CS 8803DL: Deep Learning Project
Mid-Term Progress Report

Casey Battaglino*, Min-Hung Chen†, Chih-Yao Ma‡, and Hao Yan§

1 Proposal

Motivation

Deep convolutional neural networks are a leading technique for image classification. However, *video* classification is less developed. While it is certainly possible to simply run a neural network on each individual frame of a video, this sacrifices a wealth of temporal attributes such as movement, gestures, gait, etc. There are indeed methods of preprocessing temporal information into a single 2-dimensional input [1], but a more attractive research goal is to develop a neural network that can discover temporal relationships on its own. There are multiple recent approaches towards this goal, but as of yet no consensus on which is superior.

The difficulties in training neural networks on video input include the following: memory requirements (particularly if 3D convolutions are used), fewer public data sets, size of the data sets (for instance, the Sports-1M data set is $\approx 4TB$ large), and lack of consensus on which approach is most effective.

Our goal is to develop a deep neural network that classifies videos. We have settled on two main approaches, discussed below: 3-dimensional convolutional neural networks (3D-CNNs) and recurrent neural networks that incorporate long short-term memory (LSTM).

Related Work

One popular approach is to apply a 2D-CNN to each frame of the video, followed by an RNN with LSTM [3]. Another approach is slow-fusion, which applies multiple frames to the input at the same time [5]. This approach, applied to a simply CNN, shows only a modest improvement over CNN single-frame learning. Tran, et al. demonstrated that using 3D-CNNs instead of 2D can achieve state-of-the-art results on several data sets [9]. Ng, et al. demonstrate that instead of training on 'short snippets' (such as in [5, 9]), an LSTM approach allows us to train on entire videos efficiently [7], and achieves state-of-the-art performance on several data sets.

Many of these papers incorporate additional features such as optical flow [1] and improved dense trajectory, both of which involve optimization techniques applied to subsets of frames.

Approach and Techniques

We have decided on combining two possible approaches, which address the third, temporal dimension in different ways: 3D Convolutional Neural Networks (3D-CNNs) [9, 5] and Multi-Dimensional Long Short-Term Memory (LSTM)/Recurrent Neural Networks (RNNs) [3].

Using a 3D-CNN, it is possible to build a classifier on a moving window of frames using 3D convolutions to extract useful local movement information. However, 3D-CNNs introduce a massive

*cbattaglino3@gatech.edu
†cmhungsteve@gatech.edu
‡cyma@gatech.edu
§yanhao@gatech.edu

memory blowup that may make training infeasible on GPUs. To address this, we would like to use the slow-fusion model [5] instead of 3D kernels. In addition, we also plan to apply 3D kernels to one or two layers to check the improvement if we don't meet the memory problem. On the other hand, an LSTM/RNN approach is able to build the classifier with long term memory using a much larger number of frames at once. Combining a slow-fusion CNN with LSTM would combine the local movement information with long-term memory, with possible gains in learning.

CNN + CNN (1D convolution) ==> Yan

Furthermore, the current LSTM model is built on the fully-connected layer, which does not conserve spatial information. We would like to examine the effectiveness of adding the LSTM network at earlier layers. If we apply LSTM directly after the convolution layer, the input to LSTM model is actually multi-dimensional tensor. To address this challenge, we would like to apply the multi-dimensional LSTM [2] to efficiently take advantage of the spatial information following an early convolution layer.

Data Set

We propose to begin with a well-established data set such as UCF-101 [8], which contains 13320 videos from 101 action categories. For early training purposes we may use only a small subset of these actions, such as a subset of 10 sports. The ultimate goal is applying our framework to Sports-1M [5], which contains around 1 million Youtube videos belonging to 487 categories. In this way, we can compare our methods with the current state-of-the-art methods.

Experimental Methodology

For training we plan to use the Jinx cluster at Georgia Tech. Each node is equipped with 2 nVidia Tesla M2090 "Fermi" GPU cards, and CPU nodes with large memory are available.

The following is a set of experiments that we may perform, given enough time:

- Move 2D-LSTM within convolution layers to see if this better incorporates spatial information.
- Incorporate slow fusion within a 2D-CNN to see if this input approach yields higher accuracy.
- Replace a 2D-CNN-LSTM network with a Shallow ResNet-LSTM network (with or without slow fusion)– to see if ResNet provides significant gains over the AlexNet used by existing studies.
- Study if slow-fusion *combined* with optical-flow input provides significantly better results than either approach on its own.

We may experiment with different strategies in normalization and augmentation, given enough time.

Tasks to complete (members)

data preparation by optical flow (Casey)

Build CNN as the spatial encoder (Min-Hung Chen, Hao Yan)

Before we encoder the temporal information into the final features, we need to encode spatial information first. In this task, we chose the 'Network-In-Network' [6] model as the CNN architecture, and forwarded all the frames to CNN to obtain feature vectors for these frames. Currently we used the database UCF-11 for the primary experiment. There were 11 classes and 100 videos for each class, and we only selected the first 57 frames (The shortest video has 57 frames). We extracted the features from the layer right before the last linear layer in the NIN network, which have 1024 dimension. Therefore, the final feature dimension for each video will be 1024 x 57, and we will forward the features to the next network to encode the temporal information.

To be complete: Compare different CNN architectures. Select lower frame rates instead of using all the adjacent frames and compare the results. Generate features from a more complicated database UCF-101.

RNN with LSTM as one temporal encoder (Chih-Yao Ma)

In this specific task, we aim to implement a RNN with LSTM as the baseline for performance comparison. This RNN is implemented from scratch using rnn library provided by Element-Research. In this RNN, the user can specify if he/she prefer to use linear layer, LSTM, or even GRU layers. By initializing the number of hidden size for each layers, one can create arbitrary stacked layers, i.e. stacked LSTMs by $'4096, 800, 200'$. The user can also design how the learning rate decay through the training process by using command line arguments.

To be complete: Implement CUDA usage, cross-validation, and take feature vectors extracted from CNN and experiment how different parameters and architectures affect training performance.

CNN as another temporal encoder (Min-Hung Chen)

According to the success of modeling languages using CNN architectures [10, 4], we plan to model the temporal information across different frames using CNN architectures as well, and compare the performance and results with the previous RNN architecture. We plan to design the network based on the language models [10, 4], and compare the results between 1D and 2D convolution.

To be complete: design CNN architectures using 1D and 2D convolution kernels as temporal encoders. Test different parameters for learning. Implement the CUDA version of the encoder for acceleration.

Group Tasking

The following is a tentative list of tasks. As we move closer to our experiments the list will undoubtedly grow.

- Proposal writing - Min-Hung Chen, Hao, Casey
- Proposal presentation - Chih-Yao Ma
- Read and understand papers on 2D-LSTM - Everyone
- Implement the temporal encoder with LSTM: Chih-Yao Ma, Casey
- Implement the spatial and temporal encoder with CNN: Min-Hung Chen, Hao
- Preparation of UCF data (incl. optical flow) - Casey

References

[1] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):500–513, 2011.

[2] W. Byeon, T. M. Breuel, F. Raue, and M. Liwicki. Scene labeling with lstm recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3547–3555, 2015.

[3] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, abs/1411.4389, 2014.

[4] B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences. *CoRR*, abs/1503.03244, 2015.

[5] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1725–1732, Washington, DC, USA, 2014. IEEE Computer Society.

[6] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[7] J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. *CoRR*, abs/1503.08909, 2015.

[8] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.

[9] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *CoRR*, abs/1412.0767, 2014.

[10] Y. Zhang and B. Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *CoRR*, abs/1510.03820, 2015.