CSC 311: Introduction to Machine Learning

Tutorial 12 - Final Exam Review

Harris Chan

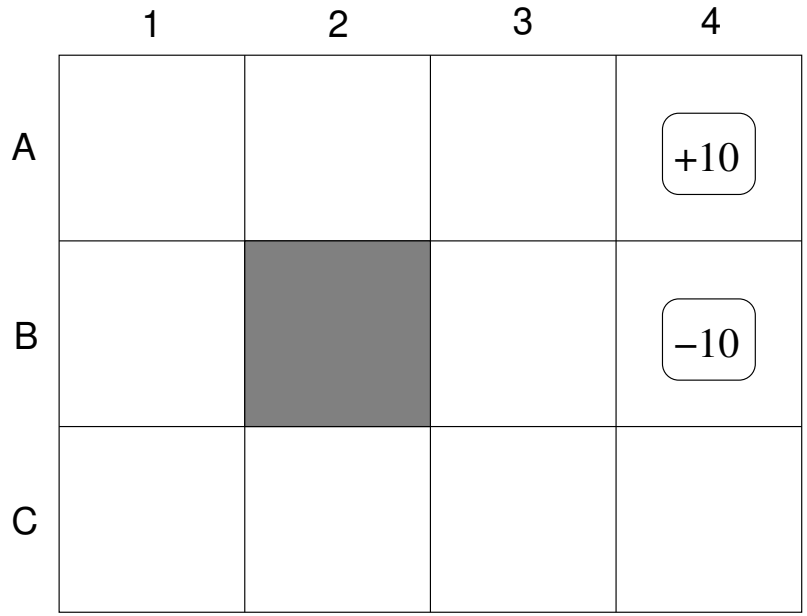
University of Toronto

This tutorial

Cover example questions from several areas:

- Reinforcement Learning
- K-Means / EM
- Principal Component Analysis
- Probabilistic Models
- Support Vector Machines / Ensembling Methods
- Neural Networks

Reinforcement Learning



Consider this familiar navigation task, shown on the left above. You can move in any of four directions (left/right/up/down). If you hit the blob at B2, you remain in the same state. The rewards are +10 for *entering* state A4 and −10 for *entering* B4; these are both absorbing states. The reward for every other state transition is 0.

Reinforcement Learning

(A). Assume that the state transitions are deterministic. Recall that under the simple Q-learning algorithm, the estimated Q values are updated using the following rule:

$$\hat{Q}(s,a) = r + \gamma \max_{a'} \hat{Q}(s',a')$$

Consider applying this algorithm when all the $\hat{Q}$ values are initialized to zero; and $\gamma = 0.9$.
Indicate Q values on the figure after repeatedly cycling through the following episodes:

- A1,A2,A3,B3,B4
- C2,C1,B1,A1,A2,A3,A4
- C4,C3,B3,A3,A4

Reinforcement Learning

	1	2	3	4
A	right: +8.1 down: 0	left: 0 right:+9	left: 0 right: +10 down: +8.1	+10
B	up: +7.29 down: 0		right: -10 up: +9.0 down: 0	-10
C	up: +6.56 right: 0	left: +5.9 right: 0	left: 0 right: 0 up: +8.1	left: +7.29 up: 0

Reinforcement Learning

(B). Assume the robot will now use the policy of always performing the action having the greatest Q value. Indicate this policy on the figure. Is it optimal?

Reinforcement Learning

(B). Assume the robot will now use the policy of always performing the action having the greatest Q value. Indicate this policy on the figure. Is it optimal?

Answer:

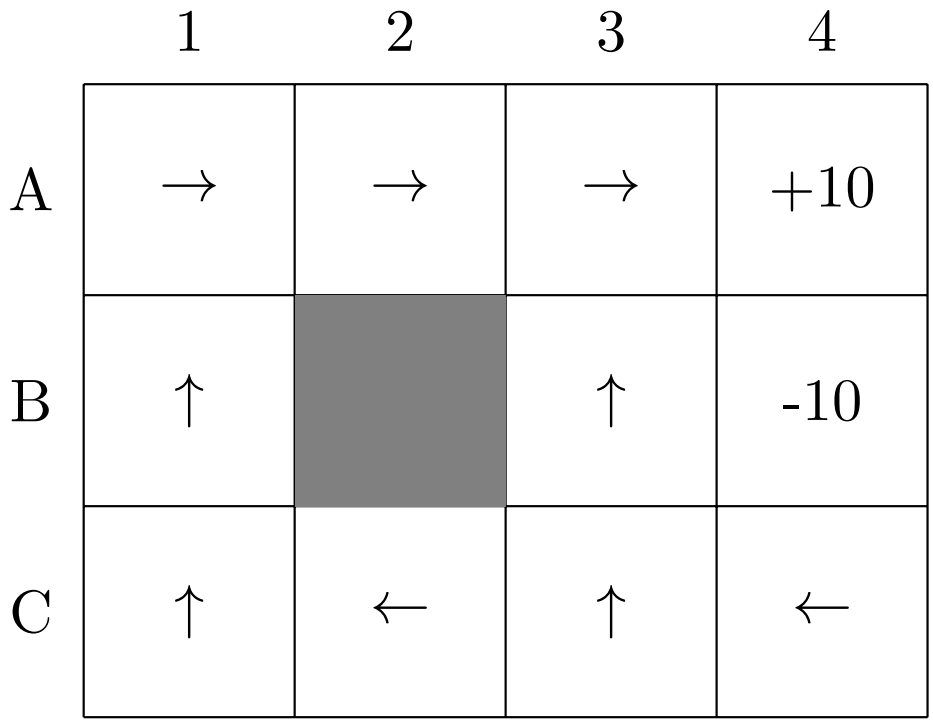


Figure: Learned policy

Reinforcement Learning

(B). Assume the robot will now use the policy of always performing the action having the greatest Q value. Indicate this policy on the figure. Is it optimal?

Answer:

- Optimal must mean shortest path to upper right (since reward worth less the further in future it occurs due to discount gamma).
- Need more exploration, go right at C2.

Reinforcement Learning

(C). We have seen several examples of greedy approaches to learning in this class, such as decision tree learning and boosting. What are the strengths and weaknesses of a greedy action strategy for reinforcement learning?

Reinforcement Learning

(C). We have seen several examples of greedy approaches to learning in this class, such as decision tree learning and boosting. What are the strengths and weaknesses of a greedy action strategy for reinforcement learning?

Answer:

- Strength: Not spending lots of time exploring useless actions; exploit knowledge; Saves computation.
- Weaknesses: May miss/never find optimal policy; sensitive to initial conditions.

Reinforcement Learning

(D) Imagine that this grid world is much larger, and contains multiple obtacles and terminal states with positive and negative rewards. Define some features that could be used to enable generalization in this setting.

Reinforcement Learning

(D) Imagine that this grid world is much larger, and contains multiple obtacles and terminal states with positive and negative rewards. Define some features that could be used to enable generalization in this setting.

Answer:

- We can represent the Q function as a parametric model (e.g. Neural Network) instead of a table, and the same model is used everywhere, that maps a state representation and an action to a value.
- This same model can therefore generalize across states.
- The states can be represented by, for example, the coordinates, distance to terminal states, etc.

K-Means / EM

1. What is the difference between K-Means and Soft K-Means algorithm?

K-Means / EM

1. What is the difference between K-Means and Soft K-Means algorithm?

Answer:

- Hard K-Means assigns a point to 1 particular cluster, whereas Soft K-Means assigns responsibilities (summing to 1) across clusters

K-Means / EM

2. K-means algorithm can be seen as a special case of the EM algorithm. Describe the steps in K-means that correspond to the E and M steps, respectively.

K-Means / EM

2. K-means algorithm can be seen as a special case of the EM algorithm. Describe the steps in K-means that correspond to the E and M steps, respectively.

Answer:

- Assignment step in K-Means is similar to the E-step in EM, computing responsibilities assesment
- Refitting step in K-Means minimizes the cluster distance while M-step in EM maximizes generative likelihood
- K-Means is equivalent to having spherical covariance (shared diagonal) while EM can have arbitrary covariance.

Principal Component Analysis

1. The principal components of a dataset can be found by either minimizing an objective or, equivalently, maximizing a different objective. In words, describe the objective in each case using a single sentence.

Principal Component Analysis

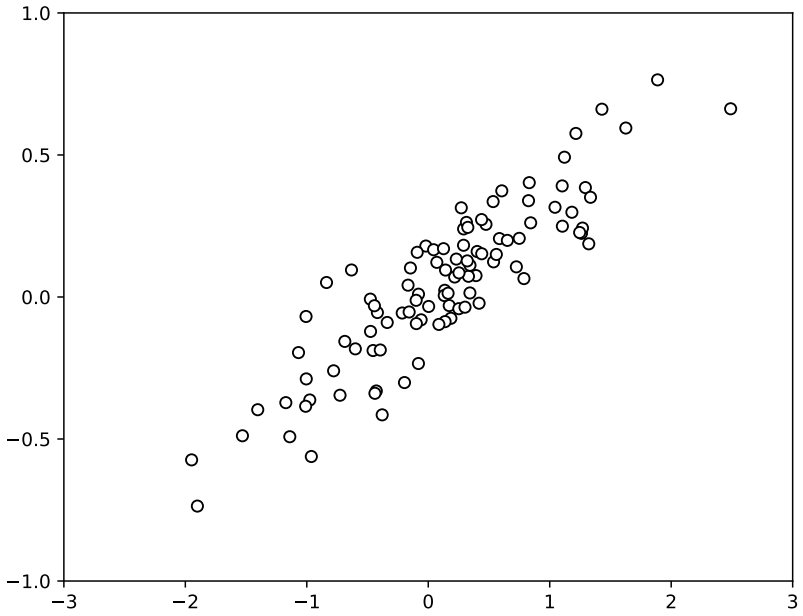
1. The principal components of a dataset can be found by either minimizing an objective or, equivalently, maximizing a different objective. In words, describe the objective in each case using a single sentence.

Answer:

- **Minimizing**: Reconstruction error i.e. the distance between the original point and its projection onto the principal component subspace
- **Maximizing**: Variance between the code vectors i.e. the variance between the coordinate representations of the data in the principal component subspace

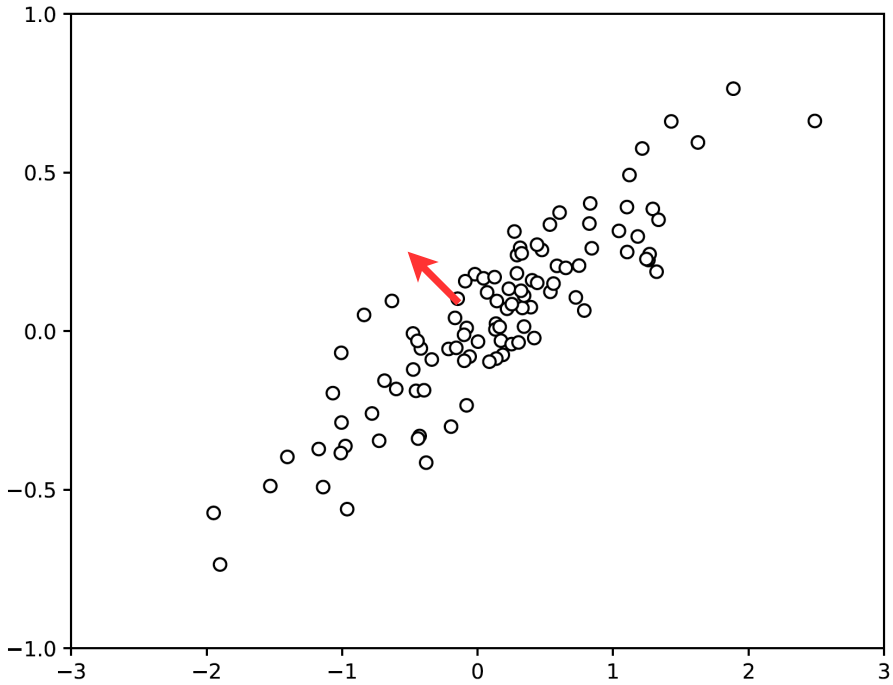
Principal Component Analysis

2. The figure below shows a two-dimensional dataset. Draw the vector corresponding to the **second** principal component.



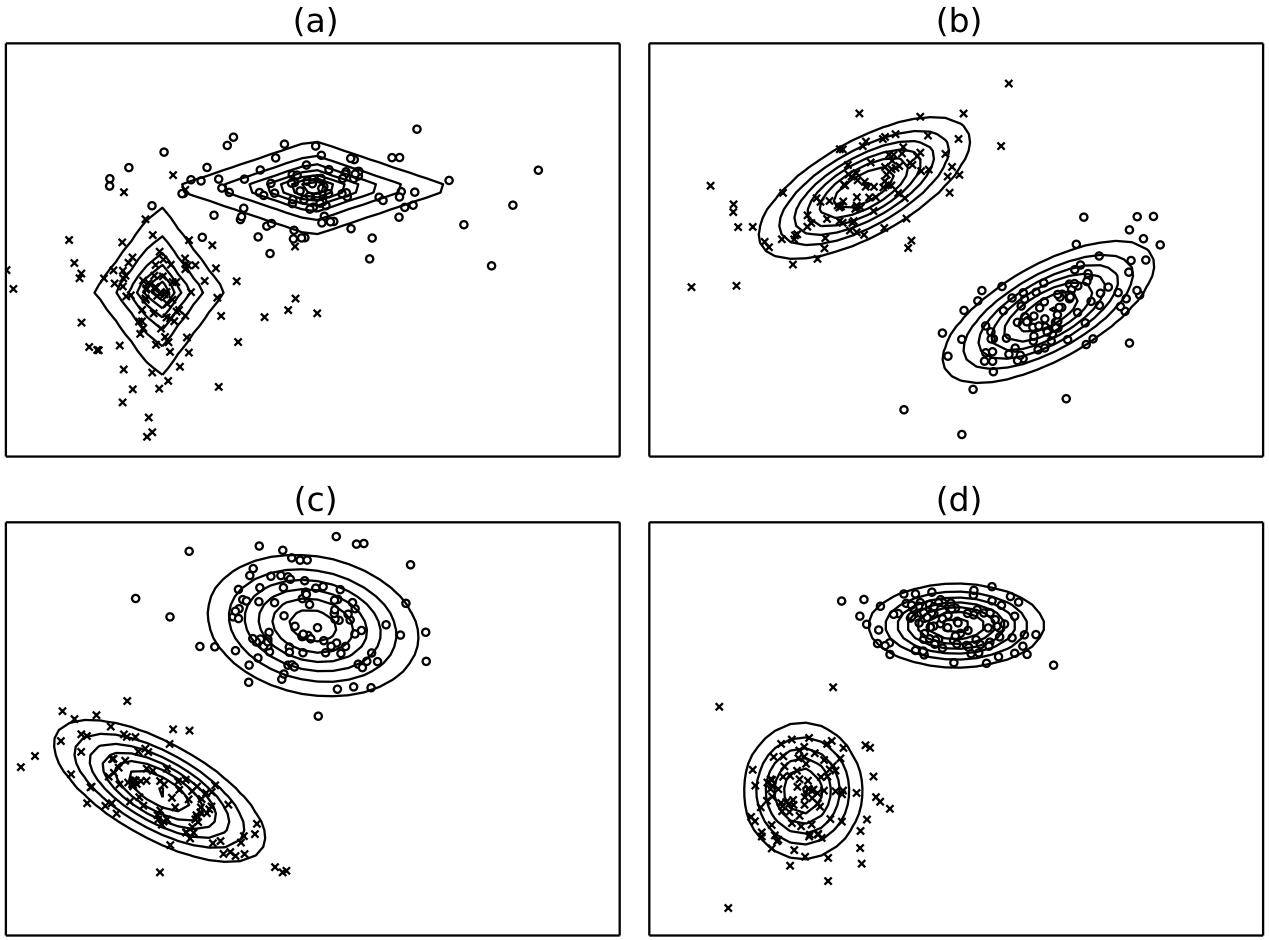
Principal Component Analysis

2. The figure below shows a two-dimensional dataset. Draw the vector corresponding to the **second** principal component.



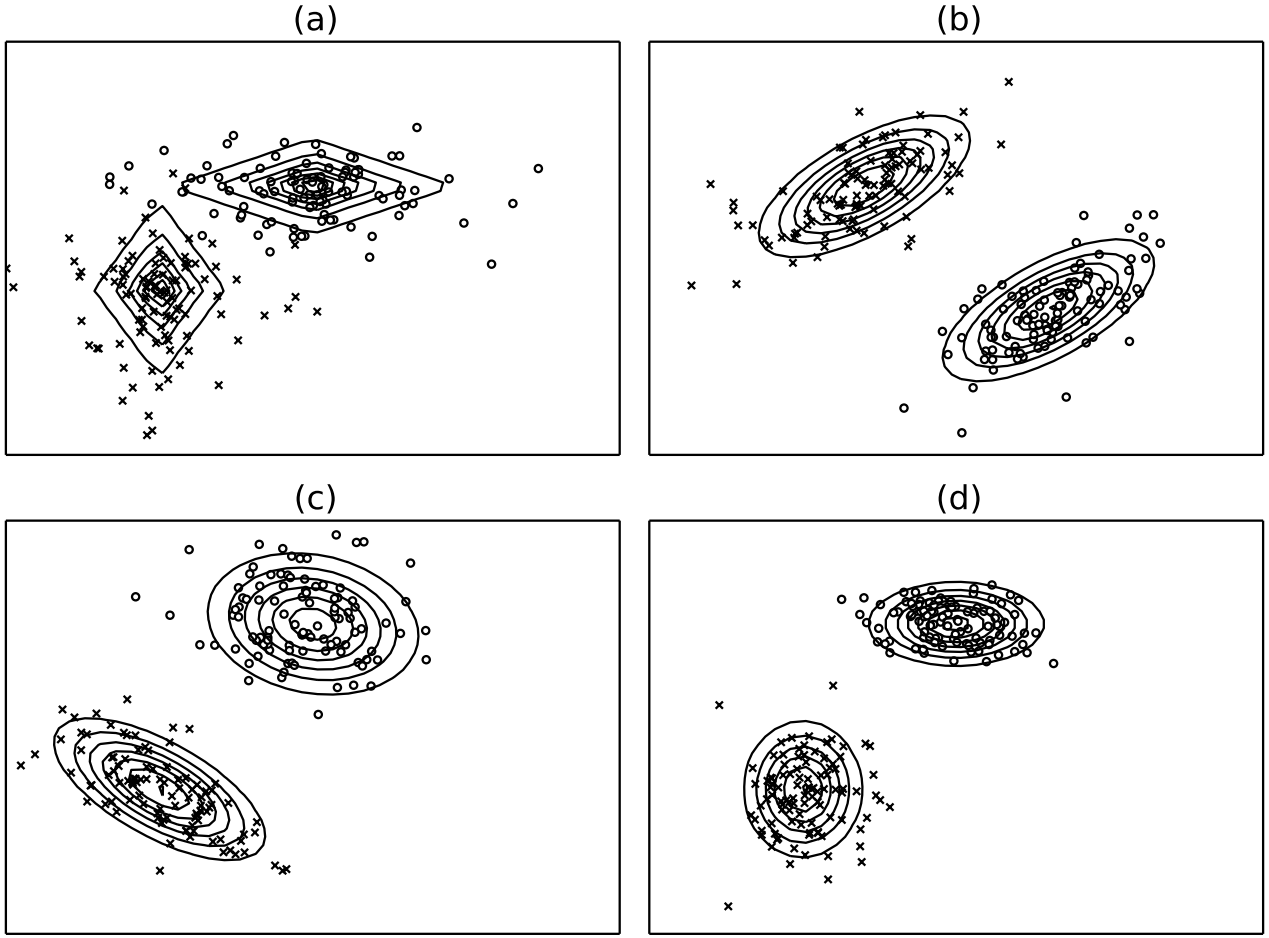
Probabilistic Models: Naive Bayes

Which of the following diagrams could be a visualization of a Naive Bayes classifier? Select all that applies.



Probabilistic Models: Naive Bayes

Which of the following diagrams could be a visualization of a Naive Bayes classifier? Select all that applies.



Answer: A, D

Probabilistic Models: Naive Bayes

Question:

- Consider the following problem, in which we have two classes: $\{Tainted, Clean\}$, and each data x has 3 attributes: (a_1, a_2, a_3).
- These attributes are also binary variables: $a_1 \in \{on, off\}$, $a_2 \in \{blue, red\}$, $a_3 \in \{light, heavy\}$.
- We are given a training set as follows:
  1. $Tainted$: $(on, blue, light)$ $(off, red, light)$ $(on, red, heavy)$
  2. $Clean$: $(off, red, heavy)$ $(off, blue, light)$ $(on, blue, heavy)$

(A) Manually construct Naive Bayes Classifier based on the above training data. Compute the following probability tables: a) the class prior probability, b) the class conditional probabilities of each attribute.

Probabilistic Models: Naive Bayes

(a) Class prior probability:

- $p(c = Tainted) = 3/6 = 1/2$,
- $p(c = Clean) = 1/2$

(b) The class conditional distributions:

- $p(a_1 = on|c = Tainted) = 2/3$, $p(a_1 = off|c = Tainted) = 1/3$
- $p(a_2 = blue|c = Tainted) = 1/3$, $p(a_2 = red|c = Tainted) = 2/3$
- $p(a_3 = light|c = Tainted) = 2/3$,
  $p(a_3 = heavy|c = Tainted) = 1/3$
- $p(a_1 = on|c = Clean) = 1/3$, $p(a_1 = off|c = Clean) = 2/3$
- $p(a_2 = blue|c = Clean) = 2/3$, $p(a_2 = red|c = Clean) = 1/3$
- $p(a_3 = light|c = Clean) = 1/3$, $p(a_3 = heavy|c = Clean) = 2/3$

Probabilistic Models: Naive Bayes

(B) Classify a new example $(on, red, light)$ using the classifier you built above. You need to compute the posterior probability (up to a constant) of class given this example.

Probabilistic Models: Naive Bayes

(B) Classify a new example $(on, red, light)$ using the classifier you built above. You need to compute the posterior probability (up to a constant) of class given this example.

Answer: To classify $\mathbf{x} = (on, red, light)$, we have:

$$p(c|\mathbf{x}) = \frac{p(c)p(x|c)}{p(c = Tainted)p(x|c = Tainted) + p(c = Clean)p(x|c = Clean)}$$

Computing each term:

$$\begin{aligned} p(c = T)p(x|c = T) = \big(&p(c = T)p(a_1 = on|c = T)p(a_2 = red|c = T) \\ &p(a_3 = light|c = T)\big) \\ = &\frac{1}{2} \times \frac{2}{3} \times \frac{2}{3} \times \frac{2}{3} \\ = &\frac{8}{54} \end{aligned}$$

(B) Classify a new example $(on, red, light)$ using the classi
er you built above. You need to compute the posterior probability (up to a constant) of class given this example.

Answer: Similarly,

$$p(c = Clean)p(x|c = Clean) = \frac{1}{2} \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{54}$$

Therefore, $p(c = Tainted|\mathbf{x}) = 8/9$ and $p(c = Clean|\mathbf{x}) = 1/9$, according to Naive Bayes classifier this example should be classified as **Tainted**.

SVM

We showed that the Support Vector Machine (SVM) can be viewed as minimizing hinge loss:

$$\min_{\mathbf{w},b} \sum_{i=1}^{N} \mathcal{L}_{\mathrm{H}}(y,t) + \frac{1}{2\gamma}\|\mathbf{w}\|_2^2$$

where $\mathcal{L}_{\mathrm{H}}$ is the Hinge loss.

(A) Write down the equation for Hinge Loss in terms of t and y

SVM

We showed that the Support Vector Machine (SVM) can be viewed as minimizing hinge loss:

$$\min_{\mathbf{w},b} \sum_{i=1}^{N} \mathcal{L}_{\mathrm{H}}(y,t) + \frac{1}{2\gamma}\|\mathbf{w}\|_2^2$$

where $\mathcal{L}_{\mathrm{H}}$ is the Hinge loss.

Answer: Hinge Loss is defined as:

$$\mathcal{L}_{\mathrm{H}}(y,t) = \max(0, 1 - ty)$$

(A) TRUE or FALSE: if the total hinge loss is zero, then every training example must be classified correctly. Justify your answer.

(A) TRUE or FALSE: if the total hinge loss is zero, then every training example must be classified correctly. Justify your answer.

Answer: TRUE. The hinge loss can only be zero if the example satisfies the margin constraint, and hence is classified correctly. This must be true for every example if the total hinge loss is zero.

Ensembling Methods

Suppose your classifier achieves poor accuracy on both the training and test sets. Which would be a better choice to try to improve the performance: bagging or boosting? Justify your answer.

Ensembling Methods

Suppose your classifier achieves poor accuracy on both the training and test sets. Which would be a better choice to try to improve the performance: bagging or boosting? Justify your answer.

Answer:

- The model is underfitting, has high bias
- Bagging reduces variance, whereas boosting reduces the bias
- Therefore, use **boosting**

Neural Network

Backprop Consider a neural network with N input units, N output units, and K hidden units. The activations are computed as follows:

$$\begin{aligned}
\mathbf{z} &= \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} \\
\mathbf{h} &= \sigma(\mathbf{z}) \\
\mathbf{y} &= \mathbf{x} + \mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)} \\
\mathcal{J} &= \frac{1}{2}\|\mathbf{y} - \mathbf{s}\|^2
\end{aligned}$$

for given vectors $\mathbf{s}$.

(A) Draw the computation graph relating $\mathbf{x}$, $\mathbf{z}$, $\mathbf{h}$, $\mathbf{y}$, and $\mathcal{J}$.

Neural Network

(A) Draw the computation graph relating $\mathbf{x}$, $\mathbf{z}$, $\mathbf{h}$, $\mathbf{y}$, and $\mathcal{J}$.

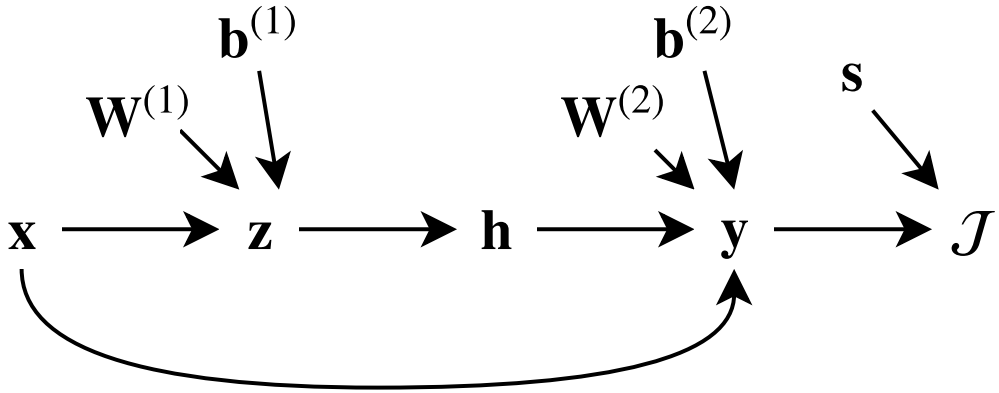


Figure: Computational graph

Neural Network

(B) Derive the backprop equations for computing $\overline{\mathbf{x}} = \partial \mathcal{J} / \partial \mathbf{x}$.

$$
\begin{aligned}
\mathbf{z} &= \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)} \\
\mathbf{h} &= \sigma(\mathbf{z}) \\
\mathbf{y} &= \mathbf{x} + \mathbf{W}^{(2)}\mathbf{h} + \mathbf{b}^{(2)} \\
\mathcal{J} &= \frac{1}{2}\|\mathbf{y} - \mathbf{s}\|^2
\end{aligned}
$$

for given vectors $\mathbf{s}$.

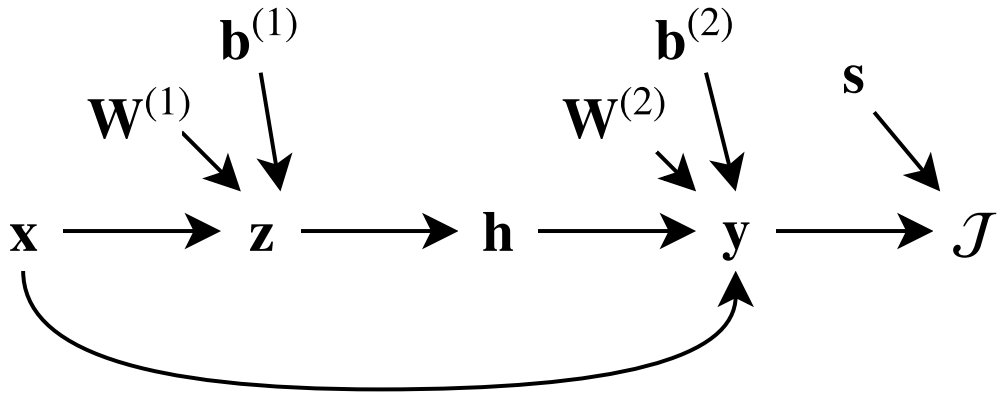


Figure: Computational graph

$$
\begin{aligned}
\overline{\mathcal{J}} &= 1 \\
\overline{\mathbf{y}} &= \overline{\mathcal{J}} \frac{\partial \mathcal{J}}{\partial \mathbf{y}} \\
&= \overline{\mathcal{J}}(\mathbf{y} - \mathbf{s}) \\
\overline{\mathbf{h}} &= \overline{\mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{h}} \\
&= [\mathbf{W}^{(2)}]^\top \overline{\mathbf{y}} \\
\overline{\mathbf{z}} &= \overline{\mathbf{h}} \frac{\partial \mathbf{h}}{\partial \mathbf{z}} \\
&= \overline{\mathbf{h}} \circ \sigma'(\mathbf{z}) \\
\overline{\mathbf{x}} &= \overline{\mathbf{z}} \frac{\partial \mathbf{z}}{\partial \mathbf{x}} + \overline{\mathbf{y}} \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \\
&= [\mathbf{W}^{(1)}]^\top \overline{\mathbf{z}} + \overline{\mathbf{y}}
\end{aligned}
$$