Final Exam

Final exam will be done individually. Use of partial or entire solutions obtained from others or online is strictly prohibited.

- There will be 7 pages in this exam (including this cover sheet and scratch paper)
- This is a **SEMI-OPEN-BOOK** exam. You can use lecture notes as reference, but do not search the internet for answer.
- Work efficiently and independently.
- You have 150 minutes.
- Good luck!

Question	Topic	Max. score	Score
1	Short Questions	30	
2	Support Vector Machine	10	
3	Boosting Machine	10	
4	Bayesian Network	20	
5	Decision Tree	15	
6	Neural Network and Back-propagation	15	
Total		100	

1. **Short Questions** (30 points)

   (a) (2 pts) **True** or **False**: Bagging algorithms attach weights $\alpha_1, \alpha_2, ...\alpha_N$ to a set of N weak learners. They re-weight the learners and convert them into strong ones. Boosting algorithms draw N sample distributions (usually with replacement) from an original data set for learners to train on.

   (b) (2 pts) **True** or **False**: An infinite depth binary Decision Tree can always achieve 100% training accuracy, provided that no point is mislabelled in the training set.

   (c) (2 pts) **True** or **False**: A neural network with multiple hidden layers and sigmoid nodes can form non-linear decision boundaries.

   (d) (2 pts) **True** or **False**: A random forest is an ensemble learning method that attempts to lower the bias error of decision trees. ↖ dec variance

   (e) (2 pts) Assume that you initialize all weights in a neural net to the same value and you do the same for the bias terms. Which of the following statements is correct.
   (a) This is a good idea since it treats every edge equally.
   (b) This is a bad idea.

   (f) (3 pts) Which of the following statements are true?
   (a) The more training examples, the more accurate the prediction of a k-nearest-neighbor classifier
   (b) k-nearest-neighbors cannot be used for regression. X ← average value
   (c) A k-nearest-neighbor classifier is sensitive to outliers.
   (d) Training a k-nearest-neighbor classifier takes more computational time than applying it / using it for prediction.

   $P(Y=1|X=1) = \frac{P(X=1,Y=1)}{\sum_Y P(X=1,Y)} = \frac{.4}{.3+.4} = \frac{4}{7}$ $3/7$

   (g) (3 pts) Consider the following joint distribution on X and Y, where both random variables take on the values 0, 1: $p(X = 0, Y = 0) = 0.1$, $p(X = 0, Y = 1) = 0.2$, $p(X = 1, Y = 0) = 0.3$, $p(X = 1, Y = 1) = 0.4$. You receive $X = 1$. What is the largest probability of being correct you can achieve when predicting Y in this case?
   (a) $\frac{1}{3}$ **(b)** $\frac{3}{4}$ **(c)** $\frac{1}{7}$ **(d)** 0 **(e)** 1 **(f)** $\frac{2}{3}$ **(g)** $\frac{6}{7}$ **(h)** $\frac{4}{7}$ **(i)** $\frac{3}{7}$ **(j)** $\frac{1}{4}$ **(k)** $\frac{2}{4}$

   $P(Y=0|X=1) = \frac{P(X=1,Y=0)}{\sum_Y P(X=1,Y)} = \frac{.3}{.3+.4}$

   (h) (3 pts) Consider the K-means algorithm. Which of the following assertions is wrong?
   (a) Regardless of the initialization the algorithm converges.
   (b) Regardless of the initialization the algorithm always finds the same clusters.
   (c) If we initialize the K-means algorithm with optimal clusters then it will find in one step optimal representation points.
   (d) If we initialize the K-means algorithm with optimal representation points then it will find in one step optimal clusters.
   → C is first step in algorithm, d would be second step either way

   (i) (3 pts) What strategies can help reduce overfitting in decision trees?
   (a) Pruning
   (b) Enforce a minimum number of samples in leaf nodes
   (c) Make sure each leaf node is one pure class ↖ bad ← lead to large trees
   (d) Enforce a maximum depth for the tree

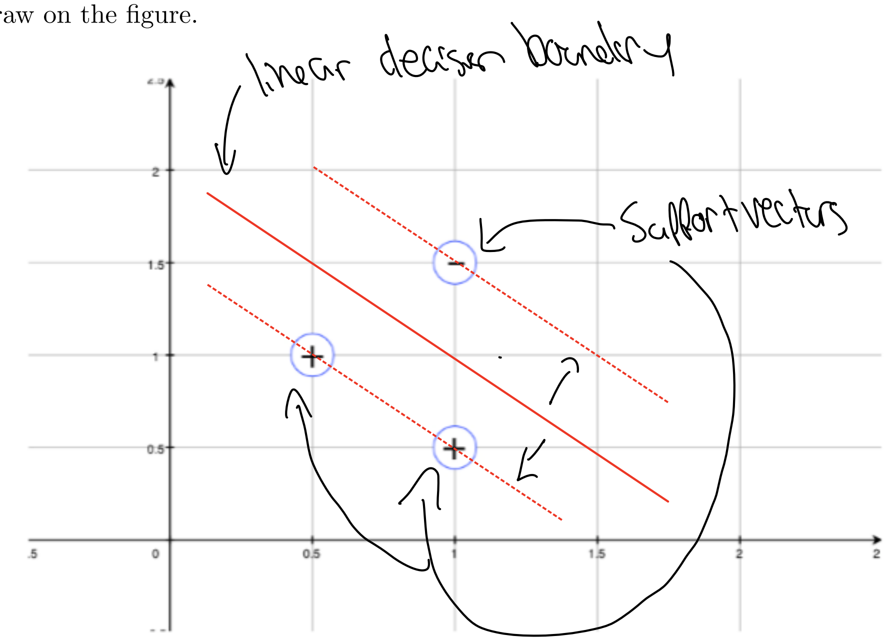
(j) (3 pts) In the setting of EM algorithm, where x_n is the data and z_n is the latent variable, what quantity is called the posterior?
(a) $p(x_n|z_n, \theta)$
(b) $p(x_n, z_n|\theta)$
(c) $p(z_n|x_n, \theta)$

(k) (5 pts) Suppose we clustered a set of N data points using two different clustering algorithms: k-means and Gaussian mixtures. In both cases we obtained 5 clusters and in both cases the centers of the clusters are exactly the same. Can 3 points that are assigned to different clusters in the kmeans solution be assigned to the same cluster in the Gaussian mixture solution? If no, explain. If so, sketch an example or explain in 1-2 sentences.

See Next Page

2. **Support Vector Machine (SVM)** (10 points)

(a) (3 pts) Consider the three linearly separable two-dimensional input vectors in the following figure. Find the linear SVM that optimally separates the classes by maximizing the margin. You only need to draw on the figure.



(b) (2 pts) In the solution for (a), how many *support vectors*?

There are 3 support vectors in the solution for (a).

(c) (5 pts) What are the advantages of SVM?

Advantages include the use of support vectors to maximize the margin around the decision boundary, to create a more generalized model for test sample classification. The addition of slack variables allows for some handling of misclassified values & the dual form of the Loss function can utilize the "kernel trick" to classify nonlinear data.

1. (k) This could be possible due to the fact that the shape of the cluster varies between k-means and GMM, ← even w/ similar mean locations. For example, in 2D, k-means clusters are circular, while GMM can be elliptic. Additionally the soft assignment the GMM produces using probabilities could also lead to variations in test sample classification.

3. **Boosting Machine-Adaboost** (10 points)

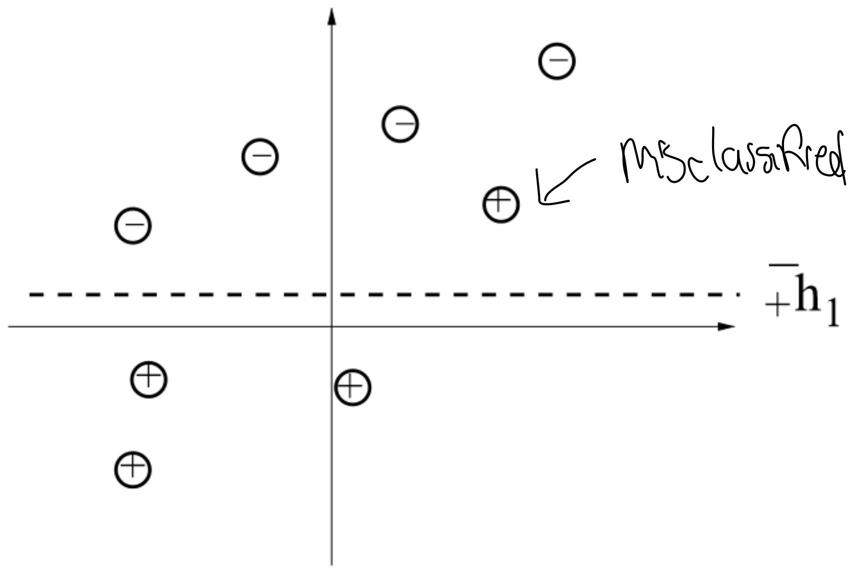


Figure 1: h_1 is chosen at the first iteration of boosting. Points above the h_1 are predicted to be negative while below the h_1 are predicted to be positive.

(a) (5 pts) The above figure shows a dataset of 8 points, equally divided among the two classes (positive and negative). The figure also shows a particular choice of decision line h_1 picked by AdaBoost in the first iteration. What is the weight α_1 that will be assigned to h_1 by AdaBoost? (Initial weights of all the data points are equal, or $\frac{1}{8}$.)

(b) (5 pts) AdaBoost will eventually reach zero training error, regardless of the type of weak classifier it uses, provided enough weak classifiers have been combined. (**True** or **False**, briefly explain)

3a.) $M_1 = \frac{1}{8}$ for all points $\rightarrow \varepsilon_1 = \frac{1}{8}$ (1 misclassification)

$$\alpha_1 = \frac{1}{2}\ln\left(\frac{1-\varepsilon_1}{\varepsilon_1}\right) = \frac{1}{2}\ln\left(\frac{1-\frac{1}{8}}{1/8}\right) = \boxed{0.973}$$

b.) I believe this is true. We should be able to achieve very high accuracy (100% on training) if the classifiers are weak and we use enough of them. As long as the classifiers achieve different misclassifications, then the ensembling while boosting should reach a 0% training error. These classifiers need to have an error $\varepsilon_t < 0.5$ (better than random).

4. **Bayesian Network** (15 points)

(a) (5 pts) Draw a Bayesian network which represents the following joint distribution:

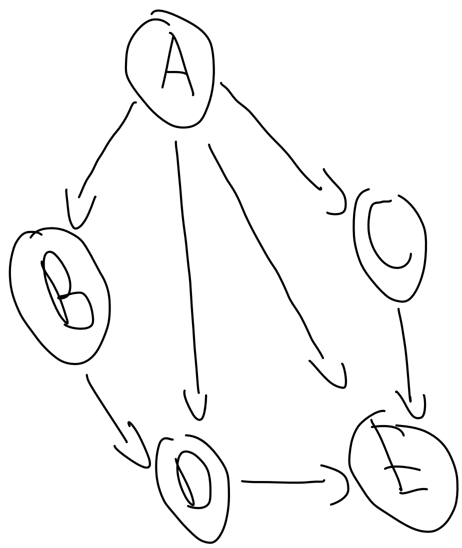
$$P(A, B, C, D, E) = P(A)P(B|A)P(C|A)P(D|A, B)P(E|A, C, D)$$

(b) (5 pts) How many independent parameters are needed to fully specify the joint distribution in (a).

(c) (3 pts) Suppose we do not have any independence assumption over all 5 random variables A, B, C, D, E, write down one possible factorization of $P(A, B, C, D, E)$.

(d) (2 pts) How many independent parameters are needed to fully specify the joint distribution in (c).

a.)



b.) $P(A,B,C,D,E) = P(A)\,P(B|A)\,P(C|A)\,P(D|A,B)\,P(E|A,C,D)$

all R.V. binary → $2^0 + 2^1 + 2^1 + 2^2 + 2^3$

$1 + 2 + 2 + 4 + 8$

$\boxed{17 \text{ parameters}}$

c.) w/o independence, Any Combination

$P(A,B,C,D,E) = \boxed{P(A)P(B)P(C|B,A)P(D|A,C)P(E|A,B,C)}$

d.) all R.V. binary → $2^0 + 2^0 + 2^2 + 2^2 + 2^3 = 1+1+4+4+8$

$= \boxed{18 \text{ parameters}}$

5. **Decision Tree** (15 points) We will use the dataset below to learn a decision tree which predicts if people pass machine learning (Yes or No), based on their previous GPA (High, Medium, or Low) and whether or not they studied.

GPA	Studied	Passed (Y)
L	F	F
L	T	T
M	F	F
M	T	T
H	F	T
H	T	T

For this problem, write your answer using $\log_2$, it may be helpful to note that $\log_2 3 \approx 1.6$.

(a) (2 pts) What is the entropy $H(Y)$?

(b) (3 pts) What is the entropy $H(Y|\text{GPA})$?

(c) (5 pts) What is the entropy $H(Y|\text{Studied})$?

(d) (5 pts) Draw the full decision tree that would be learned for this data set. You do not need to show any calculations.

a.) $H(Y) = -\sum_{i=1}^{2} P(Y=y_i)\log_2(P(Y=y_i))$

$P(Y=T) = 4/6 = 2/3, \; P(Y=F) = 2/6 = 1/3$

$H(Y) = -2/3\log_2(2/3) - 1/3\log_2(1/3) = \boxed{0.918}$

b.) $H(Y|GPA) = -\sum_{j=1}^{3} P(GPA=x_j)\sum_{i=1}^{2} P(Y=y_i|GPA=x_j)\log_2(P(Y=y_i|GPA=x_j))$

$P(GPA=L) = P(GPA=M) = P(GPA=H) = 2/6 = 1/3$

$H(Y|GPA) = -1/3\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) - \frac{1}{3}\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log\left(\frac{1}{2}\right)\right)$

$-1/3\left(1\log_2(1) + 0\log(0)\right)$

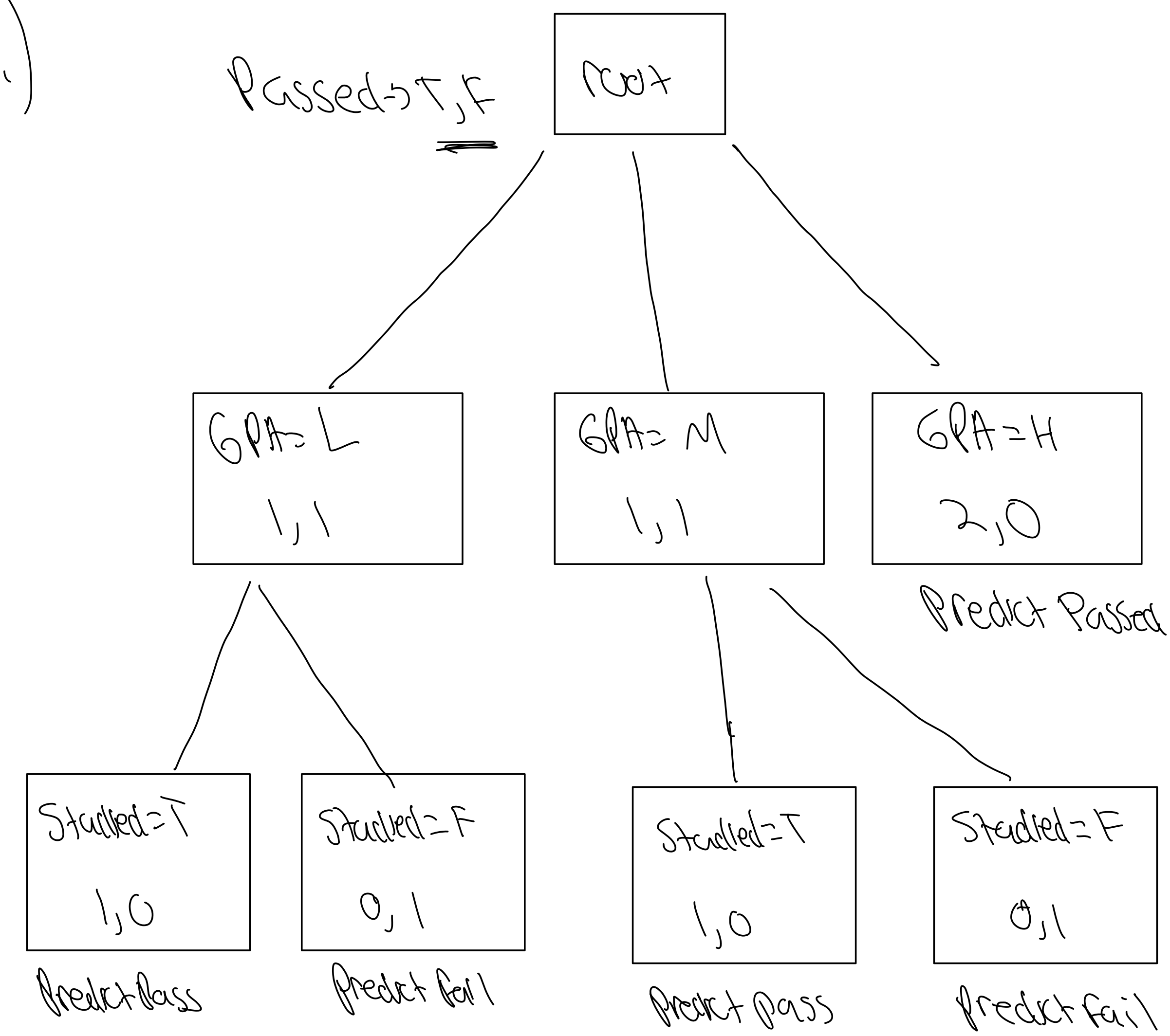
$= -2/3\left(\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) = 2/3 = \boxed{0.667}$

c.) $H(Y|\text{Studied}) \rightarrow P(\text{Studied}=T) = \frac{3}{6} = \frac{1}{2}, \; P(\text{Studied}=F) = \frac{3}{6} = \frac{1}{2}$

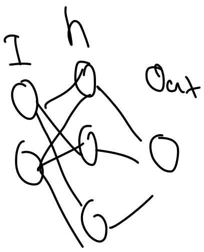
$$H(Y|\text{Studied}) = -\frac{1}{2}\left(1\log_2(1) + 0\log(0)\right) - \frac{1}{2}\left(\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2(2/3)\right)$$

$$= -\frac{1}{2}\left(\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right) = \boxed{0.459}$$

d.)



6. **Neural Network and Back-propagation** (15 points) You want to train your neural network to predict the score of exam based on inputs of how many hours we studied and how many hours we slept the day before. Your neural network consists of an input layer with 2 units (hours studied and hours slept), a hidden layer with 3 units and an output layer with 1 unit (the score of exam). You use the **sigmoid** activation function for the hidden units and **no** activation function for the outputs (or inputs). We use the following notations:



- $\mathbf{x}$ is the training input vector, $\mathbf{y}$ is the true label vector(in this case, the given exam scores), $\hat{\mathbf{y}}$ is the output of your neural network. All vectors are **column vectors**. Note that vector $\mathbf{x}$ has two elements, $\mathbf{y}$ and $\hat{\mathbf{y}}$ has only one element (i.e., scalar).
- Sigmoid function is defined as: $\sigma(x) = \frac{1}{1+e^{-x}}$. $\sigma(.)$ is applied element-wise to a vector. The derivative of sigmoid function is $\sigma'(x) = \frac{d\sigma(x)}{dx} = \sigma(x)(1-\sigma(x))$
- $\mathbf{g}$ is the vector of hidden unit values before the sigmoid activation functions are applied, $\mathbf{h} = \sigma(\mathbf{g})$ is the vector of hidden unit values after they are applied.
- V, b_V are the weight matrix and bias terms that map the input layer to the hidden layer, i.e., $\mathbf{g} = V\mathbf{x} + b_V$.
- W, b_W are the weight matrix and bias terms that map the hidden layer to the output layer, i.e., $\hat{\mathbf{y}} = W\mathbf{h} + b_W$.

(a) (5 pts) What function does this one-layer neural network represents? Write down the function expression for $\hat{\mathbf{y}}$ in terms of input $\mathbf{x}$ and all related weight/bias parameters.

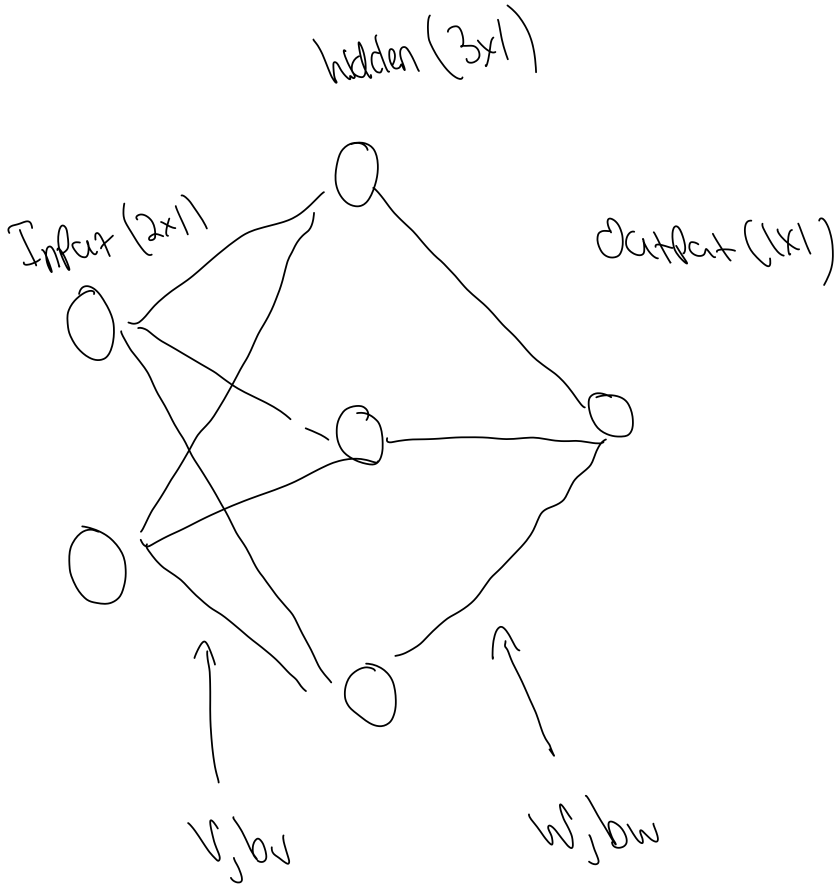
(b) (5 pts) Calculate the number of parameters (weights) in this network. You can leave your answer as an expression. Be sure to account for the bias terms. (Hint: consider the size of the weight matrices (i.e., V and W) and bias terms (i.e., b_V and b_W))

(c) (5 pts) Suppose you train your network with cost function $J = \frac{1}{2}|\mathbf{y} - \hat{\mathbf{y}}|^2$. What is $\frac{\partial J}{\partial W}$? (Hint: $\frac{\partial J}{\partial W}$ is the matrix/vector with the *same dimension* as W, express the gradient $\frac{\partial J}{\partial W}$ in terms of proper vector product using $\hat{\mathbf{y}}, \mathbf{y}$ and $\mathbf{h}$.)

a.) $\hat{y} = Wh + bw$, $h = \sigma(g) = \frac{1}{1+e^{-g}}$, $g = Vx + bv$

$$\boxed{\hat{y} = W\left(\frac{1}{1+e^{-(Vx+bv)}}\right) + bw}$$

b.)



for V → go from 2x1 to 3x1 → 3x2 = 6 parameters

bv → # hidden → 3x1 → 3 parameters

for W → go from 3x1 to 1x1 → 1x3 = 3 parameters

bw → # output → 1x1 → 1 parameters

$6+3+3+1 = \boxed{13 \text{ parameters}}$

(c) $J = \frac{1}{2}|y-\hat{y}|^2 \rightarrow 1\times1$

$$\frac{dJ}{dw} = \frac{dJ}{d\hat{y}} \cdot \frac{d\hat{y}}{dw} \rightarrow \frac{dJ}{d\hat{y}} = -(y-\hat{y}) = (\hat{y}-y)$$

$$\hat{y} = Wh + bw \rightarrow \frac{d\hat{y}}{dw} = h$$

$$\boxed{\frac{dJ}{dw} = (\hat{y}-y) \cdot h^T}$$

$\hat{y}-y$ $\quad 1\times3$

$\rightarrow$ shape is $(1\times1)\times(h^T) = \underline{\underline{(1\times3)}}$

$\hat{w}$ shape

Since $\hat{y} = Wh + bw$

$w \cdot h = 1\times1$

$w = 1\times3$

$h = 3\times1$