FSAN850

Seminar: Data Science

Summer 2021

Final Exam

June 9, 2021

9am-12am

Name:

CLOSE BOOKS, NOTES AND COMPUTERS.

Good luck!

1. (6pt) Given the training data below, answer the following questions regarding perceptron learning. Initially, we randomly set $w_0 = 0$, $w_1 = 0.1$, $w_2 = 0.1$, $w_3 = 0.1$.

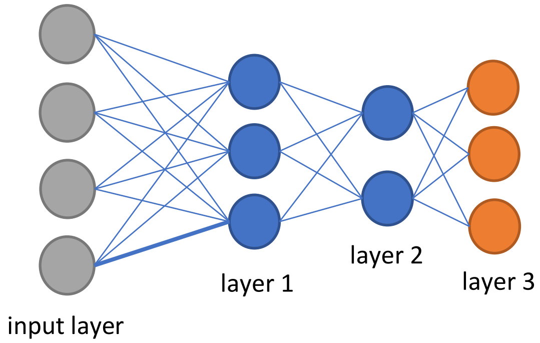
X_1	X_2	X_3	Y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1

   a. (2pt) Calculate the updated values of w_0, w_1, w_2, w_3 after the first training record using the perceptron rule, given learning rate $\eta = 0.1$.

   b. (2pt) Calculate the updated values of w_0, w_1, w_2, w_3 after the first training record using the delta rule with stochastic gradient descent, given learning rate $\eta = 0.1$.

   c. (2pt) Calculate the updated values of w_0, w_1, w_2, w_3 after the first batch using the delta rule with gradient descent, given learning rate $\eta = 0.1$, batch size = 2 and the first batch is comprised of the first two training records.

2. (9pt) The figure below shows the structure of a feedforward neural network. It is used to solve a classification problem of three classes. Suppose $x \in R^4$ is the feature vector of one training instance and $t \in R^3$ is the associated one-hot encoded label vector with $t_2 = 1$.



The activation function at layer 3 is the softmax function. In general, softmax is a vector-to-vector function. For our case, the input is $z^3 \in R^3$ and output is $o^3 \in R^3$.

a. (1pt) Give the mathematical formulation of softmax.

b. (2pt) Why should we use softmax as the activation function at layer 3?

For the neural network in the above figure, layer l has weight matrix W^l and bias vector b^l. W_{ij}^l is the weight connecting the j'th neuron in layer $l-1$ to the i'th neuron in layer l. For layer l, the input vector is a^{l-1}, the activation vector from layer $l-1$, and z^l is the output vector before activation while a^l is the output vector after activation. All layers except for layer 3 use the same scalar-to-scalar activation function f_a.

c. (3pt) Give the mathematical formulation of the neural network that takes x as input and outputs o, the vector of predicted probabilities of each class.
Note that $o = a^3$ and $x = a^0$. Please express o layer by layer. For each layer, please specify the input and the output dimension.

d. (3pt) Let $Loss(t, o)$ denote the cross-entropy loss for the training instance (x, t). Give its mathematical formulation. Explain why it is proper for solving a classification task.

3. (12pt) Suppose we are training a word2vec model on a sequence of words $[w_{t_1}, w_{t_2}, \dots, w_{t_l}]$ of length l. The vocabulary size is n_v. Vocabulary words are denoted by $w_1, w_2, \dots, w_{n_v}$.The word embedding dimension is n_e. $W^{(i)} \in R^{n_v \times n_e}$ and $W^{(o)} \in R^{n_v \times n_e}$ are the input and output word embedding matrices respectively. $v_w^{(i)}$ and $v_w^{(o)}$ are respectively the input and output word vectors of word w.
Consider a particular training instance (c, w) where c is the sequence of context words of w (the surrounding $2n_w$ words of w, left n_w and right n_w). For this specific training instance, answer the following questions.
Please use σ for the sigmoid function. For your reference, $\sigma(x) = \frac{1}{1+\exp(-x)}$ and $\sigma(-x) = 1 - \sigma(x)$.

a. (1pt) For the Continuous Bag-of-words (CBOW) model, the context c is represented as a vector v_c. Give its mathematical formulation.

b. (2pt) For CBOW, give the mathematical formulation of the loss function, $Loss(c, w)$, based on negative sampling with k negative words $[nw_1, nw_2, \dots, nw_k]$ sampled from the vocabulary.

c. (2pt) What task do we want to solve with the negative sampling loss? Why do we want to use negative sampling instead of letting $Loss(c, w) = -\log P(w|c)$ and formulating $P(w|c)$ with the softmax function?

d. (3pt) For CBOW, consider the hierarchical softmax formulation of $Loss(c, w)$. Suppose $n_v = 4$. Draw the corresponding binary tree where each leaf node corresponds to a vocabulary word. Suppose $w = w_2$, that is, the second leaf node counted from left. Give the mathematical formulation of $P(w|c)$. You should mark the internal nodes of the binary tree and use their notations to describe $P(w|c)$.

e. (2pt) Briefly explain the difference between CBOW and the Continuous Skip-gram model in terms of their learning objectives. (No need for mathematical details)

f. (2pt) What is contextualized word embedding? Explain the difference with the word vectors produced by word2vec.

4. (8pt) Negative sampling is simplified from noise contrastive estimation (NSE) which is based on the following procedure. For each observed context-word pair (c, w), match it with k noisy context-word pairs $[(c, nw_1), (c, nw_2) \dots (c, nw_k)]$ where nw_i is a noisy word sampled from a noise contrastive distribution $P_n(w)$. In this way, we obtain a dataset where observed context-word pairs are mixed with those sampled from the noisy distribution. For each pair (c, w) (w can be either an observed word or a noisy word) in this dataset, we assign a label, l, to it according to how it is generated. $l = 1$ means that (c, w) is a truly observed pair while $l = 0$ means that (c, w) is a noisy pair. The model is required to solve a binary classification task to discriminate observed pairs from noisy pairs. Specifically, a general training instance presented to the model has the form (c, w, l) where (c, w) is the input feature and l is the true label.

a. (2pt) What does $P(l = 1|c)$ equal to? (If you draw randomly from an urn containing 1 red ball and k black balls, what is the probability of getting the red ball?)

b. (2pt) What does $P(w|c, l=0)$ equal to? (How has the noisy words nw_i been sampled?)

Suppose we model $P(w|c, l=1)$ with the softmax function and $P(w|c, l=1) \propto \exp(v_c^T v_w^{(o)})$ where v_c is the vector representation of c in CBOW. We can express $\log P(w|c, l=1)$ in the form of $\log A - \log B$. That is

$$\log P(w|c, l=1) = v_c^T v_w^{(o)} - \log \sum_{w'} \exp\left(v_c^T v_{w'}^{(o)}\right)$$

The term $\sum_{w'} \exp\left(v_c^T v_{w'}^{(o)}\right)$ is interpreted as the normalization constant. Let's denote it by Z_c because it is specific to c. Then, we have

$$\log P(w|c, l=1) = v_c^T v_w^{(o)} - \log Z_c$$

Recall that the goal of word2vec is to maximize the probability of the observed word in a given context. In this sense, we can rewrite $P(w|c, l=1)$ as $P_\theta(w|c)$ to emphasize that there is a nested word embedding model parameterized by θ. For CBOW, $\theta = (W^{(i)}, W^{(o)})$. To wrap up, we have

$$\log P(w|c, l=1) = \log P_\theta(w|c) = v_c^T v_w^{(o)} - \log Z_c$$

c. (2pt) According to Bayes rule, how is $P(l=1|c, w)$ related to $P(w|c, l=0)$ and $P(w|c, l=1)$? Use the results from (a) and (b) and the notation $P_\theta(w|c)$ to simplify the expression. (Hint: context c should always appear in the conditioning side)

d. (1pt) Use the identity $x = \exp \log x$ to reformulate $P(l=1|c, w)$ as the output of the sigmoid function. Specifically, show that $P(l=1|c, w) = \sigma(\log P_\theta(w|c) - \log k - \log P_n(w))$. For your reference, $\sigma(x) = \frac{1}{1+\exp(-x)}$.

If we plug in $\log P_\theta(w|c) = v_c^T v_w^{(o)} - \log Z_c$, then we have

$$P(l = 1|c, w) = \sigma\left(v_c^T v_w^{(o)} - \log Z_c - \log k - \log P_n(w)\right)$$

The term $-\log Z_c$ is hard to compute in general. NCE claims that we can estimate it as a separate model parameter b_c and even fix this parameter to zero! Suppose that $-\log Z_c$ is fixed to 0, then we have

$$P(l = 1|c, w) = \sigma\left(v_c^T v_w^{(o)} - \log k - \log P_n(w)\right)$$

When $v_c^T v_w^{(o)} = \log k + \log P_n(w)$, $P(l = 1|c, w) = \sigma(0) = 0.5$. In other words, the compatibility score between c and w measured by $v_c^T v_w^{(o)}$ has to be higher than the baseline score $\log k + \log P_n(w)$ in order to let the model think that (c, w) is more likely to be a truly observed pair rather than a noisy pair.

e. (1pt) Because each context c is associated with $k + 1$ context-word pairs, for context c, we want to minimize

$$-\left(\log P(l = 1|c, w) + \sum_{i=1}^{k} P\,(l = 0|c, nw_i)\right)$$

use the results from (d) to expand this objective. Compare it with the loss based on negative sampling. What is the difference?

5. (5pt) Suppose we want to predict the sentiment polarities of Yelp reviews. For example, the review “this place is so much fun” has a positive label (=1) while the review “the food is disgusting” has a negative label (=0). Consider using a recurrent neural network (RNN) to solve the binary classification task based on sentence embedding.
   a. (4pt) Design a RNN where the last hidden state is used to represent the sentence and predict the sentiment label. Plot the structure of your model.

b. (1pt) What is the drawback of the previous approach? Could you give an improved solution?

6. (5pt) Let f be a non-numeric query, R be the range of outcome, D and D' be neighboring datasets. $q(D,r)$ represents the quality score of the outcome $r \in R$, given dataset D. The Exponential Mechanism A_E is defined as

$$A_E(D,R,q,\epsilon) = \frac{exp\,(\frac{\epsilon * q(D,r)}{2 * \Delta q})}{\sum_{r \in R} exp\,(\frac{\epsilon * q(D,r)}{2 * \Delta q})},$$

where

$$\Delta q = \max_{\|D-D'\|_1 \leq 1,\ r \in R} \|q(D,r) - q(D',r)\|$$

Prove the Exponential Mechanism satisfies ϵ-differential privacy.

7. (5pt) Suppose we want to elect a student senator to represent our FSAN program. We have two candidates, denoted as A and B. Assume here are n semi-honest voters, and all voters must vote for one candidate. Please design a privacy-preserving voting protocol to preserve each voter's decision and explain why your protocol works.
[Hint: Introducing any semi-honest helpers is allowed.]