Image Understanding by Captioning with Differentiable Network Architecture Search

Zeting Luan[1] Ziyan Zhu[2]

[1]Department of Electrical and Computer Engineering
University of California, San Diego

[2]Department of Mathematics
University of California, San Diego

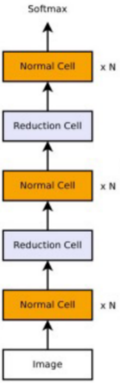
July 12, 2023

Overview

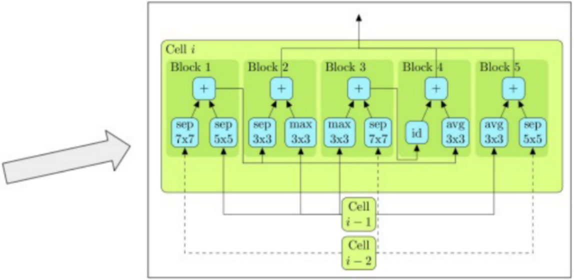
Background and Motivations

- **Image Captioning.** The process of generating textual description of an image based on the objects and actions in the image

- **Neural Architecture Search (NAS).** [1][2] [3][4]



CIFAR10 Architecture



NASNet
Zoph et al. (2018)

Background and Motivations

- **Learning from Mistakes (LFM).** [5]

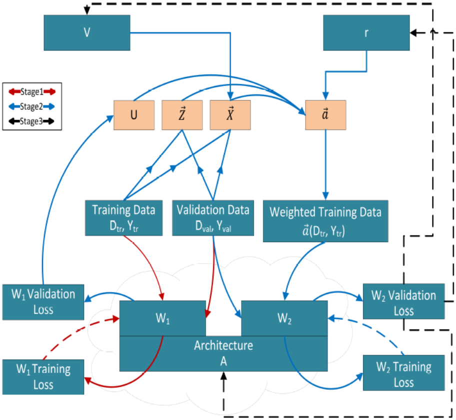


Figure: LFM : A multi-level optimization framework.

Three-Stage Learning Framework

- **Stage I.** An encoder-decoder network model is trained to create image captions.

$$E^*(A), F^*(A) = \min_{E,F} \; L(E, A, F, D^{tr})$$

where E, F denote the network weights of the encoder and the decoder correspondingly, and A denotes the architecture of the encoder.

Three-Stage Learning Framework

- **Stage I.** An encoder-decoder network model is trained to create image captions.

$$E^*(A), F^*(A) = \min_{E,F} \; L(E, A, F, D^{tr})$$

where E, F denote the network weights of the encoder and the decoder correspondingly, and A denotes the architecture of the encoder.

- **Stage II.** The trained encoder and decoder generates a pseudo image captioning dataset from some unlabeled images, which will be used to train an image captioning model.

$$W^*(E^*(A), F^*(A)) = \min_{W} \; L(W, U, E^*(A), F^*(A))$$

where U denotes some unlabeled images, which will be used to generate pseudo dataset with trained model from stage I, and W is the weights of image captioning model.

Three-Stage Learning Framework

- **Stage I.** An encoder-decoder network model is trained to create image captions.

$$E^*(A), F^*(A) = \min_{E,F} \; L(E, A, F, D^{tr})$$

where E, F denote the network weights of the encoder and the decoder correspondingly, and A denotes the architecture of the encoder.

- **Stage II.** The trained encoder and decoder generates a pseudo image captioning dataset from some unlabeled images, which will be used to train an image captioning model.

$$W^*(E^*(A), F^*(A)) = \min_{W} \; L(W, U, E^*(A), F^*(A))$$

where U denotes some unlabeled images, which will be used to generate pseudo dataset with trained model from stage I, and W is the weights of image captioning model.

- **Stage III.** The architecture A will be learned by minimizing the loss on validation set of image captioning model.

Three-Stage Learning Framework

The optimization problem can be summarized as below:

$$\begin{aligned} \min_{A} \quad & L(W^*(E^*(A), F^*(A)), D^{val}) \\ \text{s.t.} \quad & W^*(E^*(A), F^*(A)) = \min_{W} \; L(W, U, E^*(A), F^*(A)) \\ & E^*(A), F^*(A) = \min_{E,F} \; L(E, F, A, D^{tr}) \end{aligned}$$

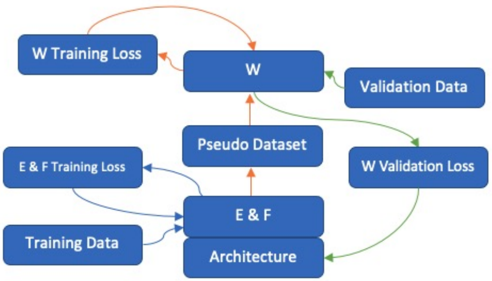


Figure: The Overall Process Flow.

Optimization Algorithm

- **Stage I** & **II.** Approximation with one-step gradient descent:

$$
\begin{aligned}
E^*(A) &\approx E' = E - \eta_e \nabla_E L(E, F, A, D^{tr}) \\
F^*(A) &\approx F' = F - \eta_f \nabla_F L(E, F, A, D^{tr}) \\
W^*(E^*(A), F^*(A)) &\approx W' = W - \eta_w \nabla_W L(W, U, E', F')
\end{aligned}
$$

Optimization Algorithm

- **Stage I & II.** Approximation with one-step gradient descent:

$$E^*(A) \approx E' = E - \eta_e \nabla_E L(E, F, A, D^{tr})$$
$$F^*(A) \approx F' = F - \eta_f \nabla_F L(E, F, A, D^{tr})$$
$$W^*(E^*(A), F^*(A)) \approx W' = W - \eta_w \nabla_W L(W, U, E', F')$$

- **Stage III.** Update the architecture A by gradient descent:

$$A \leftarrow A - \eta_a \nabla_A L(W', D^{val})$$

where the gradient $\nabla_A L$ can be obtained by chain rule as:

$$\begin{aligned}\nabla_A L(W', D^{val}) =& \eta_e \eta_w \nabla^2_{A,E} L(E, A, F, D^{tr}) \nabla^2_{E',W} L(W, U, E', F') \nabla_{W'} L(W', D^{val}) \\ &+ \eta_f \eta_w \nabla^2_{A,F} L(E, A, F, D^{tr}) \nabla^2_{F',W} L(W, U, E', F') \nabla_{W'} L(W', D^{val})\end{aligned}$$

Gradient Approximation

Recall the formula for the gradient $\nabla_A L$:

$$\begin{aligned}\nabla_A L(W', D^{val}) =& \eta_e \eta_w \nabla^2_{A,E} L(E, A, F, D^{tr}) \nabla^2_{E',W} L(W, U, E', F') \nabla_{W'} L(W', D^{val}) \\ &+ \eta_f \eta_w \nabla^2_{A,F} L(E, A, F, D^{tr}) \nabla^2_{F',W} L(W, U, E', F') \nabla_{W'} L(W', D^{val})\end{aligned}$$

Approximate matrix-vector product by finite difference for the first term:

$$\nabla^2_{E',W} L(W, U, E', F') \nabla_{W'} L(W', D^{val}) \approx \frac{\nabla_{E'} L(W^+, U, E', F') - \nabla_{E'} L(W^-, U, E', F')}{2\epsilon_w}$$

where $W^{\pm} = W' \pm \epsilon_w \nabla_{W'} L(W', D^{val})$, and ϵ_w is a small scale.

$$\nabla^2_{A,E} L(E, A, F, D^{tr}) \nabla_{E'} L(W^{\pm}, U, E', F') \approx \frac{\nabla_A L(E^+, A, F, D^{tr}) - \nabla_A L(E^-, A, F, D^{tr})}{2\epsilon_e}$$

where $E^{\pm} = E' \pm \epsilon_e \nabla_{E'} L(W^{\pm}, U, E', F')$, and ϵ_e is a small scale.

Gradient Approximation

Recall the formula for the gradient $\nabla_A L$:

$$\begin{aligned}\nabla_A L(W', D^{val}) =& \eta_e \eta_w \nabla^2_{A,E} L(E, A, F, D^{tr}) \nabla^2_{E',W} L(W, U, E', F') \nabla_{W'} L(W', D^{val}) \\ &+ \eta_f \eta_w \nabla^2_{A,F} L(E, A, F, D^{tr}) \nabla^2_{F',W} L(W, U, E', F') \nabla_{W'} L(W', D^{val})\end{aligned}$$

Approximate matrix-vector product by finite difference for the second term:

$$\nabla^2_{F',W} L(W, U, E', F') \nabla_{W'} L(W', D^{val}) \approx \frac{\nabla_{F'} L(W^+, U, E', F') - \nabla_{F'} L(W^-, U, E', F')}{2\epsilon_w}$$

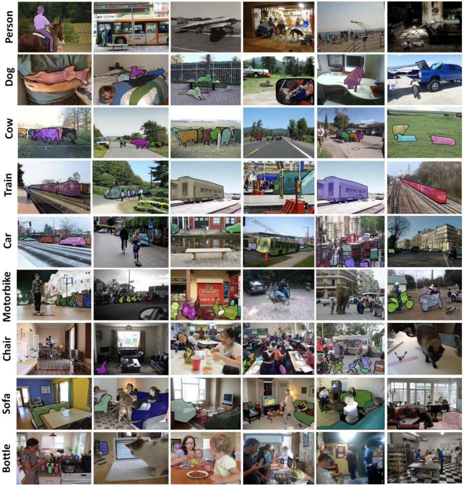
where $W^{\pm} = W' \pm \epsilon_w \nabla_{W'} L(W', D^{val})$, and ϵ_w is a small scale.

$$\nabla^2_{A,F} L(E, A, F, D^{tr}) \nabla_{F'} L(W^{\pm}, U, E', F') \approx \frac{\nabla_A L(E, A, F^+, D^{tr}) - \nabla_A L(E, A, F^-, D^{tr})}{2\epsilon_f}$$

where $F^{\pm} = F' \pm \epsilon_f \nabla_{F'} L(W^{\pm}, U, E', F')$, and ϵ_f is a small scale.

Dataset

- **MSCOCO 2015** [6]



Discussion

- **Summary**

  Three-stage learning framework

  NAS

  LFM

- **Limitations**

  Memory

References I

Barret Zoph and Quoc V Le.
Neural architecture search with reinforcement learning.
arXiv preprint arXiv:1611.01578, 2016.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le.
Learning transferable architectures for scalable image recognition.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean.
Efficient neural architecture search via parameters sharing.
In *International Conference on Machine Learning*, pages 4095–4104. PMLR, 2018.

Kenneth O Stanley and Risto Miikkulainen.
Evolving neural networks through augmenting topologies.
Evolutionary computation, 10(2):99–127, 2002.

References II

- Bhanu Garg, Li Zhang, Pradyumna Sridhara, Ramtin Hosseini, Eric Xing, and Pengtao Xie.
  Learning from mistakes – a framework for neural architecture search, 2022.
- Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick.
  Microsoft coco captions: Data collection and evaluation server, 2015.

The End
Any Questions?