Deep Learning – Autumn 2019

Assignment 3 – Deep Learning Final Project

Building an Image Captioning Neural Network

The Business Brief:

Management has greatly enjoyed seeing you transform the data science capabilities for the organisation. Your work on the object detection model as well as your industry-focused literature review and proposal of applications has certainly gotten set them thinking!

Management is now interested in more than just detecting what an image is, but a description of what is in the image. They are confident this will have a range of applications for the organisation.

The Technical Brief:

Your task, as noted above, is to build an image captioning model. This will combine a number of practical and theoretical components that you have dealt with previously.

Specifically, an image captioning model in its simplest form is typically composed of two parts:

1. A feature extractor for the images
2. A sequence-based model to generate captions from the output of 1

Typically, a pretrained CNN is used for the first part an RNN/LSTM (or variant) is used for the second component.

Some blog posts to get you started are listed below. Please see the breakdown of tasks for how you may utilise them in your work.

- https://daniel.lasiman.com/post/image-captioning/
- https://towardsdatascience.com/image-captioning-with-keras-teaching-computers-to-describe-pictures-c88a46a311b8
- https://machinelearningmastery.com/develop-a-deep-learning-caption-generation-model-in-python/

You will likely find more resources and papers online and are encouraged to broaden your search, especially for moving beyond the base requirements below.

Task 1 – A basic model:

For this component you must get a simple model working. You may draw heavily from the online resources listed above as this component is focussed on just getting a working base.

There is no need to reinvent the wheel, however you must cite which resources you decided to use and you cannot reproduce the work of others verbatim.

You must score your model (at minimum) with the scores BLEU-1, BLEU-2, BLEU-3, BLEU-4. However, there may be other metrics you wish to use to score your model including visual inspection and analysis among others.

You must ensure all metrics are *explained* (What does this metric mean?), *justified* (Why are you using this metric?) and *benchmarked* (Is this a good score? How good?).

Regardless of whether you use any of the resources above, you must clearly delineate in your submitted notebooks and report between your basic model and your experimental work to improve this model or any other models implemented.

Task 2 – Extensions:

Once you have a basic model working, you must propose and implement experiments to improve the model you have built. This is very open ended however you must (at minimum) propose and implement one experiment and discuss the results of this. Of course, to gain exceptional marks, teams must go beyond the minimum. You will find in the resources above some suggestions for next steps to take from the authors themselves if you are stuck for ideas.

Some things you *could* consider include (but are by no means limited to):

- Different pretrained image feature extraction models
    - Or not pre-trained
- Different sequence models
- Incorporation of advanced mechanisms such as Attention
- Deeper hyperparameter tuning, perhaps using an advanced methodology.
- Another dataset. Either with your existing model or a new model built for a specific focus. (Flickr 30k?)

Wider reading on well-cite papers, including searching sites such as paperswithcode may assist with finding ideas for models and techniques to try.

Some Advice:

Regardless of the path you choose, remember to document your experiments, decisions and results. You **must** make judicious use of comments and markdown cells to explain what each component of your code does.

As this task has some key technical components (Image part, sequence part, knitting together part), you may wish to split team effort that way to build your initial model and then split into sub-teams for different experimental tasks; coming back together to share and discuss next steps.

For the technical elements, teams that only replicate one of the provided or other resources without detailed commenting and notation can expect pass-level marks. Teams who do this with good explanations (especially in the code) that demonstrate understanding would expect closer to credit-level marks. Doing all of the above with some *reasonable* extensions and experiments (with associated discussions) pushes students into distinction marks and hence exceptional effort in this area can expect exceptional marks.

The Data:

The dataset you will use (at minimum) is the Flickr8k dataset, available from a variety of online sources. It has a set training and testing sets within it. I have uploaded to a google drive folder you can download from. Linked here.

The technology:

You are free to use a deep learning framework of your choice. However, given your experience in Keras and the wealth of resources available for this it would be advisable to at least get your base model working in this before pushing into Tensorflow. Caution is advised utilising other frameworks you have little or no experience with given the short time frame.

This task will likely involve utilisation of greater computation resources than your desktop computers. Many students discovered iterative methodologies for reading in images during task 1B which will prove useful in this task.

Additional resources include:

- Colab for completely free, browser-based notebooks with GPU and TPU backends.
- AWS and Azure credits available from the github developer pack (here)
    - AWS has a deep learning AMI that is compatible with G3 instances and above. (link and tutorials). If you simply select this instance you can switch to virtual environments with all your standard libraries packed in. These instances on spot requests start at 20c/hour (current pricing), hence the AWS credit in the student developer pack will get each team a few hundred hours of training.
    - Azure and GCP likely have similar pre-configured spot instances.
- UTS can arrange GPU credits as well for teams. Please contact me asap to get this process started if interested. I am unsure of limits and budgets on this.

Submission requirements:

- Fully commented and explained notebook with the results of your work. You may find it easier to utilise an alternate editor/IDE for your experimental work, however the final 'data story' and code should be presented in a notebook. The notebook should be able to be run by who you hand over the notebook to.
    - Please ensure you save out your base model and final model you wish to present. You must have a cell that reads in these (using **relative file paths**) so

the retraining doesn't have to happen to see the results of the network. Please test this.
    - You do not need to include every experimental model created, only the first base model and the best model(s) referred to in the report.
- A report following the CRISP-DM style of data projects. It is important to outline key steps and findings from your methodology. In this instance, the business has not got a distinct, single idea for what this technology could be useful for hence you may propose specific application(s) to focus on for your report.
    - The wordcount should not exceed 1500 words.

Good luck all!