IBM Watson

Semantic Word Embedding Neural Network Language Models for Automatic Speech Recognition

Kartik Audhkhasi, Abhinav Sethy

Bhuvana Ramabhadran

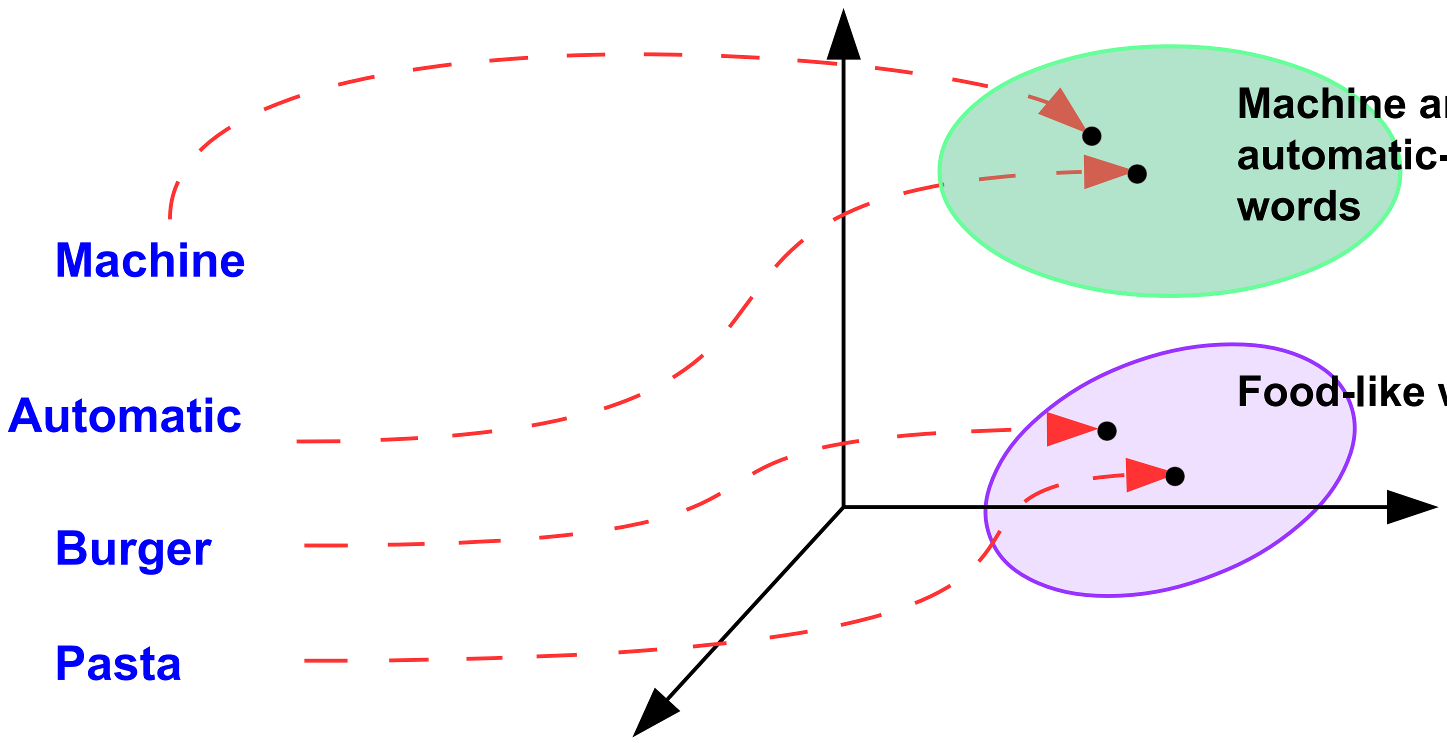
Watson Multimodal Group

IBM T. J. Watson Research Center

IBM

Motivation

- Semantic word embedding algorithms (e.g. word2vec and GloVe) aim to capture semantic information from text.



Motivation

- Semantic word embedding algorithms (e.g. word2vec and GloVe) aim to capture semantic information from text.
- Semantic embeddings are diverse compared to word embeddings learned by a neural network language model (NNLM).

Sim. Rank	Speech		Machine		Learning	
	GloVe	FNNLM	**GloVe**	FNNLM	**GloVe**	FNNLM
1	**Remarks**	Address	**Machines**	Stun	**Learn**	Learn
2	**Address**	Event	**Guns**	Pellet	**Teaching**	Learned
3	**Speeches**	Ceremony	**Gun**	Celebratory	**Learned**	Learns
4	**Comments**	Statement	**Hand**	Millimeter	**Skills**	Complain
5	**Bush**	Remarks	**Automatic**	Sharpnel	**Teach**	Confirmation

Top-5 nearest words found using cosine similarity on GloVe and feedforward NNLM (FNNLM) embeddings.

What is the GloVe Algorithm for Computing Semantic Word Embeddings?

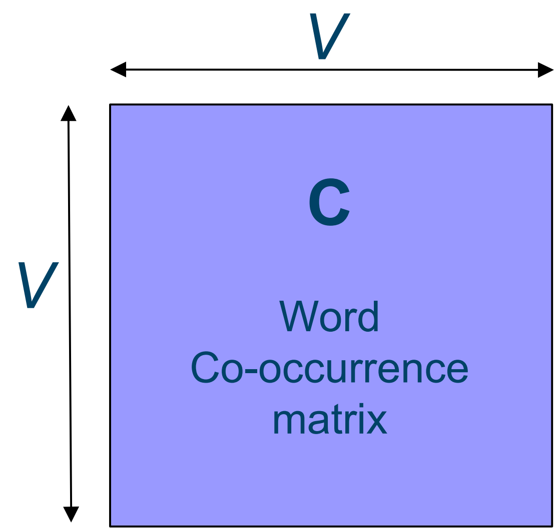
GloVe – Global Vectors for Word Representation

- GloVe performs a bilinear approximation of the word co-occurrence matrix computed over training data.
- The *V x V* dimensional word co-occurrence matrix **C** is obtained by traversing training text and counting co-occurrences.

Central word: evening

*“Good **evening** and welcome to the evening news”.*

Context: (good, **evening**), (and, **evening**)



J. Pennington, R. Socher, C. D. Manning, *“GloVe: Global Vectors for Word Representation”*, Proc. EMNLP, Vol. 14. 2014.

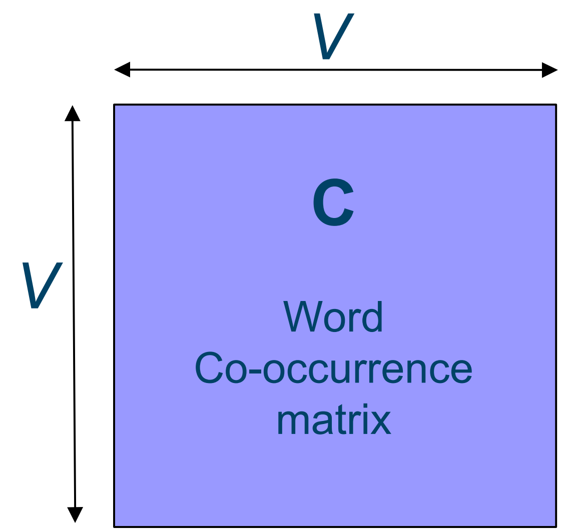
GloVe – Global Vectors for Word Representation

- GloVe performs a bilinear approximation of the word co-occurrence matrix computed over training data.
- The *V x V* dimensional word co-occurrence matrix **C** is obtained by traversing training text and counting co-occurences.

Central word: evening

“Good evening and welcome to the ***evening*** *news”.*

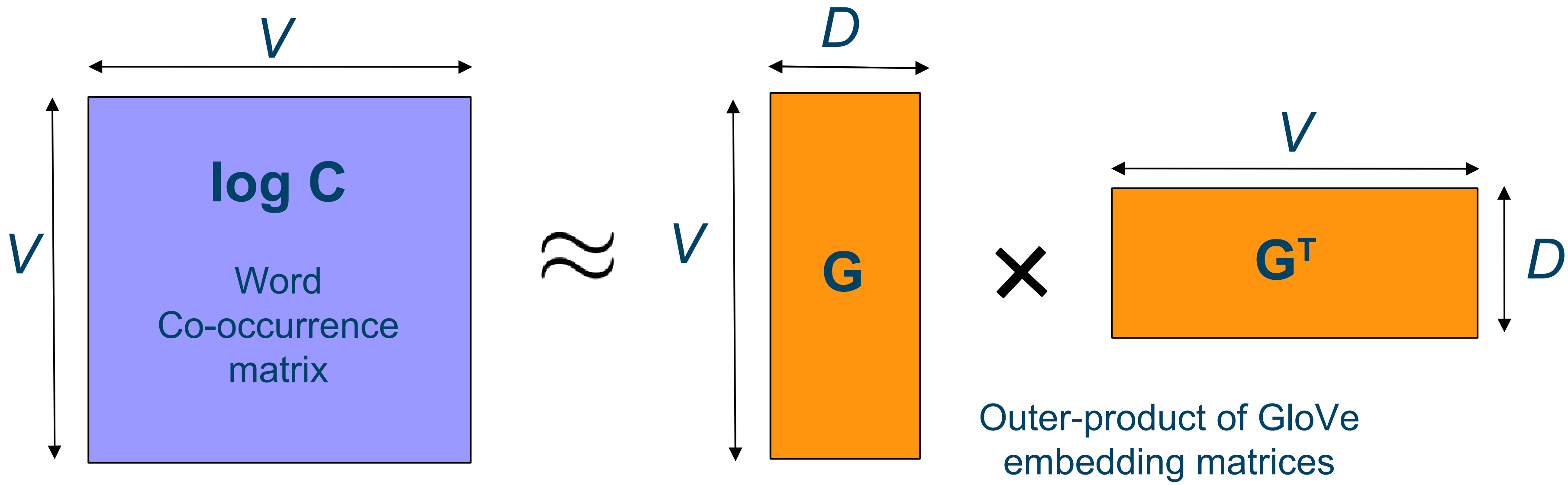
Context: (the, **evening**), (news, **evening**)



J. Pennington, R. Socher, C. D. Manning, *“GloVe: Global Vectors for Word Representation”*, Proc. EMNLP, Vol. 14. 2014.

GloVe – Global Vectors for Word Representation

- GloVe performs a bilinear approximation of the word co-occurrence matrix computed over training data.
- The *V x D* dimensional GloVe matrix **G** is obtained as follows:



J. Pennington, R. Socher, C. D. Manning, *“GloVe: Global Vectors for Word Representation”*, Proc. EMNLP, Vol. 14. 2014.

GloVe – Global Vectors for Word Representation

- GloVe performs a bilinear approximation of the word co-occurrence matrix computed over training data.
- The *V x D* dimensional GloVe matrix **G** is obtained as follows:

$$\mathbf{G}^*, \mathbf{b}^* \quad = \arg\min_{\mathbf{G},\mathbf{b}} \sum_{i,j=1}^{V} f(\mathbf{C}(i,j)) \Big(\mathbf{G}(i)\mathbf{G}(j)^T + b(i) + b(j) - \log \mathbf{C}(i,j) \Big)^2$$

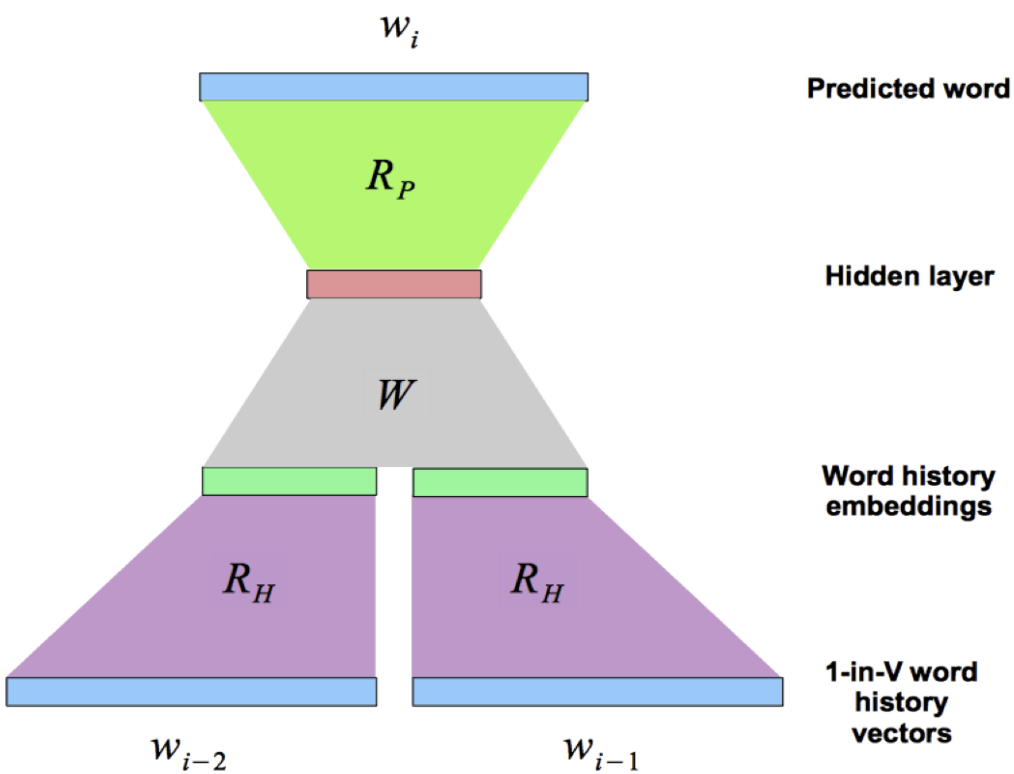
where ***b*** is a vector of biases and *f(x)* is the weighting function:

$$f(x) = \min\{1, (x/x_{\min})^{\alpha}\}$$

How do we include GloVe embeddings in a feed-forward NNLM?

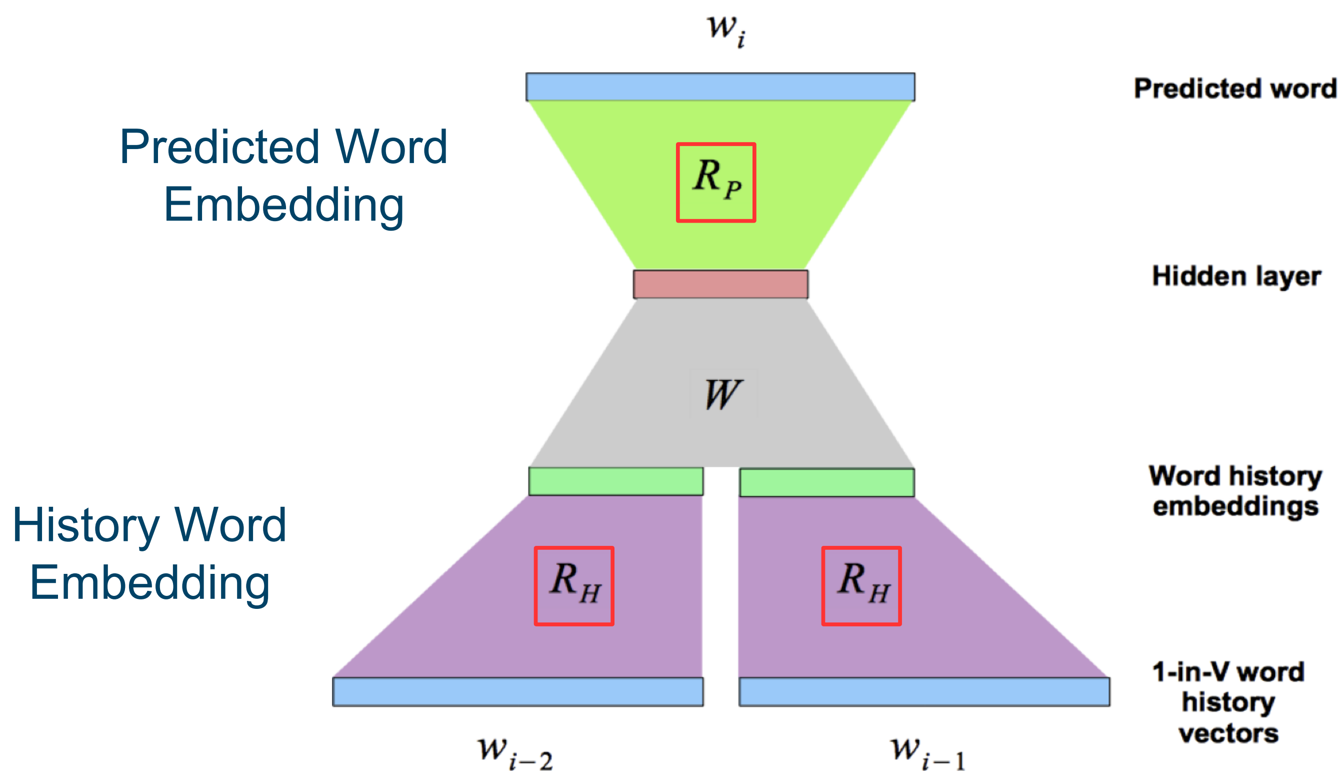
A Standard Feed-forward NNLM (FNNLM)

- A feed-forward NNLM predicts the next word by passing continuous embeddings of the history words through a feed-forward NN:



A Standard Feed-forward NNLM (FNNLM)

- A FNNLM uses two different word embeddings learned to minimize training text perplexity.

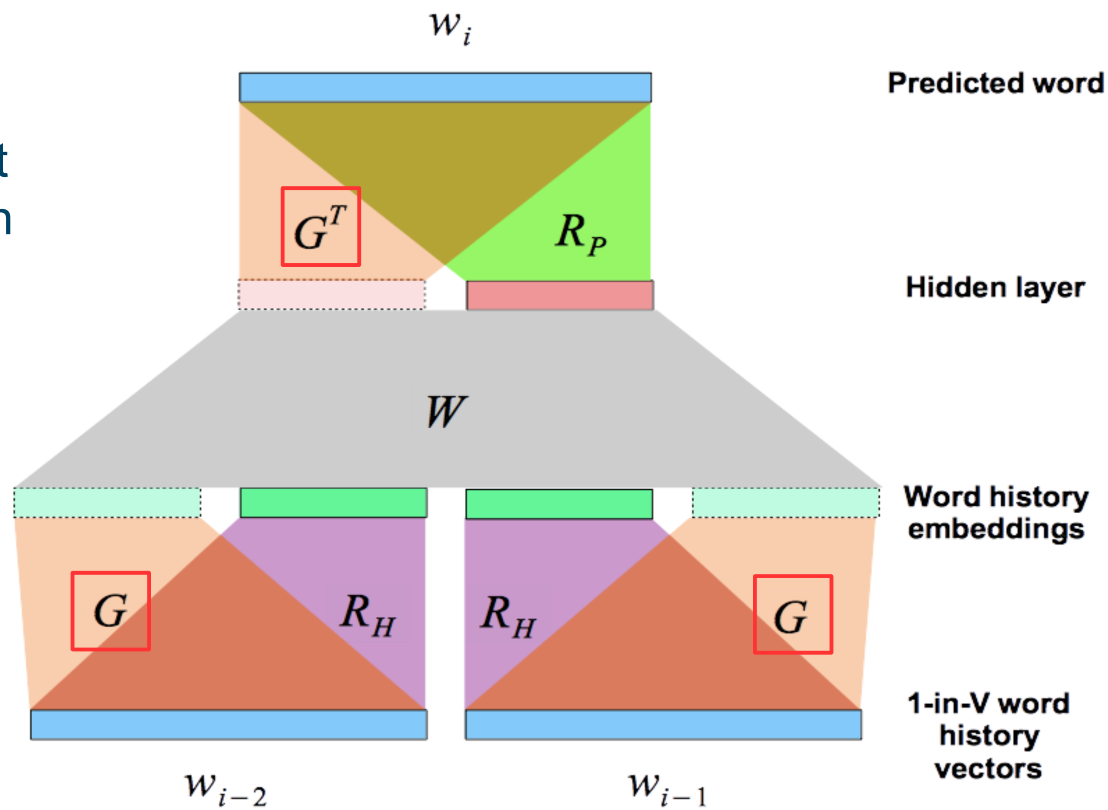


Semantic Word Embedding (SWE) FNNLM

- A SWE-FNNLM incorporates the GloVe matrix ***G*** as follows:

Hidden → Output Weight expansion

Input feature fusion

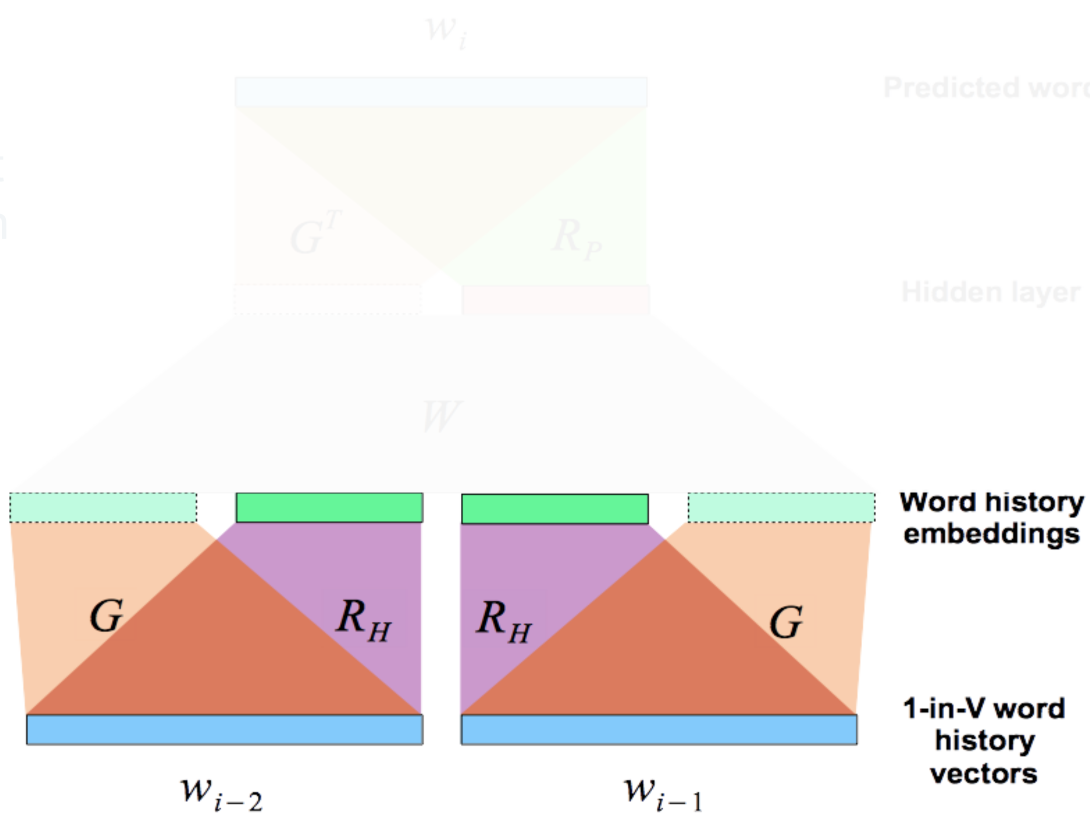


Intuition Behind SWE-NNLM

- Input feature concatenation fuses two diverse word embeddings.

Hidden → Output
Weight expansion

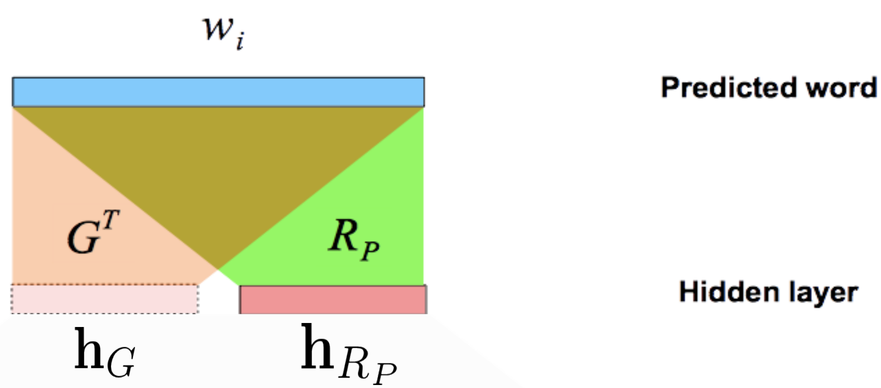
Input feature
fusion



Intuition Behind SWE-NNLM

- Output weight expansion performs log-linear interpolation of two un-normalized FNNLMs.

Hidden → Output
Weight expansion



$$P(w_i = k) \propto \exp(G^T \mathbf{h}_G + R_P \mathbf{h}_{R_P})$$
$$= \exp(G^T \mathbf{h}_G)\exp(R_P \mathbf{h}_{R_P})$$

Semantic word Embedding-based FNNLM

Standard FNNLM

LM Experimental Setup

- We trained all NNLMs on a 12M word subset of the 2007 IBM GALE English Broadcast news ASR system.
- Vocab size limited to 20K words.
- 300-dimensional GloVe word embeddings trained on the 2B word English Gigaword corpus.
- LM training used a mini-batch based stochastic gradient descent.
- We do not update the GloVe embeddings during LM training since it gave insignificant perplexity reduction.

LM Results

- LM perplexities on dev04 set:

LM	Perplexity	% Reduction
6gm KN	144.5	-
5gm FNNLM (300,500) *300-dim embeddings, 500 hidden neurons*	144.9	-0.3%
6gm KN + FNNLM	118.3	18.1%

LM Results

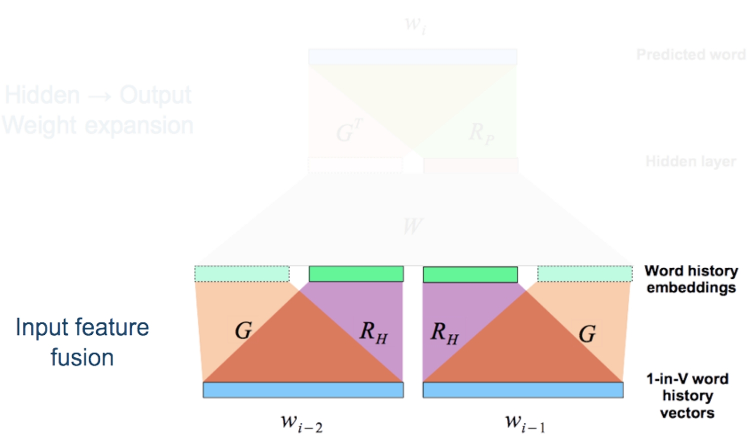
- LM perplexities on dev04 set:

LM	Perplexity	% Reduction
6gm KN	144.5	-
5gm FNNLM (300,500) *300-dim embeddings, 500 hidden neurons*	144.9	-0.3%
5gm SWE-FNNLM (300,500)+300	**128.5**	**11.1%**
6gm KN + FNNLM	118.3	18.1%
6gm KN + SWE-FNNLM	**111.8**	**22.6%**
All	109.6	24.2%

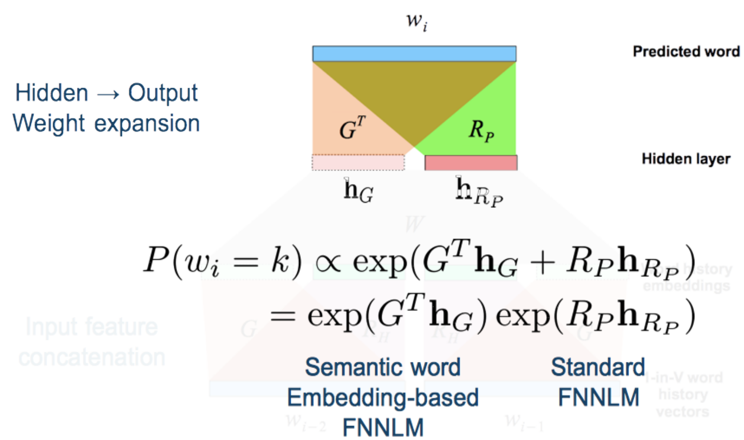
- SWE-FNNLM gives significant perplexity improvement over a standard FNNLM.
- FNNLM of similar size (600,800) gives worse perplexity of 151.

LM Results

- Is including GloVe embedding in the output layer important?



VS



$$P(w_i = k) \propto \exp(G^T \mathbf{h}_G + R_P \mathbf{h}_{R_P})$$
$$= \exp(G^T \mathbf{h}_G) \exp(R_P \mathbf{h}_{R_P})$$

LM Results

- Is including GloVe embedding in the output layer important?

LM	Perplexity	% Reduction
6gm KN	144.5	-
5gm Input-only SWE-FNNLM	134.2	7.1%

LM Results

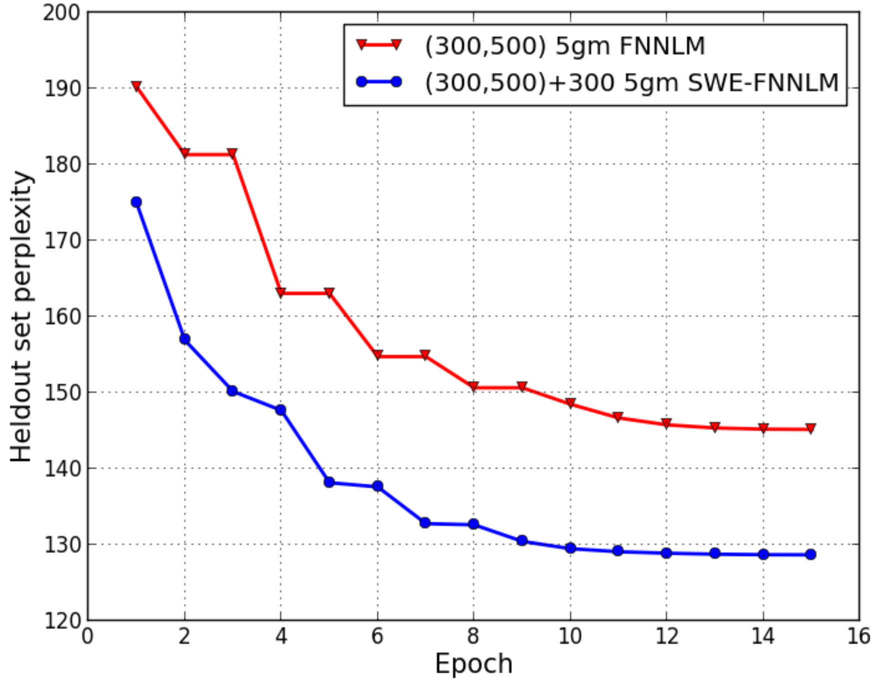
- Is including GloVe embedding in the output layer important?

LM	Perplexity	% Reduction
6gm KN	144.5	-
5gm Input-only SWE-FNNLM	134.2	7.1%
5gm full SWE-FNNLM	**128.5**	**11.1%**

- Both input feature fusion and input → output weight expansion contribute significantly to perplexity improvement.

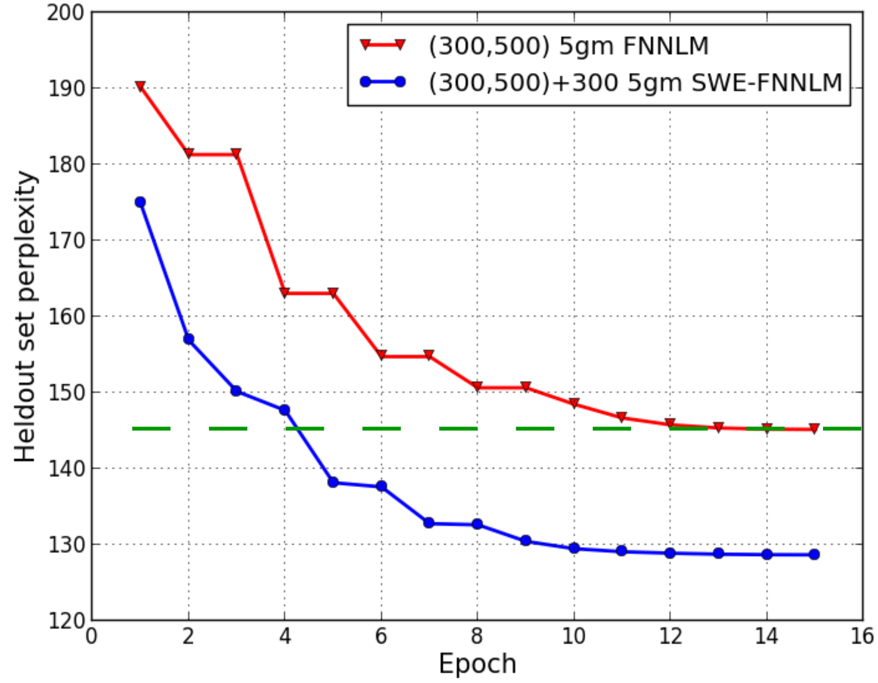
LM Results

- SWE-FNNLM gets a head-start during training due to diverse embeddings trained on large corpus.



LM Results

- SWE-FNNLM enables rapid adaptation of LMs on new in-domain data.



ASR Lattice Rescoring Setup

- Acoustic model is CNN-HMM hybrid system trained on 400 hrs of broadcast news data.
- Decoder vocabulary is 80K words.
- Baseline LM is a linear interpolation of 4gm KN LMs trained on different data sources from a 350M word corpus.
- We generated lattices on the rt04 test set using a pruned baseline LM.

ASR Lattice Rescoring Results

- WERs on the rt04 test set after lattice rescoring:

LM	WER	% Reduction
4gm KN	11.3%	-
4gm KN + FNNLM	11.0%	2.6%

ASR Lattice Rescoring Results

- WERs on the rt04 test set after lattice rescoring:

LM	WER	% Reduction
4gm KN	11.3%	-
4gm KN + FNNLM	11.0%	2.6%
4gm KN + SWE-FNNLM	**10.7%**	**5.3%**

The above WER reductions are significant ($p < 0.001$) using NIST SCTK's sc_stats.

ASR Lattice Rescoring Results

- WERs on the rt04 test set after lattice rescoring:

LM	WER	% Reduction
4gm KN	11.3%	-
4gm KN + FNNLM	11.0%	2.6%
4gm KN + SWE-FNNLM	**10.7%**	**5.3%**

The above WER reductions are significant ($p < 0.001$) using NIST SCTK's sc_stats.

- **Comparison with 350M word LMs:** Model M (10.6%) and FNNLM (10.3%).

A. Sethy, S. Chen, E. Arisoy, B. Ramabhadran , *"Unnormalized exponential and neural network language models"*, Proc. ICASSP, 2015.

Conclusion

- Semantic word embeddings trained on a large corpus improve neural network language models.
- The performance benefit appears due to diversity of semantic embeddings to the embeddings learned by a NNLM **and** large corpus used to train the semantic embeddings.
- Including semantic word embeddings through both feature fusion and input → output weight expansion helps LM performance.
- We are currently exploring application of semantic word embeddings to recurrent NNLMs.