Spring 2020 NLP HW3

Solutions

Q1

Answer:

Sample answers:

a) Maximum Likelihood Estimation:

$$P(rights) = \frac{80{,}891}{520{,}000{,}000} \qquad P(rights \mid your) = \frac{378}{883{,}614}$$

b) Estimation with add-k smoothing, $k = 0.05$:

$$P(doorposts \mid your) = \frac{0 + 0.05}{883{,}614 + 0.05 \cdot 1{,}254{,}193}$$

c) Under reasonable assumptions, higher-order models yield lower perplexity scores (or entropy scores). Thus the proper assignment is: 962, unigram; 170, bigram; 109, trigram.

Q2

Solution:

Notes:

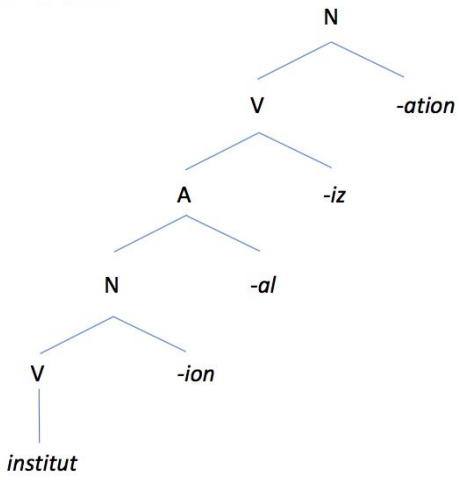
- please disregard IPA in the trees
- Trees (A) are definite answers. Responses to B and C can vary.

Institutionalization

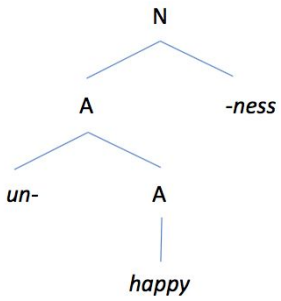
B. *-ize* - cause to become (e.g., become an institution)

C. Americanize (pattern: Adj → N)

institutionalization
[ɪnstɪtjuʃənaj͡zɛj͡ʃən]

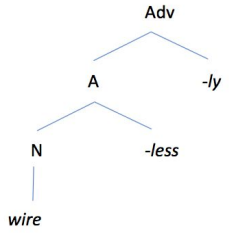


unhappiness
[ʌnhæpinɪs]



Unhappiness
B. *-ness* - state of possessing the quality of the root (i.e., state of being happy)
C. goodness (pattern: Adj → N)

Wirelessly
B. *-ly* - in the manner specified by the root (an action done without wires is done wirelessly)
C. slowly (pattern: Adj → Adv)



Q3
Solution:

- **reverse cascade**: 3 IDF = log(10000/3) ≈ 8.11
- **full shower**: 50 IDF = log(10000/50) ≈ 5.30
- **half bath**: 10 IDF = log(10000/10) ≈ 6.91
- **multiplex**: 3 IDF = log(10000/3) ≈ 8.11

Term Frequencies				
	Documents			
	Doc 1	Doc 2	Doc 3	Doc 4
reverse cascade	8	10	0	0
full shower	3	1	2	2
half bath	0	0	8	7
multiplex	2	2	2	9

TFIDF for terms in documents				
	Documents			
	Doc 1	Doc 2	Doc 3	Doc 4
reverse cascade	8.11 * 8 = 64.88	8.11 * 10 = 81.10	0	0
full shower	5.30 * 3 + 15.90	5.30 * 1 = 5.30	5.30 * 2 = 10.60	5.30 * 2 = 10.60
half bath	0	0	6.91 * 8 = 55.28	6.91 * 7 = 48.37
multiplex	8.11 * 2 = 16.22	8.11 * 2 = 16.22	8.11 * 2 = 16.22	8.11 * 9 = 72.99

Q4

Answer:

A: It will likely increase the capacity to overfit because it can make a more complex model.

B: It will likely decrease the capacity to overfit because less training gives it less time to overfit.

C: It will likely increase capacity because it can create more complex models.

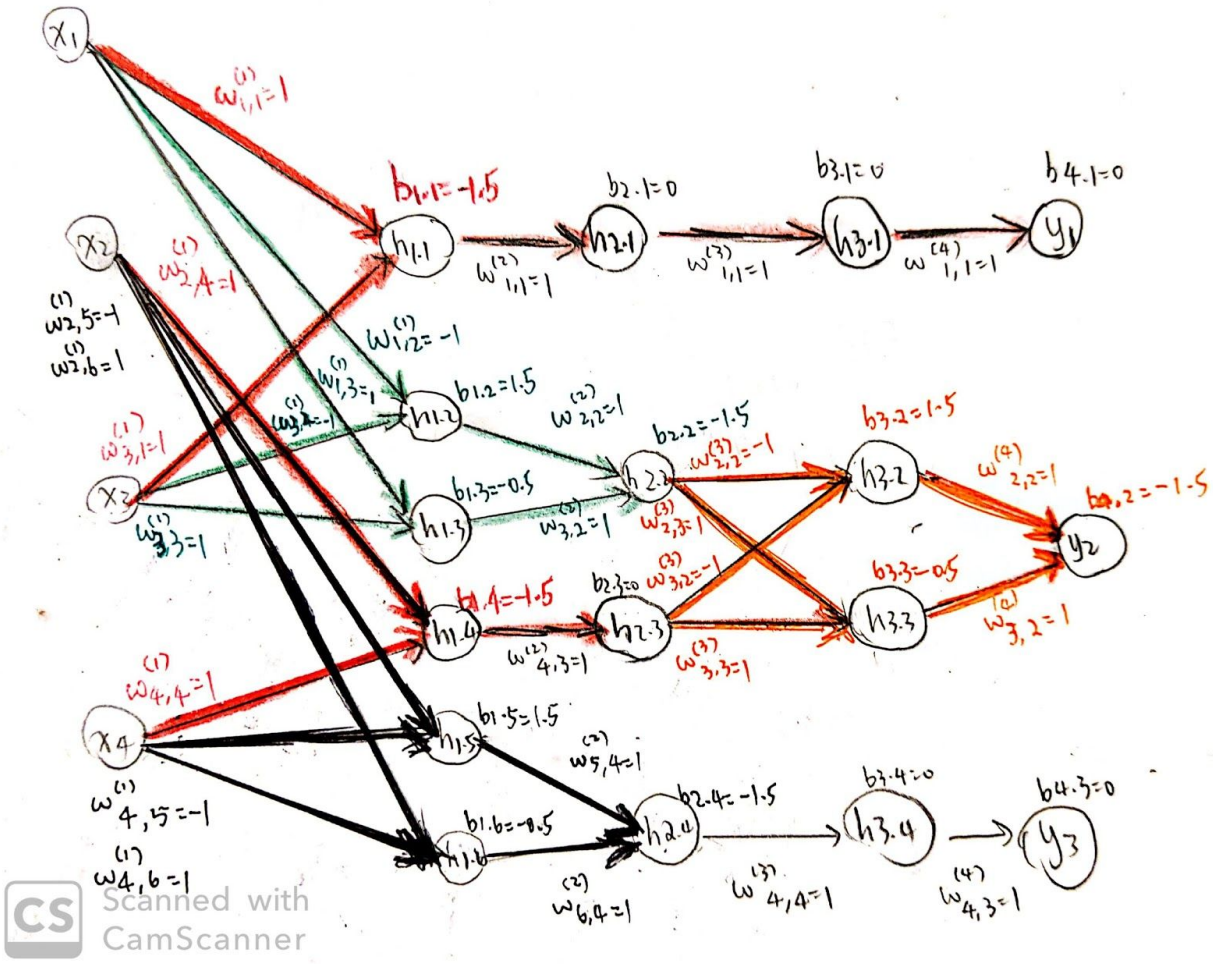
D: it will likely decrease capacity because nonlinear activations are more expressive, but also because a multilayer perceptron with linear activations is the same as a single layer perceptron because of matrix multiplication, so same answer as A.

E: It will increase because more training data increases ability to generalize and less decreases ability. It's also easier to memorize the dataset.

Q5

Answer:

A possible solution:



Q6

Answer

Loss Function:

$$L = \frac{1}{n}\sum_{i=1}^{n} - log\, p(Y = Y_i|X_i) + \frac{\lambda}{2}(\|W_1\|^2 + \|W_2\|^2 + \|W_3\|^2)$$

$$Letp(Y = Y_i|X_i) = P_k = \frac{e^{V_k}}{\sum_j e^{V_j}} where V_k = (W_3 h_2 + b_3)_k, v = W_3 h_2 + b_3$$

$$\frac{\partial L}{\partial V_k} = \frac{\partial L}{\partial P_k} * \frac{\partial P_k}{\partial V_k} = \left(\frac{1}{n}\Sigma - \frac{1}{P_k}\right) * \left(\frac{\sum_j e^{V_j} * e^{V_k} - (e^{V_k})^2}{\left(\sum_j e^{V_j}\right)^2}\right)$$
$$= \left(\frac{1}{n}\Sigma - \frac{1}{P_k}\right) * (P_k - (P_k)^2) = \frac{1}{n}\sum(P_k - 1)$$

$$\frac{\partial L}{\partial W_3} = \frac{\partial L}{\partial V_k} * \frac{\partial V_k}{\partial W_3} + \lambda\|W_3\| = \frac{1}{n}\sum(P_k - 1){h_2}^T + \lambda\|W_3\|$$
$$\frac{\partial L}{\partial b_3} = \frac{\partial L}{\partial V_k} * \frac{\partial V_k}{\partial b_3} = \frac{1}{n}\sum(P_k - 1)$$
$$\frac{\partial L}{\partial h_2} = \frac{\partial L}{\partial V_k} * \frac{\partial V_k}{\partial h_2} = \frac{1}{n}\sum {W_3}^T (P_k - 1), \frac{\partial L}{\partial h_2} = 0 if\, h_2 < 0$$

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial h_2} * \frac{\partial h_2}{\partial W_2} + \lambda\|W_2\| = \frac{1}{n}\sum {W_3}^T (P_k - 1){h_1}^T + \lambda\|W_2\|$$
$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial h_2} * \frac{\partial h_2}{\partial b_2} + \lambda\|W_2\| = \frac{1}{n}\sum {W_3}^T (P_k - 1)$$
$$\frac{\partial L}{\partial h_1} = \frac{\partial L}{\partial h_2} * \frac{\partial h_2}{\partial h_1} = \frac{1}{n}\sum {W_2}^T {W_3}^T (P_k - 1), \frac{\partial L}{\partial h_1} = 0 if\, h_1 < 0$$

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial h_1} * \frac{\partial h_1}{\partial W_1} + \lambda\|W_1\| = \frac{1}{n}\sum {W_2}^T {W_3}^T (P_k - 1)x^T + \lambda\|W_1\|$$
$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial h_1} * \frac{\partial h_1}{\partial b_1} = \frac{1}{n}\sum {W_2}^T {W_3}^T (P_k - 1)$$

Q7
Answer

Part A:

The maximum value of the perplexity is ∞.

Given any sentence $x^{(i)}$, if $p(x^{(i)}) = 0$, then $L = -\infty$ and $2^L = \infty$

Part B:

The minimum value of the perplexity is 1.

Given any sentence $x^{(i)}$, if $p(x^{(i)}) = 1$, then $L = 0$ and $2^L = 1$

Part C:

Training corpus: the a STOP

Test corpus: a the STOP

The bigram language model trained on the training corpus gives 0 probability to the sentence “a the STOP”, so perplexity is infinite.

Part D:

Training corpus: the a STOP

Test corpus: the a STOP

Likewise, the bigram language model trained on the training corpus gives the sentence “the a STOP” a probability 1 of occurring.

Q8
Answer

Part A: mapping

f(s) = a vector that contains a feature for each word w that counts the number of times w is seen in s

Part B: dynamic programming algorithm

F(0,0) = 0
F(i,0) = 0
F(0,j) = 0

F(i,j) = max[F(i-1, j-1) + s(i,j),
F(i-1, j),
F(i,j-1),
F(i-2,j-1) + s(i-1,j) + s(i,j),
F(i-1,j-2) + s(i,j-1) + s(i,j)]

Part C: modified dynamic programming algorithm
F(0,0) = 0
F(i,0) = -ip
F(0,j) = -jp

F(i,j) = max[F(i-1,j-1)+s(i,j),
F(i-1,j)-p,
F(i,j-1)-p,
F(i-2,j-1)+s(i-1,j)+s(i,j),
F(i-1,j-2)+s(i,j-1)+s(i,j)]