Learning Transferable Architectures for Scalable Image Recognition

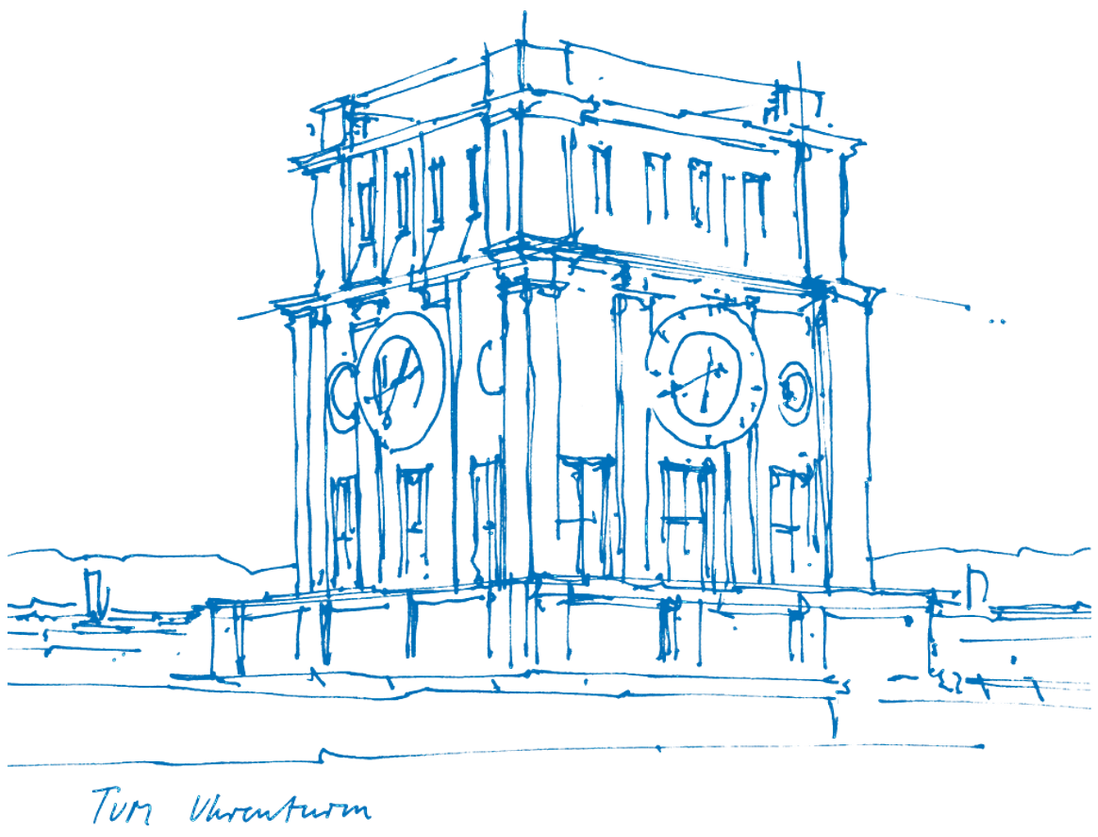
- Barret Zoph, Vijay Vasudevan, Jonathon Shlens, Quoc V. Le

Seminar - Recent trends in Automated Machine Learning

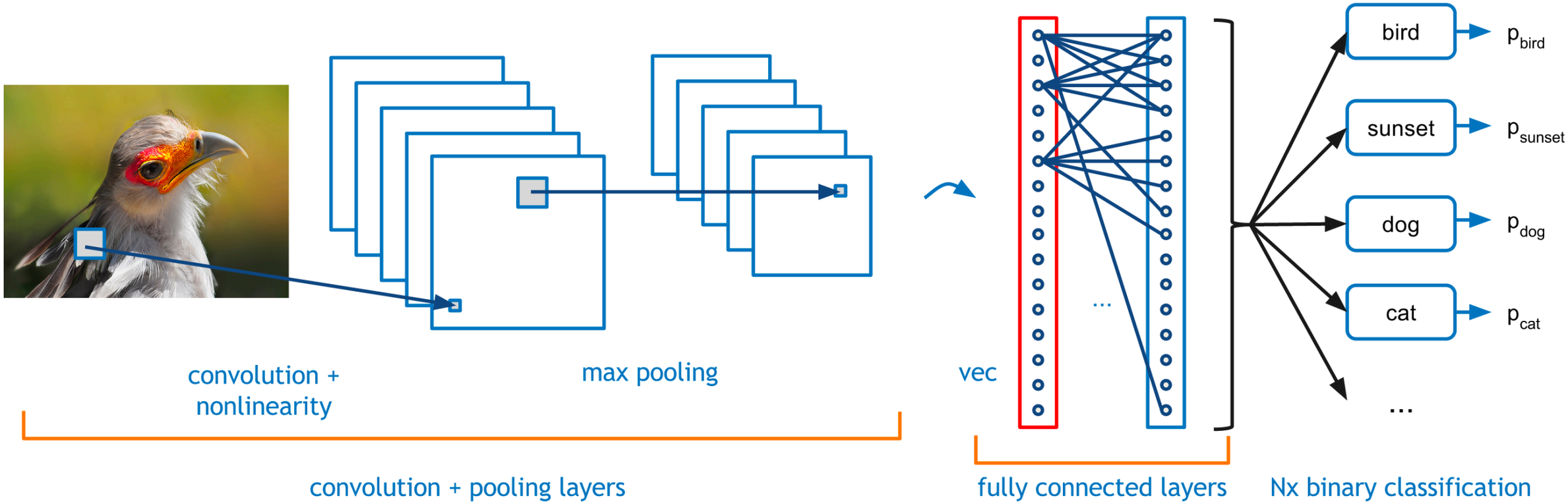
Sebastian Fellner

Technische Universität München

06. June 2019, Garching



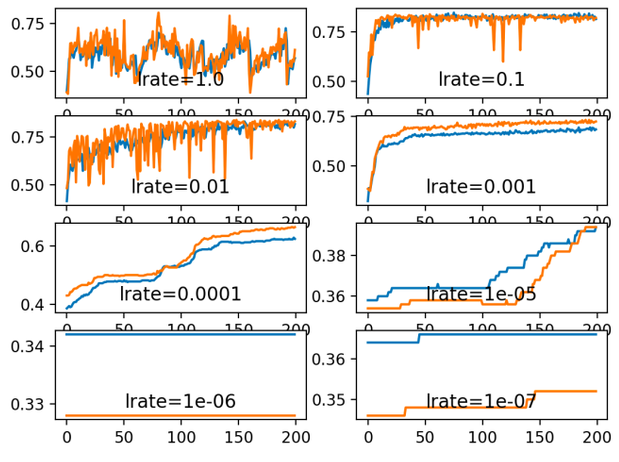
Problem statement



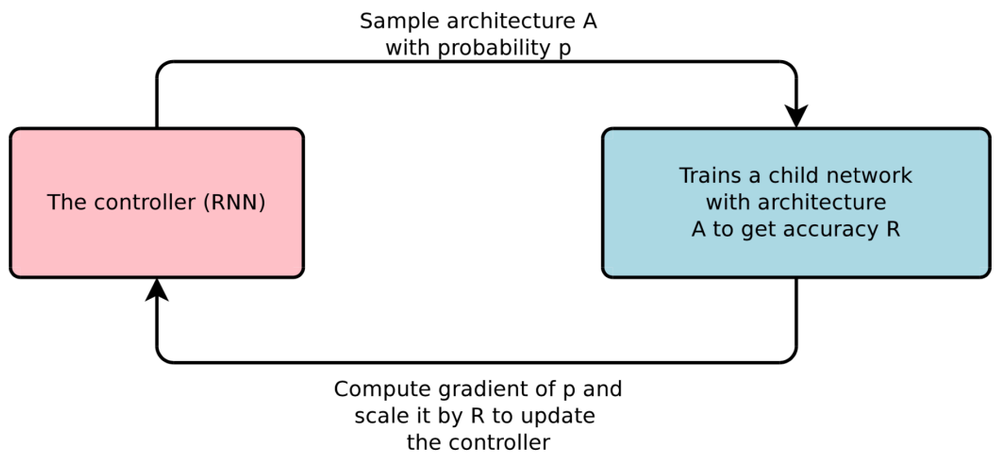
Train a neural network image classification model

Previous solutions and shortcomings

- Architecture engineering
  - Requires domain knowledge
  - Trial and error

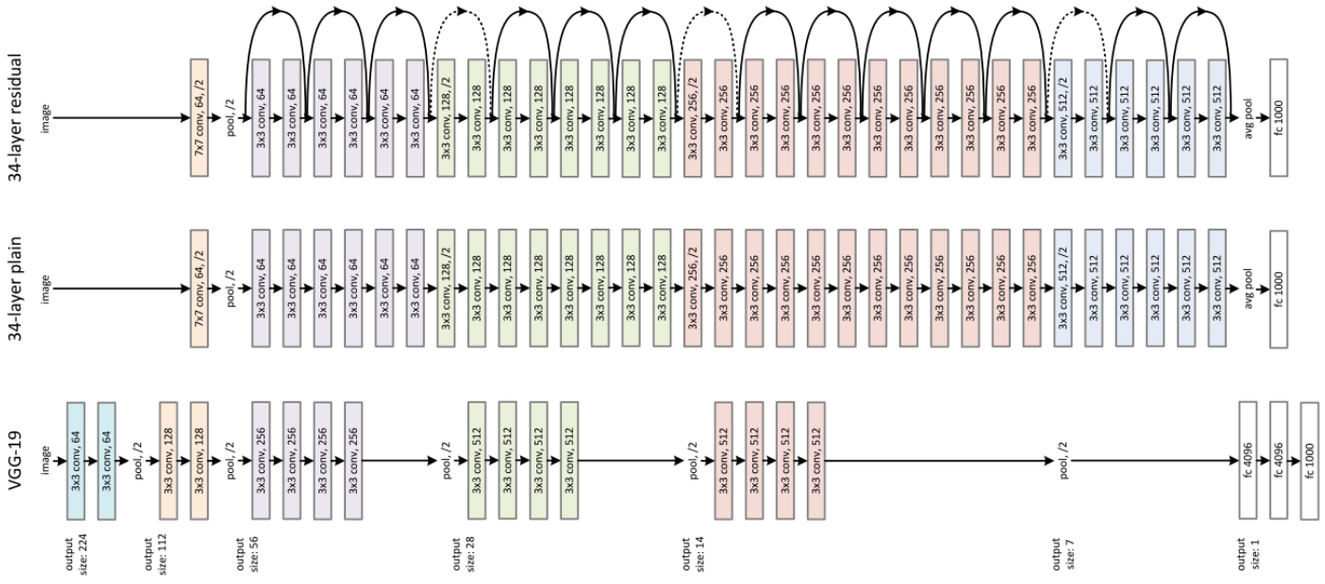


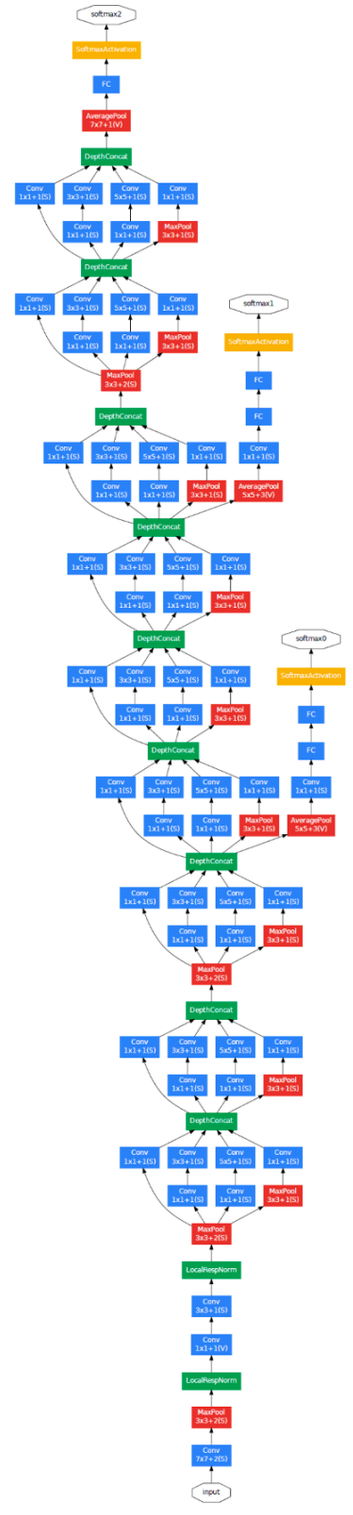
- NAS
  - Architecture search is limited to one dataset a time
  - No transferability
  - No scalability



NASNet search space - general idea

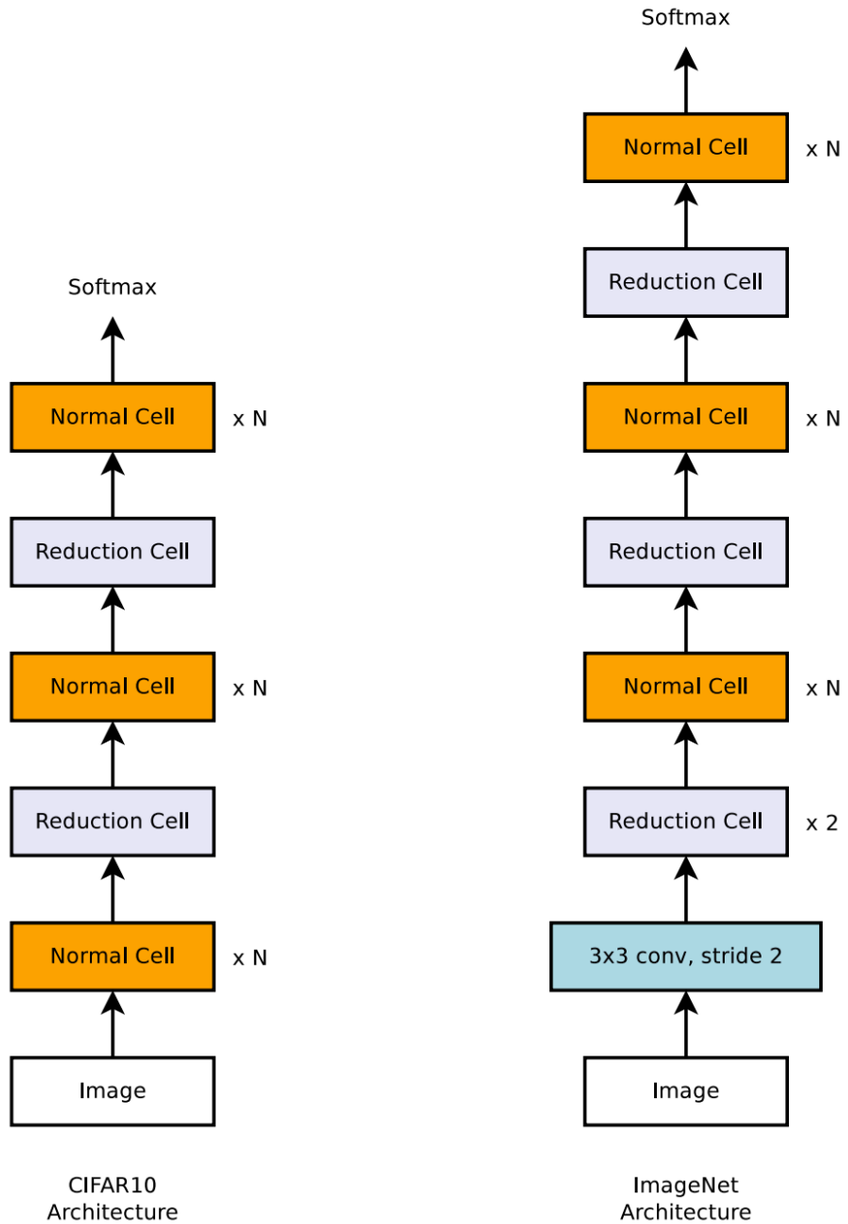
- Observation: handcrafted architectures often contain a lot of repetition
- Reduce search space to cells
  - Repeat those for whole architecture
  - Enables transferability
  - search/training is converges faster
  - Generalises better for other tasks
- Only convolutional layers



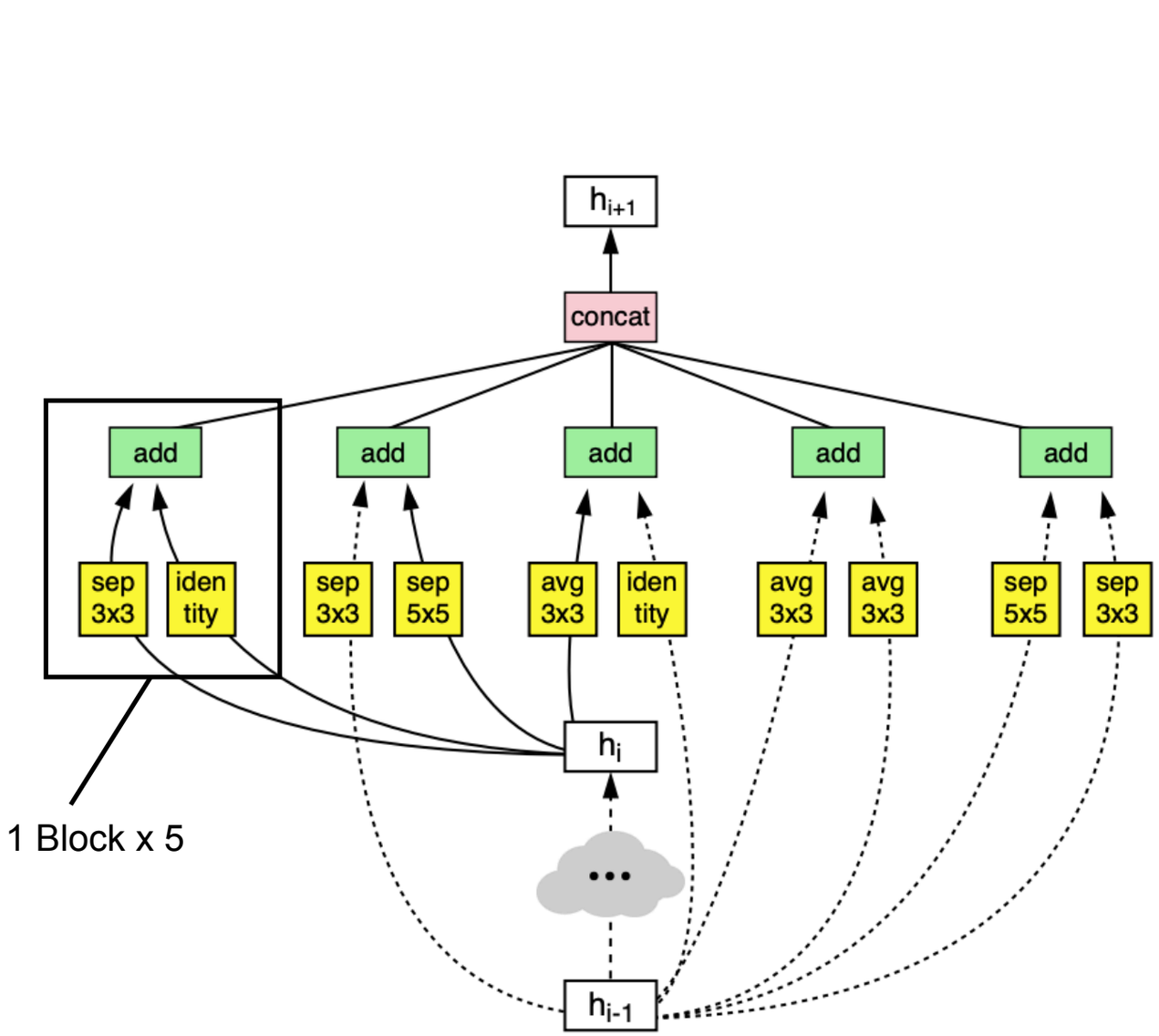


NASNet search space - architecture

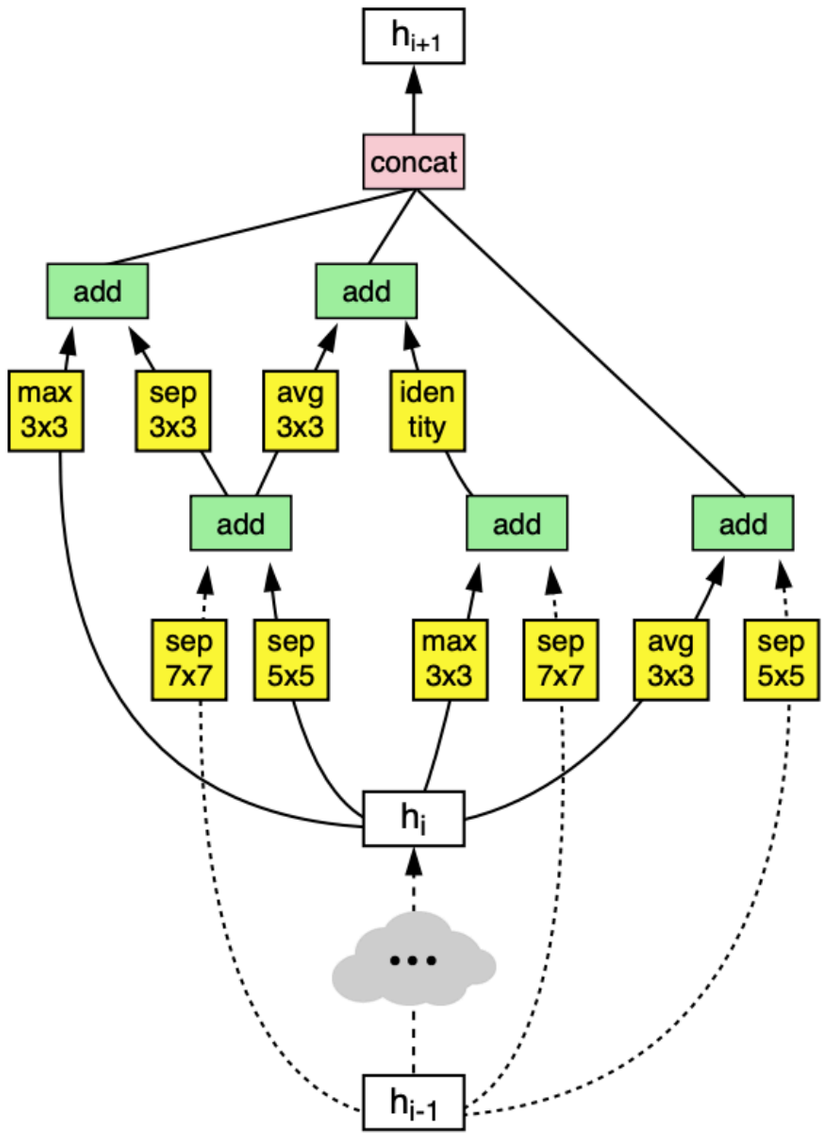
- Two cells
  - Normal cell
  - Reduction cell
- Actual architecture is predefined by cell repetitions
  - Only few hyper parameters
  - Architecture can be scaled easily



Cell generation - cell content

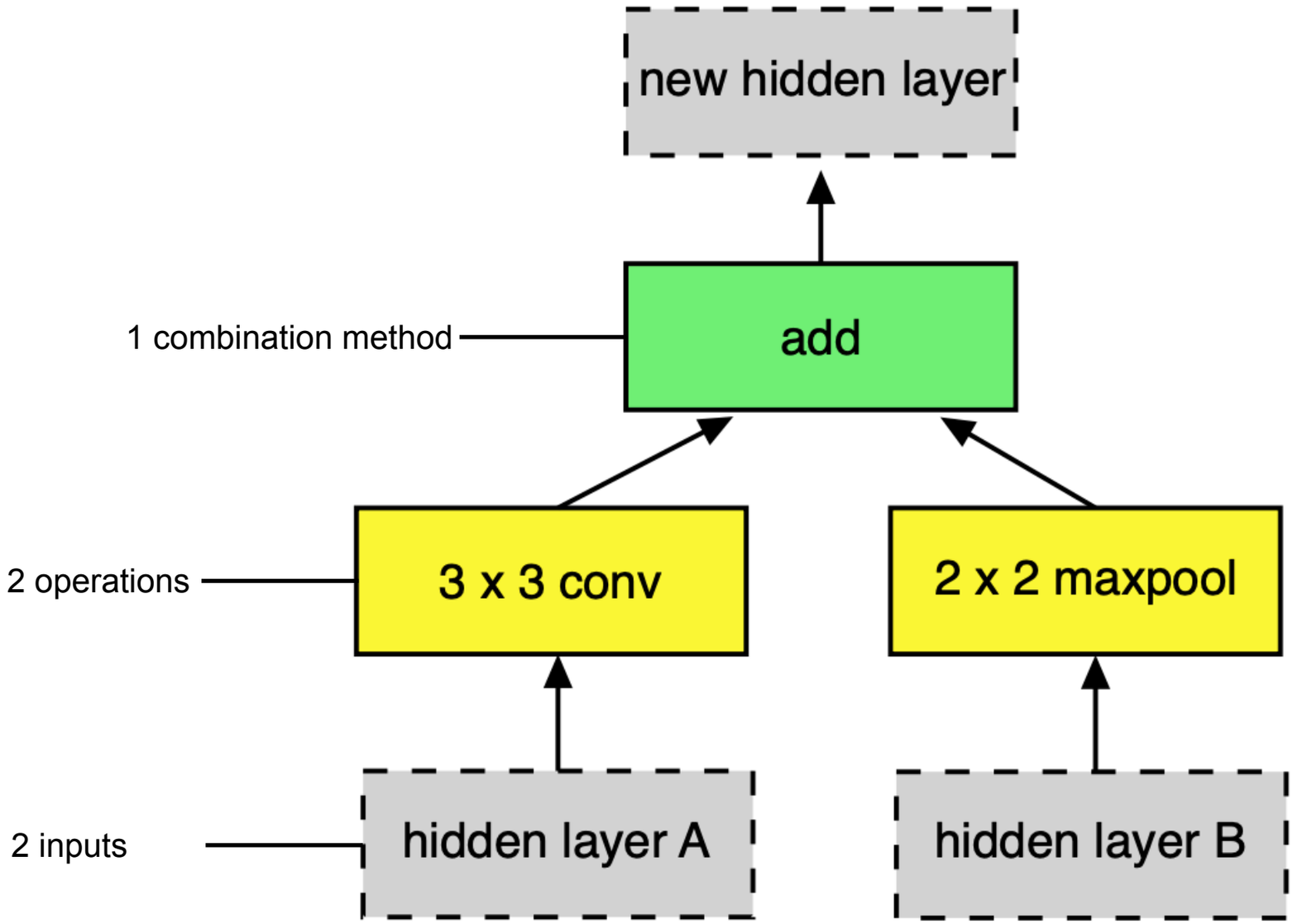


Normal Cell



Reduction Cell

Cell generation - cell content



1 Block = 5 selections

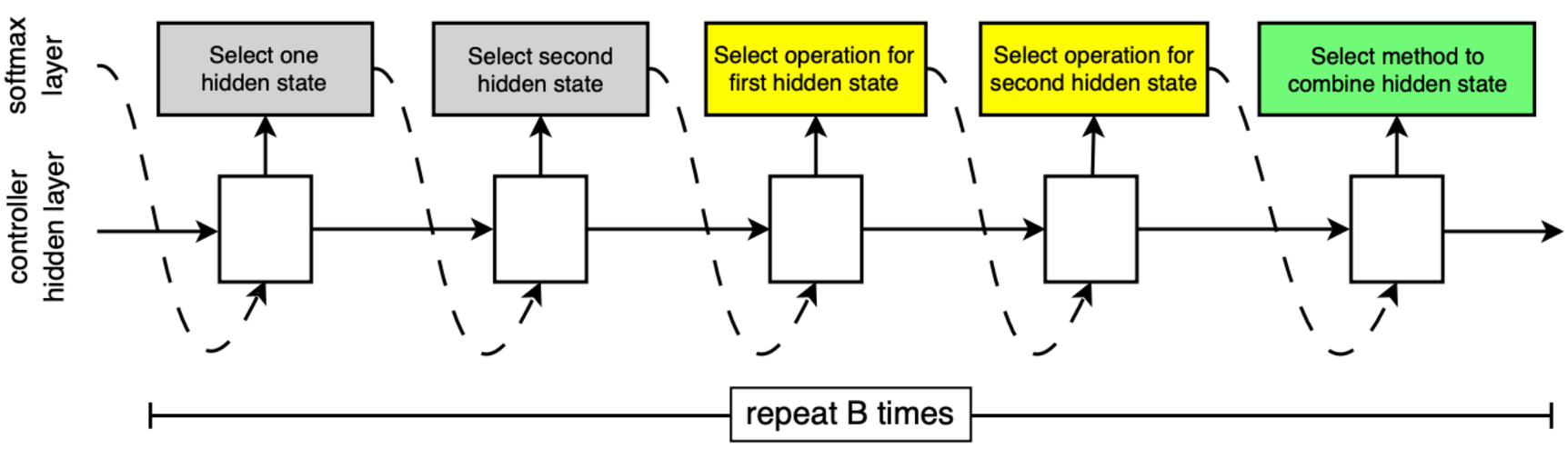
Cell generation - cell content

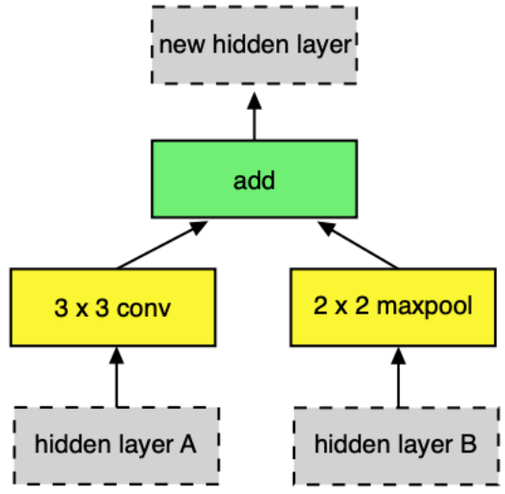
- *B* blocks
  - Each block consists of 5 selections
    - (2) Select two inputs
    - (2) Select one function for each input
      - Apply function to input
    - (1) Combine both inputs
      - element wise addition
      - concatenation
- Blocks are size invariant
  - Stride and padding are selected accordingly
- All unused hidden states are concatenated to output of cell
  - 1x1 convolutions are applied fit number of filters
- Number of filters is doubled in reduction cell

- identity
- 1x7 then 7x1 convolution
- 3x3 average pooling
- 5x5 max pooling
- 1x1 convolution
- 3x3 depthwise-separable conv
- 7x7 depthwise-separable conv
- 1x3 then 3x1 convolution
- 3x3 dilated convolution
- 3x3 max pooling
- 7x7 max pooling
- 3x3 convolution
- 5x5 depthwise-seperable conv

Cell generation - RNN

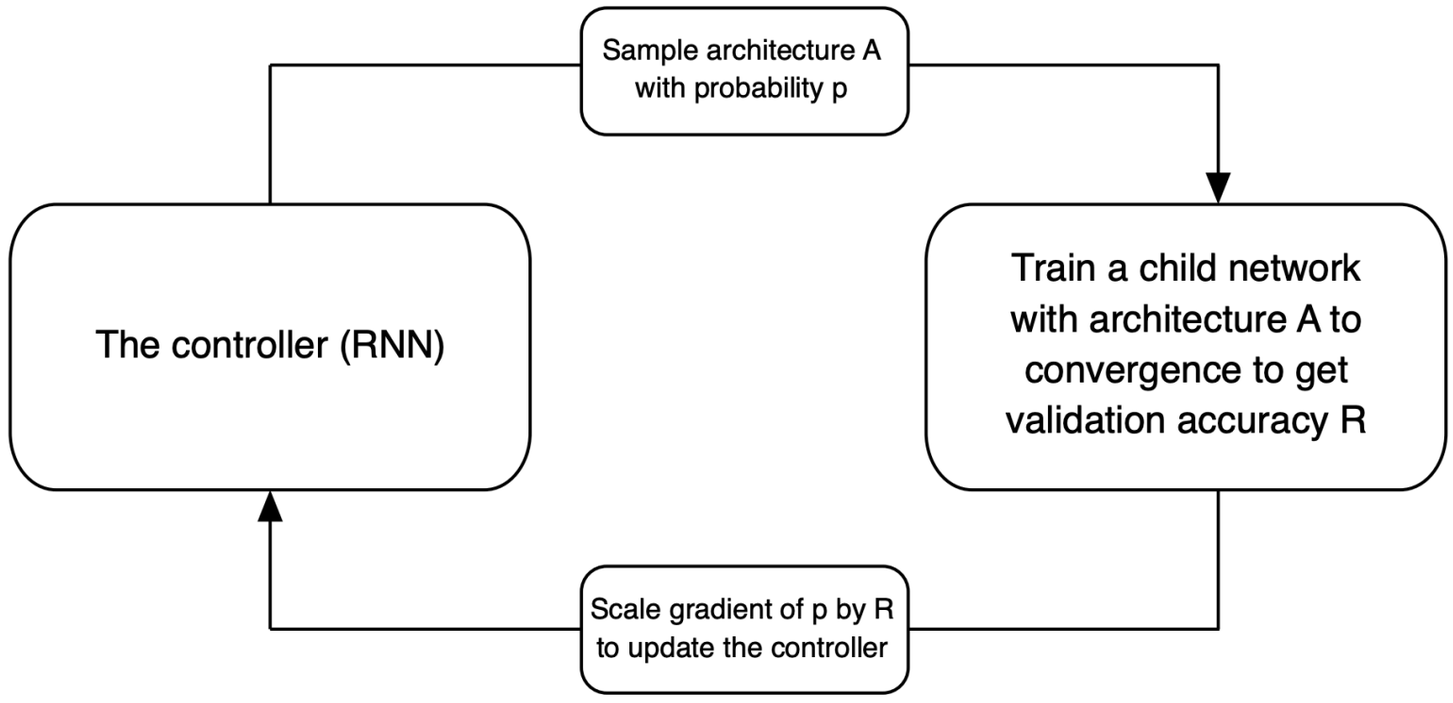
- One layer LSTM network
- Predict each block
- Two cells separate



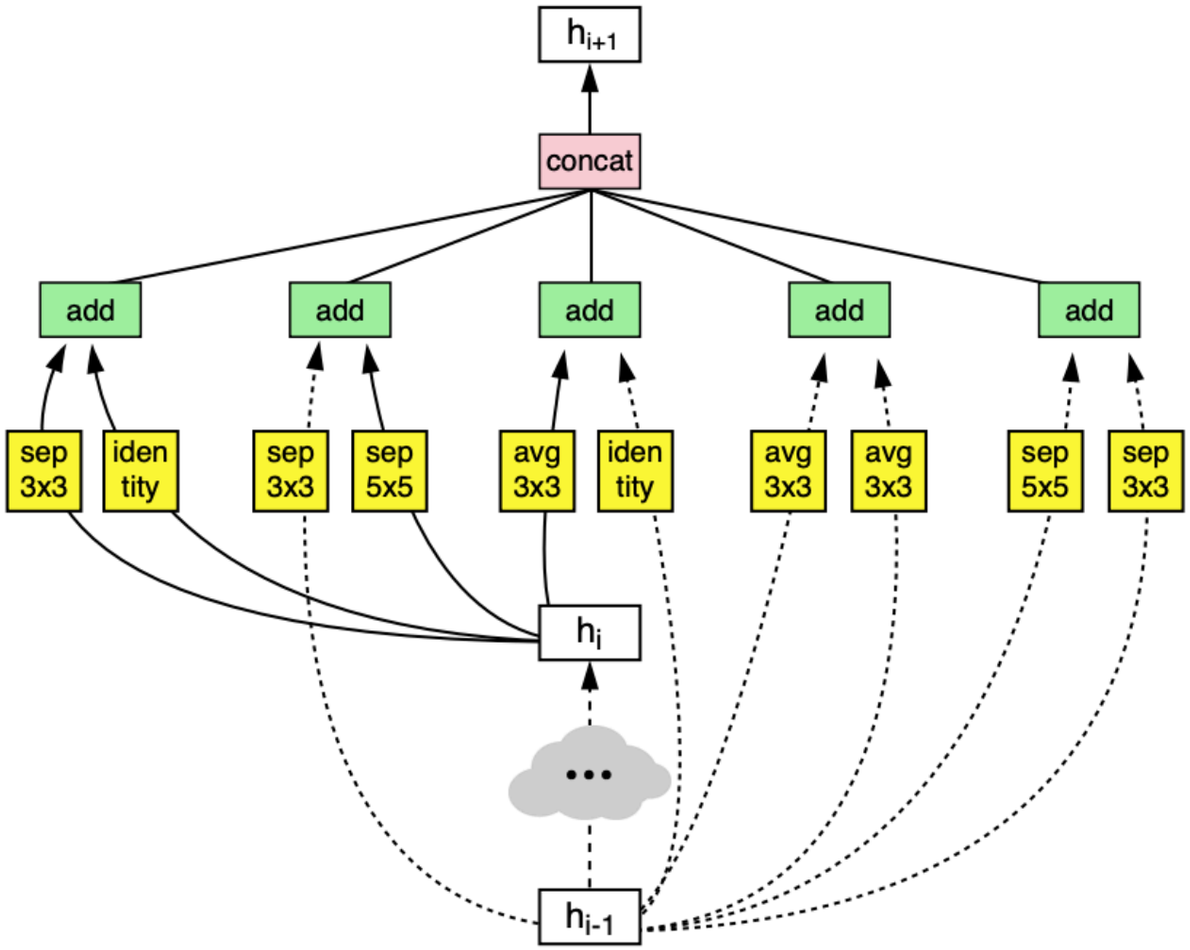


Cell generation - RNN training loop

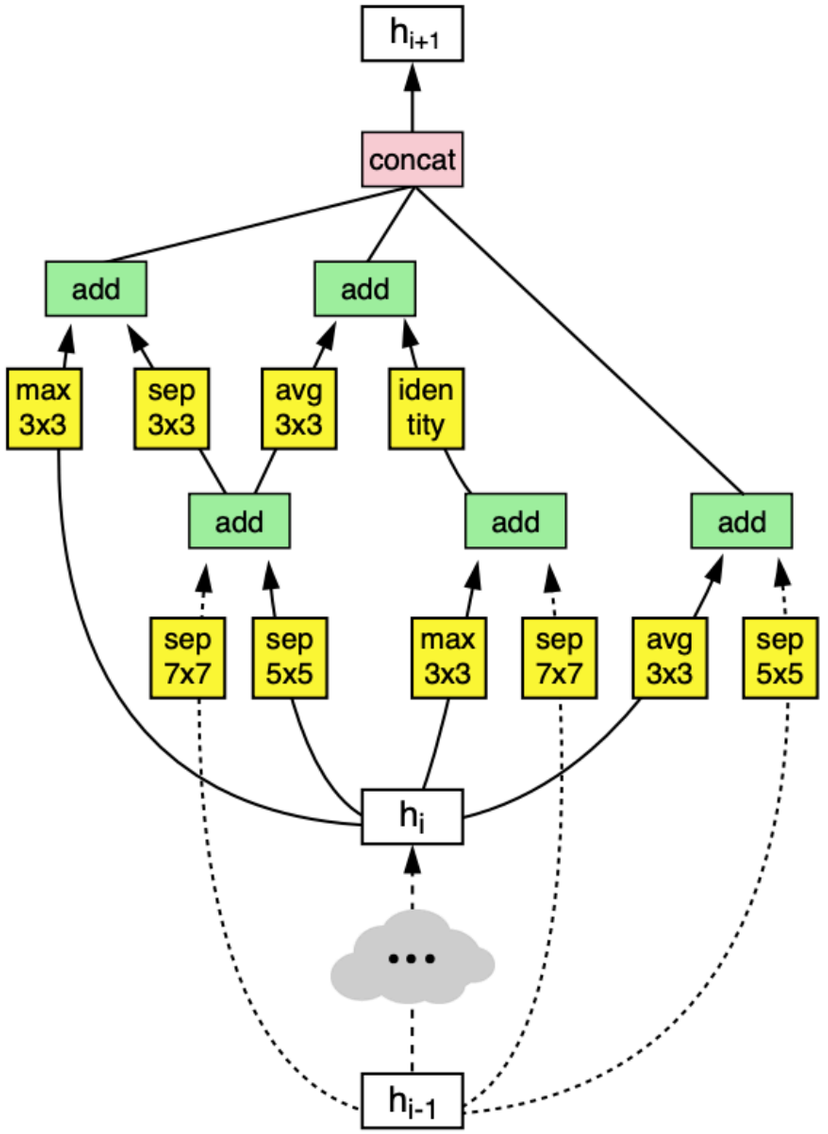
- Similar to NAS
- Predict cells
- Train resulting architecture on CIFAR10
- Scale probability of cell selection with accuracy
- Update model weights



Resulting cells



Normal Cell



Reduction Cell

Results

- State of the art performance in 2017
  - On imagenet
  - Mobile (few parameters)
  - Object detection
- RL vs random search

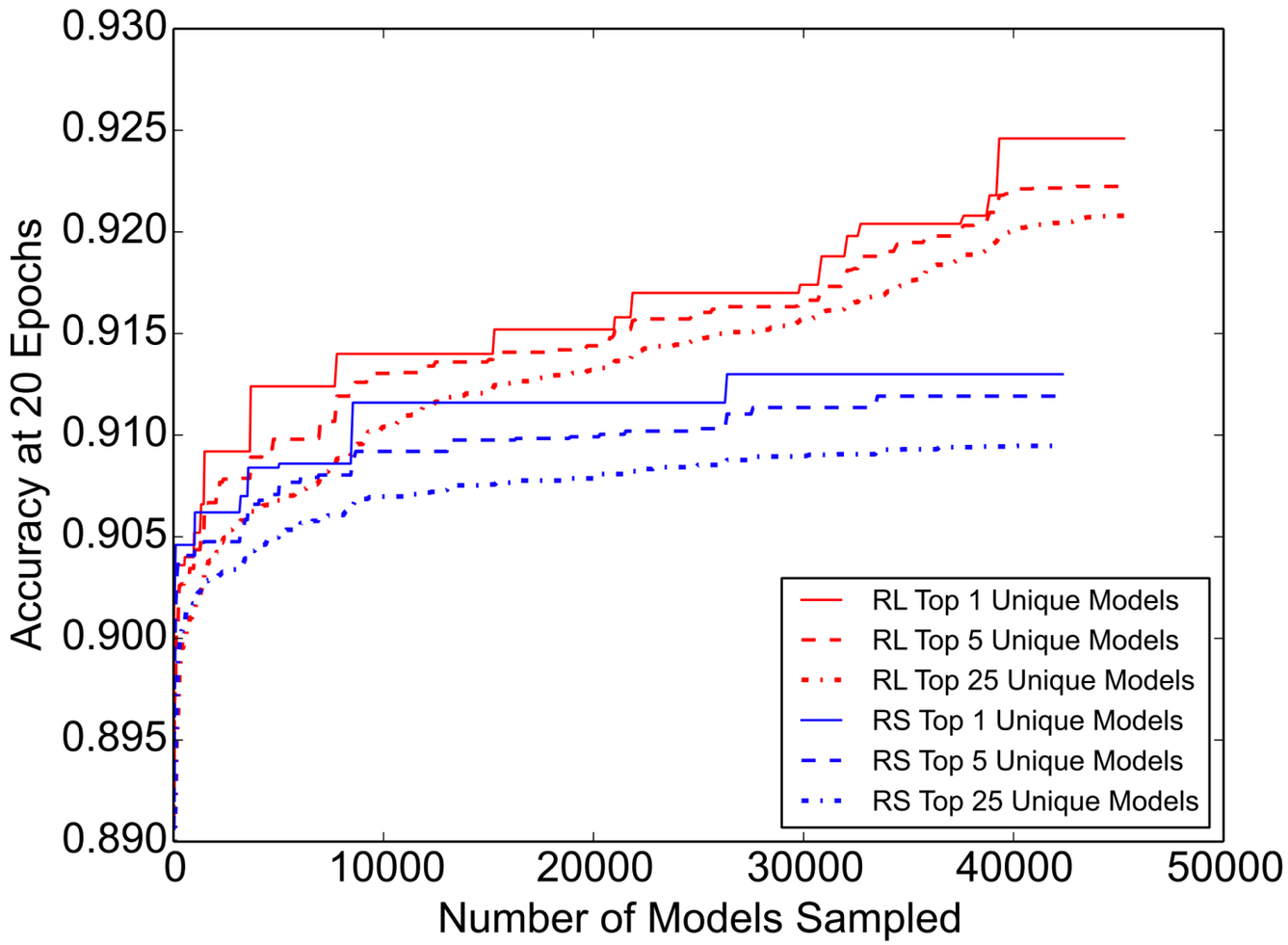
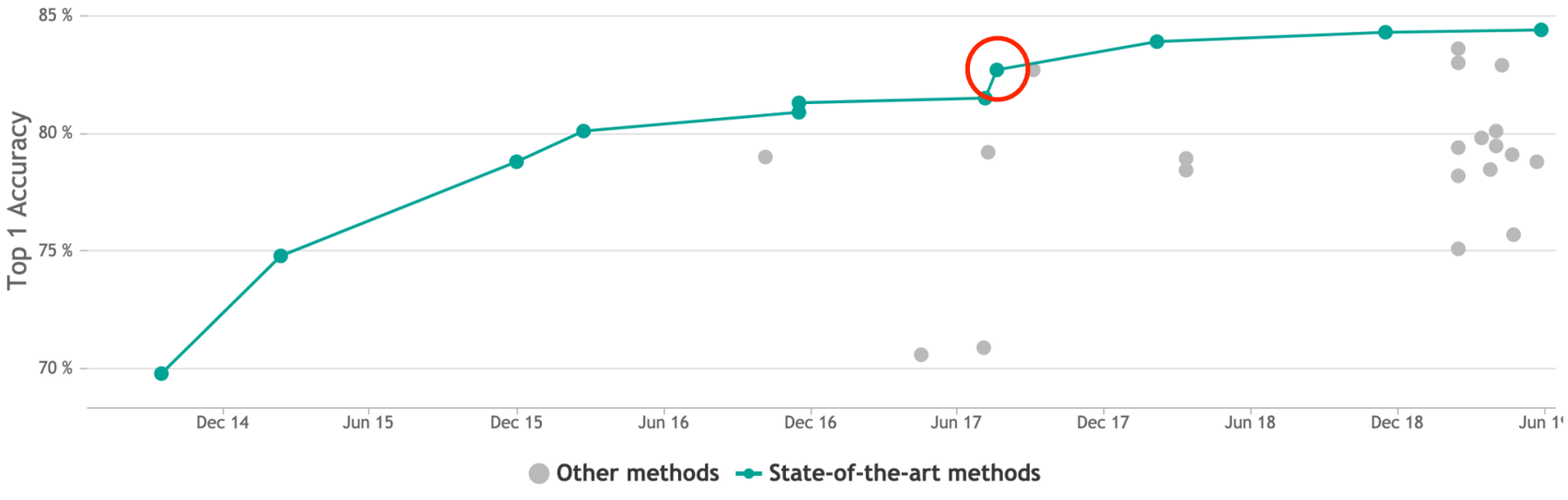


Image Classification on ImageNet



Thank you for your attention!