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?

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

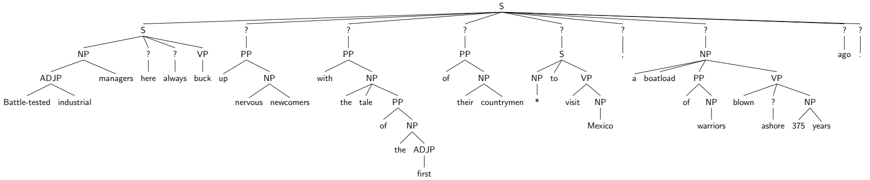
Tagging

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 .

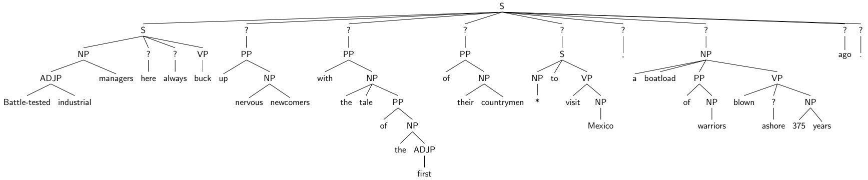
Part-of-Speech Tags

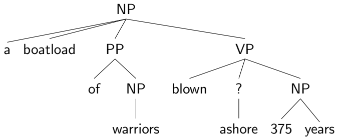
So/RB what/WP if/IN Steinbach/NNP had/VBD struck/VBN just/RB seven/CD home/NN runs/NNS in/IN 130/CD regular-season/JJ games/NNS ,/, and/CC batted/VBD in/IN the/DT seventh/JJ position/NN of/IN the/DT A/NNP 's/NNP lineup/NN ./.

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 VP buck
PP up NP nervous newcomers
PP with NP the tale PP of NP the ADJP first
PP of NP their countrymen
S NP * to VP visit NP Mexico
NP a boatload PP of NP warriors VP blown ? ashore NP 375 years
ago


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

Features used in state of the art

What if we just used words and context?

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.

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

- w_3, w_4; left context
- w_6, w_7; right context

Boundary Cases

Goal: predict t_2.

$$\left[< s > w_1 w_2 w_3 w_4\right] w_5 w_6 w_7 w_8$$

Goal: predict t_8.

$$w_1 w_2 w_3 w_4 w_5 \left[w_6 w_7 w_8 < /s >< /s >\right]$$

k Symbols $< s >$ and $< /s >$ represent boundary padding.

The Role of Features

- Recall Zipf's law.
- Many words are ..
- Can capture patterns. example.

How much does this matter?

graph of tagging.

Sparse Tagging Model

- Create training data,

$$(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$$

- Each $\mathbf{x}_i$ includes features of window.
- Each $\mathbf{y}_i$ is the one-hot tag encoding.
- Prediction accuracy is measured identically.

Naive Bayes/Logistic Regression for Tagging

- Setup is identical to text classification.
-

$$\hat{\mathbf{y}} = \mathbf{x}\mathbf{W} + \mathbf{b}$$

Contents

Collobert and Weston Natural Language Processing (almost) from Scratch

Two ideas

- Non-linear Models
- Dense Word embeddings

(1) Non-Linear Models for Classification

- Neural network represent any non-linear classifier, for example

$$NN_1 = f_1(\mathbf{x}\mathbf{W}^1 + \mathbf{b}^1))$$

$$\hat{\mathbf{y}} = f_2(NN_1\mathbf{W}^2 + \mathbf{b}^2)$$

- Where $\mathbf{W}^1 \in \mathbb{R}^{d_{\text{in}} \times dmid}$, $\mathbf{b}^1 \in \mathbb{R}^{1 \times dmid}$
- $\mathbf{W}^2 \in \mathbb{R}^{dmid \times dout}$, $\mathbf{b}^2 \in \mathbb{R}^{1 \times d_{\text{out}}}$
- Activation f_1 is non-linear.

Decision $\arg\max \hat{y}$

Can learn non-linear decision boundary. Diagram

For instance, f_1 Sigmoid and f_2 softmax

$$\frac{\partial L(y, \hat{y})}{\partial \hat{y}_j} = \frac{\mathbf{1}(y_j = 1)}{\hat{y}_j}$$

For instance, f_1 ReLU and $f_2$hinge-loss

Backpropagation

- Chain rule

Contents

(2) Dense Features

Instead of defining $\mathbf{x} = \sum_{i=1}^{n} \delta(f_i)$

Where $v : \mathcal{F} \mapsto \mathbb{R}^d$ for instance $v(f) = \delta(f)\mathbf{W}^0$

and define $\mathbf{x} = [v(f_1) \dots v(f_k)]$

(For now we assume all examples have fixed length)

Dense Features for Tagging

Instead of defining $\mathbf{x} = \sum_{i=1}^{n} \delta(f_i)$

Where $v : \mathcal{F} \mapsto \mathbb{R}^d$ for instance $v(f) = \delta(f)\mathbf{W}^0$

and define $\mathbf{x} = [v(f_1) \ldots v(f_k)]$

(For now we assume all examples have fixed length)

Dense Features for Tagging

Instead of defining $\mathbf{x} = \sum_{i=1}^{n} \delta(f_i)$

Where $v : \mathcal{F} \mapsto \mathbb{R}^d$ for instance $v(f) = \delta(f)\mathbf{W}^0$

and define $\mathbf{x} = [v^1(f_1) \dots v^1(f_k) \dots v^2(f_k + 1) \dots v^2(f_k)]$

(For now we assume all examples have fixed length)

Parameters

- With word features $|\mathcal{V}|$
- With all pair word features $|\mathcal{V}|^2$
- With word embedding features $d|\mathcal{V}|$
  Representation that allows parameter sharing.

Lookup layer is Learned too

results

Results Pretty good

objective

Diagram