Part-of-Speech Tagging

+ Neural Networks

CS 287
Last class we focused on hinge loss.

\[ L_{hinge} = \max\{0, 1 - (\hat{y}_c - \hat{y}_{c'})\} \]

Consider now the squared hinge loss, (also called \( \ell_2 \) SVM)

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

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

Part-of-Speech Data

Part-of-Speech Models

Bilinear Model

Windowed Models
( (S (CC But) (SBAR-ADV (IN while) (S (NP-SBJ (DT the) (NNP New) (NNP York) (NNP Stock) (NNP Exchange) ) (VP (VBD did) (RB n’t) (VP (VB fall) (ADVP-CLR (RB apart) ) (NP-TMP (NNP Friday) ) (SBAR-TMP (IN as) (S (NP-SBJ (DT the) (NNP Dow) (NNP Jones) (NNP Industrial) (NNP Average) ) (VP (VBD plunged) (NP-EXT (NP (CD 190.58) (NNS points) ) (PRN (: -) (NP (NP (JJS most) ) (PP (IN of) (NP (PRP it) )) (PP-TMP (IN in) (NP (DT the) (JJ final) (NN hour) ))) (: -) )))))))) (NP-SBJ-2 (PRP it) ) (ADVP (RB barely) ) (VP (VBD managed) (S (NP-SBJ (-NONE- -2) ) (VP (TO to) (VP (VB stay) (NP-LOC-PRD (NP (DT this) (NN side) ) (PP (IN of) (NP (NN chaos) )))))))))) (. .)))
Battle-tested industrial managers here always bucked up nervous newcomers with the tale of the ADJP first of the NP of their countrymen to visit NP Mexico of NP blown a boatload of NP warriors ashore 375 years
A boatload of warriors blown ashore 375 years ago.
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.
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 ./.
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 ./.
“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?

- Which of these tags are English only?
- Are there phenomena that these don’t cover?
- Should our models be language specific?
Universal Part-of-Speech Tags (Petrov et al, 2012)

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.

However note,

- Possibly have “solved” PTB tagging (Manning, 2011)
- Deep Learning skepticism
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 Word-only Tagging Models

Let,

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

- \( x = \sum_i \delta(f_i) \)

Example: Rare word tagging

in 130 regular-season/* games,

\[
\begin{align*}
x &= \delta(\text{prefix:3:reg}) + \delta(\text{prefix:2:re}) \\
    &\quad + \delta(\text{prefix:1:r}) + \delta(\text{has-hyphen}) \\
    &\quad + \delta(\text{lower-case}) + \delta(\text{suffix:3:son}) \ldots
\end{align*}
\]
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

\[
x = \delta(\text{last:CD}) + \delta(\text{prefix:3:reg}) + \delta(\text{prefix:2:re}) \\
+ \delta(\text{prefix:1:r}) + \delta(\text{has-hyphen}) \\
+ \delta(\text{lower-case}) + \delta(\text{suffix:3:son})...
\]

However, requires search. HMM-style sequence algorithms.
Exercise: What if we just used words and context?

- No word-specific features (mostly)
- No search over previous decisions

Next couple classes, we will work our way up to this paper,

1. Dense word features
2. Contextual windowed representations
3. Neural networks architecture
4. Semi-supervised training
Contents

Part-of-Speech Data

Part-of-Speech Models

Bilinear Model

Windowed Models
Motivation: Dense Features

- Strawman linear model learns one parameter for each word.
- Features allow us to share information between words.
- Can this be learned?
Bilinear Model

Bilinear model,

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

- \( x^0 \in \mathbb{R}^{1 \times d_0} \) start with one-hot.
- \( W^0 \in \mathbb{R}^{d_0 \times d_{in}}, \ d_0 = |\mathcal{F}| \)
- \( W^1 \in \mathbb{R}^{d_{in} \times d_{out}}, \ b \in \mathbb{R}^{1 \times d_{out}} \); model parameters

Notes:
- Bilinear parameter interaction.
- \( d_0 >> d_{in}, \ e.g. \ d_0 = 10000, \ d_{in} = 50 \)
Bilinear Model: Intuition

\[(x^0W^0)W^1 + b\]
Embedding Layer

\[ x^0 W^0 \]

\[
\begin{bmatrix}
0 & \ldots & 1 & \ldots & 0 \\
\vdots & & \vdots & & \vdots \\
\vdots & & \vdots & & \vdots \\
0 & \ldots & w_{d_0,1} & \ldots & w_{d_0,d_{in}}
\end{bmatrix}
\]

- Critical for natural language applications
- Informal names for this idea,
  - Feature embeddings/word embeddings
  - Lookup Table
  - Feature/Representation Learning
  - In Torch, `nn.LookupTable (x^0` one-hot)
Dense Features

When dense features implied we will write, 

\[ \hat{y} = f(xW^1 + b) \]

Example 1: single-word classification with embeddings

\[ x = \nu(f_1; \theta) = \delta(f_1)W^0 = x^0W^0 \]

\( \nu : \mathcal{F} \mapsto \mathbb{R}^{1\times d_{in}}; \) parameterized embedding function

Example 2: Bag-of-words classification with embeddings

\[ x = \sum_{i=1}^{k} \nu(f_i; \theta) = \sum_{i=1}^{k} \delta(f_i)W^0 \]
Dense Features

When dense features implied we will write,

\[ \hat{y} = f(xW^1 + b) \]

Example 1: single-word classification with embeddings

\[ x = v(f_1; \theta) = \delta(f_1)W^0 = x^0W^0 \]

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

Example 2: Bag-of-words classification with embeddings

\[ x = \sum_{i=1}^{k} v(f_i; \theta) = \sum_{i=1}^{k} \delta(f_i)W^0 \]
Log-Bilinear Model

\[ \hat{y} = \text{softmax}(xW^1 + b) \]

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

Chain Rule:

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

\[
\hat{y} = \text{softmax}(xW^1 + b)
\]

Recall,

\[
\frac{\partial L(y, \hat{y})}{\partial z_i} = \begin{cases} 
- (1 - \hat{y}_i) & i = c \\
\hat{y}_i & \text{ow.}
\end{cases}
\]

\[
\frac{\partial L}{\partial x_f} = \sum_i W^1_{f,i} \frac{\partial L}{\partial z_i} = -W^1_{f,c}(1 - \hat{y}_c) + \sum_{i \neq c} W^1_{f,i}\hat{y}_i
\]
Embedding Gradients: Cross-Entropy II

\[ x = x^0 W^0 \]

\[ \frac{\partial x_f}{\partial W^0_{k,f'}} = x^0_k 1(f = f') \]

Update:

\[ \frac{\partial L}{\partial W^0_{k,f'}} = \sum_f x^0_k 1(f = f') \frac{\partial L}{\partial x_f} = x^0_k (-W^1_{f',c} (1 - \hat{y}_c) + \sum_{i \neq c} W^1_{f',i} \hat{y}_i) \]
Contents

Part-of-Speech Data

Part-of-Speech Models

Bilinear Model

Windowed Models
Sentence Tagging

- \( w_1, \ldots, w_n; \) sentence words
- \( t_1, \ldots, t_n; \) sentence tags
- \( C; \) 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
- $w_5$; Word of interest
- $w_6, w_7$; right context
- $d_{\text{win}}$; size of window ($d_{\text{win}} = 5$)
Boundary Cases

**Goal:** predict $t_2$.

\[
\langle s \rangle \ w_1 \ w_2 \ w_3 \ w_4 \ w_5 \ w_6 \ w_7 \ w_8
\]

**Goal:** predict $t_8$.

\[
w_1 \ w_2 \ w_3 \ w_4 \ w_5 \ w_6 \ w_7 \ w_8 \langle /s \rangle \langle /s \rangle
\]

Here symbols $\langle s \rangle$ and $\langle /s \rangle$ represent boundary padding.
Dense Windowed BoW Features

- $f_1, \ldots, f_{d_{\text{win}}}$ are words in window

- Input representation is the concatenation of embeddings

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

Example: Tagging

\[
\begin{array}{ccccccc}
 w_1 & w_2 & [w_3 & w_4 & w_5 & w_6 & w_7] & w_8 \\
\end{array}
\]

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

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

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

Example: Tagging

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

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

Rows of $W^1$ encode position/feature specific weights.
Tagging from Scratch (Collobert et al, 2011)

Part 1 of the key model,