Homework 5 (100 points)

CNNs, AEs, GANs, Attention Mechanism

The homework will be due in 3 weeks from the day of release. You may divide your work into multiple notebooks (for each task). Submit links to all notebooks for full credit.

You need to perform the following tasks for this homework:

In your project, you will pick an image dataset to solve a classification task. Provide a link to your dataset. You may pick any dataset except **MNIST, CIFAR or ImageNet**.

Task 1 (30 points):

Part 1 (10 points): This step involves downloading, preparing, and visualizing your dataset. Create a convolutional base using a common pattern: a stack of Conv and MaxPooling layers. Depending on the problem and the dataset you must decide what pattern you want to use (i.e., how many Conv layers and how many pooling layers). Please describe why you chose a particular pattern. Add the final dense layer(s). Compile and train the model. Report the final evaluation and describe the metrics.

Part 2 (10 points): The following models are widely used for transfer learning because of their performance and architectural innovations:

1. VGG (e.g., VGG16 or VGG19).
2. GoogLeNet (e.g., InceptionV3).
3. Residual Network (e.g., ResNet50).
4. MobileNet (e.g., MobileNetV2)

Choose any **one** of the above models to perform the classification task you did in Part 1. Evaluate the results using the same metrics as in Part 1. Are there any differences? Why or why not? Describe in detail.

Part 3 (10 points): Use data augmentation to increase the diversity of your dataset by applying random transformations such as image rotation (you can use any other technique as well). Repeat the process from part 1 with this augmented data. Did you observe any difference in results? Why or why not?

Task 2 (15 points):

Part 1 (7 points): Variational Autoencoder (VAE): Here is a complete implementation of a VAE in TensorFlow: https://www.tensorflow.org/tutorials/generative/cvae

PyTorch implementation is fine too.

Following these steps try generating images using the same encoder-decoder architecture using a different Image dataset (other than MNIST).

Part 2 (8 points): Generative Adversarial Networks (GANs): Repeat part 1 (use same dataset) and implement a GAN model to generate high quality synthetic images. You may follow steps outlined here: https://www.tensorflow.org/tutorials/generative/dcgan

Task 3 (55 points): NLP and Attention Mechanism

Part 1 (10 points): Implement the **scaled dot-product attention** as discussed in class (lecture 14) from scratch (use NumPy and pandas only, no deep learning libraries are allowed for this step).

Part 2 (10 points): Pick any **encoder-decoder seq2seq** model (as discussed in class) and integrate the scaled dot-product attention in the encoder architecture. You may come up with your own technique of integration or adopt one from literature. **Hint:** See Bahdanau or Luong attention paper presented in class (lecture 14).

Part 3 (5 points): Pick any public dataset of your choice (**use a small-scale dataset like a subset of the Tatoeba or Multi30k dataset**) for machine translation task. Train your model from Part 2 for the machine translation task. Evaluate test set by reporting the BLEU Score.

Part 4 (30 points): In this part you are required to implement a **simplified Transformer model** from scratch (using Python and NumPy/PyTorch/TensorFlow with minimal high-level abstractions) and apply it to a machine translation task (e.g., English-to-French or English-to-German translation) using the same dataset from part 3.

We discussed Transformer architecture in depth in class (Vaswani Paper – Attention is all you need). Apply the following simplifications to the original model architecture:

1. **Reduced Model Depth:** Use **2 encoder layers** and **2 decoder layers** instead of the standard 6.
2. **Limited Attention Heads:** Use **2 attention heads** in the multi-head attention mechanism rather than 8.
3. **Smaller Embedding Size:** Set the **embedding dimension to 64** instead of 512.
4. **Reduced Feedforward Network Size:** Use a **feedforward dimension of 128** instead of 2048.
5. **Smaller Dataset:** Use a **small dataset** (e.g., about 10k sentence pairs).
6. **Tokenization Simplifications:** Use a **basic subword tokenizer** (like Byte Pair Encoding - BPE) or word-level tokenization instead of complex language-specific tokenizers.

Key components to implement:

1. Positional Encoding: Implement Sinusoidal position encoding.

2. **Scaled dot-product attention**: Use the same implementation from part 1.

3. **Multi-Head Attention**: Integrate the scaled dot-product attention into a multi-head attention framework using the specified simplifications.

4. **Encoder and Decoder Blocks**: Implement simplified encoder and decoder layers, ensuring: Layer normalization, Residual connections, Masked attention in the decoder for autoregressive generation.

5. **Final Output Layer**: Implement a linear layer followed by a SoftMax activation for generating translated tokens.

Evaluation: Compute the BLEU score on a validation set and compare the performance with your model from part 2. Explain why there are differences in performance. Also discuss any other differences you notice, for example runtime etc.

Project Progress Report (This is not graded)

Please submit a report detailing your progress on the final project. This can be a 1 (maximum 2) page (word or pdf) long description of your data-collection/modelling/preliminary results related tasks. Also, describe the next steps towards your final goal.

Task for 6000 level (Graduate level only): 100 points

Medical Image Segmentation is an important problem in healthcare domain. Polyp recognition and segmentation is one field which helps doctors identify polyps from colonoscope images. CVC-Clinic database consists of frames extracted from colonoscopy videos. The dataset contains several examples of polyp frames & corresponding ground truth for them.

The Ground Truth image consists of a mask corresponding to the region covered by the polyp in the image. The data is available in both .png and .tiff format here: https://polyp.grand-challenge.org/CVCClinicDB/

Consider this task as a minor research project in which you should research the existing models used (https://paperswithcode.com/dataset/cvc-clinicdb) to identify polyps from these images. Report on the key findings and the evaluation metrics used for this problem. Variants of the Unet architecture are often used to solve this problem. Implement either Unet or any of its variants (Unet++, ResUnet etc.) to segment the polyp images. This may be a computation intensive task (requiring GPUs). In case you do not have access to GPUs simply reduce your training data size to train your model. Report your results, compare and contrast these results with at least 2 of the other research paper results.