Part-of-Speech Tagging

+

Neural Networks

CS 287

Quiz: ReLU

Last class we focused on standard hinge loss. Consider now the squared hinge loss, (ℓ_2 SVM)

$$L_{hinge} = \max\{0, 1 - (\hat{y}_c - \hat{y}_{c'})^2\}$$

What is the effect does this have on the loss? How do the parameters gradients change?

Today's Class

- Part-of-Speech Tagging as a Motivating Example
- Bilinear Models
- Windowed Models

Contents

Penn Treebank (Marcus et al, 1993)

- The ur-dataset of statistical NLP
- Constructed from 1989-1992.
- Contains 4.5 million token
- Around 1 million make up the core PTB, text from 1989 Wall Street Journal

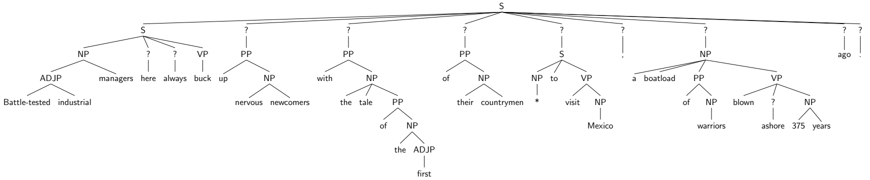
Tagging

So what if Steinbach had struck just seven home runs in 130 regular-season games , and batted in the seventh position of the A 's lineup .

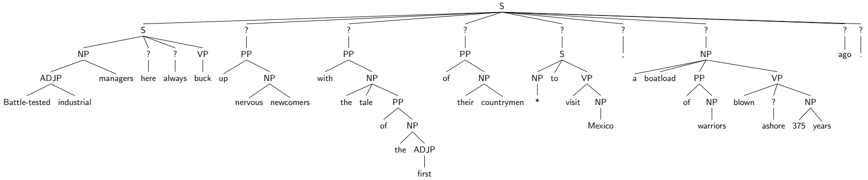
Part-of-Speech Tags

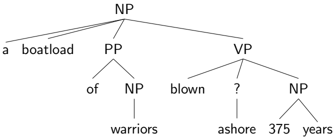
So/RB what/WP if/IN Steinbach/NNP had/VBD struck/VBN just/RB seven/CD home/NN runs/NNS in/IN 130/CD regular-season/JJ games/NNS ,/, and/CC batted/VBD in/IN the/DT seventh/JJ position/NN of/IN the/DT A/NNP 's/NNP lineup/NN ./.

Syntax


S
S ? ? ? ? ? ? ? ?
NP ? ? VP PP PP PP S , NP ago .
ADJP managers here always buck up NP with NP of NP NP to VP a boatload PP VP
Battle-tested industrial nervous newcomers the tale PP their countrymen * visit NP of NP blown ? NP
of NP Mexico warriors ashore 375 years
the ADJP
first

Syntax


S
NP ADJP Battle-tested industrial managers here always buck
up nervous newcomers
with the tale of the first
of their countrymen
* to visit Mexico
a boatload of warriors blown ashore 375 years
ago


NP
a boatload
PP of NP warriors
VP blown ? ashore
NP 375 years

“Simplified” English Tagset I

1. , Punctuation
2. CC Coordinating conjunction
3. CD Cardinal number
4. DT Determiner
5. EX Existential there
6. FW Foreign word
7. IN Preposition or subordinating conjunction
8. JJ Adjective
9. JJR Adjective, comparative
10. JJS Adjective, superlative
11. LS List item marker

“Simplified” English Tagset II

12. MD Modal
13. NN Noun, singular or mass
14. NNS Noun, plural
15. NNP Proper noun, singular
16. NNPS Proper noun, plural
17. PDT Predeterminer
18. POS Possessive ending
19. PRP Personal pronoun
20. PRP$ Possessive pronoun
21. RB Adverb
22. RBR Adverb, comparative

“Simplified” English Tagset III

23. RBS Adverb, superlative
24. RP Particle
25. SYM Symbol
26. TO to
27. UH Interjection
28. VB Verb, base form
29. VBD Verb, past tense
30. VBG Verb, gerund or present participle
31. VBN Verb, past participle
32. VBP Verb, non-3rd person singular present
33. VBZ Verb, 3rd person singular present

“Simplified” English Tagset IV

34. WDT Wh-determiner
35. WP Wh-pronoun
36. WP$ Possessive wh-pronoun
37. WRB Wh-adverb

NN or NNS

Whether a noun is tagged singular or plural depends not on its semantic properties, but on whether it triggers singular or plural agreement on a verb. We illustrate this below for common nouns, but the same criterion also applies to proper nouns.

Any noun that triggers singular agreement on a verb should be tagged as singular, even if it ends in final -s.

EXAMPLE: Linguistics NN is/*are a difficult field.

If a noun is semantically plural or collective, but triggers singular agreement, it should be tagged as singular.

EXAMPLES: The group/NN has/*have disbanded.
The jury/NN is/*are deliberating.

Language Specific?

- Chinese has circumpositions, German doesn't really gerunds, etc.

Universal Part-of-Speech Tags

1. VERB - verbs (all tenses and modes)
2. NOUN - nouns (common and proper)
3. PRON - pronouns
4. ADJ - adjectives
5. ADV - adverbs
6. ADP - adpositions (prepositions and postpositions)
7. CONJ - conjunctions
8. DET - determiners
9. NUM - cardinal numbers
10. PRT - particles or other function words
11. X - other: foreign words, typos, abbreviations
12. . - punctuation

Why do tags matter?

- Interesting linguistic question.
- Used for many downstream NLP tasks.
- Benchmark linguistic NLP task.

Why do tags matter?

- Interesting linguistic question.
- Used for many downstream NLP tasks.
- Benchmark linguistic NLP task.

However note,

- Possibly have "solved" PTB tagging (Manning, 2011)
- Deep Learning skepticism

Strawman: Sparse Tagging Models

Let,

- $\mathcal{F}$; just be the set of word type
- $\mathcal{C}$; be the set of part-of-speech tags, $|\mathcal{C}| \approx 40$
- Use a linear model, $\hat{y} = f(\mathbf{xW} + \mathbf{b})$

However this runs into clear issues.

Why is tagging hard?

1. Rare Words
   - 3% of tokens in PTB dev are unseen.
   - What can we even do with these?
2. Ambiguous Words
   - Around 50% of seen dev tokens are ambiguous in train.
   - How can we decide between different tags for the same type?

Better Tag Features: Word Properties

Representation can use specific aspects of text.

- $\mathcal{F}$; Prefixes, suffixes, hyphens, first capital, all-capital, hasdigits, etc.
- $\mathbf{x} = \sum_i \delta(f_i)$

Example: Rare word tagging

```
in 130 regular-season/JJ games ,
```

$$
\begin{aligned}
\mathbf{x} &= \delta(\texttt{prefix:3:reg}) + \delta(\texttt{prefix:2:re}) \\
&+ \delta(\texttt{prefix:1:r}) + \delta(\texttt{has-hyphen}) \\
&+ \delta(\texttt{lower-case}) + \delta(\texttt{suffix:3:son}) \ldots
\end{aligned}
$$

Better Tag Features: Tag Sequence

Representation can use specific aspects of text.

- $\mathcal{F}$; Prefixes, suffixes, hyphens, first capital, all-capital, hasdigits, etc.
- Also include features on previous tags

Example: Rare word tagging with context

```
in 130/CD regular-season/JJ games ,
```

$$
\begin{aligned}
\mathbf{x} &= \delta(\texttt{last:CD}) + \delta(\texttt{prefix:3:reg}) + \delta(\texttt{prefix:2:re}) \\
&+ \delta(\texttt{prefix:1:r}) + \delta(\texttt{has-hyphen}) \\
&+ \delta(\texttt{lower-case}) + \delta(\texttt{suffix:3:son}) \ldots
\end{aligned}
$$

Modeling Context

- Features on context require inference.
- Still standard way to do tagging.
- Very fast implementation in Stanford CoreNLP

NLP From Scratch (Collobert et al. 2011)

What if we just used words and context?

- No lexical features (mostly)
- No inference (mostly)

We will work our way up to the full paper.

1. Word representations with shared information
2. Contextual input representations
3. Semi-Supervised training.

Contents

Motivation: Dense Features

- Strawman linear model learns one parameter for each word.
- Features allow us to share information between words.
- However, requires knowledge of what to share.
- We will build up to dense, $\mathbf{x} \in \mathbb{R}^{1 \times d_{\mathrm{in}}}$.

Bilinear Model

Bilinear model,

$$\hat{\mathbf{y}} = f((\mathbf{x}^0\mathbf{W}^0)\mathbf{W}^1 + \mathbf{b})$$

- $\mathbf{x}^0 \in \mathbb{R}^{1\times d_0}$ usually $\mathbf{x}^0 \in \{0,1\}^{1\times d_0}$ or one-hot
- $\mathbf{W}^0 \in \mathbb{R}^{d_0 \times d_{\text{in}}}$, $d_0 = |\mathcal{F}|$
- $\mathbf{W}^1 \in \mathbb{R}^{d_{\text{in}} \times d_{\text{out}}}, \mathbf{b} \in \mathbb{R}^{1\times d_{\text{out}}}$; model parameters

Notes:

- Bilinear parameter interaction.
-

Bilinear Model: Intuition

$$(\mathbf{x}^0\mathbf{W}^0)\mathbf{W}^1 + \mathbf{b}$$

$$\begin{bmatrix} 0 & \dots & \color{red}{1} & \dots & 0 \end{bmatrix} \begin{bmatrix} w^0_{1,1} & \dots & w^0_{0,d_{\text{in}}} \\ & \vdots & \\ & \vdots & \\ & \vdots & \\ \color{red}{w^0_{k,1}} & \color{red}{\dots} & \color{red}{w^0_{k,d_{\text{in}}}} \\ & \vdots & \\ & \vdots & \\ & \vdots & \\ w^0_{d_0,1} & \dots & w^0_{d_0,d_{\text{in}}} \end{bmatrix} \begin{bmatrix} \color{red}{w^1_{1,1}} & \dots & \dots & \color{red}{w^1_{0,d_{\text{out}}}} \\ & \ddots & \ddots & \\ \color{red}{w^1_{d_{\text{in}},0}} & \dots & \dots & \color{red}{w^1_{d_{\text{in}},d_{\text{out}}}} \end{bmatrix}$$

Embedding Layer

$$(\mathbf{x}^0\mathbf{W}^0)$$

$$\begin{bmatrix} 0 & \dots & {\color{red}1} & \dots & 0 \end{bmatrix} \begin{bmatrix} w^0_{1,1} & \dots & w^0_{0,d_{\text{in}}} \\ & \vdots & \\ & \vdots & \\ & \vdots & \\ {\color{red}w^0_{k,1}} & {\color{red}\dots} & {\color{red}w^0_{k,d_{\text{in}}}} \\ & \vdots & \\ & \vdots & \\ & \vdots & \\ w^0_{d_0,1} & \dots & w^0_{d_0,d_{\text{in}}} \end{bmatrix}$$

- Critical for natural language applications
- Informal names for this idea,
  - Feature embeddings/ word embeddings
  - Lookup Table
  - Feature/Representation Learning
  - (In Torch, `nn.LookupTable` ($\mathbf{x}$ one-hot) or `nn.SparseLinear`)

Dense Features

Example 1: single-word classfication with embeddings

$$\mathbf{x} = v(f_1; \theta) = \delta(f_1)\mathbf{W}^0 = \mathbf{x}^0\mathbf{W}^0$$

- $v : \mathcal{F} \mapsto \mathbb{R}^{1 \times d_{\mathrm{in}}}$; parameterized embedding function

With dense features, we write bilinear model as,

$$\hat{\mathbf{y}} = f(\mathbf{x}\mathbf{W}^1 + \mathbf{b})$$

Example 2: Bag-of-words classfication with embeddings

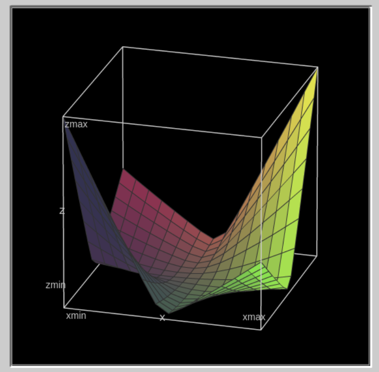
$$\mathbf{x} = \sum_{i=1}^{k} v(f_i; \theta) = \sum_{i=1}^{k} \delta(f_i)\mathbf{W}^0$$

Log-Bilinear Model

$$\hat{\mathbf{y}} = \log \mathrm{softmax}(\mathbf{x}\mathbf{W}^1 + \mathbf{b})$$

- Same form as multiclass logistic regression, but with dense features.
- However, objective is now non-convex (no restrictions on $\mathbf{W}^0$, $\mathbf{W}^1$)

Log-Bilinear Model



$$-15 \log \sigma(xy) - 5 \log \sigma(-xy) + \lambda/2 ||[x\ y]||^2$$

Does it matter?

- We are going to use SGD, in theory this is quite bad
- However, in practice it is not that much of an issue
- Argument: in large parameter spaces local optima are okay
- Lots of questions here, beyond scope of class

Embedding Gradients: Cross-Entropy

Chain Rule:

$$\frac{\partial L(f(\mathbf{x}))}{\partial x_i} = \sum_{j=1}^{m} \frac{\partial f(\mathbf{x})_j}{\partial x_i} \frac{\partial L(f(\mathbf{x}))}{\partial f(\mathbf{x})_j}$$

$$\hat{\mathbf{y}} = \log \mathrm{softmax}(\mathbf{x}\mathbf{W}^1 + \mathbf{b})$$

$$\frac{\partial L}{\partial x_f} = \sum_i W^1_{f,i} \frac{\partial L}{\partial z_{f,i}} = W^1_{f,c}(1 - \hat{y}_c) - \sum_{i \neq c} W^1_{f,i} \hat{y}_i$$

$$\mathbf{x} = \mathbf{x}^0 \mathbf{W}^0$$

$$\frac{\partial x_j}{\partial W^0_{k,j'}} = x^0_k \mathbf{1}(j = j')$$

Update:

$$\frac{\partial L}{\partial W^0_{k,j'}} = x^0_k (W^1_{j',c}(1 - \hat{y}_c) - \sum_{i \neq c} W^1_{j',i} \hat{y}_i)$$

Geometric Interpretation

[On Board]

Contents

Sentence Tagging

- $w_1, \ldots, w_n$; sentence words
- $t_1, \ldots, t_n$; sentence tags
- $\mathcal{C}$; output class, set of tags.

Window Model

Goal: predict t_5.

- Windowed word model.

$$w_1 \ w_2 \ [w_3 \ w_4 \ w_5 \ w_6 \ w_7] \ w_8$$

- w_3, w_4; left context
- w_5; Word of interest
- w_6, w_7; right context
- d_{win}; size of window ($d_{\text{win}} = 5$)

Boundary Cases

Goal: predict t_2.

$$[\langle \mathrm{s} \rangle \ w_1 \ w_2 \ w_3 \ w_4] \ w_5 \ w_6 \ w_7 \ w_8$$

Goal: predict t_8.

$$w_1 \ w_2 \ w_3 \ w_4 \ w_5 \ [w_6 \ w_7 \ w_8 \ \langle / \mathrm{s} \rangle \ \langle / \mathrm{s} \rangle]$$

Here symbols $\langle \mathrm{s} \rangle$ and $\langle / \mathrm{s} \rangle$ represent boundary padding.

Dense Windowed BoW Features

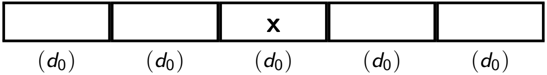
- $f_1, \ldots, f_{d_{\mathrm{win}}}$ are words in window
- Input representation is the concatenation of embeddings

$$\mathbf{x} = [v(f_1)\ v(f_2)\ \ldots\ v(f_{d_{\mathrm{win}}})]$$

Example: Tagging

$$w_1\ w_2\ [w_3\ w_4\ w_5\ w_6\ w_7]\ w_8$$

$$\mathbf{x} = [v(w_3)\ v(w_4)\ v(w_5)\ v(w_6)\ v(w_7)]$$



Rows of $\mathbf{W}^1$ encode position specific weights.

Dense Windowed Extended Features

- $f_1, \ldots, f_{d_{\text{win}}}$ are words, $g_1, \ldots, g_{d_{\text{win}}}$ are capitalization

$$\mathbf{x} = [v(f_1)\ v(f_2)\ \ldots\ v(f_{d_{\text{win}}})\ v_2(g_1)\ v_2(g_2)\ \ldots\ v_2(g_{d_{\text{win}}})]$$

Example: Tagging

$$w_1\ w_2\ [w_3\ w_4\ w_5\ w_6\ w_7]\ w_8$$

$$\mathbf{x} = [v(w_3)\ v(w_4)\ v(w_5)\ v(w_6)\ v(w_7)\ v_2(w_3)\ v_2(w_4)\ v_2(w_5)\ v_2(w_6)\ v_2(w_7)]$$

		x							
(d_0)	(d_0)	(d_0)	(d_0)	(d_0)	(d_0')	(d_0')	(d_0')	(d_0')	(d_0')

Rows of $\mathbf{W}^1$ encode position specific weights.

Tagging from Scratch (Collobert et al, 2011)

Part 1 of the key model,