

Recurrent Neural Network for Language Modeling

EE596/LING580 -- Conversational Artificial Intelligence

Hao Cheng

University of Washington

Language Model Basics

- $P(\mathbf{w}) = \prod_{i=1}^T P(w_j | w_1 \dots w_{j-1})$
- N-gram model

$$P(\mathbf{w}) = \prod_{i=1}^T P(w_j | w_{j-N+1} \dots w_{j-1})$$

- Log-likelihood:

$$\sum_{i=1,..,N} \log_b P(\mathbf{w}^{(i)})$$

- Perplexity (PPL):

$$b^{-\frac{1}{N} \sum_i \log_b P(\mathbf{w}^{(i)})}$$

Limitations of N-gram Model

- With increasing order (N) of the N-gram model, the number of possible parameters increases **exponentially**
 - Vocabulary size: V
 - Unigram: $V - 1$ parameters
 - Bigram: $V^2 - 1$ parameters
 - N-gram: $V^{N-1} - 1$ parameters
- Require tremendous amount of data to give good estimate on parameters of high-order N-gram models

Neural Network Language Models (NNLM)

- Many word histories are similar (but not exact)
- Project sparse history onto some continuous low-dimensional space
 - i.e., similar histories can be clustered
- Fewer parameters have to be estimated from the training data

Feedforward NNLM

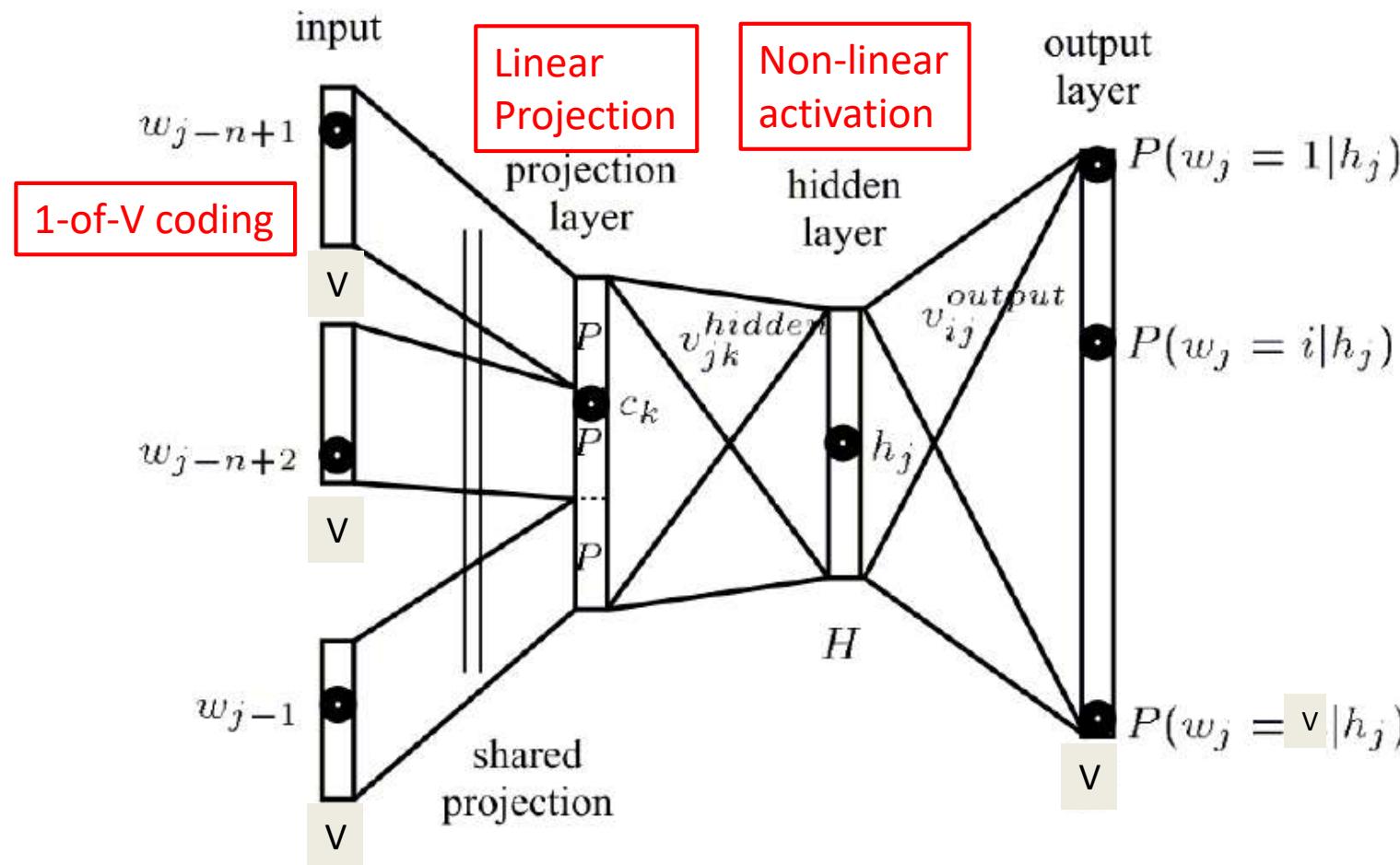
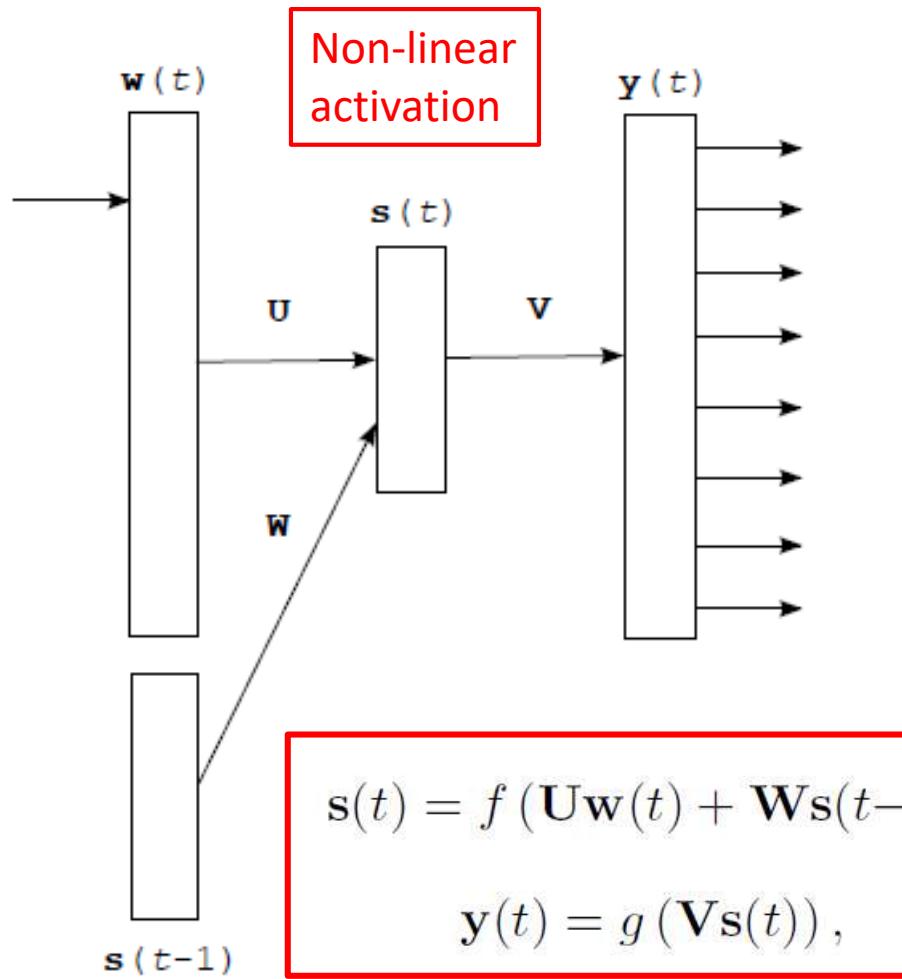


Fig. from Y. Bengio and H. Schwenk.

More effective representation of history?

Recurrent NNLM (RNNLM)



- Input layer and output layer have the same dimensionality as the vocabulary
- Hidden layer is smaller (50 – 1000 neurons)
- U, W are the matrices of weights between input and hidden layer
 - U : words
 - W : history states
- V is the matrix of weights between hidden and output layer

$$f(z) = \frac{1}{1 + e^{-z}},$$
$$g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$

Fig. from T. Mikolov.

Backpropagation Through Time (BPTT)

- The recurrent weights W are updated by unfolding them in time and training the network as a deep feedforward NN.
- The process of propagating errors back through the recurrent weights is called BPTT.

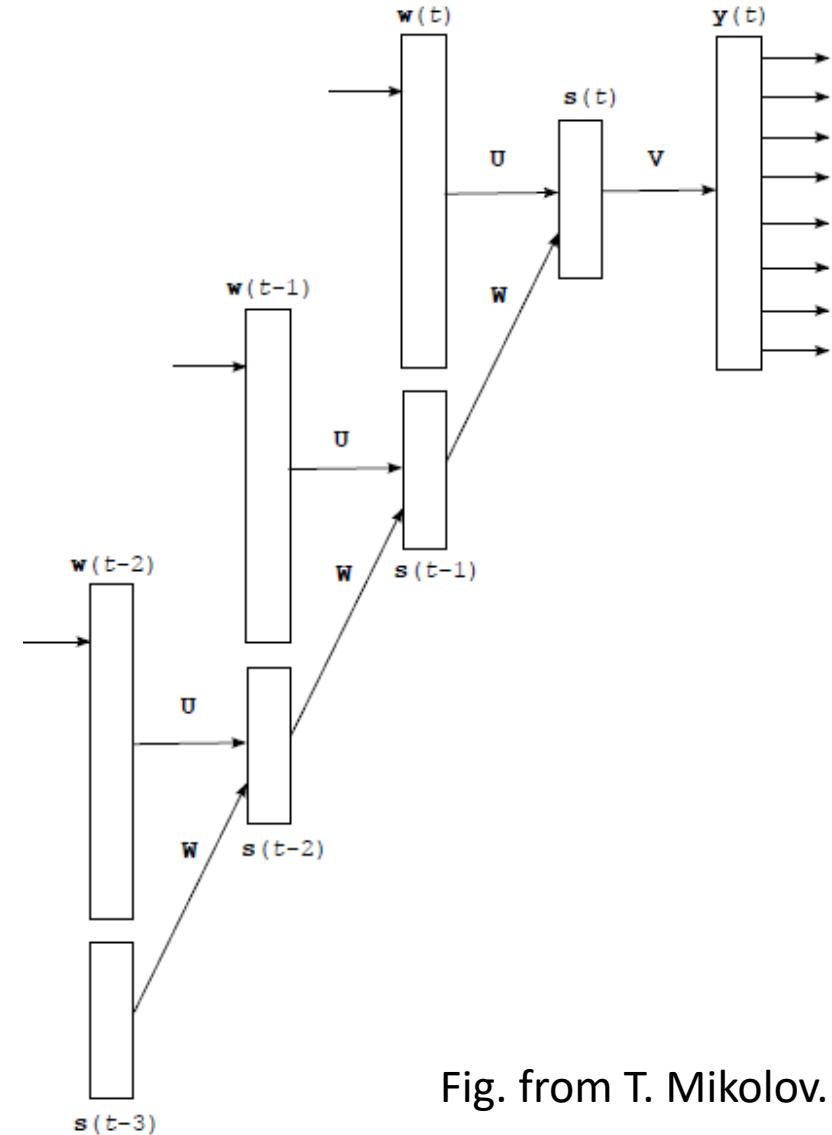


Fig. from T. Mikolov.