

# CertNexus

AIP-210

*CertNexus Certified Artificial Intelligence Practitioner (CAIP)*

**For More Information – Visit link below:**

**<https://www.examsempire.com/>**

## **Product Version**

- 1. Up to Date products, reliable and verified.**
- 2. Questions and Answers in PDF Format.**



**<https://examsempire.com/>**

---

# Latest Version: 6.0

## Question: 1

You train a neural network model with two layers, each layer having four nodes, and realize that the model is underfit. Which of the actions below will NOT work to fix this underfitting?

- A. Add features to training data
- B. Get more training data
- C. Increase the complexity of the model
- D. Train the model for more epochs

**Answer: B**

Explanation:

Underfitting is a problem that occurs when a model learns too little from the training data and fails to capture the underlying complexity or structure of the data. Underfitting can result from using insufficient or irrelevant features, a low complexity of the model, or a lack of training data.

Underfitting can reduce the accuracy and generalization of the model, as it may produce oversimplified or inaccurate predictions. Some of the ways to fix underfitting are:

Add features to training data: Adding more features or variables to the training data can help increase the information and diversity of the data, which can help the model learn more complex patterns and relationships.

Increase the complexity of the model: Increasing the complexity of the model can help increase its expressive power and flexibility, which can help it fit better to the data. For example, adding more layers or nodes to a neural network can increase its complexity.

Train the model for more epochs: Training the model for more epochs can help increase its learning ability and convergence, which can help it optimize its parameters and reduce its error.

Getting more training data will not work to fix underfitting, as it will not change the complexity or structure of the data or the model. Getting more training data may help with overfitting, which is when a model learns too much from the training data and fails to generalize well to new or unseen data.

## Question: 2

Which of the following is NOT an activation function?

- A. Additive
- B. Hyperbolic tangent
- C. ReLU
- D. Sigmoid

---

**Answer: A**

Explanation:

An activation function is a function that determines the output of a neuron in a neural network based on its input. An activation function can introduce non-linearity into a neural network, which allows it to model complex and non-linear relationships between inputs and outputs.

Some of the common activation functions are:

Sigmoid: A sigmoid function is a function that maps any real value to a value between 0 and 1. It has an S-shaped curve and is often used for binary classification or probability estimation.

Hyperbolic tangent: A hyperbolic tangent function is a function that maps any real value to a value between -1 and 1. It has a similar shape to the sigmoid function but is symmetric around the origin. It is often used for regression or classification problems.

ReLU: A ReLU (rectified linear unit) function is a function that maps any negative value to 0 and any positive value to itself. It has a piecewise linear shape and is often used for hidden layers in deep neural networks.

Additive is not an activation function, but rather a term that describes a property of some functions. Additive functions are functions that satisfy the condition  $f(x+y) = f(x) + f(y)$  for any  $x$  and  $y$ . Additive functions are linear functions, which means they have a constant slope and do not introduce non-linearity.

### Question: 3

Which of the following items should be included in a handover to the end user to enable them to use and run a trained model on their own system? (Select three.)

- A. Information on the folder structure in your local machine
- B. Intermediate data files
- C. Link to a GitHub repository of the codebase
- D. README document
- E. Sample input and output data files

**Answer: C D E**

Explanation:

A handover is the process of transferring the ownership and responsibility of an ML system from one party to another, such as from the developers to the end users. A handover should include all the necessary information and resources that enable the end users to use and run a trained model on their own system. Some of the items that should be included in a handover are:

Link to a GitHub repository of the codebase: A GitHub repository is an online platform that hosts the source code and version control of an ML system. A link to a GitHub repository can provide the end users with access to the latest and most updated version of the codebase, as well as the history and documentation of the changes made to the code.

README document: A README document is a text file that provides an overview and instructions for an ML system. A README document can include information such as the

---

purpose, features, requirements, installation, usage, testing, troubleshooting, and license of the system.

Sample input and output data files: Sample input and output data files are data files that contain examples of valid inputs and expected outputs for an ML system. Sample input and output data files can help the end users understand how to use and run the system, as well as verify its functionality and performance.

### Question: 4

Which of the following pieces of AI technology provides the ability to create fake videos?

- A. Generative adversarial networks (GAN)
- B. Long short-term memory (LSTM) networks
- C. Recurrent neural networks (RNN)
- D. Support-vector machines (SVM)

**Answer: A**

Explanation:

Generative adversarial networks (GAN) are a type of AI technology that can create fake videos, images, audio, or text that are realistic and indistinguishable from real ones. GAN consist of two neural networks: a generator and a discriminator. The generator tries to produce fake samples from random noise, while the discriminator tries to distinguish between real and fake samples. The two networks compete against each other in a game-like scenario, where the generator tries to fool the discriminator and the discriminator tries to catch the generator. Through this process, both networks improve their abilities until they reach an equilibrium where the generator can produce convincing fakes.

### Question: 5

Which database is designed to better anticipate and avoid risks of AI systems causing safety, fairness, or other ethical problems?

- A. Asset
- B. Code Repository
- C. Configuration Management
- D. Incident

**Answer: D**

Explanation:

---

An incident database is a database that is designed to better anticipate and avoid risks of AI systems causing safety, fairness, or other ethical problems. An incident database collects and stores information about incidents or events where AI systems have caused or contributed to negative outcomes or harms, such as accidents, errors, biases, discriminations, or violations. An incident database can help identify patterns, trends, causes, impacts, and solutions for AI-related incidents, as well as provide guidance and best practices for preventing or mitigating future incidents.

### Question: 6

What is the open framework designed to help detect, respond to, and remediate threats in ML systems?

- A. Adversarial ML Threat Matrix
- B. MITRE ATT&CK® Matrix
- C. OWASP Threat and Safeguard Matrix
- D. Threat Susceptibility Matrix

**Answer: A**

Explanation:

The Adversarial ML Threat Matrix is an open framework designed to help detect, respond to, and remediate threats in ML systems. The Adversarial ML Threat Matrix is inspired by the MITRE ATT&CK® Matrix<sup>1</sup>, which is a framework for describing cyberattacks across various stages of an attack lifecycle. The Adversarial ML Threat Matrix adapts this framework to address specific threats and vulnerabilities in ML systems, such as data poisoning, model stealing, model evasion, or model inversion<sup>2</sup>. The Adversarial ML Threat Matrix provides a structured way to organize and classify adversarial techniques, tactics, procedures, examples, and mitigations for ML systems<sup>2</sup>.

### Question: 7

Which two techniques are used to build personas in the ML development lifecycle? (Select two.)

- A. Population estimates
- B. Population regression
- C. Population resampling
- D. Population triage
- E. Population variance

**Answer: A D**

Explanation:

Personas are fictional characters that represent the potential users or customers of an ML system. Personas can help understand the needs, goals, preferences, and behaviors of the target audience, as well as design and evaluate the system from their perspective. Some of the techniques that are used to build personas in the ML development lifecycle are:

Population estimates: Population estimates are statistical methods that estimate the size, characteristics, and distribution of a population based on a sample or a census. Population estimates can help identify and quantify the potential market segments and user groups for an ML system, as well as their demographics, locations, and behaviors.

Population triage: Population triage is a process of prioritizing and selecting the most relevant and representative personas for an ML system based on some criteria or metrics. Population triage can help focus on the key user needs and scenarios, as well as avoid creating too many or too few personas.

## Question: 8

Which of the following text vectorization methods is appropriate and correctly defined for an English-to-Spanish translation machine?

- A. Using TF-IDF because in translation machines, we do not care about the order of the words.
- B. Using TF-IDF because in translation machines, we need to consider the order of the words.
- C. Using Word2vec because in translation machines, we do not care about the order of the words.
- D. Using Word2vec because in translation machines, we need to consider the order of the words.

**Answer: D**

Explanation:

Text vectorization is a technique that converts text into numerical vectors that can be used by machine learning models. Text vectorization can use different methods to represent text features, such as word frequency, word order, word meaning, or word context. Some of the common text vectorization methods are:

TF-IDF: TF-IDF (term frequency-inverse document frequency) is a method that assigns a weight to each word based on its frequency in a document and its rarity across a collection of documents. TF-IDF can capture the importance and relevance of words for a given topic or domain, but it does not consider the order or meaning of words.

Word2vec: Word2vec is a method that learns a vector representation for each word based on its context in a large corpus of text. Word2vec can capture the semantic and syntactic similarity and relationships among words, as well as preserve the order of words.

For an English-to-Spanish translation machine, using Word2vec would be appropriate and correctly defined, because in translation machines, we need to consider the order of the words, as well as their meaning and context.

**Thank You for Trying Our Product**

**Special 16 USD Discount Coupon: **NSZUBG3X****

**Email:** [support@examsempire.com](mailto:support@examsempire.com)

**Check our Customer Testimonials and ratings  
available on every product page.**

**Visit our website.**

**<https://examsempire.com/>**