Artificial Neural Networks
II

Ronan Collobert

ronan@collobert.com

McCulloch and Pitts
- Boolean functions
- No training

Perceptron
- Linear classification
- Convergence if separable
- Generalization?

Adaline
- Linear classification/regression
- Delta Rule
- Convergence?

Margin Perceptron
- Linear classification
- Margin: better generalization?

Kernel Perceptron
- Non-linear classification

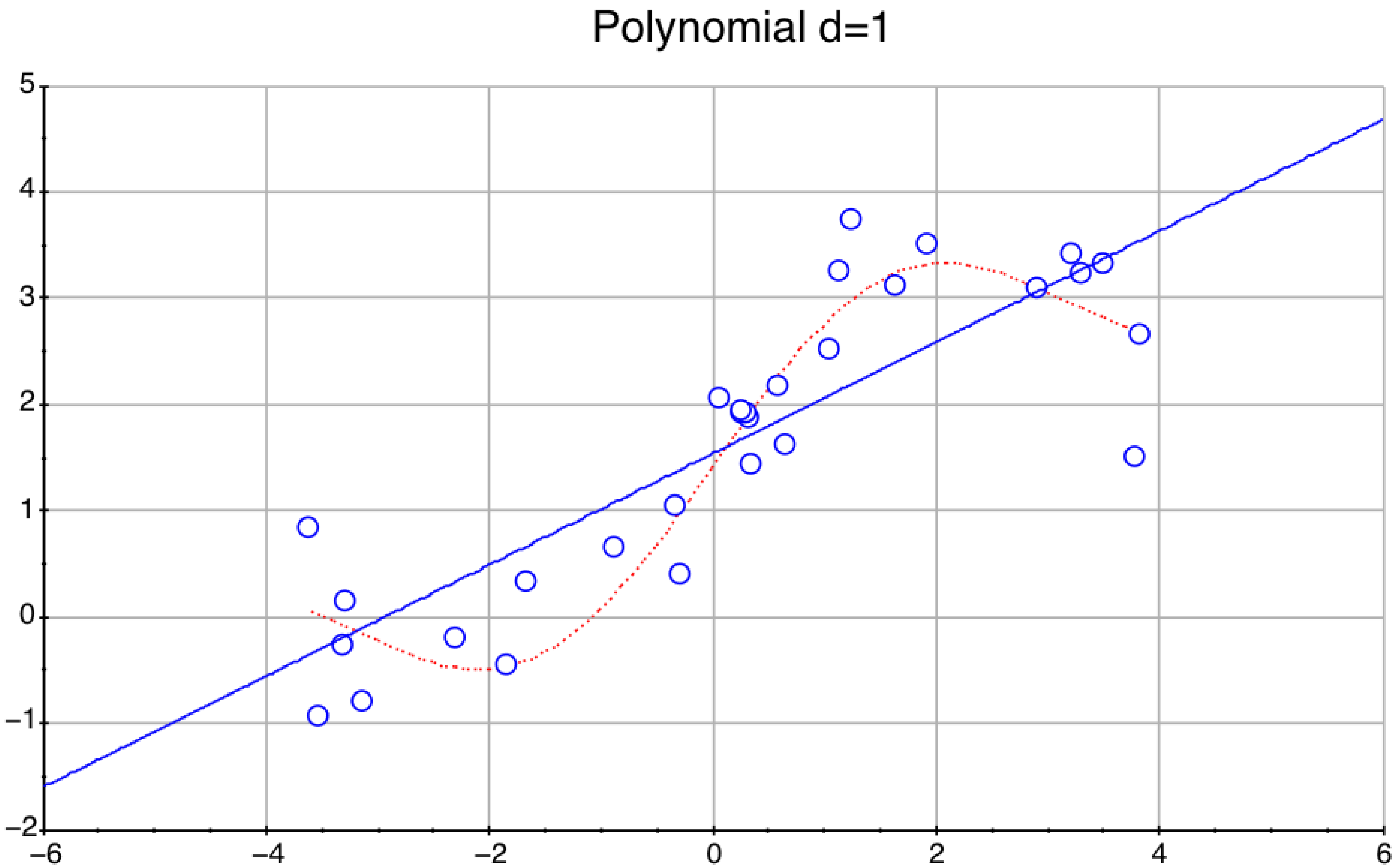
SVM
- Linear classification
- Non-linear with kernels
- Margin: better generalization?

Multi Layer Perceptron
- Non-linear classification/regression
- Gradient descent (backprop)
- Convergence?
- Generalization?

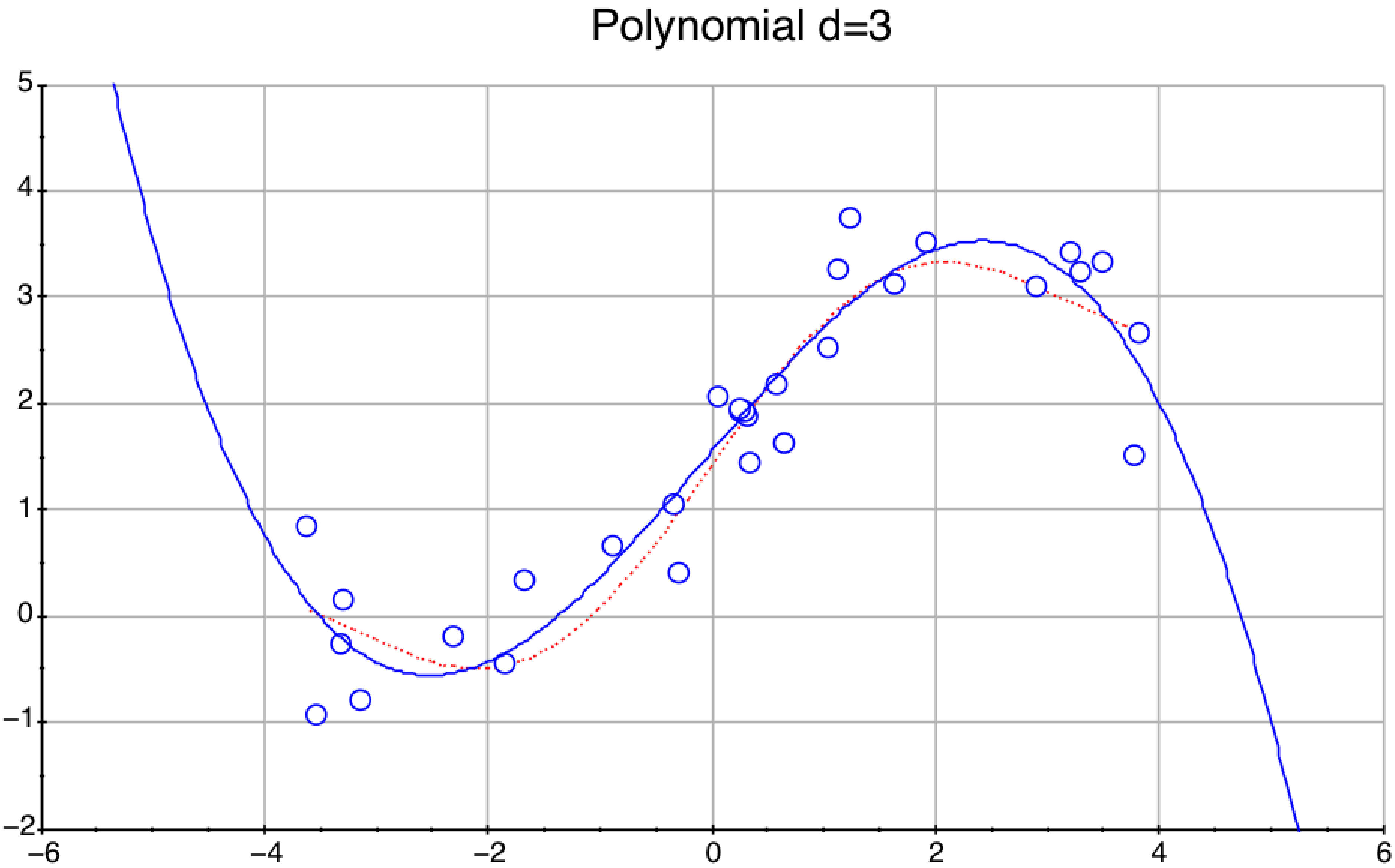
Unsupervised Training
- Reconstruction bottleneck:
  - layer size
  - sparsity
  - transpose constraint

Specializations
- RBF
- Convolutions 1D/2D
- Sequence classification

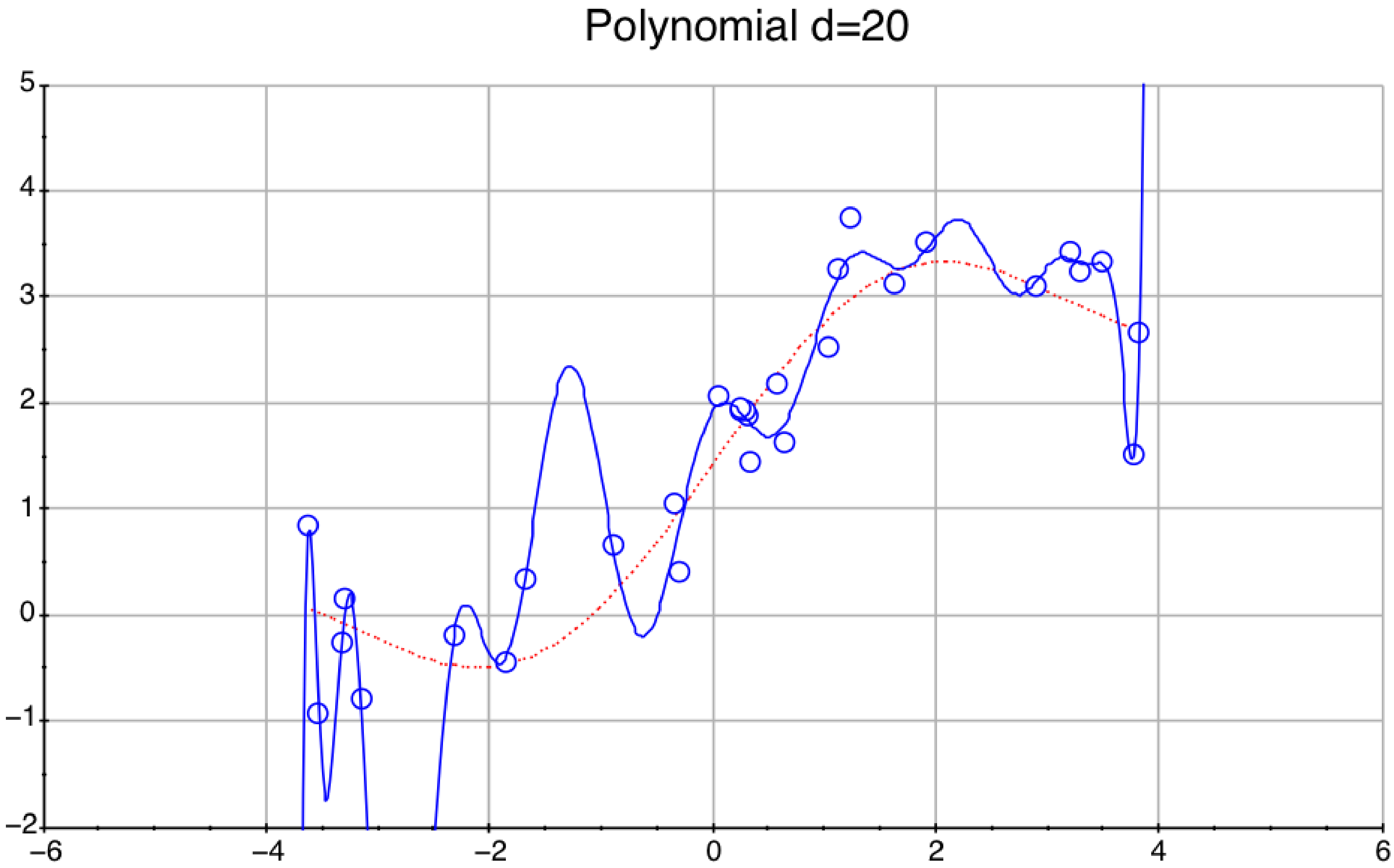
Generalization



From (Bottou, 2010)



From (Bottou, 2010)

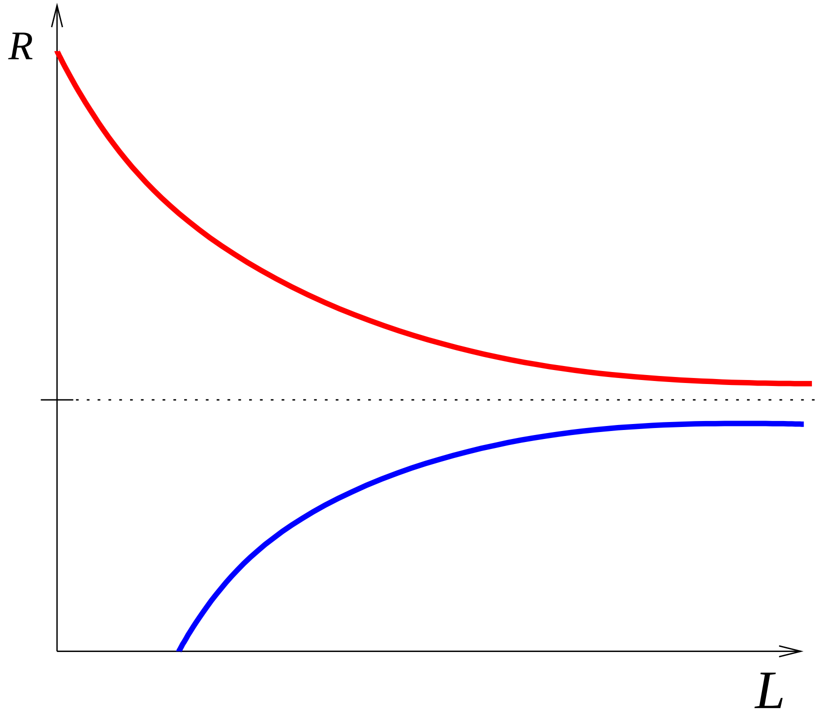


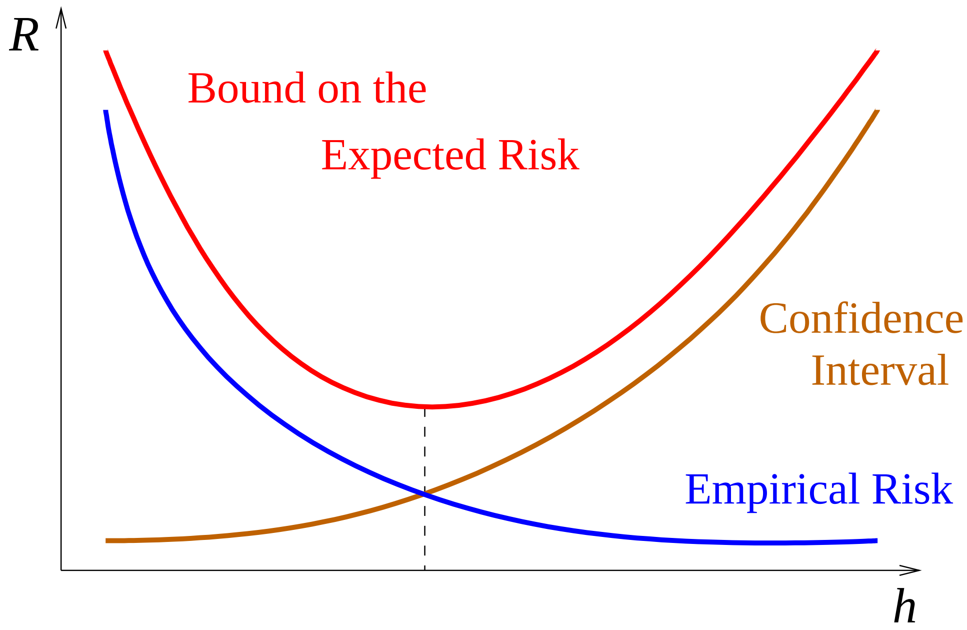
From (Bottou, 2010)

- Bound the difference train-test error
  given "complexity" measure of class of functions

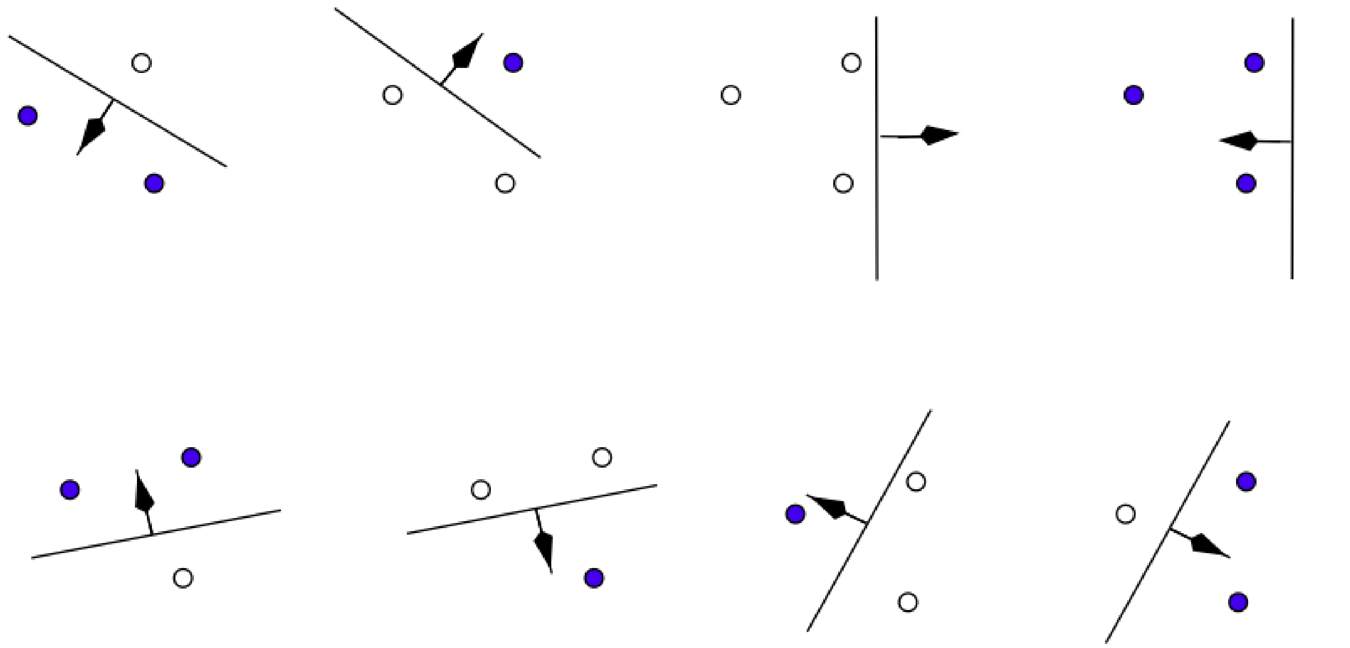
- h is the Vapnik-Chervonenkis dimension
- L training examples
- With probability $1-\eta$:

$$\text{testerr} \leq \text{trainerr} + \sqrt{\frac{h(\log(2L/h)+1) - \log(\eta/4)}{L}} \qquad (1974)$$





- VC dim of a set of functions: maximum number of points L that can be separated into two different classes in all the 2^L ways



From (Burges, 1998)

- VC dim { linear classifiers $x \mapsto w \cdot x$, dim d }: $h = d + 1$
- VC dim { linear classifiers with margin $\geq \rho$, dim d }: $h \leq \min(\frac{R^2}{\rho^2}, d) + 1$
- VC dim { neural net classifiers with n parameters }: $h \sim O(n^4)$ (Karpinski & Macintyre, 1997)

Gradient Descent Convergence

- Proofs from (Bottou, 1991)
- Given a cost function $C(w)$, we perform

$$w^{t+1} = w^t - \lambda^t \frac{\partial C(w^t)}{\partial w}$$

- Assume we have a single minimum $w^\star$ and

$$\forall \epsilon \quad \inf_{||w-w^\star||^2 > \epsilon} (w - w^\star) \frac{\partial C(w)}{\partial w} > 0$$

- Define sequence

$$h^t = (w^t - w^\star)^2$$

- Idea: if $u_t \geq 0$ and $\sum_t (u_{t+1} - u_t)_+ < \infty$ then u_t converges
- Consider

$$h^{t+1} - h^t = -2\,\lambda^t (w^t - w^\star) \frac{\partial C(w^t)}{\partial w} + \left(\lambda^t \frac{\partial C(w^t)}{\partial w} \right)^2$$

- Consider

$$h^{t+1} - h^t = -2\lambda^t (w^t - w^\star) \frac{\partial C(w^t)}{\partial w} + \left(\lambda^t \frac{\partial C(w^t)}{\partial w}\right)^2$$

- Assume

$$\left(\frac{\partial C(w)}{\partial w}\right)^2 \leq A + B\,(w - w^\star)^2 \quad (A, B \geq 0)$$

- Then we get:

$$h^{t+1} - h^t \leq A\,(\lambda^t)^2 + B(\lambda^t)^2\, h^t \quad \Rightarrow \quad h^{t+1} - (1 + B(\lambda^t)^2)\, h^t \leq A\,(\lambda^t)^2$$

- Assume

$$\sum_t (\lambda^t)^2 < \infty$$

- The following sequence converges:

$$\mu^t = \prod_{i=1}^{t} \frac{1}{1 + B(\lambda^i)^2}$$

- We have $\mu^t\, h^{t+1} - \mu^{t-1}\, h^t \leq A\,(\lambda^t)^2\, \mu^t$
  - ⋆ So $\sum_t A\,(\lambda^t)^2\, \mu^t < \infty$
  - ⋆ $\Rightarrow \mu^{t-1}\, h^t$ converges
  - ⋆ $\Rightarrow h^t$ converges

- We have

$$h^{t+1} - h^t = -2\lambda^t (w^t - w^\star) \frac{\partial C(w^t)}{\partial w} + \left(\lambda^t \frac{\partial C(w^t)}{\partial w} \right)^2$$

- h^t converges and $\sum_t (\lambda^t)^2 < \infty$, so with previous assumption

$$\sum_t \lambda^t (w^t - w^\star) \frac{\partial C(w^t)}{\partial w} < \infty$$

- Make sure learning rates do not decrease too quickly:

$$\sum_t \lambda^t = \infty$$

- In that case $(w^t - w^\star) \frac{\partial C(w^t)}{\partial w}$ converges to 0, and because of initial assumption

$$w^t \to w^\star$$

- Given a cost function $C(w)$, we perform

$$w^{t+1} = w^t - \lambda^t H(z^t, w^t)$$

such that

$$\mathbf{E}_z H(z,\, w^t) = \frac{\partial C(w^t)}{\partial w}$$

- Same idea than before, with same kind of hypothesis, but this time

$$h^t = (w^t - w^\star)^2$$

is a random variable.

- Use the same kind of "trick":
if $u_t \geq 0$ and $\sum_t \mathbf{E}(\delta_t(u_{t+1} - u_t)) < \infty$ then u_t converges a.s.
with

$$\delta_t = \begin{cases} 1 \text{ if } \mathbf{E}(u^{t+1} - u^t | \mathcal{P}^t) > 0 \\ 0 \text{ otherwise} \end{cases}$$

where $\mathcal{P}^t$ is the "history" up to time t

$$\mathcal{P}^t = z^0, \ldots, z^{t-1}, w^0, \ldots, w^t, \lambda^0, \ldots, \lambda^t$$

- More general convergence theorems exist (Bottou, 1991)
  - ⋆ Assume $C(w)$ is three time differentiable
  - ⋆ If several minima, then we can show w^t stay "confined" in the same region when λ^t decreases.
  - ⋆ Assume $C \geq C_{min}$ and consider $h^t = C(w^t) - C_{min}$
- Assumptions similar than before:

$$\sum_t \lambda^t = \infty \quad \text{and} \quad \sum_t (\lambda^t)^2 < \infty$$

and

$$\mathbf{E}_z (H(z, w))^2 \leq A + B\, w^2 \quad \text{with } A, B \geq 0$$

- Then we get

$$C(w^t) \to C^\infty \text{ a.s.} \quad \text{and} \quad \left(\frac{\partial C(w^t)}{\partial w}\right)^2 \to 0 \text{ a.s.}$$

Applications

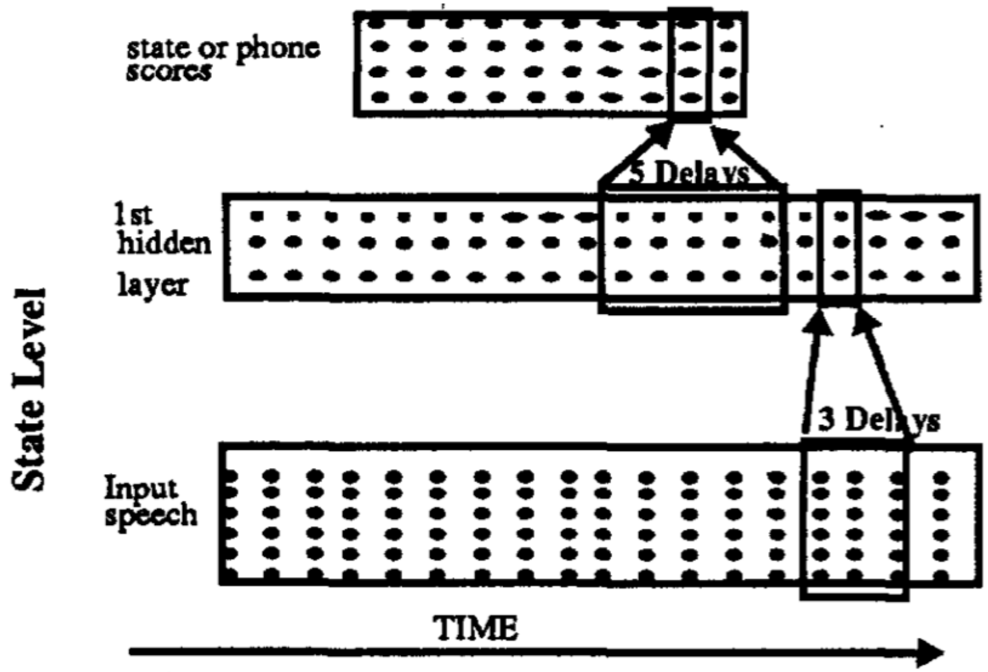


Fig. 1. baseline TDNN

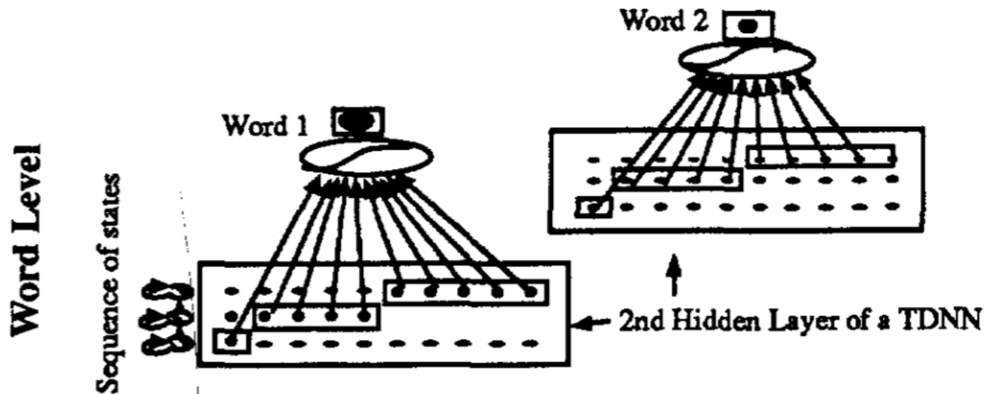


Fig. 3. 2 Word MS-TDNN

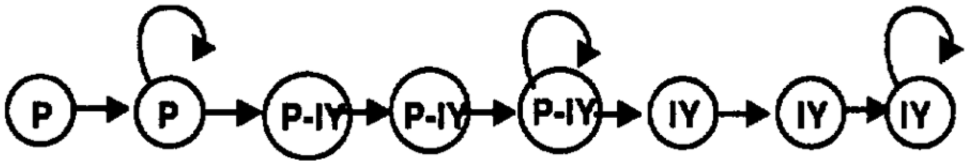
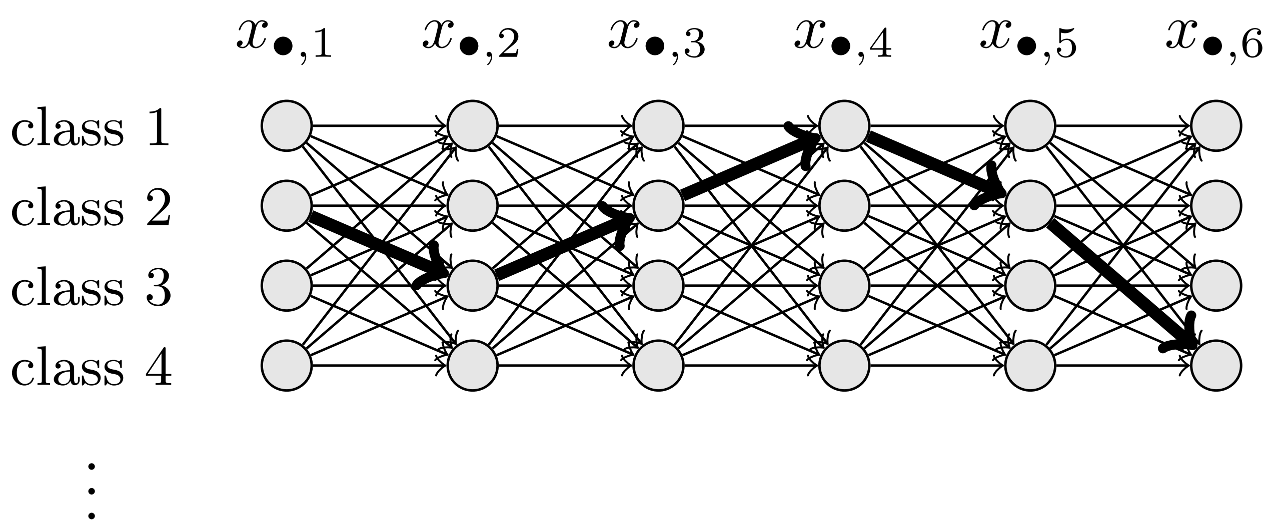


Fig. 4. Phone Model for 'p'

From (Haffner, 1992)

- Sequence of T frames $[\boldsymbol{x}]_1^T$
- The network score for class k at the t^{th} frame is $f([\boldsymbol{x}]_1^T, k, t, \boldsymbol{\theta})$
- A_{kl} transition score to jump from class k to class l



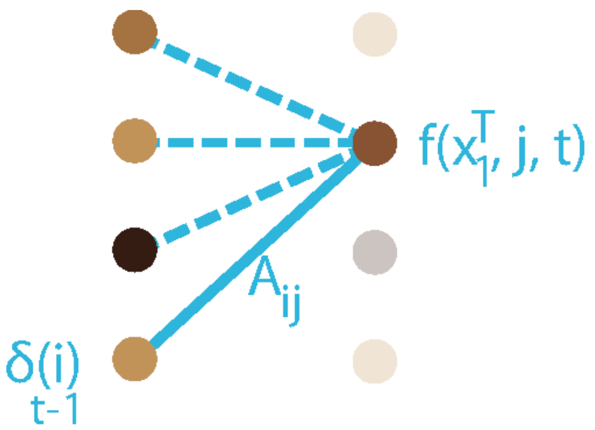
- Sentence score for a class label path $[i]_1^T$

$$s([\boldsymbol{x}]_1^T, [i]_1^T, \tilde{\boldsymbol{\theta}}) = \sum_{t=1}^{T} \left(A_{[i]_{t-1}[i]_t} + f([\boldsymbol{x}]_1^T, [i]_t, t, \boldsymbol{\theta}) \right)$$

- Conditional likelihood by normalizing w.r.t all possible paths:

$$\log p([y]_1^T \,|\, [\boldsymbol{x}]_1^T, \tilde{\boldsymbol{\theta}}) = s([\boldsymbol{x}]_1^T, [y]_1^T, \tilde{\boldsymbol{\theta}}) - \underset{\forall [j]_1^T}{\operatorname{logadd}}\, s([\boldsymbol{x}]_1^T, [j]_1^T, \tilde{\boldsymbol{\theta}})$$

- Normalization computed with recursive Forward algorithm:



$$\delta_t(j) = \text{logAdd}_i \left[\delta_{t-1}(i) + A_{i,j} + f_\theta(j, x_1^T, t)\right]$$

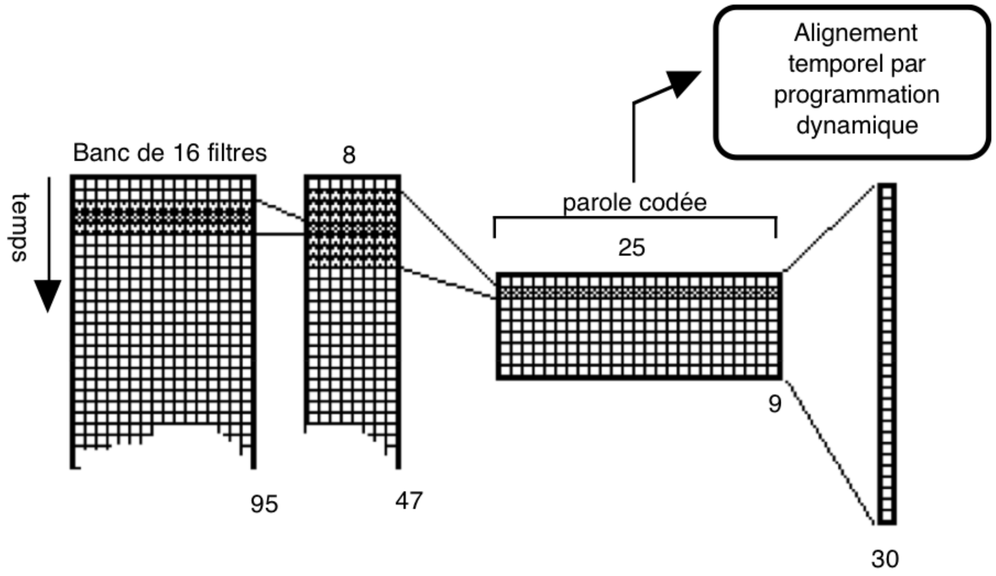
Termination:

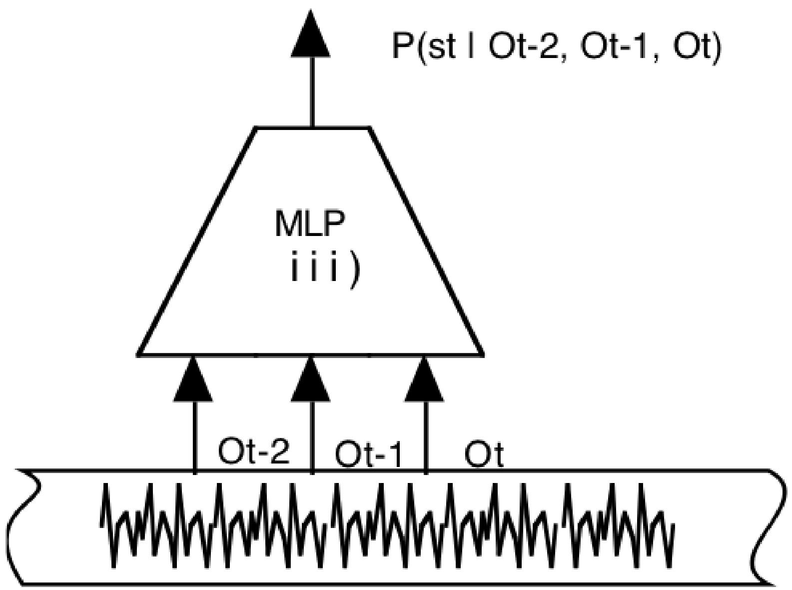
$$\underset{\forall [j]_1^T}{\text{logadd}}\, s([\boldsymbol{x}]_1^T, [j]_1^T, \tilde{\boldsymbol{\theta}}) = \text{logAdd}_i\, \delta_T(i)$$

- Simply backpropagate through this recursion with chain rule

- Non-linear CRFs: Graph Transformer Networks (Bottou et al., 1997)
- Compared to CRFs, we train features (network parameters $\boldsymbol{\theta}$ and transitions scores A_{kl})

- Inference: Viterbi algorithm (replace logAdd by max)





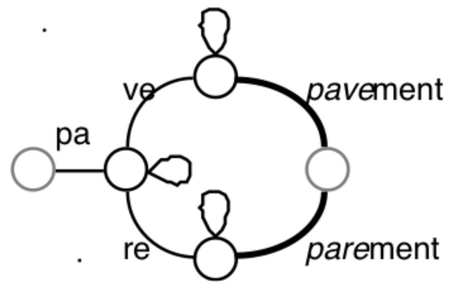


Fig 10.2 - Mise en commun des parties identiques. Aux transitions en gras sont associés les mots "pavement" et "parement".

From (Bottou, 1991)

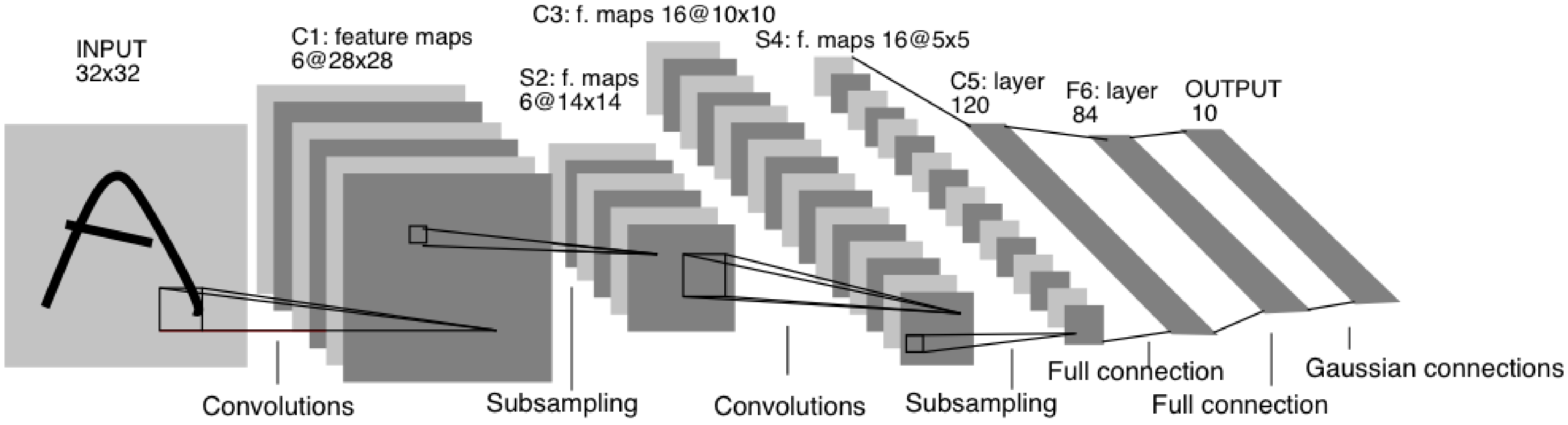
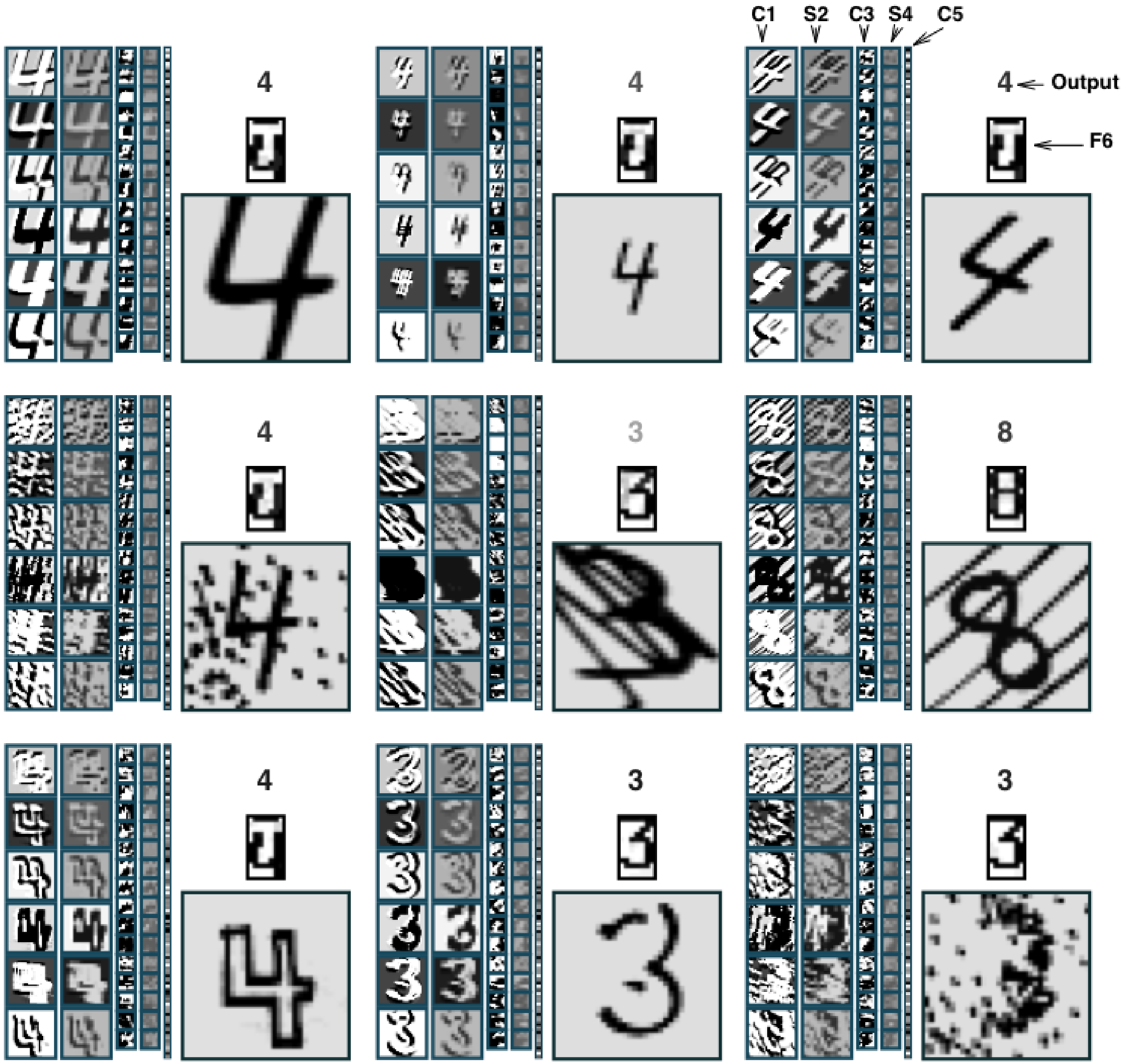


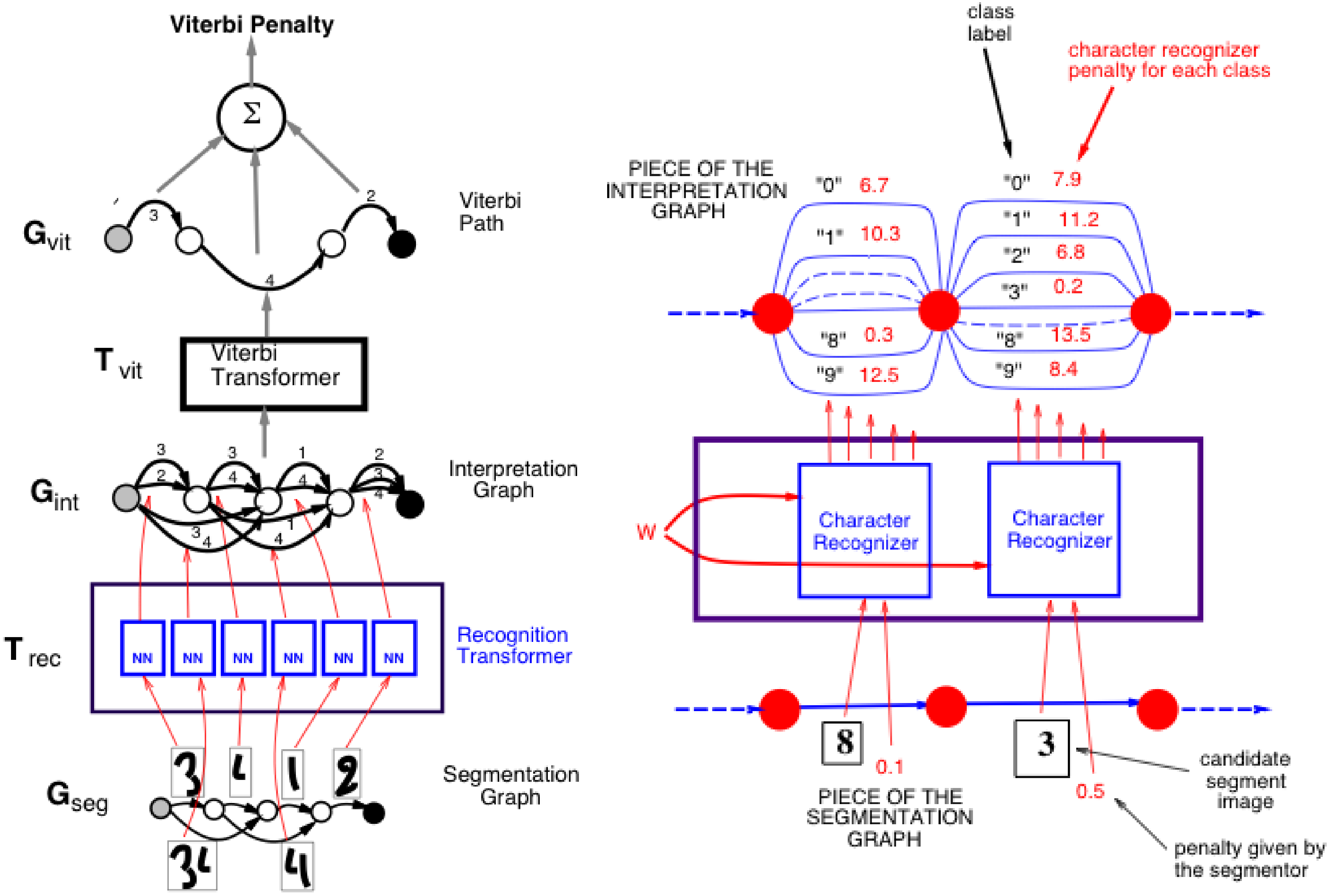
Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Fig. 4. Size-normalized examples from the MNIST database.

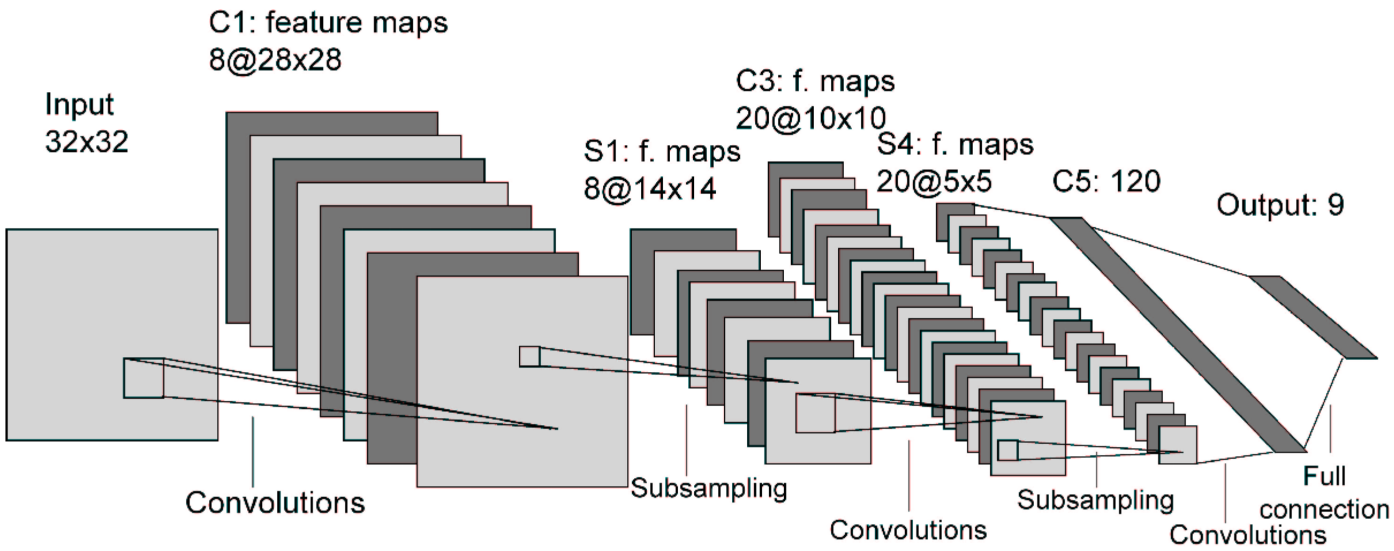
	Err. rate (%)
Gaussian SVM	1.4
1000 HU NN (MSE)	4.5
800 HU NN	1.6
CNN	0.8
CNN + distortions	0.4
6 layers NN + distortions	0.4

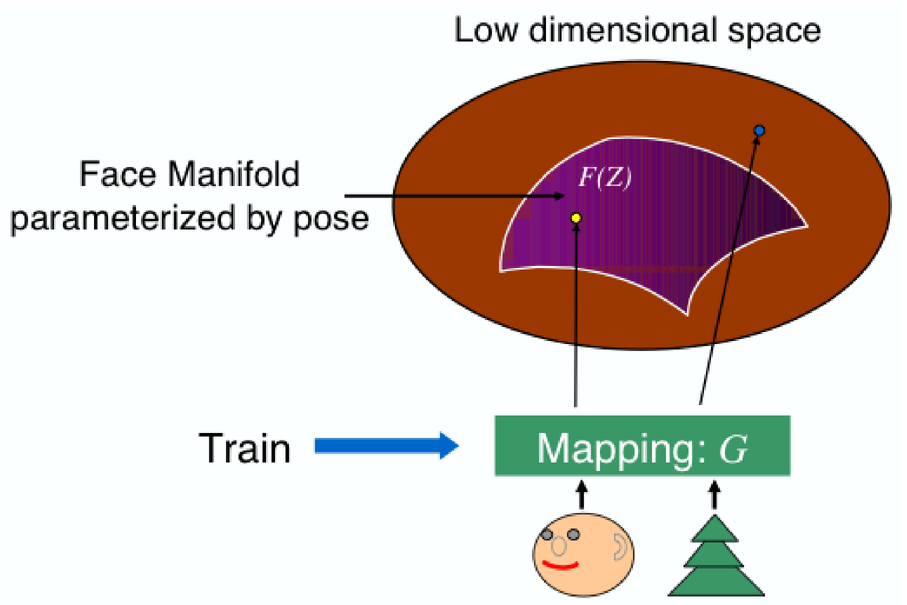


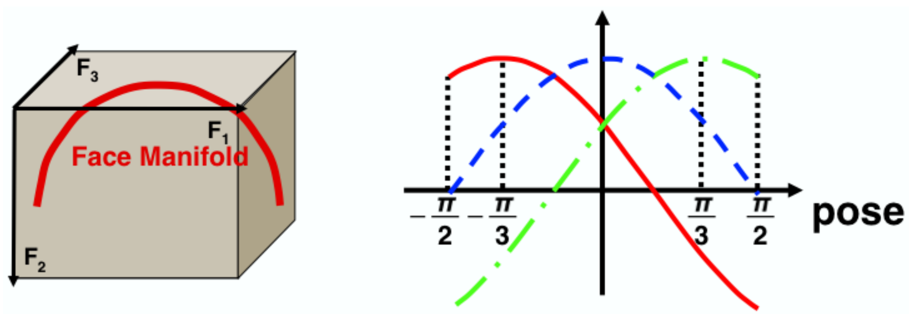
(Lecun et al., 1998)



(Lecun et al., 1998)

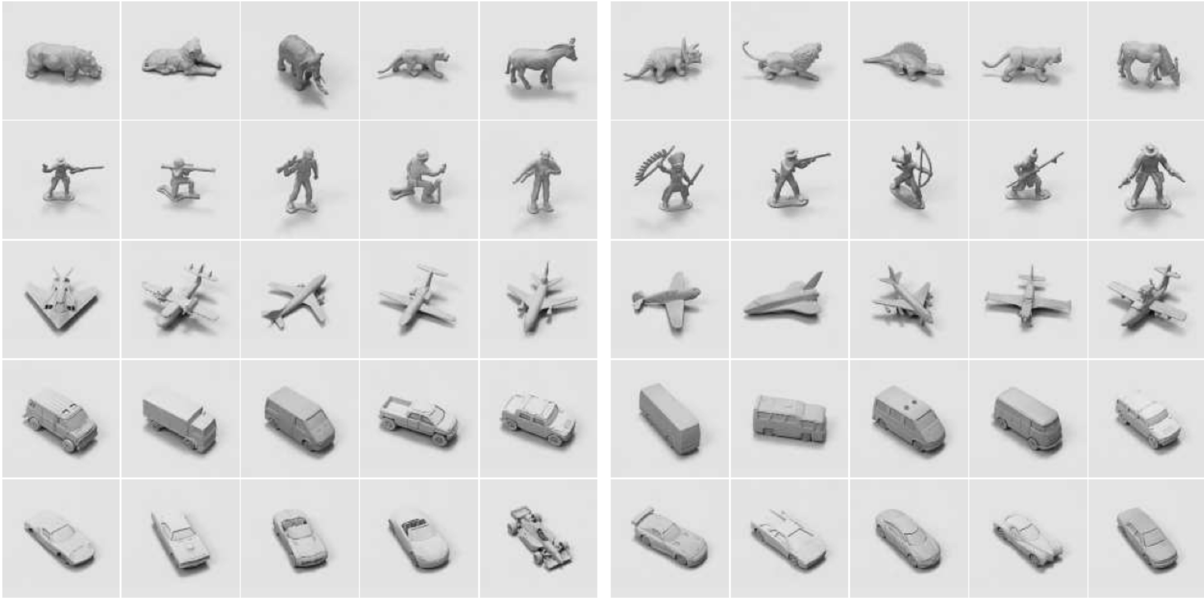






(Osadchy et al., 2007)

Image: Object Recognition



Classifi cation				
exp#	Classifi er	Input	Dataset	Test Error
1.0	Linear	raw 2x96x96	norm-unif	30.2%
1.1	K-NN (K=1)	raw 2x96x96	norm-unif	18.4 %
1.2	K-NN (K=1)	PCA 95	norm-unif	16.6%
1.3	SVM Gauss	raw 2x96x96	norm-unif	N.C.
1.4	SVM Gauss	raw 1x48x48	norm-unif	13.9%
1.5	SVM Gauss	raw 1x32x32	norm-unif	12.6%
1.6	SVM Gauss	PCA 95	norm-unif	13.3%
1.7	Conv Net 80	raw 2x96x96	norm-unif	6.6%
1.8	Conv Net 100	raw 2x96x96	norm-unif	6.8%
2.0	Linear	raw 2x96x96	jitt-unif	30.6%
2.1	Conv Net 100	raw 2x96x96	jitt-unif	7.1%
Detection/Segmentation/Recognition				
exp#	Classifi er	Input	Dataset	Test Error
5.1	Conv Net 100	raw 2x96x96	jitt-text	10.6%
6.0	Conv Net 100	raw 2x96x96	jitt-clutt	16.7%
6.2	Conv Net 100	raw 1x96x96	jitt-clutt	39.9%

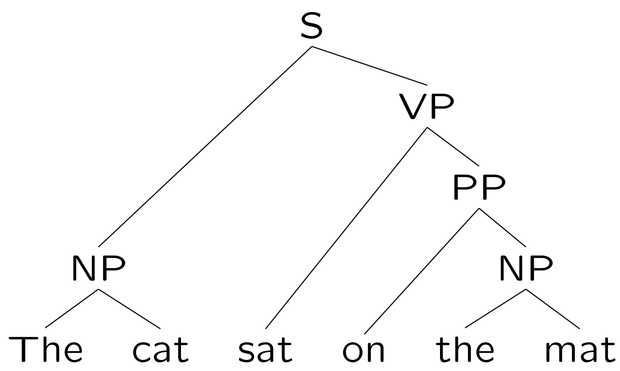
(LeCun et al., 2004)

Text: Natural Language Processing (Tasks)

- Part-Of-Speech Tagging (POS): syntactic roles (noun, adverb...)
- Chunking (CHK): syntactic constituents (noun phrase, verb phrase...)
- Name Entity Recognition (NER): person/company/location...
- Semantic Role Labeling (SRL): semantic role

  [John]$_{ARG0}$ [ate]$_{REL}$ [the apple]$_{ARG1}$ [in the garden]$_{ARGM-LOC}$

- Parsing (PSG):



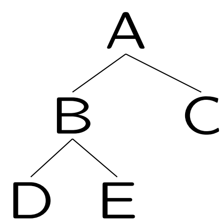
- Tagging tasks (BIOES tagging scheme):

The	black	cat	sat	on	the	mat	.
B-NP	I-NP	E-NP	S-VP	S-PP	B-NP	E-NP	O

POS (Toutanova, 2003)	Various combinations of surrounding words & tags, various caps, digit, dash, various prefixes & suffixes Dependency Network
Chunking (Sha, 2003)	surrounding words, POS tags Conditional Random Field (CRF)
NER (Ando, 2005)	Surrounding words, POS, several suffixes & prefixes, surrounding tags, bigrams, previously assigned tags to words, unlabeled data Viterbi decoding at test
SRL (Koomen, 2005)	6 parse trees, pruning heuristics, POS, voice, phrase type, head words, subparts of the trees, ... Argument identification, argument classification, integer linear programming

PCFG

$A \longrightarrow B\ C$



Parsing (Collins, 1999) (Charniak, 2000)	Lexicalized Probabilistic Context-Free Grammar (PCFG), POS, head words, chart parser, deleted interpolation, ... 30 pages of details in (Bikel, 2004)
Parsing (Charniak & Johnson, 2005 & 2006)	Re-ranking over the above, using lots of ad-hoc features
Parsing (Finkel et al, 2008) (Petrov & Klein, 2008) (Carreras & al, 2008	PCFG, dependency features CRF or similar

$$\begin{aligned} \text{a word} &= \text{index in a dictionary} \\ \text{The cat sat on the mat} &= (w_1, w_2, w_3, w_4, w_5, w_6) \end{aligned}$$

binary code $\sim$ dictionary size

$$w \longleftrightarrow \left(0, \cdots 0, \underset{\text{at index } w}{1}, 0, \cdots 0\right)^{\mathrm{T}} = (\mathbf{1}_{\cdot = w})^{\mathrm{T}}$$

word embedding

$M \sim$ feature size $\times$ dictionary size

$$M \times (\mathbf{1}_{\cdot = w}) = M_{\bullet w}$$

lookup-table operation

sentence embedding

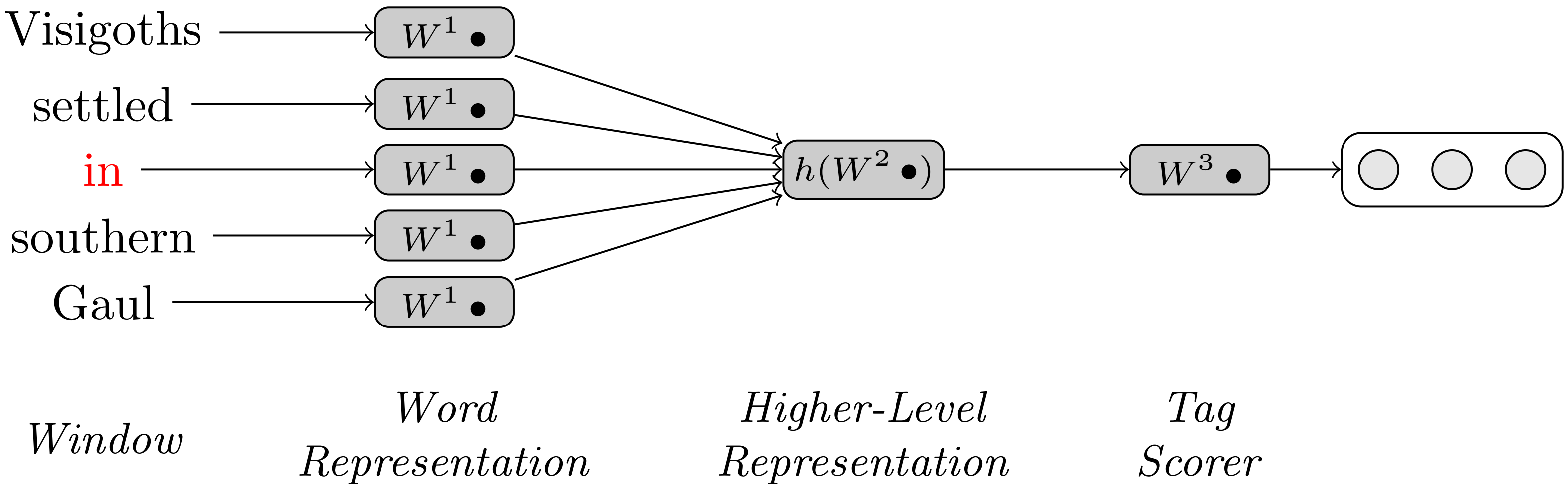
$$M \times (\mathbf{1}_{\cdot = w_1} \cdots \mathbf{1}_{\cdot = w_6}) = (M_{\bullet w_1} \cdots M_{\bullet w_6})$$

Convolution (kernel size 1)

Applicable to any discrete feature (words, caps, stems...)

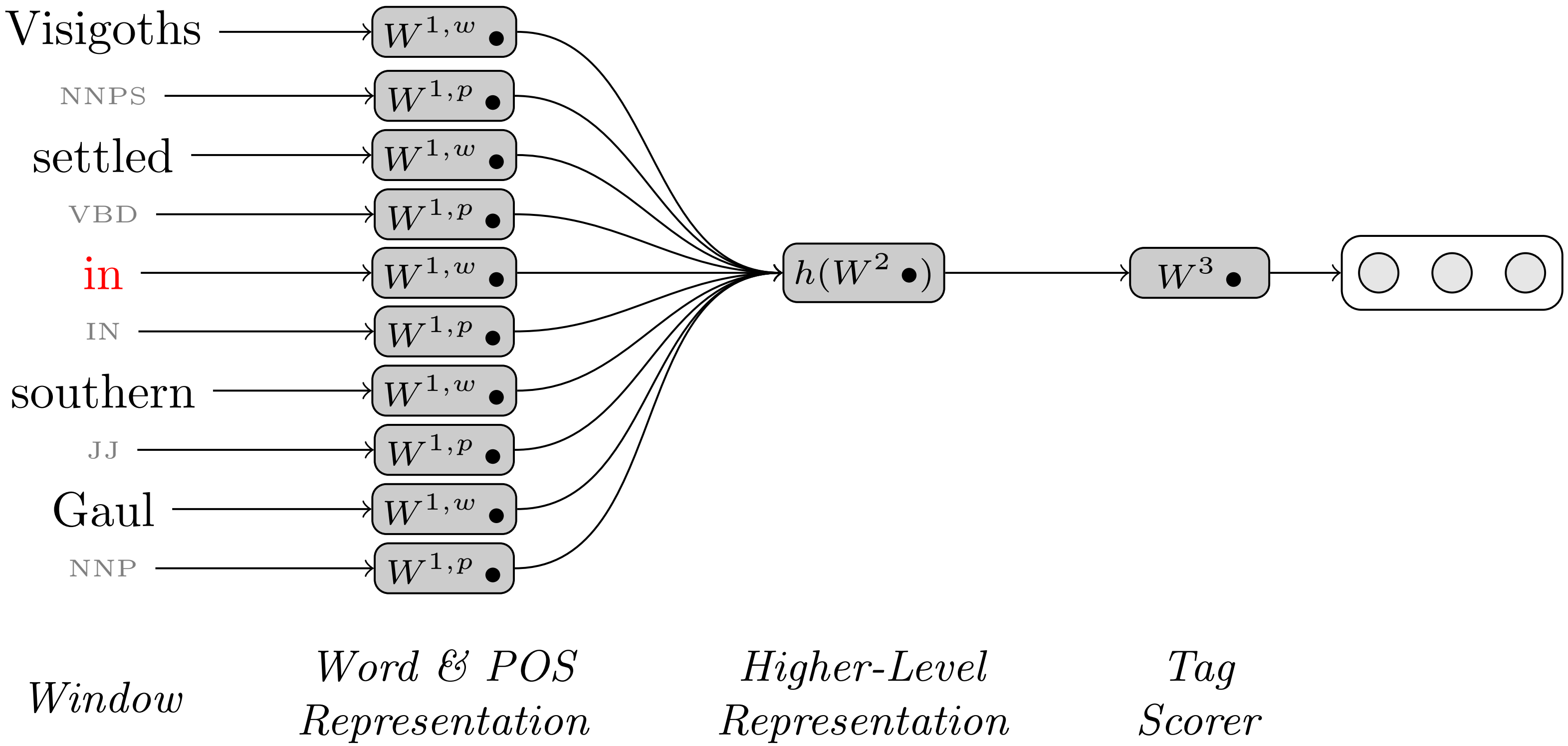
See (Bengio et al, 2001)

How to tag "in" in the sentence
"The Visigoths settled in southern Gaul"?

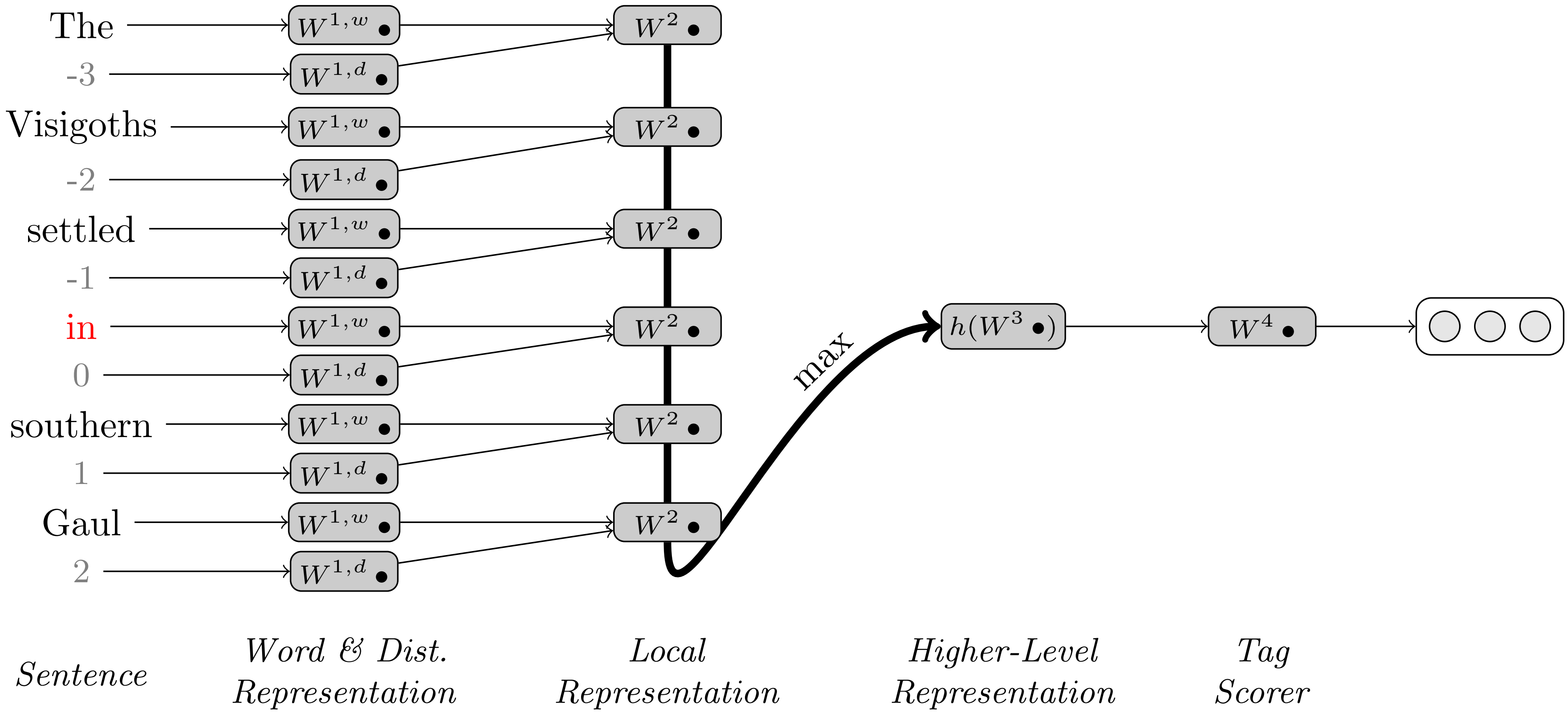


Window Approach (extra features)

How to tag "in" in the sentence
"The Visigoths settled in southern Gaul"?

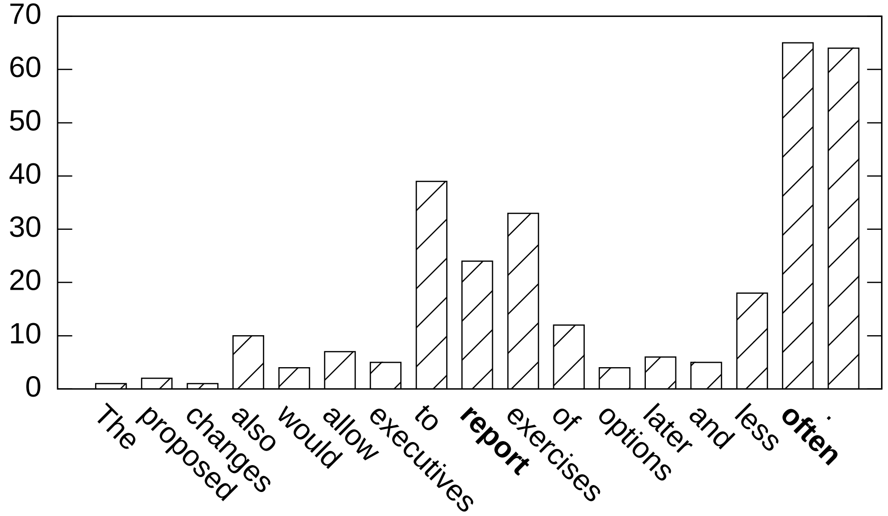


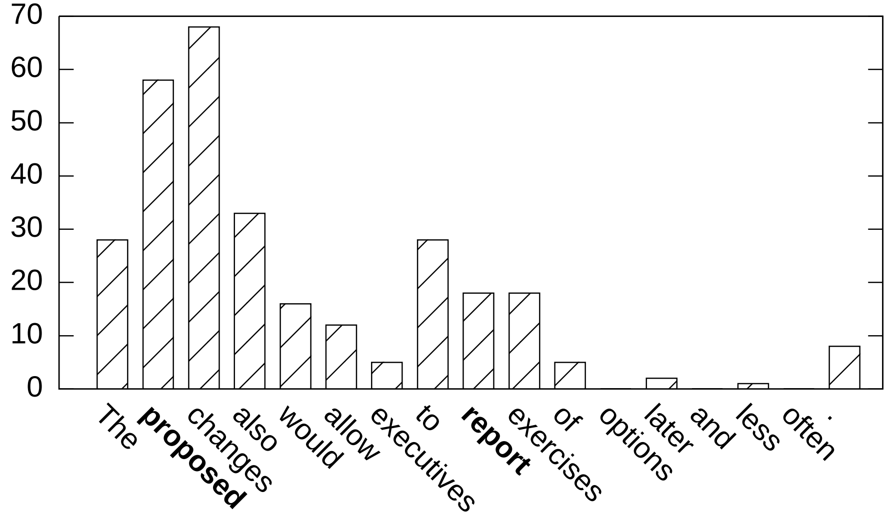
How to tag "in" in the sentence
"The Visigoths settled in southern Gaul"?



For each i, what is the chosen t ?

$$\max_t [X]_{i,t} \quad \forall i$$





Ranking Language Model

- Language Model: *"is a sentence actually english or not?"*
  Implicitly captures: ⋆ syntax ⋆ semantics
- Bengio & Ducharme (2001) Probability of next word given previous words. Overcomplicated – we do not need probabilities here
- Entropy criterion largely determined by most frequent phrases
- Rare legal phrases are no less significant that common phrases
- $f()$ a window approach network
- Ranking margin cost:

$$\sum_{s \in \mathcal{S}} \sum_{w \in \mathcal{D}} \max\left(0,\, 1 - f(s,\, w_s^\star) + f(s,\, w)\right)$$

$\mathcal{S}$: sentence windows $\mathcal{D}$: dictionary
$w_s^\star$: true middle word in s
$f(s,\, w)$: network score for sentence s and middle word w

- Stochastic training:
  - ⋆ positive example: random corpus sentence
  - ⋆ negative example: replace middle word by random word

Training Language Model

- Two window approach (11) networks (100HU) trained on two corpus:
  - ⋆ LM1: Wikipedia: **631M** of words
  - ⋆ LM2: Wikipedia+Reuters RCV1: **631M+221M=852M** of words
- Massive dataset: cannot afford classical training-validation scheme
- Like in biology: breed a couple of network lines
- Breeding decisions according to 1M words validation set
- LM1
  - ⋆ order dictionary words by frequency
  - ⋆ increase dictionary size: 5000, $10,000$, $30,000$, $50,000$, $100,000$
  - ⋆ 4 weeks of training
- LM2
  - ⋆ initialized with LM1, dictionary size is $130,000$
  - ⋆ 30,000 additional most frequent Reuters words
  - ⋆ 3 additional weeks of training

france	jesus	xbox	reddish	scratched	megabits
454	1973	6909	11724	29869	87025
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

Semi-Supervised Benchmark Results

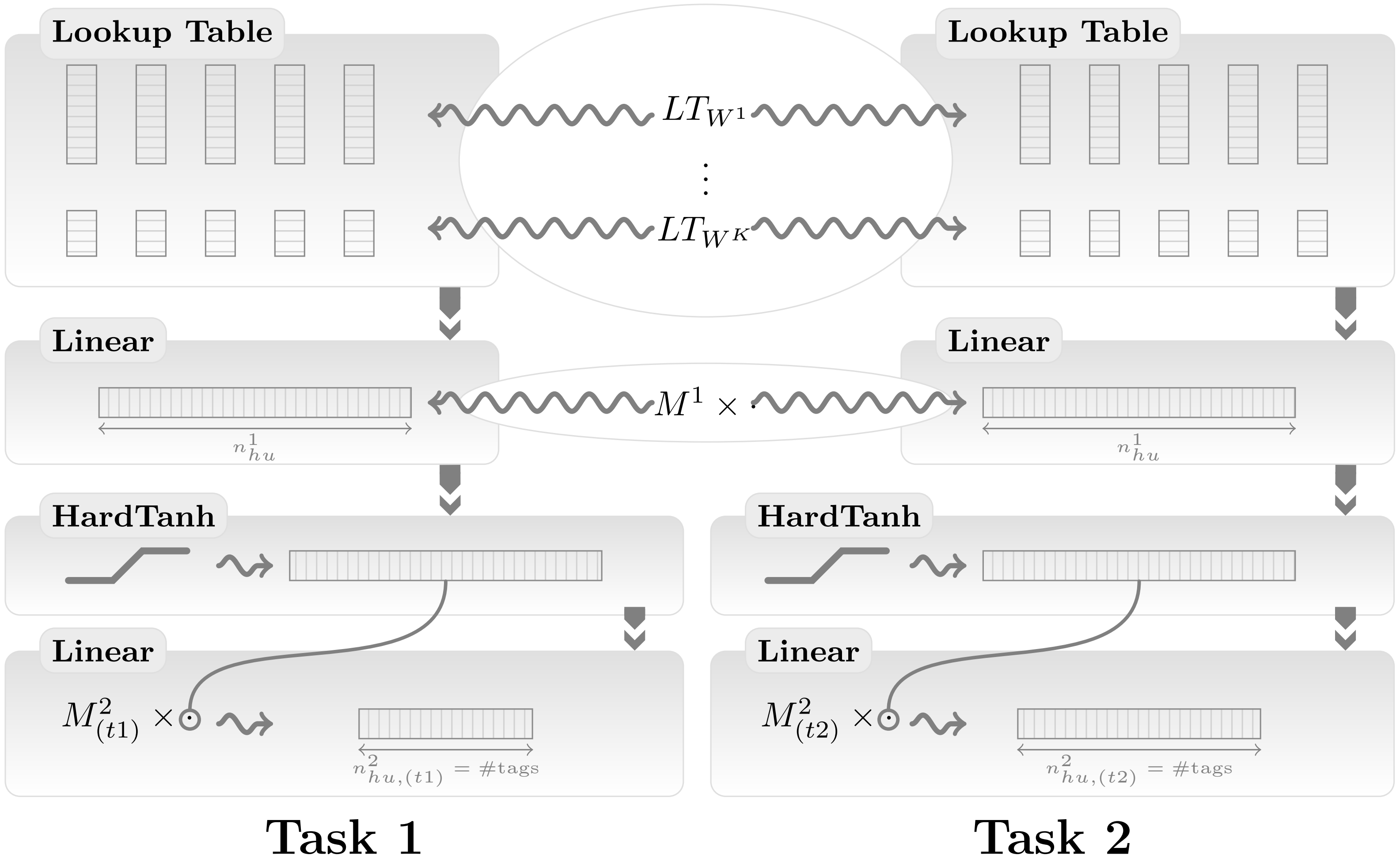
- Initialize word embeddings with LM1 or LM2
- Same training procedure

Approach	POS (PWA)	CHK (F1)	NER (F1)	SRL (F1)
Benchmark Systems	**97.24**	**94.29**	**89.31**	**77.92**
NN+WLL	96.31	89.13	79.53	54.53
NN+SLL	96.37	90.33	81.47	71.24
NN+WLL+LM1	97.05	91.91	85.68	57.32
NN+SLL+LM1	97.10	93.65	87.58	74.28
NN+WLL+LM2	97.14	92.04	86.96	56.97
NN+SLL+LM2	97.20	93.63	88.67	73.90

- Huge boost from language models
- Training set word coverage:

	LM1	LM2
POS	97.86%	98.83%
CHK	97.93%	98.91%
NER	95.50%	98.95%
SRL	97.98%	98.87%

- Joint training
- Good overview in (Caruana, 1997)



Window Approach

Approach	POS (PWA)	CHK (F1)	NER (F1)
Benchmark Systems	**97.24**	**94.29**	**89.31**
NN+SLL+LM2	97.20	93.63	88.67
NN+SLL+LM2+MTL	97.22	94.10	88.62

Sentence Approach

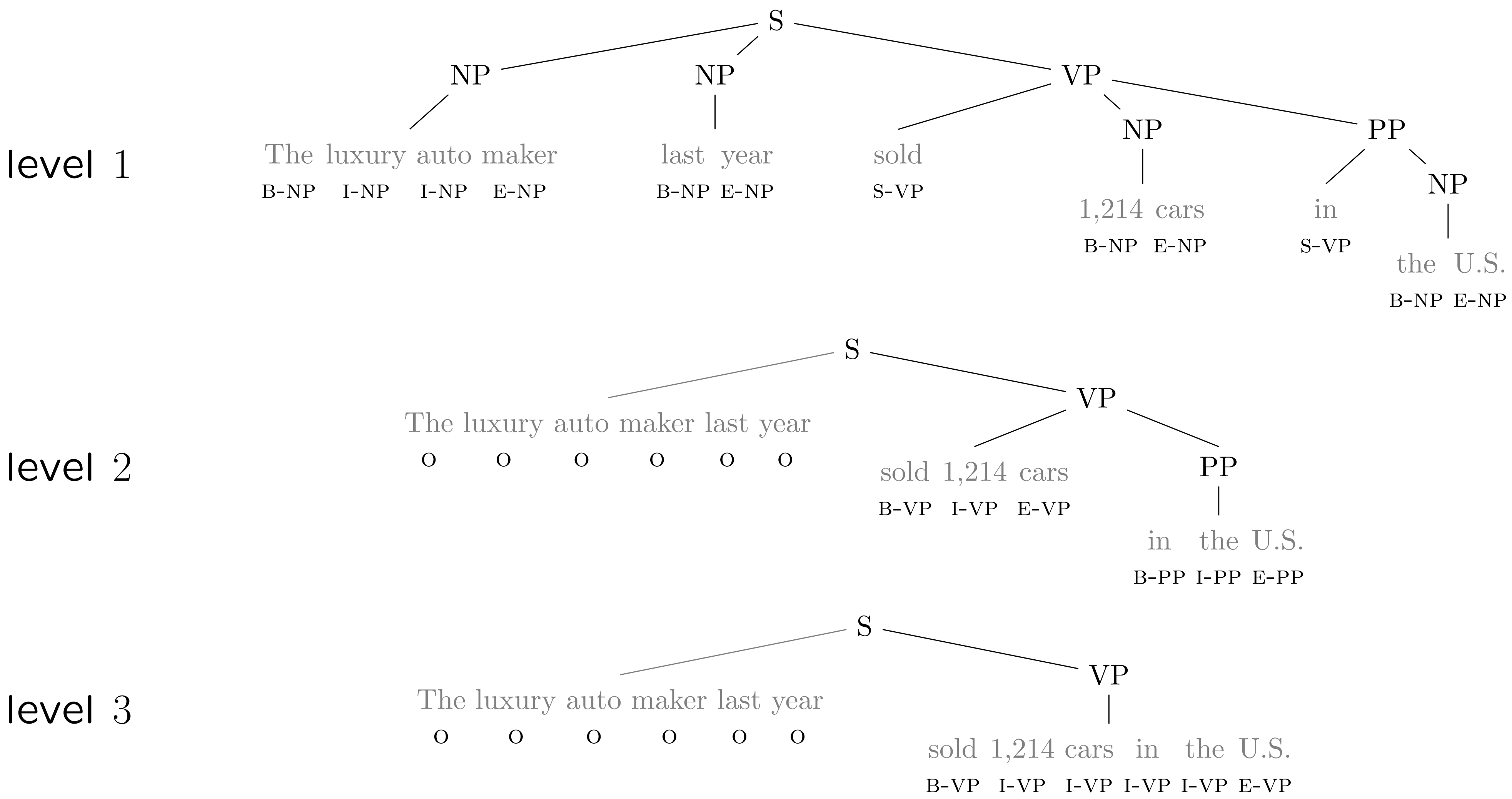
Approach	POS (PWA)	CHK (F1)	NER (F1)	SRL (F1)
Benchmark Systems	**97.24**	**94.29**	**89.31**	**77.92**
NN+SLL+LM2	97.12	93.37	88.78	73.90
NN+SLL+LM2+MTL	97.22	93.72	87.99	74.33

Increase level of engineering by incorporating common NLP techniques

- Stemming for western languages benefits POS (Ratnaparkhi, 1996)
  - ⋆ Use last two characters as feature (455 different stems)
- Gazetteers are often used for NER (Florian, 2003)
  - ⋆ $8,000$ locations, person names, organizations and misc entries from CoNLL 2003
- POS is a good feature for CHK & NER (Shen, 2005) (Florian, 2003)
  - ⋆ We feed our own POS tags as feature
- CHK is also a common feature for SRL (Koomen, 2005)
  - ⋆ We feed our own CHK tags as feature

Approach	**POS** (PWA)	**CHK** (F1)	**NER** (F1)	**SRL** (F1)
Benchmark Systems	**97.24**	**94.29**	**89.31**	**77.92**
NN+SLL+LM2	97.20	93.63	88.67	73.90
NN+SLL+LM2+Suffix2	97.29	–	–	–
NN+SLL+LM2+Gazetteer	–	–	89.59	–
NN+SLL+LM2+POS	–	94.32	88.67	75.39
NN+SLL+LM2+CHK	–	–	–	74.73

- Parsing is essential to SRL (Punyakanok, 2005) (Pradhan, 2005)
- State-of-the-art SRL systems use several parse trees (up to 6!!)
- We feed our network several levels of Charniak parse tree provided by CoNLL 2005



SRL Benchmark Results With Parsing

Approach	SRL (test set F1)
Benchmark System (six parse trees)	**77.92**
Benchmark System (top Charniak only)	**74.76**†
NN+SLL+LM2	73.90
NN+SLL+LM2+CHK	74.73
NN+SLL+LM2+Charniak (level 1 only)	76.27
NN+SLL+LM2+Charniak (levels 1 & 2)	76.24
NN+SLL+LM2+Charniak (levels 1 to 3)	76.62
NN+SLL+LM2+Charniak (levels 1 to 4)	76.50
NN+SLL+LM2+Charniak (levels 1 to 5)	76.98

† on the validation set