NAS-VQA summary

DANG ANH CHUONG

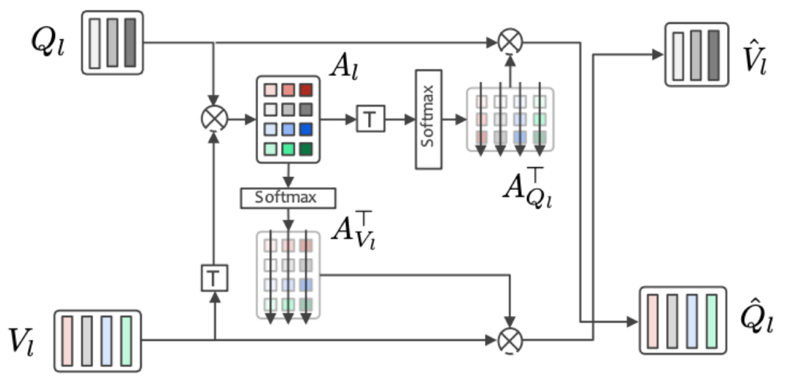
Outline:

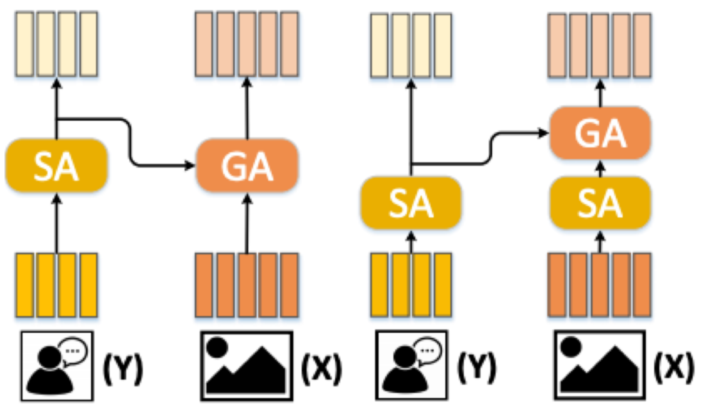
I. NAS-VQA project summary
   1. First attempt
   2. Second attempt
   3. Third attempt
   4. New publish on NAS-VQA vs ours
   5. Brief results summary

I. NAS-VQA summary

1. First attempt

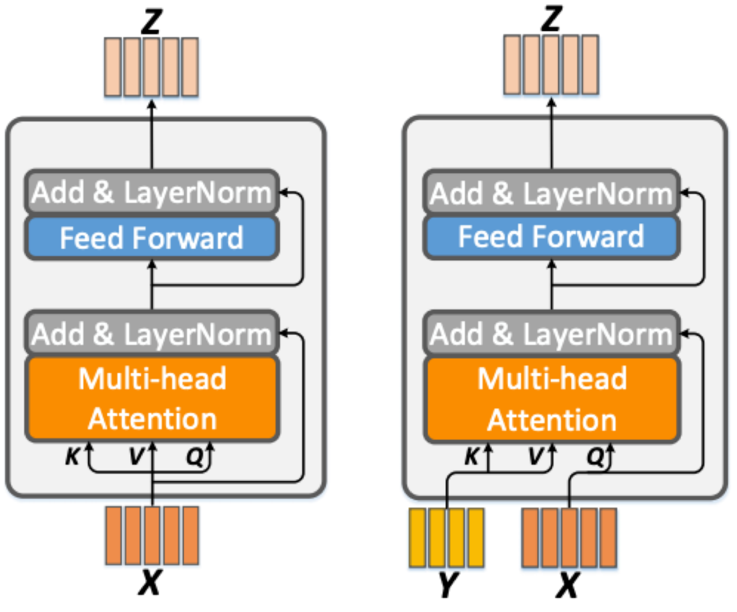
- The first attempt for applying NAS in VQA task
  - ➢ Utilizing SNAS search strategy with Gumbel Softmax, note: by changing this does not improve performance :'(
  - ➢ Utilizing MCAN & DenseCoAttention model
    - Operations: SA, SGA, modified CoA
  - ➢ Operation pool: SA-SA, ID-SGA, SGA-ID, SA-ID, ID-SA … permutations of pair-operation
  - ➢ Search whole attention network topology
    - ✓ Best performance on validation set: **67.08%**
    - ✓ Not bad but not yet giving good result





(b) SA(Y)-GA(X,Y)

(c) SA(Y)-SGA(X,Y)



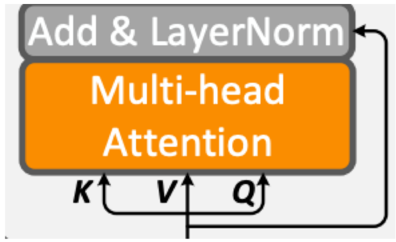
(a) Self-Attention (SA)

(b) Guided-Attention (GA)

2. Second attempt

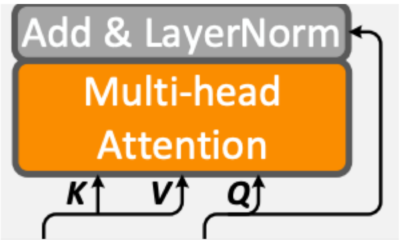
- Rethinking about why DARTS yields improvement in classification task
  - ➢ Keeping search strategy as SNAS
  - ➢ Maybe it is about search proxy:
    - Which operations?
    - Connection between operations …
  - ➢ Needed for more appropriate operations and connections proxy:
    - ○ Breaking down & utilizing MCAN operations: SA, GA, FF
    - ○ Operation pool: SA, GA, FF, ID (a.k.a: skip-connect), None
    - ○ Adding connection operations from DARTS: Identity, None

Self Attention



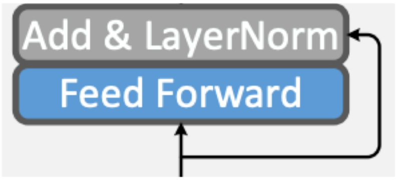
$$Z = LN(X + SA(X)) = LN(X + MHA(X, X, X, 0))$$

Guided Attention



$$Z = LN(X + GA(X)) = LN(X + MHA(X, Y, Y, 0))$$

Feed Forward



$$Z = LN(X + FF(X))$$

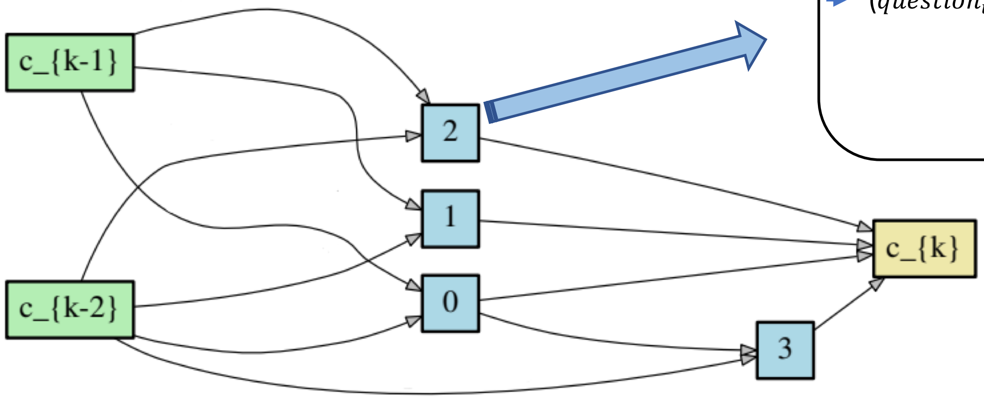
$$FF(X) = FC_{d\to 4d} \to ReLU \to Drop(0.1) \to FC_{4d\to d}$$

Where;
LN: LayerNorm
MHA: multi-head attention

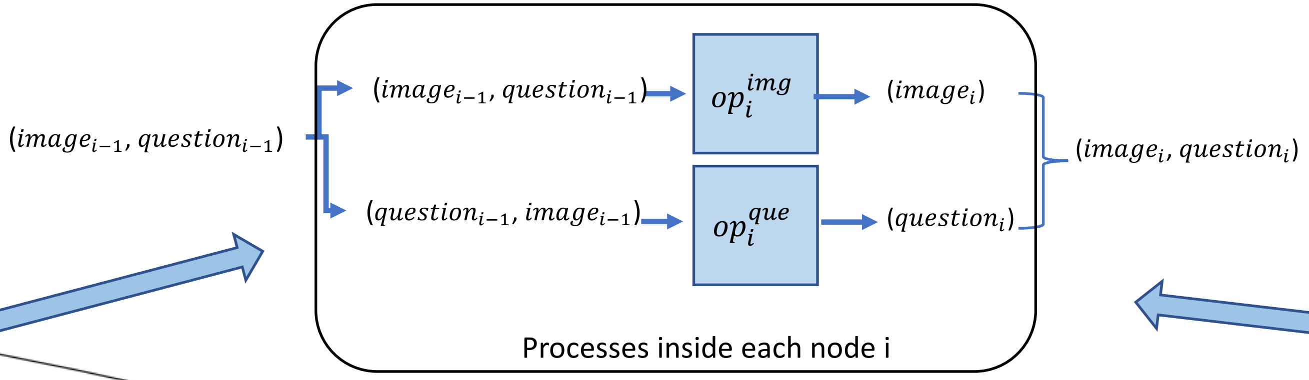
2. Second attempt

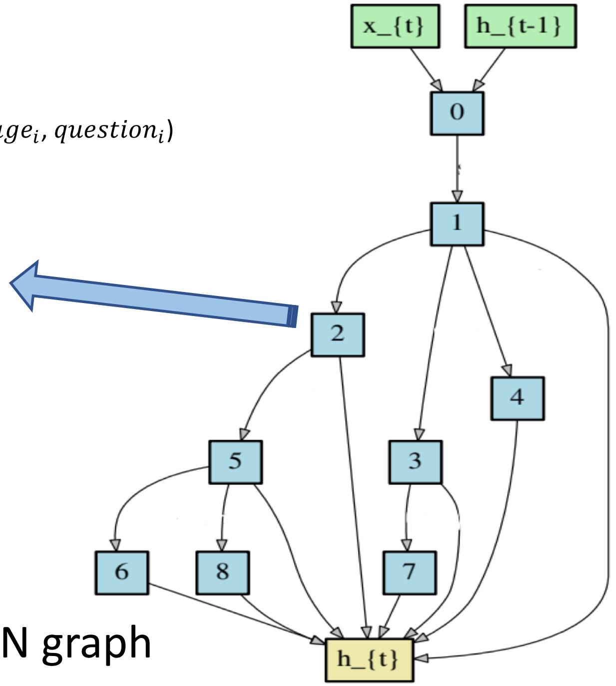
- How about connection proxy?
  - Utilizing two proxy from DARTS (however, both are stack-style connection)
    - Proxy for CNN (best on val set: **66.60%**)
    - Proxy for RNN (best on val set: **66.69%** … improved a bit with way less parameters)

    - In both implementation, I was confused about how to generate output for each cell, after all I took average features from intermediate nodes.
    - Input of each node are 2 features, while output is 1 specific feature (either image or language)



CNN graph

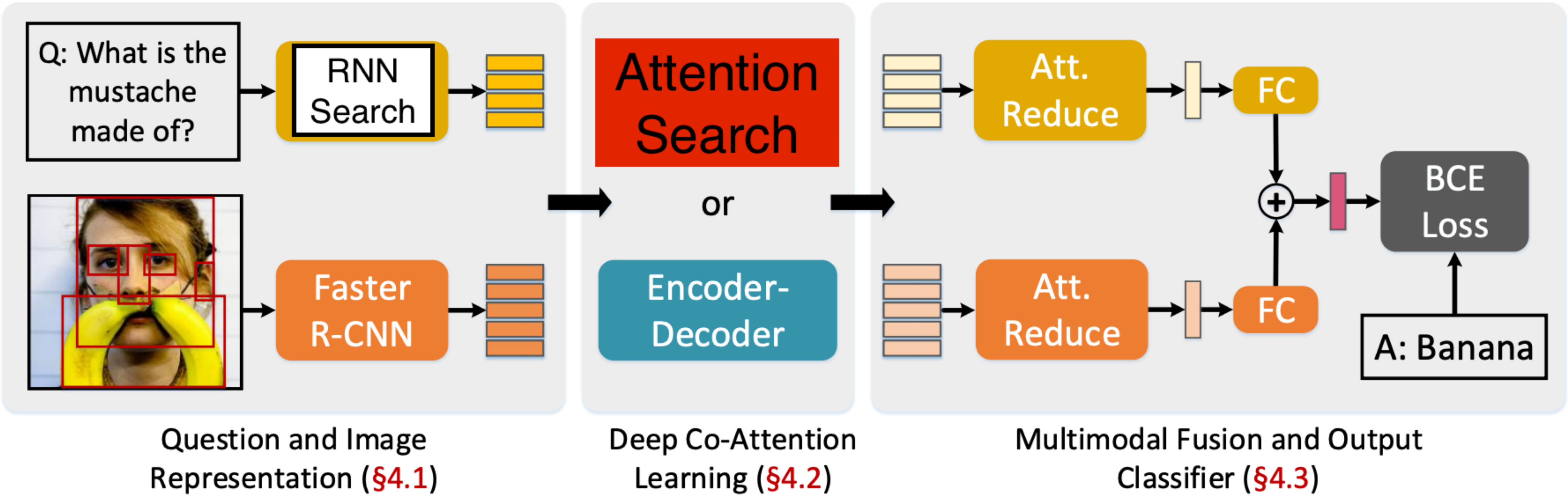




RNN graph

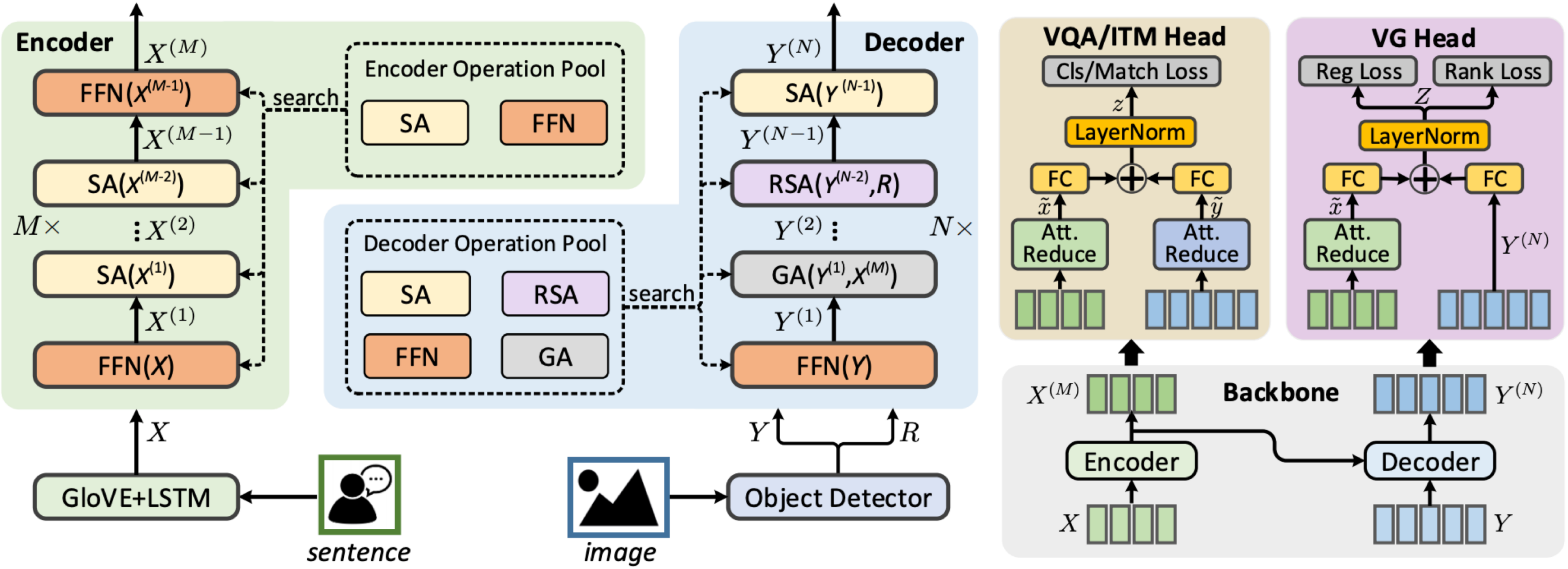
3. Third attempt

- Reconsidering about MCAN model and why it is good
- Language feature may be weak compare to image feature from Faster-RCNN
- Try to implement search on Language feature extractor:
  - ✓ Utilizing DARTS search for RNN model
  - ✓ Initial results:
    - ➢ RNN-search + Attention search = **66.44%** (not yet carefully tuned)
    - ➢ RNN-search + MCAN_ed = **67.03%** ± 0.02% (not yet carefully tuned)



4. New publish and differences in our model

- Last month, the same group of authors of MCAN paper published one paper about NAS on VQA, the solution in the paper somehow similar to my approach
  - By utilizing NAS algorithm, they improve their model performance <u>*0.6%*</u> (from **67.2%** to **67.8%**) very minimal improvement
  - Paper title: Deep Multimodal Neural Architecture Search (a.k.a MMnasNet)

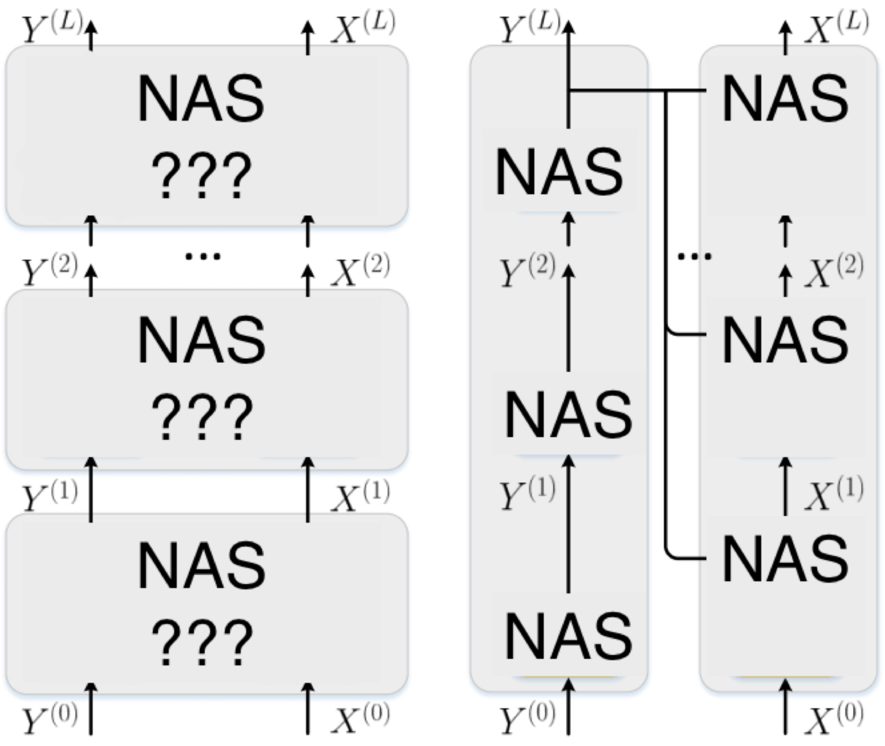


(a) Unified Encoder-Decoder Backbone

(b) Task-specific Heads

4. New publish and differences in our model

- Brief summary about differences between their proposed method and ours
  - ➢ Encoder-Decoder proxy vs Stack approach
  - ➢ Slightly different in operations pool and structure of each operation



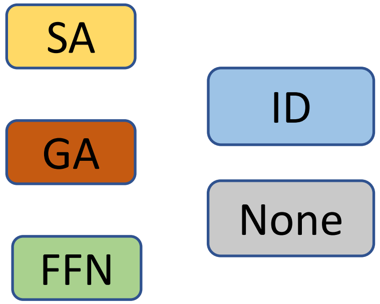
Stack Decoder-Encoder

	Mine	Theirs
SA	$Z = LN(X + SA(X)) = LN(X + MHA(X, X, X, 0))$	Z = SA(X) = MHA(X,X,X, 0)
GA	$Z = LN(X + GA(X)) = LN(X + MHA(X, Y, Y, 0))$	Z = GA(X,Y) = MHA(X,Y,Y, 0)
FFN	$Z = LN(X + FFN(X))$ $FFN(X) = FC_{d \to 4d} \to ReLU \to Drop(0.1) \to FC_{4d \to d}$	Z = FFN(X) FFN(X) = FCd ∘ Drop0.1 ∘ ReLU ∘ FC4d (X)
RSA	N/A	Z = RSA(X, R) = MHA(X,X,X, log(MLP(R) + ϵ)) MLP(R) = ReLU ∘ FC1 ∘ ReLU ∘ FCdh (R)

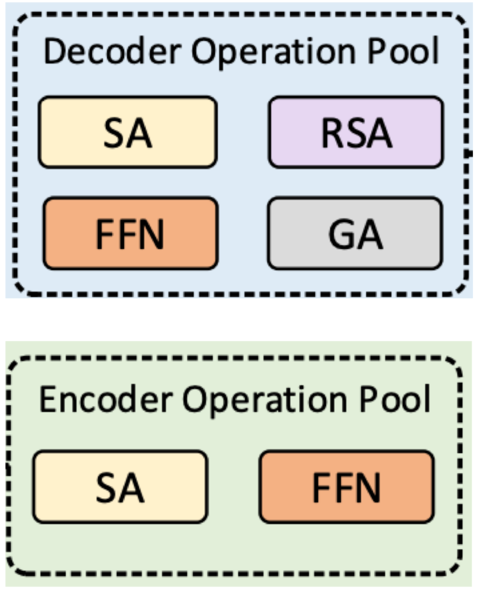
4. New publish and differences in our model

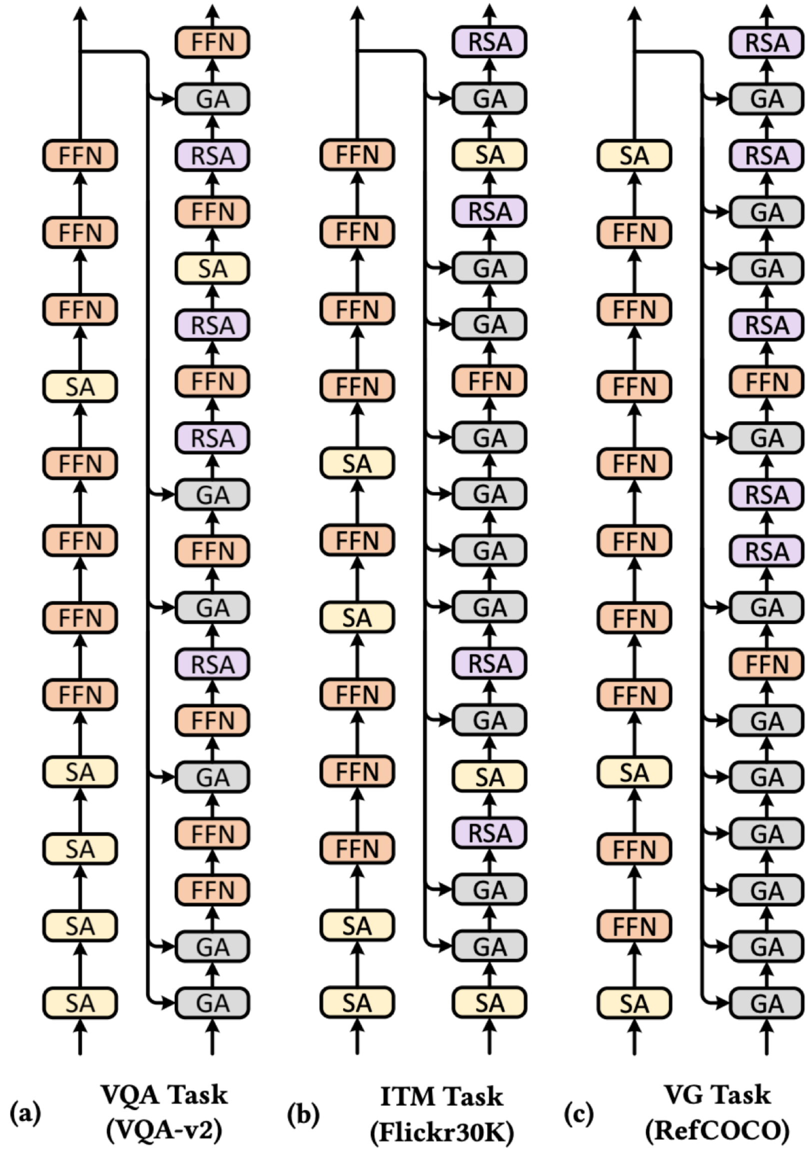
- A bit more details about new publish
    - Specifically, main different is that they have no add_norm layer in their operation primitives comparing to mine
    - They use design of encoder and decoder for their search proxy while I used stackable design
    - Their operation pool are more specific for encoder-decoder while mine is same for both image and language
    - I utilized SNAS search algorithm while they used more optimized one from ProxylessNAS (ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware)

My ops pool



Their ops pool





Result architecture of MMnasNet for each task

5. Brief performance comparison

Model	Best validation accuracy (%)
DCN (paper)	65.50
BAN (paper)	66.04
Ours (2nd)	66.69
Ours (3rd)	67.03
Ours (1st)	67.08
MCAN (paper)	67.20
MMnasNet (paper)	67.80

References:

- Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In Advances in Neural Information Processing Systems (NIPS)
- Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep Modular Co-Attention Networks for Visual Question Answering. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Duy-Kien Nguyen and Takayuki Okatani. 2018.Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Zhou Yu, Yuhao Cui, Jun Yu, Meng Wang, Dacheng Tao, and Qi Tian. 2020. Deep Multimodal Neural Architecture Search. In Proceedings of ACM Conference (ACM).