CSC 219-01 Machine Learning (Fall 2023)

Project 4: Multi-Modal Co-Attention Network for Fake News Detection

Due at 3:00 pm, Monday, November 13, 2023

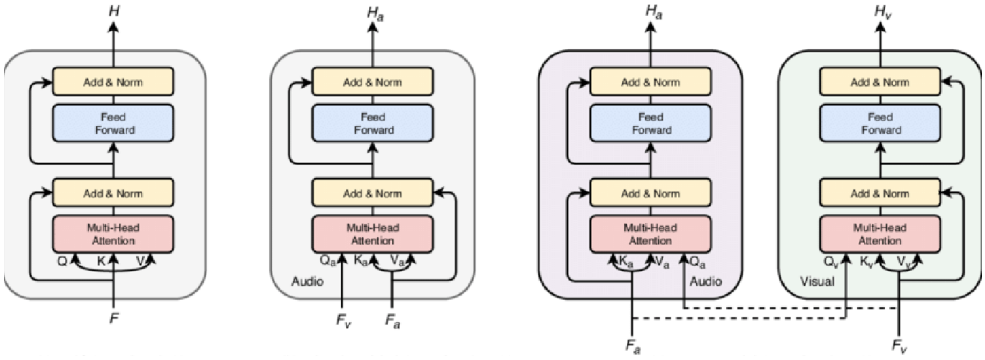
Demo: class time, Monday, November 13, 2023

1. Problem Formulation

This project builds on your project 3 but we use cross/co-attention mechanisms to improve the performance. In project 3, the fused/combined latent features learned ignore correlations across multiple modalities. To overcome this limitation, in this project, we propose to learn inter-dependencies among modalities using cross/co-attention mechanism. Concretely, we extract visual-domain and textual-domain features, and then **we fuse them through cross/co-attention mechanisms**.

Your task is to extend the model you achieved in Project 3 using self-attention, cross-attention, and co-attention. Compare those three models with your original model in Project 3

2. Self-Attention vs Cross-Attention vs Co-Attention



(a) Self-Attention (b) Cross-Attention (c) Co-Attention

If you need more hints, check the following layers:

- For implementing cross-attention and co-attention in Tensorflow: you may use https://keras.io/api/layers/attention_layers/multi_head_attention/

- For implementing cross-attention and co-attention in PyTorch: you may use https://pytorch.org/docs/stable/generated/torch.nn.MultiheadAttention.html

3. Requirements

- You may choose to use either Tensorflow or PyTorch.
- To encode the text, use pre-trained GloVe embedding. Check our lab tutorial on transfer learning using GolVe.
- Make sure you match images with their corresponding texts.
- Use EarlyStopping when training neural networks using Tensorflow.
- NO need to do any hyper-parameter tuning.
- Print out Recall, Precision, and F1 score on test data for real and fake news, respectively, for each of the four models.

4. Grading Breakdown

You may feel this project is described with some certain degree of vagueness, which is left on purpose. In other words, **creativity is strongly encouraged**. Your grade for this project will be based on the soundness of your design, the novelty of your work, and the effort you put into the project.

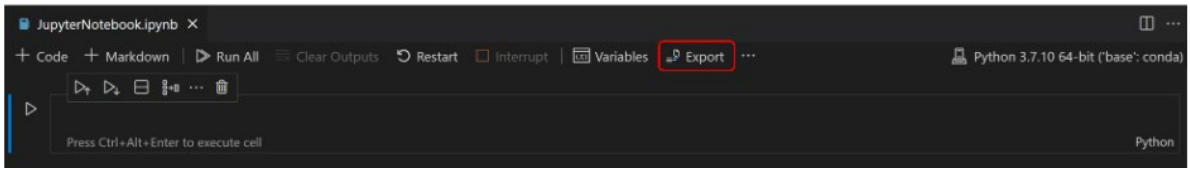
Use the evaluation form on Canvas as a checklist to make sure your work meet all the requirements.

5. Teaming

Students must work in teams of 3 people. Think clearly about who will do what on the project. Normally people in the same group will receive the same grade. However, the instructor reserve the right to assign different grades to team members depending on their contributions. So you should choose partner carefully!

6. Deliverables

(1) The HTML version of your notebook that includes all your source code. In VS Code, you can export a Jupyter Notebook as an HTML file. To export, select the Export action on the main toolbar. You'll then be presented with a dropdown of file format options.



(2) Your report in PDF format, with your name, your id, course title, assignment id, and due date on the first page. As for length, I would expect a report with more than one page. Your report should include the following sections (but not limited to):

- Problem Statement
- Methodology
- Experimental Results and Analysis
- Task Division and Project Reflection
- Additional Features

In the section "Task Division and Project Reflection", describe the following:

- who is responsible for which part,
- challenges your group encountered and how you solved them
- and what you have learned from the project as a team.

In the section "Additional Features", you describe and claim credit for additional features.

To submit your notebook and report, go to Canvas "Assignments" and use "Project 4".

All the deliverables must be submitted **by team leader** on Canvas before

3:00 pm, Monday, November 13, 2023

NO late submissions will be accepted.

7. Possible Additional Features (10 points each feature)

- Can you try transfer learning for the CNN input channel? https://keras.io/api/applications/ There are some models you may want to consider (try at least one model as listed here)
  - VGG16
  - ResNet50
  - MobileNetV2

- Use contextual word embedding like BERT to represent the text for the LSTM input channel. Here is a nice introduction: https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/ You may use any high-level implementation like the following:
  - https://www.sbert.net/docs/quickstart.html
  - https://github.com/llSourcell/bert-as-service

- Can you borrow any idea from this paper to further improve your models?

  Multimodal Fusion with Co-Attention Networks for Fake News Detection by Wu et al.
  https://aclanthology.org/2021.findings-acl.226.pdf

8. In-class Presentation.

On the due day, each team has 5 minutes to present your work in the class. Explain your solutions by referring to your notebook. You do not have to prepare the PowerPoint slides for your presentation.