NLP 1/22 Tuesday Class 1

NLP

TA: Jiacheng Xu

Syntactic semantic
propagation

eg. C D
The city counsel refused the demonstrators a permit because they _____ violence.
refer C/D
threatened (D)
prompted (D)
used (D)
dislike (C)
caused (D)

* Diffiulty in referring the pronoun they
* Ambiguities ↓

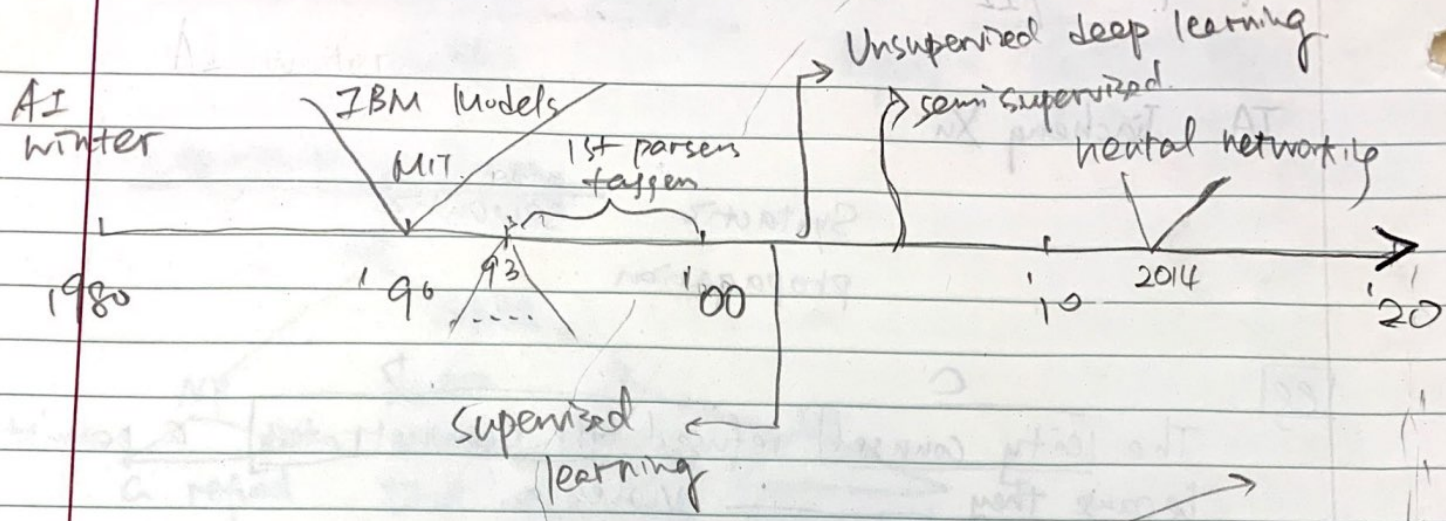
eg. Teacher Strikes Idle kids.
V. adj.
N. V.

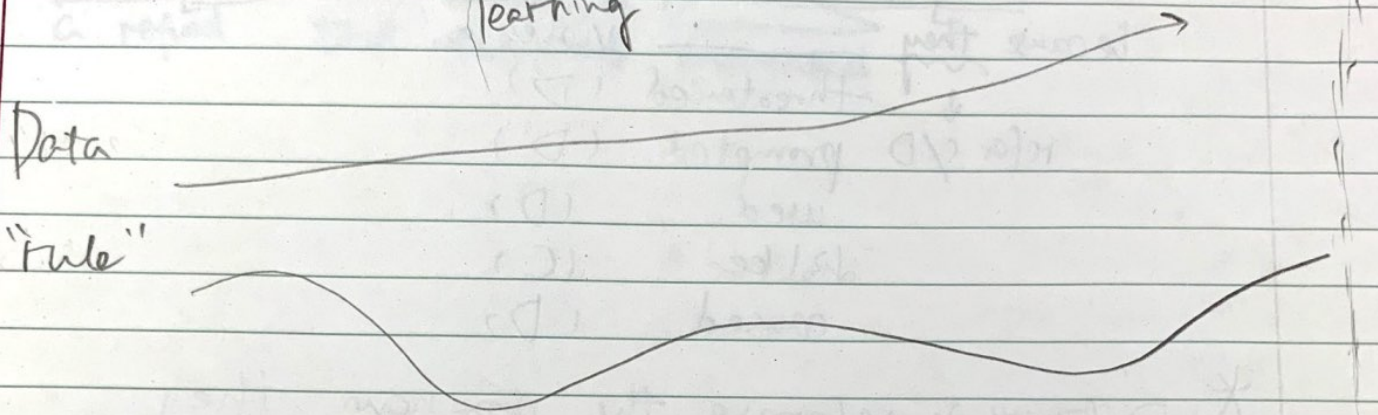
eg. Iraq Head Seeks Arms
of state weapons
of body body part

eg. kids Make Nutritious Snacks
create
are/be

eg Translation
Il fait veronout beaux => Today the weather is nice.
↳ It's nice out.
He makes today beautiful.
If fact actually handsome.

Scanned with CamScanner





NLP vs. Computational Linguistics.

words meaning

performant code

- Homework
- Midterm In-class
- Final Project : open-ended.

Scanned with CamScanner

NLP 1/24 Th | Class 2 (1)

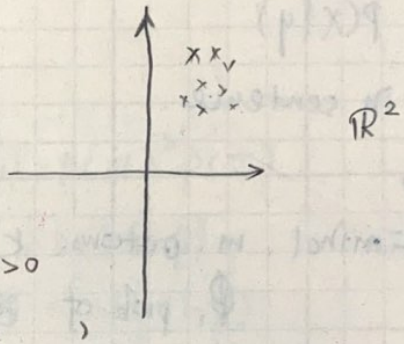
2.1, 4.1, 4.3

Outline

- Naive Bayes
- Feature Extraction
- Sentiment Analysis

Classification

Point $x \in \mathbb{R}^n$
Label $x \in \{0, 1\}$
Weights $\vec{w} \in \mathbb{R}^n$



Transformation $x \in$ ( ) ( )
$W^T x > 0$

Ex1 The movie was great! I would watch again

$\bar{x} \Rightarrow f(x) \in \mathbb{R}^n$
$W^T f(x) > 0$

Feature Extraction

Ex2 the film was awful. I'll never watch it again

Bag of words: Say we have n words
Sentence $\bar{x}$ (Ex1) $\Rightarrow$ [1 0 _ _ 1 0]^n
the a ... great awful ...

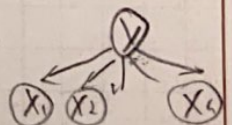
Decisions:
Value of i^{th} position = count(word i, $\bar{x}$)
Use count or just 0–1 ?
tf – idf tf (term frequency) = Count(word)
idf = $\log \frac{\#\text{ documents}}{\#\text{ docs containing words}}$

Index: Map from words to integers
the $\Rightarrow$ 0
great $\Rightarrow$ log10

unigram BOW [1 1 _ _ 0]
[the movie] [would watch]

Naive Bayes

$$P(\bar{x}, y) = P(y) P(\bar{x} | y) \triangleq P(y) \prod_{i=1}^{n} P(\bar{x}_i | y) \quad \text{defined}$$



Scanned with CamScanner

NLP Yoav Thu Class 2 (2)

P

Maximum Likelihood

$$\prod P(x, y) = \prod P(y) P(\bar{x} \mid y)$$

Weighted. H H H T

$P(H) = 0.75$

$P(H) = p \quad \prod P(y) = P(H)^3 P(T)$

$= p^3 (1-p)$

$3 \log p + \log (1-p)$

$\frac{3}{p} + \frac{1}{1-p} = 0$

+ : It was good.
+ : It was not bad.
− : It was terrible
− : It was bad.

$P(+) = P(-) = 0.5$

$P(\text{It} \mid +) = \phi_{\text{It},+} = 2/7$

$P(\text{was} \mid +) = 2/7$

$= 1/7$

$= 1/7$

$= 1/7$

It was good:

+ : $\frac{1}{7} \cdot \frac{2}{7} \cdot \frac{2}{7} \cdot \frac{1}{7} = \ldots$

− : $\frac{1}{7} \cdot \frac{2}{7} \cdot \frac{2}{7} \cdot 0 = 0$

P(negative given positive) = 0

Smoothing

$$p(x \mid y) = \frac{\text{count}(x, y)}{\text{count}(*, y)}$$

$P(\bar{x}_i \mid y) =$

Scanned with CamScanner

NLP 1/29 Tue Lecture 3 (1, 2.2, 2.4

Recap

$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

$(\bar{x}, y) \quad \bar{x} \in \mathbb{R}^n \quad y \in \{0, 1\}$

Bags of words $\bar{x}_i$ = count of word i in the sentence

Naive Bayes: $P(\bar{x}, y) = P(y)\,P(\bar{x}|y)$

$\propto \underbrace{P(y)}_{\text{categorical}} \prod_{i=1}^{n} \underbrace{\varphi_{i,y}}_{\text{multinomial}}^{\bar{x}_i} \qquad \varphi_{i,y} = P(w_i|y)$

Estimation Data $(\bar{x}^j, y^j)_{j=1}^{D}$

Maximum Likelihood: $P(y) = \frac{\text{Count}(y)}{\text{total } D}$

$\varphi_{i,y} = \frac{\text{Count of word } i \text{ in sents w/ label } y}{\text{count of all words in sents w/ label } y}$

Inference

$\underbrace{P(y|\bar{x})}_{\text{this is computed}} \propto \underbrace{P(y) \cdot P(\bar{x}|y)}_{\text{these came from model params}}$

Generative Vs **Discriminative**

$P(\bar{x}, y)$ — Can sample $(\bar{x}, y) \sim p$

$P(y|\bar{x})$ — Directly optimizes what we care about (prediction)

(Logistic Regression)

Perceptron: error-driven discriminative method

Sentiment Analysis

eg. On Powerpoint.

Scanned with CamScanner

Perceptron

Decision boundary $\bar{w}^T \bar{x} \overset{?}{\geq} 0$

if ≥ 0 +

else −

Learning Given $(\bar{x}^i, y^i)_{i=1}^{D}$ ⇒ produce $\bar{w}$

for i in range (0, epochs)

for j in range (0, D)

$y_{pred} = \text{sign}(\bar{w}^T \bar{x}^j)$ $\begin{matrix} +1 & \text{if} \geq 0 \\ -1 & \text{else} \end{matrix}$

Update $\bar{w}$: if $y_{pred} = y^j$, continue

if $y_{pred} = -1$ $y^j = +1$, $\bar{w} \leftarrow \bar{w} + \bar{x}^j$

$y_{pred} = +1$ $y^j = -1$, $\bar{w} \leftarrow \bar{w} - \bar{x}^j$

$\bar{w}_0$

$\bar{w}_1 = \bar{w}_0 + \bar{x}^j$

$\bar{w}_0 \cdot \bar{x}^j = c$

$\bar{w}_1 \cdot \bar{x}^j = c + \bar{x}^j \bar{x}^j > c$

Ex $\bar{x}^1$ = good $y^1 = +$

$\bar{x}^2$ = bad $y^2 = -$

vector: good bad not

$[1 \ 0 \ 0]$

$[0 \ 1 \ 0]$

$\bar{w} \leftarrow [0, 0]$

$[1, 0] - [0, 0]$

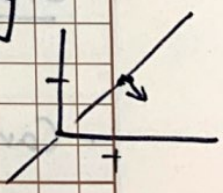
$\bar{x}$

$\bar{w}_0 \leftarrow \bar{w}_0 + [1, 0]$

$\bar{w}_1 = [1, 0]$

$\bar{w} = [1, -1]$

0,0 0,1 (0, -1).



not good

not bad

good bad not

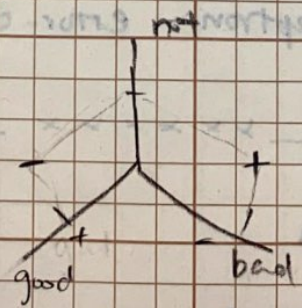
$[1 \ 0 \ 1]$

$[0 \ 1 \ 1]$

$\bar{w} = [-1 \ -1 \ 0]$

$w = [0, -1 \ -1]$.

$w = [0 \ 0 \ 0]$



seperable

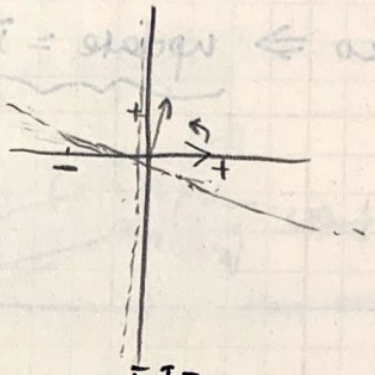
Scanned with CamScanner

Lecture 3 (2)

Ex It was good
the movie was not bad
not so good

8 words

$W_{it} = 1000$
$W_{movie} = 1000$
$W_{so} = -1000$

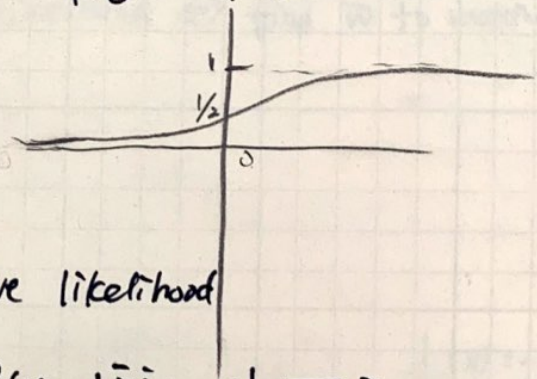


Logistic Regression

Discriminative : $P(y|\bar{x}) \triangleq \frac{e^{\bar{w}^T\bar{x}}}{1+e^{\bar{w}^T\bar{x}}} = \text{logistic}(\bar{w}^T\bar{x})$

$$\text{logistic}(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$$

$P(y|\bar{x}) > \frac{1}{2} \overset{?}{\Leftrightarrow} \bar{w}^T\bar{x} \overset{?}{>} 0$



Training

Maximizing discriminative likelihood

$$\mathcal{L} = \sum_{j:y^j=+} \log P(y=+|\bar{x}^j) + \sum_{j:y^j=-} \log P(y=-|\bar{x}^j) \approx \log \prod_j P(y \text{ is correct} | \bar{x}^j)$$

Train with stochastic gradient ascent

$\bar{w}$

Pick up $(\bar{x}^j, y)$ from data

$\bar{w} \leftarrow \bar{w} + \alpha \cdot \frac{\partial}{\partial w} \mathcal{L}((\bar{x}^j, y^j))$

Stop
$\alpha = 1$

example

$$\mathcal{L}((\bar{x}^j, y^j=+)) = \log P(y=+|\bar{x}^j)$$
$$= \log \frac{e^{\bar{w}^T\bar{x}}}{1+e^{\bar{w}^T\bar{x}}}$$
$$\mathcal{L} = \bar{w}^T\bar{x} - \log(1+e^{\bar{w}^T\bar{x}})$$
$$\frac{\partial}{\partial w_i}\mathcal{L} = x_i - \frac{1}{1+e^{\bar{w}^T\bar{x}}} \cdot e^{\bar{w}^T\bar{x}} \cdot x_i$$
$$= x_i\left(1 - \frac{e^{\bar{w}^T\bar{x}}}{1+e^{\bar{w}^T\bar{x}}}\right)$$
$$= \bar{x}_i(1 - P(y=+|\bar{x}))$$

$P(y=+|\bar{x})$

Scanned with CamScanner

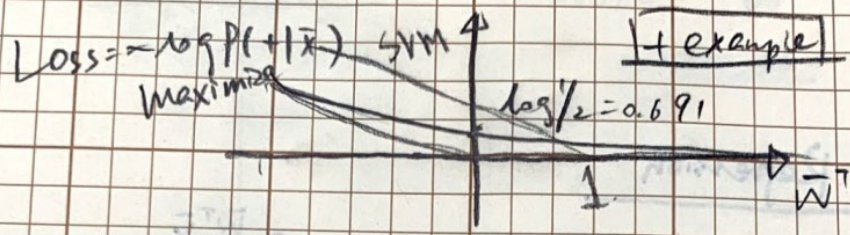
$$\frac{\partial}{\partial w} L = \bar{x}^j \left(1 - P(+ \mid \bar{x}^j)\right)$$

Recall $y^j = +$

$P(+) \approx 1 \Rightarrow$ "no update"

$P(-) \approx 0 \Rightarrow$ update $= \bar{x}^j$

$\Rightarrow$ Looks like "soft perceptron"



Scanned with CamScanner

NLP 1/31 Thursday Lecture 4 (1)

Recap Perceptron & Logistic Regression

weight vector $\vec{w}$, decision rule: $\vec{w}^T x^T > 0$ $\Leftrightarrow$ LR: $P(y=+|\vec{x}) = \frac{e^{\vec{w}^T\vec{x}}}{1+e^{\vec{w}^T\vec{x}}} \geq 0.5$

Learning

Perce

```
for i in range(0, epochs)
    for j in range(0, D)
        y_pred = sign(w^T x)
        if y_pred ≠ y^j
            if y^j = +: w ← w + α x^j      (α: step size)
                y^j = −: w ← w − α x^j
```

LR

$$P(y=+|\vec{x}) = \frac{e^{\vec{w}^T\vec{x}}}{1+e^{\vec{w}^T\vec{x}}}$$

if $y^j = +$: $\vec{w} = \vec{w} + \alpha \vec{x}(1 - P(y=+|\vec{x}))$

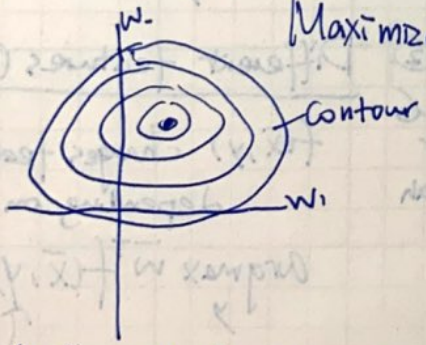
$y^j = -$: $\vec{w} = \vec{w} + \alpha \vec{x}(1 - P(y=-|\vec{x}))$

Today: Optimization, Multiclass, Application examples

Optimization LR had an objective function

$$\ell(\vec{w}) = \sum_{j:\, y^j=+} \log P(y=+|\vec{x}^j) + \sum_{j:\, y^j=-} \log P(y=-|x^j)$$

L2 regularization: $(+\|w\|^2)$



Maximize log likelihood $\Leftrightarrow$ find $\vec{w}$ to maximize $\ell(\vec{w})$

Alg1 Stochastic gradient descent ascent

Repeat $\vec{w} \leftarrow \vec{w} + \alpha \frac{\partial}{\partial \vec{w}} \ell(\vec{w})$

$\ell(x) = -x^2$ $\frac{\partial}{\partial x} -x^2 = -2x$

$x_0 = -2$

$\frac{\partial}{\partial x} -x^2 \big|_{x=-2} = 4$

$x_1 = x_0 + \alpha \cdot g = -2 + 4 = 2$

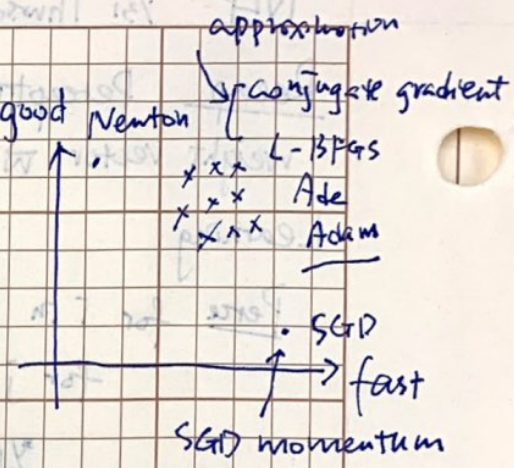
Scanned with CamScanner

Alg 2 Newton's method

$$\bar{w} \leftarrow \bar{w} + \left(\frac{\partial^2}{\partial \bar{w}^2} l\right)^{-1} \frac{\partial}{\partial \bar{w}} l$$

Hessian matrix $n \times n$: $\frac{\partial^2}{\partial w_i \partial w_j}$

gradient vector len n

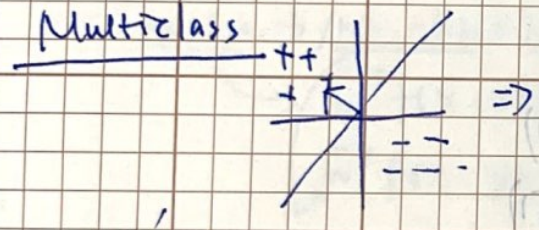


Objective

l = maximize log-likelihood

Regularization.

Multiclass



one-vs-all

BC_1 = (1 vs 2,3)

BC_2 (2 vs 1,3)

BC_3 (3 vs 1,2)

$\bar{w}_{BC_1} \bar{x} > w_{BC_2}$?

all vs all

BC_{12} (1 vs 2)

13 ⋮

23 ⋮

key idea

Two ways to do this

① Different weights (DW):

$\bar{w}_y$ for each label y

$\operatorname{argmax}_y \bar{w}_y^T \bar{x}$ = prediction

② Different features (DF): ~~$f(\bar{x}, y)$~~

$f(\bar{x}, y)$ changes feature representation depending on y

$\operatorname{argmax}_y \bar{w}^T f(\bar{x}, y)$

("hypothesized" y)

Eisenstein

Scanned with CamScanner

NLP / Lecture 4 (2)

Ex Topic classification

$\bar{x}$ = too many drug trials, too few patients → health / sports / politics

BoW: [drug patients baseball] = [1 1 0]

DW $\bar{w}_{health} = [2.0, 5.6, -3] \Rightarrow 7.6$

$\bar{w}_{sports} = [1.2, -3.1, 5.7] \Rightarrow -1.9$

DF $f(\bar{x}, y)$:= $\bar{x}$ replicated $|y|$ times (Cartesian product)

$f(\bar{x}, y=H) = [1, 1, 0, 0, 0, 0, 0, 0, 0]$

$f(\bar{x}, y=S) = [0, 0, 0, 1, 1, 0, 0, 0, 0]$ ↑ I[count of word, &label = sports]

$\bar{w} = [2.0, 5.6, -3 / 1.2, -3.1, 5.7 / \ldots]$

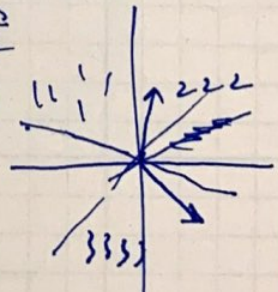
$\bar{w}^T f(\bar{x}, y=H) = 7.6$

$\bar{w}^T f(\bar{x}, y=S) = -1.9$

MC-Perception

```
for i range (0, epochs)
    for j in range (0, D)
        ypred = arg max_y  w̄_y^T x̄ / w̄^T f(x, y)
        if ypred ≠ y^j
            w̄_pred ← w̄_ypred − x̄
            w̄_j ← w̄_ypred + x̄
```

PF



$\bar{w} \leftarrow \bar{w} + f(\bar{x}, y^j) - f(\bar{x}, y_{pred})$

Scanned with CamScanner

NLP 2/5 Tuesday Lecture 5

Recap

Optimization maximize (minimize) objective function $\ell(\bar{w})$

Stochastic gradient descent (SGD for minimizing)

$$\bar{w} \leftarrow \bar{w} + \alpha \frac{\partial}{\partial w} \ell$$

α: decrease during training

evaluated on one instance

Adam / Adagrad / Adadelta ?

Multiclass

$\operatorname{argmax}_y \bar{w}_y^T \bar{x}$ — different weights

$\operatorname{argmax}_y \bar{w}^T f(\bar{x}, y)$ — different features.

eg. $\bar{x} = [\underset{\text{drug}}{1} \quad \underset{\text{patient}}{1} \quad \underset{\text{baseball}}{0}]$

$w = \begin{bmatrix} x & x & x \\ x & x & x \end{bmatrix}$ w_y

$f(\bar{x}, y=y_1) = [1,1,0,0,0,0,0,0,0]$

$f(\bar{x}, y=y_2) = [0,0,0,1,1,0,0,0,0]$

$w = [x\ x\ x\ 0 \cdots 0]$.

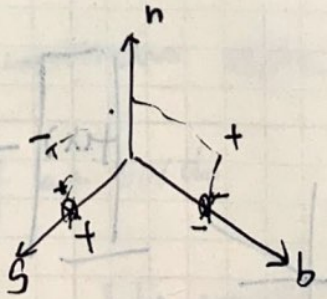
MC Perc Update: $\bar{w}_{y_{pred}} \leftarrow \bar{w}_{y_{pred}} - \alpha \bar{x}$

$\bar{w}_{y_i} \leftarrow \bar{w}_{y_i} + \alpha \bar{x}$

Neutral Nets

Linear classifier $\bar{w}^T \bar{x} \gtrless 0$

	g	b	n
good	1	0	0
bad	0	1	0
not good	1	0	1
not bad	0	1	1



add

g	b	n	ng	nb
1	0	0	0	0
0	1	0	0	0
1	0	1	1	0
0	1	1	0	1

- Bigrams require crafting. What if we need trigrams / 4-gram?
- Kernels are expensive
- We probably don't need all pairs

Scanned with CamScanner

NN: transform data into latent feature space

$\bar{z} = g(W\bar{x})$ $\bar{x} \in \mathbb{R}^3$

$w^T z > 0$

$\Rightarrow w^T(W\bar{x})$

$\Rightarrow (w^T W)^T \bar{x} \overset{?}{>} 0$

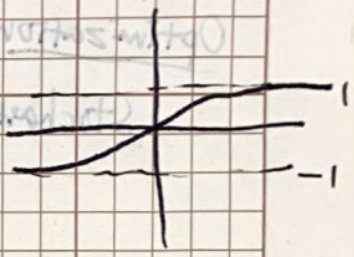
$\bar{w}'$

g = nonlinearity

tanh

$\tanh(1) \approx 1$

$\tanh(2) \approx 1$



$$W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad X = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad g(W\bar{x}) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

5×3 3×1 5×1

$y = g(Wx+b)$

$z = g(Vy+c)$

$z = g(Vg(Wx+b)+c)$

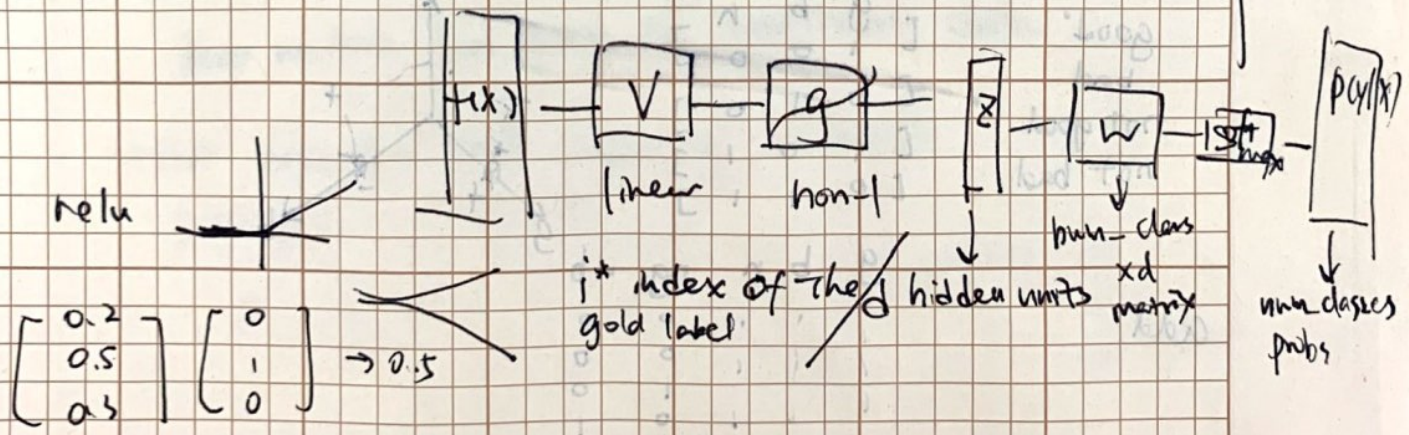
Deep Neural Network

output of 1st layer

Feed forward computation (not recurrent)

On PowerPoint

Vectorization and softmax.



i^* index of the gold label

$$L(x, i^*) = W\bar{z} \cdot e_{i^*} - \log \sum_j (W\bar{z}) \cdot e_j$$

$$\frac{\partial}{\partial W_{ij}} L(x, i^*) = \begin{cases} z_j - p(y=i|x) z_j & \text{if } i = i^* \\ -p(y=i|x) z_j & \text{otherwise} \end{cases}$$

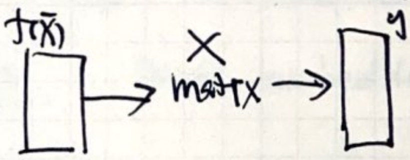
Scanned with CamScanner

NLP 2/7 Thursday Lecture 6

Recap $P(Y|X) = \text{softmax}(Wg(Vf(x)))$

---- slides ----

Batching



$x^T W \to \bar{y}$

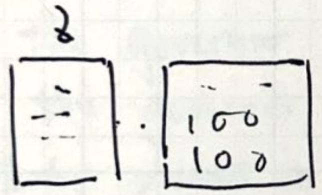
$[n] \quad [n \times m] \quad [m]$

$(\bar{X}^j, \bar{y}^j)_{j=1}^{D}$ if ~~$\bar{x}$~~ j --- , $w^T W$.

~~for ... j $f^{(j)}$ $\bar{x}$ = W~~

$x^T W \to \bar{Y}$

Refactoring: restoring competence to ~~help~~ handle multiple examples all ~~fd~~.



Slides

Word Embeddings : mapping world $n \in \mathbb{R}^n$: n=50 ~ 300

- Deep models are good at continuous data
- What does the first layer of an NN do

$\beta_0 W = [1 \; 0 \; 0 \; 1 \; 0 \; 0] = X$ (The, good, matrix)

$\bar{W}_x = \sum_{x,i} W$

the president of

— — — of

~~said~~ decreed.

NLP 2/12 Tuesday Lecture 7 (1)

Recap
- Pytorch
- Training NNs : initialization, right optimizer

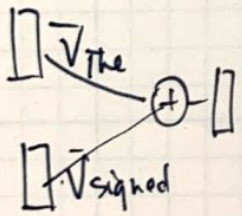
Today's Outline : Word embedding, evaluation, NNs for NLP

• Word embeddings low-dimensional (50–1000) vector representation of word types
(high-dim : $\mathbb{R}^{|V|}$ each word has its own axis)

High-low dim : "You shall know a word by the company it keeps"
- JR Firth 1957

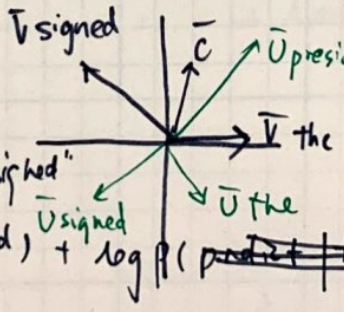
e.g. the president signed
the president announced
the governor signed
the governor announced

Mikolov predict a word from its context (and vice versa)
Each word w has a vector $\bar{v}_i$ and a context vector $\bar{u}_i$ or $\bar{v}_i$
Continous bag-of-words (CBOW)



$\bar{c} = \frac{1}{n}\sum_{j \in context} v_j$

$p(w=w_i|\bar{c}) = \frac{\exp(\bar{u}_i \cdot \bar{c})}{\sum_{i \in V} \exp(\bar{u}_i \cdot \bar{c})}$



$p(w=president|the)$
$p(signed)$

Predict "president" form context : "the — signed"
Maximize $\log p(predict|the, signed) + \log p(\cancel{predict\ the}\ signed|president, ...)$
Optimize $\bar{v}$s and $\bar{u}$s both.

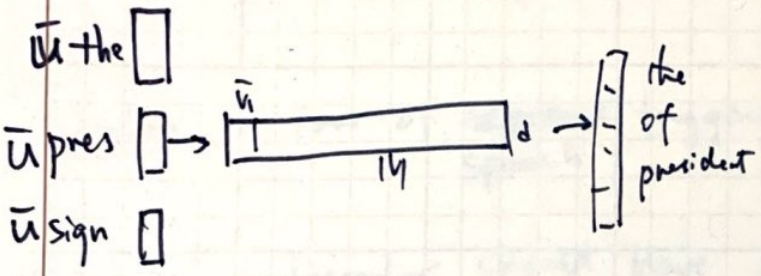
Tunable parameter : window size

... the president signed ab...
w_{s1}
w_{s2}

Scanned with CamScanner

MP Lecture 7 (2)

Skip-gram "Inverse of CBOW"

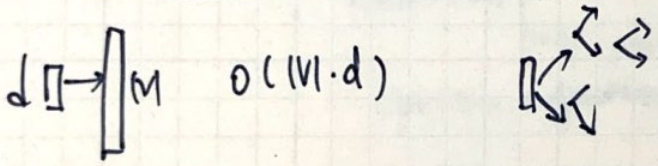


$$P(C=w_i \mid W=w_j) = \frac{\exp(\bar{U}_j \cdot \bar{V}_i)}{\sum_i \exp(\bar{U}_j \cdot \bar{V}_j)}$$

Max: $\log P(\text{the} \mid \text{president}) + \log P(\text{signed} \mid \text{president}) + \log(\text{president} \mid \text{signed}) + \log(a \mid \text{signed})$ $w_5 = 1$

$d \geq 100$
$|V| = 10{,}000$
$dV \sim 1{,}000{,}000$

Runtime Solution 1 hierarchical softmax



Solu 2 skip-gram w/ negative sampling

Data gives us (word, context) pairs

Make synthetic (fake word, center) fake word fun)

$$P(\text{real} \mid w=w_i, c=w_j) = \frac{\exp(\bar{U}_i \cdot \bar{V}_i)}{1 + (\bar{V}_i \cdot \bar{U}_i)}.$$

~~Max x log P(real | real words)~~
~~+ log P(fake | fake pairs)~~

Max x log P(real | real data)
+ log P(fake | fake data)

Scanned with CamScanner

NLP 2/14 Thursday. Lecture 8

Recap word embedding

Today Part of ~~sequence~~ speech tagging, sequence model

Cohere's ambassador should have set up the big meeting in DC yesterday
NNP NN MD (model) VB VBN (verb) RP (particle) JJ NN. IN DT (determiners) NNP NN

Closed class. pronouns. determiners. conjunctions. prep. particles
Open class, adjectives. adverbs.

Words Ambiguity: Fed raises the interest by 5% totes.
VB NN VB
NN VB n. v.

Input : X_1 X_2
Output : y_1 y_2

a multi class

Indicator features.

"Different features" ~~I [prev = raises ∧ tg = NN]~~ I [prev = raises ∧ tg = NN]

Scanned with CamScanner

Tagging Input = $\bar{x}(x_1 \cdots x_n)$

Output $\bar{y} = (y_1 \cdots y_n)$

~~Reasons~~ March out reason about a sentence's bags collectively.

① Generative Model = Hidden Markov Model

② discriminative model = conditional random fields (CRFs)

HMMS

$$P(\bar{y}, \bar{x}) = P(y_1) P(x_1|y_1) \cdot P(y_2|y_1) P(x_2|y_2) P(y_3|y_2)$$

$$P(y_1)\, P(y_2|y_1)\, P(y_3|y_2)$$

Why Markov.

Tags form a Markov process.

~~Parameter~~ Parameter initial and $P(y_1)$

$$P(y_{i+1}|y_i) = \boxed{\square}$$

$$P(x_i|y_i) = \boxed{\square}^{|y|}$$

cols: Sum to 2.

Scanned with CamScanner

NLP 2/19 Tuesday Lecture 9 th

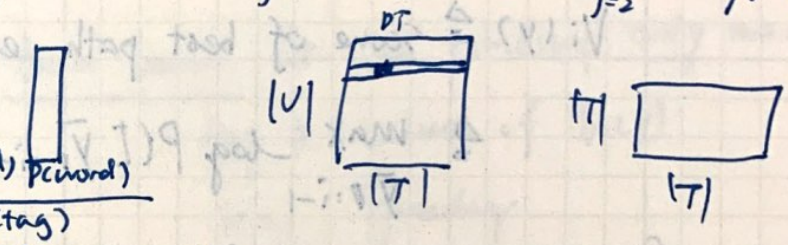
Today Estimation, Inference (NER, CRFs)

Parameter Estimation

Assume we have D examples $(\bar{x}^i, \bar{y}^i)$ for $i = 1 \dots D$

Find params to maximize $\sum_i \log p(\bar{x}^i, \bar{y}^i)$

$$= \sum_{i=1}^{D} \left[\log p(y_1^i) + \sum_{j=1}^{n} \log p(x_j^i \mid y_j^i) + \sum_{j=2}^{n} \log p(y_j^i \mid y_{j-1}^i) \right]$$



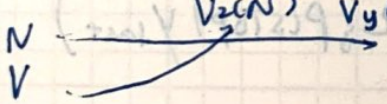
The $p(\text{word} \mid \text{tag}) = \dfrac{p(\text{tag} \mid \text{word})\, p(\text{word})}{p(\text{tag})}$

Estimation Count and normalize

Smoothing Laplace add α

Inference $\underset{\bar{y}}{\operatorname{argmax}}\, p(\bar{y} \mid \bar{x}) = \underset{\bar{y}}{\operatorname{argmax}} \left(\dfrac{p(\bar{x}, \bar{y})}{p(\bar{x})} \right) = \underset{\bar{y}}{\operatorname{arg\,max}}\, p(\bar{x}, \bar{y})$

HMM $p(\bar{x}, \bar{y}) = \pi \dots$ $= \underset{\bar{y}}{\operatorname{argmax}} \log p(\bar{x}, \bar{y})$



$$= \underset{\bar{y}}{\operatorname{argmax}} \left[\sum \log p(x_i \mid y_i) + \sum \log p(y_i \mid y_{i-1}) \right]$$

Viterbi Algorithm compute most likely tag seq from HMM

	they	can	fish	
N	-1	-3	-1	p(w|t)
V	-3	-1	-1	

rows sum to 1

Transition

START → N: -1 → V: -1 → STOP: $-\infty$

N → N: -1 → V: -1 → STOP: -1

V → N: -1 → V: -1 → STOP: -2

Scanned with CamScanner

$$\max_{\bar{y}_{1:N}} \left[\sum_{i=1}^{n} \log P(x_i|y_i) + \sum_{i=1}^{n} \log p(y_i|y_{i-1}) + \log P(\text{STOP}|y_n) \right]$$

drop — doesn't matter in computation

$$= \max_{y_n} \left[\log P(x_n|y_n) + \max_{y_{n-1}} \left[\log P(y_n|y_{n-1}) + \max_{\bar{y}_{1:n-2}} \left[\sum^{n-1} - \sum^{n-1} \right] \right] \right]$$

Define memo $V_i(y)$ = $n \times |T|$ matrix

$V_i(y) \triangleq$ score of best path ending in y at time i

$$\triangleq \max_{\bar{y}_{1:i-1}} \log P([\bar{y}_{1:i-1}, y], \bar{x}_{1:i})$$

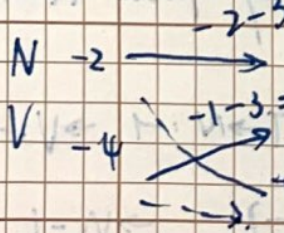
Compute $V_i(y) = \log P(x_i|y) + \max_{prev} \log P(y|y_{prev}) + V_{i-1}(y_{prev})$

$V_1(y) = \log P(x_1|y) + \log P(y|\text{START})$

Viterbi

```
for i=1 ... n          n
  for y in T:          |T|        O(n|T|^2)
    V_i(y) ← compute - V(i,y)     |T|
for y_last in T        |T|
  final_score = V_n(y_last) + log P(STOP | y_last)
Result = max final_score
```

Inference



Viterbi w/ log probs. max, + (semiring)

Viterbi w/ probs. max, *

$\sum$, * (log-Σ, Σ)

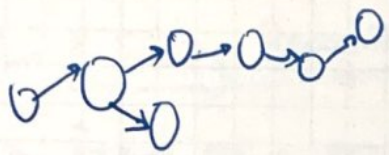
Scanned with CamScanner

MP | Lecture 9 (2)

Algorithm

Sum-product : $\Sigma, *$ $(\log\text{-}\Sigma, \Sigma)$ $\sum_{\bar{y}_{1:n}} \prod p(x_i|y_i) \ldots = p(\bar{x})$

$\sum_{\bar{y}} p(\bar{x}, \bar{y})$



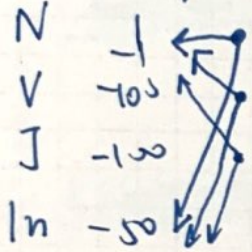
n words

|T| tags

Time complexity $O(n|T|^2)$

$V_i(y)$

they can fish



Beam Search: at ith timestep only maintain top k values of $V_i(y)$

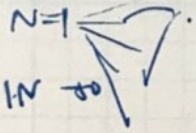
k=1 : greedy

k=2 :

Maintain priority queue over $V_i(y)$ for each i

beam

they can



n words |T| tags bs = k

$O(n|T|k \log(|T|k))$

Scanned with CamScanner

NLP, 2/21 Thursday, Lecture 16

Today

Named Entity Recognition

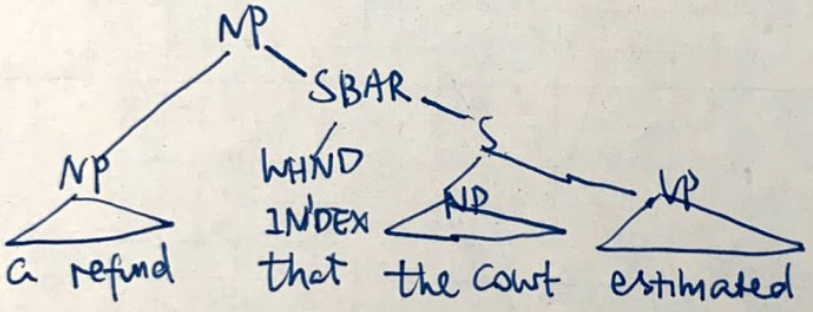
- Frame as a sequence problem with a BIO tagset: Begin, Inside, Outside
- Why might HMM not do so well now?

what if unknown words:

Conditional Random Fields (CRF)

$$P(y|x) = \frac{\prod_k \exp(\phi(x, y))}{\sum \prod_k \exp(\phi(x, y))}$$

Constituency Parsing



Scanned with CamScanner