Graphical Feelings: GIF Sentiment Learning

Presented by
Gordon Blake
gblake@stanford.edu

Overview

To improve GIF search, we implement a deep learning system to predict real-valued sentiment scores for GIFs using two network architectures: a MaxPool fully connected model and an RNN using LSTM units. The network takes a GIF as input and outputs 17 real values corresponding to different emotions. Both models achieve moderate loss but make conservative predictions close to the class means. Different GIF encoders may offer improvements.

Data and Features

Data consisted of 6,143 GIFs from the Giphy API and 2.7 pairwise emotion comparisons of GIF sentiment by internet users from MIT Media Lab's GIFGIF[1]. These were transformed into 17 normalized, real-valued scores per GIF using the Bradley-Terry model. Figure 1 shows the distribution of these scores by sentiment class.

The dataset was split into 80/10/10 train/dev/test portions. Frames from GIFs were sampled and encoded as 2048 dimensional vectors using the ResNeXt-101 CNN[2] pretrained on the Kinetics dataset of videos of human actions. Two sampling methods and data augmentation yielded three training data sets: (1) Sparse Small, (2) Dense Small, and (3) Sparse Augmented. As performance on all training sets was comparable, the Dense Small results are reported here.

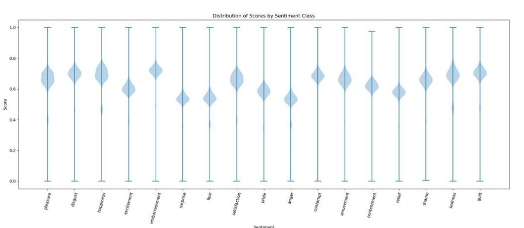

Figure 1. Distribution of sentiment scores

References
[1] http://gifgif.media.mit.edu
[2] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" In: *arXiv preprint* arXiv:1711.09577 (2017).
[3] Damian Borth et al. "Large-scale visual sentiment ontology and detectors using adjective-noun pairs". In: *Proceedings of the 21st ACM international conference on Multimedia*. ACM. 2013, pp. 223–232.
[4] Icons made by RoundIcons from www.flaticon.com

Models

Two models were explored, as shown in Figure 2:

(1) **Fully Connected MaxPool**: pools the CNN encoder output across frames as passes the result through fully connected layers.

(2) **RNN** uses a multilayer LSTM architecture to process a sequence of frames, feeding the output of the last RNN layer to a fully connected prediction layer.

The number of intermediate layers was a tuned hyperparameter ranging from 2 to 8. Other hyperparameters include batch size, number of hidden units, learning rate, dropout, and weight decay. Hyperparameters were tuned via randomized search.

Mean-squared error (MSE) loss was used on all models (see Equation 1). Performance was measured by taking the square root of the average MSE over classes, as well as by calculating the percentage of variance in the data explained by the model.

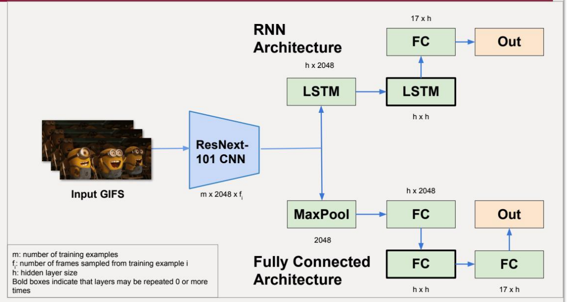

Figure 2. FC-MaxPool and RNN architectures

Equation 1. Mean-Squared Error loss function, where Y_i denotes sentiment prediction for a single class on example i

$$MSE(\hat{Y}, Y) = \frac{1}{b}\sum_{i=1}^{b}(\hat{Y}_i - Y_i)^2$$

Results

A two-layer RNN with 64 hidden units trained with dropout probability 0.1705 marginally outperformed all other models and the baselines.

Model	Train RMSE	Dev RMSE	Test RMSE	Test EV
Lin-Reg	0.07471	0.07848	0.08047	0.343
Mean Only	0.07336	0.07432	0.07611	0.406
FC-MaxPool-2	0.07336	0.07432	0.07627	0.406
FC-MaxPool-4	0.07362	0.07453	0.07627	0.406
RNN-2	**0.07120**	**0.07399**	**0.07610**	**0.408**
RNN-4	0.07341	0.07446	0.07613	0.406
RNN-8	0.07344	0.07431	0.07617	0.405

Table 1. Performance of best models

RMSE = Root-Mean-Squared Error averaged over classes
EV = Explained Variance
Train Samples: 4820, Dev Samples: 604, Test Samples: 604

Figure 3. Train/dev loss of best model over training epochs

Discussion and Future Directions

Even the best RNN model does not substantially outperform the baseline. All models trained appear to be too conservative, making predictions very close to the mean value for each sentiment class. As seen in Figure 3, dev and train loss were comparable, suggesting that the problem was one of bias, not variance. Qualitatively, GIFs with low prediction error tended to lack strong emotions while GIFs with high error had salient facial emotion markers such as smiles, tears, or wide eyes. These results suggest that the CNN encoder may not be capturing key emotional features.

Future work may use a different image encoder optimized for sentiment-specific features such as SentiBank[3] or use transfer learning to tune the last few layers of the encoder on GIF sentiment. The augmented dataset could be more densely sampled, combining the benefits of more frames per training example and more examples. Finally, an alternate loss function that more heavily weights correct predictions of strong emotions may also improve the model's ability to capture variance in the data.