(R)NN-based Language Models

Lecture 12

CS 753

Instructor: Preethi Jyothi

Word representations in Ngram models

- In standard Ngram models, words are represented in the discrete space involving the vocabulary
- Limits the possibility of truly interpolating probabilities of unseen Ngrams
- Can we build a representation for words in the continuous space?

Word representations

- 1-hot representation:
  - Each word is given an index in {1, … , V}. The 1-hot vector $f_i \in R^V$ contains zeros everywhere except for the i^{th} dimension being 1
- 1-hot form, however, doesn't encode information about word similarity
- Distributed (or continuous) representation: Each word is associated with a dense vector. Based on the "distributional hypothesis".
  E.g. dog → {-0.02, -0.37, 0.26, 0.25, -0.11, 0.34}

Word embeddings

- These distributed representations in a continuous space are also referred to as “word embeddings”
  - Low dimensional
  - Similar words will have similar vectors
- Word embeddings capture semantic properties (such as man is to woman as boy is to girl, etc.) and morphological properties (glad is similar to gladly, etc.)

Word embeddings

FRANCE	JESUS	XBOX	REDDISH	SCRATCHED	MEGABITS
AUSTRIA	GOD	AMIGA	GREENISH	NAILED	OCTETS
BELGIUM	SATI	PLAYSTATION	BLUISH	SMASHED	MB/S
GERMANY	CHRIST	MSX	PINKISH	PUNCHED	BIT/S
ITALY	SATAN	IPOD	PURPLISH	POPPED	BAUD
GREECE	KALI	SEGA	BROWNISH	CRIMPED	CARATS
SWEDEN	INDRA	PSNUMBER	GREYISH	SCRAPED	KBIT/S
NORWAY	VISHNU	HD	GRAYISH	SCREWED	MEGAHERTZ
EUROPE	ANANDA	DREAMCAST	WHITISH	SECTIONED	MEGAPIXELS
HUNGARY	PARVATI	GEFORCE	SILVERY	SLASHED	GBIT/S
SWITZERLAND	GRACE	CAPCOM	YELLOWISH	RIPPED	AMPERES

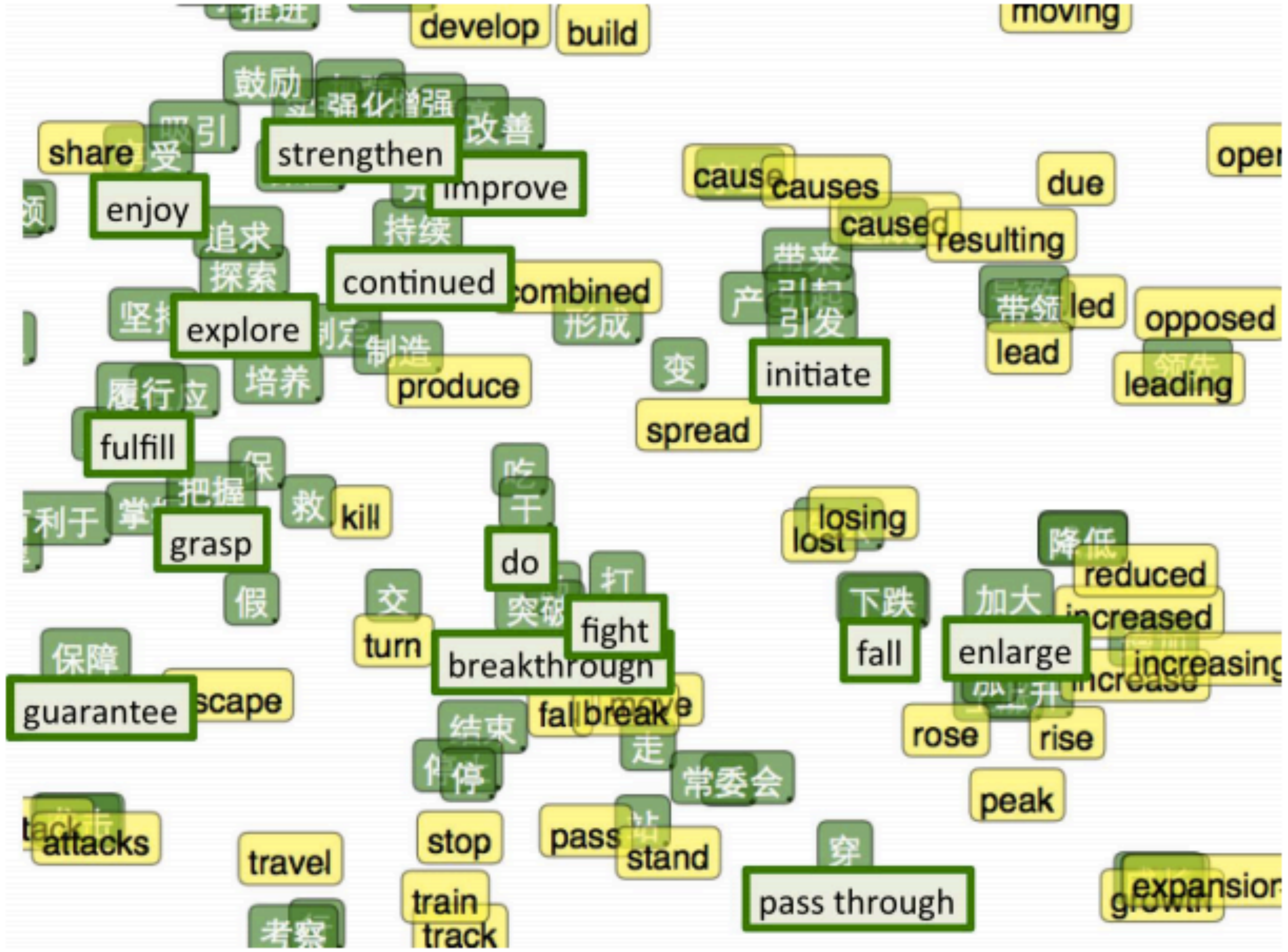
[C01]: Collobert et al.,01

Relationships learned from embeddings

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza

[M13]: Mikolov et al.,13

Bilingual embeddings

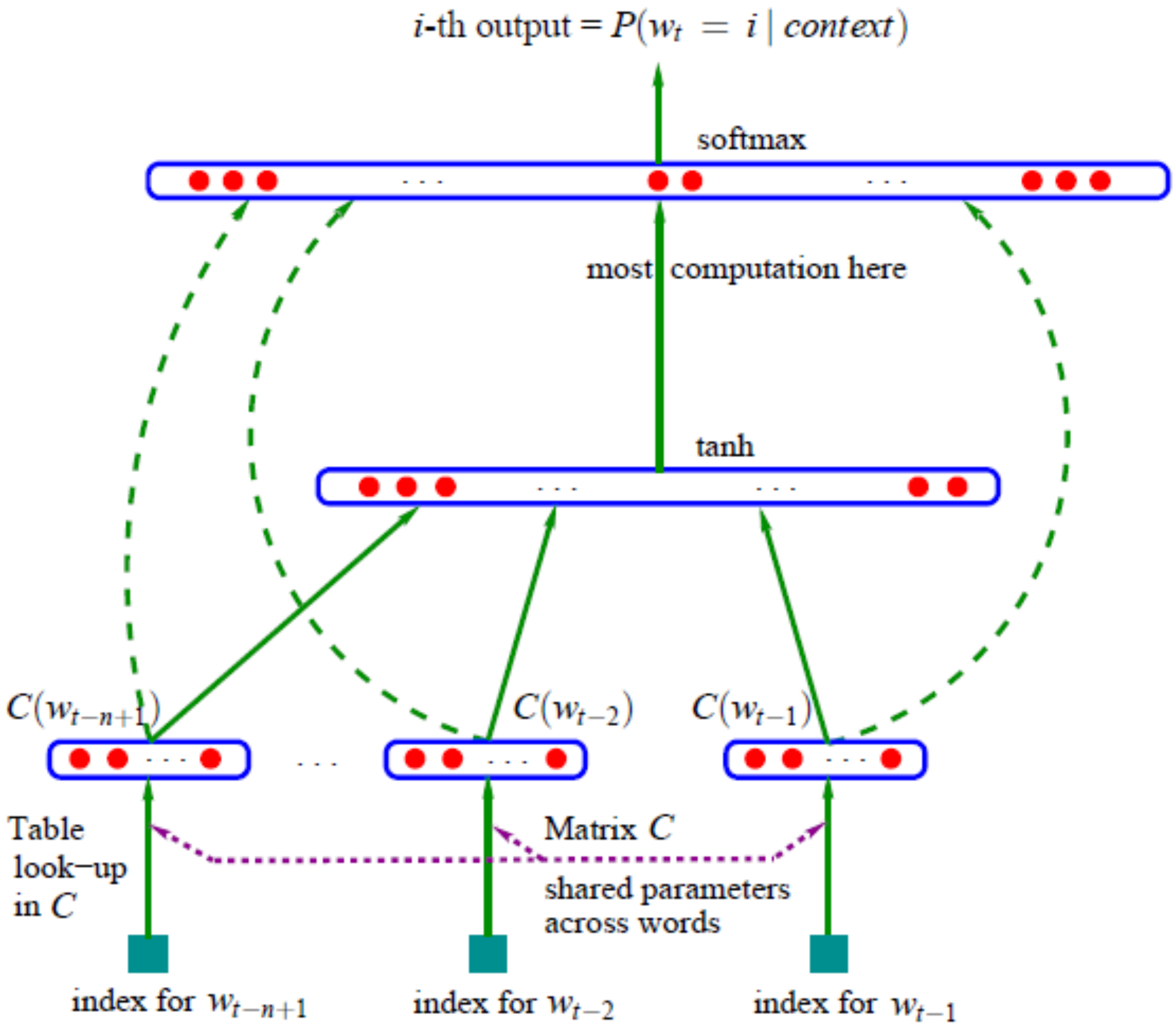


Word embeddings

- These distributed representations in a continuous space are also referred to as “word embeddings”
  - Low dimensional
  - Similar words will have similar vectors
- Word embeddings capture semantic properties (such as man is to woman as boy is to girl, etc.) and morphological properties (glad is similar to gladly, etc.)
- The word embeddings could be learned via the first layer of a neural network [B03].

[B03]: Bengio et al., “A neural probabilistic LM”, JMLR, 03

Word embeddings

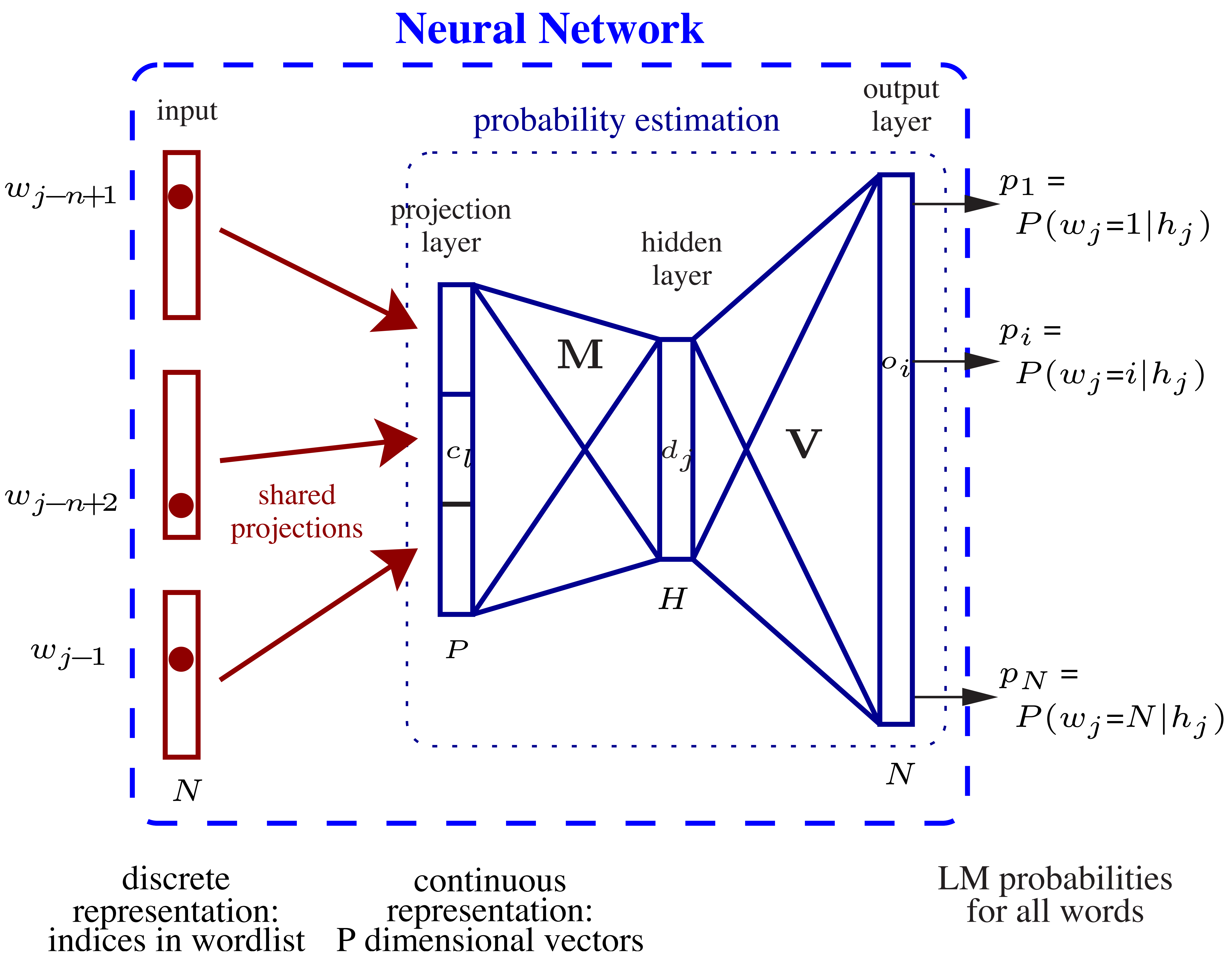


Introduced the architecture that forms the basis of all current neural language and word embedding models

- Embedding layer
- One or more middle/hidden layers
- Softmax output layer

[B03]: Bengio et al., “A neural probabilistic LM”, JMLR, 03

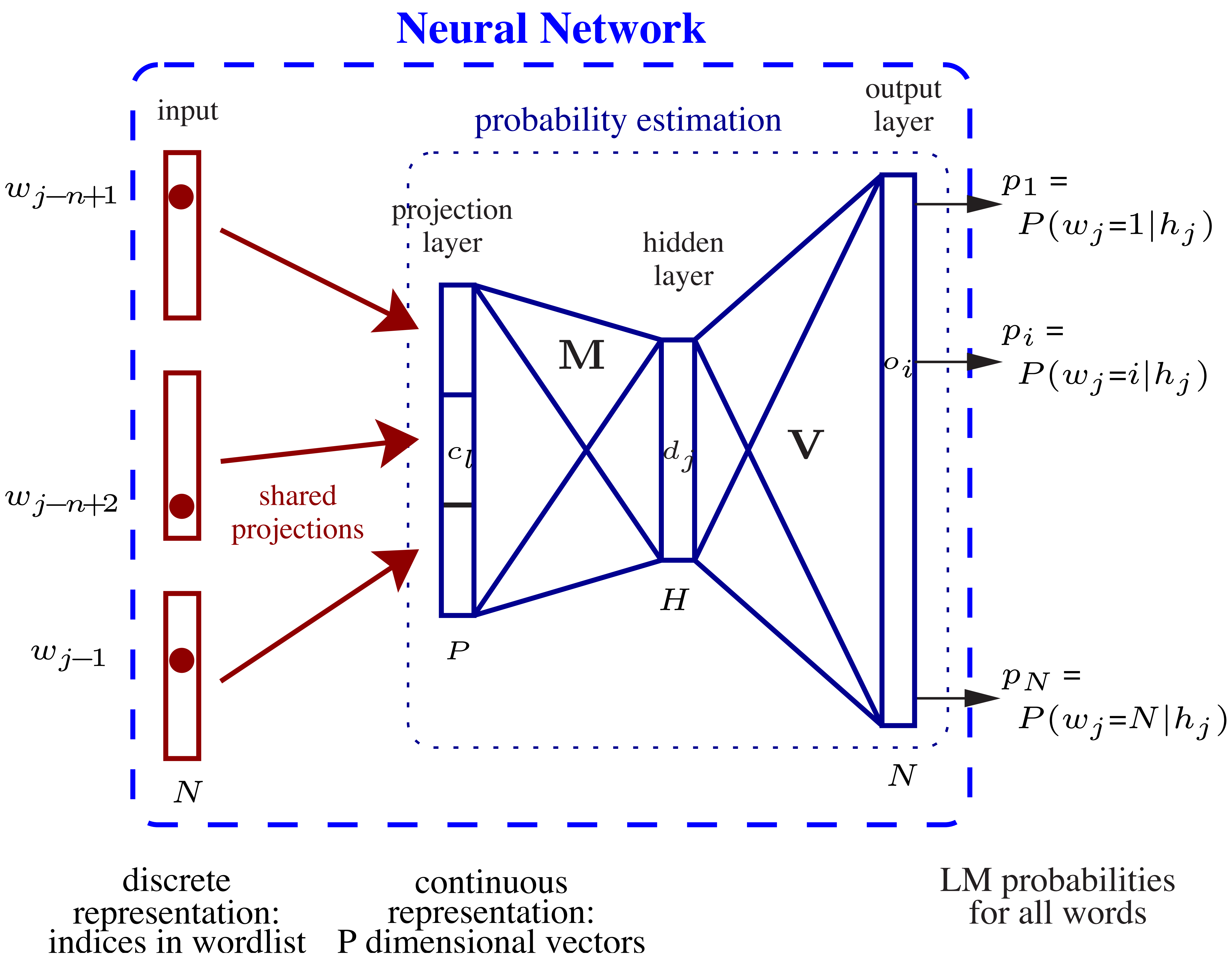
Continuous space language models



[S06]: Schwenk et al., “Continuous space language models for SMT”, ACL, 06

NN language model

- Project all the words of the context $h_j = w_{j-n+1}, \ldots, w_{j-1}$ to their dense forms
- Then, calculate the language model probability $Pr(w_j = i | h_j)$ for the given context h_j



NN language model

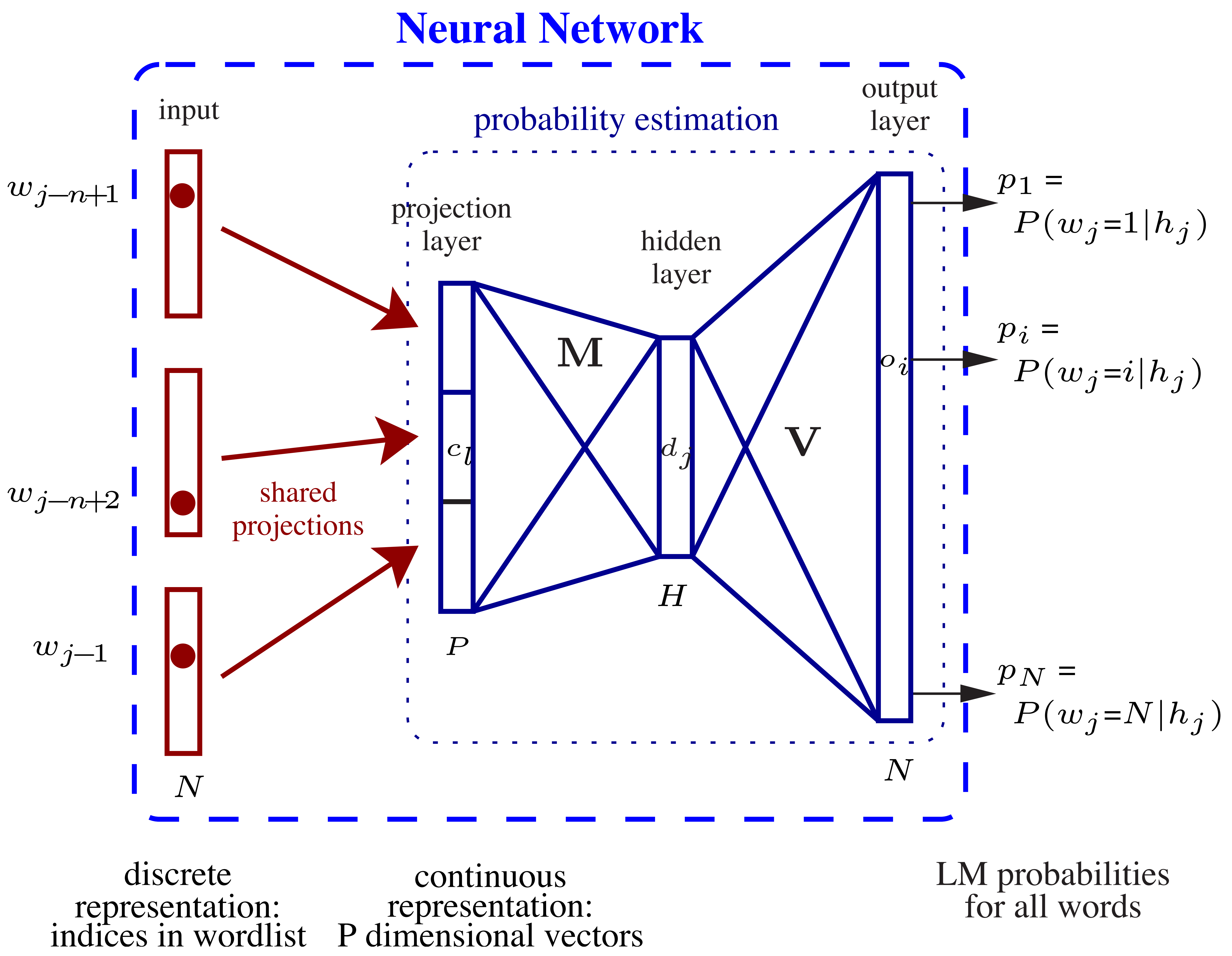
- Dense vectors of all the words in context are concatenated forming the first hidden layer of the neural network
- Second hidden layer:

  $d_j = \tanh(\Sigma m_{jl} c_l + b_j) \ \forall j = 1, \ldots, H$

- Output layer:

  $o_i = \Sigma v_{ij} d_j + b'_i \ \ \forall i = 1, \ldots, N$

- p_i → softmax output from the ith neuron → $\Pr(w_j = i \mid h_j)$



NN language model

- Model is trained to minimise the following loss function:

$$L = \boxed{\sum_{i=1}^{N} t_i \log p_i} + \boxed{\epsilon \left(\sum_{kl} m_{kl}^2 + \sum_{ik} v_{ik}^2 \right)}$$

- Here, t_i is the target output 1-hot vector (1 for next word in the training instance, 0 elsewhere)

- First part: Cross-entropy between the target distribution and the distribution estimated by the NN

- Second part: Regularization term

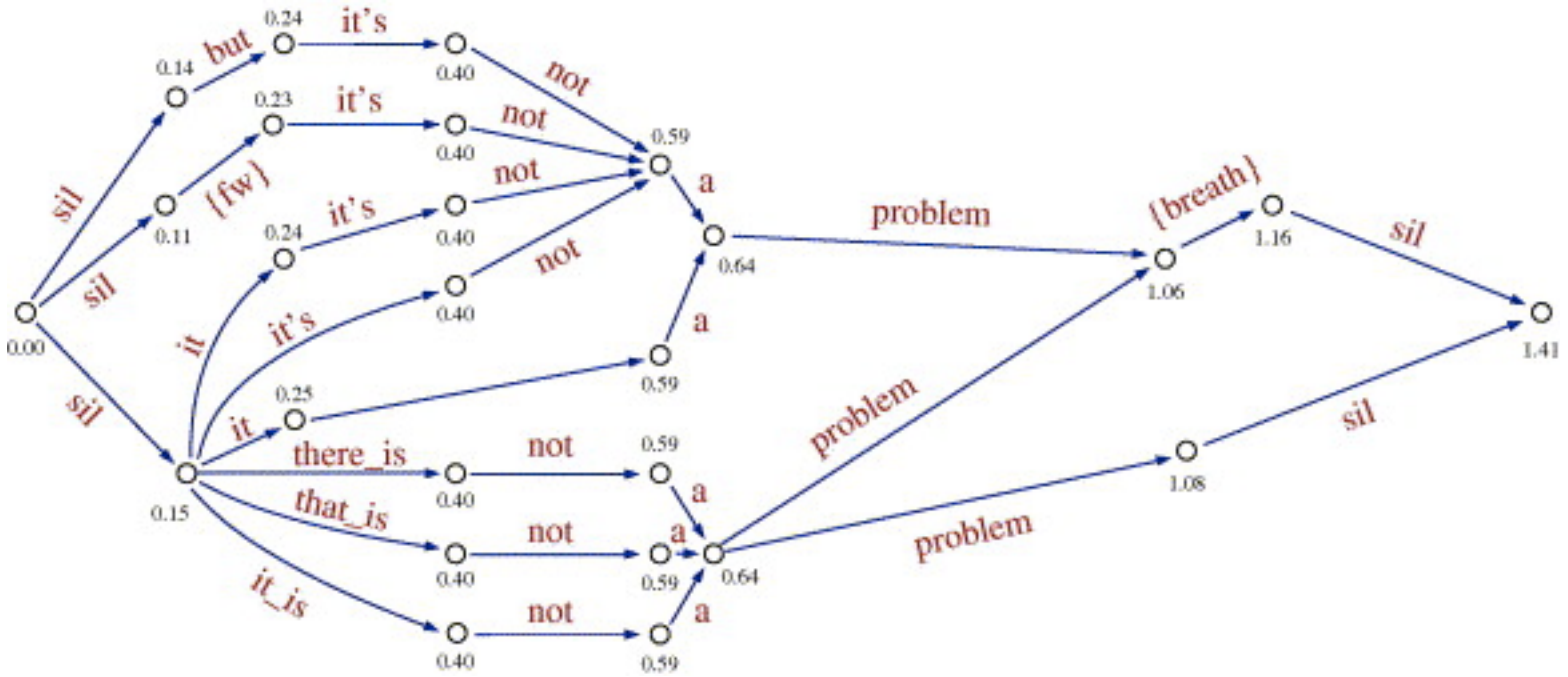
Decoding with NN LMs

- Two main techniques used to make the NN LM tractable for large vocabulary ASR systems:

  1. Lattice rescoring

  2. Shortlists

Use NN language model via lattice rescoring



- Lattice — Graph of possible word sequences from the ASR system using an Ngram backoff LM
- Each lattice arc has both acoustic/language model scores.
- LM scores on the arcs are replaced by scores from the NN LM

Decoding with NN LMs

- Two main techniques used to make the NN LM tractable for large vocabulary ASR systems:

    1. Lattice rescoring

    2. Shortlists

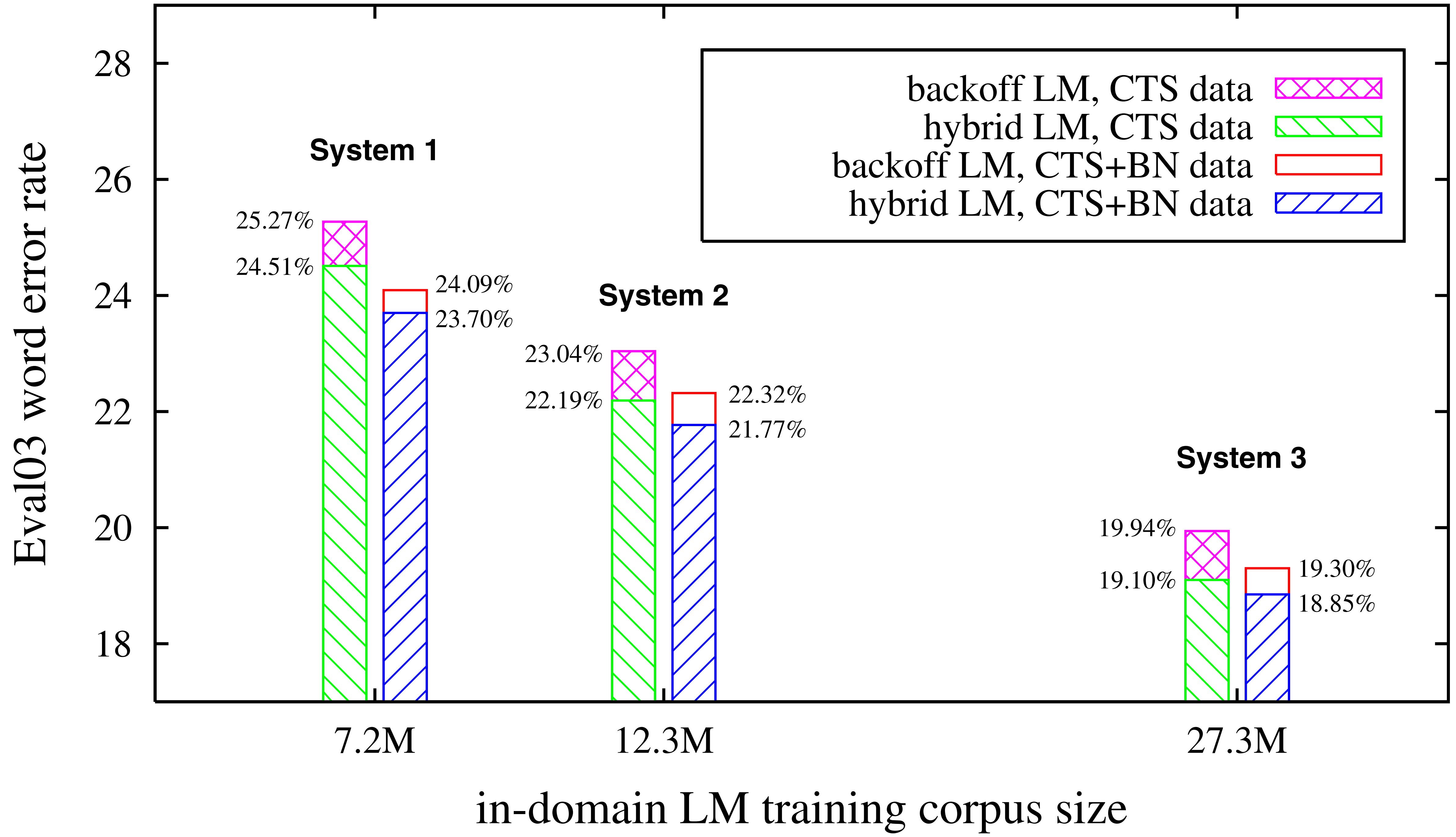
Shortlist

- Softmax normalization of the output layer is an expensive operation esp. for large vocabularies
- Solution: Limit the output to the s most frequent words.
  - LM probabilities of words in the short-list are calculated by the NN
  - LM probabilities of the remaining words are from Ngram backoff models

Results

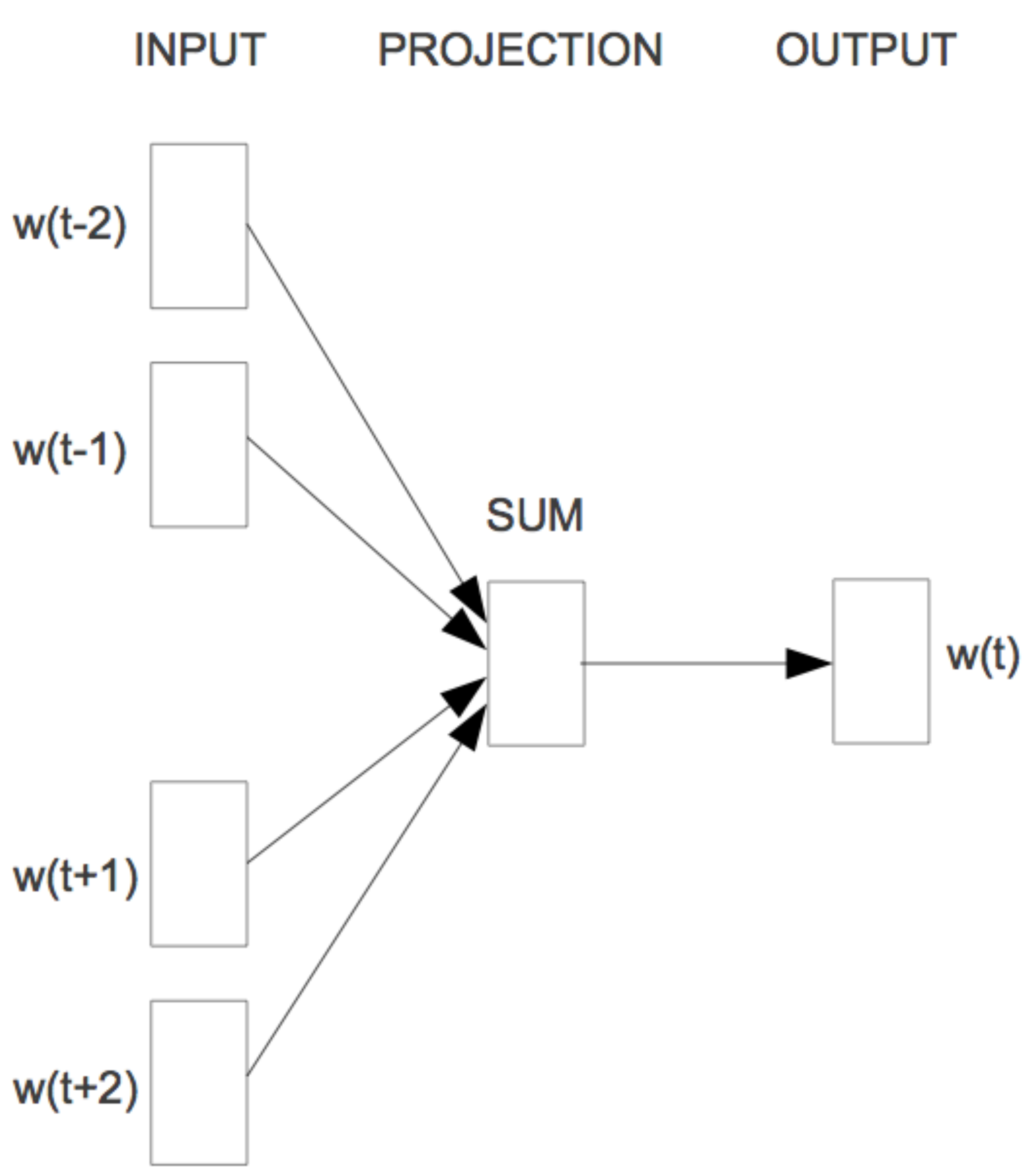
Table 3
Perplexities on the 2003 evaluation data for the back-off and the hybrid LM as a function of the size of the CTS training data

CTS corpus (words)	7.2 M	12.3 M	27.3 M
In-domain data only			
Back-off LM	62.4	55.9	50.1
Hybrid LM	57.0	50.6	**45.5**
Interpolated with all data			
Back-off LM	53.0	51.1	**47.5**
Hybrid LM	50.8	48.0	44.2

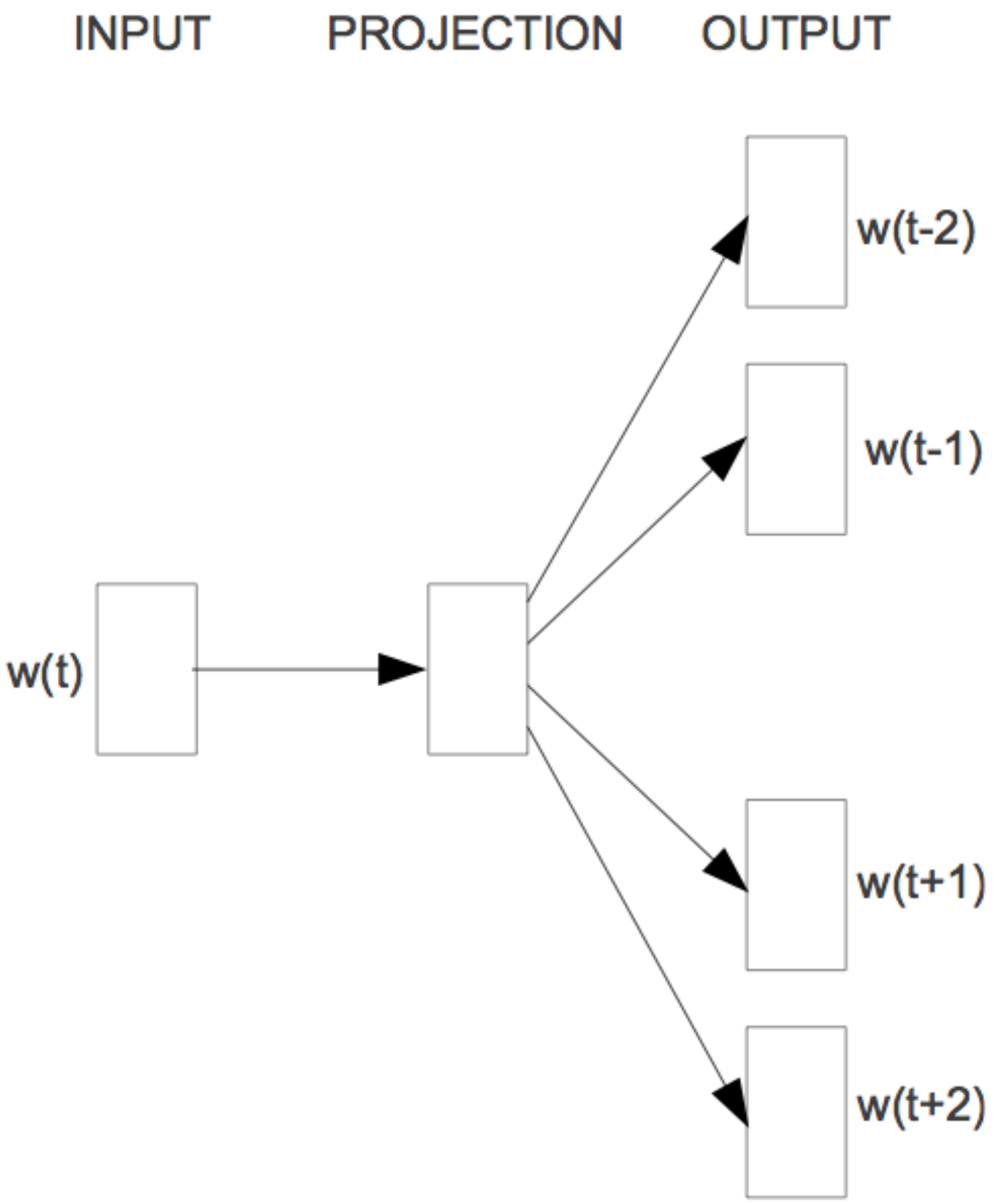


[S07]: Schwenk et al., “Continuous space language models”, CSL, 07

word2vec (to learn word embeddings)



Continuous bag-of-words
CBOW



Skip-gram

Image from: Mikolov et al., “Efficient Estimation of Word Representations in Vector Space”, ICLR 13

Bias in word embeddings

Gender bias in word embeddings

When you envision a nurse, a woman most likely pops into your mind. If you imagine an accomplished executive, on the other hand, it's quite likely you're thinking about a man.

It's not just you, though. The machine learning algorithms that target ads at us, prune our search results, or sort resumes for recruiters are all plagued by gendered stereotypes.

Algorithms that model natural language transform words into vectors, and similar words should be near each other in this vector space. Unfortunately, our models have learned to capture the biases present in the real-life data on which we train them. In word embedding space, for example, the relationship between "he" and "she" mirrors that of "programmer" and "homemaker". When we train our machine learning models on embeddings like these, a recruiter searching for "programmers" will leave female resumes at the bottom of the pile.

The following visualization allows you to define a category and see how words in that category relate to gender, by projecting them onto the axis representing gender in word embedding space. It is intended to encourage you to think critically about the tools you use and to consider carefully before treating anything as a black box. Feel free to explore the categories, or create your own.

Choose your word category...

Personality ▼

Or type your own words...

charming, accomplished, ambitious, impressive

Image from:http://wordbias.umiacs.umd.edu/

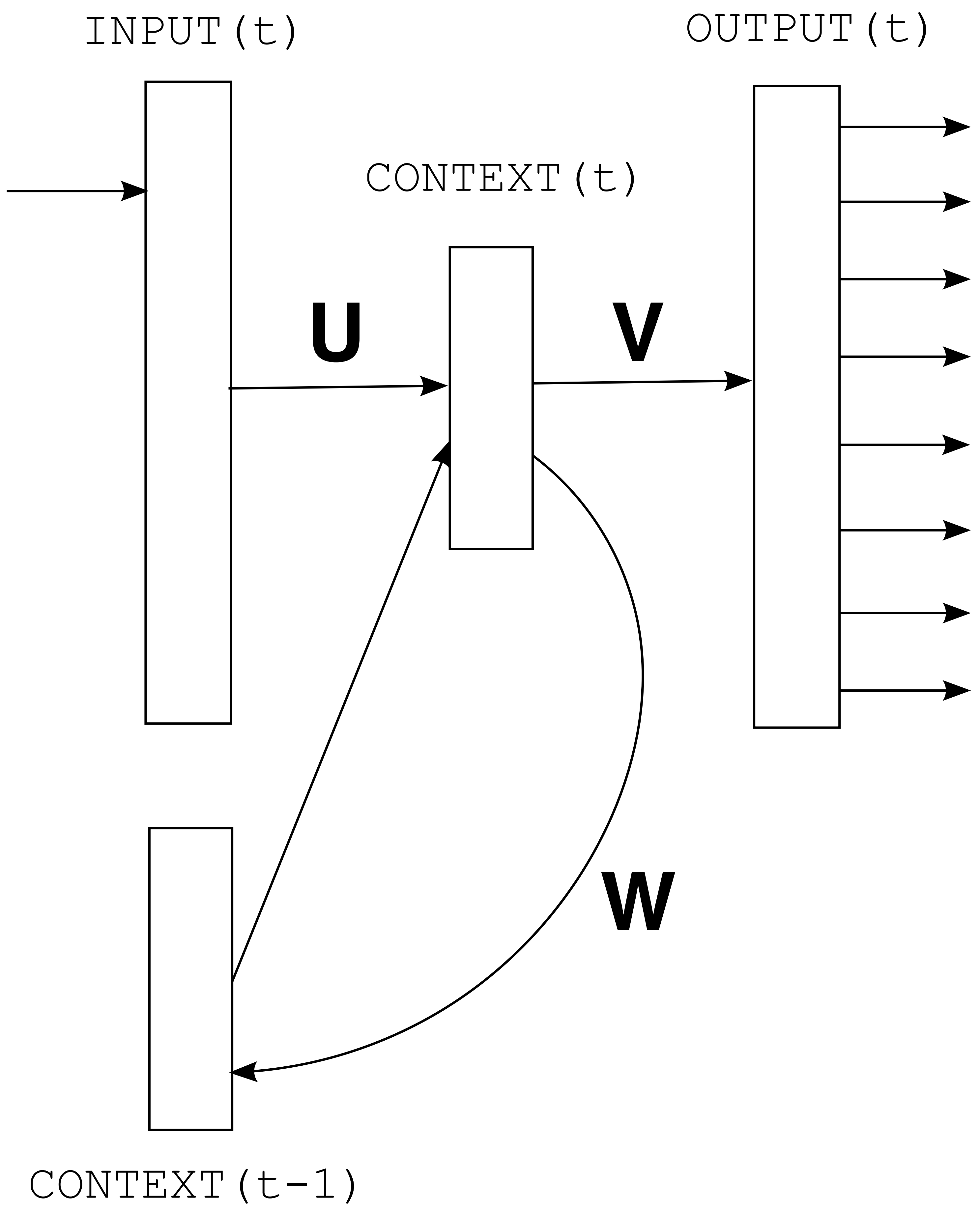
Longer word context?

- What have we seen so far: A feedforward NN used to compute an Ngram probability $Pr(w_j = i | h_j)$ (where h_j encodes the Ngram history)

- We know Ngrams are limiting:
  Alice who had attempted <u>the assignment asked</u> the lecturer

- How can we predict the next word based on the entire sequence of preceding words? Use recurrent neural networks (RNNs)

Simple RNN language model



- Current word, x_t
  Hidden state, s_t
  Output, y_t

$$s_t = f(Ux_t + Ws_{t-1})$$
$$o_t = \mathrm{softmax}(Vs_t)$$

- RNN is trained using the cross-entropy criterion

Image from: Mikolov et al., "Recurrent neural network based language model", Interspeech 10

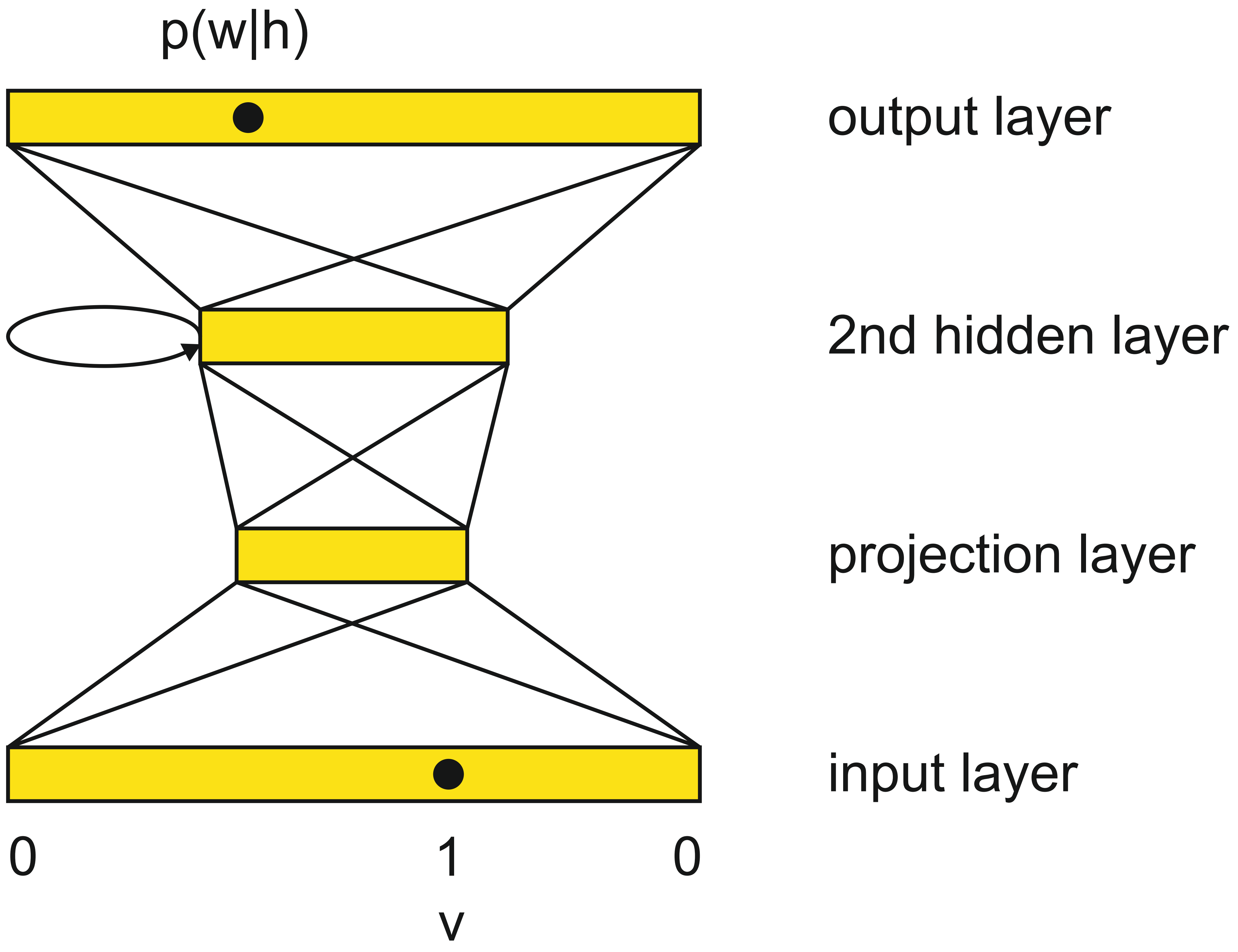
RNN-LMs

- Optimizations used for NNLMs are relevant to RNN-LMs as well (rescoring Nbest lists or lattices, using a shortlist, etc.)
- Perplexity reductions over Kneser-Ney models:

Model	# words	PPL	WER
KN5 LM	200K	336	16.4
KN5 LM + RNN 90/2	200K	271	15.4
KN5 LM	1M	287	15.1
KN5 LM + RNN 90/2	1M	225	14.0
KN5 LM	6.4M	221	13.5
KN5 LM + RNN 250/5	6.4M	156	11.7

Image from: Mikolov et al., "Recurrent neural network based language model", Interspeech 10

LSTM-LMs



- Vanilla RNN-LMs unlikely to show full potential of recurrent models due to issues like vanishing gradients
- LSTM-LMs: Similar to RNN-LMs except use LSTM units in the 2nd hidden (recurrent) layer

Image from: Sundermeyer et al., “LSTM NNs for Language Modeling”, IS 10

Comparing RNN-LMs with LSTM-LMs

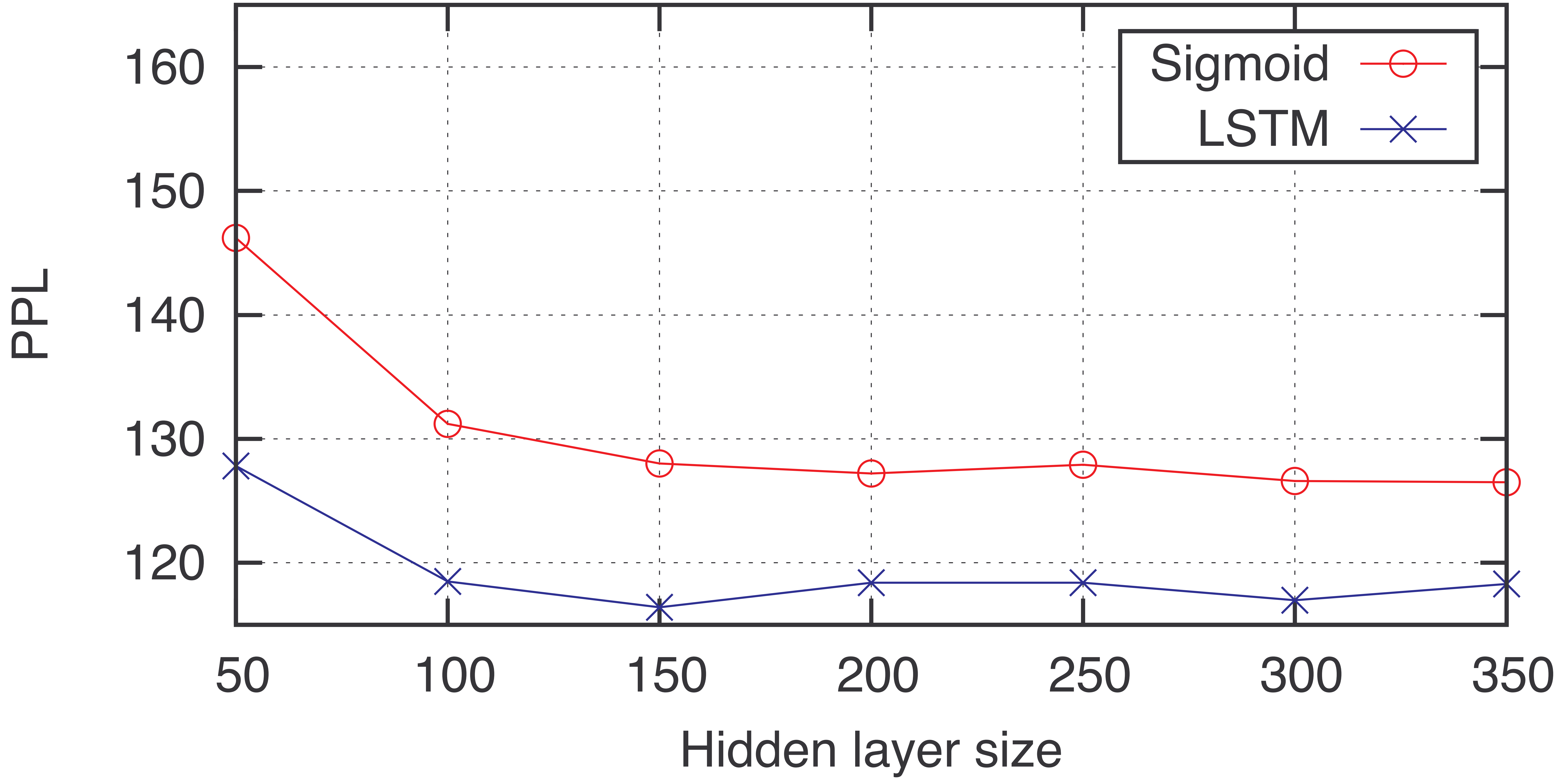


Image from: Sundermeyer et al., "LSTM NNs for Language Modeling",

Character-based RNN-LMs

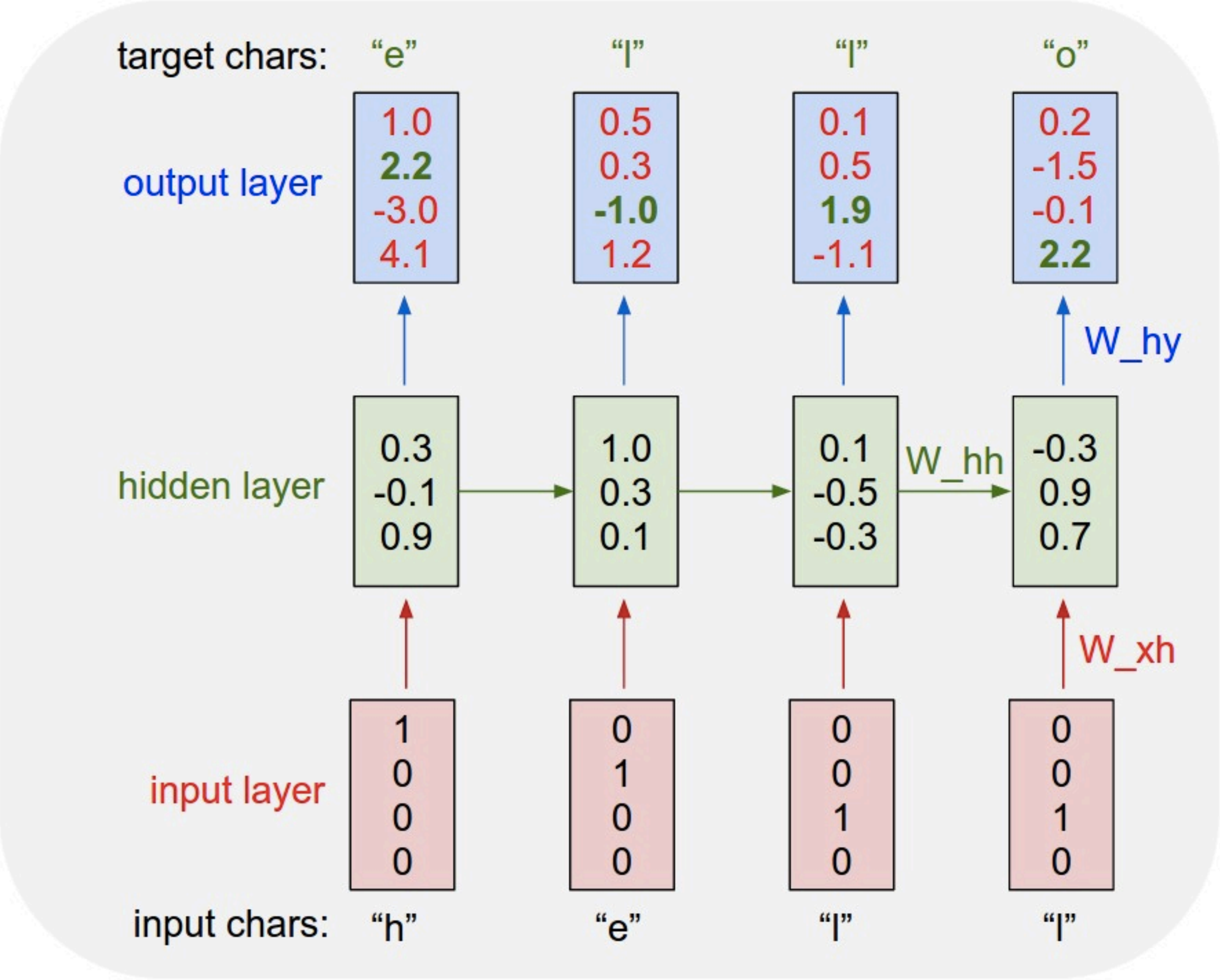


Image from: http://karpathy.github.io/2015/05/21/rnn-effectiveness/
Good tutorial available at https://github.com/yunjey/pytorch-tutorial/blob/master/tutorials/02-intermediate/language_model/main.py#L30-L50

Generate text using a trained character-based LSTM-LM

VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

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

Generate text using an LM trained on Obama speeches

Good morning. One of the borders will be able to continue to be here today. We have to say that the partnership was a partnership with the American people and the street continually progress that is a process and distant lasting peace and support that they were supporting the work of concern in the world. They were in the streets and communities that could have to provide steps to the people of the United States and Afghanistan. In the streets — the final decade of the country that will include the people of the United States of America. Now, humanitarian crisis has already rightly achieved the first American future in the same financial crisis that they can find reason to invest in the world.

Thank you very much. God bless you. God bless you. Thank you.

Source:https://medium.com/@samim/obama-rnn-machine-generated-political-speeches-c8abd18a2ea0

NN trained on Trump's speeches (now defunct)

Source:https://twitter.com/deepdrumpf?lang=en

Common RNNLM training tricks

- SGD fares very well on this task (compared to other optimizers like Adagrad, Adam, etc.).
- Use dropout regularization
- Truncated BPTT
- Use mini batches to aggregate gradients during training
  - In batched RNNLMs, process multiple sentences at the same time
  - Handle variable length sequences using padding and masking
  - To be judicious about padding, sort the sentences in the corpus by length before creating batches

Spotlight: Regularizing and Optimizing LSTM Language Models (Merity et al. 2018)

- No special model, just better regularisation + optimization
- Dropout on recurrent connections and embeddings
- SGD w/ averaging triggered when model is close to convergence
- Weight tying between embedding and softmax layers
- Reduced embedding sizes
- https://github.com/salesforce/awd-lstm-lm

Spotlight:
On the State of the art of Evaluation in Neural Language Models (Melis et al., 2018)

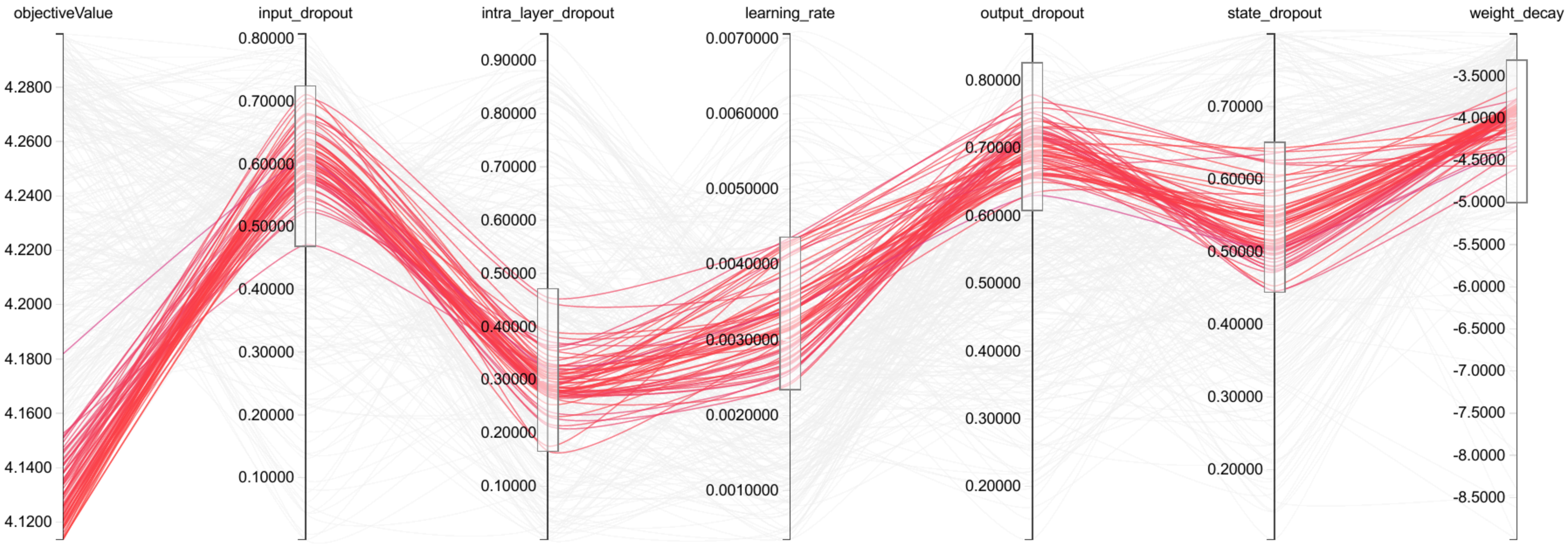


Image from:https://arxiv.org/pdf/1707.05589.pdf