Internship Progress Report

Le-Vu Tran

September 7, 2018

Contents

Chapter 1

Week 14 - September 6 2018

1.1 Introduction

I have spent the majority of my time this week improving my Long - Short Term Memory network (PyTorch version), trying to extract videos' features by using TensorHub Module[69] and TensorFlow for Poets[70]. As mentioned before, till this week I would only work with the Short-term subtask so all my experiments from this moment were performed on the Short-term memorability scores.

1.2 Feature Extracting

1.2.1 Using TensorHub

In week 9, I had already try to extract images' features but at that moment I misunderstood the concept of how TensorFlow actually worked. After learning more about how to use TensorFlow this week I finally figured out how to use TensorHub to extract features correctly.

I used TensorHub to extract images' features with pretrained Inception V3 network. As mentioned by TensorHub, the checkpoint exported into the module was inception-v3-2016-08-28/inception-v3.ckpt downloaded from TensorFlow-Slim's pre-trained models. Its weights were originally obtained by training on the ILSVRC-2012-CLS dataset for image classification (Imagenet).

All my code could be found on my Colab notebook. I also normalized the input image by subtracting it by ***128*** then dividing it by ***128*** (IMPORTANT step). This module took me more than one hour to extract 8000 videos (8 frames for each video), which were around 64000 frames total.

1.2.2 Using TensorFlow for Poets

TensorFlow provided a tutorial called TensorFlow for Poets teaching how use transfer learning, which meant starting with a model that had been already trained on another problem, then retrain it on a similar problem.

At the 4th step Retraining the network, they described correctly how their retrain script worked. I changed the architecture argument to Inception V3 to use it as pretrained model to extract bottleneck values.

One of the very first phase of their retrain script was to analyzed all the images and calculated the bottleneck values for each of them. The term bottleneck here was actually indicated the term feature. After this phase I got a folder containing all images' bottleneck values, the next job I had to do was concatenating them all to work later.

1.3 PyTorch LSTM Classification Network

1.3.1 Architecture

Beside using the same architecture as I had used in week 9, I added 2 Fully-Connected Layers separately to initialize the hidden state ***h0*** and cell state ***c0*** of the LSTM cells based on the input. The motivation behind this change was

when I thought using a randomly initialized hidden state and cell state was nothing but a terrible idea because it made no sense at all.

These 2 Fully-Connected Layers I mentioned above turned the input size of [batch-size, sequence-length, input-size] (e.g. [1000, 8, 2048]) to hidden state and cell state sizes of [num-layers, batch-size, hidden-size] (e.g. [3, 1000, 1024]).

1.3.2 Training on Dev-Set (extracted by Inception V3 with TensorFlow for Poets)

Strategy

Till this moment I would use this strategy for every new networks. I splitted the provided Dev-Set for this challenge into three parts, since the Dev-Set had 8000 videos, I picked ***6000*** videos for training, ***1000*** videos for validating and the last ***1000*** videos for testing.

Version 2.1

As described above in the architecture section, I used my version 2 architecture powered by some Fully-Connected Layers so I called this 2.1. I trained this architecture with only ***100*** epochs and at the learning rate of ***1e-4***. My network converged with only around 50 epochs and the loss value seemed to decreasing stably.

The best correlation value on Validate set was ***0.4757*** and the minimum loss value was ***0.011***. As the loss would continue decreasing, the correlation value also decreased, so I thought it was over-fitting.

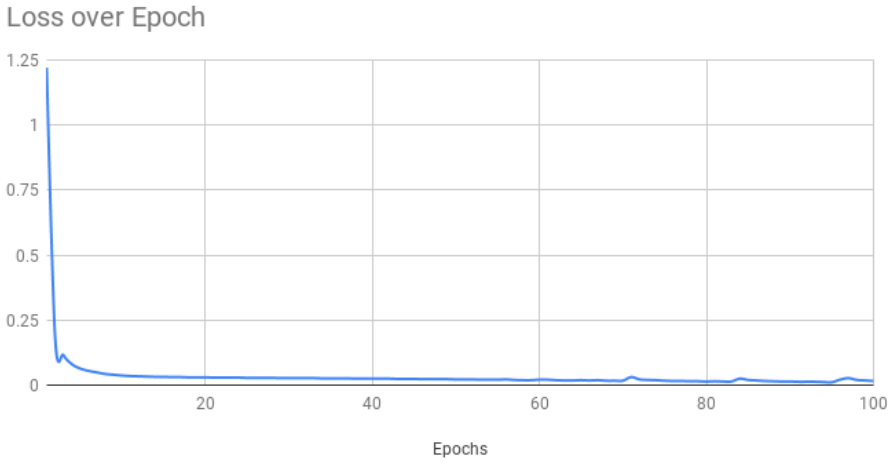

Figure 1.1: Loss over Epoch (on Train set).

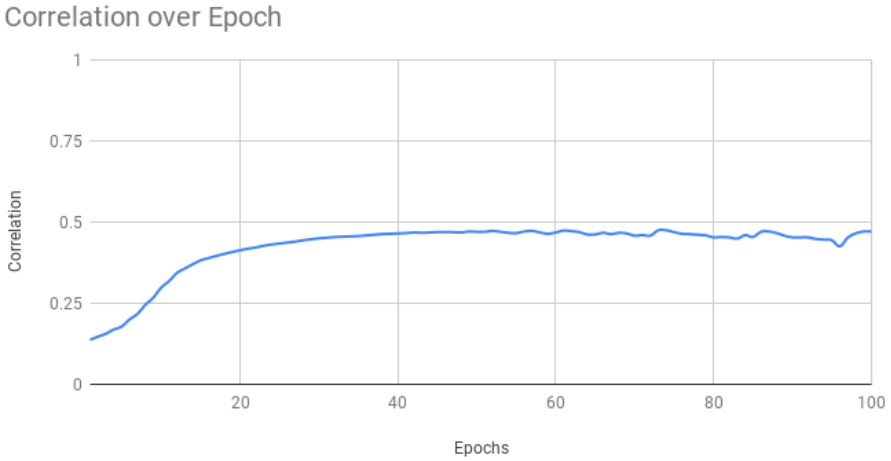

Figure 1.2: Correlation over Epoch (on Validate set).

Lately, I used this pre-trained model to calculate the correlation value on the test, the correlation was a little bit lower but acceptable, which was ***0.4591***.

1.3.3 Training on Dev-Set (extracted by Inception V3 with TensorHub)

Version 2.1

I trained this architecture with only ***100*** epochs and at the learning rate of ***1e-4***.

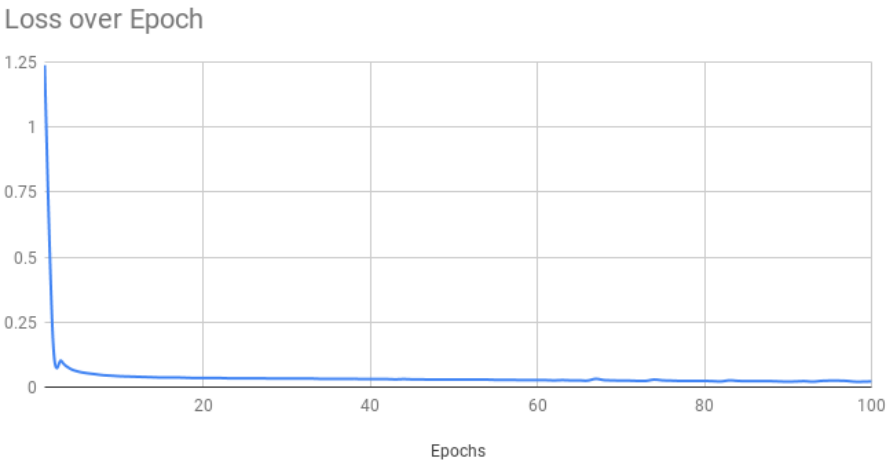

Figure 1.3: Loss over Epoch (on Train set).

The best correlation value on Validate set was ***0.3585*** and the minimum loss value was ***0.021***. As the loss would continue decreasing, the correlation

value also decreased, so I thought it was over-fitting.

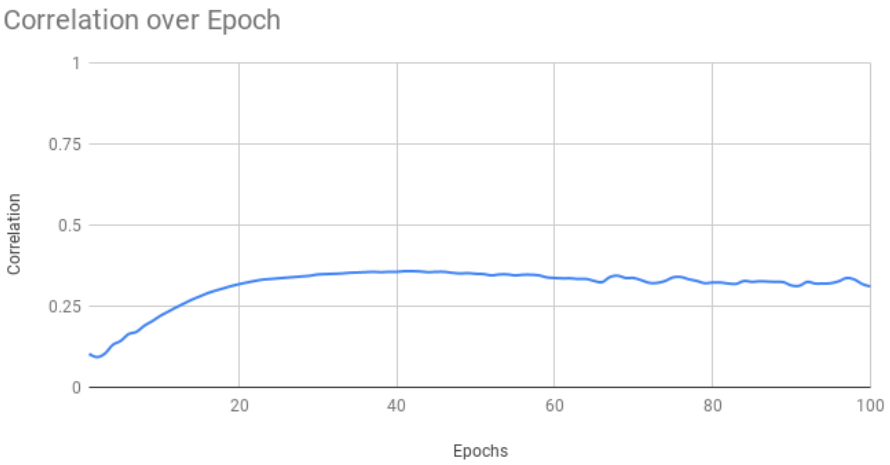

Figure 1.4: Correlation over Epoch (on Validate set).

Then I used this pre-trained model to calculate the correlation value on the test, the correlation was a little bit lower but acceptable, which was ***0.2989***.

Comment

I did not normalized the images before extracting their feautures in the previous week, and the value of correlation on Validate set was only 0.2x; the value of correlation on Test set was even worse (0.01x).

In this week I tried normalizing the images and my features performed much more better. The results was described above in the sub section above.

Bibliography

[1] École Polytechnique Fédérale de Lausanne, *EE-559 - Deep Learning.*

[2] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, *Places: A 10 million Image Database for Scene Recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

[3] Jia, Yangqing and Shelhamer, Evan and Donahue, Jeff and Karayev, Sergey and Long, Jonathan and Girshick, Ross and Guadarrama, Sergio and Darrell, Trevor, *Caffe: Convolutional Architecture for Fast Feature Embedding*, arXiv:1408.5093, 2014.

[4] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, *PyTorch.*

[5] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks.*

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, *Deep Residual Learning for Image Recognition*, arXiv:1512.03385, 2015.

[7] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, *Densely Connected Convolutional Networks*, arXiv:1608.06993, 2016.

[8] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba, *Learning Deep Features for Discriminative Localization*, arXiv:1512.04150, 2015.

[9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich, Scott Reed, *Going Deeper with Convolutions*, arXiv:1409.4842, 2014.

[10] École Polytechnique Fédérale de Lausanne *EE-559 - Deep Learning*, *Mini-project 1: Prediction of finger movements from EEG recordings.*

[11] Karen Simonyan, Andrew Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, arXiv:1409.1556, 2014.

[12] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, *You Only Look Once: Unified, Real-Time Object Detection.*

[13] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, arXiv:1311.2524, 2013.

[14] G. C. Benjamin Blankertz and K.-R. R. Muller, *Towards brain computer interfacing*, *Neural Information Processing Systems (NIPS)*, 2002.

[15] Davis E. King, *DLib*.

[16] Stanford University, CS231n - Convolutional Neural Networks for Visual Recognition.

[17] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. *CIFAR-10 dataset*.

[18] Sergey Ioffe, Christian Szegedy *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*.

[19] Conference on Computer Vision and Pattern Recognition 2017.

[20] International Conference on Computer Vision 2017, *2017.thecvf.com*.

[21] Conference on Computer Vision and Pattern Recognition 2018.

[22] Eric Tzeng, Judy Hoffman, Kate Saenko, Trevor Darrell, *Adversarial Discriminative Domain Adaptation*, arXiv:1702.05464, 2017.

[23] Artem Rozantsev, Mathieu Salzmann, Pascal Fua, *Beyond Sharing Weights for Deep Domain Adaptation*, arXiv:1603.06432, 2016.

[24] Mehdi Mirza, Simon Osindero, *Conditional Generative Adversarial Nets*, arXiv:1411.1784 2014.

[25] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas, *Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks*, arXiv:1612.03242, 2016.

[26] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas, *StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks*, arXiv:1710.10916, 2017.

[27] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, *Generative Adversarial Networks* arXiv:1406.2661, 2014.

[28] Han Zhang, Ian Goodfellow, Dimitris Metaxas and Augustus Odena, *Self-Attention Generative Adversarial Networks*, arXiv:1805.08318, 2018.

[29] Takeru Miyato, Toshiki Kataoka, Masanori Koyama and Yuichi Yoshida, *Spectral Normalization for Generative Adversarial Networks*, arXiv:1802.05957, 2018.

[30] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros, *Paired Image-to-image translation using Conditional GAN*, arXiv:1611.07004, 2016.

[31] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro, *High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs*, CVPR 2018.

[32] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*, ICCV 2017.

[33] Zhu, Jun-Yan and Zhang, Richard and Pathak, Deepak and Darrell, Trevor and Efros, Alexei A and Wang, Oliver and Shechtman, Eli, *Toward Multimodal Image-to-Image Translation*, arXiv:1711.11586, Nov 2017.

[34] Xiaolong Wang, Abhinav Shrivastava, Abhinav Gupta, *A-Fast-RCNN: Hard Positive Generation via Adversary for Object Detection*, CVPR, 2017.

[35] Ross Girshick, *Fast R-CNN*, arXiv:1504.08083, 2015.

[36] Jun-Yan Zhu, *pytorch-CycleGAN-and-pix2pix*.

[37] Cameron Smith, *neural-style-tf*.

[38] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, *Image Style Transfer Using Convolutional Neural Networks*.

[39] Manuel Ruder, Alexey Dosovitskiy, Thomas Broxn, *Artistic style transfer for videos*, arXiv:1604.08610, 2016.

[40] Leon A. Gatys, Matthias Bethge, Aaron Hertzmann, Eli Shechtman, *Preserving Color in Neural Artistic Style Transfer*, arXiv:1606.05897, 2016.

[41] The Starry Night by Vincent van Gogh, 1889.

[42] Multimedia Evaluation 2018, *mediaeval2018*.

[43] Medico: The 2018 Multimedia for Medicine Task.

[44] Joint Photographic Experts Group.

[45] The 2018 Emotional Impact of Movies Task, *emotionalimpact*.

[46] The 2018 Predicting Media Memorability Task, *memorability*.

[47] James L McGaugh, *Memory-a Century of Consolidation*, Science 287, 5451 (2000), 248-251.

[48] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, *Learning Spatiotemporal Features with 3D Convolutional Networks*, ICCV 2015.

[49] Khurram Soomro, Amir Roshan Zamir and Mubarak Shah, *UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild*, CRCV-TR-12-01, November 2012.

[50] ALMEIDA, Jurandy; LEITE, Neucimar J.; TORRES, Ricardo da S, *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011. p. 3673-3676.

[51] Dalal, N. and B. Triggs, *Histograms of Oriented Gradients for Human Detection*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1 (June 2005), pp. 886–893.

[52] DC. He and L. Wang (1990), *Texture Unit, Texture Spectrum, And Texture Analysis*, Geoscience and Remote Sensing, IEEE Transactions on, vol. 28, pp. 509 - 512.

[53] SZEGEDY, Christian et al, *Rethinking the inception architecture for computer vision*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. p. 2818-2826.

[54] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski, *Orb: an efficient alternative to sift or surf*, In Computer Vision (ICCV), 2011, IEEE International Conference on, 2011.

[55] OpenCV *Keypoints and Descriptors tutorial*.

[56] Eugenio Culurciello, TowardsDataScience, *Neural Network Architectures*.

[57] Sepp Hochreiter; Jürgen Schmidhuber, *Long Short-Term Memory*, Neural Computation.

[58] Google's AI organization, *TensorFlow*.

[59] Ronan Collobert, Koray Kavukcuoglu, Clement Farabet, *Torch*.

[60] Skymind, *From word to embeddings*.

[61] Aditya Khosla, Akhil S. Raju, Antonio Torralba and Aude Oliva, *Understanding and Predicting Image Memorability at a Large Scale*, International Conference on Computer Vision, 2015.

[62] Matt Harvey, Coastline Automation, *Five video classification methods implemented in Keras and TensorFlow*.

[63] Phillip Isola, Jianxiong Xiao, Antonio Torralba and Aude Oliva, *What makes an image memorable?*, IEEE Conference on Computer Vision and Pattern Recognition, 2011.

[64] T. Konkle, T. F. Brady, G. A. Alvarez, and A. Oliva, *Conceptual distinctiveness supports detailed visual long-term memory for realworld objects*, JEP:G, 2010.

[65] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, *LabelMe: a database and web-based tool for image annotation*, IJCV, 2008.

[66] Jiri Fajtl, Vasileios Argyriou, Dorothy Monekosso, Paolo Remagnino, *AMNet: Memorability Estimation with Attention*, arXiv:1804.03115, 2018.

[67] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio, *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*, arXiv:1502.03044, 2015.

[68] Diederik P. Kingma, Jimmy Ba, *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980, 2014.

[69] Google, *TensorFlow Hub Module*.

[70] Google, CodeLabs, *TensorFlow for Poets*.