Paper: *The Game Imitation: Deep Supervised Convolutional Networks for Quick Video Game AI*

Nirajan Koirala

CSC 9010_001

Summary

Authors show that their methods provide reasonable performance for video game tasks when compared with deep-Q learning however at a significantly lower cost in terms of hardware, time and fine tunings at inference. They propose a deep CNN on the task of playing Super Smash Bros. and the input data is gameplay frames. Ground truth values are generated from a human player. They mention that they chose this game because of its complexity in 3D camera angles and random zooms. Their method is not however appropriate for all cases, like other games, Mario Tennis dataset to investigate what parts of model are portable cross-game and what limitations such models might face in random scenarios. Their work provides evidence on how CNNs produce expressive features which are exceedingly robust on image datasets. They allow the supervised model direct access to temporal information and by doing this expect the network to be not strictly limited to examples it sees in the training dataset but also be able to generalize well in the unfamiliar situations.

Section 3 provides the description of the problem where the input dimension of the images is (4, 128, 128, 3) since the classification pipeline takes as input a concatenation of 4 temporally sequential gameplay images. Output is a softmax function containing 30 classes. During inference, they also take in account along with correct prediction button presses, the evaluation of the performance of neural network in a real game setting. Data collection is done through emulation and screen capture tools. They provide further description of the data collection techniques in section 4. For the image preprocessing, they downsample each frame of the video from 344x246 to 128x128 pixels using nearest neighbor interpolation in order to preserve stronger colors of the images. In section 5 they discuss about the model and methods used in training.

They use 3 main CNN architectures: single frame CNN, a CNN with early integration of video frames and a CNN with late integration of video frames. For the single frame CNN, they use a vanilla AlexNet structure and it is able to turn the supervised learning problem into a simple single image classification problem. This model however ignores the temporal information of the data set in the model which leads to some issues related to supervised imitation learning. The early integration of CNN incorporates the temporal information by concatenating all frames at the onset of the network. Same layers of single frames network are used but only the first layer is changed to 128x128x3F where F is the number of frames which is 4 in their case. Also the frames are 1/6 s apart which provides the network with approximately twice as much information as the human reaction limit. One of the major problems of this CNN model is that only the first convolutional layer has access to the temporal information in a separable form and the subsequent layers having no access to the temporal information are not able to properly learn the temporal features.

For the late integration model of CNN, instead of concatenating the input data before the first layer, they send each frame through its own independent layers with untied weights before merging the activations at the end through a series of dense layers. Also, the dense layers which consolidate the activations from the independent CNN branches have same architecture as of layers 16-20 in the early integration model. In section 5.4, training of the models is described where they perform hyperparameter search on learning rate and L2 regulaization penalty with a small number of trials. They find that the model training efficacy is mostly dependent on learning rate and they settle on training learning rate with 5 other parameters like batch size, L2 regularization penalty etc. Also, they mention that no data augmentation techniques were employed and data was only generated in the limited confines of game emulator. Since the collected dataset is very imbalanced, one of the ways of dealing it is through batch selection or weighted loss function. However, they found that a simple post-processing step that adds scaler weights to class scores produces good test time performance and it helps them avoid retraining the network with additional hyperparameter search. They bias each softmax class score by a simple multiplier which is chosen to ensure that each class is expressed sufficiently during inference.

The results are provided in section 6 and using the confusion matrix heat-map they found that some classes induce the most misclassifications. After 2 epochs of training, they end up with about 96% accuracy for Super Smash Bros. and Mario tennis for top 3 accuracy scores. During live gameplay also, the model is able to perform well against higher level CPU opponents. The model is able to play effectively against higher level CPU AI but without any knowledge of game objectives. Since a supervised learning setting is imposed, the models are able to train faster on fewer data compared to deep-Q learning. They also found that the model might have some limitations as it showed degrading performance on Mario Tennis dataset. Inspite of the limitations, the models work relatively good for games like Super Smash Bros where actions are relatively localized in time and some prediction uncertainty is allowed. CNN also helps the model to generalize better as many of the states during live simulation are not exactly present in training set. Additionally model's live gameplay links are provided in section 6.6.

Hence, they are able to achieve comparatively good results using the supervised learning framework for complex games. They attribute the good generalization of their models to CNN as well as using late-integration for temporal data. Authors hope that the simplicity of their approach will help gather important insights into the generalizing power of deep CNNs on difficult vision problems.

Strengths

- The paper is well organized and provides a resource-light alternative to deep-Q-learning.
- The the description of their methods is really detailed.
- Video Demonstrations are also provided for the model’s play.

Weaknesses

- I think the section 5.4 which talks about the model training could have been made a little more descriptive.

Points of Confusion

- I am little confused about their method of dealing with class imbalance which is to bias each softmax class score by a simple multiplier b_{class}.
- In table 2, I am not sure why the variance of damage dealt scores are seen increasing even though damage dealt scores themselves are decreasing as the CPU level increases.

Discussion Questions

- How would these models generalize if one performs transfer learning for other similar games of similar genre?
- As they mention that one of the limitations of their models is long term memory, will incorporating LSTMs into these networks help them learn better?
- Authors also mention that color specificity is one of the reasons for model's limitation. Could playing black and white versions of these games help the model learn better?