CS 5354/CS 4365 Intelligent Computing
Fall 2019, Test 3

Name: _____

General comments:

- you are allowed up to 5 pages of handwritten notes;
- **be careful:**
  - in some problems, you need to solve all the parts; such parts are marked as X.a, X.b. etc.;
  - in other problems, you only need to solve one part; such parts are marked as X.0, X.1, X.2, and X.3; to decide which part you solve, divide the last two digits of your UTEP ID by 4, and solve problems corresponding to the remainder; for example, if the remainder is 0, solve problems 1.0, 2.0, etc.;
- if you need extra pages, place your last name on each extra page.

Good luck!

10|10

1. Solve all parts of this problem.

1.a. If we use 5 neurons each of which uses 2 bits, how many possible models can we represent? How many models could we represent in principle if we used the same number of bits?

1.b. Explain why, to get a better accuracy, we need to go from the traditional neural network with many neurons in a single layer to a multi-layer structure with a small number of neurons in each layer?

Q 1a.

$N = a * b = 5 \times 2 = 10$

Thus the number of models is $\frac{2^N}{N!}$

i.e $\frac{2^{10}}{5!}$

In principle if we used same number of bits we could represent 2^N models i.e

2^{10}

Q 16.

If we have K neuron's & each neuron K is characterized by several weight's W_K & ω_{Ki} which require b number of bit's to describe these weights. Then the total no of bit's required to describe weight's during training is $N = K \cdot b$. Thus we should have 2^N different sequences of length N. Thus it seem's we should have 2^N model's however if we swap two neuron's the resulting function

$$f(x_1, \ldots x_n) = \sum_{K=1}^{K} W_K S\left(\sum_{i=1}^{\hat{}} \omega_{Ki} x_i - \omega_0\right) - W_0$$

will not change. Also instead of swapping if we apply any permutation we get the exact same result.

Thus for K neuron's there are K! possible permutation's thus actual no of model's we can have is $\frac{2^N}{K!}$

In traditional Neural network's all neuron's are in same layer thus K is very large resulting in small number of model's. However in case of deep neural network's we increase the number of layer's to decrease the no of neurons in each layer thus effectively making K smaller and increasing number of model's which in turn increases the accuracy.

10/10

2. Solve all parts of this problem.

2.a. Which activation function is actually used in deep learning?

2.b. Why do we need pooling in deep learning?

2.c. Which pooling operations are actually used in deep learning?

Q 2a. Rectified linear activation function is actually used in deep learning

$$S(z) = \max(z, 0)$$

Q 2b. In case of deep neural Networks we have fewer number of neurons in each layer thus we cannot have a large number of input paramaters. Thus before we start procesing we need to combine several input paramaters into a single input value, this proces of combining several inputs into one input value is known as pooling. Thus we need pooling in deep neural networks to match number of input's to the number of neurons in first layer.

Q 2c. we use following 3 pooling operations in deep learning

i) max pooling $P(x, y) = \max(x, y)$

ii) min pooling $P(x, y) = \min(x, y)$

iii) Average pooling $P(x, y) = \frac{x+y}{2}$

10 + 3
10

3-4. Solve only one part of this problem -- the part corresponding to your UTEP ID.

3-4.0. Explain, in detail, why for deep neural networks, the activation function should be piece-wise linear. Explain why we cannot have $c_- = c_+$.

3-4.1. Why cannot we require shift-invariance instead of scale-invariance? Explain in detail.

3-4.2. Which pooling operations work most successfully in deep learning? Provide a theoretical explanation for these operations. Suppose that instead of the three operations, we use the operation $p(x,y) = a * \max(x,y) + (1 - a) * \min(x,y)$, with $a = 0.4$. Provide an example when for this operation, the result of pooling 4 numbers depends on how we divide these 4 numbers into pairs.

3-4.3. Explain what researchers mean when they say that deep neural networks are vulnerable to noise. Explain why deep neural networks are much more vulnerable to the noise in the original data than the traditional neural networks. If neurons on each layer amplify the noise by a factor of 4, how much will the noise by amplified after passing a 5-layer neural network? How to deal with this sensitivity of deep neural networks?

3-4.3

Deep Neural Networks are more vulnerable to noise because a small change in input (noise) results in a large change in the output. In each layer noise (δ) is amplified by factor c where $c > 1$ therefore for many layers we get $c^L \delta$ where L = number of layers.

Suppose $L = 5$ & $C = 4$

then the total noise amplification is $4^5 = 1024$

To deal with this problem we train Deep neural networks with noise.

3-4.0 want activation function independent of shifts

$y' = S(x)$; $x' = \lambda x$; $y' = \lambda y \Rightarrow y' = S(x')$

$x' = \lambda x$ & $y' = \lambda y \Rightarrow \textcircled{1}\ \lambda y = S(\lambda x)$

$y \Rightarrow S(x)$ $\quad S(\lambda) \Rightarrow \lambda S(x)$

$x = 1 \Rightarrow S(\lambda) = \lambda S(1) \Rightarrow$

$S(1) = c_+$; $\lambda \to z$ then, for $(z \geq 0)$

$S(z) = c_+ z$

$x = -1 \Rightarrow S(-\lambda) = \lambda S(-1)$

$S(-1) = c_-$ $\quad \lambda \to z$

then, for all $(z < 0)$ $S(z) = c_- z$

$$S(z) = \begin{cases} c_+ z & x > 0 \\ c_- z & x < 0 \end{cases}$$

If $c_- = c_+$ then function is linear so c_- must not be c_+

10/10

3-4. Solve only one part of this problem -- the part corresponding to your UTEP ID.

3-4.0. Explain, in detail, why for deep neural networks, the activation function should be piece-wise linear. Explain why we cannot have $c_- = c_+$.

3-4.1. Why cannot we require shift-invariance instead of scale-invariance? Explain in detail.

3-4.2. Which pooling operations work most successfully in deep learning? Provide a theoretical explanation for these operations. Suppose that instead of the three operations, we use the operation $p(x,y) = a * \max(x,y) + (1 - a) * \min(x,y)$, with $a = 0.4$. Provide an example when for this operation, the result of pooling 4 numbers depends on how we divide these 4 numbers into pairs.

3-4.3. Explain what researchers mean when they say that deep neural networks are vulnerable to noise. Explain why deep neural networks are much more vulnerable to the noise in the original data than the traditional neural networks. If neurons on each layer amplify the noise by a factor of 4, how much will the noise by amplified after passing a 5-layer neural network? How to deal with this sensitivity of deep neural networks?

if $s(x) = y$ and $x' = \lambda x$, $y' = \lambda y$ then $s(x') = y'$

$s(\lambda x) = \lambda y$, $s(\lambda x) = \lambda s(x)$ $\rightarrow \lambda > 0$. $s(\lambda) = \lambda \cdot s(1)$

$s(z) = c_+ \cdot z = s(1)$ if $z > 0$, $s(-\lambda) = \lambda s(-1)$, $x = -1$, $z = -\lambda$

if $z < 0$ $s(z) = (-z) \cdot (c_-) = c_- z$

we must have $c_+ \neq c_-$, since otherwise,

the function $s(z)$ would be linear,

and we know that with linear functions,

we can only describe linear dependencies.

10/10

3-4. Solve only one part of this problem -- the part corresponding to your UTEP ID. 80625697

3-4.0. Explain, in detail, why for deep neural networks, the activation function should be piece-wise linear. Explain why we cannot have $c_- = c_+$.

3-4.1. Why cannot we require shift-invariance instead of scale-invariance? Explain in detail.

3-4.2. Which pooling operations work most successfully in deep learning? Provide a theoretical explanation for these operations. Suppose that instead of the three operations, we use the operation $p(x,y) = a * \max(x,y) + (1 - a) * \min(x,y)$, with $a = 0.4$. Provide an example when for this operation, the result of pooling 4 numbers depends on how we divide these 4 numbers into pairs.

3-4.3. Explain what researchers mean when they say that deep neural networks are vulnerable to noise. Explain why deep neural networks are much more vulnerable to the noise in the original data than the traditional neural networks. If neurons on each layer amplify the noise by a factor of 4, how much will the noise by amplified after passing a 5-layer neural network? How to deal with this sensitivity of deep neural networks?

24
4) 97
96
1 → remainder

Q 3.4.1

If we require shift-invariance we must have

$y = s(x)$

$x' = x + x_0$

$y' = y + x_0$

$s(x') = y'$

$s(x + x_0) = y + x_0$

$s(x + x_0) = s(x) + x_0$ for all possible values of x and x_0

for $x = 0$

$s(x_0) = s(0) + x_0$

Renaming $s(0)$ as a and x_0 as z we have

$s(z) = a + z$

This is a linear function. Thus if we require shift-invariance we end up with a linear activation function which cannot be used to represent non-linear processes and since not all the processes in real world are linear we need non-linear activation function to account for non-linear processes.

10/10

3-4. Solve only one part of this problem -- the part corresponding to your UTEP ID.

3-4.0. Explain, in detail, why for deep neural networks, the activation function should be piece-wise linear. Explain why we cannot have $c_- = c_+$.

3-4.1. Why cannot we require shift-invariance instead of scale-invariance? Explain in detail.

3-4.2. Which pooling operations work most successfully in deep learning? Provide a theoretical explanation for these operations. Suppose that instead of the three operations, we use the operation $p(x,y) = a * \max(x,y) + (1 - a) * \min(x,y)$, with $a = 0.4$. Provide an example when for this operation, the result of pooling 4 numbers depends on how we divide these 4 numbers into pairs.

3-4.3. Explain what researchers mean when they say that deep neural networks are vulnerable to noise. Explain why deep neural networks are much more vulnerable to the noise in the original data than the traditional neural networks. If neurons on each layer amplify the noise by a factor of 4, how much will the noise by amplified after passing a 5-layer neural network? How to deal with this sensitivity of deep neural networks?

minimum, maximum and arithmetic average.

pooling operations require scale-invariance and shift invariance.

$$\begin{cases} p(\lambda a, \lambda b) = \lambda p(a,b) & ① \\ p(a+a_0, b+a_0) = p(a,b) + a_0 & ② \end{cases}$$

substitute $a=0$, $a_0 = x$, $b = y-x$ in ②, we get

$$p(x,y) = p(0, y-x) + x \quad ③$$

substitute $\lambda = y-x$, $a=0$, $b=1$ in equation ④, we get $p(0, y-x) = (y-x)\,p(0,1)$ ④

substitute ④ into ③, we get

$$p(x,y) = (y-x)\,p(0,1) + x$$

$$[\alpha]$$

$$= x + \alpha(y-x)$$

$$= (1-\alpha)x + \alpha y$$

pool 0, 1, 1 2

case 1, (0,1) & (1,2)

$$\begin{cases} \alpha \cdot 1 + (1-\alpha)\cdot 0 = \alpha \\ \alpha \cdot 2 + (1-\alpha) 1 = 1 + \alpha \end{cases}$$

$1+\alpha > \alpha$

thus, combining α and $1+\alpha$ leads to

$\alpha(1+\alpha) + (1-\alpha)\alpha = 2\alpha$

case 2, combine (1, 1) and (0, 2)

similarly we get $\begin{cases} \alpha \cdot 2 + (1-\alpha)\cdot 1 = 2 \\ \alpha \cdot 2 + (1-\alpha)\cdot 0 = 2\alpha \end{cases}$

results depends on which value is large

If $2\alpha > 1$, then $\alpha(2\alpha) + (1-\alpha)1 = 2\alpha^2 + 1 - \alpha$

$\alpha = 0.5$ or 1

If $2\alpha \le 1$, then $\alpha 2 + (1-\alpha) 2\alpha = 3\alpha - 2\alpha^2$

$\alpha = 0.5$ or 0

$$\begin{cases} p(x,y) = \min(x,y) & \text{when } \alpha = 0 \\ p(x,y) = \frac{x+y}{2} & \text{when } \alpha = .5 \\ p(x,y) = \max(x,y) & \text{when } \alpha = 1. \end{cases}$$

pool 0, 1, 1, 2 with

$p(x,y) = 0.4\max(x,y) + 0.6\min(x,y)$

case 1 :

(0, 1) & (1, 2)

$p(0,1) = 0.4 \cdot 1 - 0.6 \times 0 = 0.4$

$p(1,2) = 0.4 \times 2 - 0.6 \times 1 = 0.2$

$p(0.4, 0.2) = 0.4 \times 0.4 - 0.6 \times 0.2 = 0.04$

case 2

(1, 1) & (0, 2)

$p(1,1) = 0.4 \times 1 + 0.6 \times 1 = 1$

$p(0,2) = 0.4 \times 2 + 0.6 \times 0 = 0.8$

$p(1, 0.8) = 0.4 \times 1 + 0.6 \times 0.8 = 0.88$

two ways of combinations generates two different results. this is not a valid pooling operation.

10/10 23 mod 4 = 3

3-4. Solve only one part of this problem -- the part corresponding to your UTEP ID.

3-4.0. Explain, in detail, why for deep neural networks, the activation function should be piece-wise linear. Explain why we cannot have $c_- = c_+$.

3-4.1. Why cannot we require shift-invariance instead of scale-invariance? Explain in detail.

3-4.2. Which pooling operations work most successfully in deep learning? Provide a theoretical explanation for these operations. Suppose that instead of the three operations, we use the operation $p(x,y) = a * \max(x,y) + (1 - a) * \min(x,y)$, with $a = 0.4$. Provide an example when for this operation, the result of pooling 4 numbers depends on how we divide these 4 numbers into pairs.

3-4.3 Explain what researchers mean when they say that deep neural networks are vulnerable to noise. Explain why deep neural networks are much more vulnerable to the noise in the original data than the traditional neural networks. If neurons on each layer amplify the noise by a factor of 4, how much will the noise by amplified after passing a 5-layer neural network? How to deal with this sensitivity of deep neural networks?

Researchers say that DNNs are vulnerable to noise since one minor change could result in a wrong conclusion due to DNN's sensitivity.

DNNs are more vulnerable to noise than traditional NNs since the noise is amplified in every layer, per layer we have $c \cdot \delta$ where δ is the noise level, so in multiple layers we will have the exponential function $c^L \cdot \delta$ where L = no. of layers.

If neurons on each layer amplified the noise by a factor of 4 and we had 5 layers, we would have a noise level value of $4^5 \cdot \delta = 1024 \cdot \delta$

To deal with DNN's sensitivity we can artificially add noise to the observed training patterns.

10/10

5. Solve all parts of this problem.

5.a. Is $0.5i|0\rangle + 0.5|1\rangle$ a valid quantum state? Is $0.8i|0\rangle - 0.6|1\rangle$ a valid state? Explain why.

5.b. What is the probability that when we measure the bit in the state $0.6|0\rangle - 0.8|1\rangle$, we will get 0? What is the probability that we will get 1? How are similar probabilities explaining which combinations are valid states and which are not?

5.c. What is the state of a system of two independent particles if:

- the first particle is in the state $0.6|0\rangle - 0.8|1\rangle$, and
- the second particle is in the state $0.8i|0\rangle - 0.6|1\rangle$?

Hint: use tensor product.

Q 5a. For states of type $\alpha|0\rangle + \beta|1\rangle$ is a valid state if $|\alpha|^2 + |\beta|^2 = 1$

i) $(0.5)^2 + (0.5)^2 = 0.25 + 0.25 = 0.5 \neq 1$ thus this is not a valid state

ii) $(0.8)^2 + (-0.6)^2 = 0.64 + 0.36 = 1$ thus this is a valid state

Q 5b. For a state $\alpha|0\rangle + \beta|1\rangle$ if we measure the bit in this state we will get 0 with probability $|\alpha|^2$ and 1 with probability $|\beta|^2$ and these probabilities should add up to 1 for a valid state i.e $|\alpha|^2 + |\beta|^2 = 1$

Here

$$P(|0\rangle) = (0.6)^2 = 0.36$$
$$P(|1\rangle) = (0.8)^2 = 0.64$$

$0.36 + 0.64 = 1$

Thus $0.6|0\rangle - 0.8|1\rangle$ is a valid state

Q 5c. The state of system is given by tensor product.

$$(0.6|0\rangle - 0.8|1\rangle) \otimes (0.8i|0\rangle - 0.6|1\rangle)$$
$$= 0.48i|00\rangle - 0.36|01\rangle - 0.64i|10\rangle + 0.48|11\rangle$$

10/10

6. Solve all parts of this problem.

6.a. What is the result of applying Hadamard transform to the state $0.6|0\rangle - 0.8|1\rangle$?

6.b. What is the result of applying the quantum version of negation to the state $0.6|0\rangle - 0.8|1\rangle$?

6.c. What is the result of applying the quantum version of "or"-operation to the states $0.6|0\rangle - 0.8|1\rangle$ and $0.8i|0\rangle - 0.6|1\rangle$?

Q6a.

$$H(0.6|0\rangle - 0.8|1\rangle) = H(0.6|0\rangle) - H(0.8|1\rangle)$$

$$H(0.6|0\rangle) = 0.6\left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{3}{5\sqrt{2}}|0\rangle + \frac{3}{5\sqrt{2}}|1\rangle$$

$$H(0.8|1\rangle) = 0.8\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) = \frac{4}{5\sqrt{2}}|0\rangle - \frac{4}{5\sqrt{2}}|1\rangle$$

Thus

$$H(0.6|0\rangle - 0.8|1\rangle) = \frac{3}{5\sqrt{2}}|0\rangle + \frac{3}{5\sqrt{2}}|1\rangle - \frac{4}{5\sqrt{2}}|0\rangle + \frac{4}{5\sqrt{2}}|1\rangle$$

$$= \frac{-1}{5\sqrt{2}}|0\rangle + \frac{7}{5\sqrt{2}}|1\rangle$$

Q6b.

$$(0.6|0\rangle - 0.8|1\rangle) \otimes |0\rangle$$

$$0.6|00\rangle - 0.8|10\rangle$$

Now we apply $f(x) = 7x$

$|x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$

$|00\rangle \rightarrow |01\rangle$

$|10\rangle \rightarrow |10\rangle$

Thus we have

$$0.6|01\rangle - 0.8|10\rangle$$

Q6c.

$$(0.6|0\rangle - 0.8|1\rangle) \otimes (0.8i|0\rangle - 0.6|1\rangle) \otimes |1\rangle$$

$$= 0.48i|001\rangle - 0.36|011\rangle - 0.64i|101\rangle + 0.48|111\rangle$$

$f(x) = x_1$ or x_2

$|x_1, x_2, y\rangle \longrightarrow |x_1, x_2, y \oplus f(x)\rangle$

|0|10

8. Solve only one part of this problem -- corresponding to your UTEP ID.

8.0 What will happen if we apply the function f(x) = 1 to a general 2-bit state

a|00> + b|01> + c|10> + d|11>? → a|01> + b|00> + c|11> + d|10>

8.1 What will happen if we apply the function f(x) = not x to a general 2-bit state

a|00> + b|01> + c|10> + d|11>? → a|01> + b|00> + c|10> + d|11>

8.2. What will happen if we apply the function f(x) = x to a general 2-bit state

a|00> + b|01> + c|10> + d|11>? → a|00> + b|01> + c|11> + d|10>

8.3. What will happen if we apply the function f(x) = 0 to a general 2-bit state

a|00> + b|01> + c|10> + d|11>? → a|00> + b|01> + c|10> + d|11>

Q 8.1 $f(x) = \overline{x}$

$(x, y) \rightarrow |x, y \oplus f(x)\rangle$

$|00\rangle \rightarrow |01\rangle$
$|01\rangle \rightarrow |00\rangle$
$|10\rangle \rightarrow |10\rangle$
$|11\rangle \rightarrow |11\rangle$

Thus we will have

$a|01\rangle + b|00\rangle + c|10\rangle + d|11\rangle$

10|10

9-10. Solve only one part of this problem -- corresponding to your UTEP ID.

9-10.0. Explain, step-by-step, what the Deutsch-Josza algorithm will do when the tested function is

$$f(x) = 1.$$

9-10.1. Explain, step-by-step, what the Deutsch-Josza algorithm will do when the tested function is

$$f(x) = \text{not } x.$$

9-10.2. Explain, step-by-step, what the Deutsch-Josza algorithm will do when the tested function is

$$f(x) = x.$$

9-10.3. Explain, step-by-step, what the Deutsch-Josza algorithm will do when the tested function is

$$f(x) = 0.$$

Q 9-10.1

- We start with the state $|01\rangle = |0\rangle \otimes |1\rangle$ where the first bit is in state $|0\rangle$ and second bit is in state $|1\rangle$
- Now we apply the Hadamard transformation to both bits we get:

$$H(|0\rangle) \otimes H(|1\rangle) = \left(\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle\right) \otimes \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)$$

$$= \frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$$

- Now we apply $f(x) = 7x$.

$$(x, y) \longrightarrow |x, y \oplus f(x)$$

$$|00\rangle \longrightarrow |01\rangle$$
$$|01\rangle \longrightarrow |00\rangle$$
$$|10\rangle \longrightarrow |10\rangle$$
$$|11\rangle \longrightarrow |11\rangle$$

Thus we have

$$\frac{1}{2}|01\rangle - \frac{1}{2}|00\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$$

$$= -\left\{\frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle + \frac{1}{2}|11\rangle - \frac{1}{2}|10\rangle\right\}$$

$$= -\left\{\frac{1}{2}|0\rangle\otimes|0\rangle - \frac{1}{2}|0\rangle\otimes|1\rangle + \frac{1}{2}|1\rangle\otimes|1\rangle - \frac{1}{2}|1\rangle\otimes|0\rangle\right\}$$

$$= -\left\{\frac{1}{2}|0\rangle\otimes(|0\rangle - |1\rangle) - \frac{1}{2}|1\rangle\otimes(|0\rangle - |1\rangle)\right\}$$

$$= -\left\{\frac{1}{2}(|0\rangle - |1\rangle)\otimes(|0\rangle - |1\rangle)\right\}$$

$$= -\left\{\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)\otimes\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)\right\}$$

$$= -\{H|1\rangle\otimes H|1\rangle\}$$

$$= -H(|11\rangle)$$

- Now if we apply the Hadamard transformation again we simply get the state $-|11\rangle$ back

- Finally we measure the first bit i.e 1 so the function is not constant.

4/10

9-10. Solve only one part of this problem -- corresponding to your UTEP ID.

9-10.0. Explain, step-by-step, what the Deutsch-Josza algorithm will do when the tested function is

$f(x) = 1$.

9-10.1. Explain, step-by-step, what the Deutsch-Josza algorithm will do when the tested function is

$f(x) = \text{not } x$.

9-10.2. Explain, step-by-step, what the Deutsch-Josza algorithm will do when the tested function is

$f(x) = x$.

9-10.3. Explain, step-by-step, what the Deutsch-Josza algorithm will do when the tested function is

$f(x) = 0$.

case of $f(0)=0$ and $f(1)=1$.

$|00\rangle \Rightarrow |00\rangle$, $|01\rangle \Rightarrow |01\rangle$, $|10\rangle \Rightarrow |11\rangle$,

$|11\rangle \Rightarrow |10\rangle$

$$f(H(|0\rangle) \otimes H(|1\rangle)) = \frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle + \frac{1}{2}|11\rangle - \frac{1}{2}|10\rangle$$

$$= \frac{1}{2}|0\rangle\otimes|0\rangle - \frac{1}{2}|0\rangle\otimes|1\rangle + \frac{1}{2}|1\rangle\otimes|1\rangle - \frac{1}{2}|1\rangle\otimes|0\rangle$$

the first two terms have a comon factor $|0\rangle$, the third and the fourth have a common factor $|1\rangle$, so we have

$$f(H|0\rangle + H(|1\rangle)) = \frac{1}{2}|0\rangle\otimes\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) + \frac{1}{\sqrt{2}}|1\rangle\otimes\left(\frac{1}{\sqrt{2}}|1\rangle - \frac{1}{\sqrt{2}}|0\rangle\right)$$

$$= \frac{1}{\sqrt{2}}|0\rangle\otimes\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right) - \frac{1}{\sqrt{2}}|1\rangle\otimes\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)$$

$$= \left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)\otimes\left(\frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle\right)$$

after applay Hadamard transform back, we get $|1\rangle$

1st bit is 1

10/10

23

9-10. Solve only one part of this problem -- corresponding to your UTEP ID.

9-10.0. Explain, step-by-step, what the Deutsch-Josza algorithm will do when the tested function is

$f(x) = 1$.

9-10.1. Explain, step-by-step, what the Deutsch-Josza algorithm will do when the tested function is

$f(x) =$ not x.

9-10.2. Explain, step-by-step, what the Deutsch-Josza algorithm will do when the tested function is

$f(x) = x$.

9-10.3. Explain, step-by-step, what the Deutsch-Josza algorithm will do when the tested function is

$f(x) = 0$.

9-10.3) s1) apply H gate : $\frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$

s2) apply f : $\frac{1}{2}|00\rangle - \frac{1}{2}|01\rangle + \frac{1}{2}|10\rangle - \frac{1}{2}|11\rangle$

s3) apply H again : since the state didn't change when applying f we know it's still equivalent to $H(|01\rangle)$, thus by applying H again we get

$H(H(|01\rangle)) = |01\rangle$

s4) measure 1st bit : first bit is 0, therefore we would know that the function is constant