Q1. Machine Learning: Potpourri

(a) What is the minimum number of parameters needed to fully model a joint distribution \( P(Y, F_1, F_2, ..., F_n) \) over label \( Y \) and \( n \) features \( F_i \)? Assume binary class where each feature can possibly take on \( k \) distinct values.

(b) Under the Naive Bayes assumption, what is the minimum number of parameters needed to model a joint distribution \( P(Y, F_1, F_2, ..., F_n) \) over label \( Y \) and \( n \) features \( F_i \)? Assume binary class where each feature can take on \( k \) distinct values.

(c) You suspect that you are overfitting with your Naive Bayes with Laplace Smoothing. How would you adjust the strength \( k \) in Laplace Smoothing?

- [ ] Increase \( k \)
- [ ] Decrease \( k \)

(d) While using Naive Bayes with Laplace Smoothing, increasing the strength \( k \) in Laplace Smoothing can:

- [ ] Increase training error
- [ ] Decrease training error
- [ ] Increase validation error
- [ ] Decrease validation error

(e) It is possible for the perceptron algorithm to never terminate on a dataset that is linearly separable in its feature space.

- [ ] True
- [ ] False

(f) If the perceptron algorithm terminates, then it is guaranteed to find a max-margin separating decision boundary.

- [ ] True
- [ ] False

(g) In multiclass perceptron, every weight \( w_y \) can be written as a linear combination of the training data feature vectors.

- [ ] True
- [ ] False

(h) For binary class classification, logistic regression produces a linear decision boundary.
○ True  ○ False

(i) In the binary classification case, logistic regression is exactly equivalent to a single-layer neural network with a sigmoid activation and the cross-entropy loss function.

○ True  ○ False

(j) (i) You train a linear classifier on 1,000 training points and discover that the training accuracy is only 50%. Which of the following, if done in isolation, has a good chance of improving your training accuracy?

☐ Add novel features  ☐ Train on more data  ☐ Train on less data

(ii) You now try training a neural network but you find that the training accuracy is still very low. Which of the following, if done in isolation, has a good chance of improving your training accuracy?

☐ Add more hidden layers  ☐ Add more units to the hidden layers
Q2. A Nonconvolutional Nontrivial Network

You have a robotic friend MesutBot who has trouble passing Receptchas (and Turing tests in general). MesutBot got a 99.99% on the last midterm because he could not determine which squares in the image contained stop signs. To help him ace the final, you decide to design a few classifiers using the below features.

- $A = 1$ if the image contains an octagon, else 0.
- $B = 1$ if the image contains the word STOP, else 0.
  - $S = 1$ if the image contains the letter S, else 0.
  - $T = 1$ if the image contains the letter T, else 0.
  - $O = 1$ if the image contains the letter O, else 0.
  - $P = 1$ if the image contains the letter P, else 0.
- $C = 1$ if the image is more than 50% red in color, else 0.
- $D = 1$ if the image contains a post, else 0.

(a) First, we use a Naive Bayes-inspired approach to determine which images have stop signs based on the features and Bayes Net above. We use the following features to predict $Y = 1$ if the image has a stop sign anywhere, or $Y = 0$ if it doesn’t.

(i) Which expressions would a Naive Bayes model use to predict the label for $B$ if given the values for features $S = s, T = t, O = o, P = p$? Choose all valid expressions.

- $b = \arg \max_b P(b|a)P(s|b)P(t|b)P(o|b)P(p|b)$
- $b = \arg \max_b P(s|b)P(t|b)P(o|b)P(p|b)$
- $b = \arg \max_b P(b|s,t,o,p)$
- $b = \arg \max_b P(b,s,t,o,p)$
- $b = \arg \max_b P(s,t,o,p|b)$
- None

(ii) Which expressions would we use to predict the label for $Y$ with our Bayes Net above? Assume we are given all features except $B$. So $A = a, S = s, T = t, etc$. For the below choices, the underscore means we are dropping the value of that variable. So $y_{-b} = (0, 1)$ would mean $y = 0$.

- $y_{-b} = \arg \max_y P(y)P(a|y)P(b|y)P(c|y)P(d|y)P(s|b)P(t|b)P(o|b)P(p|b)$
- $y_{-b} = \arg \max_y P(s)P(t)P(o)P(p)P(a)P(b|s,t,o,p)P(c)P(d)P(y|a,b,c,d)$
- First compute $b' = \arg \max_b$ of the formula chosen in part (ii).
- Then compute $y = \arg \max_y P(y)P(a|y)P(b'|y)P(c|y)P(d|y)$
- First compute $b' = \arg \max_b$ of the formula chosen in part (ii).
- Then compute $y = \arg \max_y P(y|a,b',c,d)$
- $y = \arg \max_y \sum_{b'} P(y|a|y)P(b'|y)P(c|y)P(d|y)P(s|b')P(t|b')P(o|b')P(p|b')$
- None

(iii) One day MesutBot got allergic from eating too many cashews. The incident broke his letter $S$
detector, so that he no longer gets reliable $S$ features. Now what expressions would we use to predict the label for $Y$? Assume all features except $B, S$ are given. So $A = a, T = t, O = o, \text{ etc.}$

$\square \ y = \arg\max_y P(y)P(a|y)P(c|y)P(d|y)$

$\square \ y, --, -- = \arg\max_{y,b,s} P(y)P(a|y)P(b|y)P(c|y)P(d|y)P(s|b)P(t|b)P(o|b)P(p|b)$

$\square \ y, -- = \arg\max_{y,b,o} P(y)P(a|y)P(b|y)P(c|y)P(d|y)P(s|b)P(t|b)P(o|b)P(p|b)$

$\square \ y, -- = \arg\max_{y,b} P(y)P(a|y)P(b|y)P(c|y)P(d|y)P(t|b)P(o|b)P(p|b)$

$\square \ y, -- = \arg\max_{y,b} P(y)P(a|y)P(b|y)P(c|y)P(d|y)P(s|b)P(t|b)P(o|b)P(p|b)$

$\square \ y = \arg\max_{y,b} P(y)P(a|y)P(c|y)P(d|y) \sum_{b', s'} P(b'|y)P(s'|b')P(t|b')P(o|b')P(p|b')$

$\square \ None$

(b) You decide to try to output a probability $P(Y|\text{features})$ of a stop sign being in the picture instead of a discrete ±1 prediction. We denote this probability as $P(Y|\tilde{f}(x))$. Which of the following functions return a valid probability distribution for $P(Y = y|\tilde{f}(x))$? Recall that $y \in \{-1, 1\}$.

$\square \ \frac{e^y u^T \tilde{f}(x)}{e^{-y u^T \tilde{f}(x)} + e^{y u^T \tilde{f}(x)}}$

$\square \ \frac{\frac{1}{2}}{1 + e^{-\sigma^T \tilde{f}(x)}}$

$\square \ \frac{0.5}{1 + e^{-\sigma^T \tilde{f}(x)}}$

$\square \ \frac{1}{1 + e^{\sigma^T \tilde{f}(x)}} + 1$

$\square \ None$
Unimpressed by the perceptron, you note that features are inputs into a neural network and the output is a label, so you modify the Bayes Net from above into a Neural Network computation graph. Recall the logistic function \( s(x) = \frac{1}{1+e^{-x}} \) has derivative \( \frac{\partial s(x)}{\partial x} = s(x)[1 - s(x)] \).

(c) For this part, ignore the dashed edge when calculating the below.

(i) What is \( \frac{\partial \text{Loss}}{\partial w_A} \)?
- \( \frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot A \)
- \( 2(s(X) - y^*) \cdot [s(X) \cdot (1 - s(X))] \cdot A \)
- \( \frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot 2A + 1 \)
- \( \frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot 2A \)
- \( 2(s(X) - y^*) \cdot [s(X) \cdot (1 - s(X))] \cdot A + 1 \)
- \( \frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot A + 1 \)
- None
(ii) What is $\frac{\partial \text{Loss}}{\partial w_B}$? Keep in mind we are still ignoring the dotted edge in this subpart.

- $\frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot w_B \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot E \geq 0 \cdot S$
- $2(s(X) - y^*) \cdot [s(X) \cdot (1 - s(X))] \cdot w_B \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot E \geq 0 \cdot S$
- $\frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot w_B \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 2S + S$
- $\frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot w_B \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 2S$
- $2(s(X) - y^*) \cdot [s(X) \cdot (1 - s(X))] \cdot w_B \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot E \geq 0 \cdot S + S$
- $\frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot w_B \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot E < 0 \cdot S + S$
- None

(d) MesutBot is having trouble paying attention to the S feature because sometimes it gets zeroed out by the ReLU, so we connect it directly to the input of s(·) via the dotted edge. For the below, treat the dotted edge as a regular edge in the neural net.

(i) Which of the following is equivalent to $\frac{\partial \text{Loss}}{\partial w_A}$?

- $\frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot A$
- $2(s(X) - y^*) \cdot [s(X) \cdot (1 - s(X))] \cdot A$
- $\frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot 2A + A$
- $\frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot 2A$
- $2(s(X) - y^*) \cdot [s(X) \cdot (1 - s(X))] \cdot A + A$
- $\frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot A + A$
- None

(ii) Which of the following is equivalent to $\frac{\partial \text{Loss}}{\partial w_S}$? Keep in mind we are still treating the dotted edge as a regular edge.

- $\frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot w_B \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot E \geq 0 \cdot S$
- $2(s(X) - y^*) \cdot [s(X) \cdot (1 - s(X))] \cdot w_B \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot E \geq 0 \cdot S$
- $\frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot w_B \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 2S + S$
- $\frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot w_B \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot 2S$
- $2(s(X) - y^*) \cdot [s(X) \cdot (1 - s(X))] \cdot w_B \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot E \geq 0 \cdot S + S$
- $\frac{\partial \text{Loss}}{\partial s(X)} \cdot [s(X) \cdot (1 - s(X))] \cdot w_B \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \cdot E < 0 \cdot S + S$
- None