Image Captioning

Ran Schreiber

Submitted as a final project report for Machine Learning Applications,
Tel Aviv University, 2019

1 Introduction

Image captioning is the task of generating human-like natural language descriptions of given images. Algorithms that aim to solve it are required to identify correctly the semantically important objects in the image, and to capture their relationships. Also, the generated caption should be a grammatically valid natural language sentence.

1.1 Applications

The task of image captioning may be used as a very effective building block in many applications; It may be the basis for image-based search engines (that is, search engines that receive images as inputs), used in social media, and most importantly - it is an essential component in nearly every technology aimed to assist visually impaired people.

2 Solution

2.1 General approach

I've decided to tackle the task with a deep learning approach, as a concrete and precise mapping between images and textual captions is rather vague and (apparently) cannot be calculated by hand. The general structure of the architecture used in the project follows the encoder-decoder scheme: An input image is encoded into a lower dimensional vector (that functions as a latent variable), and then the encoded vector is used as a seed for the decoder which returns the output sequence.

2.2 Design

2.2.1 Encoder

The encoder is a deep convolutional neural network (CNN) that extracts features out of the input image and outputs a one-dimensional encoded vector of size 1000. The convolutional network that is used in the project is ResNet-34.

2.2.2 Decoder

The decoder is a two-layered long short-term memory recurrent network (LSTM). It uses the encoder's output as its initial hidden state and cell state, and produces a sequence of vectors.

2.2.3 Output Network

The sequence of hidden states that the RNN produces is the basis for the output caption. It is processed by a small fully-connected neural network of one hidden layer that uses the leaky-ReLU nonlinearity. At each iteration, the network gets one hidden state and turns it into a probability

distribution over the English language vocabulary (the vocabulary has been prepared using the training data). Finally, words are sampled randomly and the caption is constructed word-by-word.

2.2.4 Architecture initialization

Each building block of the architecture has been initialized differently; The encoder CNN has been pretrained on ImageNet, the weights of the LSTM are sampled over a uniform distribution and the output network is initialized using the normal He initialization.

2.3 The dataset

The dataset used in the project is COCO (Common Objects in Context). The training set consists of 118,287 images (about 18GB), from which 5000 samples have been extracted for evaluation during the training. The testing set consists of 5000 images (about 800MB). Each image is associated with multiple possible captions. Each sample from the dataset returns with a single randomly selected caption, and data augmentation techniques have been applied for helping to achieve better generalization and for minimizing overfitting.

2.4 Optimization

The optimization algorithm that has been used is the Adam algorithm. During training, it has tried to minimize the negative log-likelihood loss.

3 Experimental results

The network consists of 172,602,224 parameters, it has been trained in batches of 55 samples for 12 epochs; The training process has taken 7 days, 1 hour and 12 minutes. Each epoch has been followed by calculating the loss over the total training set and the loss over the total evaluation set, and when the second has been reduced, the model has been saved onto the filesystem. Finally, after training, the model has been tested and the negative log-likelihood loss, the perplexity and three BLEU scores (BLEU-2, BLEU-3, BLEU-4) have been calculated. Both training and testing have been performed on Google Cloud on an NVIDIA® Tesla® K80 GPU. The training process has been monitored via the Stackdriver Logging service.

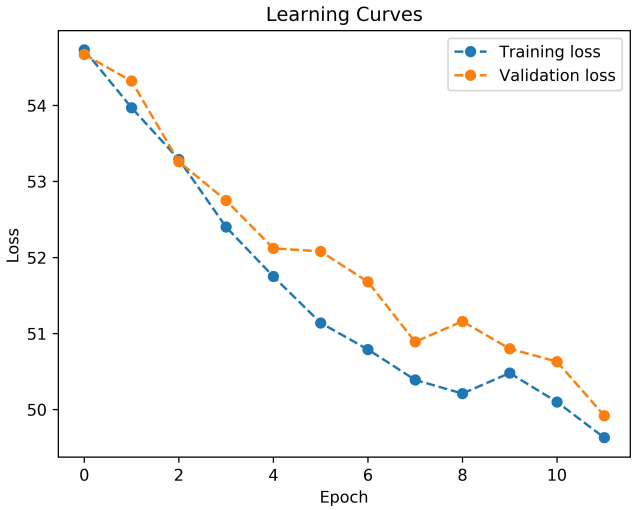
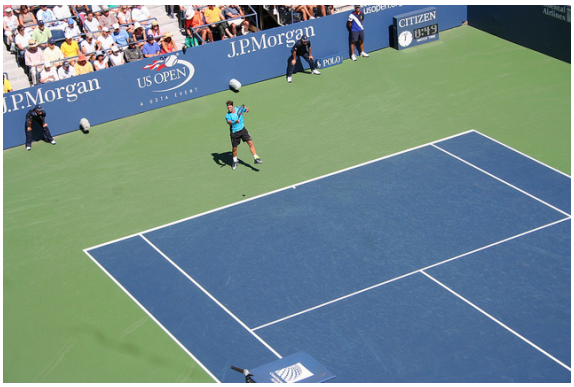


Figure 1: Learning Curves



(a) Epoch 1:
Truth: A tennis player plays on a court while many people watch
Predicted: A person in a down a down a a

(b) Epoch 2:
Truth: A picture of some food on the table
Predicted: A man filled with with is of a the a

(c) Epoch 3:
Truth: A cat is standing inside of a suitcase
Predicted: A black of at a laying on a a on of

(d) Epoch 4:
Truth: A group of older people sitting on a park bench with a dog on a leash
Predicted: A woman in an sitting motorcycle with on in to a motorcycle

(e) Epoch 5:
Truth: A white toilet sitting next to a white sink
Predicted: A bathroom of with a a a toilet a a

(f) Epoch 6:
Truth: Several young men playing a game of frisbee together
Predicted: A man player in a frisbee ball park field

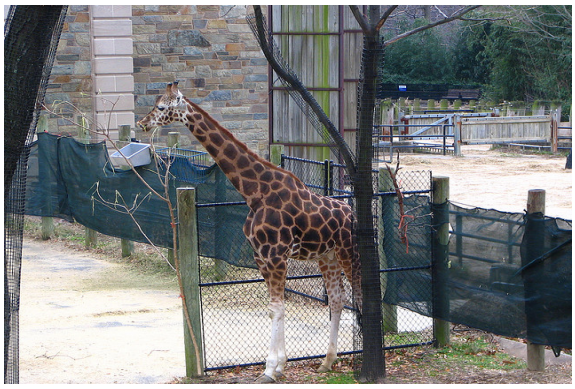
(a) Epoch 6:
Truth: The men on horses are playing a game of polo
Predicted: Several horses and are people a in a

(b) Epoch 6:
Truth: Two men with tennis rackets on a grass playing court
Predicted: A man boy a hit court tennis a ball match court on

(c) Epoch 6:
Truth: Two men playing tennis with one man preparing to hit the ball
Predicted: A young tennis up tennis a tennis during racket

(d) Epoch 7:
Truth: Sandwich on a plate, along with bread in tin
Predicted: A vase is with next and on on

(e) Epoch 8:
Truth: The young people are texting on their cell phones
Predicted: Woman is that at to a phone a her of her on to



(f) Epoch 10:
Truth: A giraffe standing near a fence and tree
Predicted: A giraffes of is giraffe giraffe a enclosure fence building

(a) Epoch 11:
Truth: A man riding a white surfboard on a wave in the ocean
Predicted: A little riding in suit surfboard surfboard in surfboard a the

(b) Epoch 12:
Truth: A bathroom with a sink a toilet and a mirror
Predicted: This bathroom with has white with a a on and

4 Discussion

The project has been conducted a bit differently from the original plans. Originally, several architectures should have been considered and tested - all follow the same encoder-decoder scheme, but the sub-networks types should be different (i.e. trying different CNNs). Also, an Embedding component should have been trained and used (rather than a one-hot encoding), and the output network should have been pruned to reduce evaluation time and to lower the model size. However, my private GPU is unfortunately and unexpectedly not powerful enough for the task, and therefore, the training and testing have been performed on Google Cloud. Because of the high price, I couldn't perform all the experimented I've planned, but only one model.

During the work I've learned a lot about dealing with images, and about working with a big real-world dataset. This project has taught and strengthened many skills of deep-learning.

5 Links

- The complete fully-documented source code of the project, a prepared vocabulary and a trained model may be found at

  https://github.com/ransch/ApplicationsForML

- A few samples of the COCO dataset (after applying data-augmentation techniques) may be viewed at

  https://colab.research.google.com/drive/1QAwcP$_R$NDJDbbUyGCv8CFU9d8oe4FFqG

- The test results may be viewed at

  https://drive.google.com/file/d/15Qz$_Z$J9Mj9XFzig6fF9SPYAukhDbG6xh