Name and Wiscmail:

CS760 Fall 2019 Midterm Exam

If I can't read your handwriting, I can't give you points.

1. Let $\hat{\theta}$ be a trained model and $(x_1, y_1), \dots (x_m, y_m)$ be a separate iid test set that was not used for training. Let ℓ be the loss function. The test set error is an unbiased estimate of the true error: $\mathbb{E}\frac{1}{m}\sum_{i=1}^{m} \ell(x_i, y_i, \hat{\theta}) = \mathbb{E}\ell(x, y, \hat{\theta})$. Explain in English what the two $\mathbb{E}$'s are, respectively.
   0.5 points for each E. Solution: The first $\mathbb{E}$ is over test sets of size m drawn from P^m. The second $\mathbb{E}$ is over a single item (x, y) drawn from P. An answer is correct if it distinguishes test set vs. single item.

2. An experiment consists of flipping the same coin 100 times and writing down the sequence $S = (s_1, s_2, \dots, s_{100})$ where $s_i \in \{H, T\}$ for all i. Suppose $P(s_1 = H) = p$. What is the entropy of the sequence $H(S)$?
   Full points if the answer is correct.
   0.5 points for mentioning the entropy of the coin. Solution: $P(S) = P(s_1, s_2, \dots, s_{100}) = \prod_i P(s_i)$ by independence. $H(S) = -\sum_S P(S) \log P(S) = -\sum_S P(S) \sum_i \log P(s_i) = -\sum_i \sum_S P(S) \log P(s_i)$. By marginalization, $H(S) = -\sum_i \sum_{s_i} P(s_i) \log P(s_i) = \sum_i H(S_i) = 100H(S_1)$, where $H(S_1) = -p \log p - (1-p) \log(1-p)$. All log are base 2. We accept answers if you say it's 100 times the entropy of the coin, or $100H(p)$.

3. Consider a fair 6-sided die with faces 1 to 6. Compute the mutual information between the outcome of a roll and the question "is the outcome a prime number?" (Recall 1 is not a prime.)
   0.5 for the steps.
   0.5 for the final answer.
   Solution: The easiest way is to notice that the question has half-half chance of being yes (2,3,5). It thus provides 1 bit, which is the mutual information.

4. Consider the following 2D regression dataset with five training items: We will build a regression decision tree stump where at each leaf we predict a constant $\hat{y}$ that minimizes the mean squared error of that leaf. Show the decision stump for the question $x_1 \geq 4$.

x_1	12	2	4	10	1
x_2	3	4	4	5	1
y	1	2	3	4	5

   0.5 for each branch.
   Solution: A constant prediction that minimizes MSE is simply the average. Our decision stump is: YES branch predicts (1+3+4)/3=8/3; NO branch predicts (2+5)/2=7/2.

5. Use the same dataset from the previous question. Find the three nearest neighbors of the test point $x = (7, 6)^\top$ in Euclidean distance.
   Full points for correct answer.
   0.5 points for mentioning atleast 2 correct points.
   Solution:

$(x_1 - x_{i1})^2$	25	25	9	9	36
$(x_2 - x_{i2})^2$	9	4	4	1	25
$dist^2$	34	29	13	10	61

   So the three nearest neighbors are: $(10, 5)^\top, (4, 4)^\top, (2, 4)^\top$.

6. In distance-weighted nearest neighbor we define $w_i = \frac{1}{d(x, x_i)^2}$ where x is the query and x_i is the i-th training point. Why don't we use $w_i = d(x, x_i)^2$?

Any explanation that tells us that the latter increases the importance of points farther away from the query. No partial points.

Solution: Because the prediction is of the form $\hat{y} = \frac{\sum_i w_i y_i}{\sum_i w_i}$. The latter weight will put more emphasis on points far away from the query point.

7. Consider online 1NN where you know $x \in \mathbb{R}$ (1D feature), and you know there is a unknown threshold such that any points to the left are labeled negative and to the right (or on the threshold) positive. Training points (x_t, y_t) arrive sequentially and indefinitely for $t = 1, 2, \ldots$. At any moment the original online 1NN algorithm could run 1NN on the training data available so far to make a prediction. Unfortunately, your computer has limited memory: at any moment it can hold at most m earlier training points (not including (x_t, y_t); your algorithm can choose which earlier points to hold). Design an algorithm so that your memory-limited online 1NN performs as well as the original online 1NN, and tell us how small m can be.
Full points for m=2 with explanation.
0.5 points for m=2 and incorrect explanation.
0.5 points for m¿2 with correct explanation.
Solution: Keep two points: the closest negative, positive pairs. 1NN on these two points makes same predictions as online 1NN. So $m = 2$.

8. You have a data set with n items and you want to evaluate neural network's performance. For each of the following methods, how many neural networks do you need to train, and how many training items will each neural network be trained on?

   (a) Split data into 70% training and 30% test.

   (b) K fold cross validation.

   (c) Leave-one-out.

   Full points for completely correct answers.
0.5 points for at least 2 correct pairs.
Solution: 1, $0.7n$; K, $\frac{K-1}{K}n$; $n, n-1$.

9. There are 1000 photos in a data set, 100 of which are of fish. A fish detector applied on the data set returned 120 photos, of which 80 truly are of fish. What is the precision and recall of the fish detector?

   0.5 points for precision and 0.5 points for recall.
Solution: Precision = 80/120=2/3. Recall = 80/100 = 0.8.

10. In the previous question, what is the true positive rate and the false positive rate?
0.5 points for TPR and 0.5 points for FPR.
Solution: True positive rate = recall = 0.8. False positive rate = (120-80)/(1000-100) = 40/900 = 4/90 = 2/45.

11. Coins A,B,C,D have head probability 0.1, 0.2, 0.3, 0.4, respectively. They are stored in a box in such a way that if you reach in the box and grab a coin, your chance of getting A,B,C,D is 0.1, 0.2, 0.3, 0.4, respectively. You reached in the box and grabbed a coin, flipped it, and it turns out heads. Given this observation, what is the probability that your coin is A, B, C, D, respectively?
0.5 points for atleast 2 correct probabilities.
Solution: $P(x|H) = \frac{P(H|x)p(x)}{\sum'_x P(H|x')p(x')}$. It's easier to start with the unnormalized version: $P(H|x = A)P(x = A) = 0.1^2$, $P(H|x = B)P(x = B) = 0.2^2$, $P(H|x = C)P(x = C) = 0.3^2$, $P(H|x = D)P(x = D) = 0.4^2$. Normalizing, $P(x = A|H) = \frac{0.01}{0.01+0.04+0.09+0.16} = \frac{1}{30}$, $P(x = B|H) = \frac{4}{30}$, $P(x = C|H) = \frac{9}{30}$, $P(x = D|H) = \frac{16}{30}$.

12. In a Bernoulli Naive Bayes model with k classes and vocabulary size v, how many parameters are needed? No partial points.
Solution: k-sided die for $p(y)$ which has $k-1$ parameters due to normalization, v coins for $p(x_i \mid y)$ for each class y. The total is $(k-1)+vk$. We also accept $k+vk=(v+1)k$.

13. In linear regression if the design matrix X is $n \times d$ with $n \geq d$, then we can invert $X^\top X$ to obtain the solution $(X^\top X)^{-1}X^\top y$. Is this true? Justify your answer.
0.5 points for false. 0.5 points for justification.
0 points for true.
Solution: It is not true. X could still have rank less than d. For example, two features may be scaled version of each other; or all items are scaled versions of each other. This happens in practice quite often, just because people mess up with the data collection process.

However, we will accept answers in which you assume that this happens with zero probability from a continuous distribution (and thus $X^\top X$ is invertible with probability 1).

14. For logistic regression in 1D with encoding $y \in \{-1, 1\}$ and without offset, we have $p(y \mid x) = \frac{1}{1+e^{-ywx}}$ where the parameter $w \in \mathbb{R}$. Given the training set $\{(x_1=-1, y_1=-1), (x_2=1, y_2=1)\}$, derive the MLE of w.
No partial points.

Solution: Likelihood $p(y_1 \mid x_1)p(y_2 \mid x_2) = \left(\frac{1}{1+e^{-w}}\right)^2$. To maximize it is to minimize $1+e^{-w}$, or to maximize e^w. This leads to $w=\infty$. This is a case where the sigmoid wants to be a hard step function.

15. Given five numbers $\ln 0.01, \ln 0.99, 0, \ln 3, \ln 5$, pass them through softmax to produce a probability vector.
No partial points.
Solution: Softmax exponentiates to produce (0.01, 0.99, 1, 3, 5) then normalizes (0.001, 0.099, 0.1, 0.3, 0.5).

16. A fully-connected feedforward neural network has input $x \in R^d$, a first hidden ReLU layer with h_1 units, a second hidden ReLU layer with h_2 units, and a single sigmoid output unit. How many parameters are there in the neural network? (Don't forget the offset weights)
0.5 points for $dh_1 + h_1h_2 + h2$. That is, 0.5 points for not considering offset weights.
Solution: Each hidden and output unit has an offset parameter. Input to first hidden layer: $(1+d)h_1$. First to second hidden layer: $(1+h_1)h_2$. To output: $1+h_2$. The total is the sum of these.

17. A ReLU unit has input $x \in \mathbb{R}$, weight $w \in \mathbb{R}$, offset $b \in \mathbb{R}$, and outputs $\hat{y} = \max(xw+b, 0)$. Let the loss be $L := \frac{1}{2}(\hat{y}-y)^2$ where $y \in \mathbb{R}$ is the training label. For $x=1, w=1, b=1, y=0$, compute the gradient.
0.5 points for computing $\partial L/\partial w$ or $\partial L/\partial b$.
Solution: Since $xw+b=2>0$, the max can be ignored. $\partial L/\partial w = (\hat{y}-y)x = 2$, $\partial L/\partial b = (\hat{y}-y) = 2$. The gradient is $(2,2)^\top$.

18. Consider $w \in \mathbb{R}$, the objective function to be minimized is the regularized loss $L(w)+\lambda w^2$. With $w_t = 1$, $dL(w_t) = 1$, step size $\eta = 0.1$, $\lambda = 2$, perform one step of gradient descent by computing the value of w_{t+1}.
No partial points.
Solution: $w_{t+1} = w_t - \eta d(L+\lambda w^2) = w_t - \eta dL(w_t) - \eta\lambda 2w_t = 1-0.1-0.1\times2\times2 = 1-0.1-0.4 = 0.5$.

19. Given the input and the convolutional neural network kernel with stride=1 and no padding, compute the output.

200	200	0	0
200	200	0	0
10	200	200	200
10	200	200	200

1	-1
1	-1

No partial points. Solution:

0	400	0
-190	200	0
-380	0	0

20. What should a product manager for deep learning projects do?
    0.5 points for any other reasonable answer.
    Full points for the answer similar to given solution.
    Solution: This was discussed in Andrew Ng's NIPS'16 deep learning tutorial, which is linked on the course webpage. One important job is for the product manager to acquire dev / test data, because that represents the ultimate product need and is a form of indirect product specification.