Tell Me What I See

Viktor Makoviichuk (06086584)
viktorm@stanford.edu

Peter Lapko (06167608)
plapko@stanford.edu

Boris Kovalenko (06201315)
kboris@stanford.edu

Mentor: Christopher Manning
manning@cs.stanford.edu

Abstract

In this project we explore performance of different approaches to the image caption task and in particular application of GRU units instead of LSTM based architectures. We created a model with modular architecture that allows to easy replacement and testing of different visual, CNN's and language, RNN's networks and made a number of experiments. As a result we found a well performing model with a simple GRU based architecture and rather short training time.

1 Introduction

AI researchers have a an ultimate goal to create agents that can perceive and understand the visual world around us and who are able to communicate with us in natural language. Humans can accomplish a lot of tasks involving complex visual recognition and scene understanding or using natural language to express thoughts and talk to each other. With just a quick glance at an picture human can point out and describe a wide variety of details about it. And while this ability feels easy and natural for us it is a very difficult task for a computer. It has to find a high-level semantic concepts describing patterns of brightness values of a few millions of pixels from the image. And even more complex task is to determine and describe complex high-level concepts that require difficult inferences from the objects in the scene.

The recent rapid progress in the area of visual recognition shows that current state of the art image recognition models based on deep convolution neural networks able of detecting thousands of visual categories at accuracies on the level with humans, or even surpassing them in some cases. The applications of Deep Learning approach for natural language processing show a lot of promises too, so using combination of such models seems like a good choice for tasks like generation of caption describing a given image. It has a wide variety of possible application: from automatic labeling of photos from one's vacation trip to helping blind people to perceive the world around us, so we decided to choose it as a goal for our project.

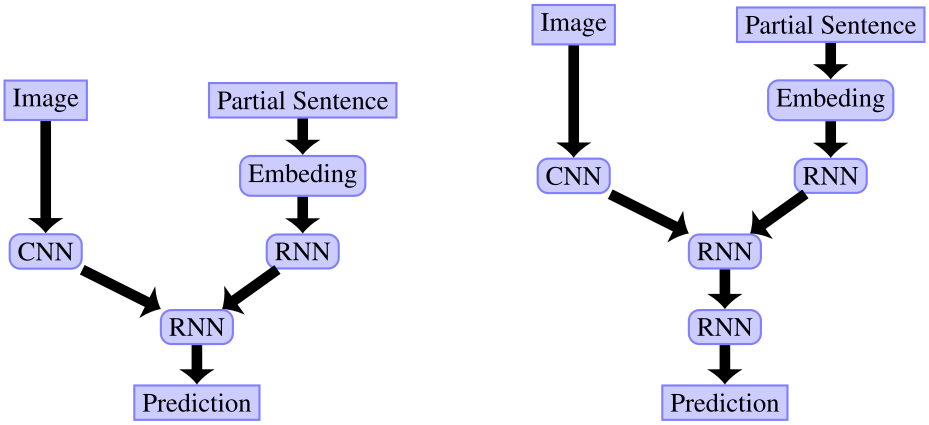
2 Background/Related Work

Some of the classical works that inspired us to choose this topic were: Grounded Compositional Semantics for Finding and Describing Images with Sentences [1], Dense Captioning project [2] that allows efficiently identification and capturing all the things in an image with a single forward pass of a network. In [3] authors suggested improved dynamic memory network beating previous state of the art models in usual and visual question answering. In [4] using hierarchical recurrent neural networks was suggested for generation consistent stories describing the image.

3 Approach

At the first step we preprocess images into tensors with the shape (image width, image height, 3 channels). Then we create a 1-indexed vocabulary from all of the words from the training captions in the training data and three special tokens <BEGIN>, <END> and <UNK>. <BEGIN> and <END> are used to mark the beginning and ending of a sentence. Token <UNK> is used for rare words to reduce the result vocabulary size. Then the result vocabulary is used to create a numerical representation of all partial sentences generated from the training captions. 0 index is used for masking.

Our model consists of 2 interacting parts: visual, CNN based and language, RNN module. Convolutional Neural Network is used to extract features from image while the language part embeds partial sentences into dense representation. On the next step the feature vectors from image and text are concatenated and fed to the next, recurrent layer. Depending on the chosen architecture there can be a few such layers, and the final one is softmax classification layer.



(a) General model architecture

(b) Extended model architecture

Figure 1: Model architectures

Left image 1a shows the graph of our simplest model. The CNN we used in experiments included VGG-16, SqueezeNet, Xception, ResNet-50. In the RNN units we used GRU or LSTM units and tried different architectures (1b). The Embedding layer was either randomly initialized or we used GloVe vectors for initialization.

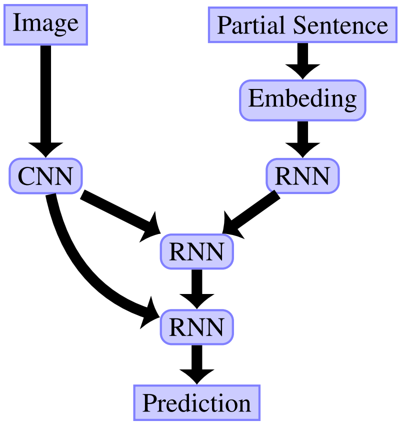


Figure 2: Stack model architecture

4 Experiments

All of our work was performed using COCO datasets [5] as they have a large collection of images with convenient format for their descriptions.

For our experiments we used our own implementation of captioning model described above. We have done hyperparameter search and evaluated lots of variants of architectures. Our implementations allows fast assembly of captioning models with the usage of different visual and language segment models. In any variant of captioning model the last layer is the ”Prediction” layer - a dense layer with softmax activation, followed by cross entropy loss.

We used Nadam optimization algorithm. It is a variant of stochastic gradient descent algorithm with additional improvements. In paper [6] it is recommended to leave most of algorithm hyperparameters as is. We cross validated learning rate, using our base model. Our base model uses Resnet-50 in visual part and two GRU units, one for sentence embedding and one for processing concatenated vector of image and sentence features. Train and validation loss histories with different learning rates presented on Figure 3. Learning rate 1e-03 provides the best speed of convergence on train and validation data. We also evaluated Adam algorithm with the best learning rate for Nadam. We have found that for our task there is no big difference between two algorithms and decided to go with Nadam.

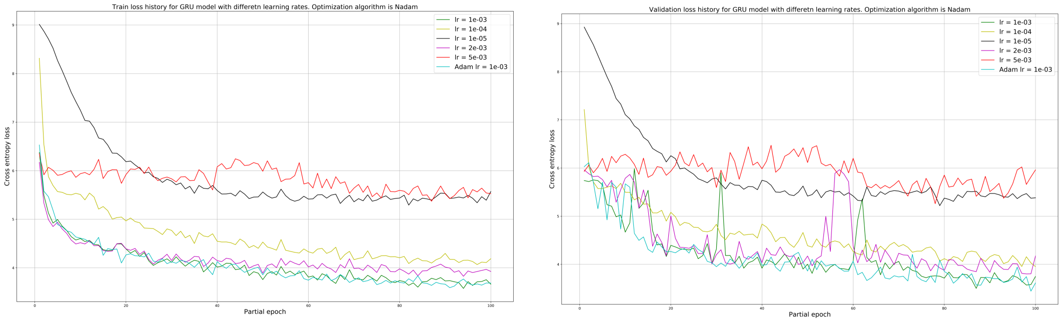


Figure 3: Loss history for different learning rates. Optimization algorithm is Nadam

We evaluated a number of language models. Train and validation loss histories are presented on Figure 4. We started with basic GRU and LSTM units (on Figure 4 it's GRU_vanilla and LSTM), evaluated it's extension - bidirectional unit (on figure 4 it's GRU_BIDIR). Tried adding second GRU (Model from Figure 1b, GRU_2 on Figure 4). Also, we tried few versions of stacked model with different hidden dimensions (Figure 2, GRU_stacked_128 and GRU_stacked_256), in this model features from images go to all GRU units. The best resulting model was basic model with GRU unit as it trained faster than others and generated good sentences.

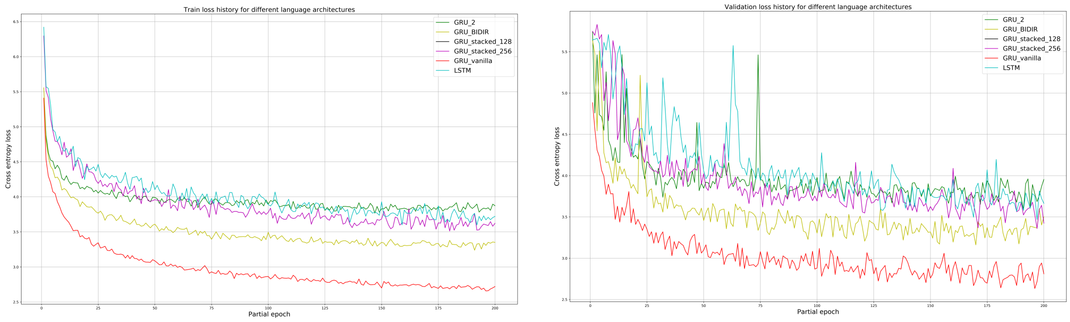


Figure 4: Loss history for different language architectures.

One of the important parts of our model is the embedding layer, which is used to learn representation of words. This layer can be initialized using different random schemes, or through use of pre-trained word vectors. For this task we used pre-trained GloVe vectors from [7], we evaluated 100, 200 and 300 - dimensional versions of GloVe vectors. All version of pre-trained vectors give slight boost in learning speed vs. randomly initialized embeddings. Out of three dimensions the best one was 300 dimensional version.

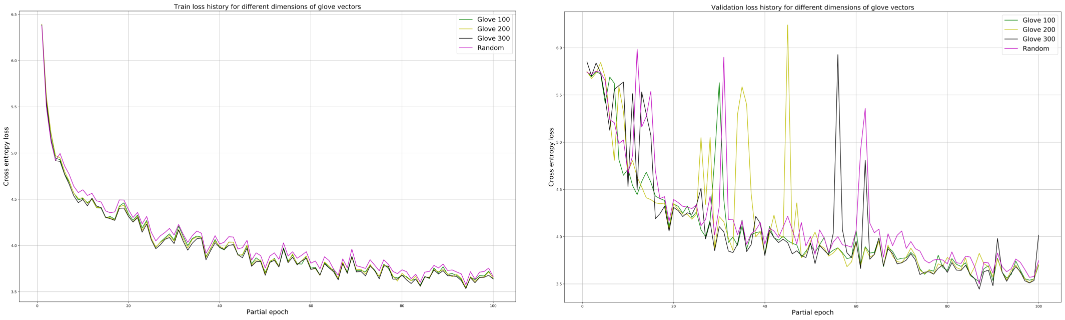
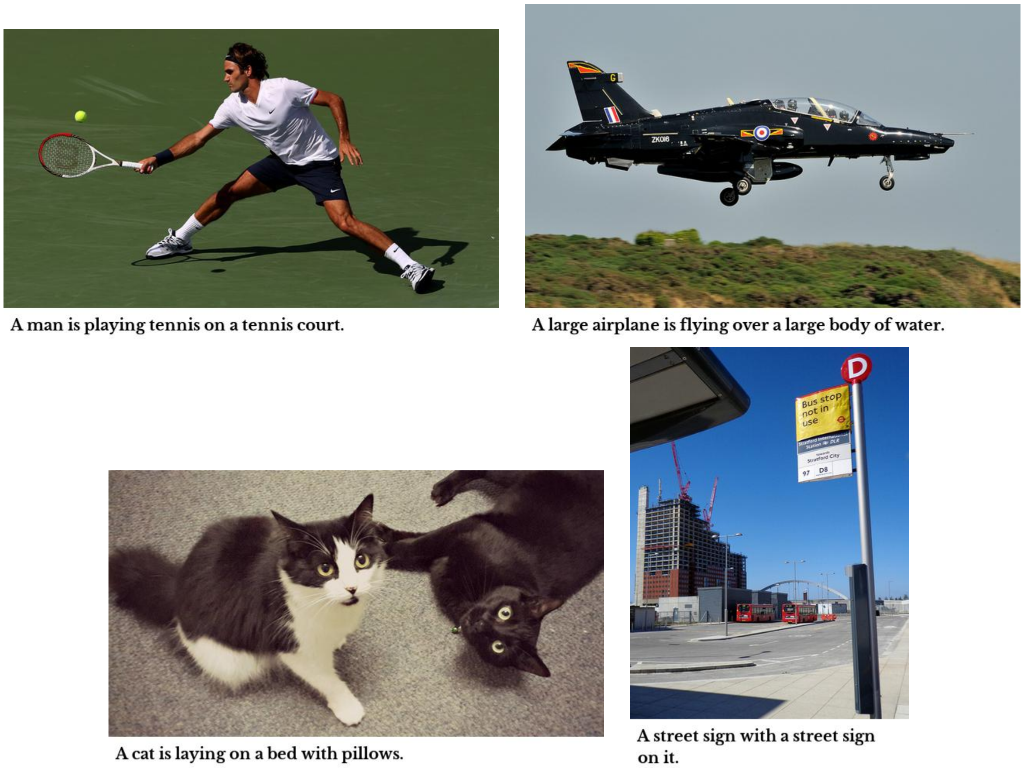


Figure 5: Loss history for different dimensions of glove vectors.

We also evaluated the captions our model generated using MS COCO validation dataset. For evaluation we used Bleu, METEOR CIDEr and ROUGE_L metrics. Obtained results allow us to say that our best model after enough training in most cases can generate meaningful captions that describe provided image very close to what is really present on the scene.

	CIDEr	Bleu_1	Bleu_2	Bleu_3	Bleu_4	METEOR	ROUGE_L
Bidirection GRU	0.268	0.518	0.318	0.188	0.114	0.141	0.376
GRU	0.489	0.583	0.389	0.254	0.17	0.173	0.426

Below are examples of test images and captions that were generated for them.



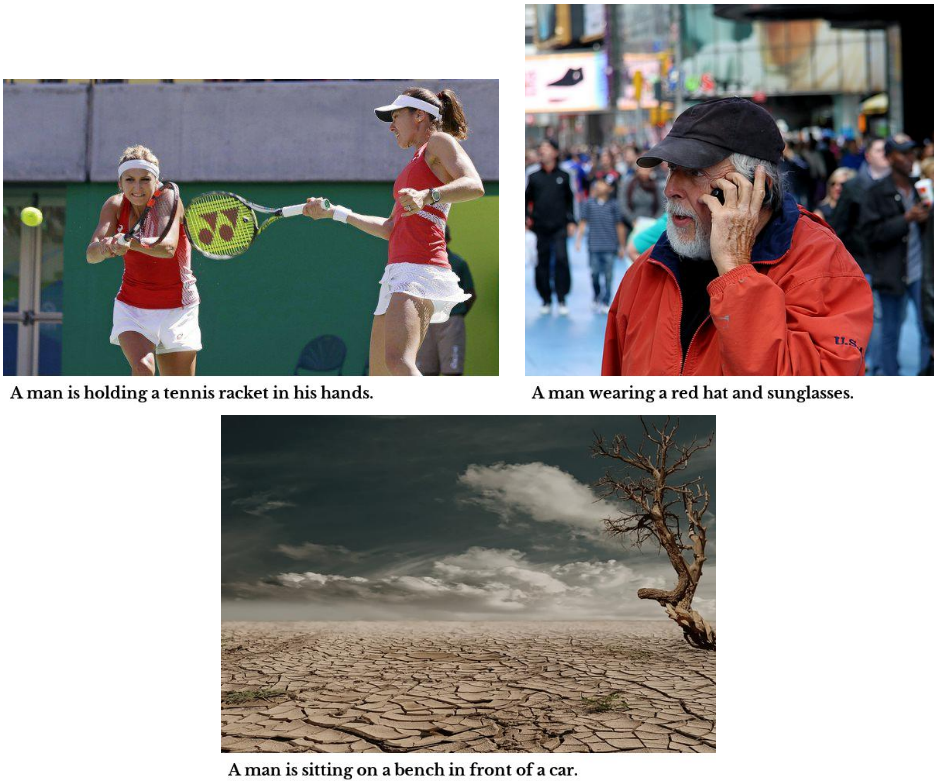
A man is playing tennis on a tennis court.

A large airplane is flying over a large body of water.

A cat is laying on a bed with pillows.

A street sign with a street sign on it.

Figure 6: Examples of good generated captions



A man is holding a tennis racket in his hands.

A man wearing a red hat and sunglasses.

A man is sitting on a bench in front of a car.

Figure 7: Examples of bad generated captions

5 Conclusion

#	User	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr-D
53	kolarmartin	0.716 (55)	0.541 (55)	0.392 (55)	0.278 (55)	0.252 (53)	0.509 (55)	0.536 (53)
54	Kovalenko	0.758 (53)	0.605 (53)	0.464 (53)	0.351 (53)	0.233 (54)	0.547 (53)	0.451 (54)
55	gabriel.j	0.725 (54)	0.564 (54)	0.427 (54)	0.323 (54)	0.219 (56)	0.524 (54)	0.443 (55)

Figure 8: c40-part of the COCO leaderboard

We achieved quite impressive results using models with much less capacity (number of parameters) compared to that described in the original papers and with GRE units instead of the LSTM. A lot of generated sentences are totally relevant to the content of the picture but there are still quite a lot of captions that are partially or totally wrong. We have submitted out results to the COCO site, and while the scores in CIDER-D, ROUGE-L, Meteor and BLEU-4 still are not yet close to the state of the art, we have got quite good result for the start as you can see on Figure 8.

The main source of the errors at the moment is the large difference in words frequencies in the pictures descriptions of the train set that we have not compensated yet. At the first stages of the training almost every capture begins with ”a man is sitting on ...” and as you can see all the words in the phrase are at the top of frequencies table in the Figure 9. One of the classical examples you can see on the desert picture on Figure 7.

On the image with girls playing tennis you can see that this error can manifest itself even much later. Also on this picture and picture with man in red jacket you can see an example of other kinds of errors that can be found later, when the model is trained quite good enough - similar concepts (features) required longer time for resolution. Women and man are both humans, man talking by phone is wearing some kind of ”hat” and his clothing contains red color, but it’s not a hat but his jacket.

Token	Count
a	659499
on	144689
of	136143
the	130731
in	122914
with	101363
and	91217
is	66095
man	49271
to	45422
sitting	35619
an	33602
two	32850
standing	29111
people	28712

Figure 9: Counts of tokens in the train Data

We have implemented a potential solution to this problem - initializations of the softmax layer with the logarithms of words frequencies, but didn’t have enough time to perform extensive testing for a large number of different architectures. Experiments with the base GRU model showed improvements in convergence and overall quality of the captions.

Our short term plans include improving quality and test scores of our models with the architectures close to the current and then try more advanced models starting from the attention one [8] to get closer to the state of the art. Also we plan to test adding more features to our language model, for example windows feature - concatenated consecutive words in sentence instead of standalone words. And to implement and test beam search for better sampling of the words for the caption sentence and test with different values of beam size.

In more distant future we would like to look into related but more complex tasks like DensCap [2], paragraph generation base on the approach suggested in [4] and visual question answering using memory networks, described in [3].

Source code for our project can be found on github [9].

References

[1] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded Compositional Semantics for Finding and Describing Images with Sentences. *ACL*, 2014.

[2] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016.

[3] Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. *arXiv*, 1603, 2016.

[4] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. A Hierarchical Approach for Generating Descriptive Image Paragraphs. *arXiv preprint arXiv:1611.06607*, 2016.

[5] COCO dataset. `http://mscoco.org/`.

[6] Dozat, Timothy Incorporating Nesterov Momentum into Adam. Stanford University, Tech. Rep (2015). `http://cs229.stanford.edu/proj2015/054_report.pdf`.

[7] GloVe: Global Vectors for Word Representation. `https://nlp.stanford.edu/projects/glove/`.

[8] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. *arXiv preprint arXiv:1612.01887*, 2016.

[9] Project link. `https://github.com/lps-stanf/cs224n_prj`.