Model Acceleration

Work in Hardware Acceleration

- Conv 层:每次从bram取出5*5个weigths参数和1个bias参数,然后对 于feature map 每次从bram中取出一行数据计算一次卷积,记录在linebuffer 中,存取5行之后,然后再回到第2行再次计算.

- Linear: 每次取出连续20个数据以及其对应的权值,存储在linebuffer中,然后一直计算完成输出的第一个点,接着再计算输出的第二个点,一直到计算完成

- Bram: 规定conv层的权重每次存取25个8位bit,全连接层的存取每次为20个8位bit.

- 定点化: 采用2个位表示整数部分6个位表示小数部分,溢出就直接让其溢出即可

Work in Hardware Acceleration

- LeNet运行pipleline

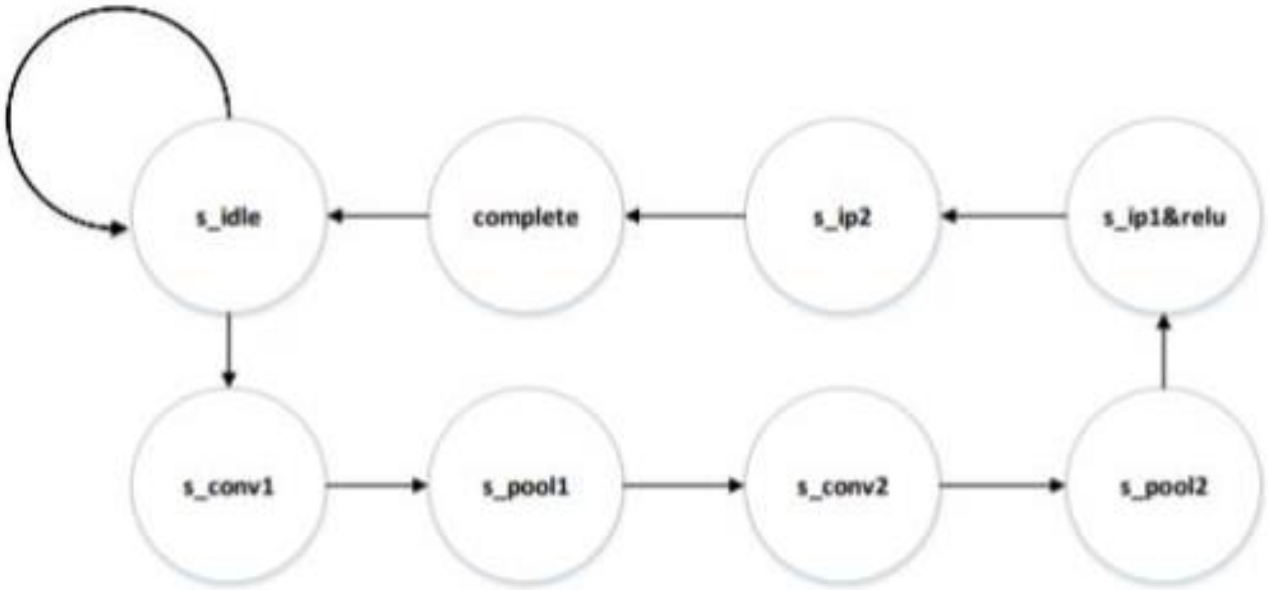


图 5：顶层状态机

Work in Hardware Acceleration

- 使用Verilog HDL实现卷积层、池层、全连接层，

```
always@* begin
    case(state)
        CONV_IDLE : begin
            if(mode == CONV1) n_state = PREPARING;
            else n_state = CONV_IDLE;
        end
        PREPARING : n_state = LOAD_CONV1_BIAS;
        LOAD_CONV1_BIAS : n_state = LOAD_CONV1_WEIGHT;
        LOAD_CONV1_WEIGHT : n_state = LOAD_CONV1_WRITE_DATA;
        LOAD_CONV1_WRITE_DATA : begin
            if(conv1_done) n_state = CONV1_FINISH;
            else n_state = (conv1_weight_done)? LOAD_CONV1_WEIGHT : LOAD_CONV1_WRITE_DATA;
        end

        CONV1_FINISH : n_state = PREPARING_CONV2_BIAS0;
        PREPARING_CONV2_BIAS0 : n_state = LOAD_CONV2_BIAS0;
        LOAD_CONV2_BIAS0 : n_state = PREPARING_CONV2_BIAS1;
        PREPARING_CONV2_BIAS1 : n_state = LOAD_CONV2_BIAS1;
        LOAD_CONV2_BIAS1 : n_state = LOAD_CONV2_WEIGHT;
        LOAD_CONV2_WEIGHT : n_state = LOAD_CONV2_WRITE_DATA;
        LOAD_CONV2_WRITE_DATA : begin
            if(conv_done) n_state = CONV_FINISH;
            else n_state = (conv2_weight_done) ? LOAD_CONV2_WEIGHT : LOAD_CONV2_WRITE_DATA;
        end
        CONV_FINISH : n_state = CONV_IDLE;
        default : n_state = CONV_IDLE;
    endcase
end
```

Research in Algorithm Acceleration

- Taylor-Pruning Model
- ResNet50 Channel-Pruning
- BN-Pruning Model
- Attention Transfer Mimic
- Net-Adapt

Taylor-Pruning Model

- Implement Paper **Pruning Convolutional Neural Networks for Resource Efficient Inference**(ICLR2017) on VGG, ResNet, Inception Strutureand Classification, Pose Estimation, Detection Tasks.

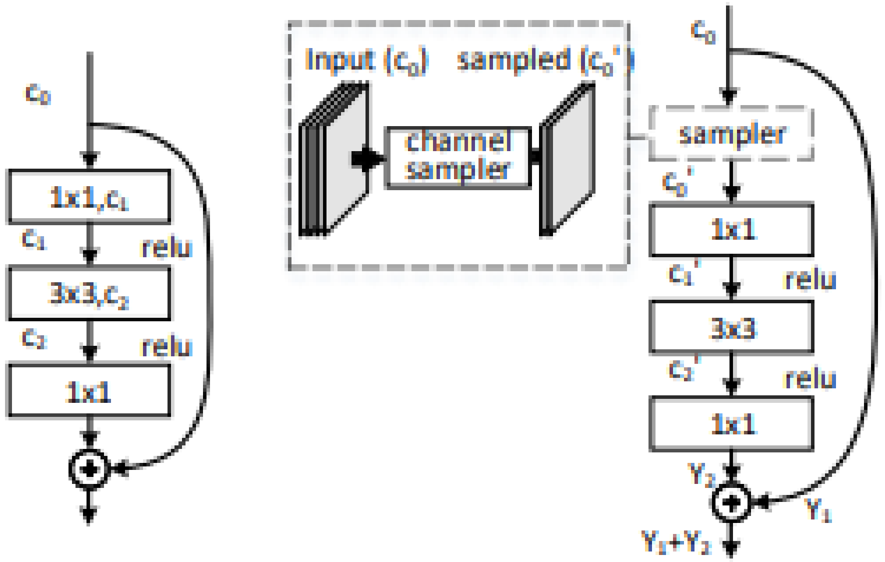
- My contribution: I use the flops constrain and weight reconstruction to enhance the result from 83 to 85.304(Top5@Accuracy) on ImageNet. The algorithm has been used in products developed by SenseTime

$$\left|\Delta\mathcal{C}(h_i)\right| = \left|\mathcal{C}(\mathcal{D}, h_i = 0) - \mathcal{C}(\mathcal{D}, h_i)\right|,$$

ResNet50 Channel-Pruning

- Implement for Paper **Channel Pruning for Accelerating Very Deep Neural Networks** (ICCV2017) on Resnet50(ImageNet).

- My contribution: I achieve 2x acceleration and Top-5 Accuracy reduces 3.11(from 92.98 to 89.87).



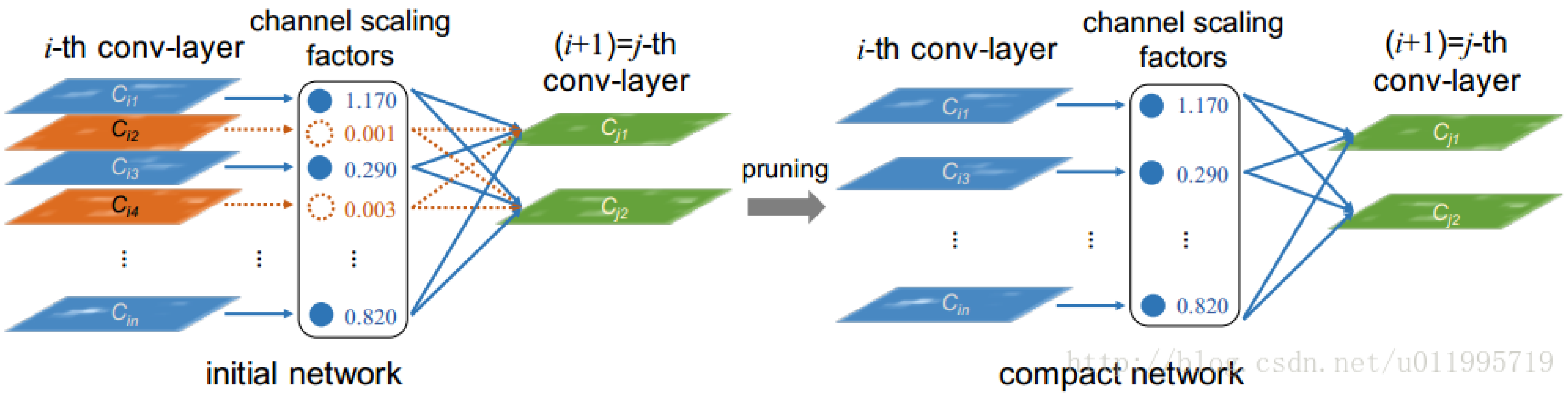
BN-Pruning Model

- Implement for Paper Learning Efficient Convolutional Networks through Network Slimming(ICCV2017) on VGG, ResNet, Inception Structure.

- My Contribution:

1 I find that this way is sensitive to the hyper parameters.

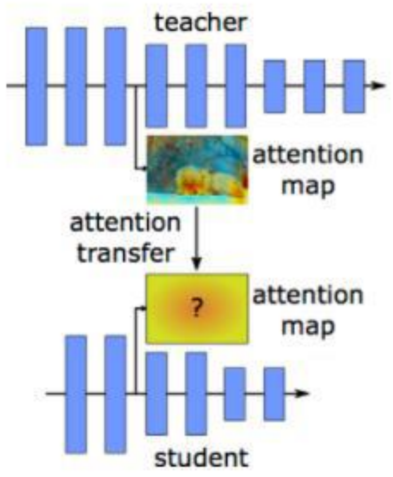
2 For detection, key point tasks, pretrained model as backbone is prone to generate better model, and BN-Pruning which relys on BatchNorm Layer is a way naturally trained from scratch model



Attention Transfer Mimic

- Implement for Paper **Improving Convolutional Networks via Attention Transfer** (ICLR 2017)on Classification, Detection Tasks

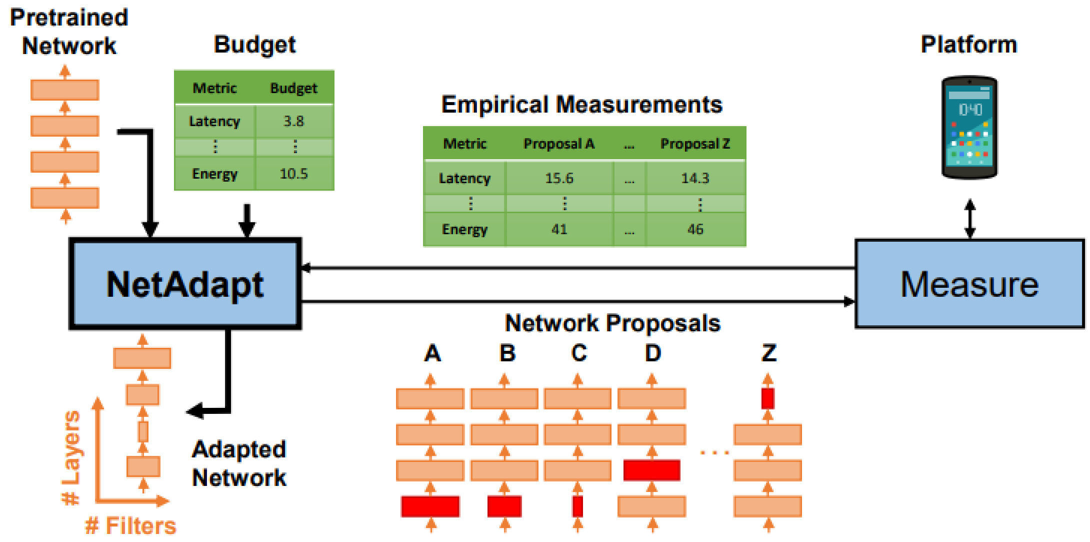
- My contribution: Transfer layer can remedy the gap betweendteacher model. For example, detection task(faster-rcnnresnet18 pascal VOC), the result of model trained by naive L2 loss as mimic loss(map 58.7) is lower 1.3 than that of model trained by L2 loss adding transfer layer.



Net-Adapt

- Pruning model with a greedy search strategy. Implement for Paper **NetAdapt: Platform-Aware Neural Network Adaptation for Mobile Applications**

- My contribution: 1 Utilize Multi-process distribute training to finetune generated child network 2 build offline time lookup table



Pruning with Hints: A framework for model acceleration