Deep Learning for Natural Language Processing

Stephen Clark
University of Cambridge and DeepMind

2. Feedforward Neural Networks for NLP

Stephen Clark
University of Cambridge and DeepMind

Named Entity Recognition (as a tagging task)

```
England|I-LOC 's|O fencers|O won|O gold|O on|O
day|I-TIME 4|I-TIME in|O Delhi|I-LOC with|O a|O medal|O
-winning|O performance|O .|O

This|O is|O Prof.|I-PER Black|I-PER 's|O second|O
gold|O of|O the|O Games|O .|O
```

Example tagset: {I-PER, I-ORG, I-LOC, I-TIME, O}

DeepMind

UNIVERSITY OF CAMBRIDGE

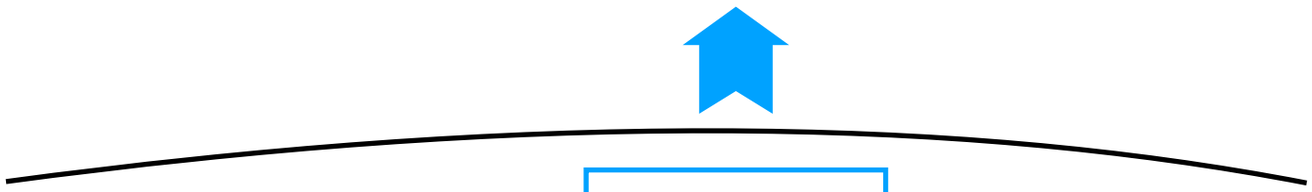
NER as Ambiguity Resolution

DeepMind

UNIVERSITY OF CAMBRIDGE

Discrete Features for NER

curr_word='Somerset'
next_word='cricket'
next_next_word='club'
next_bigram='cricket club'
...
prev_word='for'
prev_prev_word='bowled'
prev_bigram='bowled for'
...
curr_pos='NNP'
next_pos='NN'
next_next_pos='NN'
next_bigram_pos='NN NN'
....



Botham bowled for Somerset cricket club in the 1980s

NNP VBD IN NNP NN NN IN DT NNS

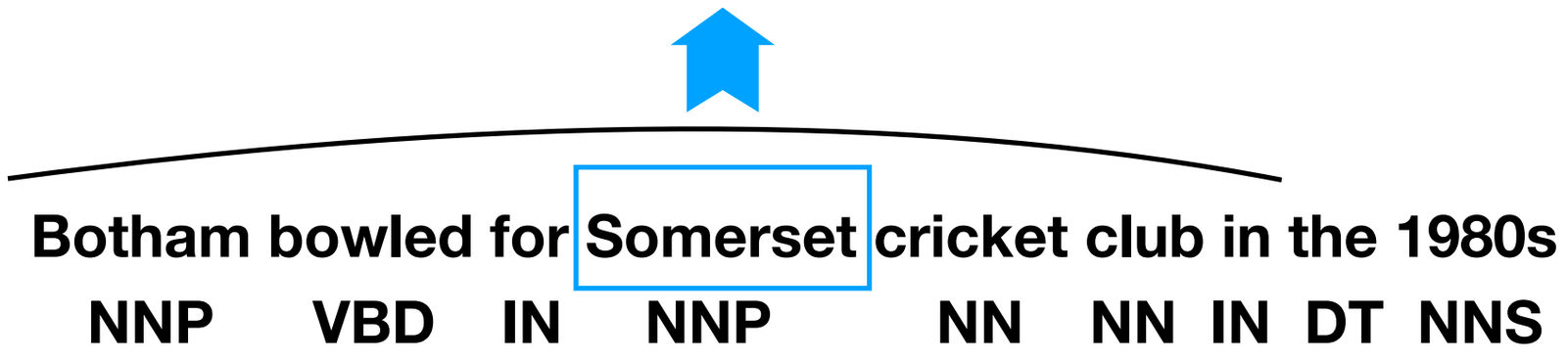
DeepMind

UNIVERSITY OF CAMBRIDGE

Indicator Features for NER

$$f_i(C, t) = \begin{cases} 1 & \text{if } \texttt{word(C) = Somerset} \ \& \ \text{t} = \text{I-ORG} \\ 0 & \text{otherwise} \end{cases}$$

$\texttt{word(C) = Somerset}$ is a *contextual predicate*



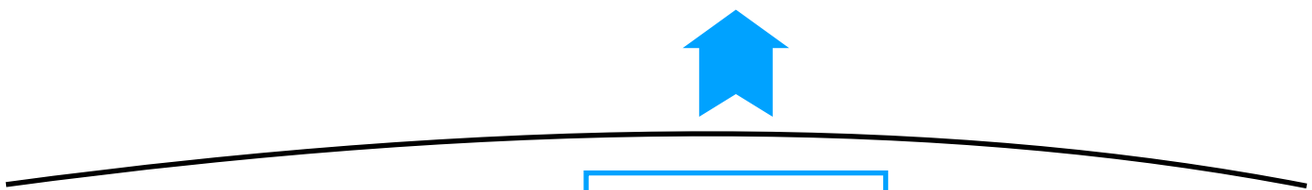
DeepMind

UNIVERSITY OF CAMBRIDGE

Log-Linear Classification Model

$$p(t|C) = \frac{1}{Z(C)} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(C,t)\right)$$

$$Z(C) = \sum_{t \in T} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(C,t)\right) \quad \text{where } T \text{ is the tagset}$$



Botham bowled for Somerset cricket club in the 1980s

NNP VBD IN NNP NN NN IN DT NNS

DeepMind

UNIVERSITY OF CAMBRIDGE

Decoding a Log-Linear Tagging Model

- The conditional probability of a tag sequence $t_1 \dots t_n$ is

$$p(t_1 \dots t_n | w_1 \dots w_n) \approx \prod_{i=1}^{n} p(t_i | C_i)$$

given a sentence $w_1 \dots w_n$ and contexts $C_1 \dots C_n$

- Beam search or Viterbi is used to find the most probable sequence

I-PER	O	O	I-ORG	I-ORG	I-ORG	O	O	I-TIME	
Botham	bowled	for	Somerset	cricket	club	in	the	1980s	
NNP	VBD	IN	NNP	NN	NN	IN	DT	NNS	

DeepMind

UNIVERSITY OF CAMBRIDGE

NER Feature Set I

Condition	Contextual predicate
$freq(w_i) < 5$	X is prefix of w_i, $\|X\| \leq 4$ X is suffix of w_i, $\|X\| \leq 4$ w_i contains a digit w_i contains uppercase character w_i contains a hyphen
$\forall w_i$	$w_i = X$ $w_{i-1} = X, w_{i-2} = X$ $w_{i+1} = X, w_{i+2} = X$
$\forall w_i$	$\text{POS}_i = X$ $\text{POS}_{i-1} = X, \text{POS}_{i-2} = X$ $\text{POS}_{i+1} = X, \text{POS}_{i+2} = X$
$\forall w_i$	$\text{NE}_{i-1} = X$ $\text{NE}_{i-2}\text{NE}_{i-1} = XY$

Taken from Curran and Clark (2003)

DeepMind

UNIVERSITY OF CAMBRIDGE

NER Feature Set II

Condition	Contextual predicate
$freq(w_i) < 5$	w_i contains period w_i contains punctuation w_i is only digits w_i is a number w_i is {upper,lower,title,mixed} case w_i is alphanumeric length of w_i w_i has only Roman numerals w_i is an initial (X.) w_i is an acronym (ABC, A.B.C.)
$\forall w_i$	memory NE tag for w_i unigram tag of w_{i+1} unigram tag of w_{i+2}
$\forall w_i$	w_i in a gazetteer w_{i-1} in a gazetteer w_{i+1} in a gazetteer
$\forall w_i$	w_i not lowercase and $f_{\text{lc}} > f_{\text{uc}}$
$\forall w_i$	unigrams of word type bigrams of word types trigrams of word types

DeepMind

UNIVERSITY OF CAMBRIDGE

Some Accuracy Numbers

English devel.	Precision	Recall	$F_{\beta=1}$
LOC	91.75%	93.20%	92.47
MISC	88.34%	82.97%	85.57
ORG	83.54%	85.53%	84.52
PER	94.26%	95.39%	94.82
Overall	90.15%	90.56%	90.35

English test	Precision	Recall	$F_{\beta=1}$
LOC	84.97%	90.53%	87.66
MISC	76.77%	75.78%	76.27
ORG	79.60%	79.41%	79.51
PER	91.64%	90.79%	91.21
Overall	84.29%	85.50%	84.89

Taken from Curran and Clark (2003)

DeepMind

UNIVERSITY OF CAMBRIDGE

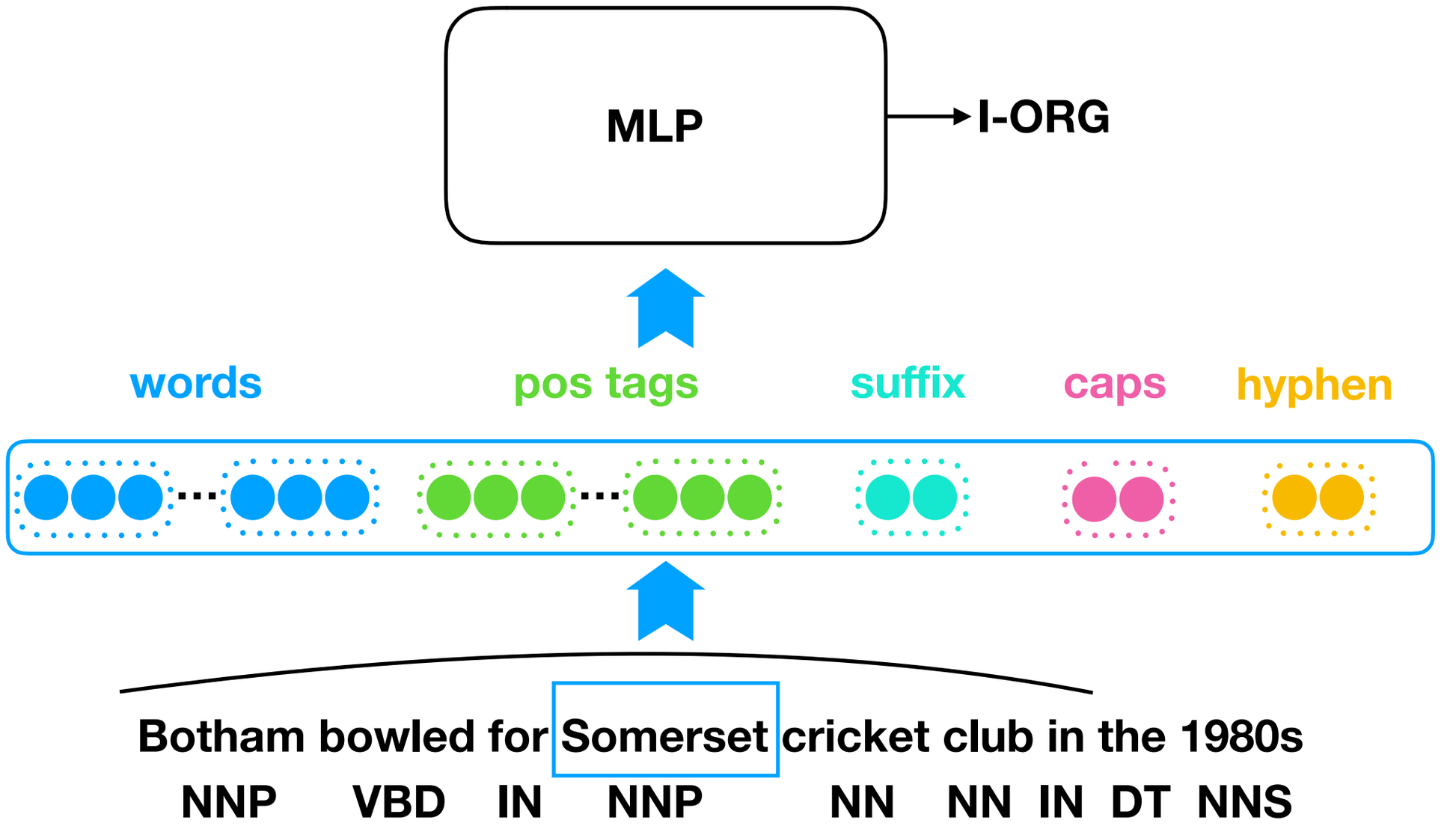
What's the Problem?

- Huge number of features (potentially millions)
- All features (esp. the combinations) defined manually
- All features are equally unlike each other (no 'sharing of statistical strength')
    - curr_word='Somerset' no closer to curr_word='Kent' than curr_word='Botham' or curr_word='cat'

DeepMind

UNIVERSITY OF CAMBRIDGE

Dense Features for Classification



DeepMind

UNIVERSITY OF CAMBRIDGE

The MLP Classifier

probabilistic output

softmax

linear trans

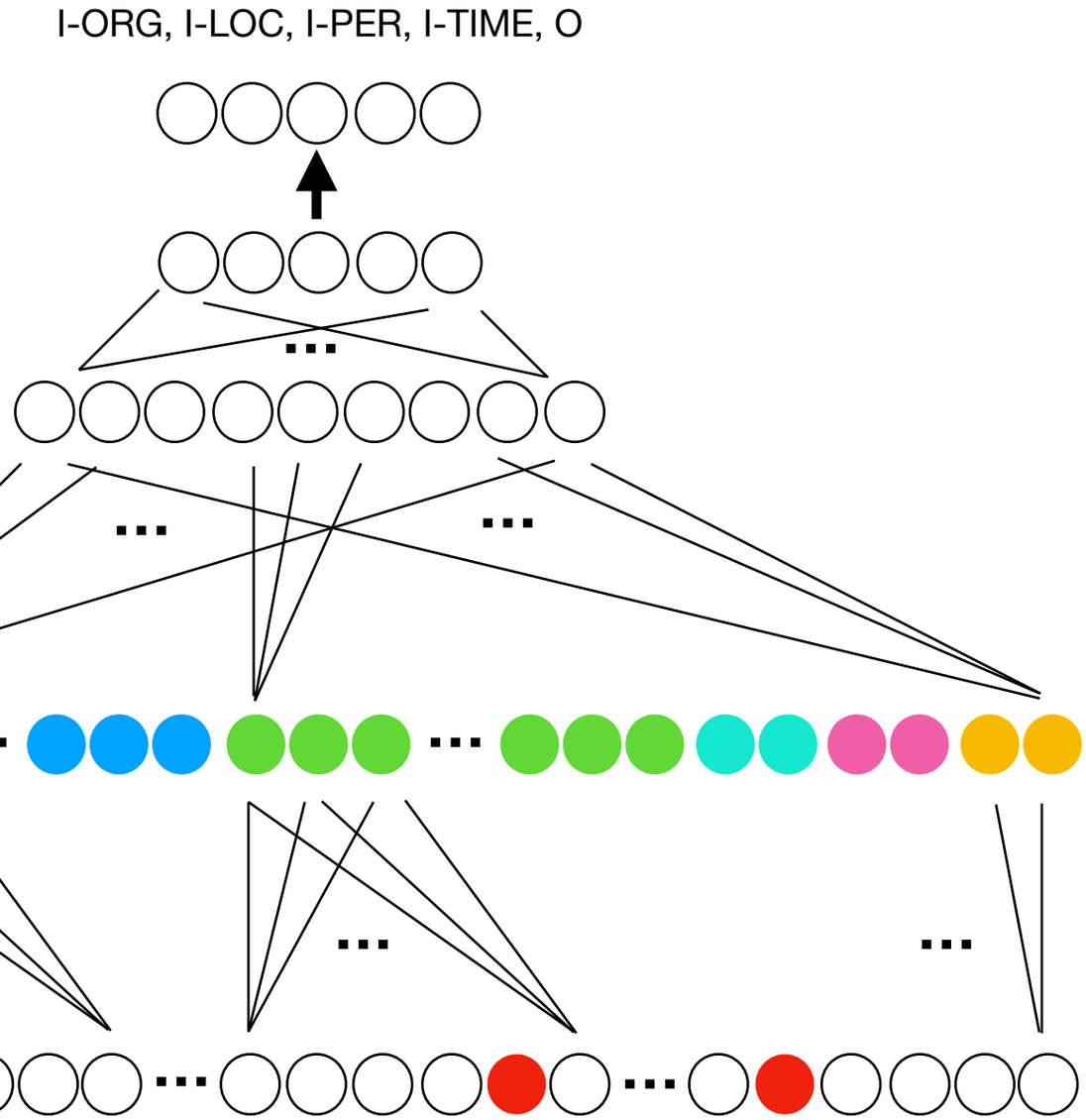
hidden layer

affine trans + non-linearity

dense input

embedding matrix

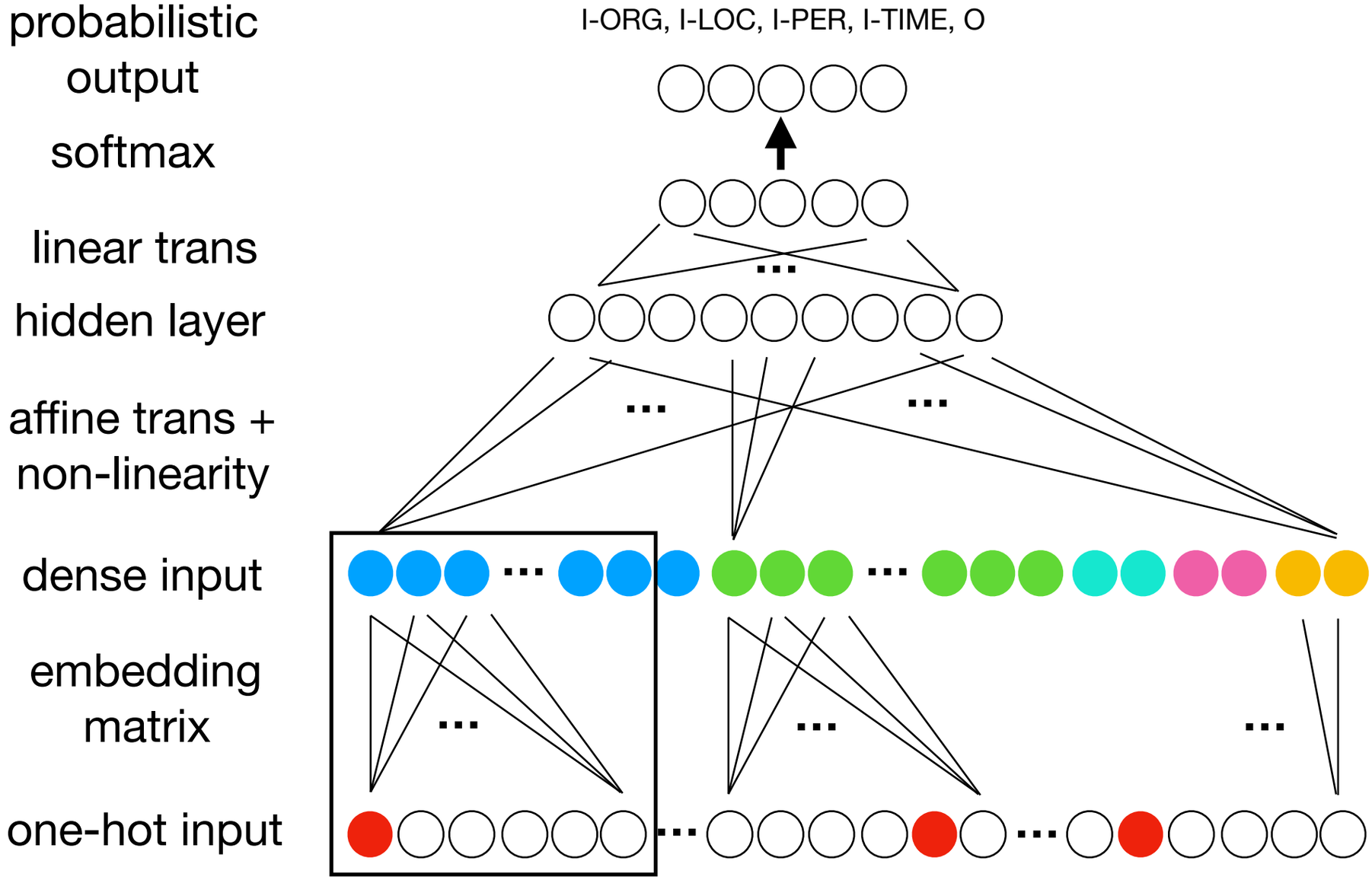
one-hot input



DeepMind

UNIVERSITY OF CAMBRIDGE

The MLP Classifier



DeepMind

UNIVERSITY OF CAMBRIDGE

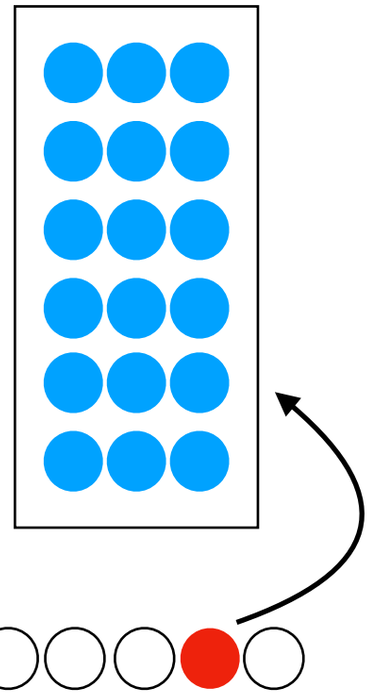
Word Embeddings

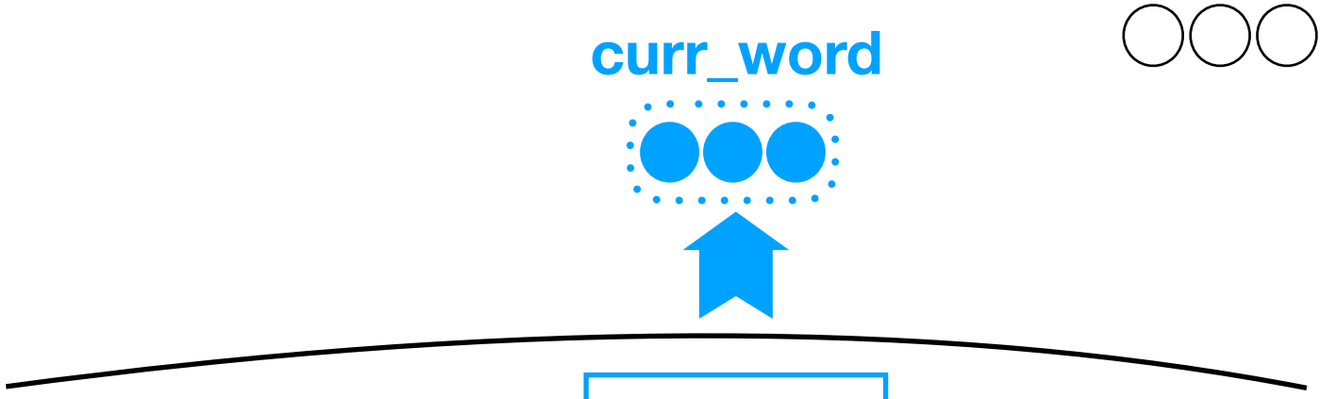
$$\mathbf{w^i} = \mathbf{f^i}\,\mathbf{E} \qquad \mathbf{w^i} \in \mathbf{R}^{1\times n},\ \mathbf{f^i} \in \{0,1\}^{1\times|V|},\ \mathbf{E} \in \mathbf{R}^{|V|\times n}$$

$\mathbf{w^i}$ is the embedding for the ith word

$\mathbf{f^i}$ is the one-hot vector for the ith word

$\mathbf{E}$ is the word embedding matrix (lookup table)

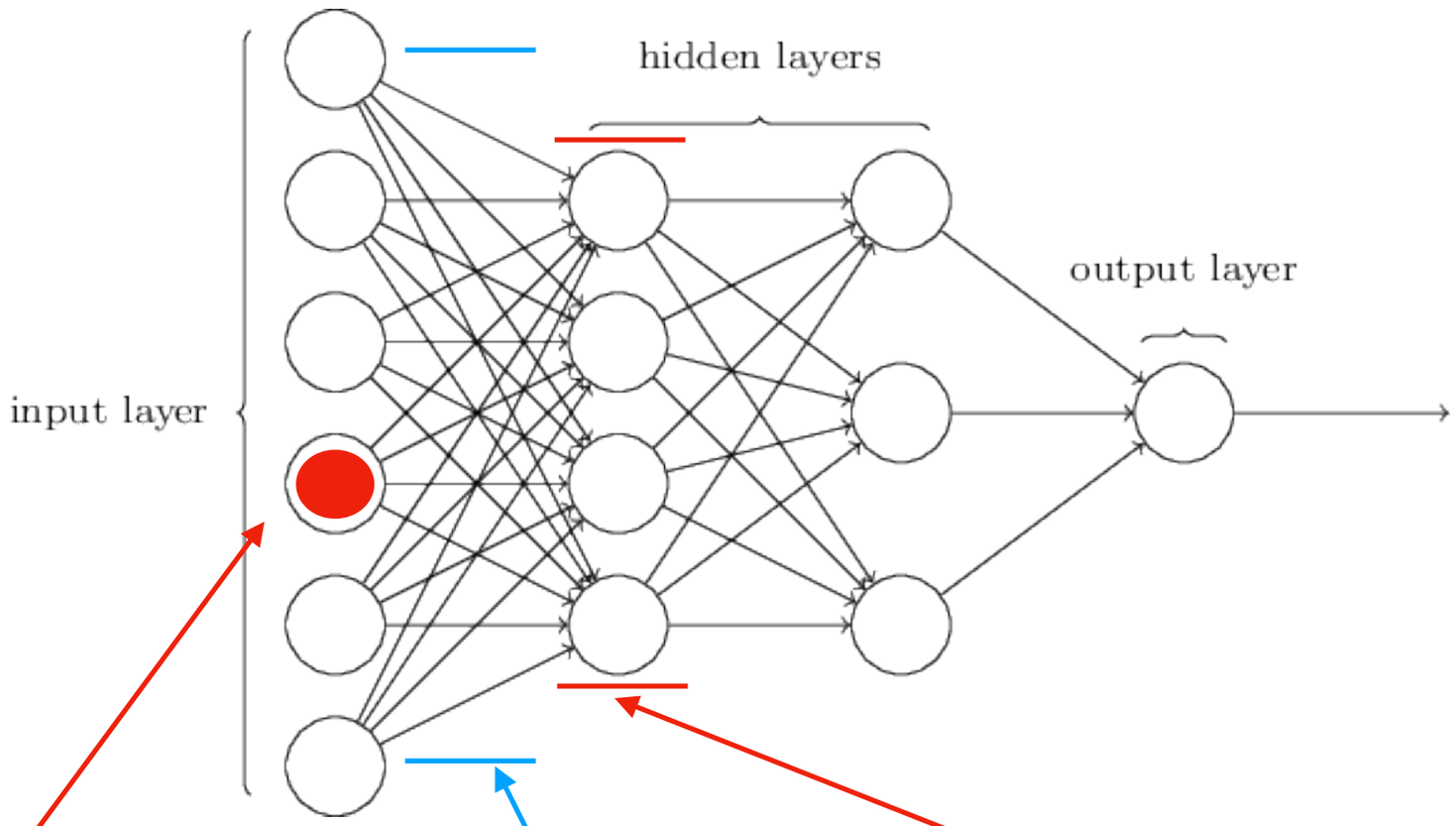




DeepMind

UNIVERSITY OF CAMBRIDGE

Word Embeddings



DeepMind

UNIVERSITY OF CAMBRIDGE

The MLP Classifier

probabilistic output

softmax

linear trans

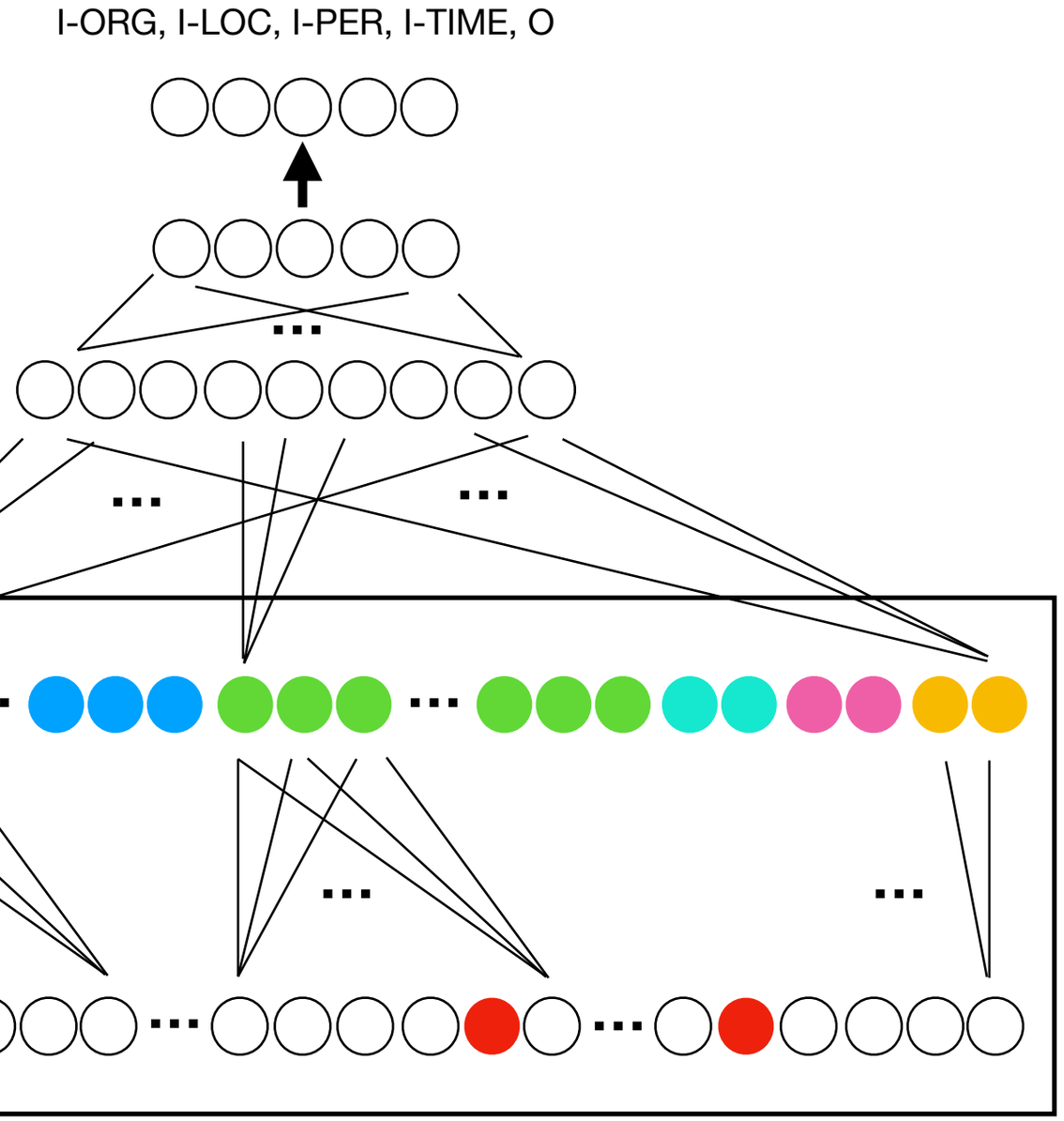
hidden layer

affine trans + non-linearity

dense input

embedding matrix

one-hot input



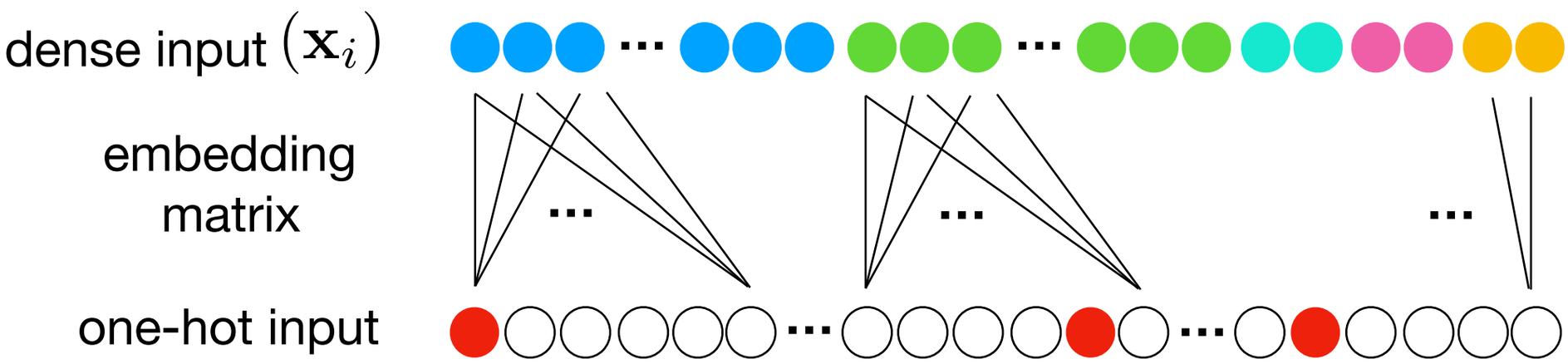
DeepMind

UNIVERSITY OF CAMBRIDGE

The MLP Classifier

$$\mathbf{x}_i = [\mathbf{e}_{w_{i-\lfloor k/2 \rfloor}}; \ldots; \mathbf{e}_{w_{i-1}}; \mathbf{e}_{w_i}; \mathbf{e}_{w_{i+1}}; \ldots; \mathbf{e}_{w_{i+\lfloor k/2 \rfloor}}; \mathbf{s}_{w_i}; \mathbf{c}_{w_i}; \mathbf{h}_{w_i}]$$

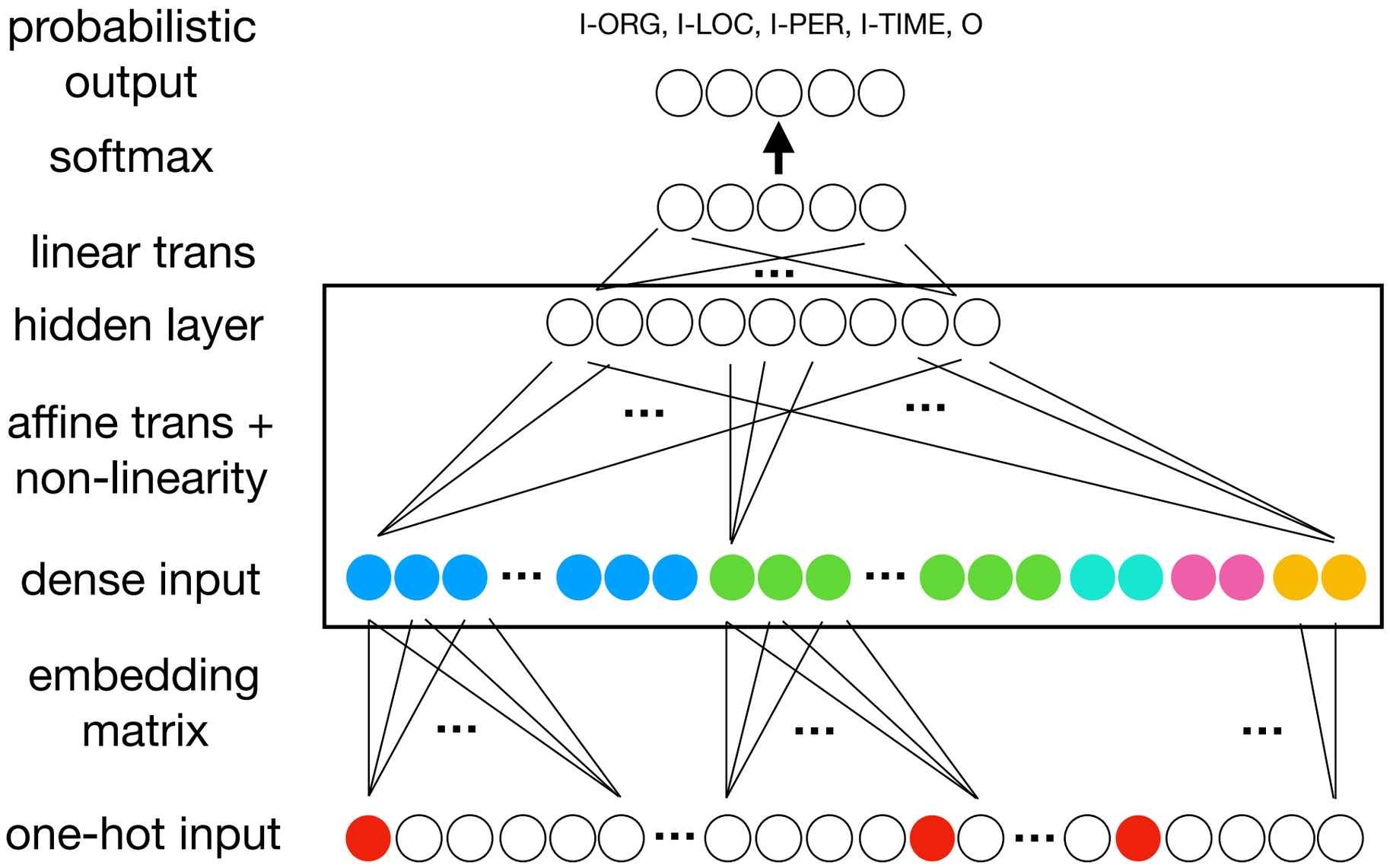
; is concatenation, *k* is the window size



DeepMind

UNIVERSITY OF CAMBRIDGE

The MLP Classifier

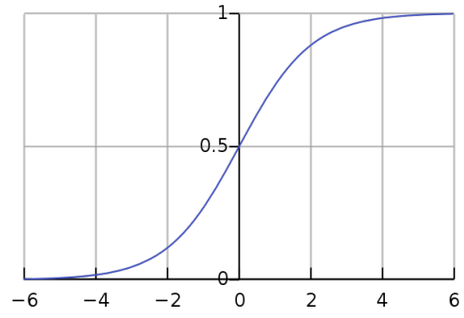


DeepMind

UNIVERSITY OF CAMBRIDGE

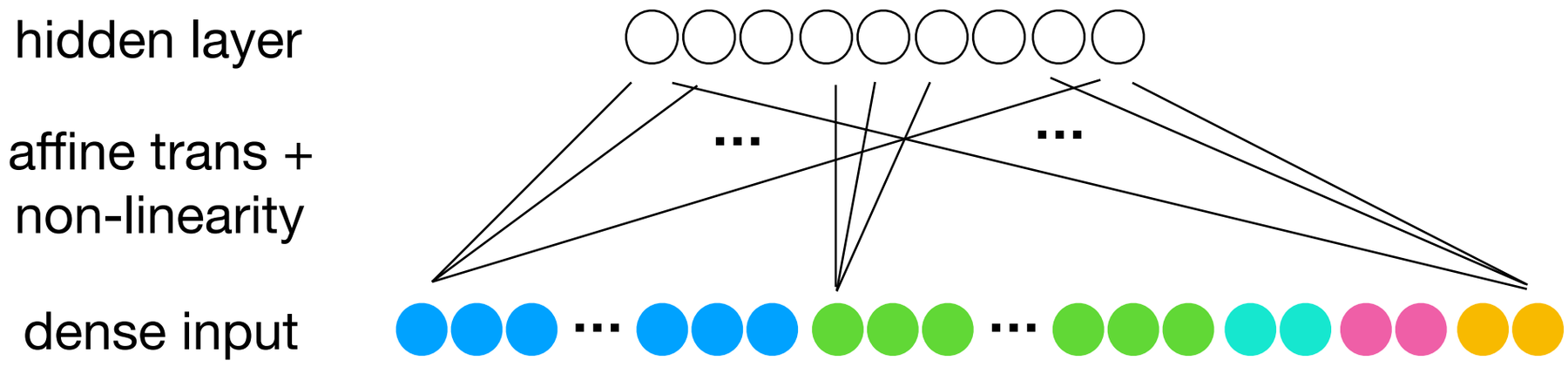
The MLP Classifier

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}.$$



$$\mathbf{h}_i^n = S(\mathbf{h}_i^{n-1}\,\mathbf{W}^n + \mathbf{b}^n)$$

$$\mathbf{h}_i^1 = S(\mathbf{x}_i\,\mathbf{W}^1 + \mathbf{b}^1) \qquad \mathbf{x}_i \in \mathbf{R}^{1\times N},\ \mathbf{W}^1 \in \mathbf{R}^{N\times h^1},\ \mathbf{b}^1 \in \mathbf{R}^{1\times h^1}$$



DeepMind

UNIVERSITY OF CAMBRIDGE

The MLP Classifier

probabilistic output

softmax

linear trans

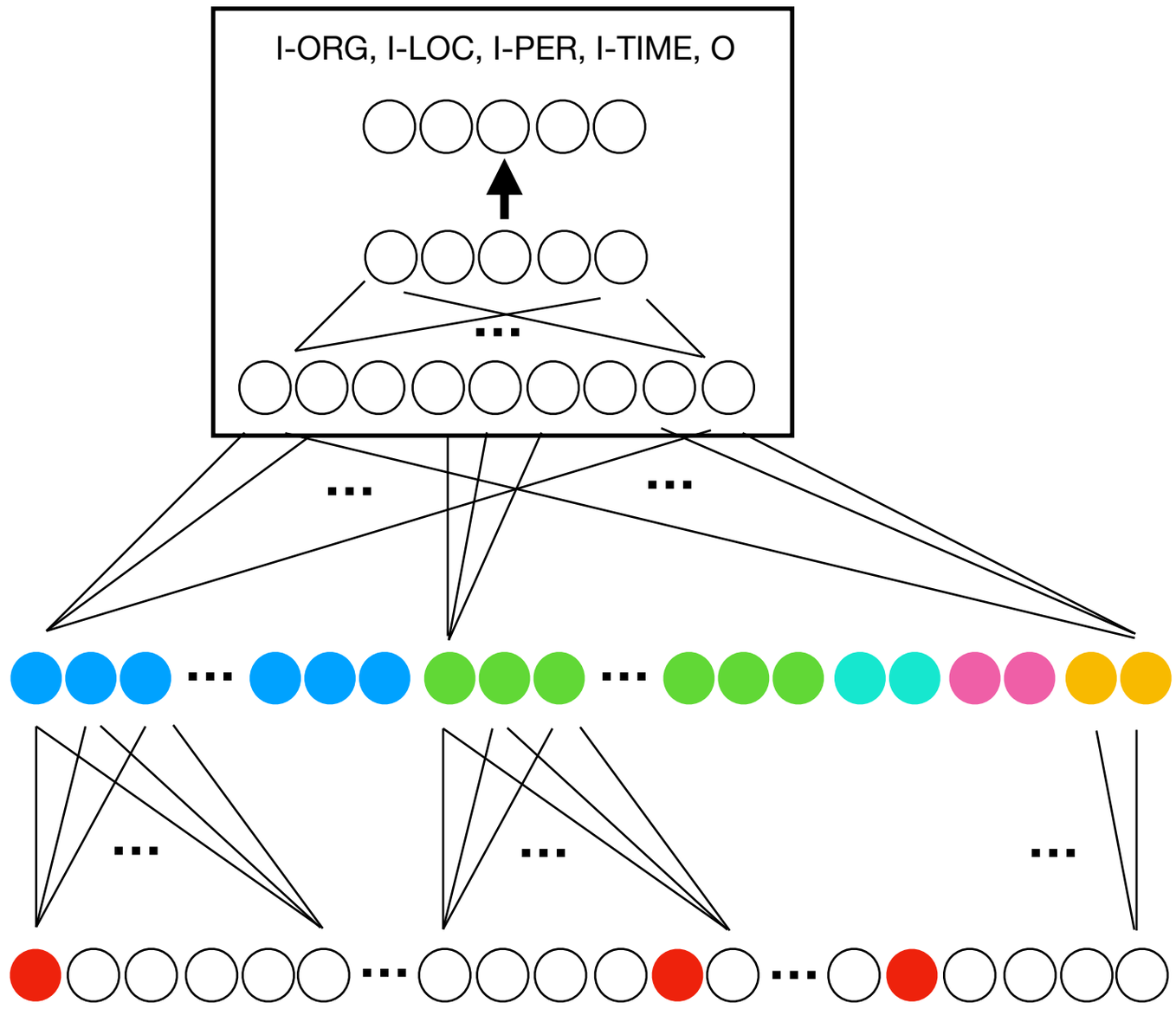
hidden layer

affine trans + non-linearity

dense input

embedding matrix

one-hot input



DeepMind

UNIVERSITY OF CAMBRIDGE

The MLP Classifier

$$\mathbf{L}_j = P(L_j) = \frac{e^{\mathbf{y}_j}}{\sum_k e^{\mathbf{y}_k}}$$

$$\mathbf{y} = \mathbf{h}_i^m \mathbf{U} \qquad \mathbf{U} \in \mathbf{R}^{h^m \times |L|}$$

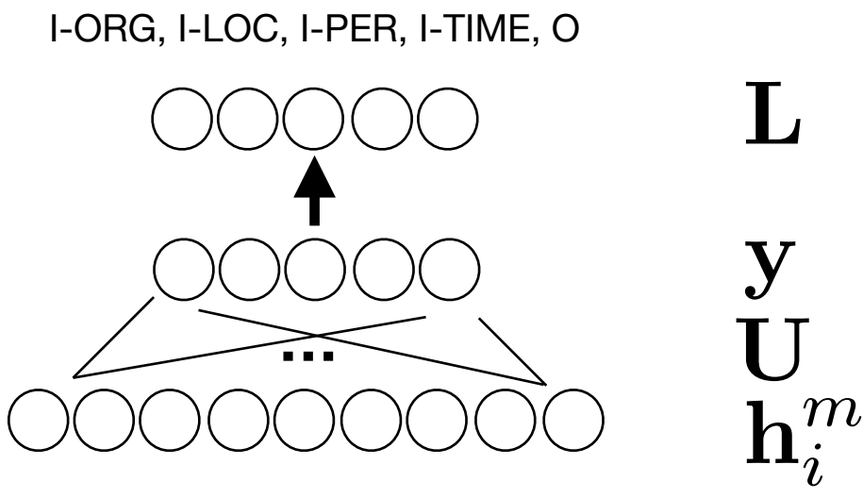
m is the number of layers, L is the NER label set

probabilistic output

softmax

linear trans

hidden layer



DeepMind

UNIVERSITY OF CAMBRIDGE

Supervised Training

- (*Context, Label*) pairs for training data

Ian Botham *bowled* for Somerset	**O**
Botham bowled *for* Somerset cricket	**O**
bowled for *Somerset* cricket club	**I-ORG**
for Somerset *cricket* club in	**I-ORG**
Somerset cricket *club* in the	**I-ORG**
cricket club *in* the 1980s	**O**

DeepMind

UNIVERSITY OF CAMBRIDGE

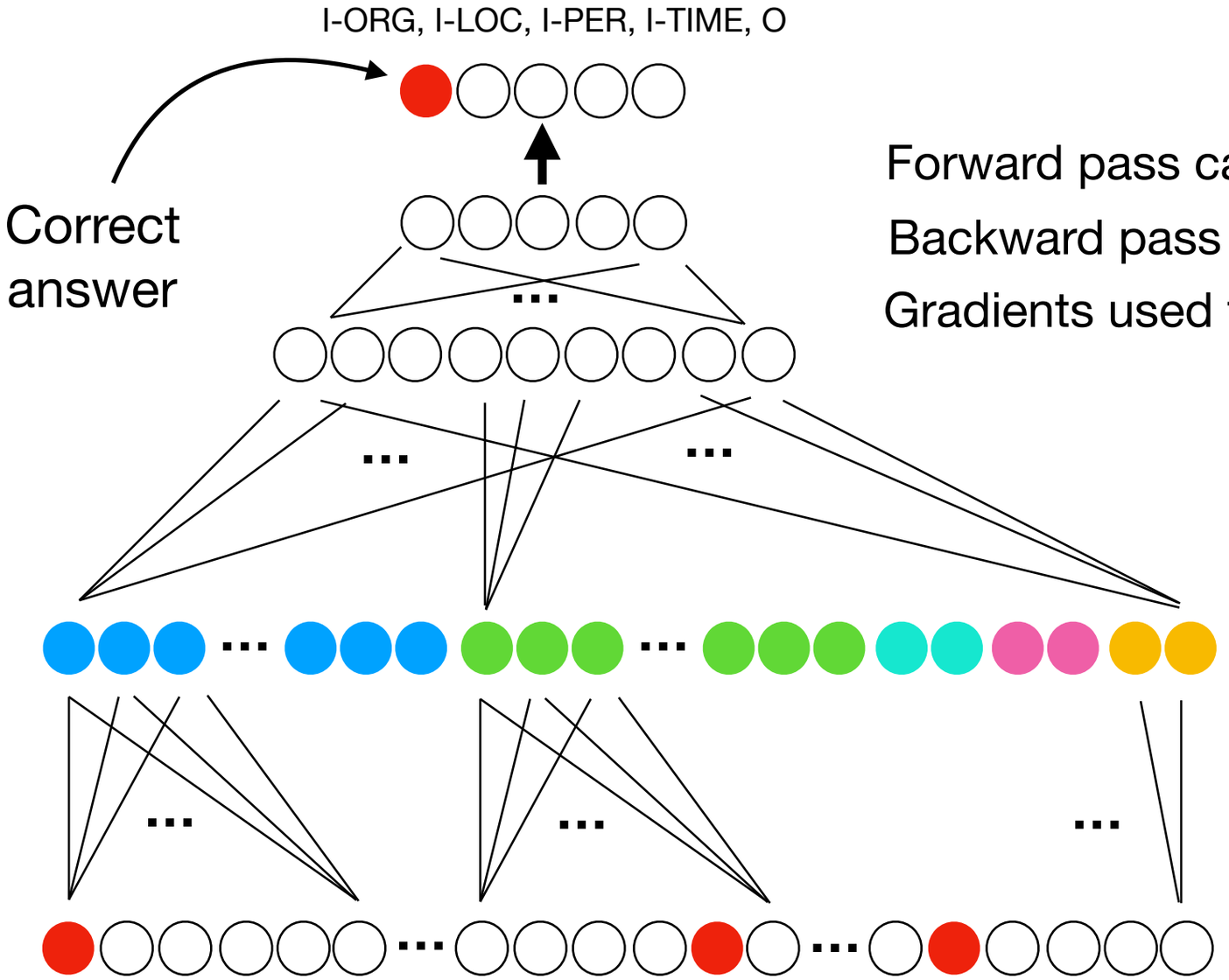
Supervised Training

- (*Context, Label*) pairs for training data
- Maximize log-likelihood of the training data
    - Initialize all parameters
    - Calculate gradients (using a subset of the data)
    - Update parameters
    - Repeat

DeepMind

UNIVERSITY OF CAMBRIDGE

Supervised Training



Forward pass calculates loss

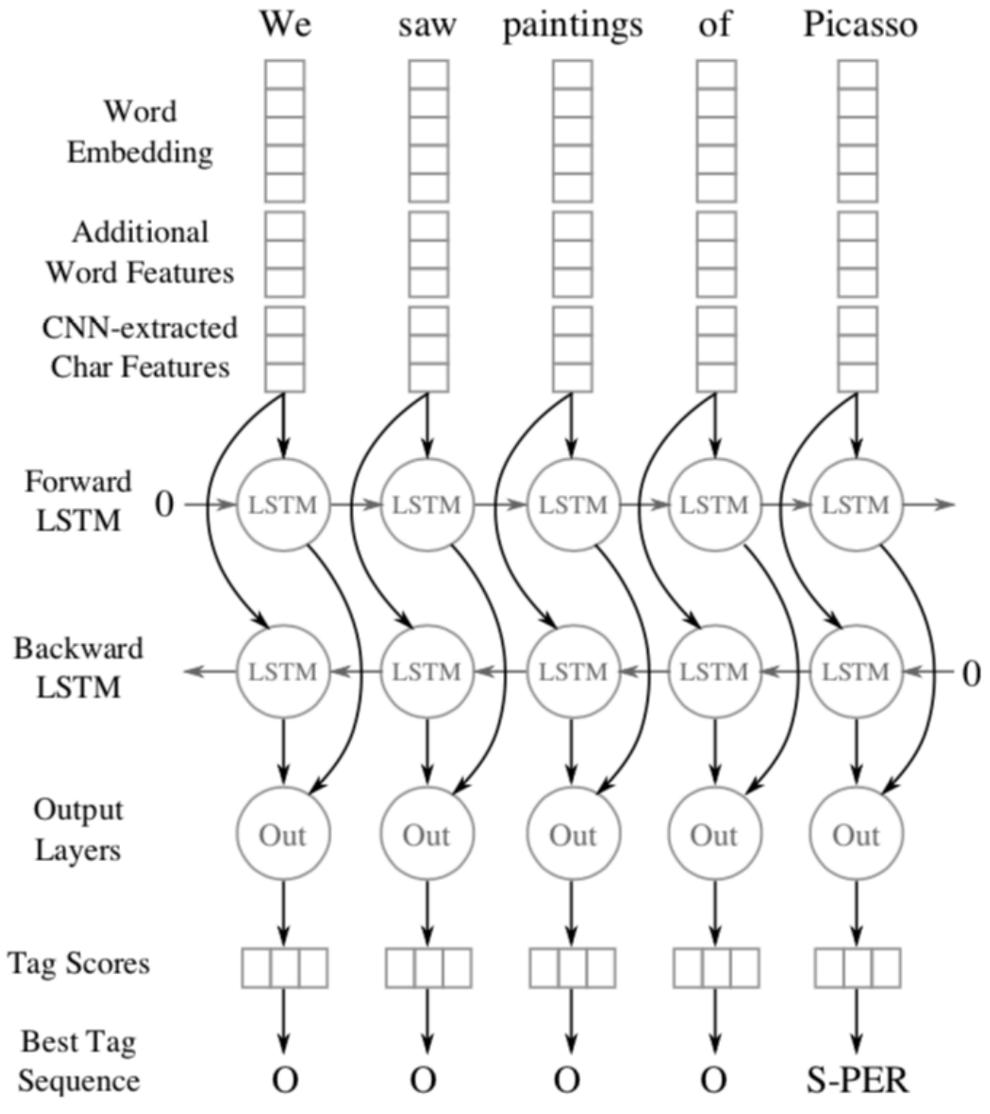
Backward pass (backprop) calculates gradients

Gradients used for parameter updates

DeepMind

UNIVERSITY OF CAMBRIDGE

A State-of-the-Art System



Taken from Chiu and Nichols (2016)

DeepMind

UNIVERSITY OF CAMBRIDGE

Some State-of-the-Art Numbers

Model	CoNLL-2003		
	Prec.	**Recall**	**F1**
FFNN + emb + caps + lex	89.54	89.80	89.67 (± 0.24)
BLSTM	80.14	72.81	76.29 (± 0.29)
BLSTM-CNN	83.48	83.28	83.38 (± 0.20)
BLSTM-CNN + emb	90.75	91.08	90.91 (± 0.20)
BLSTM-CNN + emb + lex	91.39	**91.85**	**91.62** (± 0.33)
Collobert et al. (2011b)	-	-	88.67
Collobert et al. (2011b) + lexicon	-	-	89.59
Huang et al. (2015)	-	-	90.10
Ratinov and Roth (2009)[18]	91.20	90.50	90.80
Lin and Wu (2009)	-	-	90.90
Finkel and Manning (2009)[19]	-	-	-
Suzuki et al. (2011)	-	-	91.02
Passos et al. (2014)[20]	-	-	90.90
Durrett and Klein (2014)	-	-	-
Luo et al. (2015)[21]	**91.50**	91.40	91.20

Taken from Chiu and Nichols (2016)

DeepMind

UNIVERSITY OF CAMBRIDGE

References

- Language Independent NER using a Maximum Entropy Tagger, James R. Curran and Stephen Clark (2003)

- Natural Language Processing (Almost) from Scratch, Collobert, Weston, et al. (2011)

- Named Entity Recognition with Bidirectional LSTM-CNNs, Chiu and Nichols (2016)

DeepMind

UNIVERSITY OF CAMBRIDGE