Machine Learning & Data Mining
Final Exam

Caltech CS/CNS/EE 155
March 14th, 2023 9AM

- **Due 9PM, March 17th, via Gradescope**.
- No extensions (unless emergency), and you cannot use late hours.
- No collaboration, you must answer these questions all on your own.
- Open book. Refer to the course policy for details.
- You are free to ask clarification questions on Piazza.
- This exam is designed to take 3-5 hours to complete. It is not timed.
- **Strongly encouraged** to write the solutions by hand and submit a scanned copy (or use a tablet).
- All questions have short answers. If your answer is very long, you're probably over-thinking it.
- Do not interpret answer box sizes to indicate answer complexity or length, it is mostly due to spacing.
- **Good luck!**

Question Outline

Machine Learning & Data Mining
Final Exam

Caltech CS/CNS/EE 155
March 14th, 2023 9AM

1 Multiple Choice Questions (35 points)

Just specify the correct answer. No need to justify.

Lasso (2 points)

Question A: (True or False) **Every L1-constrained regression problem is equivalent to some L1-regularized regression problem.**
Context: Recall that an L1-constrained regression problem is:

$$\arg\min_{w,b} \sum_{(x,y)\in S} \left(y - (w^T x + b)\right)^2, \quad \text{s.t.} \quad \|w\|_1 \leq C,$$

and that an L1-regularized regression problem is:

$$\arg\min_{w,b} \lambda \|w\|_1 + \sum_{(x,y)\in S} \left(y - (w^T x + b)\right)^2.$$

Solution A: True

Bagging & Bootstrap Sampling (3 points)

Question B: (Multiple Choice) Suppose we generate M bootstraps $S'_1, \ldots, S'_M$ of a training set $S = \{(x_i, y_i)\}_{i=1}^N$. For any specific $(x, y) \in S$, what is the probability that (x, y) does not appear in ANY of $S'_1, \ldots, S'_M$?
(Recall that a bootstrap is a dataset with the same size as S generated by sampling uniformly with replacement from S.)

Option A:

$$\left(1 - \frac{1}{N}\right)^{NM}$$

Option B:

$$\left(1 - \frac{1}{N}\right)^{N} \cdot M$$

Option C:

$$\left(1 - \frac{1}{NM}\right)^{NM}$$

Option D:

$$\left(1 - \left(1 - \frac{1}{NM}\right)^{N}\right)^{M}$$

Solution B: A

Convolutional Filters (4 points)

Consider the three convolutional filters below:

$$K_1 = \begin{bmatrix} 0.0041 & 0.0140 & 0.0216 & 0.0146 & 0.0046 \\ 0.0147 & 0.0481 & 0.0715 & 0.0481 & 0.0147 \\ 0.0217 & 0.0715 & 0.1065 & 0.0715 & 0.0217 \\ 0.0147 & 0.0481 & 0.0715 & 0.0481 & 0.0147 \\ 0.0041 & 0.0147 & 0.0217 & 0.0147 & 0.0046 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 0.0071 & 0.0225 & 0.0335 & 0.0225 & 0.0063 \\ 0.0228 & 0.074 & 0.1109 & 0.0743 & 0.0222 \\ 0.0348 & 0.111 & 0.1654 & 0.1109 & 0.0332 \\ 0.0232 & 0.0743 & 0.111 & 0.0742 & 0.0221 \\ 0.0072 & 0.0222 & 0.0332 & 0.0222 & 0.0061 \end{bmatrix}$$

$$K_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.75 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Question C: (Multiple Choice) If we directly apply K_1, K_2, and K_3 to Figure 1, which filter corresponds to which image in Figure 2?

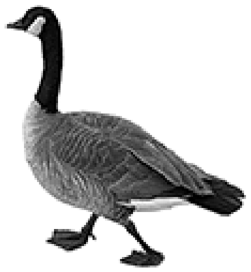


Figure 1: An image of a Canada goose taken by the instructor. All pixel intensities are clipped between 0 and 1 (black=0 and white=1), and the image is 128x128 pixels. The instructor survived this interaction.

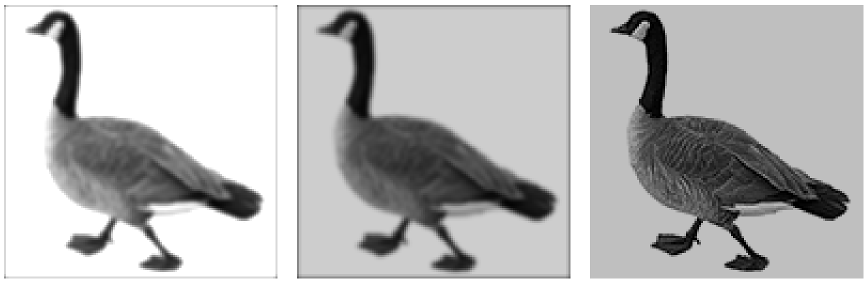


Figure 2: Convolved images. Gray backgrounds are intentional.

Solution C: *(Specify which K for each) Left =* K_2 *Middle =* K_1 *Right =* K_3

Multiclass SVMs (3 points)

Consider the following 4-class multiclass SVM objective:

$$\arg\min_{w,\xi} \frac{1}{2}\|w\|^2 + \frac{C}{N}\sum_{i=1}^{N} \xi_i$$

s.t.

$$\forall i, \forall y' \in \{1,2,3,4\}: \ w_{y_i}^T x_i - w_{y'}^T x_i \geq \mathbf{1}_{[y_i \neq y']} - \xi_i$$

where:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix},$$

and predictions are made via $\arg\max_y w_y^T x$. In other words, each ξ_i can be defined as:

$$\xi_i = \max_{y' \in \{1,2,3,4\}} \left\{ \mathbf{1}_{[y_i \neq y']} - \left(w_{y_i}^T x_i - w_{y'}^T x_i \right) \right\}. \tag{1}$$

Suppose for training data (x_i, y_i) we have $y_i = 1$ and:

$$w_1^T x_i = 1.0,$$

$$w_2^T x_i = -0.8$$

$$w_3^T x_i = 1.5,$$

$$w_4^T x_i = 0.1$$

Question D: (Multiple Choice) Which $\hat{y} \in \{1,2,3,4\}$ is the maximizer of (1)?

Solution D: $\hat{y} =$ 3

Feature Maps (3 points)

Consider the following 3-class multiclass feature map:

$$\phi(x,y) = \begin{bmatrix} \mathbf{1}_{[y=1]}x \\ \mathbf{1}_{[y=2]}x \\ \mathbf{1}_{[y=3]}x \\ -\mathbf{1}_{[y\in\{1,2\}]}x \end{bmatrix}, \tag{2}$$

where predictions are made via $\arg\max_{y\in\{1,2,3\}} w^T\phi(x,y)$. The model score for a class y' is $w^T\phi(x,y')$.

Question E: (Multiple Choice) What is the effect of using (2) versus a "conventional" multiclass feature map of:

$$\phi(x,y) = \begin{bmatrix} \mathbf{1}_{[y=1]}x \\ \mathbf{1}_{[y=2]}x \\ \mathbf{1}_{[y=3]}x \end{bmatrix}. \tag{3}$$

Option A: Compared to (3), in (2) the model scores for Class 1 & 2 are encouraged to be correlated.

Option B: Compared to (3), in (2) the model scores for Class 1 & 2 are encouraged to be anti-correlated.

Option C: Either A or B are possible depending on the specific training data.

Solution E: A

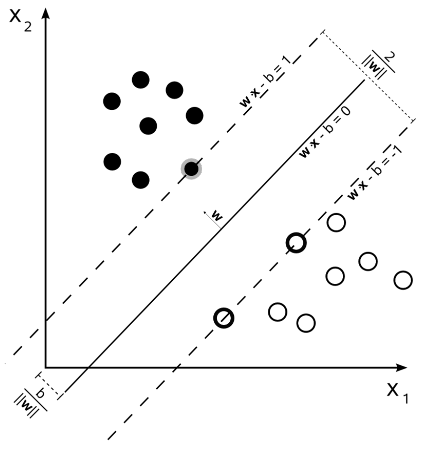
Hard-Margin SVMs (5 points)

A hard margin SVM can be written as:

$$\arg\min_{w,b} \frac{1}{2}\|w\|^2$$

s.t.

$$\forall (x, y) \in S : y(w^T x - b) \geq 1$$



The figure to the right depicts a visualization of the hard-margin SVM for a 2-dimensional example (image source: Wikipedia). The data points that are on the margin (for which the inequality constraint in the training optimization problem is a tight equality) are the "support vectors".

Question F: (True or False, 2pt) **If the data is not linearly separable, then a hard-margin SVM returns** $w = 0$.

Solution F: False

Question G: (Multiple Choice, 3pt) Assume the data is linearly separable. What happens when we remove a data point that is NOT a support vector from the training set, and then retrain the model?

Option A: The solution changes, and the training error is reduced.

Option B: The solution changes, and the training error is increased.

Option C: The solution w and b do not change from before.

Solution G: C

AdaBoost (2 points)

Question H: (True or False) **Each iteration of AdaBoost is guaranteed to reduce the training 0/1 Loss of the aggregate model.**
Context: Recall that the aggregate model of AdaBoost at iteration T is:

$$f_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \in \mathbb{R},$$

and predictions are made via the sign of $f_T(x)$: $h_T(x) = \texttt{sign}(f_T(x))$. Also recall that the Exponential Loss of a data point (x, y) with $y \in \{-1, +1\}$ is $L(y, f_T(x)) = \exp\{-y f_T(x)\}$.

Solution H: False

Tensor Model Training (2 points)

Question I: (True or False) **Every L2-regularized tensor latent-factor regression problem can be optimized via alternating closed-form optimization (i.e., converges to a local optimum).**
Context: Recall that a L2-regularized tensor regression problem can be written as:

$$\arg\min_{U,V,W} \frac{1}{2}\left(\|U\|_{\text{Fro}}^2 + \|V\|_{\text{Fro}}^2 + \|W\|_{\text{Fro}}^2\right) + \frac{1}{2} \sum_{(a,b,c)\in S} \left(Y_{a,b,c} - \langle u_a, v_b, w_c\rangle\right)^2,$$

where u_a, v_b, and w_c correspond to the a-th column of U, the b-th column of V, and the c-th column of W, respectively, and $\|\cdot\|_{\text{Fro}}$ represents the Frobenius norm. The three-way dot product is defined as:

$$\langle u_a, v_b, w_c\rangle = \sum_{k=1}^{K} \langle u_{a,k} v_{b,k}, w_{c,k}.$$

Recall that the alternating closed-form optimization implies that U (likewise V or W) can be solved optimally in closed-form if one holds the other two matrices V and W fixed.

Solution I: True

Bias-Variance Decomposition (2 points)

Let $f_S \in F_S$ denote a set of linear regression models trained on training set S using different learning algorithms (e.g., each f_S is trained using a different regularization strength). Let $\hat{f}_S$ denote the $f_S \in F_S$ that has the lowest bias in the bias-variance decomposition.

Question J: (True or False) **The minimum bias learning algorithm is guaranteed to have the smallest test squared loss.** In other words, the test error of $\hat{f}_S$ is guaranteed to be lower than the test error of any other f_S:

$$\forall f_S \in F_S : \ \mathbb{E}_{(x,y)\sim P(x,y)}\mathbb{E}_S\left[\left(y - \hat{f}_S(x)\right)^2\right] \leq \mathbb{E}_{(x,y)\sim P(x,y)}\mathbb{E}_S\left[(y - f_S(x))^2\right],$$

where $P(x,y)$ denotes the test distribution.

Solution J: False

HMM EM Learning (3 points)

Question K: (Multiple Choice) During EM training in the unsupservised setting, what best describes what happens when you initialize $\Pr(y^j|y^{j-1})$ and $\Pr(x^j|y^j)$ to both be the uniform distribution?

Option A: Nothing special, you just converge to some local optimum.

Option B: You converge to a degenerate local optimum where all the hidden states learn the same thing.

Option C: You get a divide-by-0 error and training crashes.

Context: Recall that an HMM model is specified by:

$$\Pr(x,y) = \Pr(\text{End}|y^M)\prod_{j=1}^{M}\Pr(x^j|y^j)\Pr(y^j|y^{j-1}),$$

the unsupervised learning problem is specified by

$$\arg\max_\Theta \prod_{x\in S}\Pr(x) = \arg\max_\Theta \prod_{x\in S}\sum_{y'}\Pr(x,y'),$$

where S denotes the training set and Θ denotes all the HMM model parameters. The EM algorithm alternates between inferring the distribution of sequences y' for each training example x and using that inferred distribution to estimate better model parameters Θ.

Solution K: B

Non-Negative Matrix Factorization (2 points)

Consider the matrix factorization problem of minimizing squared reconstruction error:

$$\arg\min_{U\in\mathbb{R}^{N\times K}, V\in\mathbb{R}^{M\times K}} \|Y - UV^T\|_{\text{Fro}}^2,$$

where $Y \in \mathbb{R}^{N\times M}$ is the matrix we wish to encode in a low-rank model, U and V are our model components, and $\|\cdot\|_{\text{Fro}}$ is the Frobenius norm of a matrix.

Question L: (True or False) Suppose Y is non-negative. If we enforce U and V to also be non-negative, then we typically need a larger latent dimension K to achieve the same squared reconstruction error compared to allowing U and V to take negative values.

Solution L: True

Decision Trees (2 points)

Question M: (True or False) Given infinite training data (sampled i.i.d. from the test distribution), decision trees can essentially learn arbitrary classifier functions. (Ignore the distinction between countably vs uncountably infinite, this is a high-level conceptual question.)

Solution M: True

Overfitting (2 points)

Question N: (True or False) Given infinite training data (sampled i.i.d. from the test distribution), it is essentially impossible to overfit. (Ignore the distinction between countably vs uncountably infinite, this is a high-level conceptual question.)

Solution N: True

2 Naive Bayes (15 points)

We consider the following Naive Bayes model:

$$\Pr(\texttt{Happy?}, \texttt{Grade}, \texttt{Year}) = \Pr(\texttt{Happy?}) \Pr(\texttt{Grade}|\texttt{Happy?}) \Pr(\texttt{Year}|\texttt{Happy?}).$$

In other words, the y is the Happy? variable, and the two x's are the `Grade` and `Year` variables. We assume that all variables take two values, $\texttt{Happy?} \in \{\text{Yes}, \text{No}\}$, $\texttt{Grade} \in \{\text{A}, \text{C}\}$, and $\texttt{Year} \in \{\text{Freshman}, \text{Senior}\}$. Consider the following training data:

Grade	Year	Happy?
A	Senior	Yes
A	Senior	Yes
A	Senior	No
A	Freshman	Yes
C	Freshman	No
C	Freshman	No
C	Senior	No
C	Senior	Yes

Question 1: (5 points) Fit the model parameters of the Naive Bayes using maximum likelihood with uniform unit pseudocount regularization at $\lambda = 2$. For instance, the maximum likelihood estimate for $\Pr(\texttt{Grade}|\texttt{Happy?})$ is:

$$\Pr(\texttt{Grade} = \text{A}|\texttt{Happy?} = \text{Yes}) = \frac{2(0.5) + \sum_{(x,y)} 1_{[x_{\texttt{Grade}}=A \wedge y=\text{Yes}]}}{2 + \sum_{(x,y)} 1_{[y=Yes]}}.$$

(Check Lecture 12, Slide 10, PDF Page 21). Write out the final probability tables, i.e. $\Pr(\texttt{Grade}|\texttt{Happy?})$, $\Pr(\texttt{Year}|\texttt{Happy?})$ and $\Pr(\texttt{Happy?})$.

Solution :

$$P(x|y) = \frac{\lambda P_{x|y} + N_{x,y}}{\lambda + N_y} \qquad P(y) = \frac{\lambda P_y + N_y}{\lambda + N}$$

Happy	Grade	prior(grade|happy)	N(grade,happy)	N(happy)	P(grade|happy)
yes	A	0.5	3	4	**0.667**
no	A	0.5	1	4	**0.333**
yes	C	0.5	1	4	**0.333**
no	C	0.5	3	4	**0.667**

Happy	Year	prior(year|happy)	N(year,happy)	N(happy)	P(year|happy)
yes	Freshman	0.5	1	4	**0.333**
no	Freshman	0.5	2	4	**0.500**
yes	Senior	0.5	3	4	**0.667**
no	Senior	0.5	2	4	**0.500**

Happy	prior(happy)	N(happy)	N	P(happy)
yes	0.5	4	8	**0.5**
no	0.5	4	8	**0.5**

Question 2: (5 points) Compute Pr(`Year` = Freshman, `Grade` = C, `Happy?` = No) using the trained model from Question 1.

Solution :

$$P(\text{Freshman}, C, \text{No}) = P(\text{No})\,P(C|\text{No})\,P(\text{Freshman}|\text{No}) = \tfrac{1}{2}\cdot\tfrac{2}{3}\cdot\tfrac{1}{2} = \tfrac{1}{6}$$

Question 3: (5 points) Write out the pseudocode for drawing a sample from any trained model. Assume you have repeated access to a function `random`() that returns a uniform random number in $[0, 1]$.
Instructor's Note: If you are not familiar with pseudocode, it's like writing code in a looser sentence format to represent an algorithm. For example, for flipping a coin:

- If random() < 0.5, set result = Head
- Else, set result = Tail
- Return result

Solution :

```
if random() < P(Happy = yes):   happy = yes
else:  happy = no

if random() < P(Year = freshman | Happy = happy):  year = freshman
else:  year = senior
if random() < P(Grade = A | Happy = happy):  grade = A
else:  grade = C
return happy, year, grade
```

Note:
prob lookup can be an array lookup, like
$P(\text{Year} = y \mid \text{Happy} = h)$
$=$ p_year_happy[y][h]

3 Data Transformations (15 points)

We consider the problem of linear regression, where we predict real values given input features x via $w^T x$ using a linear model w (ignoring the bias term). Suppose we want to transform the data points x to a new representation via a transformation matrix: $\tilde{x} = Ax$. For instance, A can be a rescaling of the dimensions of x:

$$A = \begin{bmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_D \end{bmatrix}, \tag{4}$$

where each $a_d > 0$ scales each dimension.

Question 1: (5 points) What is the relationship between w and $\tilde{w}$? In other words, write w as a function of $\tilde{w}$ and A such that $w^T x = \tilde{w}^T \tilde{x}$ holds for all x. Assume that A is a square, invertible matrix.

Solution :

$$w^T x = \tilde{w}^T \tilde{x} = \tilde{w}^T A x \quad \Rightarrow \quad w^T = \tilde{w}^T A \quad \Rightarrow \quad w = A^T \tilde{w}$$

$$\tilde{w}^T = w^T A^{-1} \qquad \tilde{w} = (A^T)^{-1} w$$

Question 2: (5 points) Consider the ridge regression learning objective on the transformed data:

$$\arg\min_{\tilde{w}} \frac{\lambda}{2}\|\tilde{w}\|^2 + \sum_i (y_i - \tilde{w}^T \tilde{x}_i)^2, \tag{5}$$

Rewrite (5) using w, x, and A. In other words, what is the optimization problem that yields the w that corresponds to the $\tilde{w}$ learned in (5) (with the correspondence established in Question 1)? Assume that A is a square, invertible matrix.

Solution :

$$\arg\min_w \frac{\lambda}{2} \| w^T A^{-1} \|^2 + \sum_i (y_i - w^T x_i)^2$$

Question 3: (5 points) Interpret your answers to Question 1 and Question 2 when A is a rescaling transformation such as (4). In other words, how is your answer to Question 2 different from standard ridge regression for w:

$$\arg\min_w \frac{\lambda}{2}\|w\|^2 + \sum_i (y_i - w^T x_i)^2.$$

Solution :

When A is diagonal, $\tilde{w} = (A^T)^{-1} w \Rightarrow \tilde{w}_i = \frac{1}{a_i} w_i$

$$\|\tilde{w}\|^2 = \sum_i \hat{w}_i^2 = \sum_i \frac{1}{a_i^2} w_i^2$$

So we optimize:

$$\arg\min_w \frac{\lambda}{2} \sum_i \frac{1}{a_i^2} w_i^2 + \sum_i (y_i - w^T x_i)^2$$

So the transformation only changes how much effective regularization is applied to each dimension of w.

4 Latent Markov Embedding (15 points)

This problem deals with the song embedding model from Playlist Prediction via Metric Embedding. The paper describes two models, a dual-point model and a single-point model. In the dual-point model, the transition probability from song s to song s' is:

$$\Pr(s'|s) = \frac{e^{-\|U(s')-V(s)\|_2^2}}{Z(s)}. \tag{6}$$

The probability of a playlist $p = \langle p^{[1]}, \dots, p^{[M_p]} \rangle$ is:

$$\Pr(p) = \prod_{i=1}^{M_p} \Pr(p^{[i]}|p^{[i-1]}).$$

Note that there is a special "start" song, which you can ignore for the questions below. The total probability of a dataset S of playlists is then:

$$\Pr(S) = \prod_{p \in S} \Pr(p) = \prod_{p \in S} \prod_{i=1}^{M_p} \frac{e^{-\|U(p^{[i]})-V(p^{[i-1]})\|_2^2}}{Z(p^{[i-1]})}, \quad \text{with: } Z(s) = \sum_{s'} e^{-\|U(s')-V(s))\|_2^2}. \tag{7}$$

In the single-point model, there is only one vector X representing each song (in contrast to the two vectors U and V used in the dual-point model). The transition probability from song s to song s' is:

$$\Pr(s'|s) = \frac{e^{-\|X(s')-X(s)\|_2^2}}{Z(s)}. \tag{8}$$

This results in the total probability of the dataset being:

$$\Pr(S) = \prod_{p \in S} \Pr(p) = \prod_{p \in S} \prod_{i=1}^{M_p} \frac{e^{-\|X(p^{[i]})-X(p^{[i-1]})\|_2^2}}{Z(p^{[i-1]})}, \quad \text{with: } Z(s) = \sum_{s'} e^{-\|X(s')-X(s))\|_2^2}. \tag{9}$$

For the following problems, assume that the optimal choices for U, V, and X are selected (i.e., the ones that maximize $\Pr(S)$).
Hint: Neither question requires a verbose proof to answer. One or two sentences may be sufficient. Refer to the equations and consider the relationships between U, V, and X. The two questions are related.

Question 1: (8 points) Show that the data likelihood $\Pr(S)$ for the dual-point model (7) is never less than the data likelihood for the single-point model (9).

Solution :

The dual-point model can always choose $U = V = X$ to match th performence of the single-point model, so with optimal U, V, X, th dual-point model is always at least as good as th single-point model.

Question 2: (7 points) If (6) is equal to (8) for every pair of songs s and s', what does that imply about the relationship between U, V, and X?

Solution :

$$||U(s') - V(s)||^2 = ||X(s') - X(s)||^2 = ||X(s) - X(s')||^2 = ||V(s') - U(s)||^2$$

So swapping U and V must not change anything.

Thus $U = V = X$

5 Neural Net Backprop Gradient Derivation (15 points)

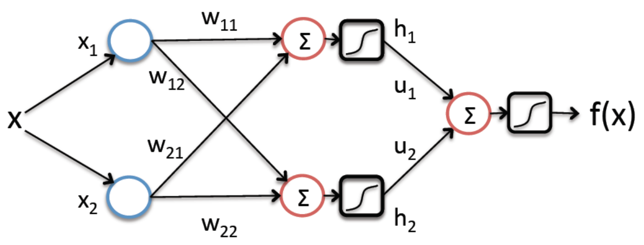


Figure 3: Illustration of Simple Neural Network.

In this question, we will consider the following neural network depicted in Figure 3. There are six parameters in the model, u_1, u_2, w_{11}, w_{21}, w_{12}, and w_{22}.
The network takes in a 2-dimensional input x and outputs a real value $f(x) \in [0, 1]$. The final output $f(x)$ is a weighted combination of two hidden node activations with a sigmoid transfer function:

$$f(x) = \sigma\left(\sum_{i=1}^{2} u_i h_i(x)\right), \tag{10}$$

where:

$$\sigma(s) = \frac{e^s}{1+e^s}, \qquad \text{and} \qquad h_i(x) = \sigma\left(\sum_{j=1}^{2} w_{ji} x_j\right).$$

Question 1: (5 points) For a given training data point (x, y), compute/expand out the gradient of the squared-loss of (x, y) w.r.t. w_{11}:

$$\frac{\partial}{\partial w_{11}} L(y, f(x)) \equiv \frac{\partial}{\partial w_{11}} (y - f(x))^2 .$$

Your answer should be in terms of x, y, $f(x)$, $\sigma()$, $h(x)$, u, w with appropriate subscripts.
Hint: write the formula using the chain rule and use the following definition of the derivative of $\sigma(s)$:

$$\frac{\partial}{\partial s}\sigma(s) = \sigma(s)(1 - \sigma(s)).$$

Solution :

$$\frac{\partial}{\partial w_{11}} L = \frac{\partial}{\partial w_{11}} (y - f(x))^2 = -2(y - f(x)) \frac{\partial}{\partial w_{11}} f(x) = -2(y - f(x)) \frac{\partial}{\partial w_{11}} \sigma(u_1 h_1(x) + u_2 h_2(x))$$

$$= -2(y - f(x))\, \sigma(u_1 h_1(x) + u_2 h_2(x)) [1 - \sigma(u_1 h_1(x) + u_2 h_2(x))] \frac{\partial}{\partial w_{11}} (u_1 h_1(x) + u_2 h_2(x))$$

$$= -2(y - f(x))\, f(x)(1 - f(x)) \frac{\partial}{\partial w_{11}} u_1 h_1(x)$$

$$= -2(y - f(x))\, f(x)(1 - f(x))\, u_1 \frac{\partial}{\partial w_{11}} \sigma(w_{11} x_1 + w_{21} x_2)$$

$$= -2(y - f(x))\, f(x)(1 - f(x))\, u_1\, \sigma(w_{11} x_1 + w_{21} x_2)[1 - \sigma(w_{11} x_1 + w_{21} x_2)] \frac{\partial}{\partial w_{11}} (w_{11} x_1 + w_{21} x_2)$$

$$= \underline{-2(y - f(x))\ f(x)(1 - f(x))\ u_1\ h_1(x)(1 - h_1(x))\ x_1}$$

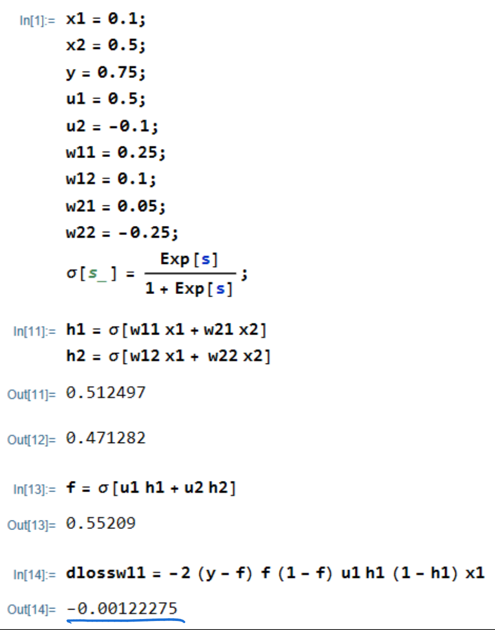
Question 2: (5 points) Find $\frac{\partial}{\partial w_{11}} L(y, f(x))$ where:

$$(x, y) = ((0.1, 0.5), 0.75)$$

$$(u_1, u_2) = (0.5, -0.1)$$

$$(w_{11}, w_{12}, w_{21}, w_{22}) = (0.25, 0.1, 0.05, -0.25)$$

Solution :



```
In[1]:= x1 = 0.1;
        x2 = 0.5;
        y = 0.75;
        u1 = 0.5;
        u2 = -0.1;
        w11 = 0.25;
        w12 = 0.1;
        w21 = 0.05;
        w22 = -0.25;
                  Exp[s]
        σ[s_] = ────────── ;
                1 + Exp[s]

In[11]:= h1 = σ[w11 x1 + w21 x2]
         h2 = σ[w12 x1 + w22 x2]

Out[11]= 0.512497

Out[12]= 0.471282

In[13]:= f = σ[u1 h1 + u2 h2]

Out[13]= 0.55209

In[14]:= dlossw11 = -2 (y - f) f (1 - f) u1 h1 (1 - h1) x1

Out[14]= -0.00122275
```

Question 3: (5 points) Use your derivations from the previous two questions to identify which term in the gradient derivation results in the vanishing gradient problem, and briefly explain how the problem is exacerbated in neural networks with more layers.

Solution :
The terms $f(x)(1-f(x))$ and $h_1(x)(1-h_1(x))$ cause the gradient to vanish when $f(x)$ or $h_1(x)$ saturate (get close to 0 or 1, which makes $f(x)$ or $1-f(x)$ very small).

With more layers, there are more of these terms being multiplied together, so the gradients for early layers become very small. If each layer saturates to $S \approx 0$, then the first layer in an N layer network has its gradient multiplied by S^N.

6 Introspection (5 points)

Question A: (2 points) In at least 5 sentences, explain the importance of following best practices when using ML methods. Give one example of such "best practice", and explain the risk or problem it addresses.

Solution A:

An important "best practice" in ML is using an uncontaminated validation set (such that no training uses any data from it). This gives us a way to estimate how a model will perform on real data, which is essential in cases where the actual deployment is high-stakes. It also allows us to detect overfitting during training, so we can employ early stopping or determine the best amount of regularization to use. Without this, we wouldn't be able to use high-capacity models without the risk that they will just memorize the training data instead of picking up on patterns.

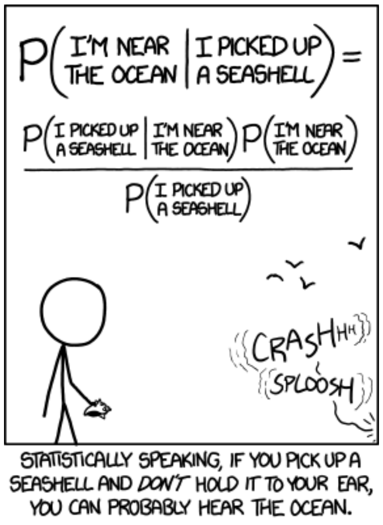
Question B: (2 points) Throughout the course, our lecturers mentioned several potential ethical concerns with ML/AI systems. In at least 5 sentences, describe one such concern and describe how it might be addressed. It does not have to be a specific topic mentioned during lecture.

Solution B:

There is a serious concern with the potential of using uninterpretable AI to make important decisions. We have strict laws about what kinds of reasoning can be used to approve/deny a loan, detain a suspect of a crime, etc. If current AI models, which can make complex extrapolations from input data, are used for such tasks, they could easily run the risk of discriminating based on race/sex/age/etc, even if that information is not explicitly present in the data. For a related example, see the de-anonymization of the Netflix Prize dataset. As deep models can extract increasingly exotic features from data, there is no robust way of interrogating why a model makes a certain classification, so we can't guard against this potential bias without more interpretable models.

Question C: (1 point) Describe your favorite topic covered in the course and why.

Solution C:
Log loss (in all the different places it showed up), because after taking information theory last year I found the connections between it and ML very beautiful.


P(I'M NEAR THE OCEAN | I PICKED UP A SEASHELL) =
P(I PICKED UP A SEASHELL | I'M NEAR THE OCEAN) P(I'M NEAR THE OCEAN)
P(I PICKED UP A SEASHELL)
CRASHHH
SPLOOSH
STATISTICALLY SPEAKING, IF YOU PICK UP A SEASHELL AND DON'T HOLD IT TO YOUR EAR, YOU CAN PROBABLY HEAR THE OCEAN.

(a) xkcd 1236: Seashell

THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

(b) xkcd 1838: Machine Learning

DESPITE OUR GREAT RESEARCH RESULTS, SOME HAVE QUESTIONED OUR AI-BASED METHODOLOGY. BUT WE TRAINED A CLASSIFIER ON A COLLECTION OF GOOD AND BAD METHODOLOGY SECTIONS, AND IT SAYS OURS IS FINE.

(c) xkcd 2451: AI Methodology

Figure 4: Congratulations! Here are some relevant xkcd comics. Shared under CC BY-NC 2.5