ECS 171 Discussion 9

HW3 review and Midterm review
Claudio Spiess and Ziwen Kan

Announcements

- Ziwen is sick, I’m doing today’s discussion and his extra OH today, 4-5 pm 55 Kemper
- Extra OHs for Midterm Review:
    - Ziwen Kan: Wed, Feb 5th 9 AM - 10 AM in Kemper 55 (depending on health)
    - Claudio Spiess: Wed, Feb 5th 11 AM - 12 PM on Zoom
    - Trevor Chan: Thursday, Feb 6th 11 AM - 12 PM on Google Meet
- OHs this week:
    - Dr. Simmons: 2-3pm Thu 3052 Kemper 53 Kemper
    - Ziwen: 1-2pm Wed (swapped with me)
    - Claudio: today, 4-5 pm 55 Kemper
- HW1 & Activity 1 grades are out
- Discussion 7 forward pass example recording in media gallery

Today

- HW3 review
- Start of midterm review

Jupyter Notebook …

ECS171 Winter 2024
Midterm Study Guide

Cheat sheet

Sigmoid and derivative of sigmoid:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$\sigma'(x) = \frac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x))$$

ReLU and Derivative for ReLU:

$$f(x) = max(0,x) = \begin{cases} x & if\ x > 0, \\ 0 & otherwise. \end{cases}$$

$$f'(x) = \begin{cases} 1 & if\ x > 0, \\ 0 & otherwise, \end{cases}$$

Gradient descent weight update:

$$W^{t+1} = W^{t} - \lambda \cdot \nabla J(W^{t})$$

Errors:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Variance:

$$Var = \sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}$$

Chain Rule:

$$if\ h(x) = f\big(g(x)\big)$$

$$h'(x) = f'(g(x)) \cdot g'(x)$$

Performance Metrics:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall\ (True\ Positive\ Rate) = \frac{TP}{TP+FN}$$

$$F_1 score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

$$(True\ Negative\ Rate)\ TNR = \frac{TN}{TN+FP}$$

$$(False\ Positive\ Rate)\ FPR = \frac{FP}{TN+FP}$$

Q1:

In machine learning, what is the dropout technique, and how does it help prevent overfitting in neural networks?

Q1: In machine learning, what is the dropout technique, and how does it help prevent overfitting in neural networks?

- Regularization technique
- Random neurons are turned off with probability p
- Intuition: Forces network to learn different “pathways”
- More robust network, less likely to become overfit

Q2: Given the hours of exercising per week measured in hours …

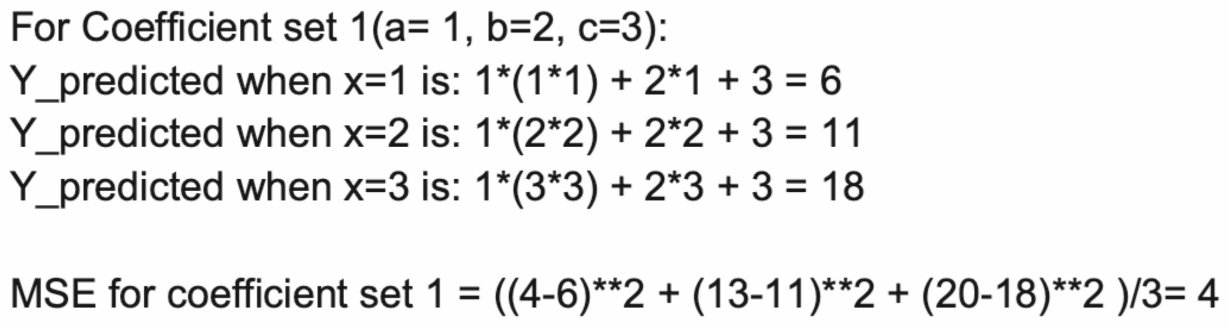
x_1	x_2	y	$\hat{y}$	$(y - \hat{y})^2$
0.0	4.0	0.70	0.68	0.0004
3.0	10.0	0.95	0.95	0
2.0	3.0	0.60	0.51	0.0081
5.0	1.0	0.15	0.22	0.0049
8.0	5.5	0.25	0.385	0.0182
12.0	7.5	0.23	0.325	0.0009

- Fill out y hat + squared error, compute SSEs
- SSE_train = sum of all (y − ŷ) = 0.041
- SSE_test = sum of all (y − ŷ)"= 0.0324
- Variance = 0.041 - 0.0324 = 0.0086
- Lower error and variance -> the model is better fit compared to the base case.

metrics for the base case:
SSE_train = 0.050
SSE_test = 0.049
Variance = 0.01

Q3: Find the coefficients of a polynomial with degree 2 which gives the lowest mean square error

- Calculate y_predicted for all 3 given values of x, for each set of coefficients
- Calculate MSE -> set 1 has lowest MSE



For Coefficient set 1(a= 1, b=2, c=3):
Y_predicted when x=1 is: 1*(1*1) + 2*1 + 3 = 6
Y_predicted when x=2 is: 1*(2*2) + 2*2 + 3 = 11
Y_predicted when x=3 is: 1*(3*3) + 2*3 + 3 = 18

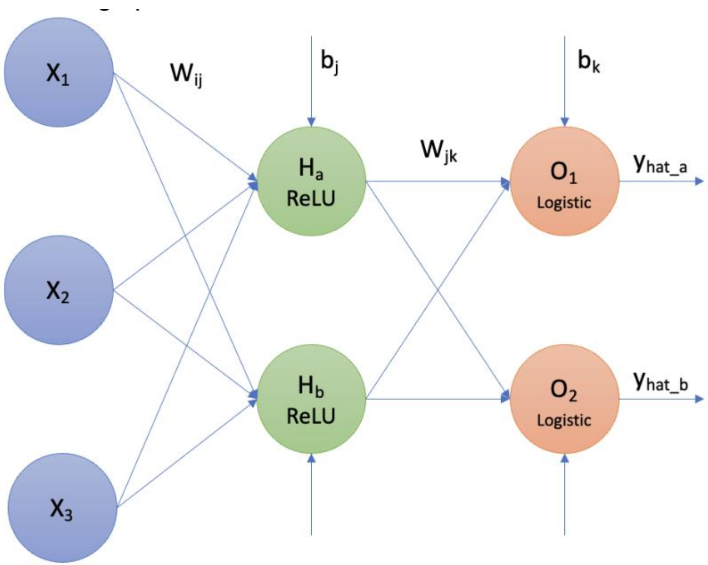
MSE for coefficient set 1 = ((4-6)**2 + (13-11)**2 + (20-18)**2)/3= 4

Q4:

If we designed this neural network with ReLU add after each hidden neural as activation function, and logistic(sigmoid) add in the end before we get the predicted value(y_hat), use the table below to calculate that which class does x1 = 7, x2 = 10, and x3 = 9 belongs. (Assume that as the result of one output has a value over a threshold τ = 0.9, it will be classified into that class.)

W_1a = 0.4	W_a1=0.2	b_a=0.7
W_1b =-0.7	W_a2=0.5	b_b=0.4
W_2a=0.6	W_b1=-0.1	b_1=0.9
W_2b=-0.5	W_b2=0.6	b_2=-0.8
W_3a=0.3		
W_3b=0.7		

Q4:



- H_a = 7*0.4+10*0.6+9*0.3+0.7=12.2
- H_b = 7*-0.7+10*-0.5+9*0.7+0.4=-3.2
- O_1 = max(0,12.2)*0.2+max(0,-3.2)*-0.1+0.9=3.34
- Y_hata = σ(O_1) = 1/(1+e^(-3.34)) = 0.9658
- O_2 = max(0,12.2)*0.5+max(0,-3.2)*0.6+(-0.8)=5.3
- Y_hatb = σ(O_2) = 1/(1+e^(-5.3)) = 0.9950
- Therefore, this data point should belong to both category a and b

Q5: if the data points mentioned in Q7 have y_a= 1 and y_b = 0, update weight w_a1, w_a2, w_1a, and w_3a, consider the learning rate η =0.2. Assume we use SSE for the loss of this question.

$$\frac{\partial \text{error}}{\partial w_{a1}} = \frac{\partial \text{error}}{\partial \hat{y}_a} \cdot \frac{\partial \hat{y}_a}{\partial O_1} \cdot \frac{\partial O_1}{\partial w_{a1}}$$

$$= -(y_a - \hat{y}_a) \cdot \hat{y}_a(1 - \hat{y}_a) \cdot \max(0, H_a)$$

$$= -(1 - 0.9658) \cdot 0.9658 \cdot (1 - 0.9658) \cdot \max(0, 12.2)$$

$$= -0.01378$$

$$W_{a1}^{\text{new}} = 0.2 - 0.2 \times (-0.01378) = 0.2027$$

Q6: Similarly, update all the biases.

- Same procedure, calc derivative of error w.r.t. bias

Q7: In logistic regression, what is the hypothesis function and what does the predicted output of the hypothesis function represent, given an input data point x(1)?

- The hypothesis function is a sigmoid function of the linear combination of weights and input variables
- Predicted output is the integral of the probability density function (sigmoid)
- Represents the probability that a given input x(1) belongs to a category

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}} \qquad t = \beta_0 + \beta_1 x$$

Q8: Joshua claims that in Machine Learning, most of the data are used in testing because accuracy in predicting unseen data is more important than the accuracy of the seen data in the training set. Do you agree with this claim? Justify your answer.

- No, because if the model doesn’t work well with the current training set, there’s no guarantee that it will work even better for the unseen testing set.

Q9: Can OLS method be used to train a logistic regression model as a common practice?

- No, because
- Different objectives
- Different error metrics
- Logistic is non-linear, OLS assumes linear relationship

Q10: Show mathematically how to obtain the weight of a linear regression model with attribute X using the OLS method.

$$y = Xw + \epsilon$$

$$\min_{w} \quad ||y - Xw||^2$$

$$\min_{w} \quad (y - Xw)^T(y - Xw)$$

gradient: $(y - Xw)^T(y - Xw) = y^T y - 2w^T X^T y + w^T X^T X w$

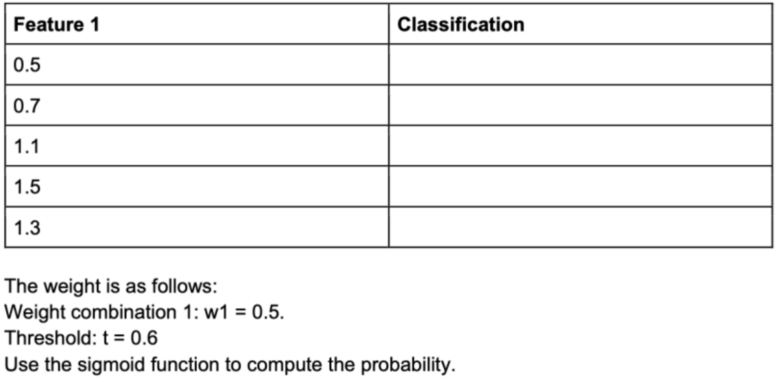
Matrix calculus: $-2X^T y + 2X^T X w = 0$

Solve: $w = (X^T X)^{-1} X^T y$

Q11: What is the number of neurons in the input layer of an ANN if the number of attributes
in the dataset is 3?

- 3

Q12: Classify the feature based on the weight and threshold given below:



Feature 1	Classification
0.5	
0.7	
1.1	
1.5	
1.3	

The weight is as follows:
Weight combination 1: w1 = 0.5.
Threshold: t = 0.6
Use the sigmoid function to compute the probability.

- Observation 1:
- weighted sum = w1 * x1 = 0.5 * 0.5 = 0.25
- sigmoid(weighted sum) = 1 / (1 + exp(-0.25)) = 0.562
- Classification: 0
- …

Q13: See activity 1

Q14: What is the difference between BatchGradient Descent (BGD) and Stochastic GD (SGD)? Why is Stochastic more widely used compared to Batch GD and Newton's method?

- BGD uses the entire dataset to compute gradients in each iteration.
- SGD uses one random data point in each iteration.
- BGD is computationally expensive, especially for large datasets.
- SGD is more efficient as it processes only one data point at a time.
- Newton's method can be very computationally expensive due to Hessian computations.
- Therefore, SGD is more widely used than BGD and Newton's method because it strikes a balance between efficiency and effectiveness.

Q15: In batch gradient descent, if the number of batches is equal to the number of observations in the training dataset, the gradient descent approach is the same as “Stochastic gradient descent ”.

True

Q16: In the context of training artificial neural networks, which of the following best describes the role of gradient descent?

- a) It's a type of activation function applied to the neurons.
- b) It's the process of adding layers to the neural network.
- **c) It's an optimization algorithm used to minimize the error by adjusting the weights.**
- d) It's a method to regularize the network and prevent overfitting.

Q17:
Which of the following statements about L1 and L2 regularization is correct?

A. L1 regularization adds the squared value of the loss to the weight update

B. L2 regularization adds the absolute value of the weight to the loss function

C. L1 regularization adds the absolute value of the weight to the loss function

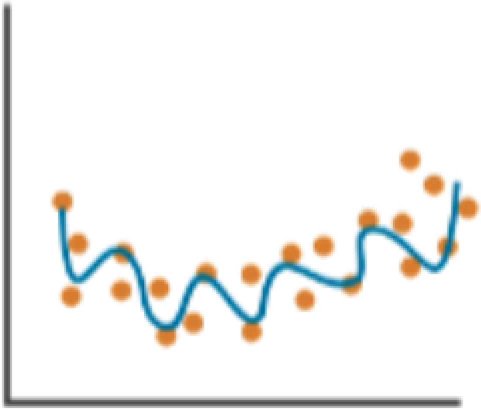
D. L2 regularization adds the squared value of the loss to the weight update

L1: Adds the **absolute value** of the weights to the loss function.

L2: Adds the **squared value** of the weights to the loss function

Q18: Ture or False: This graph on the right is an example of overfitting

True



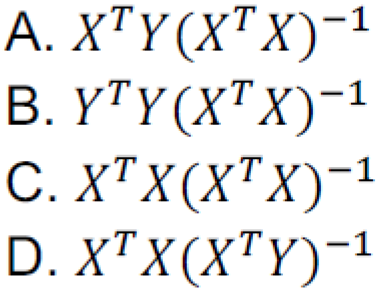
Q19:
Ture or False: Newton's method is a method that completely outperforms gradient descent in any setting because it can always find the minimum loss function value with fewer weight updates.

False

It requires computing and inverting the Hessian matrix, which is **computationally expensive.**

Can **fail** or become unstable if the Hessian is singular or poorly conditioned.

Q20:
Which of the following is the correct formula to find the weight of a linear regression model with the OLS method.



A. $X^TY(X^TX)^{-1}$

B. $Y^TY(X^TX)^{-1}$

C. $X^TX(X^TX)^{-1}$

D. $X^TX(X^TY)^{-1}$

A

Q21:
A company conducted a study to analyze the performance of two machine learning models, Model A and Model B, on a dataset of 500 instances. The confusion matrices for both models are as follows:

Model A:	Predicted Positive	Predicted Negative
Actual Positive	50	10
Actual Negative	20	420

Model B:	Predicted Positive	Predicted Negative
Actual Positive	60	40
Actual Negative	30	370

Using these confusion matrices, calculate and compare the following performance metrics for Model A and Model B: Accuracy, Precision for the positive class, Recall (Sensitivity) for the positive class, F1 score for the positive class.

Based on these metrics, determine which model (A or B) performed better on the dataset

- Model A
- Acc: (TP + TN) / (TP + TN + FP + FN) = (50 + 420) / (50 + 10 + 20 + 420) = 470 / 500 = 0.94
- Precision: TP / (TP + FP) = 50 / (50 + 20) = 50 / 70 = 0.7143
- Recall (Sensitivity): TP / (TP + FN) = 50 / (50 + 10) = 50 / 60 = 0.8333
- F1: 2 * (precision * recall) / (precision + recall) = 2 * (0.8333 * 0.7143) / (0.8333 + 0.7143) = 0.7692