CSE481 NLP Capstone: Blog Post 1

Team name: Bitmaps - **BI**narized **T**ransformers for **M**aking f**A**st **P**rediction**S**
List of members: Tobias Rohde, Anirudh Canumalla
GitHub URL: nlp-capstone

Project 1: Binarized Transformers

In recent years, Transformer neural networks have led to SOTA results in a wide variety of NLP tasks. Still, Transformers have high memory usage and are not fast enough to be run in real-time on edge devices. The goal of this project is to speed up the inference time and reduce the memory usage of transformers by using binary weights. This project was inspired by the XNOR-Net paper (XNOR-Net), in which the authors have binarized convolutional neural networks leading to a reduction in memory usage by a factor of 32 and a speed up by a factor or 58 (in the most extreme case of binarization). Of course binarization comes at a performance cost and the main challenge will be to binarize the transformer while still maintaining high performance. To our knowledge there are currently no publications on binarizing transformers. This makes for a novel and very promising capstone project. If successful, we believe transformer NLP models could run much more cost-effectively on edge devices in real-time.

To approach this project, we would start by finding an existing Transformer implementation. We are planning on using either BERT or RoBERTa. Arguably the most popular implementation is by huggingface (huggingface/transformers). We would start by simply taking BERT and converting weights to binary values using the sign function (In the XNOR-Net paper the authors report that using -1 and 1 rather than 0 and 1 works significantly better). We don't expect this to work well, but it would serve as a baseline and help us become familiar with the huggingface implementation. Next we would start working on binarizing each of the Transformer components: Multihead Attention, PositionalEncoding, Linear layer, etc. Splitting up the task into binarizing the submodules will likely help simplify the binarization. A key question will be to determine if binarizing a single component can still lead to good overall performance or if all components need to be binarized to retain good performance. In the latter case, we need to carefully consider how to evaluate the effect of binarizing individual components. As in the XNOR-Net paper, we are planning on trying to find the optimal way of binarizing the weights in order to approximate the full precision weights. We will also have to adjust the training procedure. This will be the core of our work and determine if the project is viable. To determine the performance of the binarized Transformers, we will evaluate it on the language modeling task, since it is very simple and should easily indicate how detrimental our binarization approach is to the performance.

If time allows, there are several ideas we would like to explore. Firstly, we would like to implement other SOTA models using our binarized transformer and compare the performance against the original, non-binarized version. We would like to make this comparison for different models and different NLP tasks, such as QA or machine translation. For the comparisons, we would not only compare performance metrics, but also inference time, FLOPs, and memory usage. We would also like to try running our binarized transformer model on an edge-device, such as a Jetson Nano or Raspberry Pi to see how well it performs in a real-world scenario.

Project 2: Text to Image Generation

In the past, many papers related to combining computer vision and natural language processing have focused on taking an image as the input and generating text from it. For instance, in image captioning, a caption is generated describing a given image. In visual question answering, an image and question are given as input and an answer is generated in text form. It seems like fewer papers reverse the directionality, taking only text as input and generating an image. This is often referred to as the text-to-image generation task and while current state of the art approaches are impressive, there is room for improvement. Most current approaches use GANs for this task. Our goal for the project would be to reproduce one of the SOTA models for this task. Papers we are considering are StackGAN++ (StackGAN++ Paper) and AttnGAN (AttnGAN Paper) or the newer, yet unpublished version of AttnGAN, OP-GAN (OP-GAN paper).

Our minimal viable action plan for this project is simple, since the main goal is to reproduce one of the aforementioned papers. We think that for the text-to-image generation tasks it makes the most sense to start with a paper reproduction project, since the state of the art architectures for this task involve many different components including GANs and models from both computer vision and natural language processing. This makes it a difficult task and training/fine-tuning all of the different components will be time consuming in itself.

However, if time allows, we have several ideas for how one could improve current approaches. To our knowledge, no current published approaches (including SOTA) use Transformers, which is something we would be interested in trying. Furthermore, we noticed that all current datasets only contain very short image descriptions, sometimes only single words, which are used to generate images. Providing longer and more detailed image captions as inputs could lead to higher quality generated images. This would involve building a new dataset.

Project 3: Better Methods for Commonsense Reasoning

Our third idea is to develop a method for commonsense reasoning that outperforms the current SOTA model on the Winogrande challenge. Until recently, the standard dataset for evaluating the commonsense reasoning abilities of natural language processing models was the Winograd Schema Challenge (WSC paper). The current state of the art approach is from the paper “A Simple Method for Commonsense Reasoning”. In the paper, the authors simply use a pre-trained language model and devise a simple scoring mechanism to choose a response. A key downside of the Winograd Schema Challenge is that the set of questions is very small, and SOTA models achieving high accuracy are likely susceptible to biases in the questions (which were all written by a single team). Winogrande dramatically improves upon WSC by providing a crowdsourced data bank of questions for commonsense reasoning that uses the AFLite algorithm to throw away poorly constructed questions. Since Winogrande is a new database, we believe there is value in reproducing and modifying previous approaches to WSC and applying them to the newer and more difficult Winogrande.

Our minimal viable action plan is to take an open-source transformer language model (like GPT2) and evaluate its accuracy on Winogrande by using both the partial and full scoring mechanisms as specified in Simple Method for Commonsense Reasoning. We want to devise more complex scoring mechanisms, which make better use of the language model’s latent information. We believe that potential exists in finding better ways to extract knowledge from language models for scoring questions in a Winograd schema format. For example, consider how in natural language generation not all sampling strategies work equally well. Even though we’re not changing the underlying language model, a simple choice of nucleus sampling vs. beam search has a dramatic impact on language generation results. If time allows, we would also like to explore transfer learning for our best devised methods to other commonsense reasoning datasets.