1. **Which of the following activation functions is not zero-centered?**
2. Sigmoid
3. Tanh
4. ReLU
5. Softmax

**Sigmoid**: This function outputs values between 0 and 1, which means it isn't zero-centered.

**Tanh**: This function outputs values between -1 and 1, making it zero-centered.

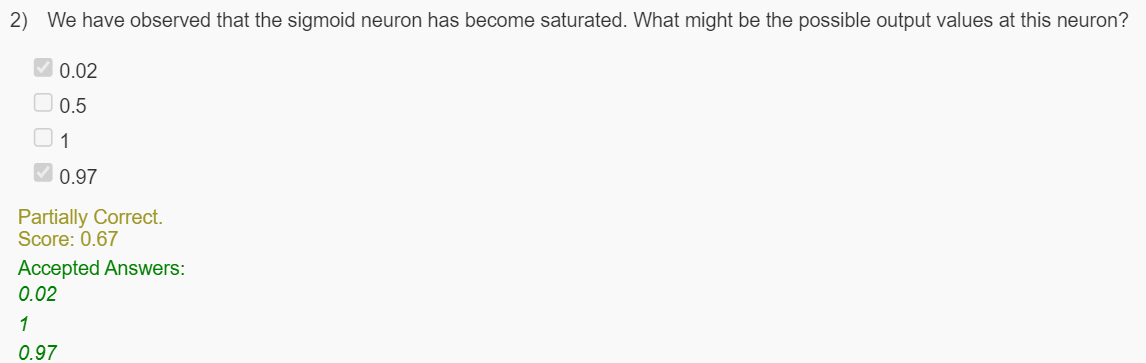
**ReLU (Rectified Linear Unit)**: This outputs values as either 0 or positive, thus it isn't zero-centered.

**Softmax**: While not inherently zero-centered, it's typically used for output layers in classification tasks to produce probability distributions.

So, the ones that aren't zero-centered are **Sigmoid**, **ReLU**, and **Softmax**. Did you expect something different

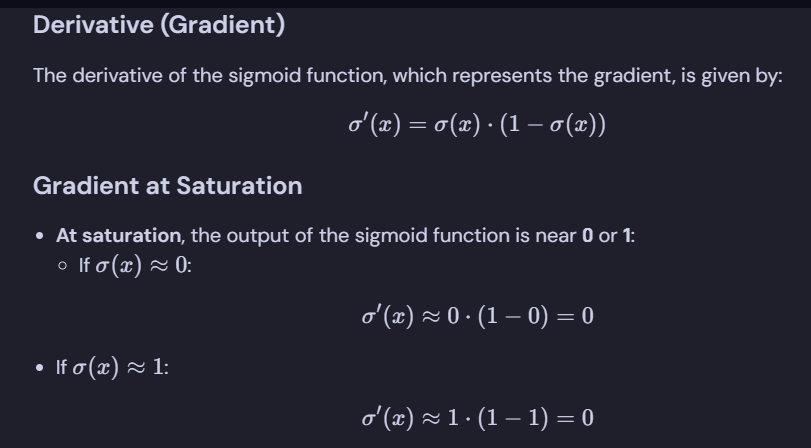
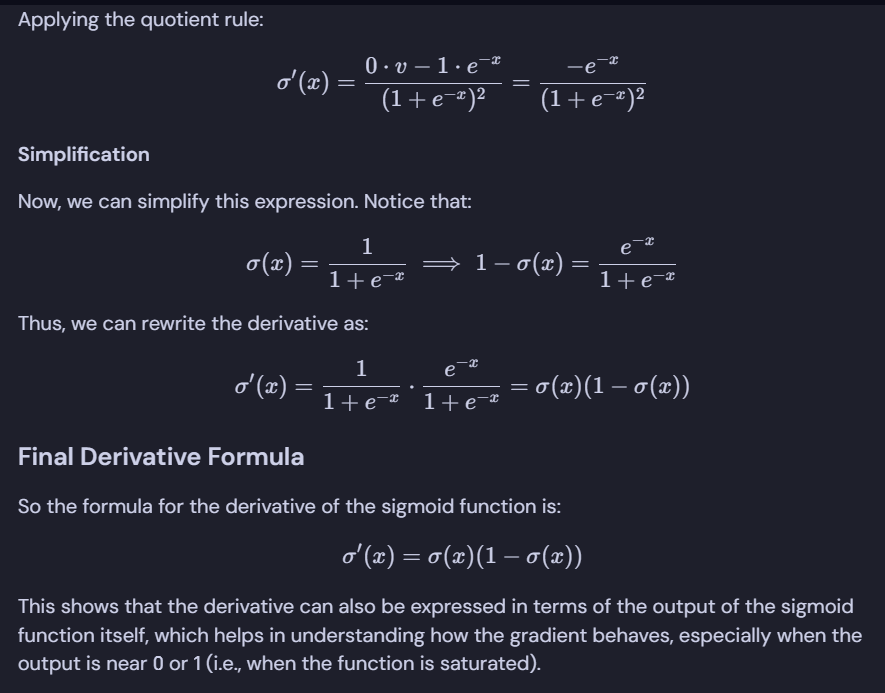
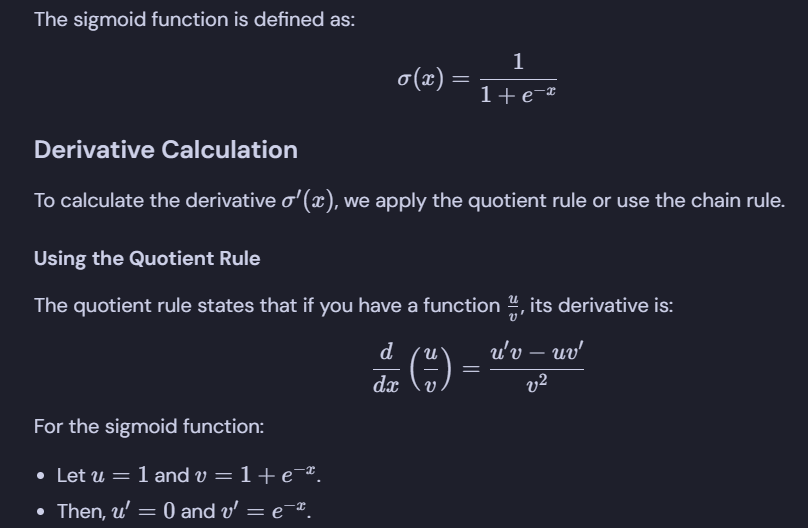
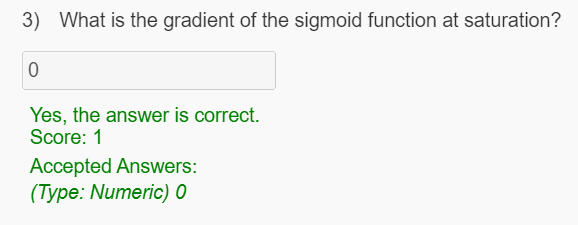
A zerocentered function is one where the mean of its output is around zero. This can make optimization algorithms like gradient descent more efficient because it tends to balance the gradients, avoiding large positive or negative biases. It helps in faster convergence during the training process.

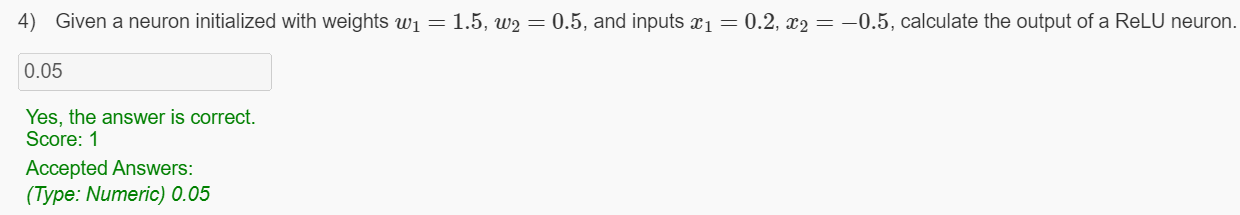
For example, the **Tanh** activation function outputs values in the range of -1 to 1, which means its mean is approximately zero. On the other hand, functions like **Sigmoid** output values between 0 and 1, which shifts the mean away from zero.

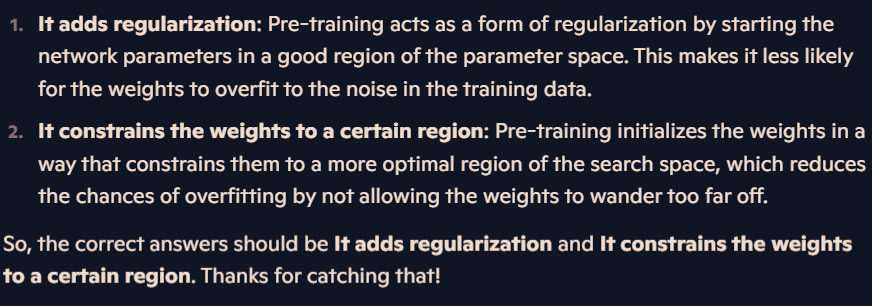
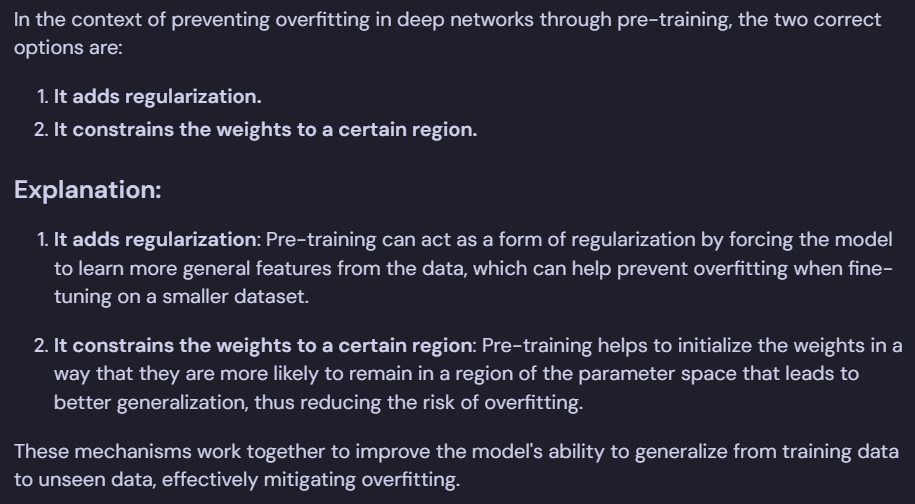
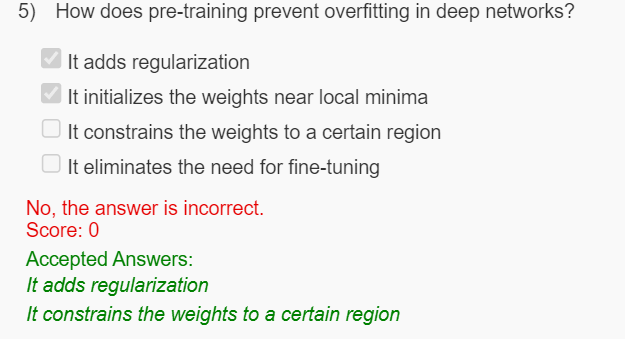
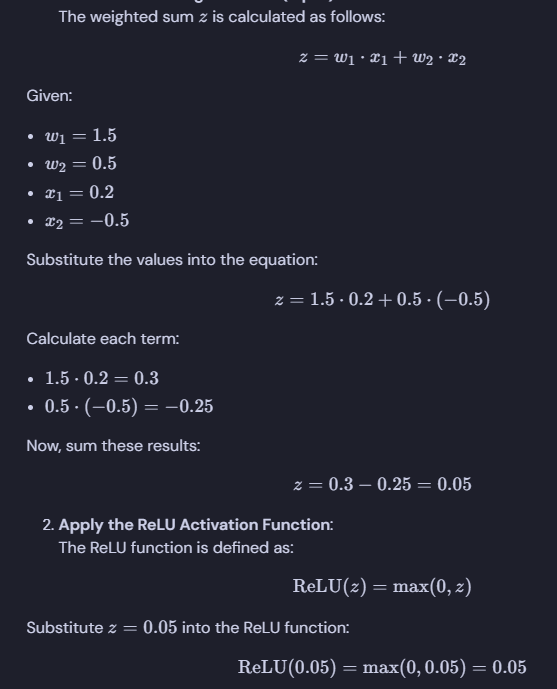
****

when a sigmoid neuron becomes saturated, it means that the output values are pushed towards the extreme ends of the sigmoid function's range, which is between 0 and 1. Saturation typically occurs when the neuron receives very large positive or negative input values, causing the output to be close to 0 or 1.

The saturated output values are 0.02, 1 and 0.97. These values indicate that the sigmoid neuron is operating in the flat regions of the sigmoid curve, leading to very small gradients, which slows down learning.

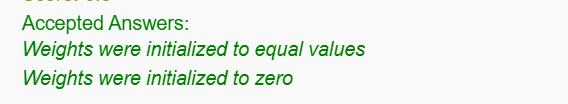
****

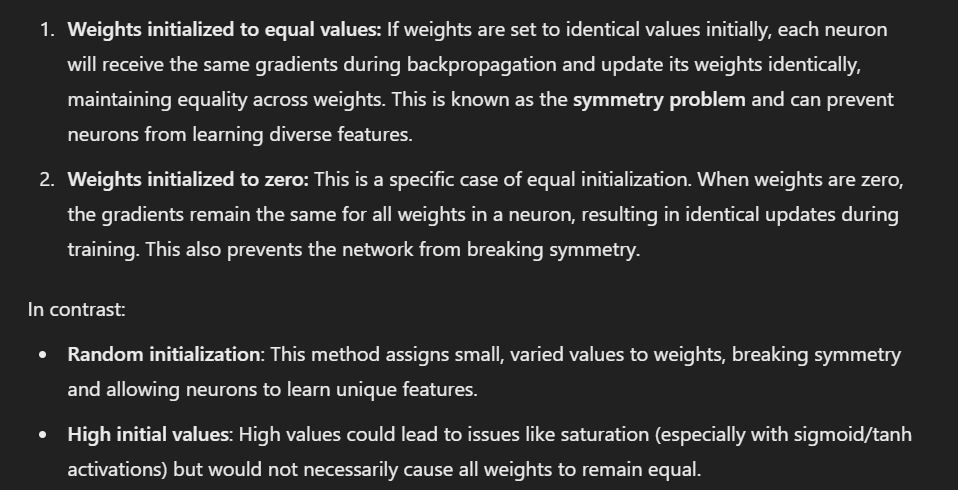
****

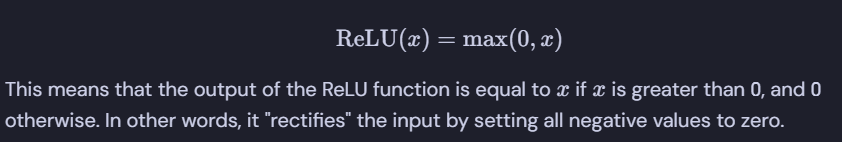
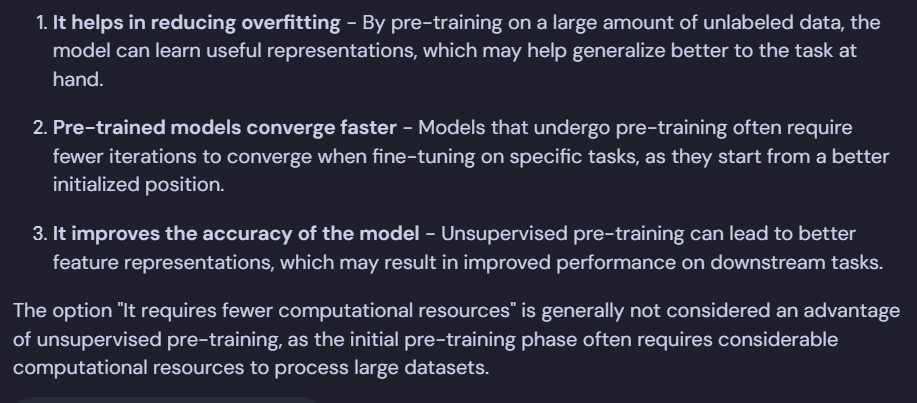
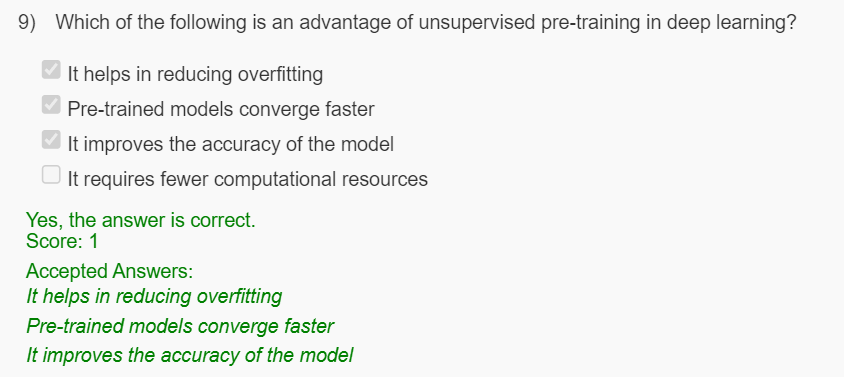
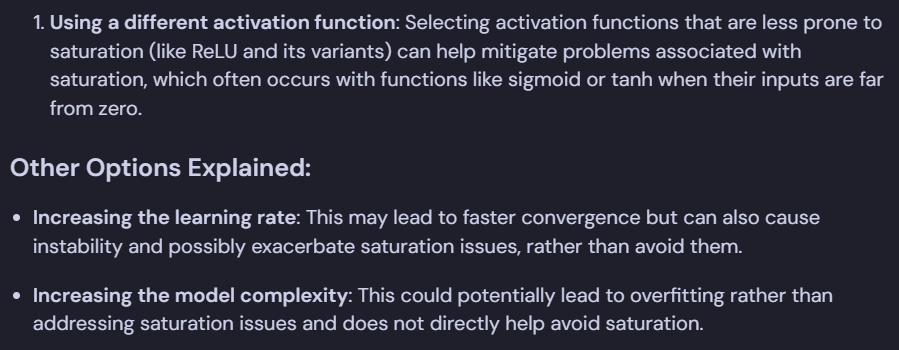
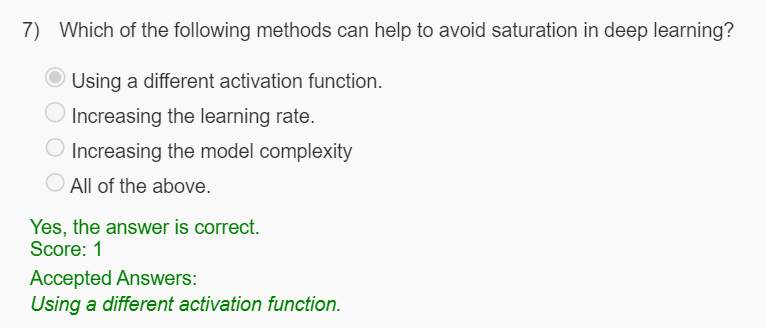
****

**We train a feed-forward neural network and notice that all the weights for a particular neuron are equal. What could be the possible causes of this issue?**

1. Weights were initialized randomly
2. Weights were initialized to high values
3. Weights were initialized to equal values
4. Weights were initialized to zero

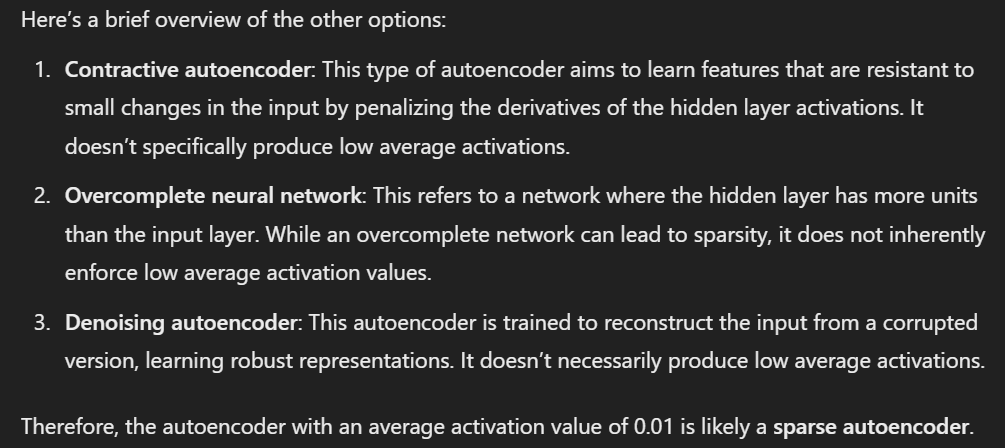
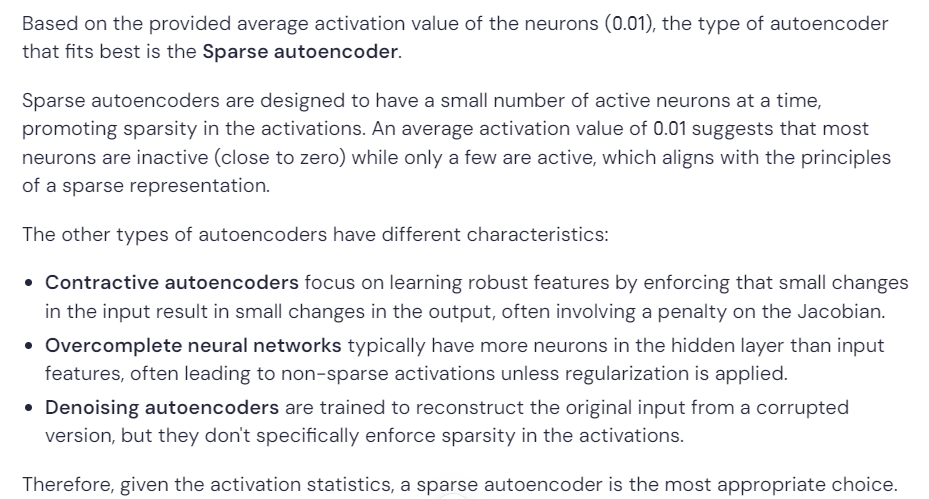






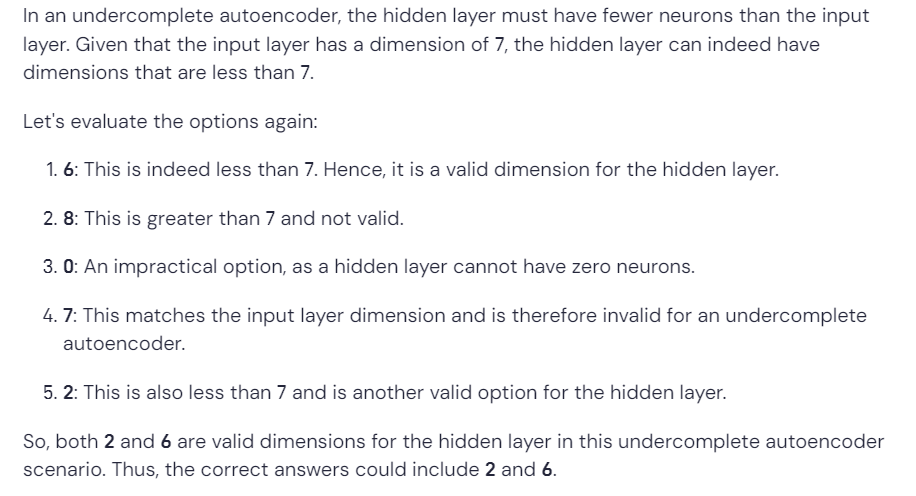
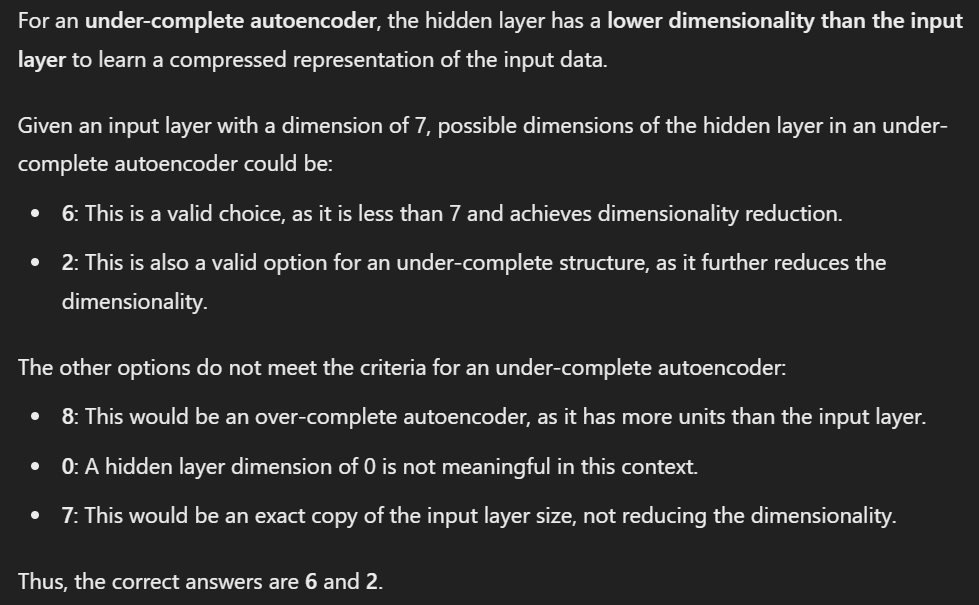
Q. We are given an autoencoder A. The average activation value of neurons in this network is 0.01. The given autoencoder is:

* Contractive autoencoder
* Overcomplete neural network
* Denoising autoencoder
* Sparse autoencoder



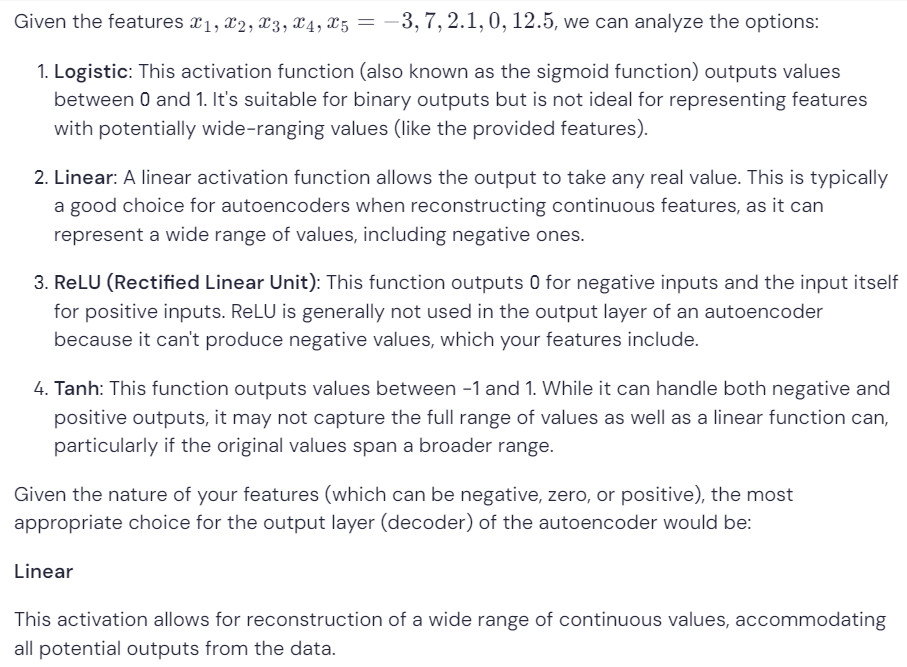
If an under-complete autoencoder has an input layer with a dimension of 7, what could be the possible dimension of the hidden layer?

* 6
* 8
* 0
* 7
* 2



Suppose for one data point we have features x1, x2, x3, x4, x5 = -3, 7, 2.1, 0, 12.5, then which of the following functions should we use on the output layer (decoder)?

* Logistic
* Linear
* ReLU
* Tanh



What is/are the primary advantages of Autoencoders over PCA?

* Autoencoders are less prone to overfitting than PCA.
* Autoencoders are faster and more efficient than PCA.
* Autoencoders can capture nonlinear relationships in the input data.
* Autoencoders require fewer input data than PCA.

1. **Autoencoders can capture nonlinear relationships in the input data**: Unlike PCA, which is a linear method that projects data onto a lower-dimensional space through linear combinations, autoencoders can learn complex nonlinear mappings, allowing them to capture intricate patterns in the data effectively. This is a significant advantage when the data has nonlinear structures.

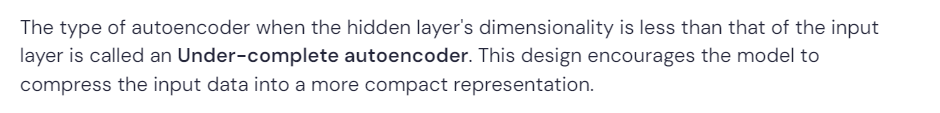
The other statements are not necessarily true or require further clarification:

1. **Autoencoders are less prone to overfitting than PCA**: This statement is not accurate. Autoencoders can be prone to overfitting, especially if they are complex and the dataset is small. In contrast, PCA, a linear method, has fewer parameters to optimize and can be less susceptible to overfitting in certain cases.
2. **Autoencoders are faster and more efficient than PCA**: This statement is generally false. Autoencoders, particularly deep neural networks, can be computationally intensive and may take longer to train compared to PCA, which is computationally efficient for linear projections.
3. **Autoencoders require fewer input data than PCA**: This statement is misleading. The amount of input data needed depends on the complexity of the autoencoder architecture and the data distribution. In many cases, autoencoders may require a significant amount of data to generalize well, especially if they have many layers or parameters.

**Question 5:**

What type of autoencoder is it when the hidden layer's dimensionality is less than that of the input layer?

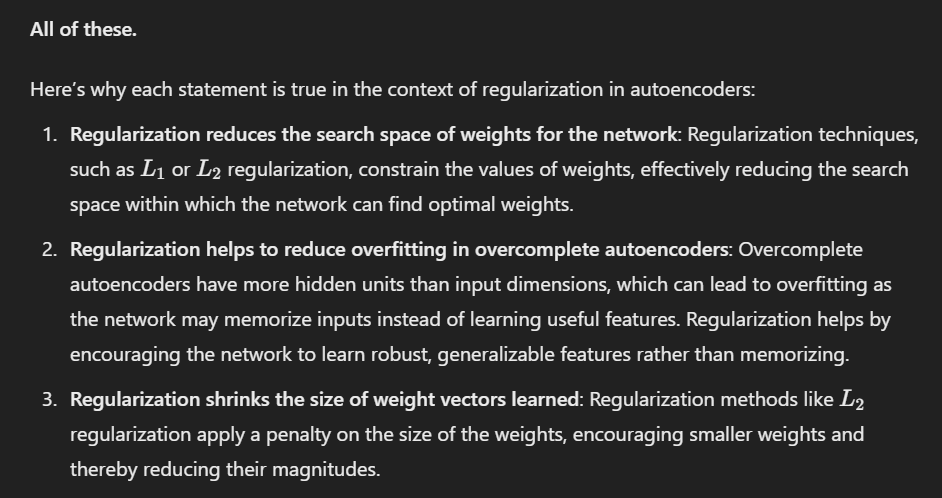
* Under-complete autoencoder
* Complete autoencoder
* Overcomplete autoencoder
* Sparse autoencoder



**Question 6:**

Which of the following statements about regularization in autoencoders is always true?

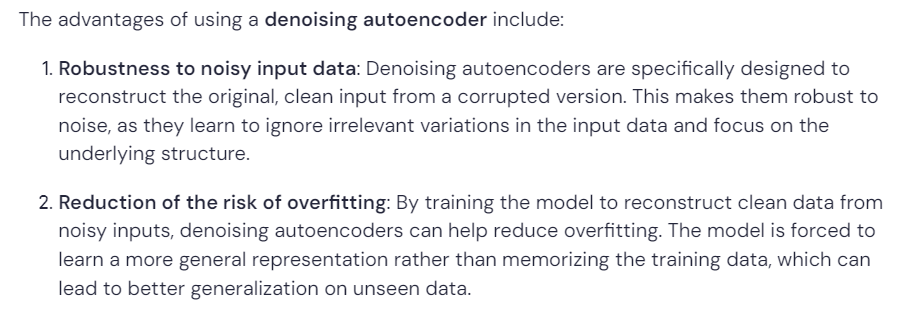
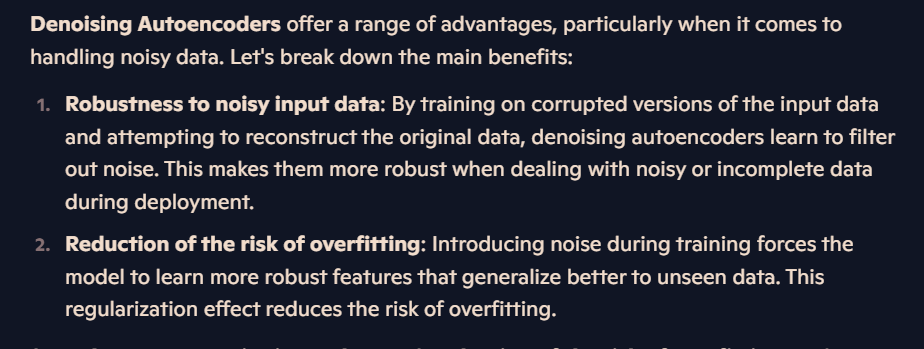
* Regularisation reduces the search space of weights for the network.
* Regularisation helps to reduce the overfitting in overcomplete autoencoders.
* Regularisation shrinks the size of weight vectors learned.
* All of these.



**Question 7:**

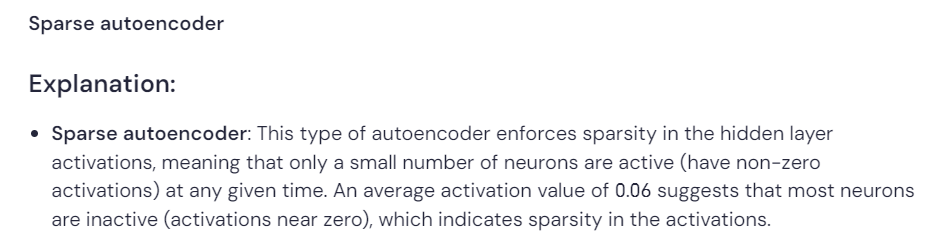
**What are the advantages of using a denoising autoencoder?**

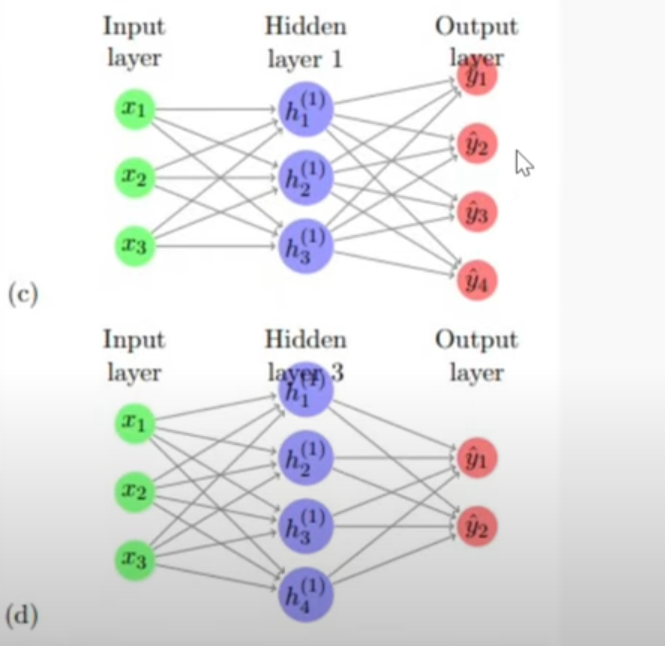
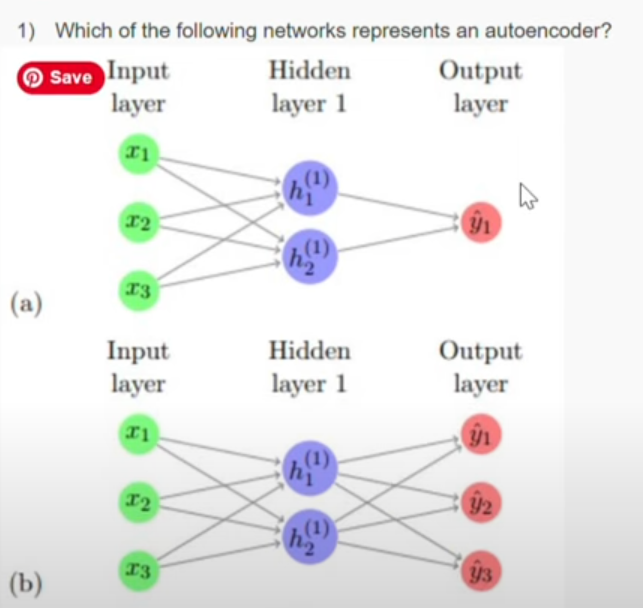
1. Robustness to noisy input data
2. Reduction of the risk of overfitting
3. Faster training time
4. It promotes sparsity in the hidden layer

****

**We are given an autoencoder A. The average activation value of neurons in this network is 0.06. The given autoencoder is:**

1. Contractive autoencoder
2. Overcomplete neural network
3. Sparse autoencoder
4. Denoising autoencoder





Answer (B)

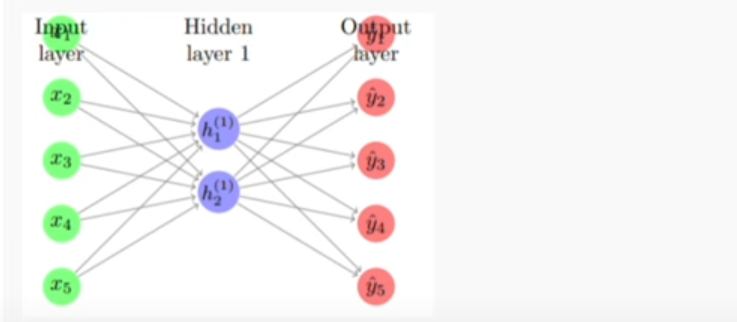
What is the primary reason for adding corruption to the input data in a denoising autoencoder?

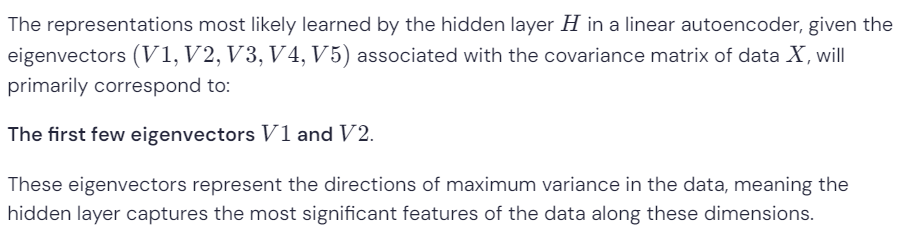
* To increase the complexity of the model.
* To improve the model's ability to generalize to unseen data.
* To reduce the size of the training dataset.
* To increase the training time.

The correct answer is that adding corruption to the input data primarily aims **to improve the model's ability to generalize to unseen data**.

**Question 7:**

We are using the following autoencoder with linear encoder and linear decoder. The eigenvectors associated with the covariance matrix of our data X is (V1, V2, V3, V4, V5). What are the representations most likely to be learned by our hidden layer H? (Eigenvectors are written in decreasing order to the eigenvalues associated with them)

****

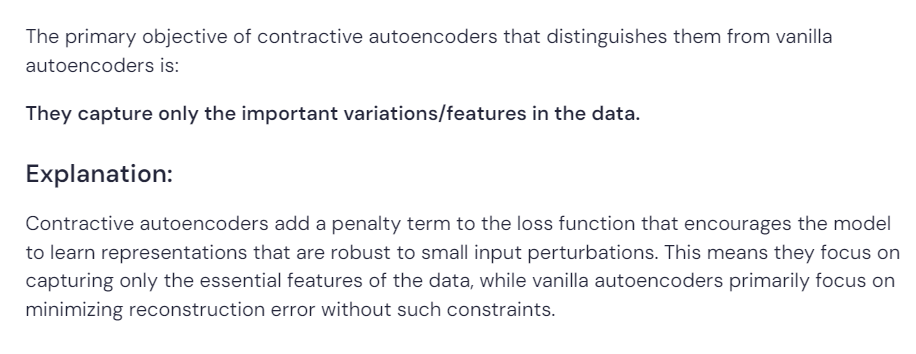
****

1. **What is the purpose of a decoder in an autoencoder?**

* To reconstruct the input data
* To generate new data
* To compress the input data
* To extract features from the input data
* **Answer: To reconstruct the input data.**

What is the primary objective of contractive autoencoders that distinguishes them from vanilla autoencoders?

* They learn a low-dimensional representation of the input data
* They minimize the reconstruction error between the input and the output
* They capture only the important variations/features in the data
* They maximize the mutual information between the input and the output



**Question 10:**

What are the possible applications of autoencoders?

* Data Compression
* Extraction of important features
* Reducing noise
* All of these

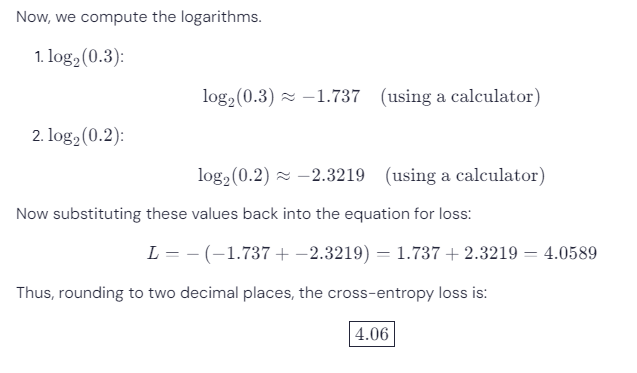
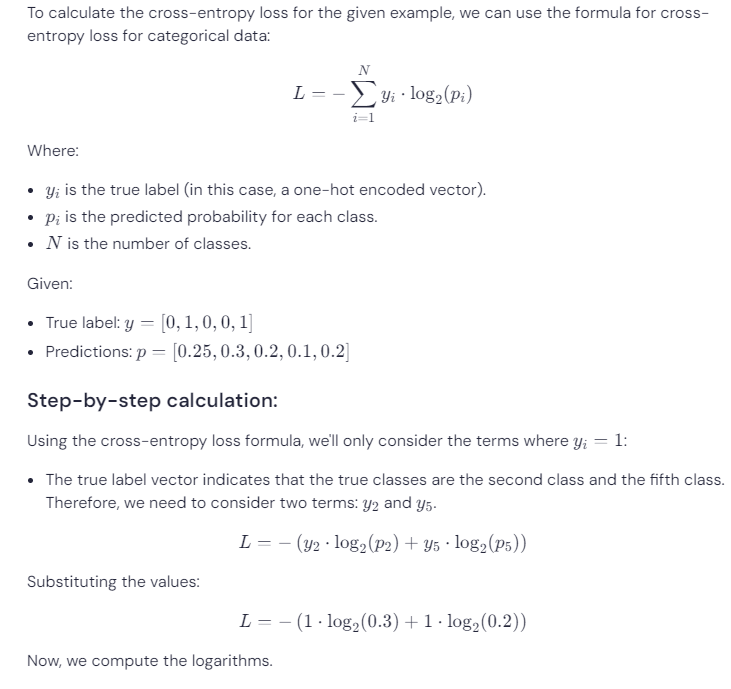
Autoencoders can be applied for:

* **Data Compression**: Reducing the dimensionality of data while preserving important information.
* **Extraction of important features**: Learning useful representations for various tasks.
* **Reducing noise**: Denoising input data by reconstructing clean versions from noisy inputs.

Thus, the correct answer is **All of these**.

The question is:

1. Suppose we build a neural network for a 5-class classification task. Suppose for a single training example, the true label is [0 1 0 0 1] while the predictions by the neural network are [0.25 0.3 0.2 0.1 0.2]. What would be the value of cross-entropy loss for this example? (Answer up to two decimal places, Use base 2 for log-related calculations)

****

Question 7:

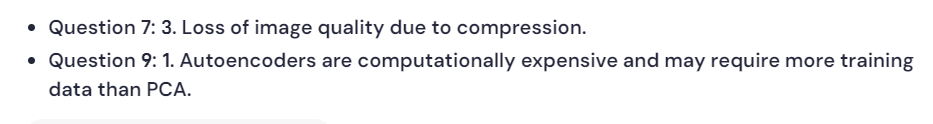
Which of the following problems prevents us from using autoencoders for the task of image compression?

1. Images are not allowed as input to autoencoders
2. Difficulty in training deep neural networks
3. Loss of image quality due to compression
4. Auto encoders are not capable of producing image output

Question 9:

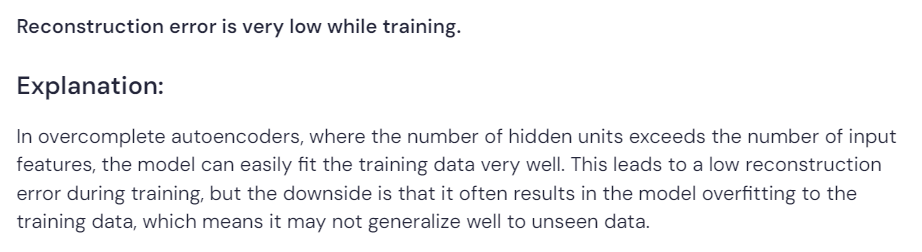
Which of the following is a potential disadvantage of using autoencoders for dimensionality reduction over PCA?

1. Autoencoders are computationally expensive and may require more training data than PCA.
2. Autoencoders are bad at capturing complex relationships in data
3. Autoencoders may overfit the training data and generalize poorly to new data.
4. Autoencoders are unable to handle linear relationships between data.

****

Which of the following statements about overfitting in overcomplete autoencoders is true?

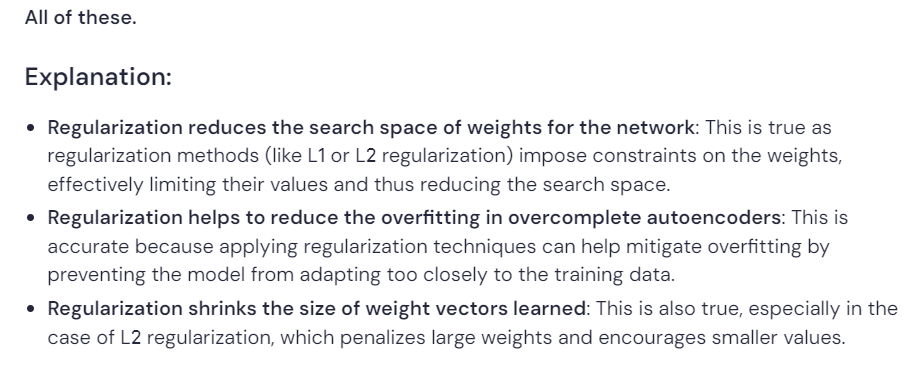
1. Reconstruction error is very low while training
2. Reconstruction error is very high while training
3. Network fails to learn good representations of input
4. Network learns good representations of input



Question 4:

Which of the following statements about regularization in autoencoders is always true?

1. Regularisation reduces the search space of weights for the network.
2. Regularisation helps to reduce the overfitting in overcomplete autoencoders.
3. Regularisation shrinks the size of weight vectors learned.
4. All of these.

****