http://inst.eecs.berkeley.edu/~ee290-2/sp21/

Hardware for Machine Learning

Lecture 18: Training

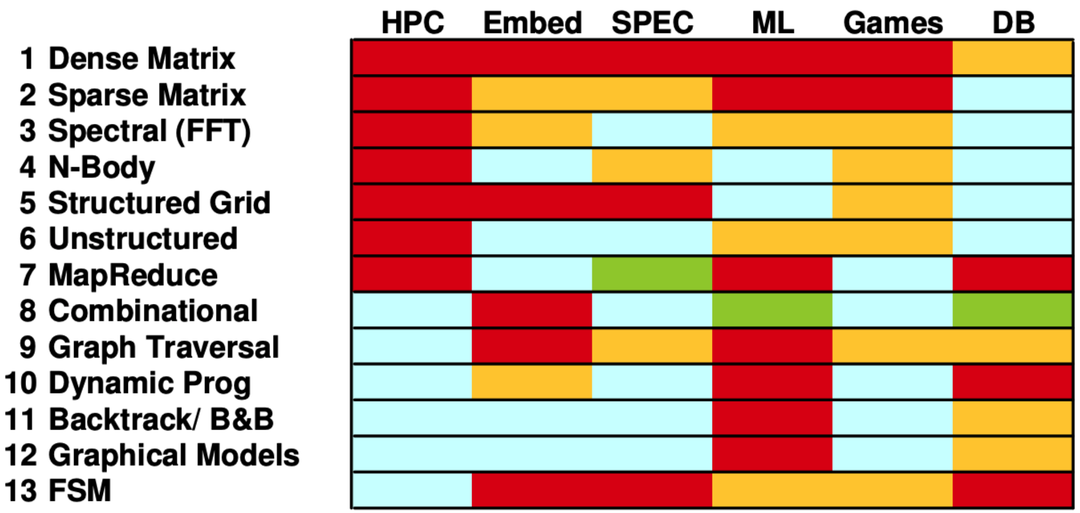
Sophia Shao

The Landscape of Parallel Computing Research: A View from Berkeley

Our goal is to delineate application requirements in a manner that is not overly specific to individual applications or the optimizations used for certain hardware platforms, so that we can draw broader conclusions about hardware requirements. Our approach, described below, is to define a number of "dwarfs", which each capture a pattern of computation and communication common to a class of important applications.

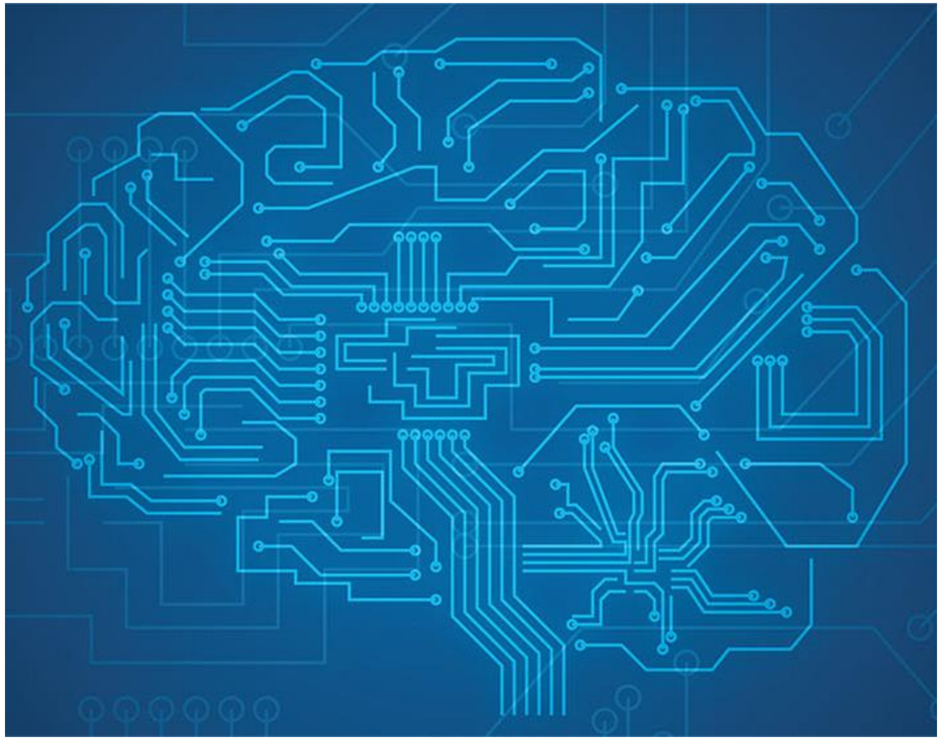
https://www2.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.pdf

The Parallel Computing Laboratory



Review

- Core computation in DNN
- Execution order of the core computation
- Hardware realization of the core computation
- Mapping DNNs to hardware
- Data transfer mechanisms across storage hierarchy
- Sparsity in DNNs
- Codesign example
- Other Operators and Near-Data Processing
  - Element-wise, Embedding, DLRM
  - Case studies of near-data processing for low compute density applications.



Training

- Training Flow Recap
- Key Kernels
  - Forward Propagation
  - Backward Propagation
  - Weight Update

Machine Learning Algorithms

- “A computer program is said to ***learn*** from ***experience (E)*** with respect to some ***task (T)*** and some ***performance measure (P),*** if its performance on T, as measured by P, improves with experience E.”, Tom Mitchell, 1998

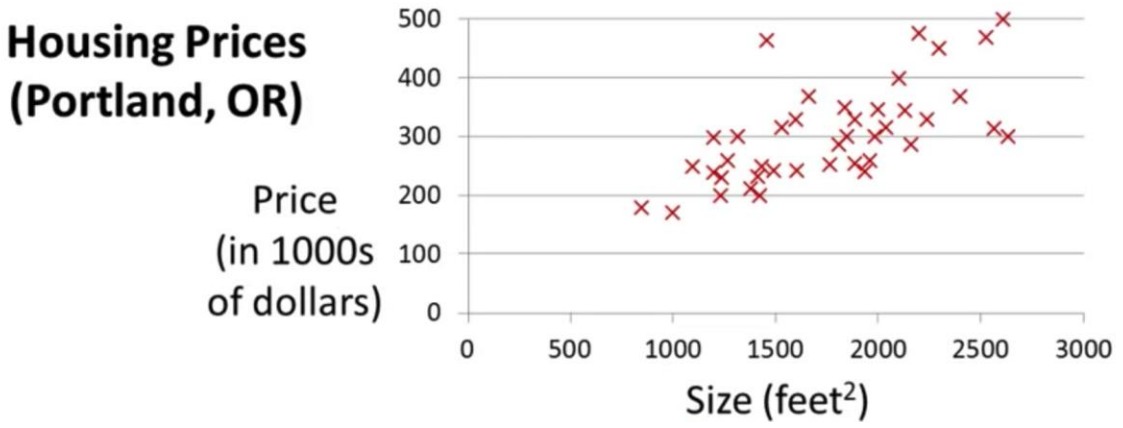
- Example: spam classification
  - Task (T): Predict emails as spam or not spam.
  - Experience (E): Observe users label emails as spam or not.
  - Performance (P): # of emails that are correctly predicted.

Building a ML Algorithm

- Nearly all ML algorithms can be described as particular instances of a simple recipe:
  - A dataset -> Experience (E)
  - A cost (loss) function -> Performance Measure (P)
  - A model + An optimization method -> Task (T)
- Use this recipe to see the different algorithms:
  - As part of a taxonomy of methods for doing related tasks that work for similar reasons
  - Rather than as a long list of algorithms that each have separate justifications

Example: Linear Regression

- Dataset:
  - (x, y) where x is size and y is price
  - m training examples
- Cost function:
  - Mean Squared error
  - $MSE = \frac{1}{m}\sum_{i=0}^{m}(h(x_i) - y_i)^2$
- Model:
  - $h = w_0 + w_1 * x$
- Optimization method:
  - Solve for where its gradient is 0.
  - Gradient descent

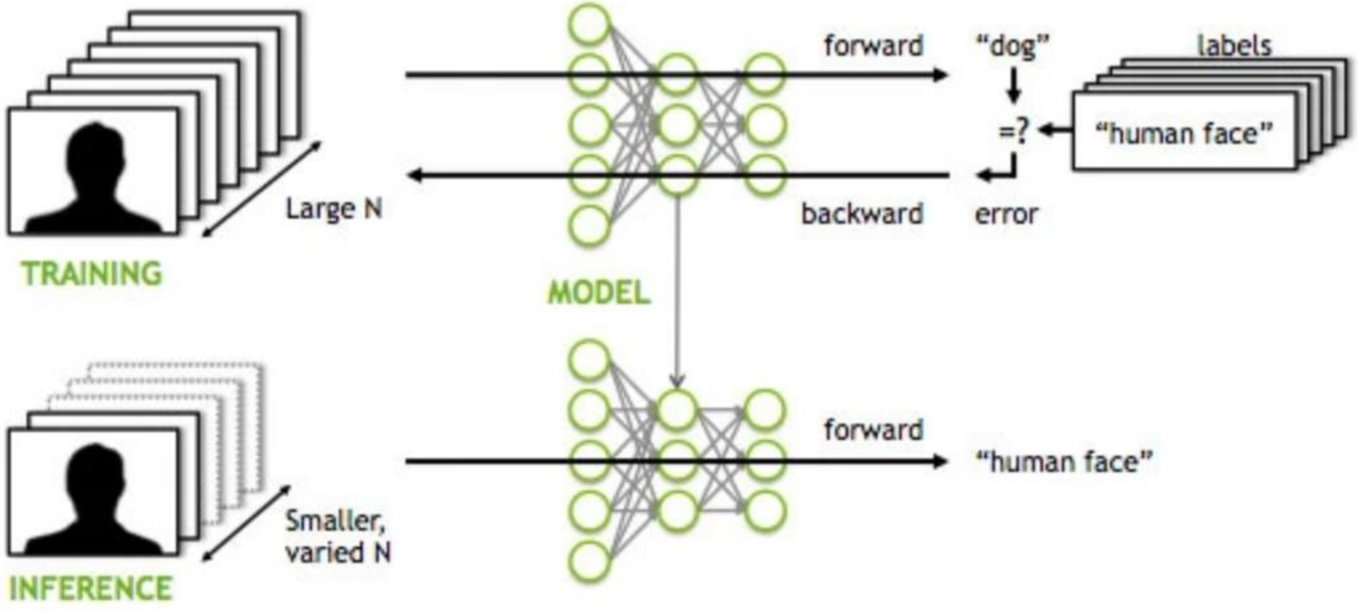


Size in feet² (x)	Price ($) in 1000's (y)
2104	460
1416	232
1534	315
852	178
…	…

Machine Learning, Andrew Ng

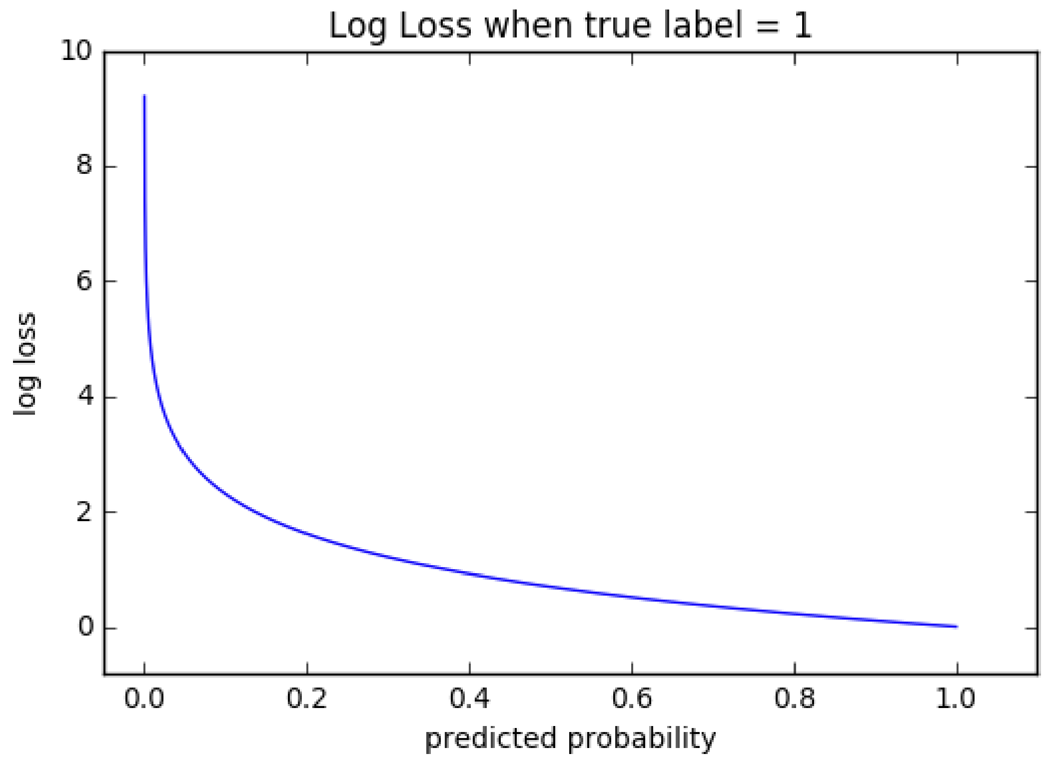
Training vs Inference

- Training
  - Dataset
  - Cost function
  - Optimization function
  - Model
- Inference
  - Dataset
  - Model



AlexNet Cost (loss) function

- Minimize the cross-entropy loss function
- Measures the performance of a classification model whose output is a probability value between 0 and 1
- $CF = -\frac{1}{N}\left(\sum_{i-1}^{N} y_i \cdot \log(\hat{y}_i)\right)$

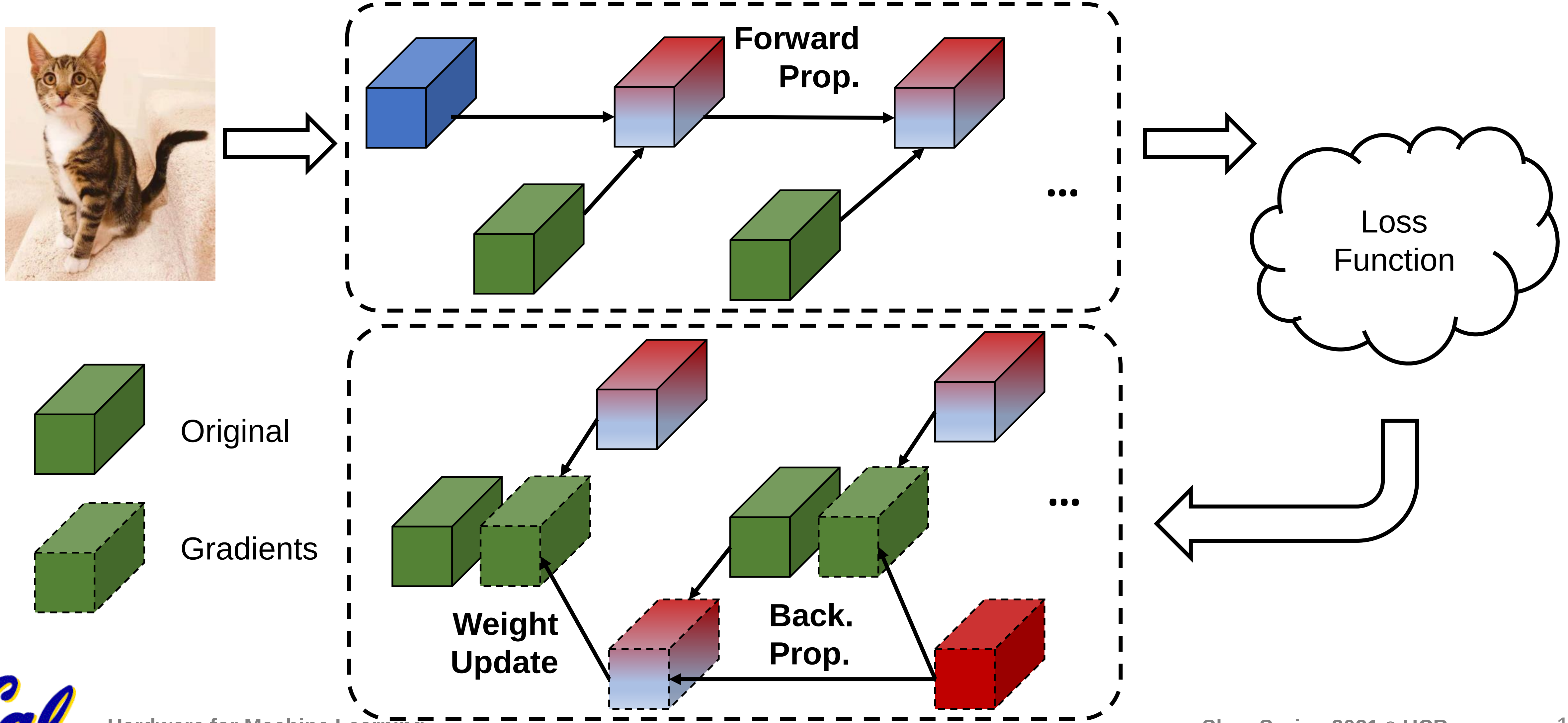


AlexNet Optimization Method

- Stochastic Gradient Descent
  - Batch size = 128
  - Update rule:

$$
\begin{aligned}
v_{i+1} &:= \underbrace{0.9}_{\text{Momentum}} \cdot v_i - \underbrace{0.0005}_{\text{Weight Decay}} \cdot \epsilon \cdot w_i - \underbrace{\epsilon}_{\text{Learning rate}} \cdot \underbrace{\left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}}_{\text{Gradient of cost function}} \\
w_{i+1} &:= w_i + v_{i+1}
\end{aligned}
$$

Training Data Dependencies



Administrivia

- Project (50%):
  - Proposal (10%)
  - Checkpoint * 2 (5% * 2. Required.)
  - Presentation (10%)
  - Final report (15%)
  - Results (5%)

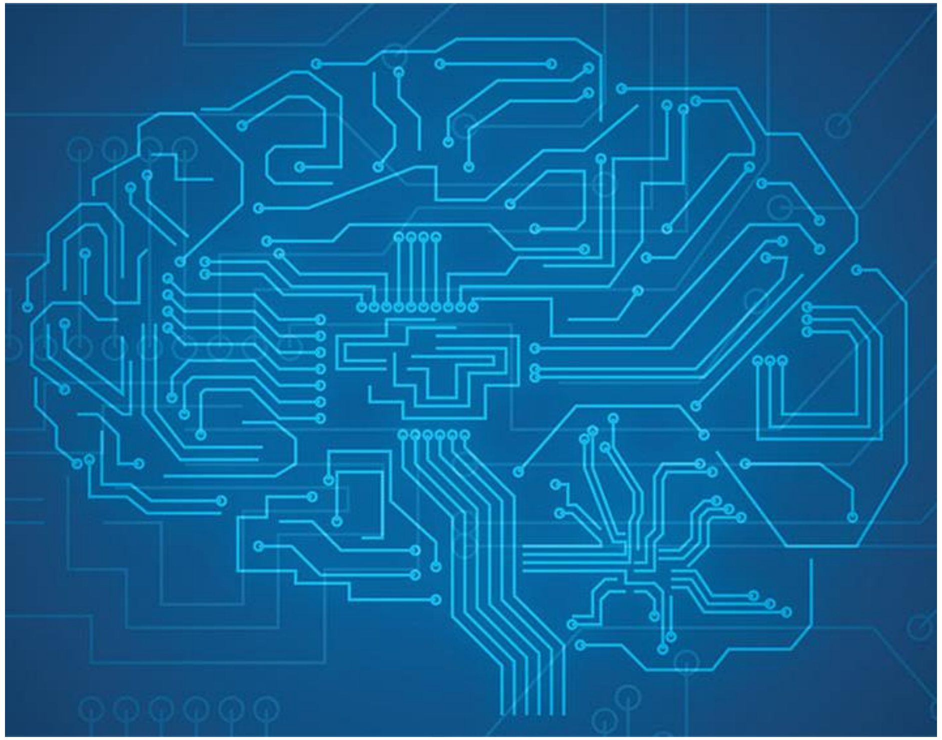
Administrivia

- Project Checkpoint 1:
  - Sign up here:
    - https://docs.google.com/spreadsheets/d/1Mer3sGy5jVTP2KKykFJ_QABfyE9nEy44-fW1diAVCCU/edit?usp=sharing
    - Please prepare 2 or 3 slides showing your current progress.
  - The slides should include:
    - A diagram representing your hardware/software design and showing how it fits into your full system.
    - An account of the progress you've made, and whether you've reached your planned Checkpoint 1 goals.
    - A description of any changes you may have made to your plans or your project.

Administrivia

- Project Presentation (5/3 tentative)
  - 10 mins presentation + 5 mins Q&A

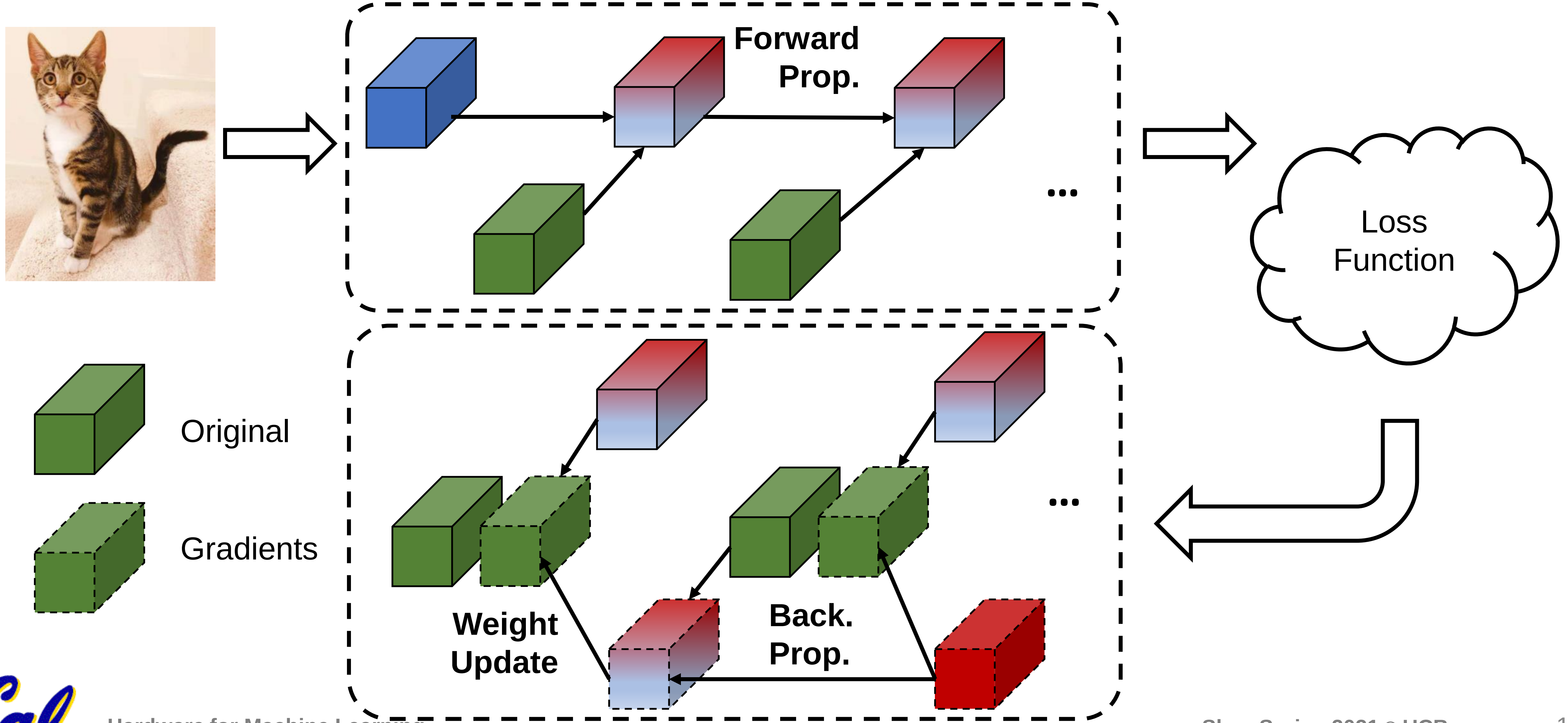
- Project Report (due 5/7)
  - Max 8 pages
  - 2-column format. Use Latex (e.g., with overleaf)
  - Provide a link to your code repository.



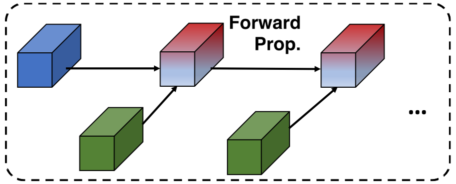
Training

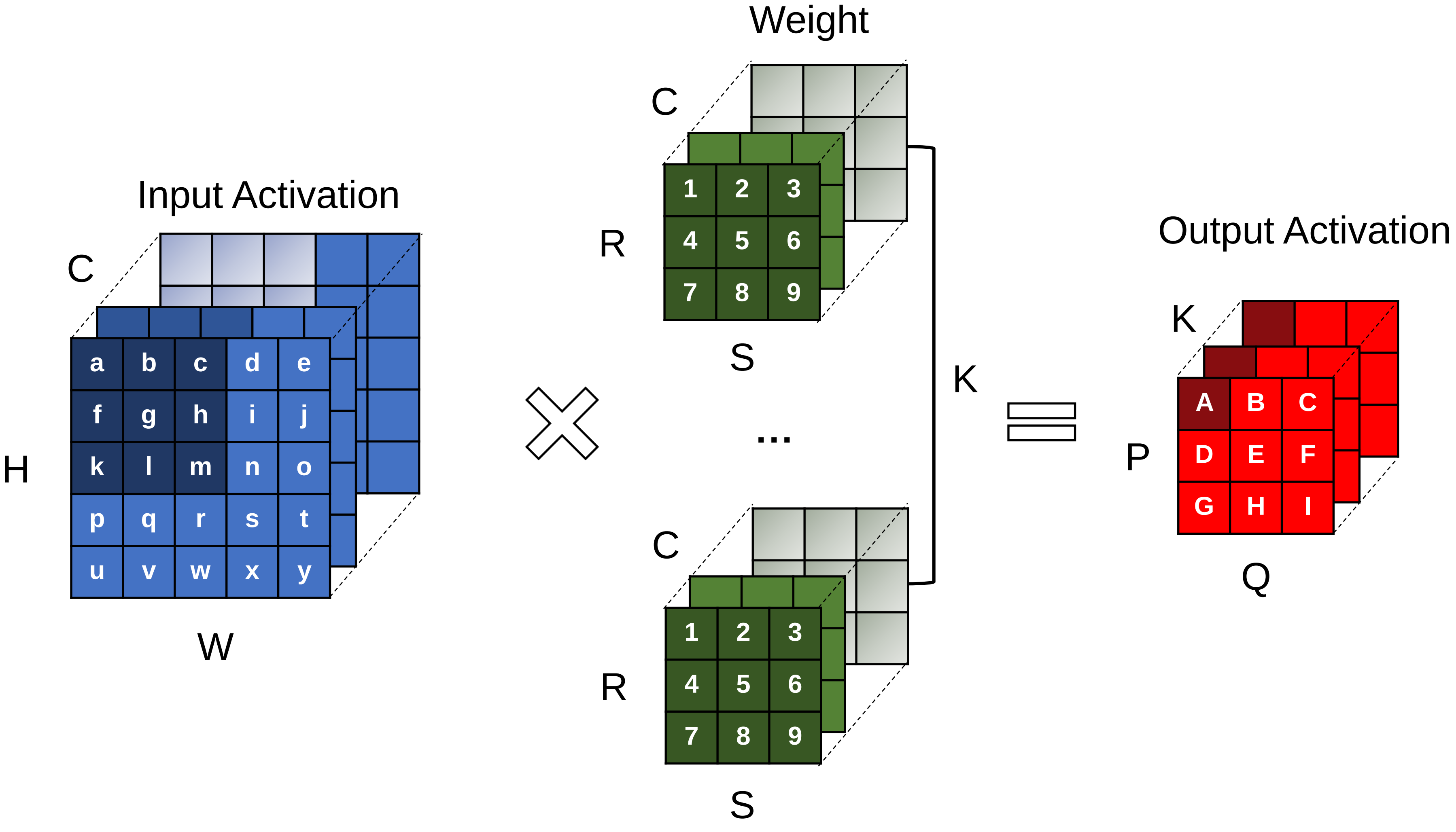
- Training Flow Recap
- Key Kernels
  - Forward Propagation
  - Backward Propagation
  - Weight Update

Training Data Dependencies



Forward Propagation



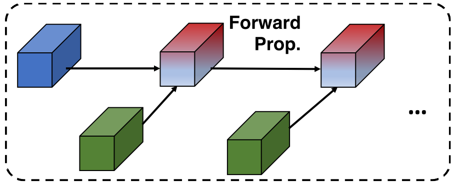


H: Height of Input Activation
W: Width of Input Activation
R: Height of Weight
S: Width of Weight
P: Height of Output Activation
Q: Width of Output Activation
stride: # of rows/columns traversed per step
padding: # of zero rows/columns added

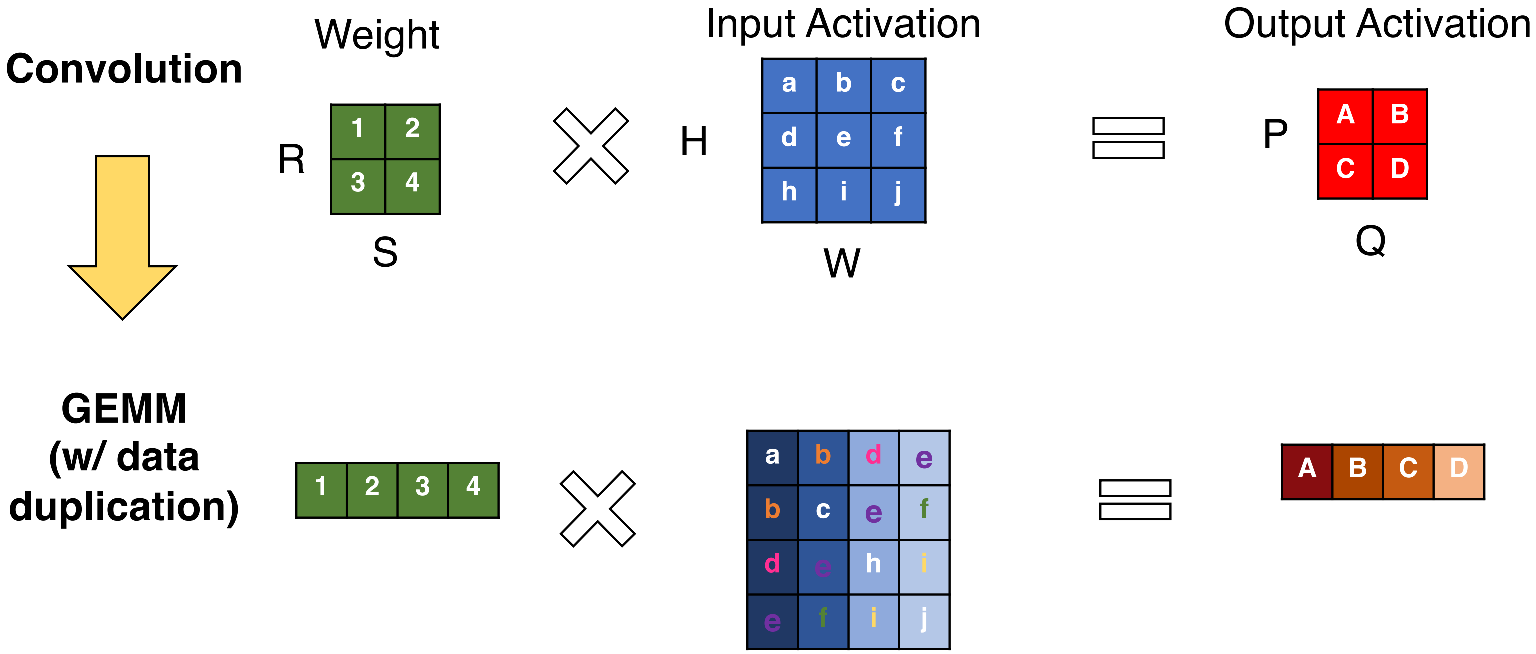
C: # of Input Channels
K: # of Output Channels
N: Batch size

Reduction Dimension:
RSC

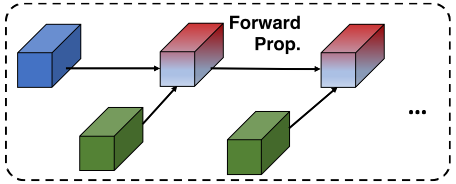
Forward Propagation

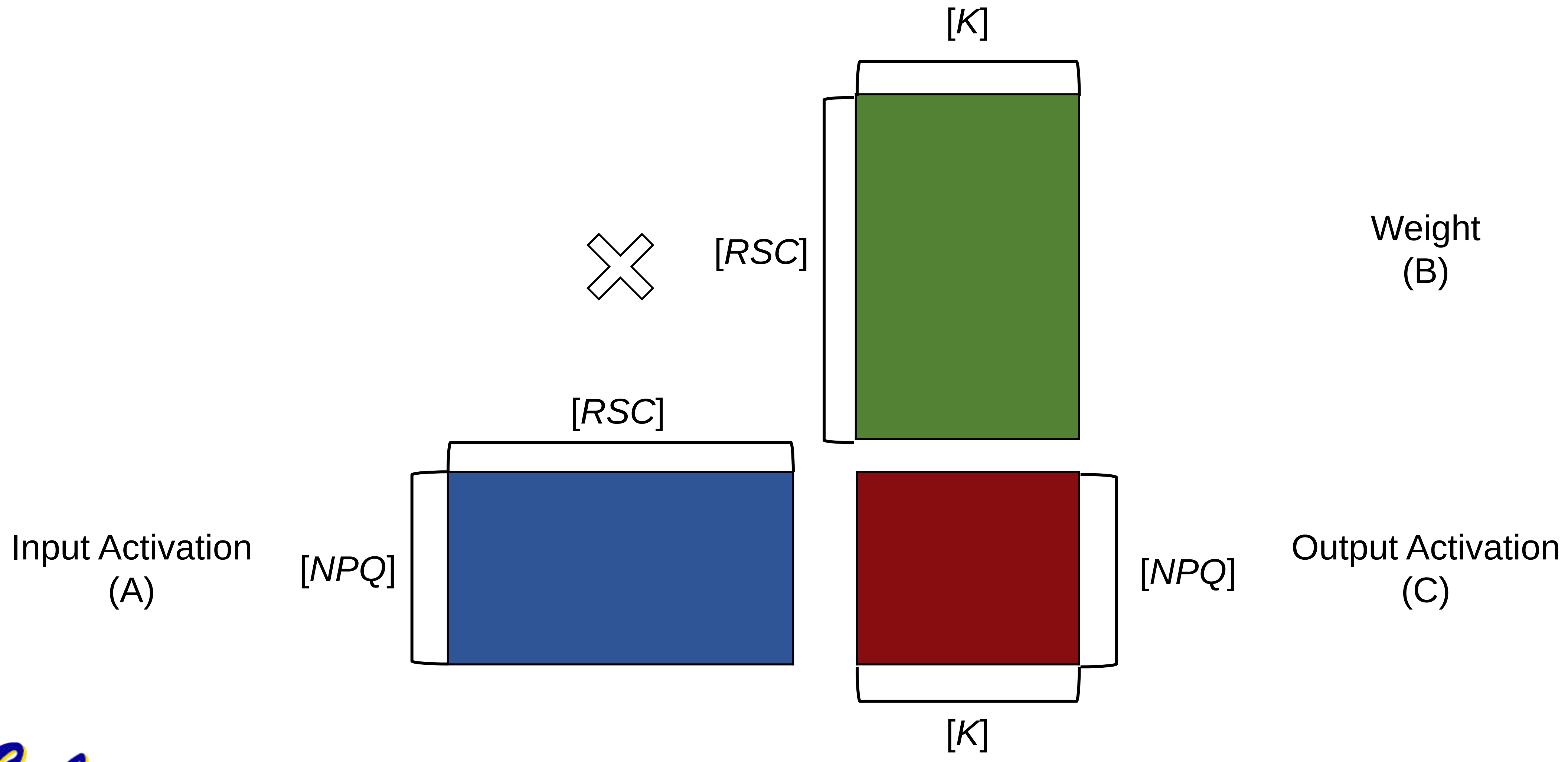


- Converting convolution to GEMM via **im2col**

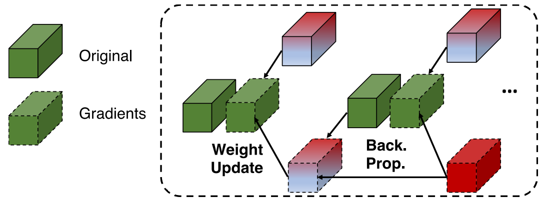


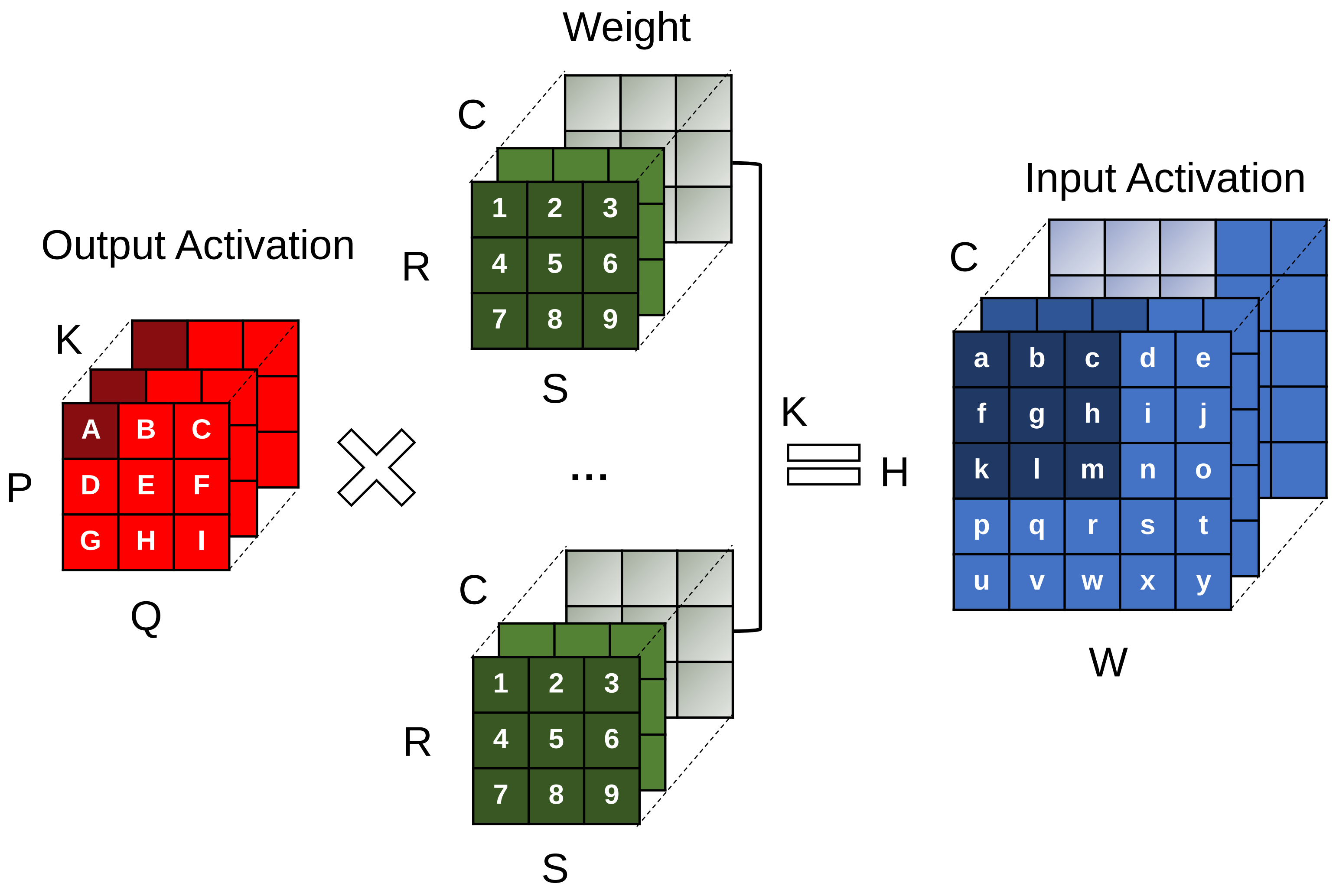
Forward Propagation

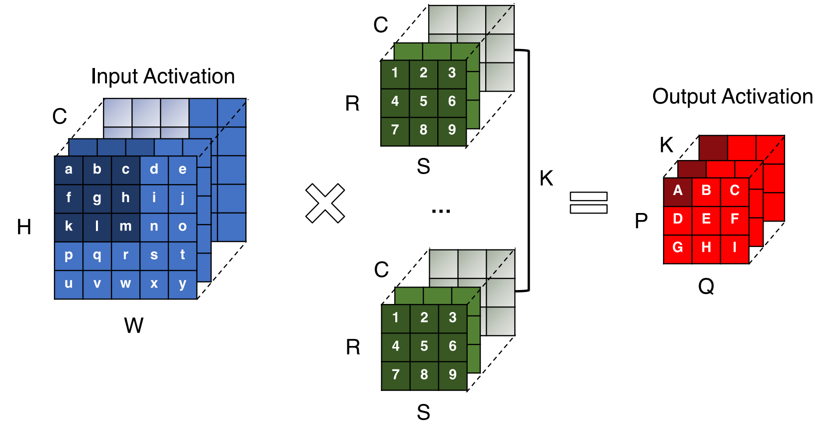




Backward Propagation

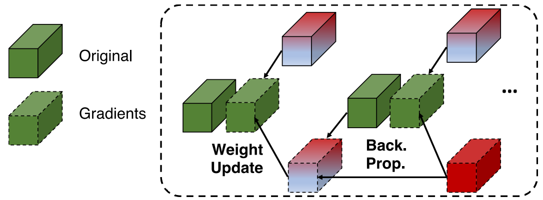


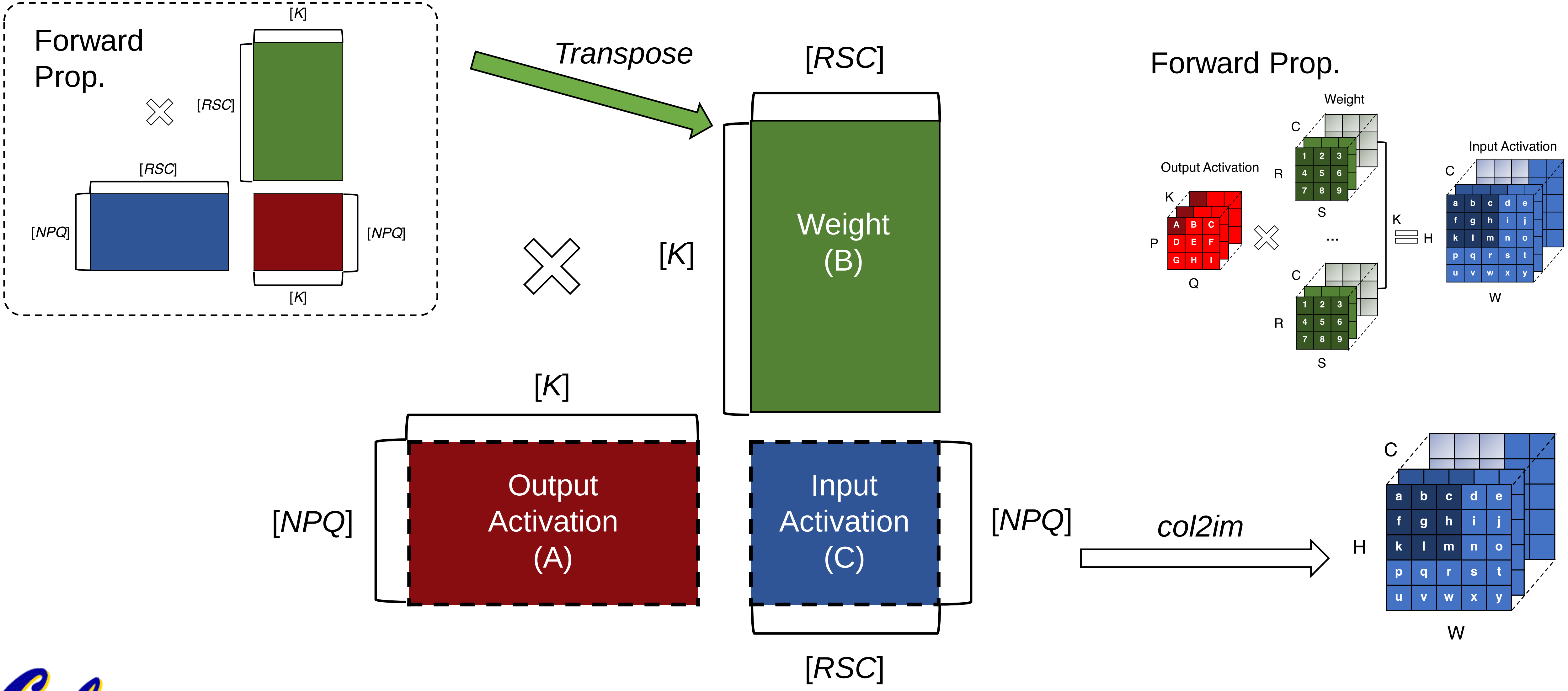




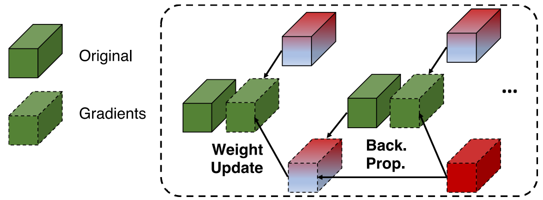
Reduction Dimension: K

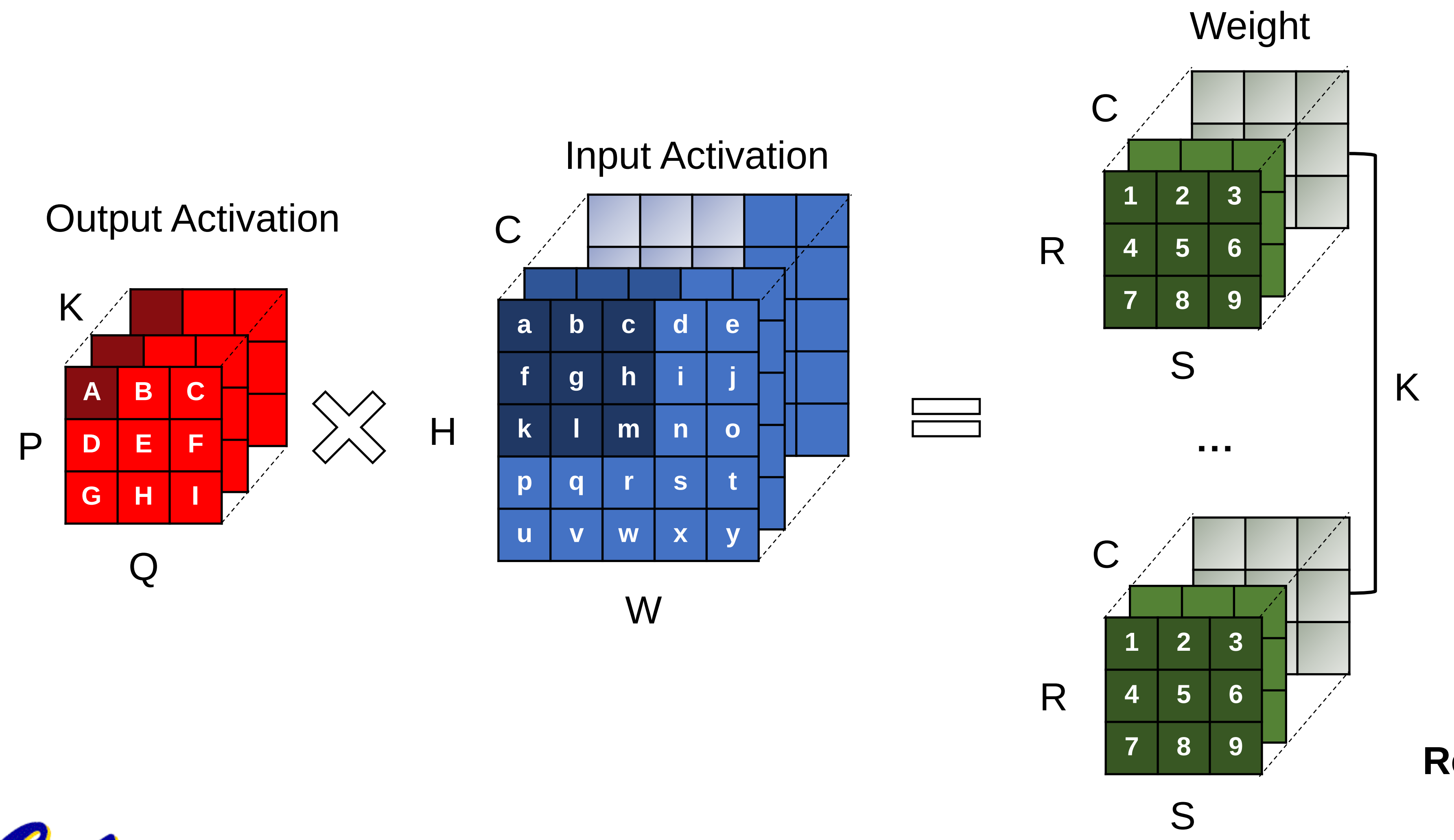
Backward Propagation

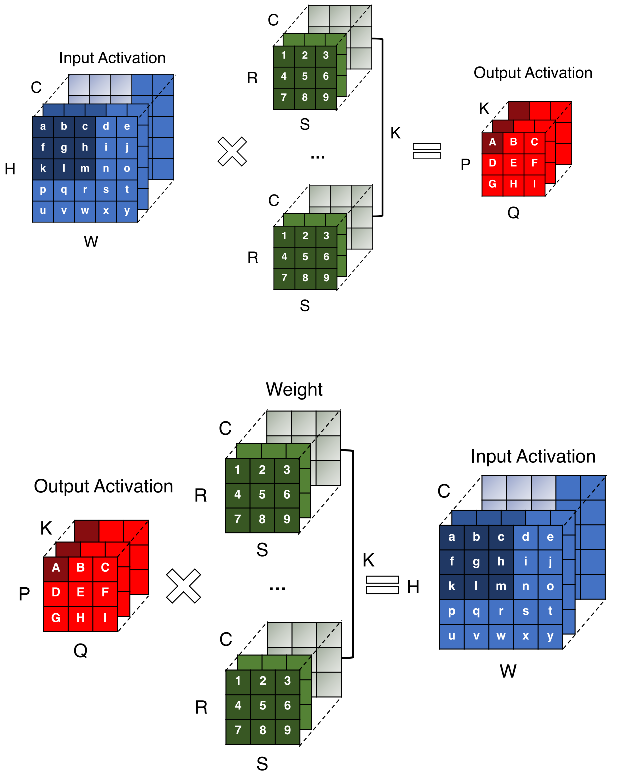




Weight Update

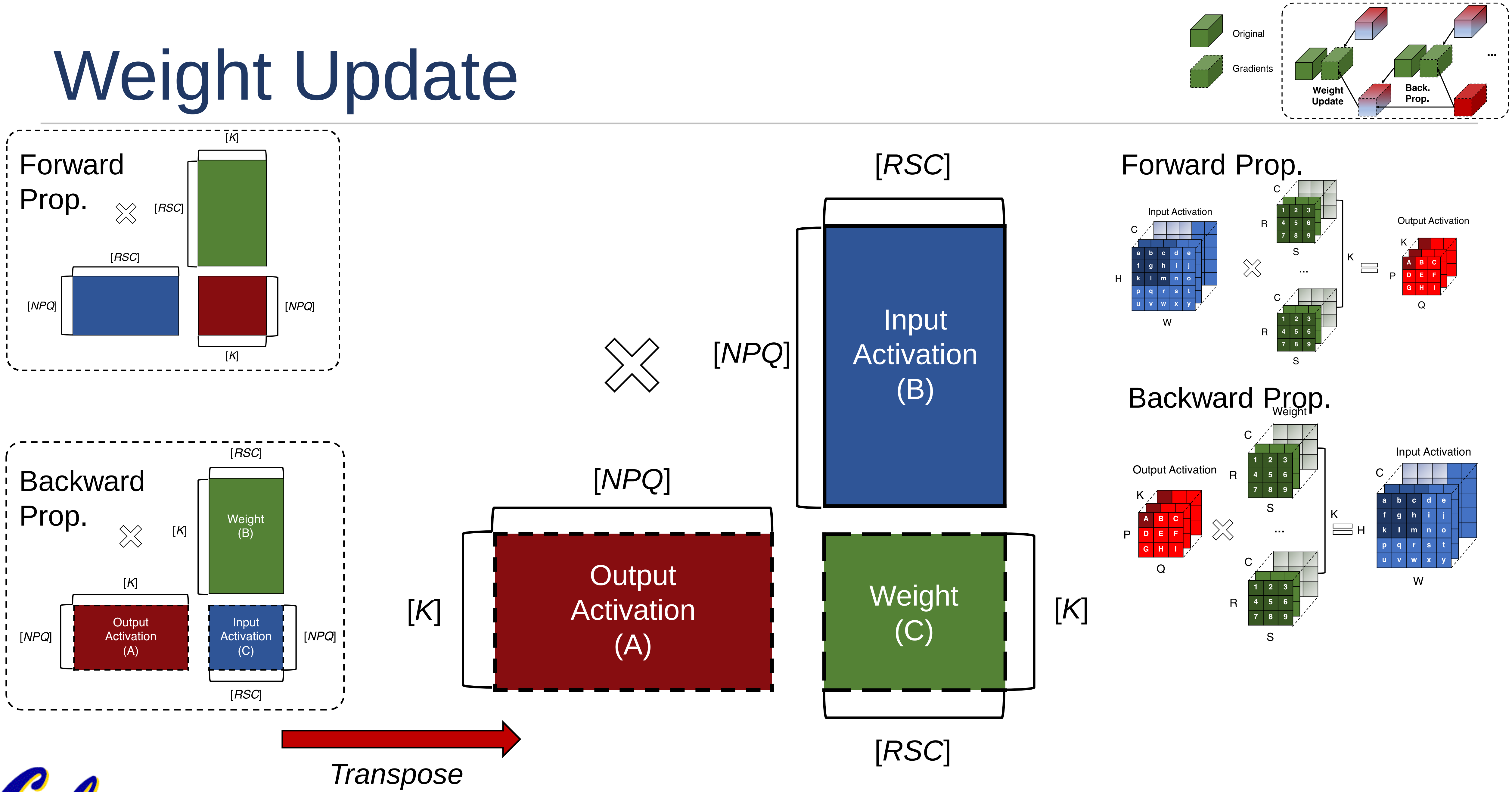




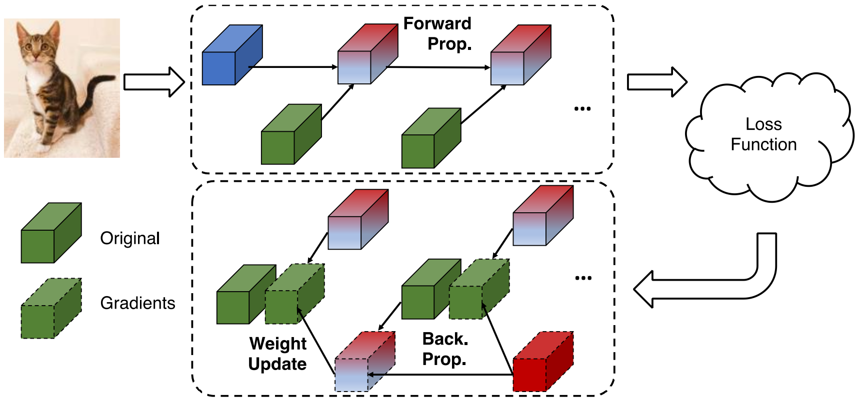


Reduction Dimension: PQ

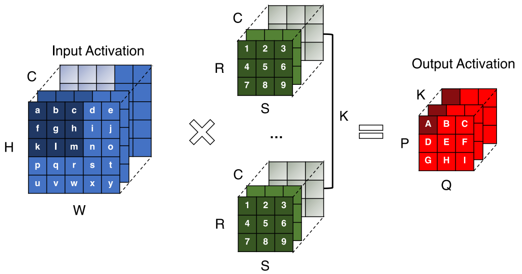
Weight Update

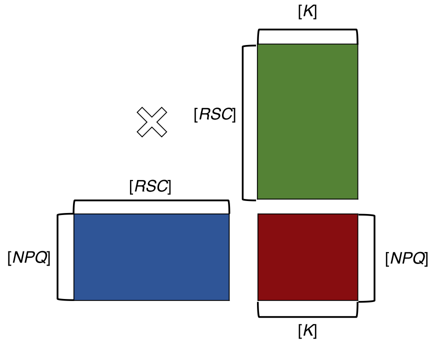


Training Flow Overview

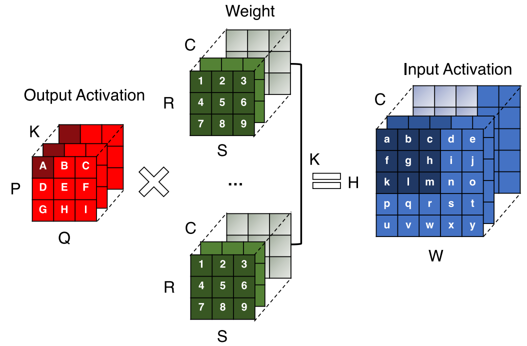


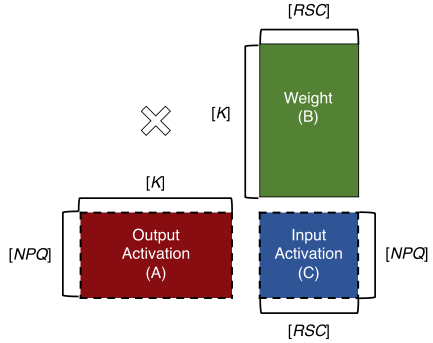
Forward Prop.



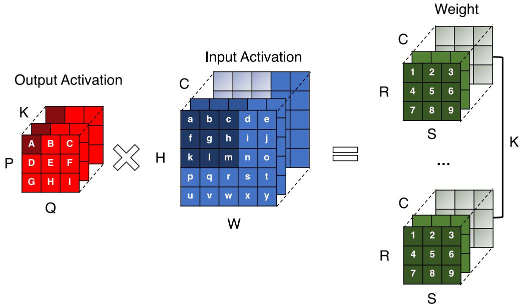


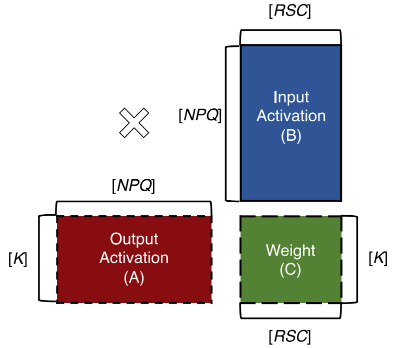
Backward Prop.





Weight Update





Review

- Core computation in DNN
- Execution order of the core computation
- Hardware realization of the core computation
- Mapping DNNs to hardware
- Data transfer mechanisms across storage hierarchy
- Sparsity in DNNs
- Codesign example
- Other Operators and Near-Data Processing
- Training Kernels
  - Training Flow Overview
  - Core kernels:
    - Forward Propagation
    - Backward Propagation
    - Weight Updates