Sentence classification on TREC dataset

Slide by: Hoang-Trieu Trinh

Supervisor: Le-Minh Nguyen

School of Information Science

Japan Advanced Institute of Science and Technology

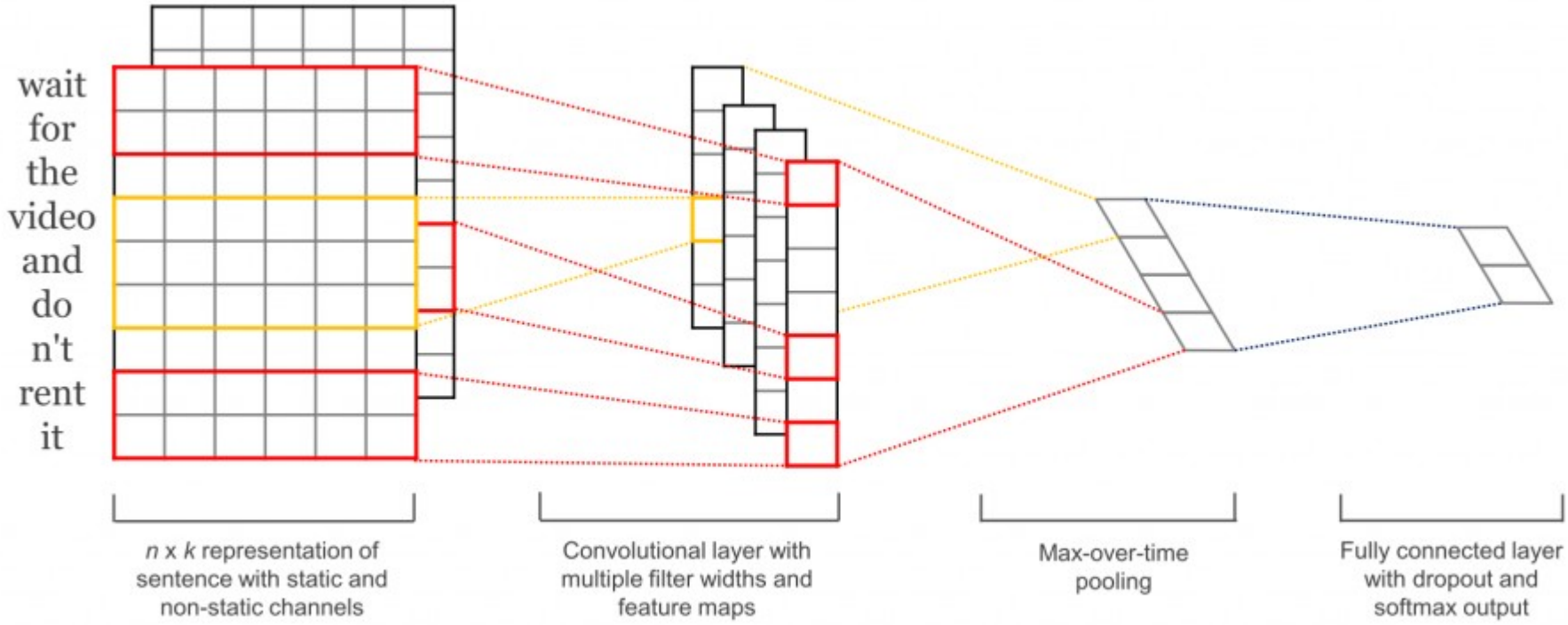
February 2016

DATA

- TREC dataset:
  - **DESC:manner** How did serfdom develop in and then leave Russia ?
  - **ENTY:cremat** What films featured the character Popeye Doyle ?
  - **DESC:manner** How can I find a list of celebrities ' real names ?
  - **ENTY:animal** What fowl grabs the spotlight after the Chinese Year of the Monkey ?
  - **ABBR:exp** What is the full form of .com ?
  - **HUM:ind** What contemptible scoundrel stole the cork from my lunch ?

- 50 labels, 6 main labels
- Average length: 10
- Training set size: 5452, Test set size: 500
- Vocabulary size: 8700
- Pre-trained word2vec words in vocab: 7500

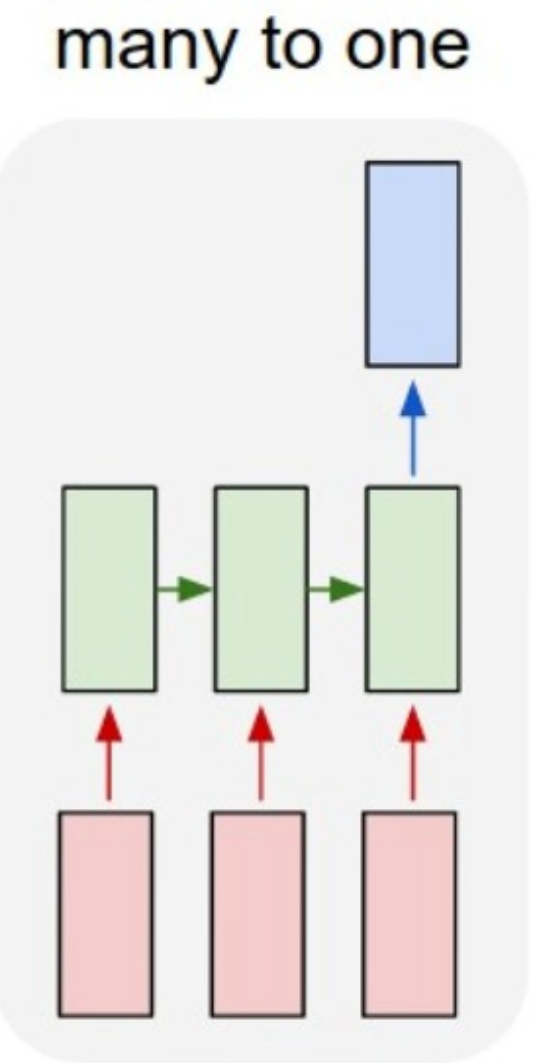
CNN



- Layer 1: 3 types of window size 3x300, 4x300, 5x300. 100 feature maps each.
- Layer 2: max-pool-over-time
- Layer 2: Fully connected with softmax output. Weight norm constrained by 3. Dropout with p = 0.5

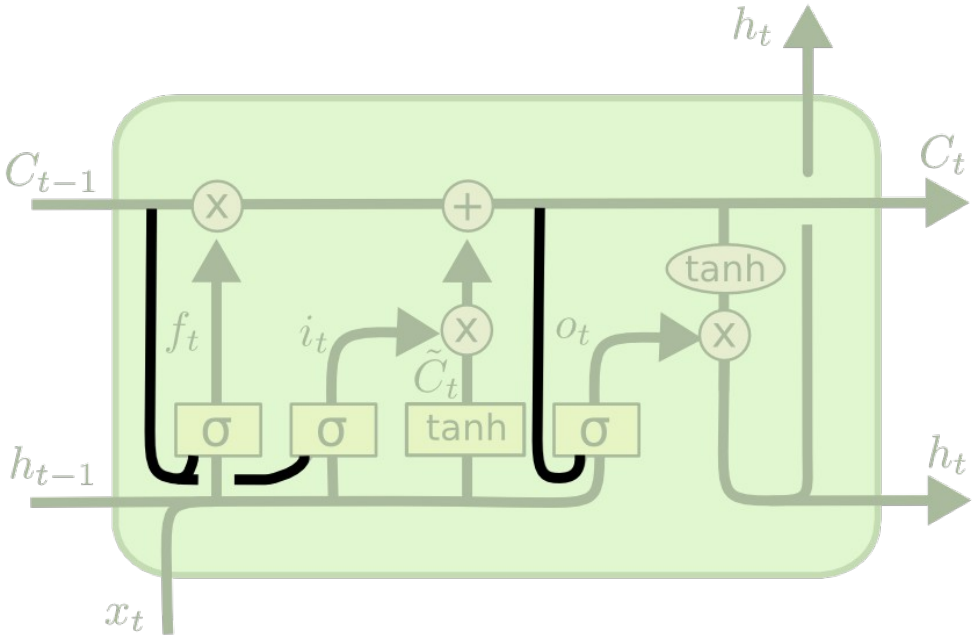
LSTM

- Layers:
  - Layer 1 (red): Embedding (300x1)
  - Layer 2 (green): LSTM cell
  - Layer 3 (blue): output of the network (300x1)
  - Layer 4: fully connected, softmax output to 6 classes. Dropout p = 0.5



http://karpathy.github.io/2015/05/21/rnn-effectiveness/

LSTM with peepholes

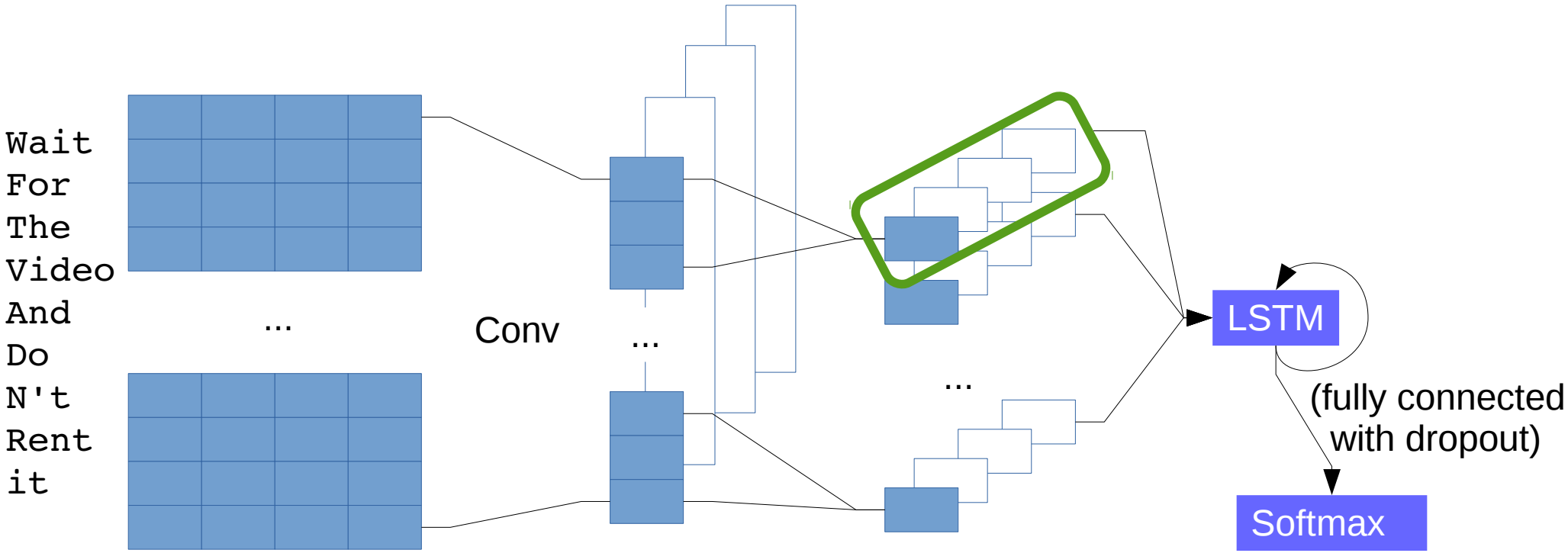


$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \; + \; b_o\right)$$

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

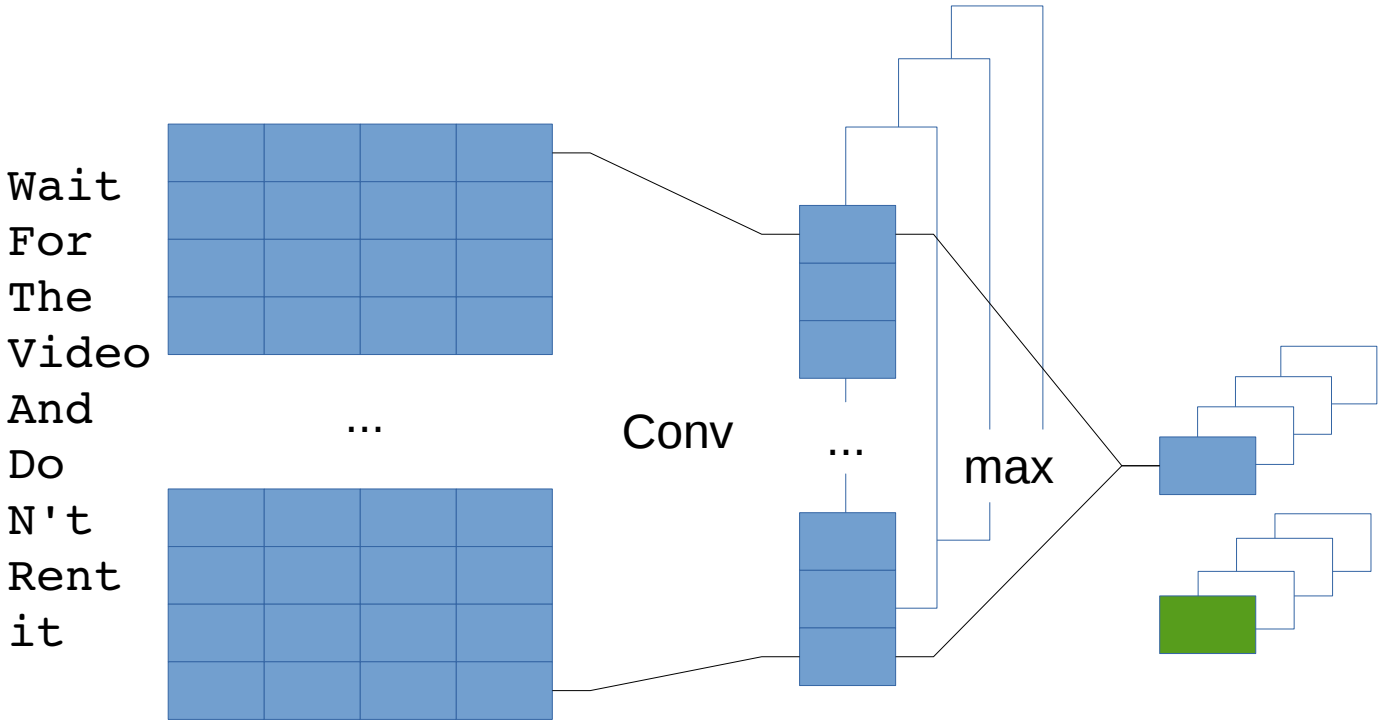
- Perform better than GRU/ regular LSTM on TREC

CNN-LSTM



Suggestion: perform a softer version of max-pool-over-time. Sequential information is better retained.

With position



$$maxpool = max(conv)$$
$$position = argmax(conv)/L$$

$$argmax(x) \approx [1\,2\,3 \ldots L]^T softmax_{\alpha \to \infty}(\alpha x)$$

Changes

- Do not use padding:
  - Too much noise: ave. seq. Length: **10**, After padding: **37**
  - Problem: varying sentence length
- Early stop:
  - Only 5% is used instead of 10%
  - Do not actually use “early stop”, instead run to a maximum epoch and reverse to the time step where the model best perform on validation set.

Vietnamese TREC dataset

- Vietnamese translation of TREC dataset
- 4600 / 8700 words covered by pre-train Vietnamese Word2Vec (most of the uncovered words are in English)
- Re-tune all hyper-parameters

Result

	TREC	TRECvn
CNN	93	91.8
CNN-LSTM	94.2	92.8
LSTM	95.4	94.2

SVM as final layer

- **CNN**
- **LSTM**
- **CNN-LSTM**
- **Any other classifier**

→ Final layer is always a fully connected layer with softmax output. Equivalent to a Linear Classifer.

→ If the previous layers is able to extract useful features that represent points separable well by a simple Linear Classifier, these features should also be useful to other Linear Classifiers.

→ Replacing this layer by another Linear Classifier (e.g. Linear SVM) is done by replacing the loss function.

SVM as final layer

- Instead of

$$Loss = \texttt{CrossEntropy}(\boldsymbol{target}, \mathbf{softmax}(W\,\phi(input)))$$

- Where phi(input) is the extracted features obtained from previous layers (conv/ conv-lstm / lstm)
- SVM (One vs Rest), t is -1 or 1

$$Loss = \sum_{i}^{n}\sum_{j}^{6} \mathrm{RELU}(-t_i W_j \phi_i + 1)$$

- With soft-margin

$$Loss = \sum_{i}^{n}\sum_{j}^{6} \mathrm{RELU}(-t_i W_j \phi_i + 1 - \xi_{ij}) + RELU(-\xi_{ij})$$

Result on TREC

CNN	CNN – SVM	CNN – SVM soft
93	93.6	93.6

Encourage Separability

- When data is linearly separable, SVM is expected to generalise better than a Linear Classifier using softmax output

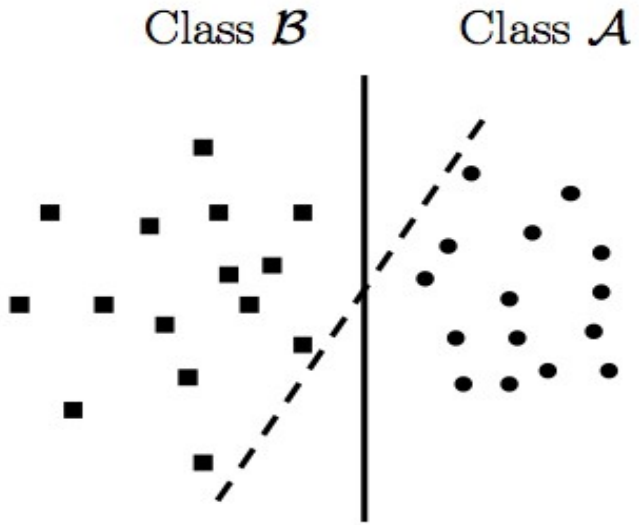


Figure 1. Which plane is best?

http://stats.stackexchange.com/questions/23391/how-does-a-support-vector-machine-svm-work

Encourage Separability

- So, it should benefitial to encourage separability when using SVM as the final layer. This can be done by adding another term to Loss function

$$Loss \mathrel{+}= \alpha \frac{\sum within\ class\ variance}{across\ class\ variance}$$

TO DO

- Implement SVM One vs. One
- Experiment SVM / SVM soft / SVM + varianceLoss on all types of models

Thank you

- Q & A.