Image Captioning using an LSTM network

Fangzhou Ai
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093
faai@eng.ucsd.edu

Yue Qiao
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093
yuq021@eng.ucsd.edu

Zunming Zhang
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093
zuz008@eng.ucsd.edu

Abstract

Image captioning is an important problem in artificial intelligence, related to both computer vision and natural language processing. This work explores the power of Recurrent Neural Networks to deal with image captioning problem. We used a pre-trained convolutional network, ResNet50, as the encoder and an LSTM model as the decoder. We also tried to replaced the LSTM with Vinalla RNN and GRU and compared the performances of different models. For evaluation, we tried to compare the test set Cross Entropy Loss and BLEU scores for different models. For baseline model, we achieved a test loss of 1.966, BLEU1 socre of 56.48 and BLEU4 score of 7.75. We got a better result for GRU method and the comparison could be found in Table 1 and Table 2. We also tried to fine tuned the baseline model and the model performance

1 Introduction

Automatically generating a natural language description of an image, a problem known as image captioning, has recently received a lot of attention in Computer Vision. The problem is interesting not only because it has important practical applications, such as helping visually impaired people see, but also because it is regarded as a grand challenge for image understanding which is a core problem in Computer Vision. Generating a meaningful natural language description of an image requires a level of image understanding that goes well beyond image classification and object detection. The problem is also interesting in that it connects Computer Vision with Natural Language Processing which are two major fields in Artificial Intelligence.

In this assignment, we will explore the power of Recurrent Neural Networks to deal with data that has temporal structure. Specifically, we will generate captions for images. In order to achieve this, we will need an encoder-decoder architecture for this assignment. Simply, the encoder will take the image as input and encode it into a vector of feature values. This will then be passed through a linear

Preprint. Under review.

layer for providing the input to LSTM. It will be trained to predict the next word at each step. we used a pre-trained convolutional network as the encoder and an LSTM model as the decoder and tried to fine tune the encoder and train the decoder by backpropagating error into it. The error will come from caption generation. The training uses images and several captions for each image generated by humans. We then run the network in generative mode to generate captions on images it has never seen before. We also tried to replace the LSTM in the encoder part with Vanilla RNN and GRU and compared the performances of them.

2 Related Work

Recently, several approaches have been proposed for image captioning. We can roughly classify those methods into three categories. The first category is template based approaches that generate caption templates based on detecting objects and discovering attributes within image. For example, the work [12] was proposed to parse a whole sentence into several phrases, and learn the relationships between phrases and objects within an image. In [9], conditional random field (CRF) was used to correspond objects, attributes and prepositions of image content and predict the best label. Other similar methods were presented in [17, 11, 10]. These methods are typically hard-designed and rely on fixed template, which mostly lead to poor performance in generating variable-length sentences. The second category is retrieval based approach, this sort of methods treat image captioning as retrieval task. By leveraging distance metric to retrieve similar captioned images, then modify and combine retrieved captions to generate caption [11]. But these approaches generally need additional procedures such as modification and generalization process to fit image query.

Inspired by the success use of CNN [8, 22] and Recurrent Neural Network [1, 15, 16]. The third category is emerged as neural network based methods [5, 4, 7, 19, 21]. Our work also belongs to this category. Among those work, Kiro et al.[6] can be as pioneer work to use neural network for image captioning with multimodal neural language model. In their follow up work [7], Kiro et al. introduced an encoder-decoder pipeline where sentence was encoded by LSTM and decoded with structure-content neural language model (SC-NLM). Socher et al.[18] presented a DT-RNN (Dependency Tree-Recursive Neural Network) to embed sentence into a vector space in order to retrieve images. Later on, Mao et al.[14] proposed m-RNN which replaces feed-forward neural language model in [7]. Similar architectures were introduced in NIC [19] and LRCN [3], both approaches use LSTM to learn text context.

Our work is based on LSTM method to do the image captioning. In our approach, we used a pre-trained concolutional network as the encoder and an LSTM model as the decoder to do the image captioning. We also tried to replaced LSTM with the Vanilla RNN and GRU to compared the model performances.

3 Methods

3.1 Dataset

For this image captioning task, we used the dataset from the well-known Common Objects in Context[13] (COCO) repssitory. COCO is a large-scale object detection, segmentation, and captioning dataset. In this report, we used a subset (around 1/5) of the COCO 2015 Image Captioning Task. The training set contains around 82k images with roughly 410k captions while the test set has around 3k images with almost 15k captions. The original images in the dataset are of different sizes and aspect ratios, which we are resizing to 256x256 before the training.

3.2 Model

3.2.1 Baseline Model

Our captioning system is implemented based on a Long Short-Time Memory (LSTM) network (baseline model). For the encoder part, we use a forzen pretrained convolutional network, namely ResNet50, as the encoder. We removed the last layer of the network and added a trainable linear layer that outputs a feaure vector with a fixed size for each image. This feature becomes the initial hidden state and the cell state of LSTM network. For baseline model, we resized the image to 256x256.

For LSTM decoder, we first initialize the hidden state and the cell state using the encoded image from CNN encoder. Then, we send all captions, seperated by space, into the embedding layer. The embedding layer basicly converts the words feature into one hot encoding format and reduces the feature dimensions so that the features fit LSTM network's input size. Later, we feed the embedded features into LSTM cells. Instead of using the output from the last LSTM cell, we use a techique called teacher forcing, which feed the network with the ground truth last word regardless of what the network outputs last time. The outputs of LSTM cell will feed to a linear layer to scale up the dimensions. In the end, the scaled output features are feed into the softmax layer.

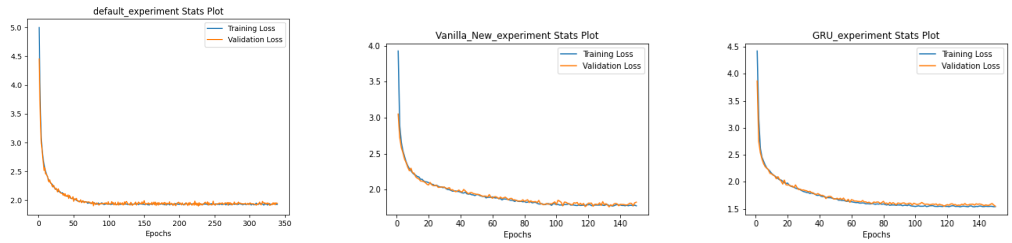
3.2.2 Vanilla RNN and GRU

For the model comparison, we also tried Vanilla RNN and GRU[2]. GRU is a varient of recurrent neural networks. GRU is like a long short-term memory (LSTM) with a forget gate, but has fewer parameters than LSTM, as it lacks an output gate.[20] For these two models, we simply replaced LSTM module in the encoder with either a Vanilla or a GRU, while the others remained to be same. Then, trained and compared the performance of these three different models.

4 Results

4.1 Learning Curve

The learning curve for LSTM, Vanilla RNN and GRU could be found in Figure 1.



(a) LSTM Loss (b) Vanilla RNN Loss (c) GRU Loss

Figure 1: Training and Validation loss for LSTM, Vanilla RNN and GRU

4.2 Cross Entropy Loss

The test set Cross Entropy Loss of different models could be found in Table 1

Table 1: Cross Entropy Loss for different models

Model	Testset Cross Entropy Loss
Baseline	1.966
Vanilla RNN	1.819
GRU	1.586
Fine tuning model	TBD

4.3 BLEU Score

For this report, we set the weight for BLEU1 to be [1, 0, 0, 0] and BLEU4 to be [0.25, 0.25, 0.25, 0.25]. The BLEU scores for different models could be found in Table 2

4.4 Images Visualization of the Best Performance Model

The good predictions are shwon in Figure 2 and bad predictions are shown in Figure 3.

Table 2: BLEU scores for different models

Model	BLEU1	BLEU4
Baseline	56.48	7.75
Vanilla RNN	54.59	8.93
GRU	65.16	17.17
Finee tuning model	65.02	16.83

5 Discussion

5.1 Baseline with different Temperatures

We try to use a stochastic approach and generate the output words using the distribution from softmax layer. The temperature controls what the distribution looks like. When the temperature approaches 0, the distribution is nearly determinstic. When the temperature approaches ∞, the distribution is completely uniform. The BLEU scores is listed in Table 3.

Table 3: BLEU scores for baseline model with different temperature

Temperature	BLEU1	BLEU4
0.01	0.117	0.022
0.1	0.107	0.020
0.2	0.099	0.018
0.7	0.106	0.020
1	0.118	0.022
1.5	0.142	0.026
2	0.152	0.028
5	0.169	0.032

From the table we can see that a higher temperature produces higher BLEU score. All of the score is lower than the determinstic approach. We suppose that the baseline model is too small to support the stochastic approach to produce good sentences.

5.2 Baseline LSTM vs Vanilla RNN vs GRU

5.2.1 Learning Curve

From the Figure 1, we can see that the Vanilla RNN takes 110 epochs to converge, LSTM takes about 80 epochs to converge, and GRU also takes about 80 epochs to converge. We believe that since RNN does not contains any kinds of forgot gate, the model suffers from vanishing gradients problem and the training is slower.

5.2.2 Cross Entropy Loss

Compare with the loss value in Table 1, we can see that GRU has a lower loss. It proves that GRU works better on networks with small feature sizes.

5.2.3 BLEU Score

Compare with the BLEU score value in Table 2, we can see that the Vanilla RNN is worse than LSTM and GRU on BLEU-1 score. GRU has the highest score on both BLEU-1 and BLEU-4. It again shows that GRU works better on networks with small feature sizes.

5.3 Baseline model vs fine tuning model

For fine tuning part, we use grid search method to find the best model, all tasks are submitted to GPU cluster as batch jobs. The result shows that increasing the embedding size is mode effective than increasing the hidden size, which is reasonable since the length of the vector for each word is 60,000

(a) a bathroom with a toilet, sink and shower.

(b) a woman sitting under an umbrella .

(c) a herd of sheep walking down a dirt road .

(d) a living room filled with furniture and a flat screen tv .

(e) a little girl sitting at a table with a plate of food .

Figure 2: Good predictions.

in this dataset, the original 300 embedding size is apparently too small to fit large vocabulary. In our experiment we found that when hidden size is 1200 and embedding size is 2000, we achieve the best BLEU score.

5.3.1 Learning Curve

The learning curve is shown in Figure 4.

5.3.2 Cross Entropy Loss

The loss for each model with different parameters are shown in Table 4. All the results are collected after 100 epochs training. The validation loss shows that larger embedding size is better than larger hidden size.

(a) a train on a train track near a building

(b) A man standing in front of a refrigerator.

(c) a person riding a horse on a beach

(d) a parking meter on the side of the road .

(e) a man is holding a cell phone in his hand .

Figure 3: Bad predictions.

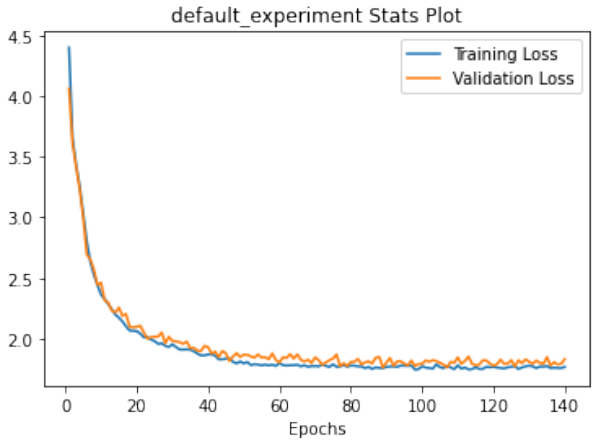


Figure 4: Fine tuning model loss.

5.3.3 BLEU Score

The BLEU1 scores are shown in Table 5 and BLEU4 scores are shown in Table 6. We can also see that larger embedding size is better than larger hidden size.

5.3.4 Fine tuning for Vanilla RNN and GRU

Though this is not required, we still tuned the vanilla RNN and GRU model to check the differences between these tow models and LSTM. During these extra fine tuning process, we also observed that

Table 4: Validation loss of different models.

Hidden/embedding size	1000	1200	1400	1600	1800	2000
1000	1.681337369	1.691209305	1.580959492	1.551394041	1.583891191	1.560777765
1200	1.816109741	1.618265422	1.57035513	1.536661038	1.524779346	1.522998657
1400	1.662981744	1.641106087	1.616746906	1.559747254	1.637271137	1.543163765
1600	1.75068753	1.587794039	1.563622826	1.529088398	1.550765577	1.519091342
1800	1.646753683	1.591480875	1.586649266	1.591782318	1.537952337	1.521097336
2000	1.639377112	1.586326071	1.671221515	1.595450379	1.560300544	1.517007219

Table 5: BLEU1 score of different models.

Hidden/embedding size	1000	1200	1400	1600	1800	2000
1000	59.41769621	59.62020083	63.35089825	63.32872151	63.73170132	64.43839568
1200	56.05131035	62.30367823	62.35710535	63.26050116	64.60560355	65.01930785
1400	60.39761923	60.39945857	60.63426329	62.73186609	61.13974605	64.93391073
1600	57.34373218	61.60656617	62.77856346	62.47185297	64.2897595	63.74235525
1800	58.97915755	61.51667164	62.34537701	60.52766041	63.5199832	63.17110813
2000	59.78645267	60.58827569	58.05156176	61.15798722	62.67753716	63.65793274

the embedding size has a more important effect on validation loss and BLEU score than hidden size, the difference is RNN and GRU model seems to be saturated when embedding size is around 1200 1400, namely after that increasing embedding size cannot bring us more improvement, while for LSTM, the saturation threshold is approaching 2000.

6 Individual contributions to the project

Fangzhou Ai Fine tuning.

Yue Qiao Baseline LSTM Model.

Zunming Zhang

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[2] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[3] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[4] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[5] Andrej Karpathy, Armand Joulin, and Li Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. *arXiv preprint arXiv:1406.5679*, 2014.

[6] Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. Multimodal neural language models. In *International conference on machine learning*, pages 595–603. PMLR, 2014.

Table 6: BLEU4 score of different models.

Hidden/embedding size	1000	1200	1400	1600	1800	2000
1000	11.84089542	12.40819432	15.3686782	15.82054416	15.70298138	16.54373736
1200	8.772820601	14.55022853	15.01156252	15.60198284	16.4523467	16.83235187
1400	12.8796874	12.91051223	13.49639036	15.16698644	13.21635365	16.18174622
1600	10.58407818	14.31981595	14.94549483	15.37257563	16.52898121	15.7704121
1800	12.36232193	14.22257743	14.84009182	13.30323795	15.66635975	15.41644636
2000	12.19404039	13.25380803	11.32768971	13.89880156	15.30576858	15.89286523

[7] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[9] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C Berg, and Tamara L Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12): 2891–2903, 2013.

[10] Polina Kuznetsova, Vicente Ordonez, Alexander Berg, Tamara Berg, and Yejin Choi. Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 359–368, 2012.

[11] Polina Kuznetsova, Vicente Ordonez, Tamara L Berg, and Yejin Choi. Treetalk: Composition and compression of trees for image descriptions. *Transactions of the Association for Computational Linguistics*, 2:351–362, 2014.

[12] Siming Li, Girish Kulkarni, Tamara Berg, Alexander Berg, and Yejin Choi. Composing simple image descriptions using web-scale n-grams. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 220–228, 2011.

[13] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[14] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.

[15] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[16] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5528–5531. IEEE, 2011.

[17] Margaret Mitchell, Jesse Dodge, Amit Goyal, Kota Yamaguchi, Karl Stratos, Xufeng Han, Alyssa Mensch, Alexander Berg, Tamara Berg, and Hal Daumé III. Midge: Generating image descriptions from computer vision detections. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 747–756, 2012.

[18] Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2:207–218, 2014.

[19] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[20] Wikipedia contributors. Gated recurrent unit — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Gated_recurrent_unit&oldid=997015931, 2020. [Online; accessed 27-February-2021].

[21] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[22] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.