Quiz Submissions - Midterm Exam

Attempt 1

Submission View

Question 1 **1 / 1 point**

Generalization error is defined as the performance on the validation set.

- ( ) True
- ✓ (•) False

Hide Feedback

Generalization error is **approximated** by the performance on the validation set. It is defined as expected loss on unseen data.

Question 2 **0 / 1 point**

Generalization error can be approximated by the performance on the training set if no hyperparameter tuning is performed.

- ✗ (•) True
- ➡ ( ) False

Hide Feedback

Generalization error is the expected loss on unseen data, and is approximated by the performance on the validation/test set.

Question 3 **1 / 1 point**

Consider a model that has 90% accuracy on the training set and 50% accuracy on the validation set. Is this model overfitting, underfitting, or neither?

- ✓ (●) Overfitting
- ( ) Overfitting and underfitting
- ( ) Underfitting
- ( ) Not enough information is given

▼ Hide Feedback

The correct answer is "overfitting". The model has extremely good accuracy on the training set, but it has very poor accuracy on the validation set, which is a clear signal of overfitting.

Question 4 **1 / 1 point**

Which of the following methods can reduce overfitting?

- ✓ [✓] A) Increase dataset size.
- ✓ [ ] B) Decrease dataset size.
- ✓ [ ] C) Use a more complex model.
- ✓ [✓] D) Use a simpler model.
- ✓ [✓] E) Add regularizations.

Question 5 **1 / 1 point**

In terms of the statistical bias-variance tradeoff, a high-bias model is equivalent to a model that is suffering from overfitting.

- ○ True
- ✓ ◉ False

▼ Hide Feedback

A model that has high-variance in the bias-variance tradeoff is equivalent to overfitting. A model that has very high bias could be underfitting.

Question 6 **1 / 1 point**

Increasing number of neighbors in K nearest neighbor classifier may result in:

- ✓ ☑ smoother decision boundry
- ✓ ☐ lower training error
- ✓ ☐ overfitting

▼ Hide Feedback

increasing k makes the classification boundary smoother, and training error might increase. With k too small we may overfit, with k too large may underfit.

Question 7 **1 / 1 point**

Consider following data:

$$X = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 2 \\ 2 & 3 \\ 3 & 3 \\ 3 & 4 \end{pmatrix}, y = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 2 \\ 2 \end{pmatrix}$$

What is p(y=1| x=[2,1]) using a 3-nearest neighbor classifier when using euclidean distance?

- 1
- ✔ 1/3
- 0.5
- 2/3

Question 8 **1 / 1 point**

Consider following data:

$$X = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 2 & 2 \\ 2 & 3 \\ 3 & 2 \\ 3 & 3 \end{pmatrix}, y = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 2 \\ 2 \end{pmatrix}$$

What is p(y=1| x=[2,1]) using a 2-nearest neighbor classifier when using euclidean distance?

- 1
- 0.1
- 0.3
- ✓ 0.5

Hide Feedback

two closest points to [2,1] are [1,1] and [2,2], which are labelled 0 and 1 respectively.

Question 9 **1 / 1 point**

Given a linear model $y = Xw$ where X is the feature matrix, w is the weight vector, y is the predicted label vector, what's the analytical solution for the optimal w?

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

Question 10 **1 / 1 point**

Under what conditions linear regression has a unique solution?

- ✓ [x] Features are linearly independent
- ✓ [x] When using L2 regularization
- ✓ [ ] When number of datapoints is larger than number of features
- ✓ [x] When we have a single feature

Question 11 **1 / 1 point**

Suppose you are optimizing a linear regression function using the closed form approach on a dataset with *m* features and *n* training examples. What is the time complexity of running leave-one-out cross validation on this training set?

- ✓ (●) $O(nm^3 + n^2m^2)$
- ( ) $O(nm^2 + n^2m^2)$
- ( ) $O(n^2m^3 + n^3m^2)$
- ( ) $O(m^3 + nm^2)$

▼ Hide Feedback

The complexity of computing the closed form solution is

$$O(nm^3 + n^2m^2)$$

, as discussed in class. Since we are doing leave-one-out cross-validation, we need to run the entire optimization process *n* times, which gives us the final complexity value.

Question 12 **1 / 1 point**

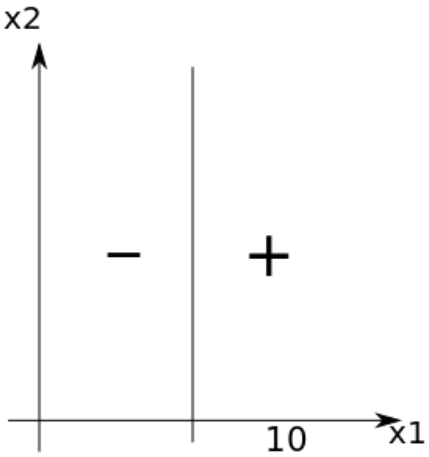
Suppose the parameters of your trained logistic regression classifier

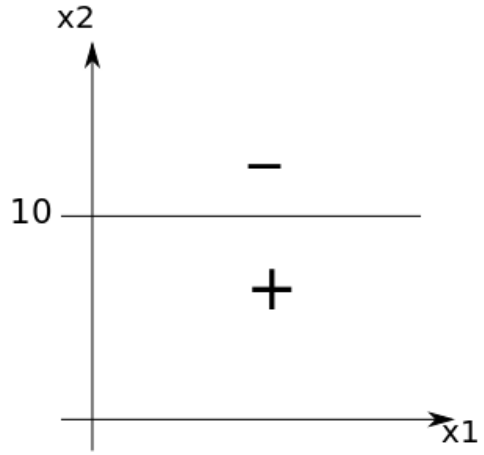
$$h_w = \sigma(w_0 + w_1 x_1 + w_2 x_2)$$

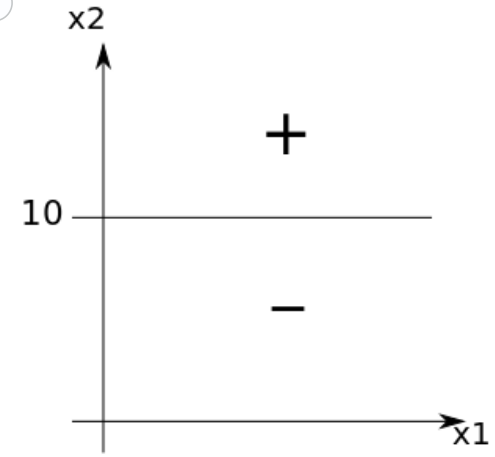
are

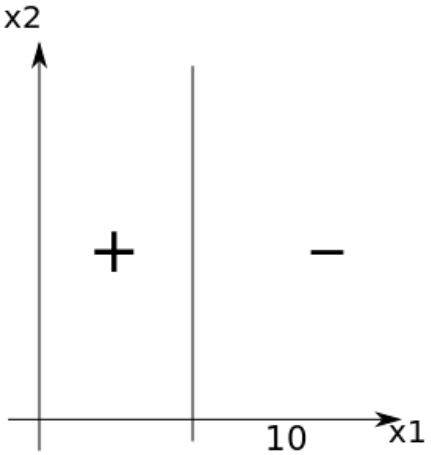
$$w_0 = 10, w_1 = 0, w_2 = -1$$

. What is the decision boundary of your classfier:




x2
10
−
+
x1


x2
10
+
−
x1



Question 13 **1 / 1 point**

Suppose you are given a binary classfication task with m features and n training instances. What is the time complexity of running leave-one-out cross validation on this training set when you optimize the linear classfier by runing gradient descent of I iterations?

- Im^2n^3
- Im^2n
- ✓ Imn^2
- Imn

Question 14 **0 / 1 point**

Select all correct statements

- ✗ ☑ logistic regression is a regression method to estimate class posterior probabilities

✓ ☐ In logistic regression, we model the ratio of class probabilities (a.k.a odds ratio = p/(1-p)) as a linear function

➡ ✓ ☑ The cost function of logistic regression is convex.

✓ ☐ The gradient descent for logistic regression model might converge to a local optimum, and fail to find the global optima.

Question 15 **1 / 1 point**

An employee at a movie production company is prototyping a Naive Bayes model to predict whether a movie will be successful (a binary classification task). So far in the prototype there are three binary features:

- *fresh*, which is 1 if the movie is "certified fresh" on Rotten Tomatoes and 0 otherwise.
- *summer*, which is 1 if the movie was released in the summer and 0 otherwise.
- *rock*, which is 1 if the movie is starring Dwayne "The Rock" Johnson and 0 otherwise.

Suppose the model is trained on the following data:

- *success=1, [fresh=0, summer=0, rock=1]*
- *success=1, [fresh=1, summer=0, rock=1]*
- *success=1, [fresh=1, summer=1, rock=1]*
- *success=0, [fresh=0, summer=1, rock=1]*
- *success=0, [fresh=1, summer=0, rock=0]*

Would this model predict success or failure for a movie with the following attributes: *[fresh=0, summer=0, rock=1]*

○ Impossible to tell (i.e., not enough information given)

✓ ◉ Success

○ Failure

▼ Hide Feedback

the maximum likelihood parameters for this model are:

$$\theta_1 = ⅗, \theta_{1,fresh} = ⅔, \theta_{0,fresh} = ½, \theta_{1,summer} = ⅓, \theta_{0,summer} = ½, \theta_{1,rock} = 1, \theta_{0,rock} = ½$$

And from these we can get that

$$P(success = 1|[fresh = 0, summer = 0, rock = 1]) \propto \theta_1(1 - \theta_{1,fresh})(1 - \theta_{1,summer})\theta_{1,rock} \approx 0.13$$

and that

$$P(success = 0|[fresh = 0, summer = 0, rock = 1]) \propto \theta_0(1 - \theta_{0,fresh})(1 - \theta_{0,summer})\theta_{0,rock} = 0.05$$

Question 16 **1 / 1 point**

Suppose you have two **binary** classification datasets: Dataset A has *m* binary features and Dataset B has *m* continuous (i.e., real-valued) features. You plan to run "Bernoulli Naive Bayes" (i.e., Naive Bayes with binary features) on Dataset A and Gaussian Naive Bayes on Dataset B. Which dataset/model requires more parameters to learn?

- ○ Binary Naive Bayes requires more parameters
- ✓ ◉ Gaussian Naive Bayes requires more parameters
- ○ They require the same number of parameters
- ○ Not enough information

▼ Hide Feedback

Both methods require that you learn the class distribution P(y), so there is no difference there. Since the output is binary, estimating P(y) requires a single parameter (i.e., we need to estimate

$$\theta_y$$

which gives the estimated marginal probability that y is equal to 1).

For Binary Naive Bayes, we also need to estimate P(x|y=1) and P(x|y=0) for each feature. Each of these estimates is a single parameter value, so we need to estimate 2m+1 parameters in total for the binary case.

In the Gaussian Naive Bayes case, we also need to estimate P(x|y=1) and P(x|y=0) for each feature. In this case, we estimate P(x|y=k) as a Gaussian and we need to learn the mean and variance for each feature, which requires 2 parameters to learn. Thus in total we have (2 classes)*(2 parameters to learn the conditional distribution of each feature for each class)*(m features)+(1 parameter to learn the marginal likelihood of the target class)=4m+1 parameters.

Question 17 **1 / 1 point**

Suppose we are giving the following training points for a classification task:

$$X = \begin{pmatrix} 0 & 2 \\ 1 & 0 \\ 1 & 0.5 \\ 3 & 1.5 \end{pmatrix}, y = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

What would be the relative ranking of the four training points in terms of the log-likelihood assigned by the Gaussian Naive Bayes model?

- 1<2<3=4
- 1>2>3>4
- ✓ 1=2=3=4
- 1<2<3<4

Hide Feedback

he parameter values are:

$$\mu_{:,1} = [0.5, 1], \sigma^2_{:,1} = [0.5, 2]; \quad \mu_{:,0} = [2, 1], \sigma^2_{:,0} = [2, .5]$$

Using this, we can see that all the points should have the exact same likelihood.

Question 18 **1 / 1 point**

Suppose you have a multi-class classification dataset with **5 possible classes and 10 binary features.** How many parameters do you need to learn to fit a Naive Bayes model to this dataset? Note that this dataset is multi-class, meaning that the domain of target *y* value is a discrete set with four possible values.

- 123
- ✓ 54
- 105
- 50

Hide Feedback

We need to learn the class distribution P(y) and the conditional distribution of the features given each class. Since there are 5 classes, P(y) requires 4 parameters to learn, as specifying the marginal probability of three of classes determines the probability of the fourth class. (Note, for instance, that in the binary classification case we only needed 1 feature to represent P(y) when there was two classes).

In addition, to estimate P(x|y=k) for each class, we need 10 parameters. Thus, in total, there are 4+5*10=54 parameters to learn.

Question 19 **1 / 1 point**

Suppose

$$x_1, x_2, \ldots, x_n$$

are independent samples draw from the following disribution:

$$p(x|\theta) = \theta x^{-\theta-1}$$

, where

$$\theta > 1, x \geq 1$$

. the maximum likelihood estimaton of

$$\theta$$

is

- $\frac{n+1}{\sum_{i=1}^{n} lnx_i}$
- ✔ $\frac{n}{\sum_{i=1}^{n} lnx_i}$
- $\frac{\sum_{i=1}^{n} lnx_i}{n}$
- $\frac{\sum_{i=1}^{n} lnx_i}{n+1}$

Question 20 **1 / 1 point**

Given the following input matrix (*X*), weight vector (*w*), and ground truth target vector (*y*) calculate the gradient of the linear regression model.

$$X = \begin{pmatrix} 4 & 0 \\ 3 & 1 \\ 1 & 3 \end{pmatrix}, \; w = \begin{pmatrix} 4 \\ 8 \end{pmatrix}, \; y = \begin{pmatrix} 26 \\ 21 \\ 5 \end{pmatrix}$$

Answer for blank # 1: -20 ✔(50 %)

Answer for blank # 2: 68 ✔(50 %)

Question 21 **1 / 1 point**

Given two convex functions f and g, select all of the statements that are true.

- ✔ ☑ $f + g$ is convex.
- ✔ ☐ Second derivatives of either f or g could be negative.
- ✔ ☐ $min(f, g)$ is convex.
- ✔ ☑ If g is monotonically increasing then $g(f(x))$ is convex.

Question 22 **1 / 1 point**

By using subgradients we can incorporate non-smooth or non-differentiable functions into our cost functions while still using gradient based methods.

- ✔ ◉ True
- ○ False

Question 23 **1 / 1 point**

If a hard linear SVM is able to achieve 100% accuracy on a training data set, then the perceptron algorithm is guaranteed to achieve 100% training accuracy on that same dataset

- ✔ ◉ True
- ○ False

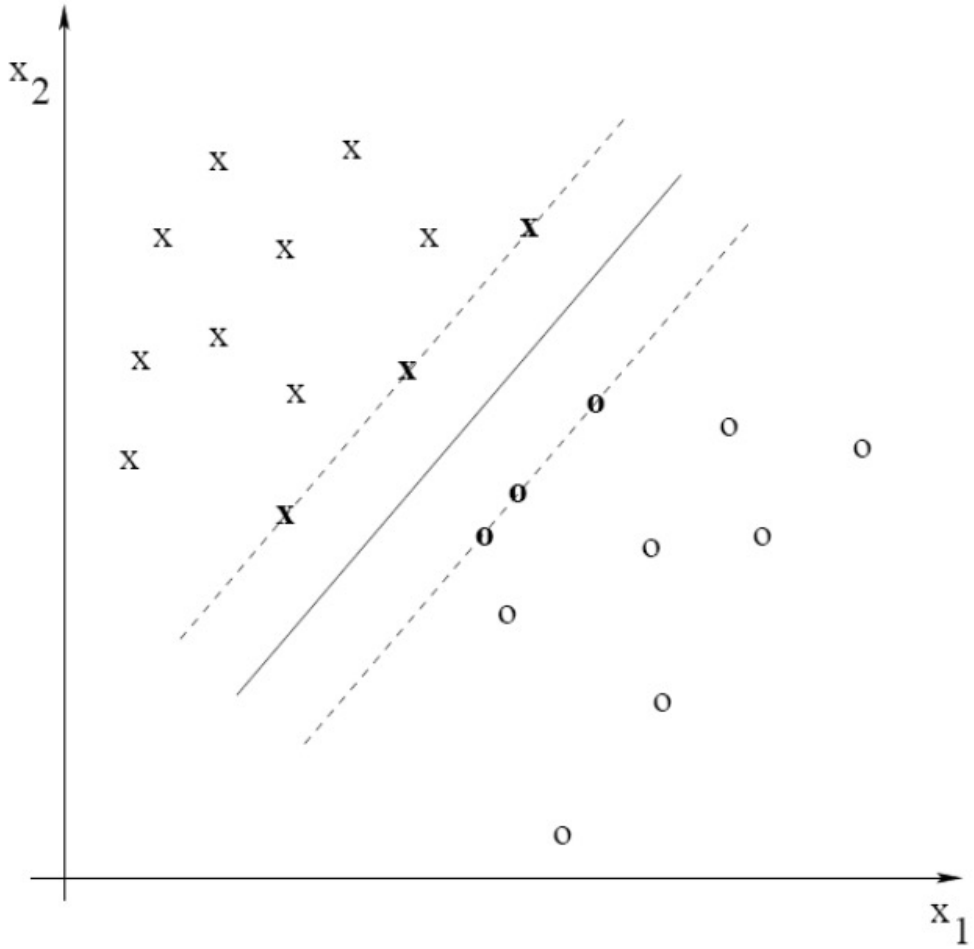
▼ Hide Feedback

The linear SVM will only achieve 100% accuracy if there is a linear decision boundary that perfectly separates the data. And the perceptron convergence theorem guarantees that the Perceptron will be able to successfully separate linearly separable data, so since

we know that there is a linear decision boundary, the Perceptron algorithm will find it.

Question 24 **1 / 1 point**

When we use leave-one-out-cross-validation for the SVM shown in the figure, what is the difference between the maximum and the minimum of the wrongly classified data points?



Answer: 0 ✓

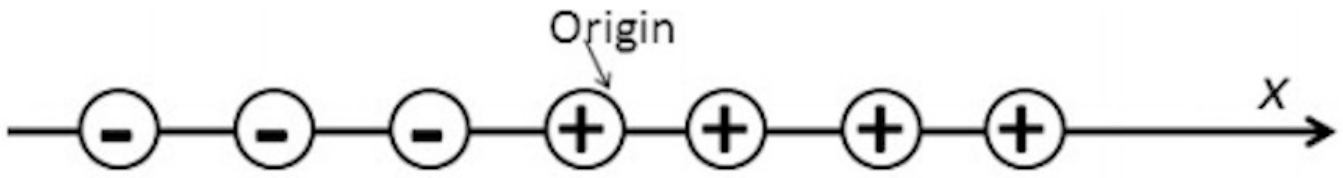
Question 25 **0.5 / 1 point**

Given the following dataset, suppose we want to learn a soft margin linear SVM.

$$\text{argmin}_{\{w,b\}} \ \frac{1}{2}w^t w + C\sum_{i=1}^{m}\epsilon_i$$

$$\text{Subject to : } y_i(w^t x_i + b) \geq 1 - \epsilon_i$$

$$\epsilon_i \geq 0 \ \forall i$$



1. If C = 0, how many support vectors do we have? (Fill in Blank 1)

2. If C is approaching infinity, how many support vectors do we have? (Fill in Blank 2)

(The answers to this question should be between 0 and 7.)

Answer for blank # 1: 0 ✖ (7)

Answer for blank # 2: 2 ✔(50 %)

Decision Trees

Question 26 **1 / 1 point**

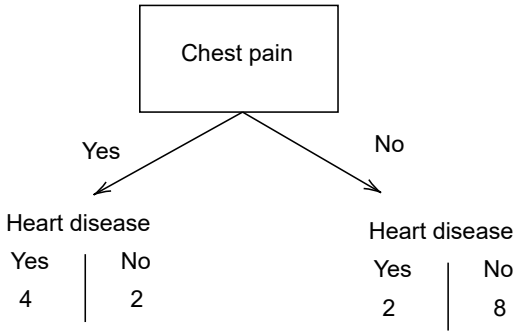
A large decision tree suffers from high variance.

- (●) True

✔

◯ False

Question 27 **1 / 1 point**

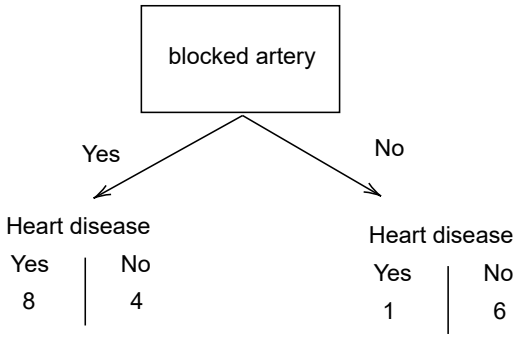


Suppose this one layer decision tree classifies the patients into having heart disease or not based on whether they have experienced chest pain. What is the misclassification cost of this tree?

Answer: 0.25 ✔

Question 28 **0 / 1 point**

Suppose this one layer decision tree classifies the patients into having heart disease or not based on whether they have blocked artery. What is the entropy cost of this tree? {when using basis 2 for the logarithm}



Answer: 0.79796 ✖ **(0.797961024143693, 0.8, 0.79, .8, .79, 0.798)**

▼ Hide Feedback

-(12/19)*((8/12)*np.log2(8/12)+(4/12)*np.log2(4/12))-(7/19)*((1/7)*np.log2(1/7)+(6/7)*np.log2(6/7))

Question 29 **1 / 1 point**

Three attributes have following GINI index; Attribute A: 0.320, Attribute B: 0.364, Attribute C: 0.380. Which attribute is more desirable to choose when building a decision tree?

✔ ◉ A

○ B

○ C

○ Not enough information is given

▼ Hide Feedback

GINI index represents impurity of a child (or leaf) node and less impure attribute is more desired.

Bootstrap

Question 30 — 1 / 1 point

In random forest, which of the following is randomly selected?

- ○ For each node, we randomly select a feature as the split candidate from the full set of features.
- ○ the number of trees in the forest.
- ○ For each tree in the forest, we randomly select a feature as the split candidate from the full set of features.
- ✓ ◉ For each node, we randomly select a subset of features as the split candidates from the full set of features.

Question 31 — 1 / 1 point

Which of the following statements are true?

- ✓ ☐ Logistic regression is not suitable for bagging since all the learners have the same decision boundary.
- ✓ ☑ We can use out of bag samples for validation instead of cross validation for random forest.
- ✓ ☐ We choose random subsamples of input without replacement in bagging.
- ✓ ☐ Bagging is suitable for models with large bias and low variance.

MLP

Question 32 — 1 / 1 point

Which of the following statements are true?

- Sigmoid and hyperbolic tangent activation cause exploding gradients as their pre-activation value moves away from zero.
- ✓ ReLU activation reduces vanishing gradient problem.
- Achieving universal approximation with MLPs requires a large amount of parameters and therefore gradients are more likely to vanish.
- A and B
- None of the above.

Question 33 **1 / 1 point**

Suppose we have a 2-hidden-layer neural network with the following characteristics:

- The input feature dimension is 30, and the inputs are denoted **x**.
- The first hidden layer has dimension 25, and the vector(output) of hidden units is denoted $\mathbf{h}^{(1)}$. The weight matrix at this hidden layer is denoted $\mathbf{W}^{(1)}$ and the bias vector $\mathbf{b}^{(1)}$.
- The second hidden layer has dimension 10, and the vector(output) of hidden units is denoted $\mathbf{h}^{(2)}$. The weight matrix at this hidden layer is denoted $\mathbf{W}^{(2)}$ and the bias vector $\mathbf{b}^{(2)}$.
- The output layer is a binary classification task using the standard cross-entropy loss.
- Sigmoid activations are used in every layer and there is no regularization applied.

Given a single training example, **x,** what would be the derivative of the weights $\mathbf{w}_{\text{out}}$ at the output layer?

$$\frac{\partial Err(\mathbf{w}_{\text{out}})}{\partial \mathbf{w}_{\text{out}}} = \mathbf{x}(\sigma(\mathbf{w}_{\text{out}}^{\top}\mathbf{h}^{(2)} + b_{\text{out}}) - y)$$

✓

$$\frac{\partial Err(\mathbf{w}_{\text{out}})}{\partial \mathbf{w}_{\text{out}}} = \mathbf{h}^{(2)}(\sigma(\mathbf{w}_{\text{out}}^{\top}\mathbf{h}^{(2)} + b_{\text{out}}) - y)$$

$$\frac{\partial Err(\mathbf{w}_{\text{out}})}{\partial \mathbf{w}_{\text{out}}} = \mathbf{x}(\sigma(\mathbf{w}_{\text{out}}^{\top}\sigma(\mathbf{W}^{(2)}\sigma(\mathbf{W}^{(1)}\mathbf{x} + b^{(1)}) + b^{(2)}) + b_{\text{out}}) - y)$$

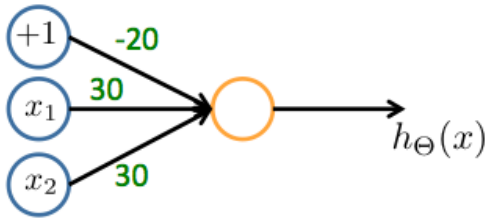
Not enough information is given.

Hide Feedback

Note that most of the details provided are not necessary. Since we are using a sigmoid activation function at all layers and we are performing binary classification with a cross-entropy loss, the final output layer is identical to a logistic regression classifier that takes $\mathbf{h}^{(2)}$ as input. Thus, the derivative for the weight function is identical to the derivative computed in Lecture 5, Slide 5.8, except we replace $\mathbf{x}$ with $\mathbf{h}^{(2)}$.

Question 34 **1 / 1 point**

Given following net where $x_1, x_2 \in 0, 1$ and $h_\theta(x)$ is sigmoid function, this net calculates (approximately) which of the following logical functions?



✓ OR

AND

XOR

NAND

Hide Feedback

We could easily compute that

$$x_1 = x_2 = 1 \quad h > 0 \quad -> 1$$
$$x_1 = 1 \quad x_2 = 0 \quad h > 0 \quad -> 1$$
$$x_1 = 0 \quad x_2 = 1 \quad h > 0 \quad -> 1$$
$$x_1 = 0 \quad x_2 = 0 \quad h < 0 \quad -> 0$$

Gradient Computation

Question 35 **1 / 1 point**

Choose the correct statement about gradient computation from the options given below:

- Numerical differentiation is generally faster and more accurate than automated algorithmic differentiation.
- ✓ Numerical differentiation is useful to compute derivative of black box cost functions.
- Automated algorithmic differentiation is generally used to check the correctness of gradient function from Numerical differentiation.
- All of the above

Question 36 **1 / 1 point**

Consider a function

$$f : \mathbb{R}^P \rightarrow \mathbb{R}^Q \qquad where \quad P \gg Q$$

Suppose we are trying to compute the Jacobian, then reverse mode of automatic differentiation is more efficient.

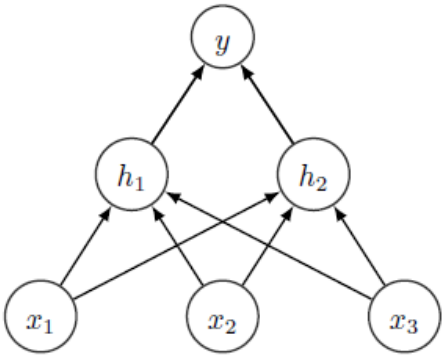
✔ True

False

Question 37 **1 / 1 point**

[Forward and Backward Propagation 1]

The following graph is an MLP with input $x = (x_1, x_2, x_3)$ and hidden layer $h = (h_1, h_2)$ and output y. We ignore bias terms for simplicity.



Let W and V denote the weight matrices connecting input and hidden layer, and hidden layer and output respectively. Both these layers are using sigmoid activation function. What is the function this MLP models?

✔ $y = \sigma(V\sigma(Wx))$

$y = \sigma(W\sigma(Vx))$

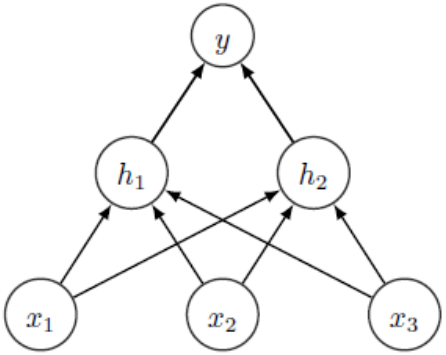
$y = \sigma(VWx)$

$y = \sigma(WVx)$

Question 38 **1 / 1 point**

[Forward and Backward Propagation 2]

The following graph is an MLP with input $x = (x_1, x_2, x_3)$, a hidden layer with two units $h = (h_1, h_2)$, and an output layer of single unit y. Both these layers are using sigmoid activation function. We ignore bias terms for simplicity.



Let W and V denote the weight matrices connecting input and hidden layer, and hidden layer and output respectively. Assume these weights are set as:

$$W = \begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

and

$$V = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

What is the output of this MLP for

$$x = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

- ✔ $\sigma(\sigma(-1))$
- $\sigma(\sigma(0))$
- $\sigma(-1)$

$$\sigma(0)$$

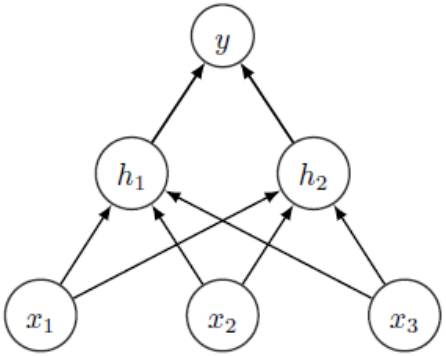
Hide Feedback

$$h = \sigma(Wx) = (\sigma(2), \sigma(-1))$$
$$y = \sigma(Vh) = \sigma(\sigma(-1))$$

Question 39 **1 / 1 point**

[Forward and Backward Propagation 3] (2 points)

The following graph is an MLP with input $x = (x_1, x_2, x_3)$, hidden layer $h = (h_1, h_2)$ and output y. We ignore bias terms for simplicity.



We use sigmoid as activation function for the hidden and the output layer.
Moreover, the loss function is defined as

$$l(y, t) = \frac{1}{2}(y - t)^2$$

, where t denotes the target value for the output unit y.
Let W and V represent the weight matrices connecting input and hidden layer, and hidden layer and output respectively, which are set as:

$$W = \begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}$$

and

$$V = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

What is the gradient

$$\frac{\partial l}{\partial V}$$

for an input

$$x = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

with target value $t = 1$, passed throught this MLP?

- ( ) -0.0286 -0.0937
- (✓) -0.0937 -0.0286
- ( ) -0.0628 -0.0521
- ( ) -0.0521 -0.0628

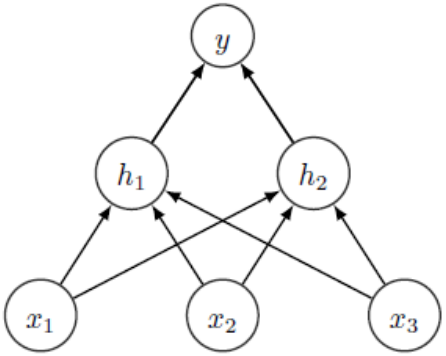
▼ Hide Feedback

$$\frac{\partial l}{\partial V} = \frac{\partial l}{\partial y}\frac{\partial y}{\partial Vh}\frac{\partial Vh}{\partial V} = (y - t) * y(1 - y) * h$$

Question 40 **1 / 1 point**

[Forward and Backward Propagation 4]

The following graph shows the structure of a simple neural network with a single hidden layer. The input layer consists of three dimensions$x = (x_1, x_2, x_3)$, the hidden layer includes two units $h = (h_1, h_2)$, and the output layer includes one unit y. We ignore bias terms for simplicity.



Let W and$V$ represent the weight matrices connecting input and hidden layer, and hidden layer and output respectively. Both of which use sigmoid as their activation function. How can we propagate back the gradient to the weights in the first layer?

$$\frac{\partial l}{\partial W} = \frac{\partial l}{\partial y}\frac{\partial y}{\partial V}\frac{\partial V}{\partial h}\frac{\partial h}{\partial Wx}\frac{\partial Wx}{\partial x}$$

$$\frac{\partial l}{\partial W} = \frac{\partial l}{\partial Vh}\frac{\partial Vh}{\partial h}\frac{\partial h}{\partial Wx}\frac{\partial Wx}{\partial x}$$

✓ $$\frac{\partial l}{\partial W} = \frac{\partial l}{\partial y}\frac{\partial y}{\partial Vh}\frac{\partial Vh}{\partial h}\frac{\partial h}{\partial Wx}\frac{\partial Wx}{\partial W}$$

$$\frac{\partial l}{\partial W} = \frac{\partial l}{\partial y}\frac{\partial y}{\partial Vh}\frac{\partial Vh}{\partial h}\frac{\partial h}{\partial Wx}\frac{\partial Wx}{\partial x}\frac{\partial x}{\partial W}$$

Hide Feedback

$$\frac{\partial l}{\partial W} = \frac{\partial l}{\partial y}\frac{\partial y}{\partial Vh}\frac{\partial Vh}{\partial h}\frac{\partial h}{\partial Wx}\frac{\partial Wx}{\partial W}$$

Attempt Score: 36.5 / 39 - 93.59 %

Overall Grade (highest attempt): 36.5 / 39 - 93.59 %

Done