SC4001 CE/CZ4042: Neural Networks and Deep Learning

Group Project

Last Update: 08 Oct 2024

Start Date: 11 October 2024 Deadline: 15 November 2024 (11:59 PM)

Students are to propose and execute a final project on an application or a research issue that is related to neural networks and deep learning. The project can be carried out in a group consisting of no more than three members. Students are to come up with a potential technique for the application or to mitigate the issue, to develop associated codes, and to compare with existing methods. Students may choose, focus, and expand on the project ideas **A – F** given below.

By the deadline, students are to submit a project report in a .pdf file of **ten A4 pages** (Arial 10 font) and associated **code** in a .zip file to NTULearn.

The project report should have the names of the team members on the front page and contain an introduction to the project idea, a review of existing techniques, a description of the methods used, experiments and results, and a discussion. The 10-page limit is exclusive of references, content page, and cover page. The code needs to be commented properly. Make sure the code can be tested easily.

The assessment is based on the project execution (30%), experiments and results (30%), report presentation (15%), and novelty (15%), and peer review (10%). We apply the same late submission penalty as in Assignment 1, i.e., 5% for each day up to three days.

A. Deep Learning for ECG Heartbeat Classification

This task challenges you to explore the fascinating world of deep learning for healthcare by tackling the task of **ECG heartbeat, which can reveal the health of a person's heart**. You will be working with a dataset of ECG signals, each representing a single heartbeat categorized into different classes based on their underlying rhythm (normal or various types of arrhythmias).

Interesting projects would be:

Develop a deep learning model that can accurately classify heartbeats from ECG signals. You are encouraged to explore different neural network architectures and techniques, going beyond simple feedforward networks. Consider experimenting with:

- **Recurrent Neural Networks (RNNs):** Can you leverage the sequential nature of ECG data to improve classification?
- **Convolutional Neural Networks (CNNs):** How effective are CNNs at extracting relevant features from the ECG waveforms?
- **Hybrid Architectures:** Can you combine the strengths of RNNs and CNNs, or explore other advanced architectures?

- **Data Augmentation:** How can you augment the training data to improve model robustness and generalization?
- **Explainability:** Can you provide insights into which features your model is using to make its classifications?

Datasets:

- ECG Heartbeat Categorization Dataset: https://www.kaggle.com/datasets/shayanfazeli/heartbeat
- ECG Arrhythmia Classification Dataset: https://www.kaggle.com/datasets/sadmansakib7/ecg-arrhythmia-classification-dataset

References:

1. Mohammad Kachuee, Shayan Fazeli, and Majid Sarrafzadeh. "ECG Heartbeat Classification: A Deep Transferable Representation." arXiv preprint arXiv:1805.00794 (2018).
   Goldberger, A., Amaral, L., Glass, L., Hausdorff, J., Ivanov, P. C., Mark, R., ... & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation [Online]. 101 (23), pp. E215–e220.

B. Deep Learning for Protein Structure Prediction

Proteins are like the "programs" that run biological processes in living organisms. Their ability to function depends on how their "code" (the sequence of amino acids) folds into a specific 3D structure. Protein Secondary Structure Prediction (PSSP) involves predicting the local structural elements (like helices, strands, and loops) of a protein based on its primary amino acid sequence. Traditionally, techniques like X-ray crystallography or NMR are used to solve the protein’s 3D structure, and from this, tools like DSSP assign secondary structure elements. However, these experimental methods are costly and time-consuming.

The goal of this assignment is to predict the secondary structure (*sst3* and *sst8* values) from just the primary sequence (*seq*) using deep learning techniques, which can significantly reduce the need for expensive lab work. In this task, secondary structure can be classified into eight categories (Q8) or simplified into three states (Q3), which offers different levels of granularity in prediction.

Interesting projects could involve:

1. Developing deep learning techniques (such as RNNs, CNNs, or transformers) to predict the Q3 and Q8 secondary structures from the protein sequence. This will test your model's ability to handle both short- and long-range dependencies in the amino acid sequence.
2. Creating models that focus on improving Q3 and Q8 prediction by exploring novel architectures or feature representations.

Dataset:

Protein Secondary Structure (labels are sst3 and sst8):
https://www.kaggle.com/datasets/alfrandom/protein-secondary-structure

References:

1. Baldi, Pierre, Søren Brunak, Paolo Frasconi, Gianluca Pollastri and Giovanni Soda. "Bidirectional Dynamics for Protein Secondary Structure Prediction." Sequence Learning (2001). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.7092&rep=rep1&type=pdf

2. Chen, J. and Chaudhari, N. S.. "Protein Secondary Structure Prediction with bidirectional LSTM networks." Paper presented at the meeting of the Post-Conference Workshop on Computational Intelligence Approaches for the Analysis of Bio-data (CI-BIO), Montreal, Canada, 2005. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.7092&rep=rep1&type=pdf
3. Sepp Hochreiter, Martin Heusel, Klaus Obermayer; Fast model-based protein homology detection without alignment, Bioinformatics, Volume 23, Issue 14, 15 July 2007, Pages 1728–1736, https://doi.org/10.1093/bioinformatics/btm247

C. Sentiment Analysis

Text sentiment analysis (TSA) refers to identification of sentiments, usually positive or negative, expressed in text or document. One may want to develop deep learning techniques for TSA

1. To deal with domain adaptation, that is, how can one adapt a network train on one domain to work in another domain
2. To compare the performance of different Transformers architectures
3. To deal with small datasets, that is, with insufficient number of training samples

References:

1. T. Gui *et al.*, "Long Short-Term Memory with Dynamic Skip Connections," *Proc. AAAI Conf. Artif. Intell.*, 2019, doi: 10.1609/aaai.v33i01.33016481.
2. A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.
3. X. Zhang, J. Zhao, and Y. Lecun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, 2015.

Datasets:

1. Stanford Sentiment Treebank: https://www.kaggle.com/atulanandjha/stanford-sentiment-treebank-v2-sst2
2. IMDB movie review dataset: https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews
3. YELP review dataset: http://xzh.me/docs/charconvnet.pdf

D. Automated Image Captioning with CNNs and Transformers

The objective of this project is to develop an automated image captioning system that generates natural language descriptions for input images. By leveraging Convolutional Neural Networks (CNNs) for image feature extraction and Recurrent Neural Networks (RNNs) or Transformer architectures for sequence modeling, the project bridges computer vision and natural language processing. Students will first implement a CNN-RNN model to generate captions and then enhance the system using Transformer-based architectures to potentially improve performance. Training will be conducted on datasets containing images paired with descriptive captions. The system's effectiveness will be evaluated using metrics such as BLEU, METEOR, and CIDEr. Additional tasks include experimenting with more advanced attention mechanisms, comparing different architectural choices, and optimizing hyperparameters for improved results.

Main Task: Develop an image captioning model using CNNs for feature extraction and RNNs or Transformers for caption generation.

Dataset:

- **MS COCO Dataset**: Contains over 120,000 images with five captions each.
    - URL: http://cocodataset.org/#download
- **Alternative**: Flickr8k or Flickr30k datasets for smaller scale.
    - Flickr8k URL: https://www.kaggle.com/datasets/adityajn105/flickr8k

Additional Tasks:

- Implement more advanced attention mechanisms to enhance caption quality.
- Compare the performance of RNN-based models with Transformer-based models.
- Experiment with different evaluation metrics and optimization techniques.

E. Artistic Style Transfer with Generative Adversarial Networks

This project aims to implement an artistic style transfer system that applies the style of one image (e.g., a painting) to the content of another image (e.g., a photograph) using Generative Adversarial Networks (GANs). Students will explore GAN architectures like CycleGAN to perform style transfer between domains without paired data. By training on datasets containing artworks and photographs, the system will learn to translate images from one style to another, enabling applications like turning landscape photos into Van Gogh-style paintings. The project involves data preprocessing, model training, and qualitative and quantitative evaluation of the generated images. Additional tasks include experimenting with different GAN architectures, and extending the model to support multiple styles.

Main Task: Implement a style transfer model using GANs to apply artistic styles to images.

Dataset:

- **WikiArt Dataset**: A large collection of artwork images.

    - URL: https://www.wikiart.org/
- **CycleGAN Datasets**: Datasets used in CycleGAN research.
    - URL: https://github.com/junyanz/CycleGAN

Additional Tasks:

- Compare different GAN architectures (e.g., CycleGAN, StyleGAN).
- Implement multi-style transfer capabilities.
- Analyze the impact of training parameters and data augmentation techniques.

F. Flowers Recognition

The Oxford Flowers 102 dataset is a collection of 102 flower categories commonly occurring in the United Kingdom. Each class consists of between 40 and 258 images. The images have large scale, pose and light variations. In addition, there are categories that have large variations within the category and several very similar categories. The dataset is divided into a training set, a validation set and a test set. The training set and validation set each consist of 10 images per class (a total of 1020 images each). The test set consists of the remaining 6149 images (minimum 20 per class). Some tasks to consider:

Main Task: Implement a classificatoin model to classify the flower images.

Datasets:

- The dataset is available in TorchVision https://pytorch.org/vision/main/generated/torchvision.datasets.Flowers102.html
- The Oxford Flowers 102 Dataset https://www.robots.ox.ac.uk/~vgg/data/flowers/102/

Additional tasks:

- Modify some previously published architectures e.g., increase the network depth, reducing their parameters, etc. Explore more advanced techniques such as deformable convolution or visual prompt tuning for Transformers.
- Analyze the results of using fewer training images, i.e., few-shot learning
- Use more advanced transformation techniques such as MixUp (see the original paper and its PyTorch implementation here)
- Try more advanced loss function such as triplet loss

Computational Resource

You can use the computational resources assigned by the course. Alternatively, you can use Google Colab for computation. Note that the free version of Colab has a session duration limit, after which the environment needs to be reset. This can be disruptive for long-running experiments or processes.