ECE1512 Project B Assignment Details

Samir Khaki (samir.khaki@mail.utoronto.ca)

Below we include the grade distribution for Project B.
The due date is currently set for 6 PM EST, December 10

Part 1&2 State Space Models (SSMs) ([60 Points])

1. Summary Document detailing: key points, technical contributions, areas for improvement **[15 points]**
2. Extension 1 (or 1 & 2 for groups) – Detailing either extensions (i.e applying to a new field) or accuracy/efficiency improvements. For whichever you choose, please include the following: how you would extend it, framework diagrams, tasks it would solve, etc. For example this can include Mamba being applied to a new field or concept like medical vision, chain of thought. Alternatively discuss some new techniques to improve efficiency or accuracy **[15 points]**
3. Extension 2 (or 3 for groups) – Same as above, except it includes the actual rudimentary experimental results. **[30 points]**

Relevant references for this part include:

1. Relevant SSM works [2, 8, 7, 3]
2. Main mamba paper [1]
3. Mamba Extensions: [10, 6, 11]

Part 3&4 Vision Language Models (VLMs) ([40 Points])

1. Summary Document on either Qwen2VL[9], VILA[4] or LLaVA[5]. Include key points, technical contributions, areas for improvement **[10 points]**
2. Efficiency: Discuss the efficiency bottlenecks of these architectures, and propose a new approach to circumvent this. This requires a carefully thought out method, with specefic details and implementations. You must also consider the implications of accuracy – although you are not required to measure end-to-end accuracy – you are required to provide some type of metric to determine the *lossy-ness* of your method. **[30 points]**

I suggest the following steps to best organize this section of the report:

- Identify the efficiency bottlenecks using some type of profiling or justification. This could be accomplished by measuring the time or flops distribution of different components in a block and/or citing previous works.
- Pick the item you would like to optimize, some suggestions could include, input-dependent reductions, channel pruning, kernel fusion etc.
- Desing an approach or method to optimize that component – here you would typically present algorithm pseudo code and a method figure.
- Finally measure your efficiency improvements, likely out of place (can be simulated with tensor ops) against some naive baseline. Likewise report considerations on your primary objective metric – information loss, entropy, accuracy on a toy/simulated dataset etc. One simple approach for this could be, take a few transformer blocks, attach a small learnable classifier head and train on mnist-like data or binary classification, this way you could report accuracy of a naive approach and your desired method easily on a simulated (toy) dataset while also showing efficiency improvements.

What to submit & Evaluation

1. You need to submit a written report (one per group) **in PDF format using the ECE1512 Quercus page submission utility.** Your report should be approximately **in IEEE style** (**one column format**). List any additional references you have used in your work. You need to provide the code, data, figures and any other auxiliary material) used in your work. To that end, your written report should include a link to your Project B GitHub repository. Since both parts of this project include code development, either separate the sections or provide multiple GitHub links.

2. **Project evaluation will include an examination of the code, notebooks, figures, and other auxiliary material posted on your GitHub repo.** It is highly recommended that you include a README document on your project's GitHub page.

3. **Ouriginal "plagiarism" scores** will be visible upon (report) submission, so make sure that you are not penalized for plagiarism in uncited text and code portions (if applicable).

4. Make sure your report submission is desigend to be used simultaneously with the Github-Repo. To do this, use **Machine Learning Paper Reproducibility Checklist** that clearly acknowledge that the report and the code are two separate artefacts, each with their own checklist.

5. **Late submissions will not be accepted.**

References

[1] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces, 2024.

[2] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces, 2022.

[3] Albert Gu, Ankit Gupta, Karan Goel, and Christopher Ré. On the parameterization and initialization of diagonal state space models, 2022.

[4] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models, 2023.

[5] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[6] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model, 2024.

[7] Eric Nguyen, Karan Goel, Albert Gu, Gordon W. Downs, Preey Shah, Tri Dao, Stephen A. Baccus, and Christopher Ré. S4nd: Modeling images and videos as multi-dimensional signals using state spaces, 2022.

[8] Jimmy T. H. Smith, Andrew Warrington, and Scott W. Linderman. Simplified state space layers for sequence modeling, 2023.

[9] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution, 2024.

[10] Changqian Yu Jusnshi Huang Zhengcong Fei, Mingyuan Fan. Scalable diffusion models with state space backbone. *arXiv preprint*, 2024.

[11] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model, 2024.