Deep Learning

Midterm Assignment: Image Captioning Solution

P.L.D. Tien (520K0220)

Ton Duc Thang University

March 10, 2023

Outline

What is this?

An image captioning system, using ResNet50 and VGG19, both of with them attention. This submission uses Flikcr8K[1] for it's dataset.

Note

The snippets of code are coming from the original notebook. It's advised to look at it for a better understanding.

[1] https://github.com/jbrownlee/Datasets/releases/tag/Flickr8k

What is image captioning?

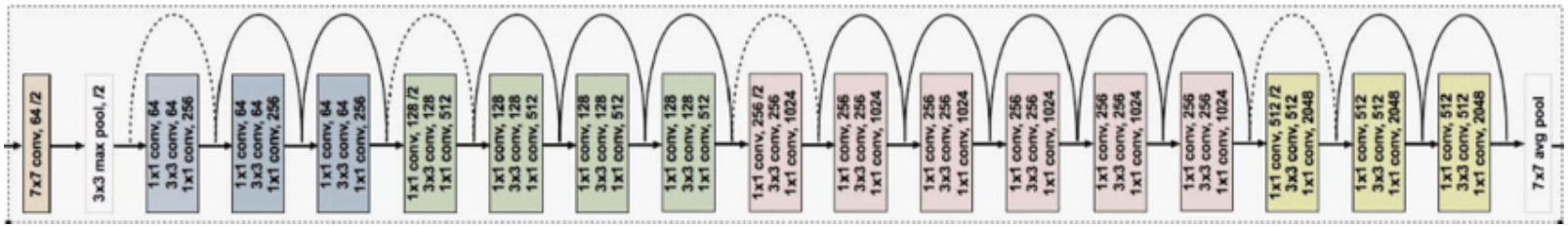
- A process of generating natural language descriptions for images.
- It has been successful in making accurate and meaningful captions.
- Applications can include:
  - Image search engines
  - Healthcare (Things like X-rays, MRI and CT scans)
  - Enable machines to understand and describe visual content.

Context

- Image captioning has been a thing in the 1960s and 1970s
- Basic objects and shapes were done.
- 2000s: Natural language descriptions considered.
- 2014: Show and Tell
  - Used CNN and LSTM.
- Also included deep learning models and large-scale image caption datasets
  - ImageNet, for instance.

ResNet50V2: What is it?

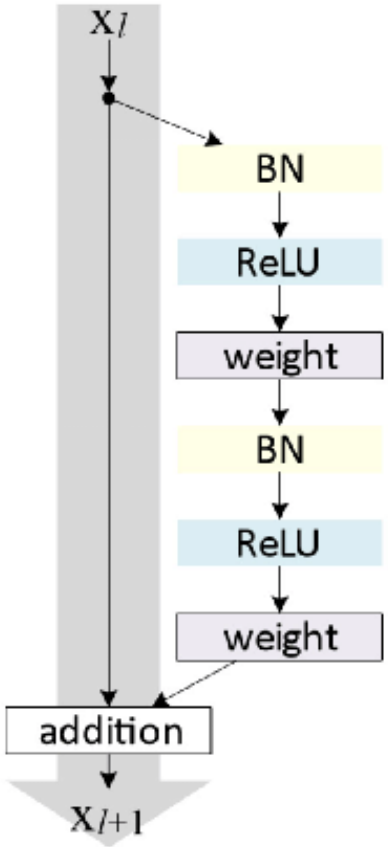
- The second version of ResNet50 thinks to improvements over the original Resnet50.
- Fewer parameters, optimizations for modern hardware, etc.



(This is ResNet50)

High-level view of how ResNet50V2 works

- Multiple residual blocks, each having several convolutional layers.
- The output is passed to a skip connection to learn the original input.
- Lastly, it outputs a probability distribution model over a predefined set.



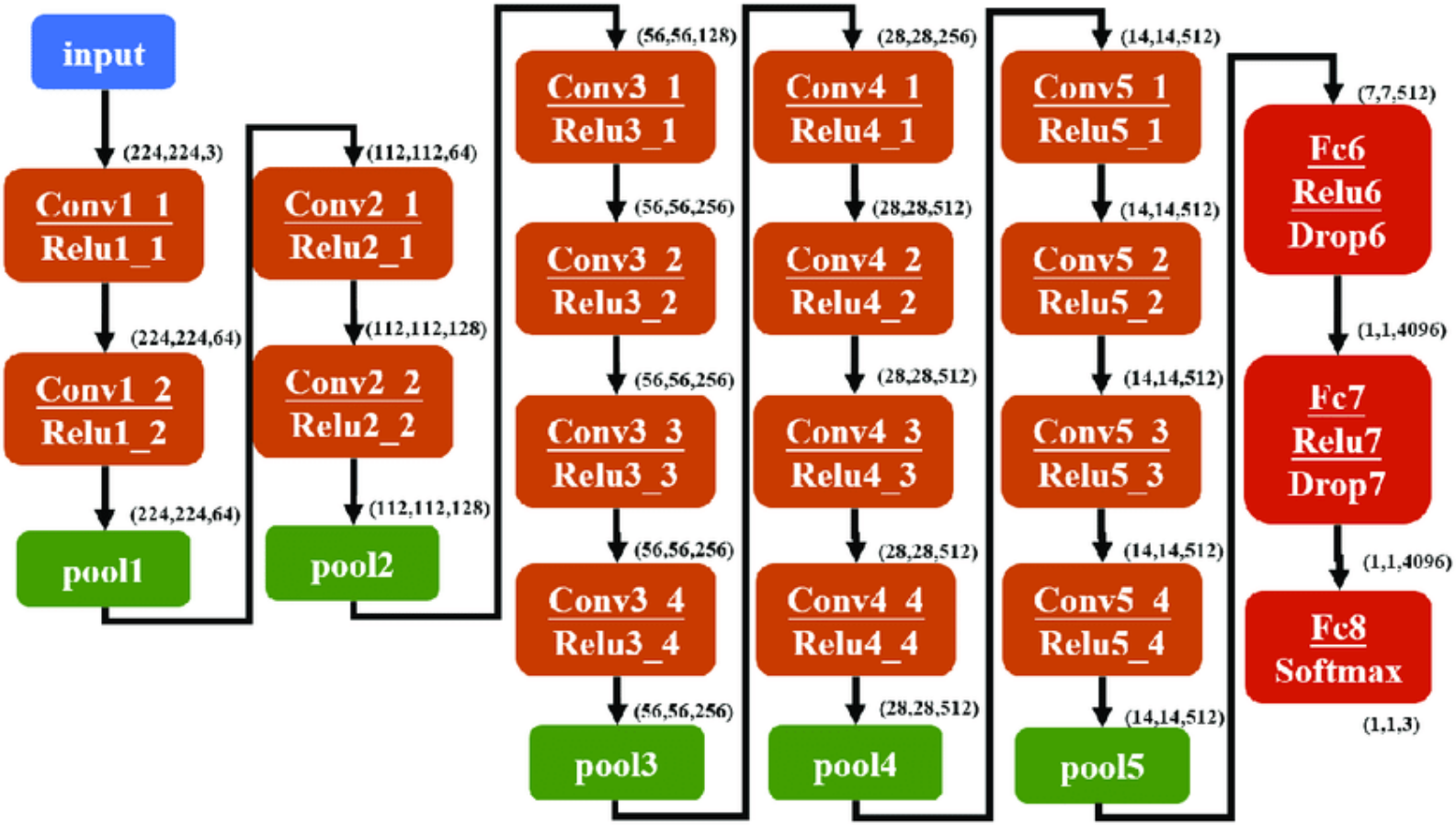
VGG19: What is it?

- Named after it's creators: The **V**isual **G**eometry **G**roup at the University of Oxford.
- A **c**onvulutional **n**eutral **n**etwork.
- An extension of VGG16, adding 3 fully connected layers along with the original 16 convolutionals.

VGG19: How it works

- Convolutional layers for feature extraction from an input.
- Passed through fully connected layers to classify the input.
- Using a variety of techniques for improvements.
  - Dropout
  - Batch normalization
  - Data augmentation

VGG19: Typical example



[1]

[1] https://www.researchgate.net/figure/Detailed-architecture-diagram-of-the-VGG19-network-Detailed-architecture-diagram-of-the_fig3_357830513/actions#reference

How do we evaluate it?

- It's crucial to determine the accuracy of any model.
- Involves comparing model-generated captions with a set of references.
- Here, many candiates are good for this.
  - ROUGE
  - METEOR
  - CIDEr
- Here, we'll be using BLEU score.

BLEU score: How it works

- Calculated computing the n-gram precision of the captions.

$$\text{BLEU} = \text{BP} \cdot \exp\left(\frac{1}{n}\sum_{i=1}^{n}\log p_i\right)$$

- Followed by a brevity penalty.

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{1-\frac{r}{c}} & \text{if } c \leq r \end{cases}$$

BLEU score's Pros and Cons

- It's widely accepted
- Easy to compute
- Quantitative measure
- *Has problems with sementics and is length-sensitive*

Set up

- Our dataset will be the Flickr8K[1], with it's images and captions
- We'll be using VGG19 and ResNet50V2, both with Attention.

[1] https://github.com/jbrownlee/Datasets/releases/tag/Flickr8k

The training

- Training will be at 10 epochs
- Both of them trained similarly, each being 10 minutes
- But there is a catch for time, since there is also:
  - Feature extraction (15 minutes)
  - Validation (7 minutes)

The results

- Overall, both models perform well enough, but it could be better.
- The current implementation of how to do BLEU score warrants further testing and evaluation
  - Since it uses the averages of all trials
  - Meaning that the results is about $\approx 0.4 - 0.5$ on both, on 10 captions.
  - Ignoring that, each would be about around $\approx 0.5 - 0.8$

Anything to talk about?