ECE661 Proposal
Project 2: Input-Dependent Dynamic CNN Model

Wei-Kai Liu, Chung-Hsuan Tung, Yi-Chen Chang
Huanrui Yang (Lead TA)

1) What are you trying to do?
   Implementing input-dependent dynamic CNN models by applying two techniques: dynamic convolution and dynamic channel gating, which provide dynamic kernel combination and dynamic filter selection, respectively.
2) How is it done today?
   The inputs of CNN models may have different features, which can vary between classes. Thus, dynamic CNN models uses the input-dependent features to increase the model computational efficiency. [2] proposed dynamic convolution, which combines several small-size kernels dynamically depending on inputs. This method significantly increases the model accuracy with only a small raise in computational costs. [1] introduced a new residual block architecture to gate convolutional channels. Thus, the filters are selected based on the input features during model inference time. This method achieves a higher model accuracy while the inference computational costs reduces or remains the same.
3) Your approach and why do you think it will be successful?
   We will use the dynamic convolution method as a substitute for the original convolution neural network (By Wei-Kai). By introducing the attention concept, k parameters $\pi_1 \sim \pi_k$ are utilized to aggregate k convolution kernels. In this case, we can get a more computationally efficient model but has more representative power as we aggregate these small kernels in a non-linear way. Also, by applying the channel gated networks, the model can choose to turn on/off each filter to reduce FLOPs without shrinking the model (By Chung-Hsuan and Yi-Chen). With the additional flexibility provided by these two methods, we believe that we can generate better results than the original convolution network. Lastly, we will try to perform PGD adversarial or other kinds of attacks on both [2] and [1] and observe the differences and performance compared to the original model (By all).
4) What are the risks?
   Training the attention weights for each kernel in [2] may be difficult. If the number of convolution kernels is larger, the model may take more time to converge. The complexity implicitly indicates a longer training time. The batch-shaping loss in [1] may be difficult to implement and also need more training time.
5) How long will it take?
   - Week 11/8: Read papers and start to implement.
   - Week 11/15: Finish basic implementations. For [2], trained with 2 to 5 kernels.
   - Week 11/22: For [2], visualize the attention map. For [1], visualize the gate distribution. Read materials about the optional requirement.
   - Week 11/29: Implement the optional requirement.
   - Week 12/6: Collect the results and write reports.
6) What are the final "exams" to check for success?
   - [2] Plot accuracy (both training and validation) versus kernel numbers $k = 2, 3, 4, 5$.
   - [2] Visualize the attention map distribution for selected testing images and show the similarity based on classes with T-SNE.
   - [1] Report the accuracy and calculate the averaged FLOPs. The result should show that the model uses fewer FLOPs than the original ResNet-20 model.
   - [1] Visualize the channel usage and show the usage pattern is similar for similar classes.

References

[1] Babak Ehteshami Bejnordi, Tijmen Blankevoort, and Max Welling. "Batch-shaping for learning conditional channel gated networks". In: *International Conference on Learning Representations*. 2020. URL: https://openreview.net/forum?id=Bke89JBtvB.

[2] Yinpeng Chen et al. "Dynamic convolution: Attention over convolution kernels". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11030–11039.