Deep Learning for Lip Reading
Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman

Objectives
- Improve the architecture and training procedure of Stafylakis et al [3] for lip reading on word-level datasets (LRW [2]).
- CTC vs Seq2Seq: Compare these two methods for character-level lip reading.
- Assess the transferability of the features pretrained on the LRW dataset to the sentence-level GRID dataset.

CTC loss
\[
p(\pi|F) = \prod_{t=1}^{T} p_c^{CTC}(\pi(t))
\]
\[
p(Y|F) = \sum_{\pi \in \Pi(Y)} p(\pi|F)
\]
\[
\mathcal{L}_{CTC}(F; Y^*) = -\log(p(Y^*|F))
\]

Seq2Seq with Attention
\[
a_i^t, h_i^t = encRN(f_i^t, h_i^{t-1}) \\
d_i^t, h_i^{t+1} = decRN([i], h_i^t, h_i^t) \\
s_i^t = u^T \tanh(W h_i^t + V h_i^{t-1} + b) \\
c_i^t = \sum_{t=1}^{T} a_i^t e^{c_i^t} \\
h_i^t = \frac{\sum_{t=1}^{T} e^{c_i^t}}{\sum_{t=1}^{T} e^{c_i^t}}
\]
\[
\mathcal{L}_{seq}(X, Y^*) = \prod_{i=1}^{L} CE(softmax(a_i^t), y_i^t)
\]

CTC Beam Search Decoding
- Greedy decoding selects the token with the maximum posterior probability on every time step and ignores rest.
- Better decoding with a beam search incorporating a simple language model (LM).
- The search considers only paths that are prefixes of words in the dictionary.

Training procedure
- 3-stage training routine similar to Stafylakis et al [3].
- First pre-train word-level with a 1D convolutional backend, spatial & temporal data augmentation and CE loss.
- Then dump features on disk and train recurrent backends using CE loss for word-level and CTC or Seq2Seq for character-level.
- Finally fine-tune end-to-end.
- For word-level training on LRW dataset, we add dropout to the recurrent layers and increase the cell size compared to [3].

Data Augmentation
- 3-stage training routine similar to Stafylakis et al [3].
- First pre-train word-level with a 1D convolutional backend, spatial & temporal data augmentation and CE loss.
- Then dump features on disk and train recurrent backends using CE loss for word-level and CTC or Seq2Seq for character-level.
- Finally fine-tune end-to-end.
- For word-level training on LRW dataset, we add dropout to the recurrent layers and increase the cell size compared to [3].

Results
- LRW dataset: we improve the state-of-the-art by almost 2%.
- GRID Dataset: We train recurrent backends with CTC and Seq2Seq on features obtained with a frontend trained on LRW. Our best results are similar to Assael et al [1].

References: