Kiel University
Christian-Albrechts-Universität zu Kiel

Distributed Systems
Research Group

Activation Sparsity and Dynamic Pruning for Split Computing in Edge AI

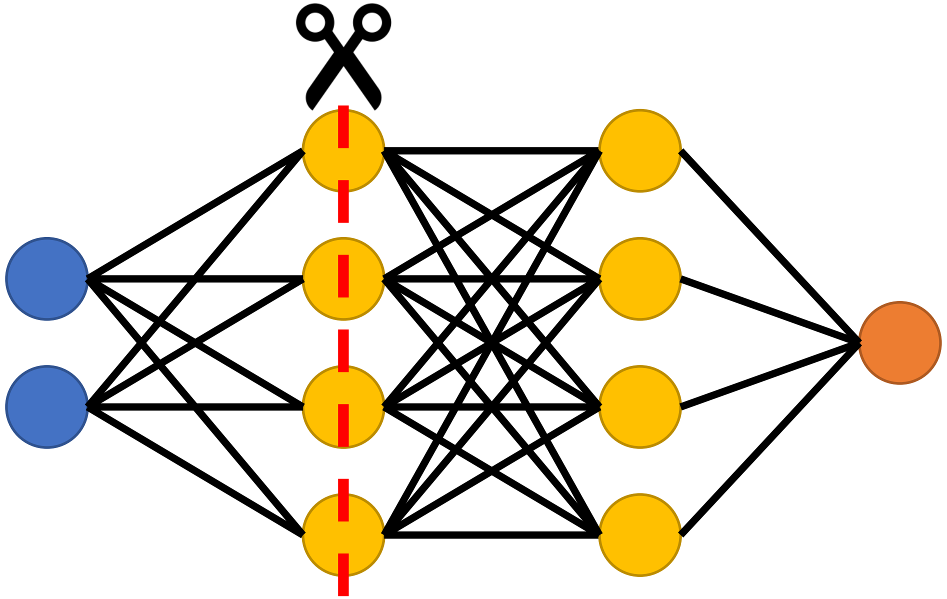
Janek Haberer [1], Olaf Landsiedel [1, 2]

[1] *Kiel University, Germany*
[2] *Chalmers University of Technology, Sweden*

Motivation

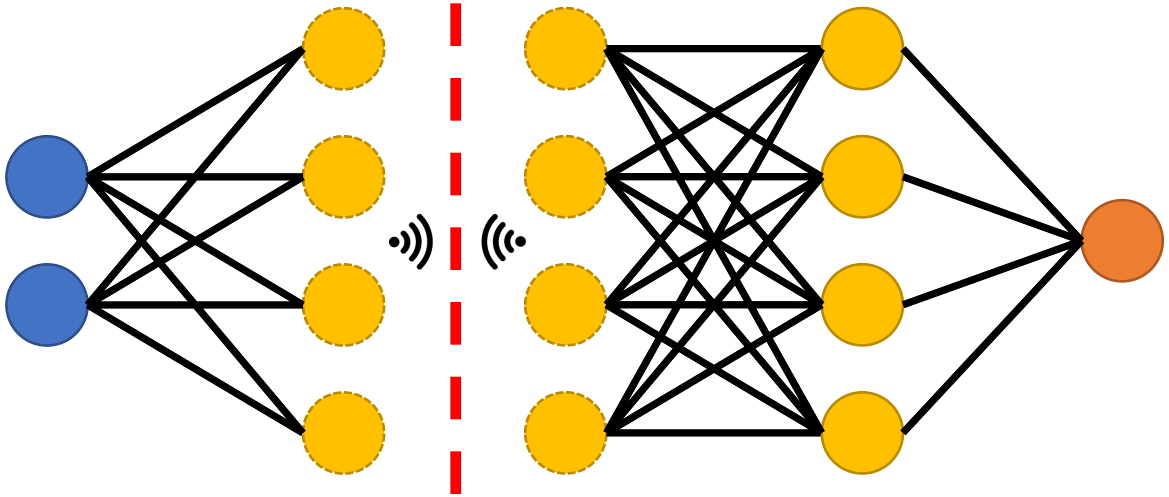
- Networks are increasing in size
  - Highest accuracies usually need **many parameters**
- Sensor devices have limited resources
- Big models can exceed these constraints
  - Use cloud to help via offloading
- Using wireless is **very expensive**
  - Latency
  - Energy

Use Split Computing!



- Split network
  - Part of inference on sensor
  - Remaining part on a server

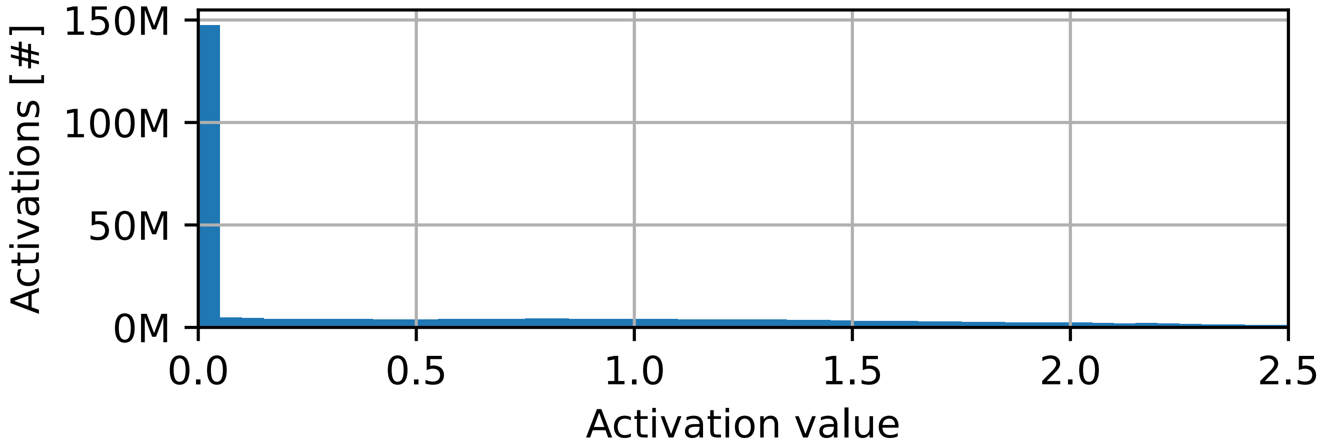
- Why?
  - Reduces load on server
  - Slightly more privacy aware
  - Reduces network communication

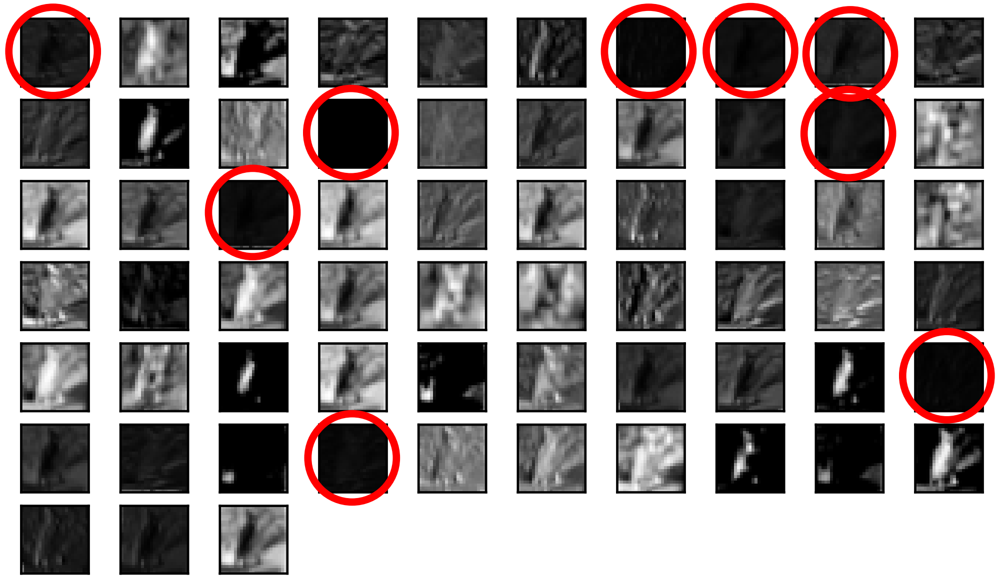


Let’s look into intermediate outputs

- ResNet-50 on CIFAR10
  - Most values are near zero
  - Many feature maps are black

- Zeros have no impact





How can we use sparsity for split computing?

- We analyze sparsity in individual layers
  - Activation sparsity
  - Feature map sparsity
  - Guidance for choosing splitting points

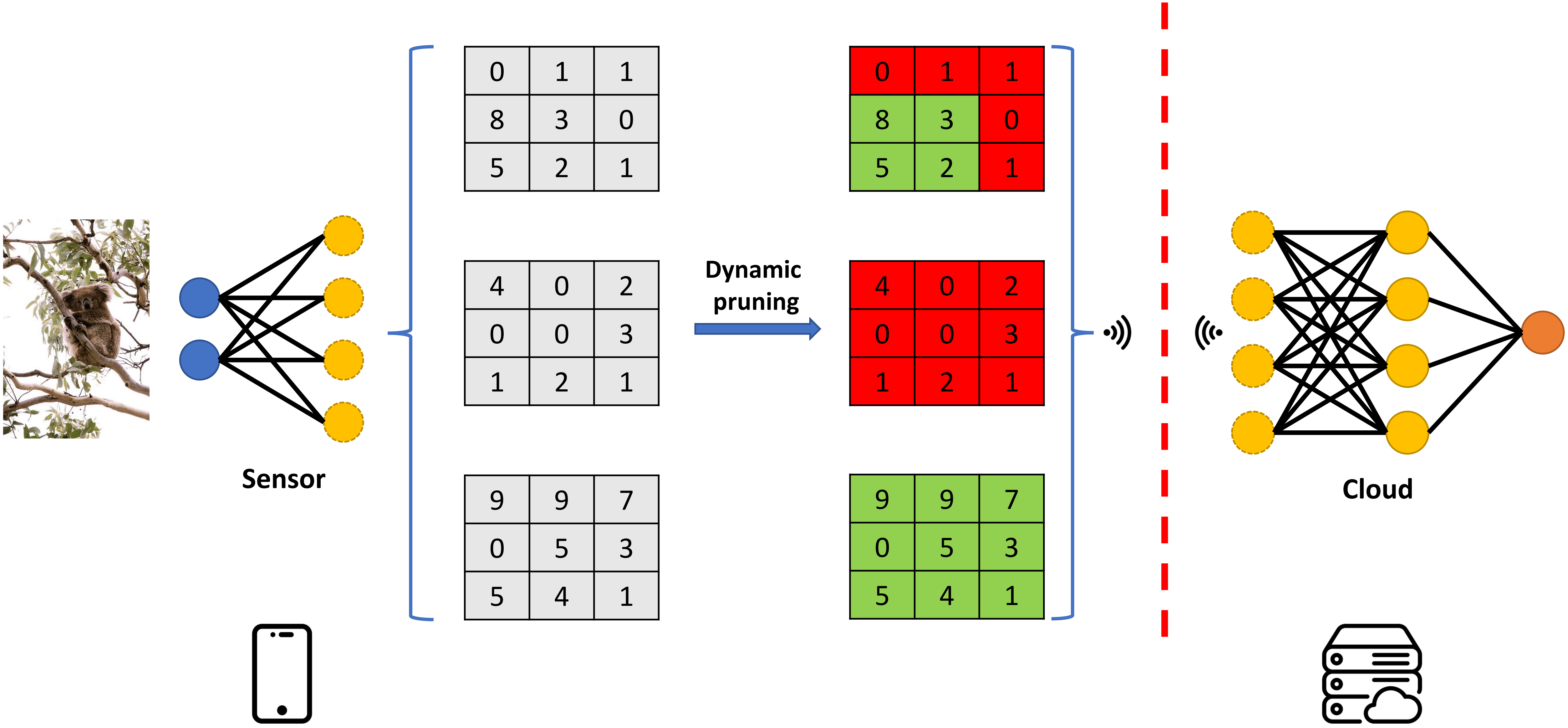
- We apply and evaluate dynamic pruning
  - Dynamic activation pruning
  - Dynamic feature map pruning
  - Show potential for compressing data

Classical pruning versus activation pruning

- Many works use pruning
  - Show high sparsity especially in dense layers
  - Usually based on weights

- Problem: weights are static
  - We are transmitting activations, not weights
  - Activations change depending on the input

How does dynamic pruning work?



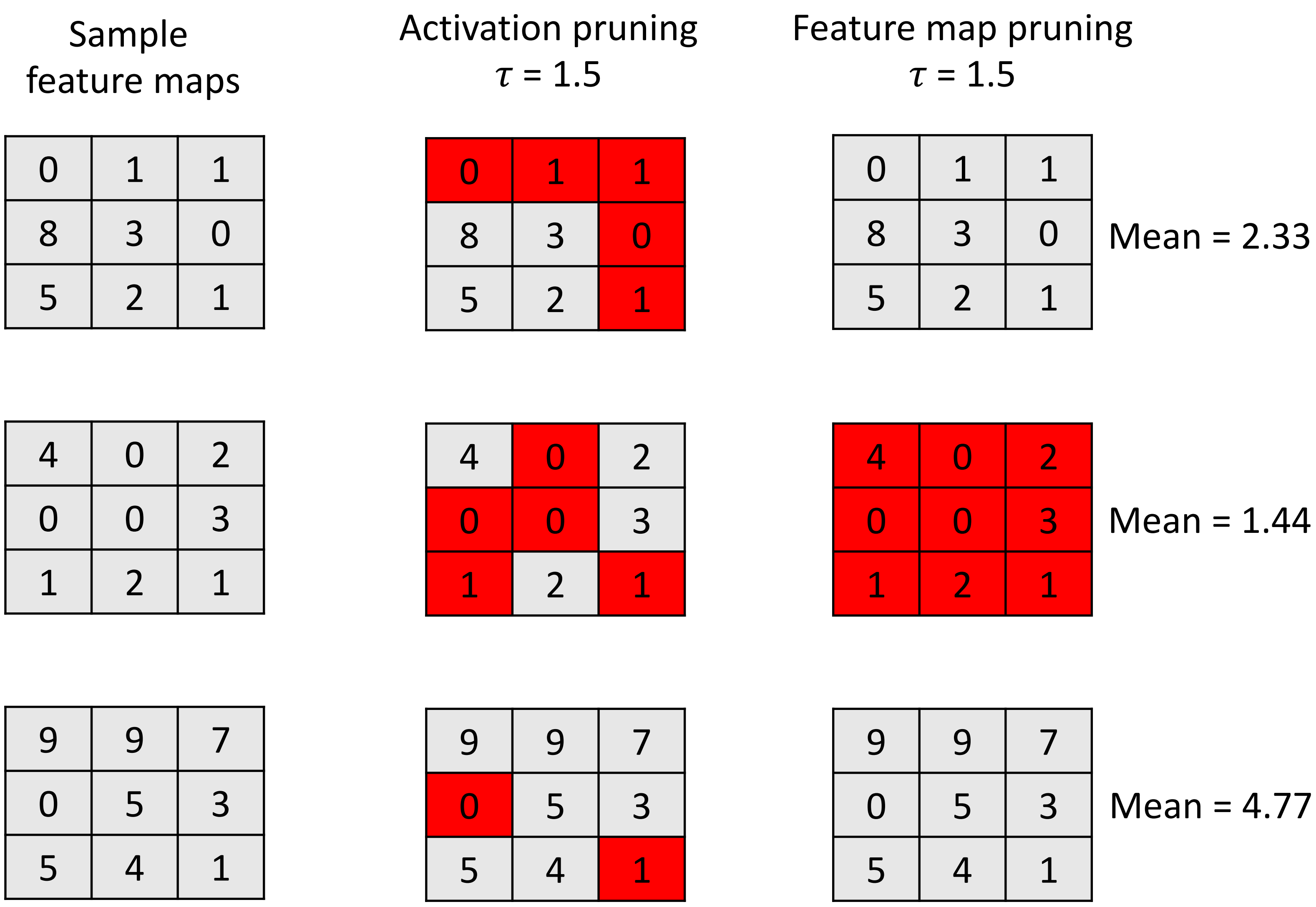
0	1	1
8	3	0
5	2	1

4	0	2
0	0	3
1	2	1

9	9	7
0	5	3
5	4	1

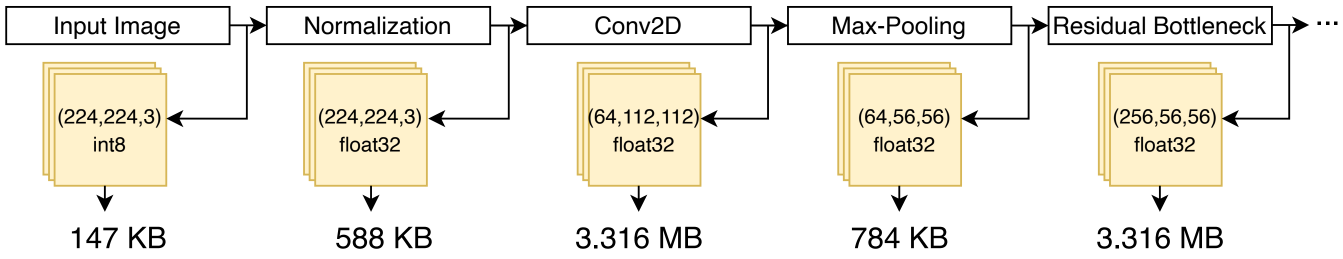
How does dynamic pruning work?

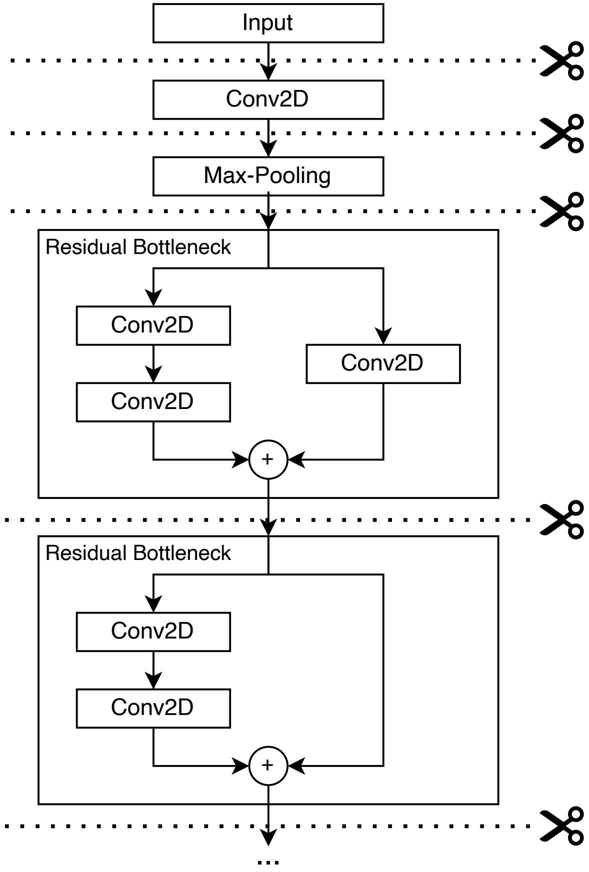
- Prune according to threshold
  - Individual values
  - Feature maps
- We are **not** removing **statically**



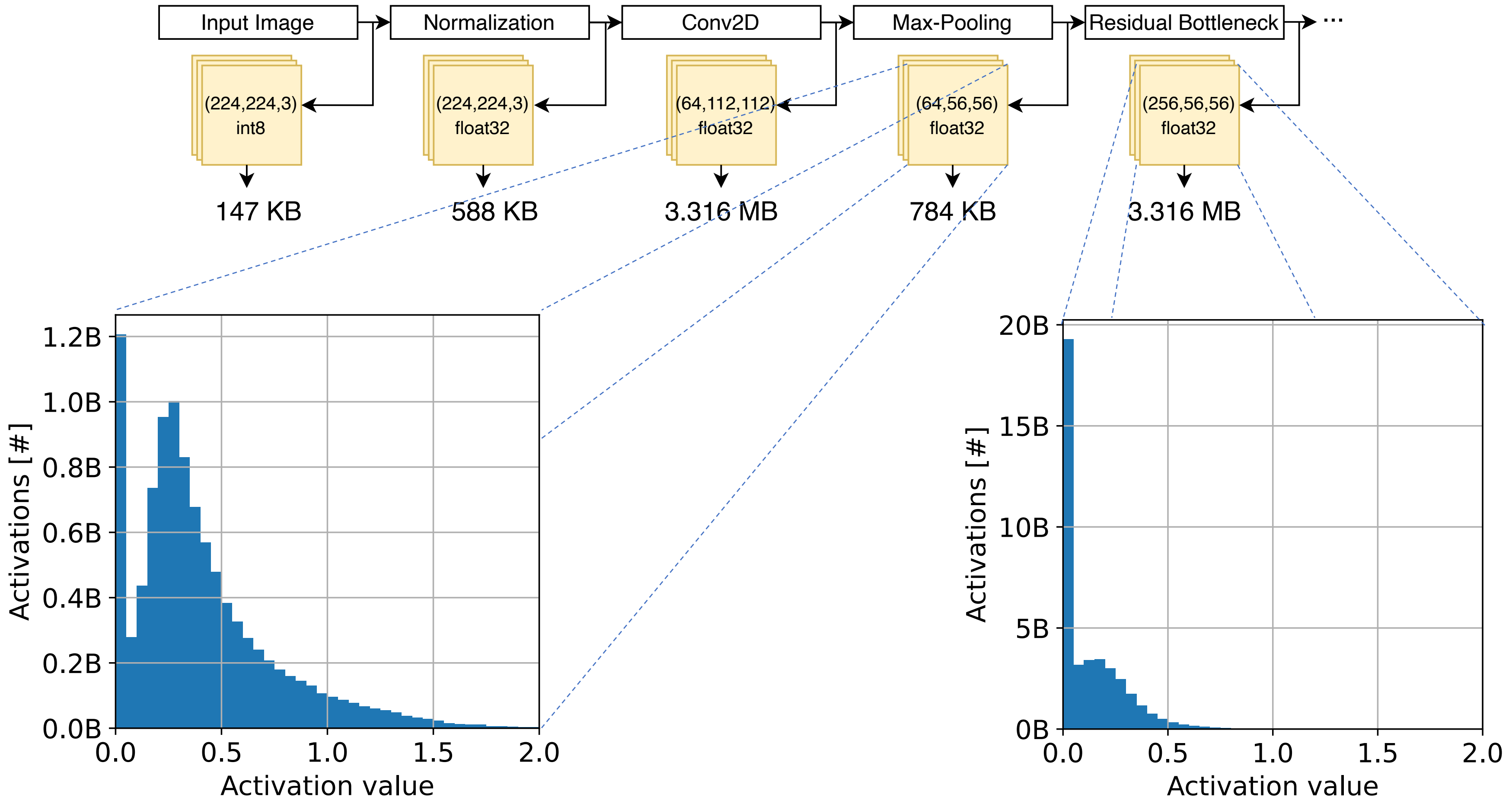
Where can we place splitting points?

- Trend goes towards parallel branches
  - Adds additional data
  - Choose points where branches end
- CNNs contain more data in early layers
  - Choose point as late as possible





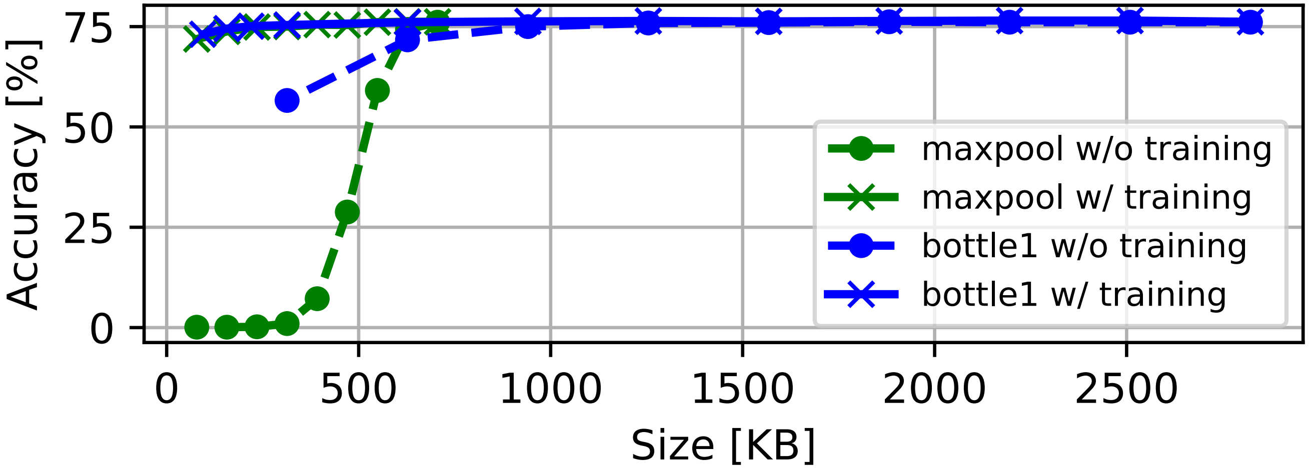
Let's look into activation sparsity

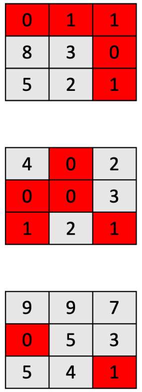


- 12% of values between 0 and 0.05
    - Amounts to 94 KB
    - 690 KB remain

- 48% of values between 0 and 0.05
    - Amounts to 1.5 MB
    - 1.6 MB remain

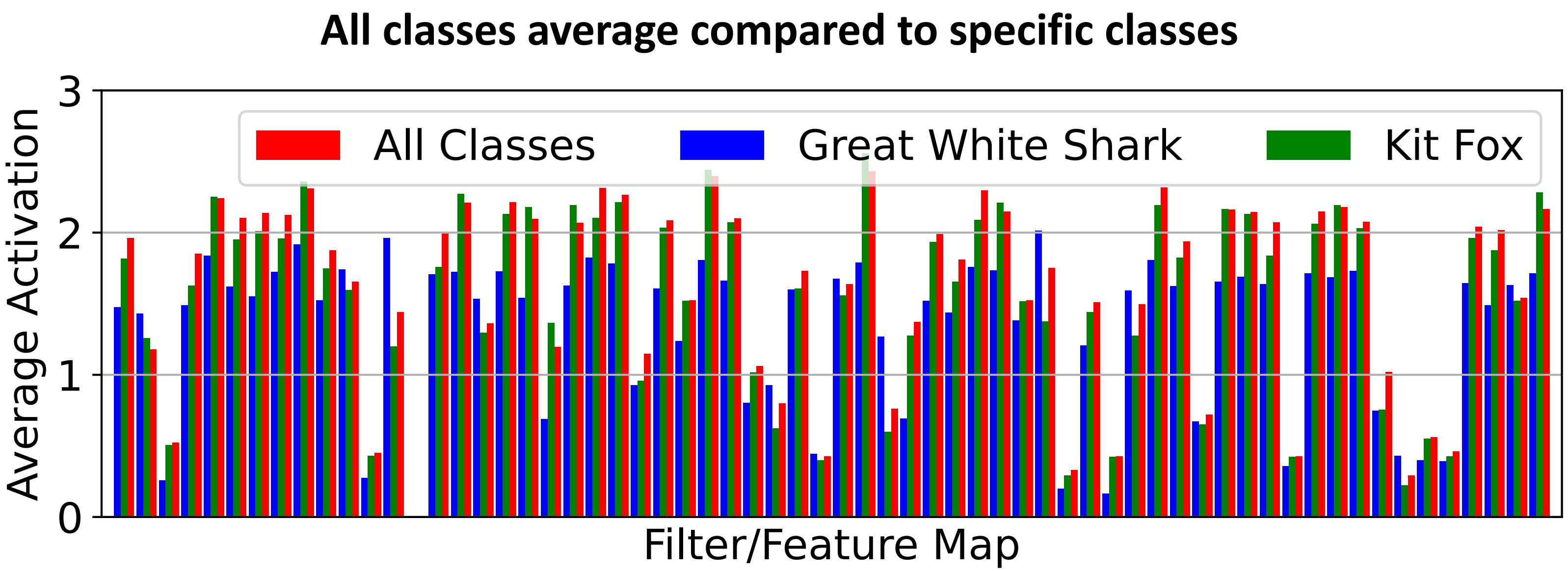
How does dynamic activation pruning affect inference?





- Without fine-tuning:
  - Up to 70% reduction with 1% loss of accuracy
  - Going below 700 KB hurts
- With fine-tuning:
  - Up to 93% reduction with 1% loss of accuracy
  - Going below 200 KB hurts

Why should we use dynamic pruning?



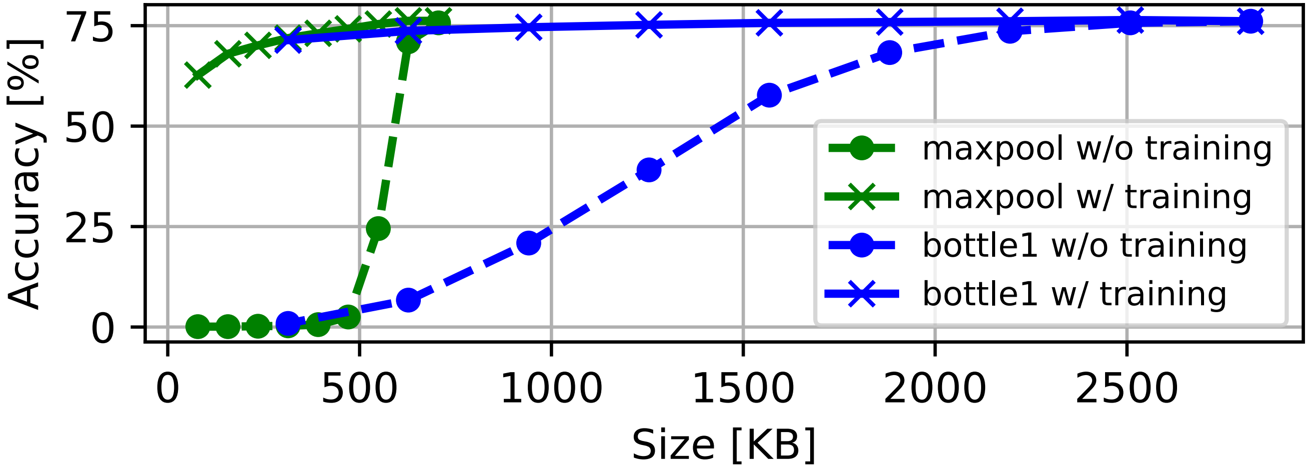
- Significant differences between classes
  - Images of sharks are very different than images of foxes

How does dynamic feature map pruning affect inference?

0	1	1
8	3	0
5	2	1

4	0	2
0	0	3
1	2	1

9	9	7
0	5	3
5	4	1



- Without fine-tuning:
  - Only up to 20% reduction with 1% loss of accuracy
  - Can reduce to 700 KB
- With fine-tuning:
  - Up to 60% reduction with 1% loss of accuracy
  - Can reduce to 550 KB

Conclusion

- Up to 48% near-zero values in activations
- Dynamic pruning allows for efficient splitting of DNNs
  - Compression of up to 93% with minimal loss of accuracy
  - Feature map pruning worse than activation pruning

Future Work

- Analyze sparsity inducing techniques
- Evaluate effect of quantization and encoding schemes

Activation Sparsity and Dynamic Pruning for Split Computing in Edge AI