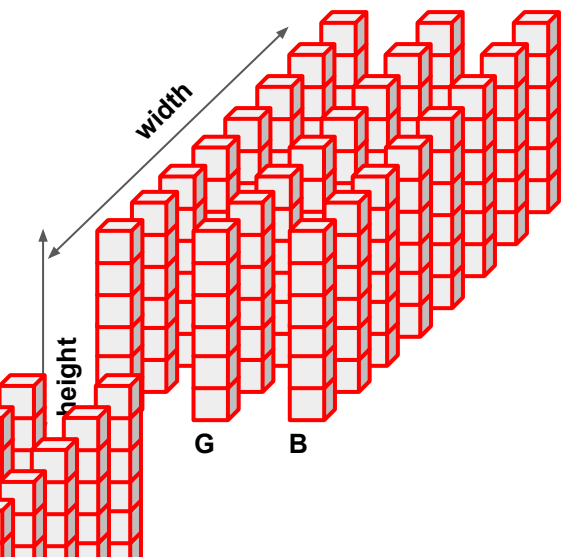


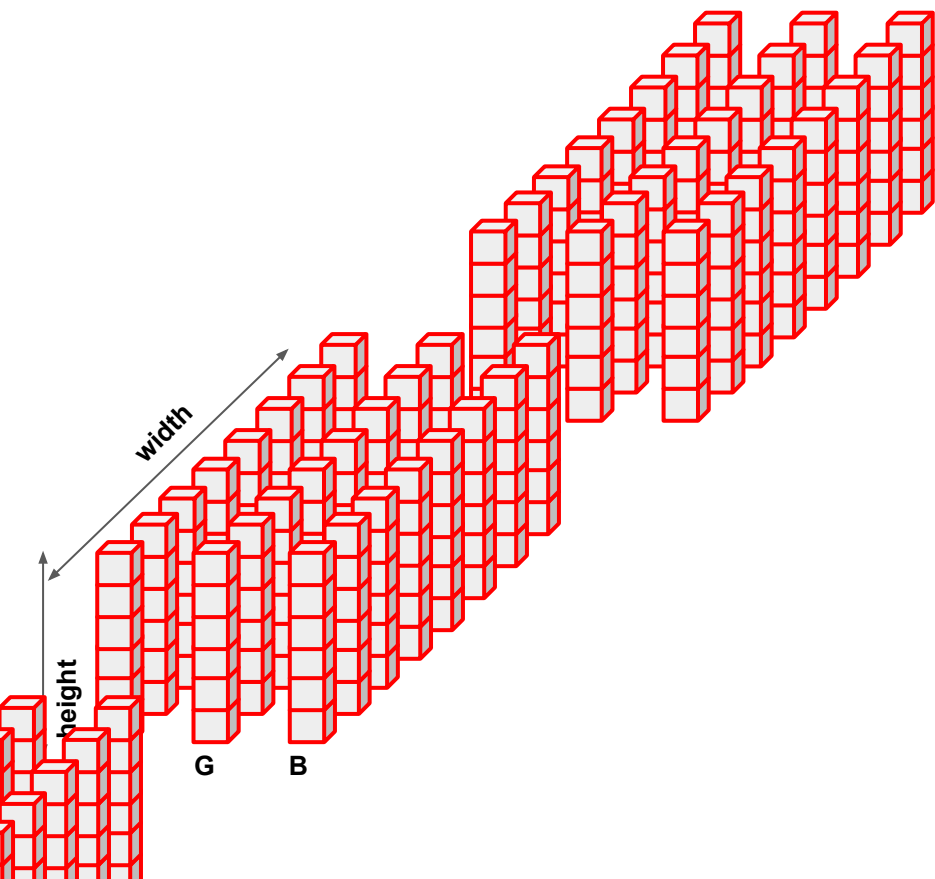
batched time series - order three tensor



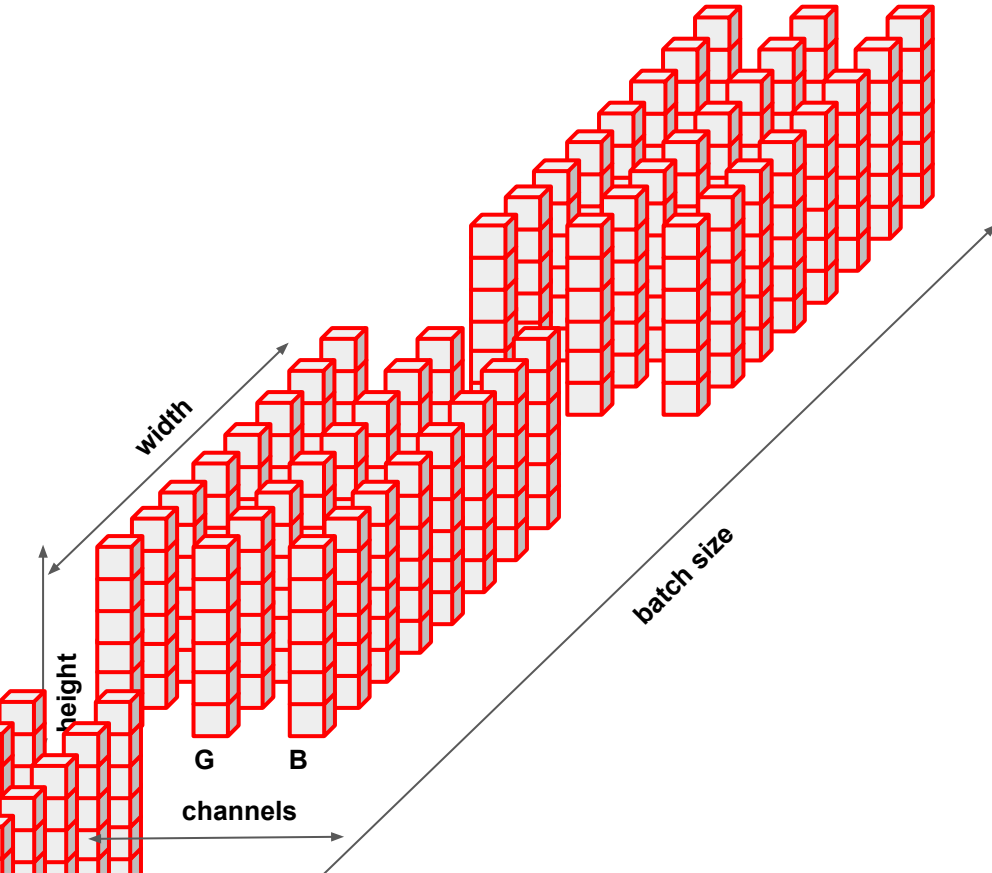
an RGB image - order three tensor







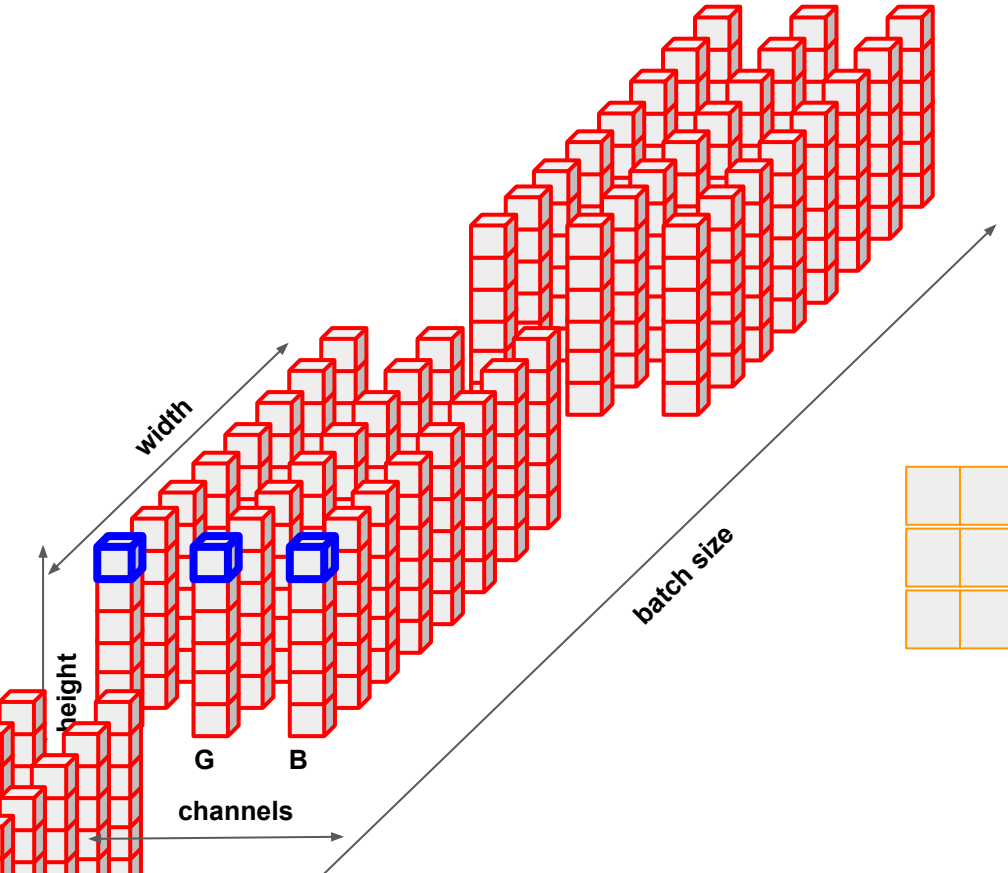
batched images - order four tensor



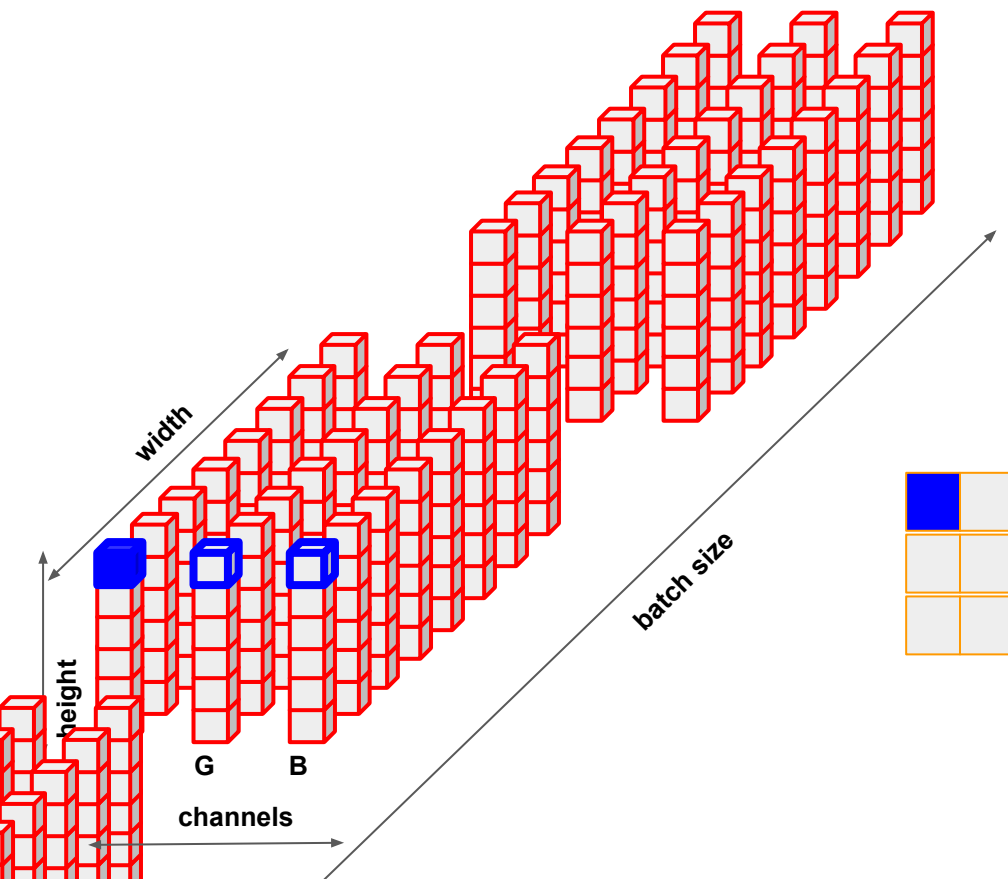
Layers

linear layer

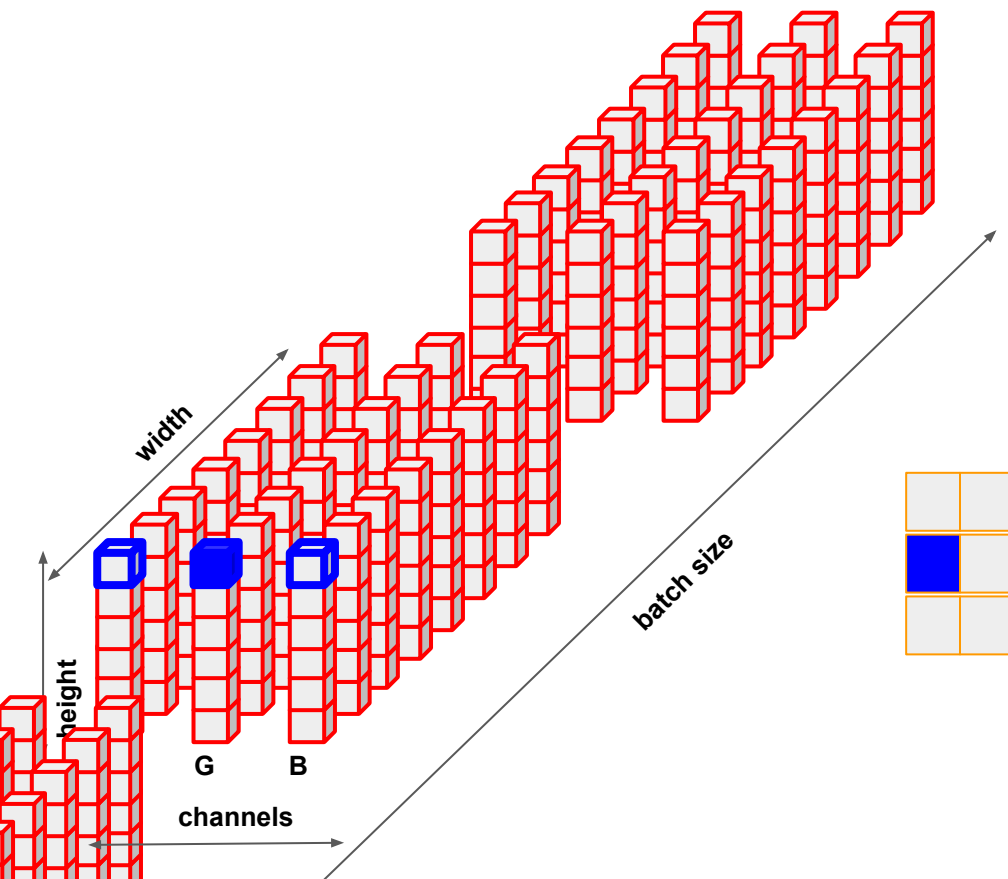
Linear layer is just a matrix multiplication



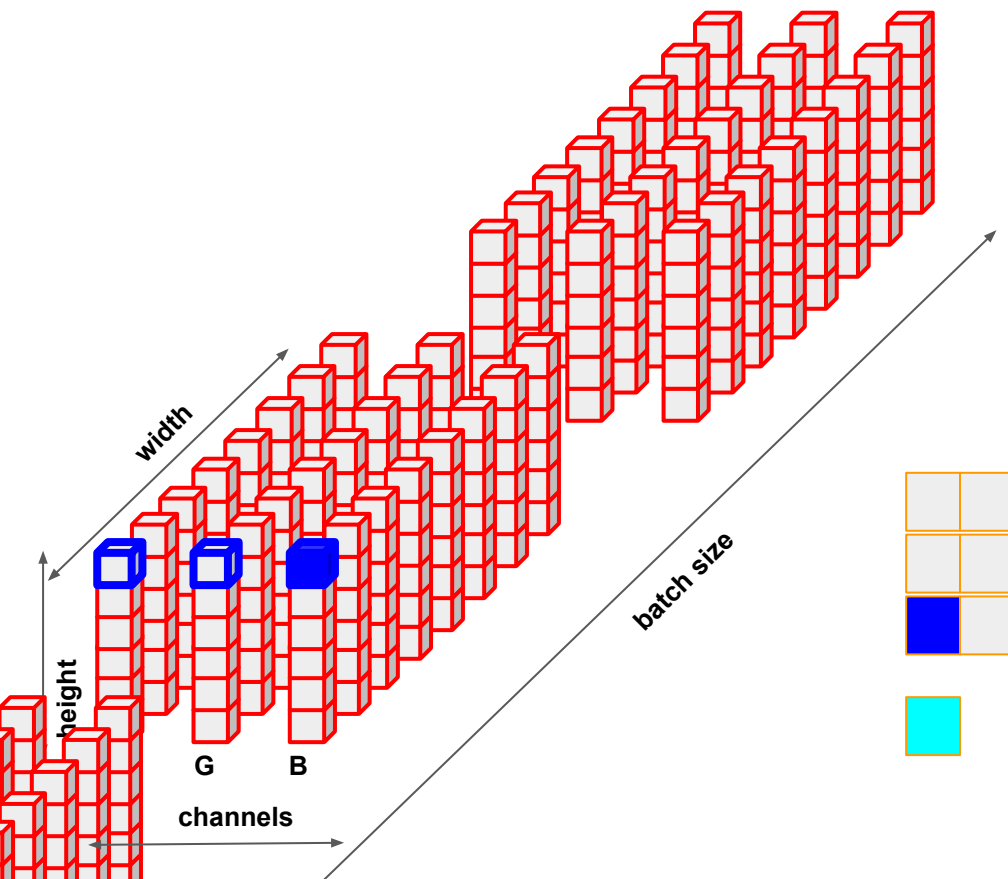
Linear layer is just a matrix multiplication



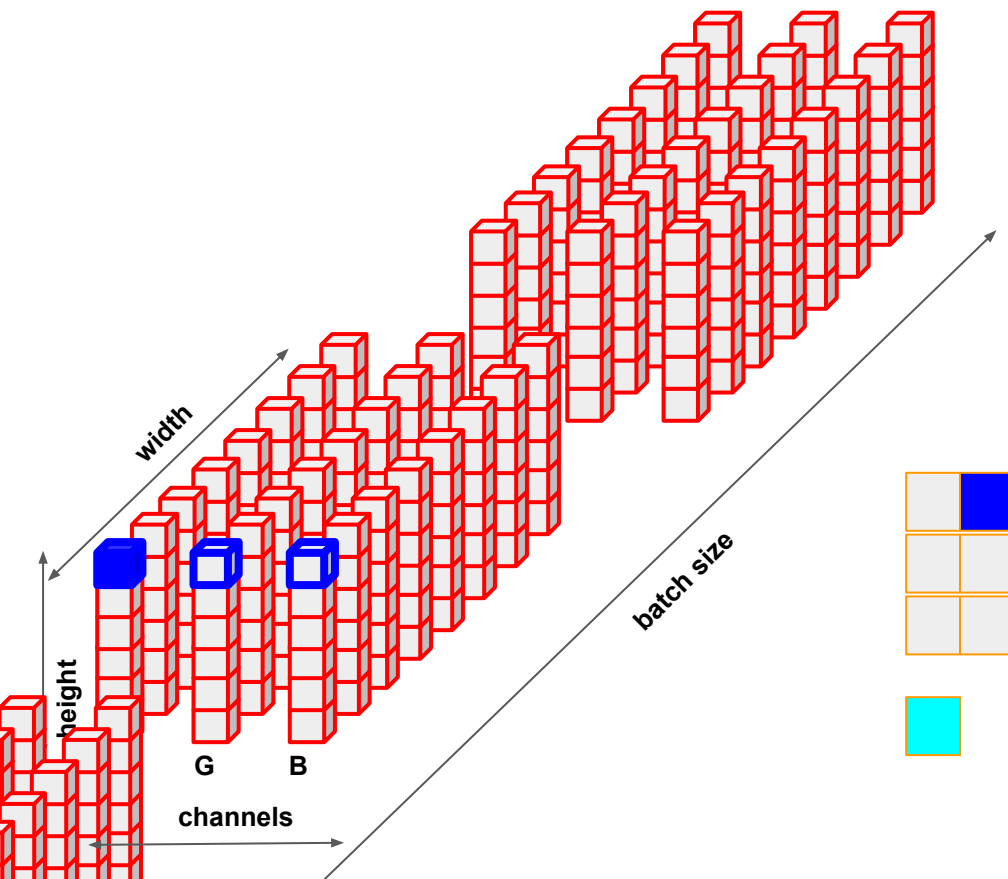
Linear layer is just a matrix multiplication



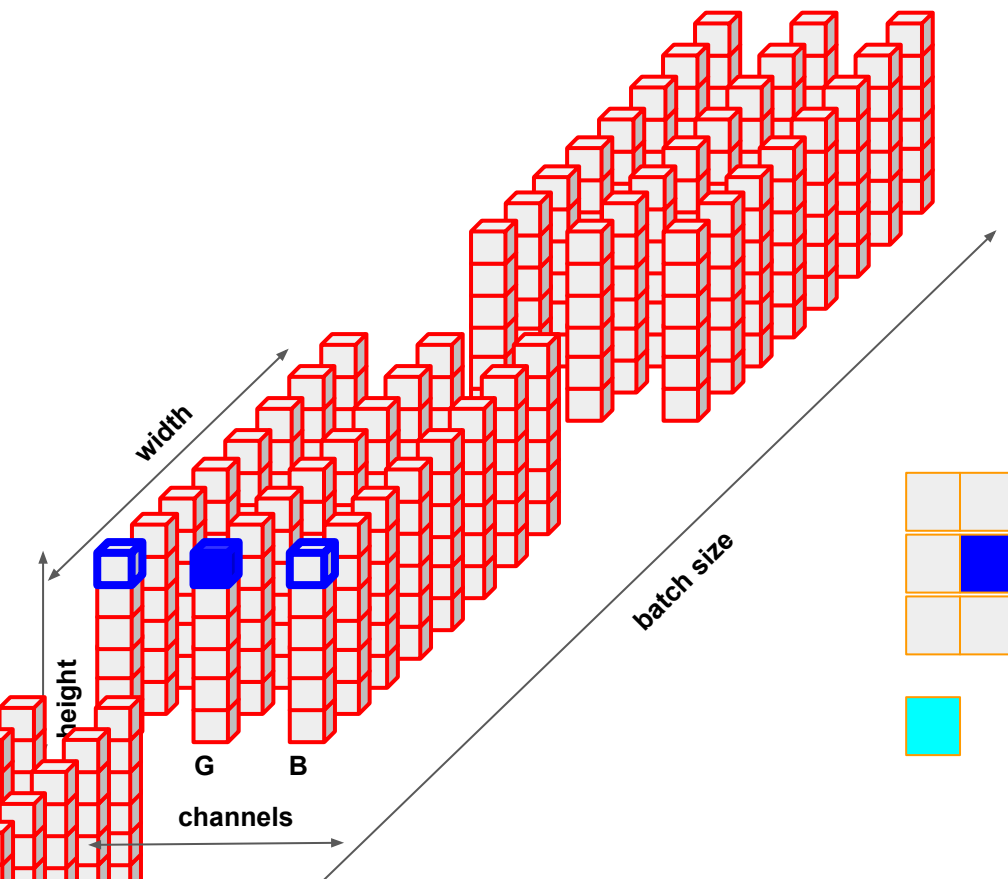
Linear layer is just a matrix multiplication



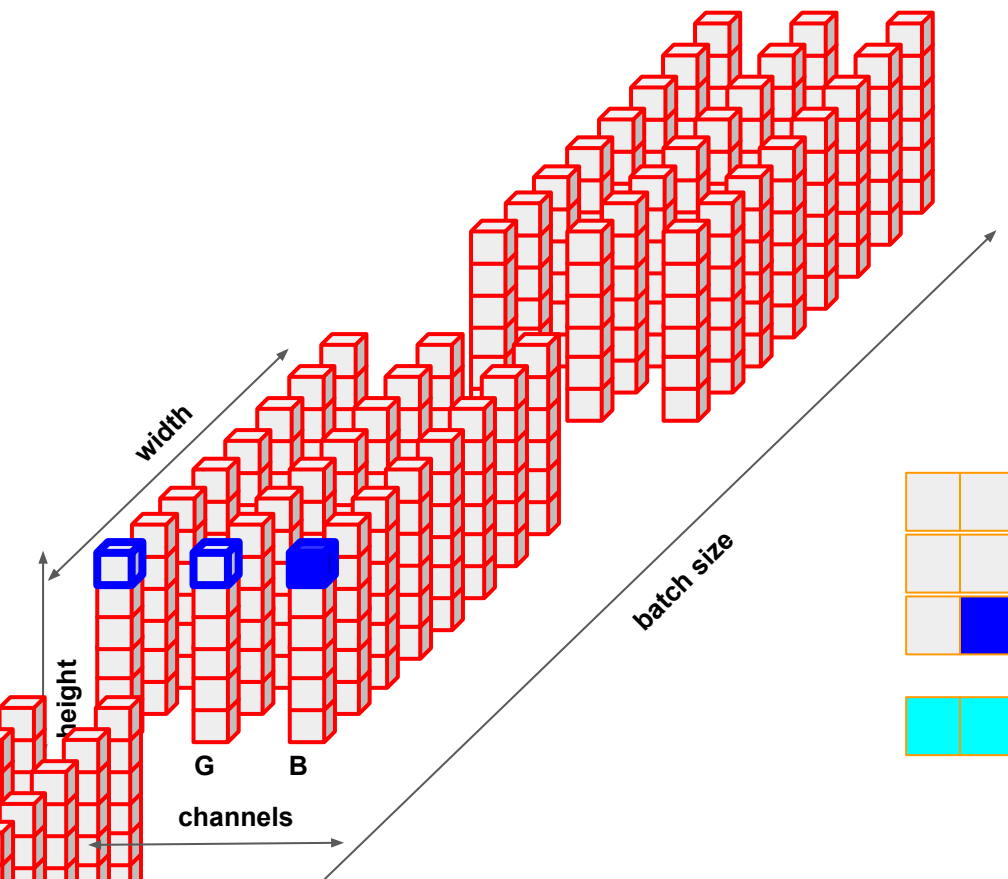
Linear layer is just a matrix multiplication



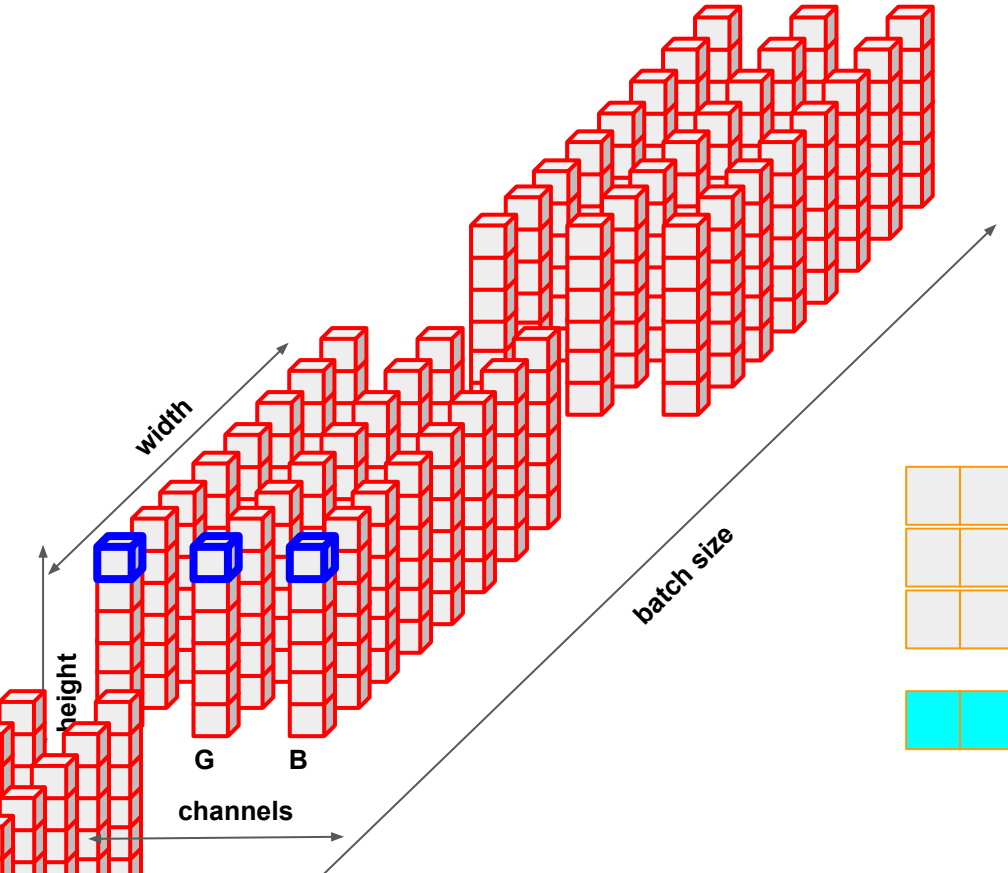
Linear layer is just a matrix multiplication



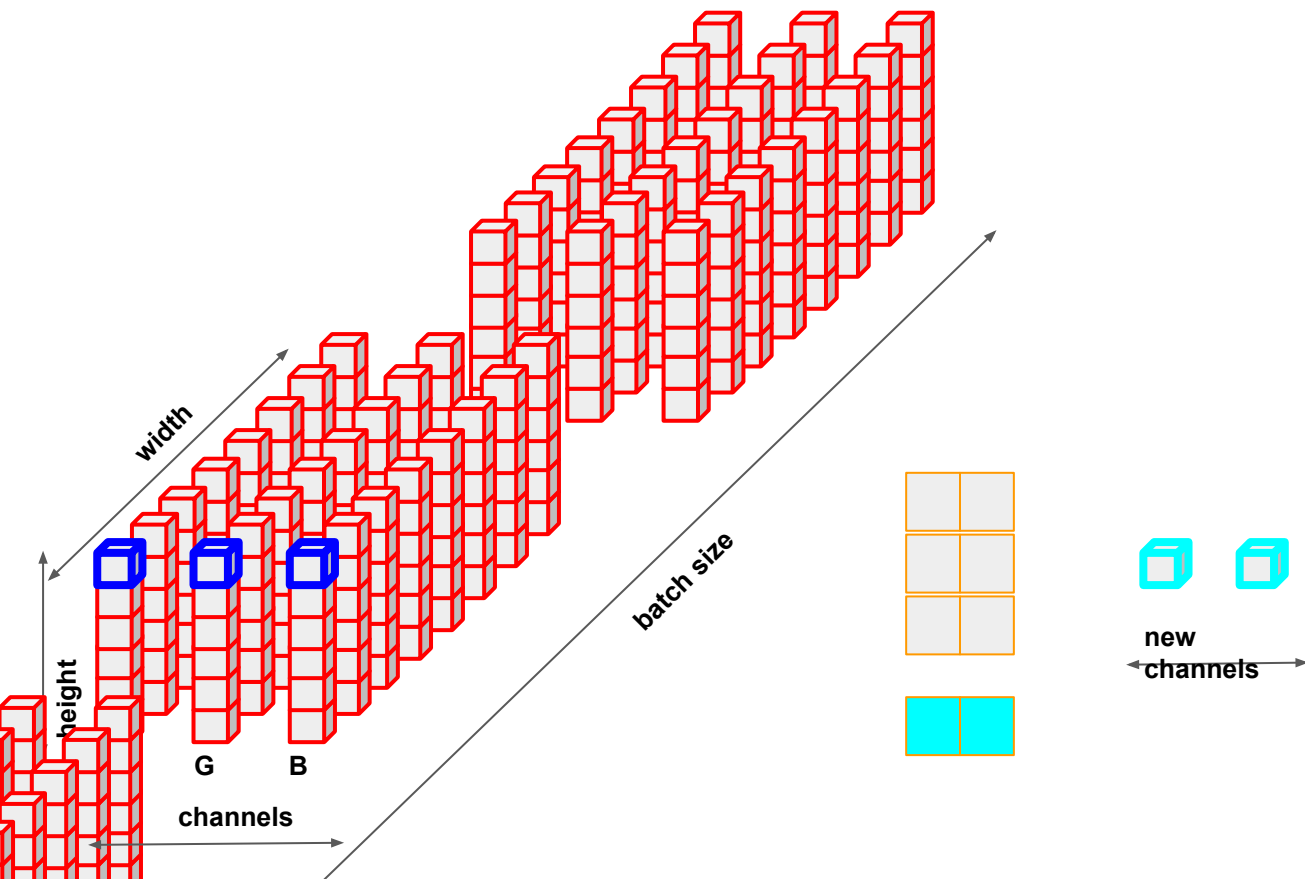
Linear layer is just a matrix multiplication



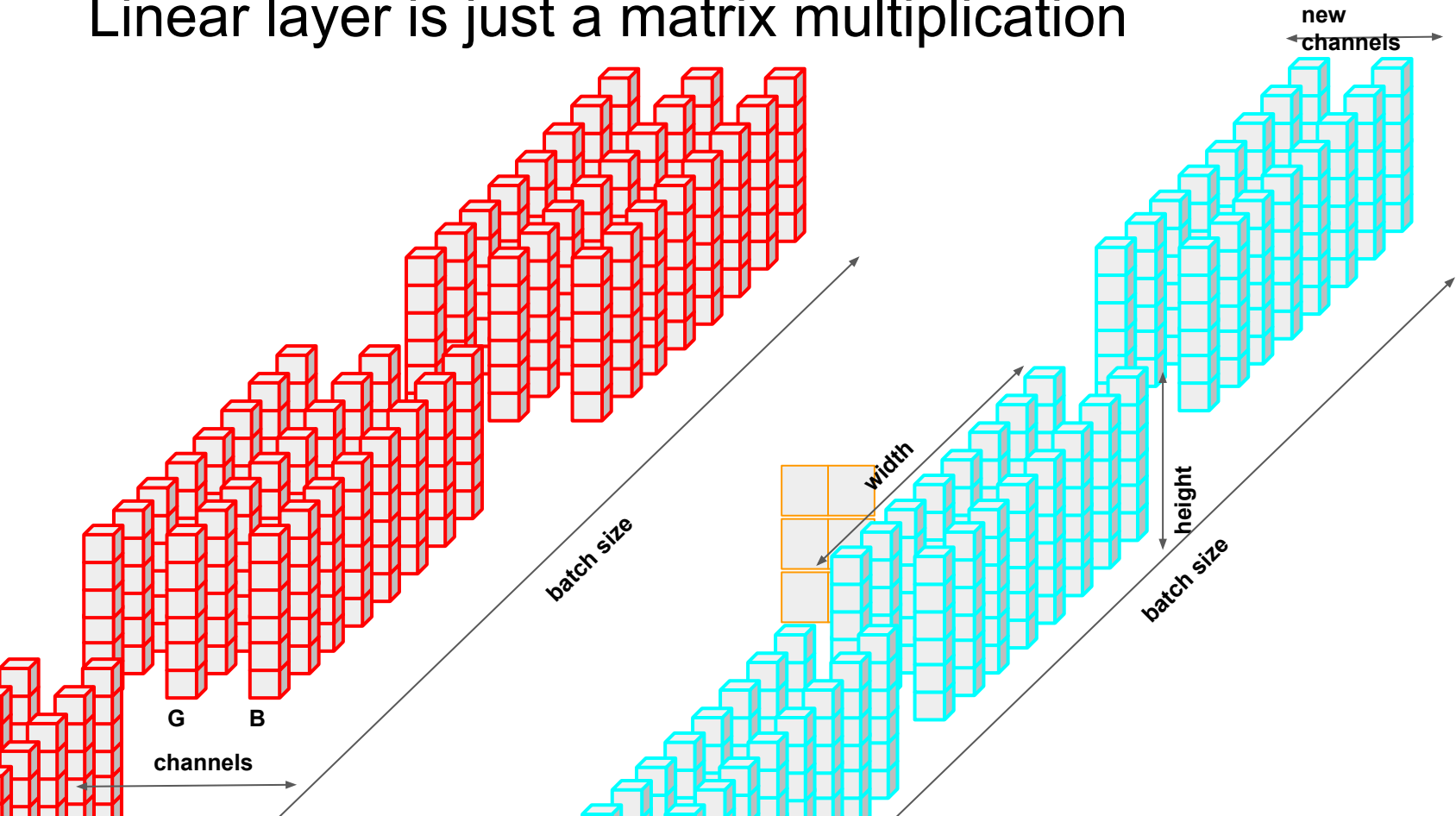
Linear layer is just a matrix multiplication



Linear layer is just a matrix multiplication



Linear layer is just a matrix multiplication



Linear Layer - Takeaways

- Matrix multiplication acts on a last tensor order (features, channels), possibly changing its dimension
- The order structure is preserved, but the dimensionality may be changed for the last (feature/channel) order
- a Pytorch convention to index things from the end with negative indices is sometimes convenient