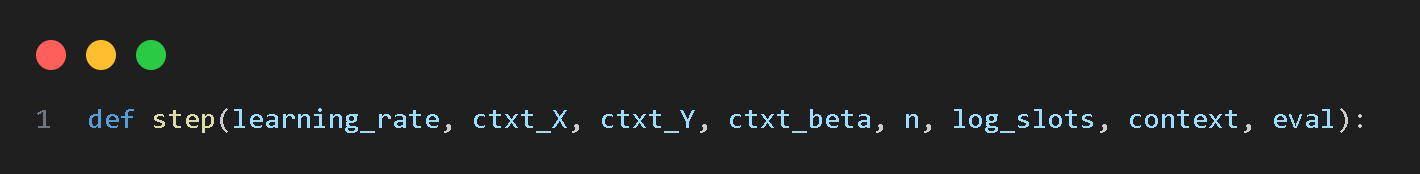
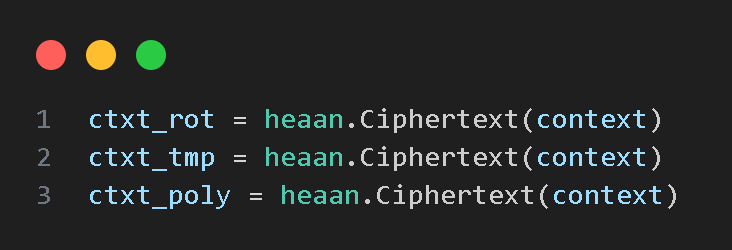
Updated Step Function Explanation with Rationale

This function implements a gradient descent step using homomorphic encryption. Each part of the function is designed to work on encrypted data to maintain privacy while performing machine learning operations. Below is a detailed explanation of each part of the function, including the rationale for the choices made.



* learning\_rate: The rate at which the model updates its parameters.
* ctxt\_X: Encrypted features of the dataset.
* ctxt\_Y: Encrypted labels of the dataset.
* ctxt\_beta: Encrypted model parameters.
* n: Number of elements to process.
* log\_slots: Logarithm of the number of slots used in encryption.
* context: Encryption context.
* eval: Evaluator for performing homomorphic operations.



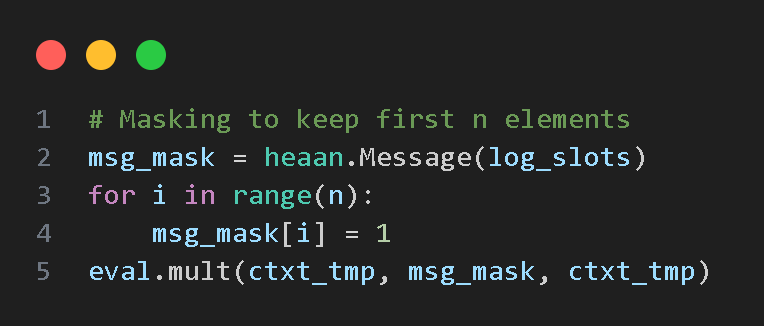
* ctxt\_rot, ctxt\_tmp, ctxt\_poly: Temporary ciphertexts used for intermediate calculations.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

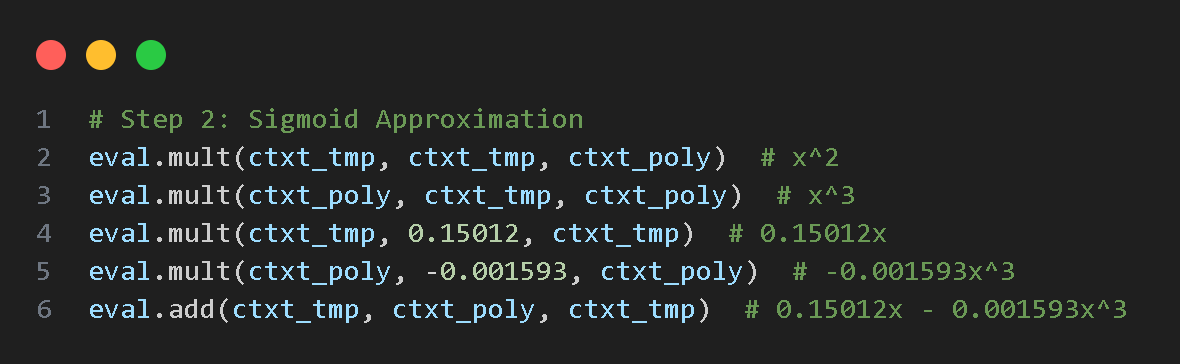
Rationale: Efficiently computes the linear combination of beta and X plus beta0 using rotations and additions to reduce the number of operations.

* ctxt\_beta0: Holds rotated ctxt\_beta for efficient linear combination.
* eval.mult: Multiplies ctxt\_beta with ctxt\_X to combine features with parameters.
* eval.left\_rotate: Rotates the ciphertext to align elements for summation, leveraging the structure of homomorphic encryption to perform parallel operations.
* eval.add: Adds rotated and multiplied ciphertexts to accumulate results, efficiently summing components for the linear combination.



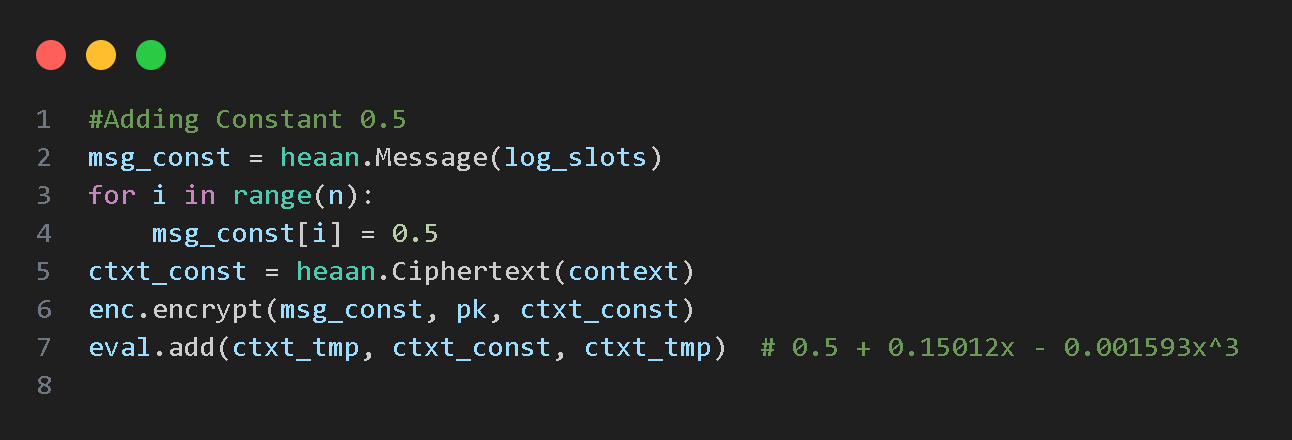
Rationale: Ensures that only the first n elements are affected by the computations, which is crucial for maintaining the structure of the encrypted data.

* msg\_mask: A mask message to isolate the first n elements, allowing selective operations on specific parts of the ciphertext.
* eval.mult: Applies the mask to the result, ensuring that only relevant data points are updated.



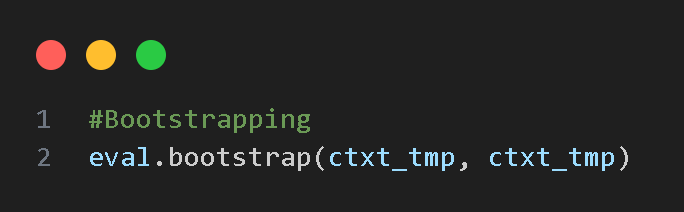
Rationale: Approximates the sigmoid function using a polynomial. The sigmoid function is key for logistic regression, but direct computation is not feasible in encrypted form, so polynomial approximation is used.

* Polynomial approximation: Uses the polynomial 0.5+0.15012𝑥−0.001593𝑥\*\*3 to approximate the sigmoid function because polynomials can be efficiently evaluated homomorphically.



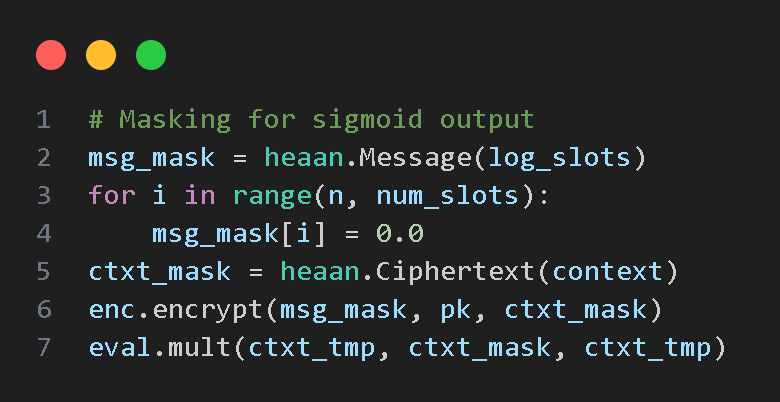
Rationale: The polynomial approximation of the sigmoid needs a constant term of 0.5. Adding this constant ensures the approximation correctly matches the sigmoid behavior.

* msg\_const: Message with constant value 0.5 for adding to the polynomial result.
* eval.add: Adds the constant to the polynomial result, finalizing the sigmoid approximation.



Rationale: Bootstrapping refreshes the ciphertext to maintain precision and allows further computations without significant noise accumulation.

* eval.bootstrap: Refreshes the ciphertext to maintain precision and computational integrity.



Rationale: Ensures that only valid elements are retained in the result after sigmoid computation, similar to earlier masking.

* msg\_mask: Mask message to isolate valid elements, ensuring that operations only affect relevant parts of the data.
* eval.mult: Applies the mask to the sigmoid result, preserving only the necessary components.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Rationale: Adjusts the learning rate dynamically and computes the gradient, which is essential for updating model parameters. The rotations and summations efficiently combine elements for the gradient calculation.

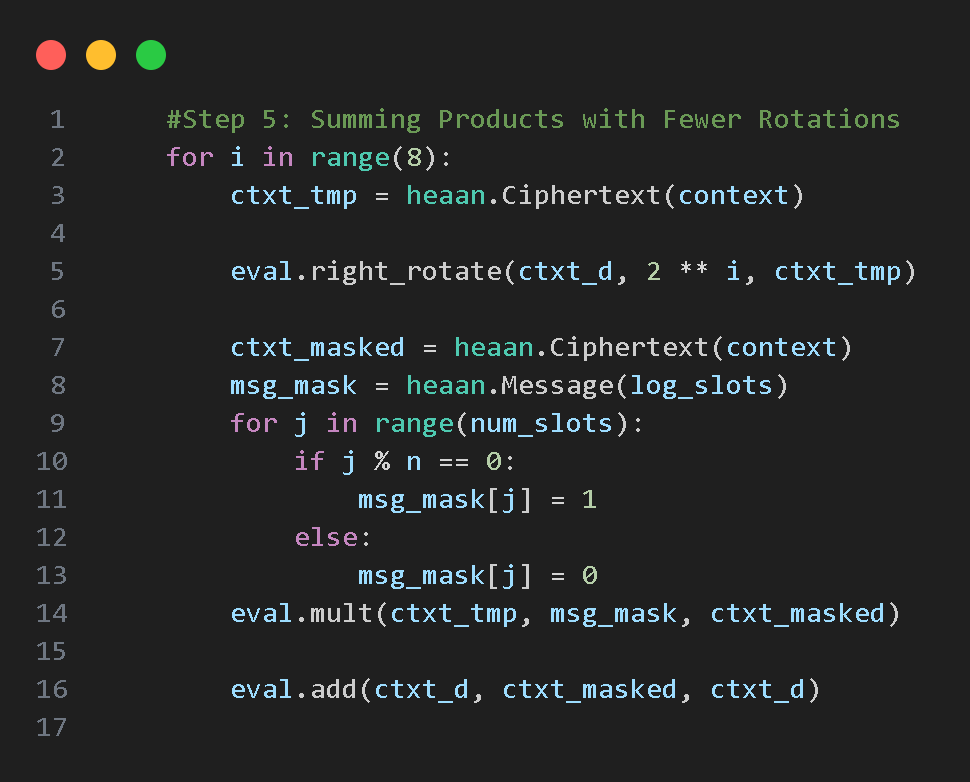
* dynamic\_learning\_rate: Adjusted learning rate to control the step size in gradient descent, preventing overshooting and improving convergence.
* eval.sub: Subtracts predicted values from actual labels to compute the error.
* eval.mult: Scales the error by the dynamic learning rate and divides by n to normalize the gradient.
* eval.right\_rotate: Rotates for efficient summation, aligning elements to facilitate cumulative addition.
* eval.add: Accumulates rotated results, ensuring the gradient is correctly computed.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

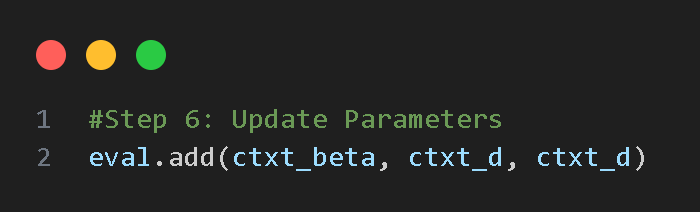
Rationale: Incorporates features into the gradient computation and adds noise for differential privacy. Noise addition ensures privacy while performing homomorphic operations.

* ctxt\_X\_j: Intermediate ciphertext for feature vector, selectively combining relevant features.
* msg\_X0: Mask for feature extraction, ensuring only the relevant feature components are considered.
* noise: Adds random noise to feature vector to enhance privacy, preventing exact recovery of original data.
* eval.add: Adds noise to feature vector, ensuring privacy-preserving computations.
* eval.mult: Multiplies gradient by feature vector to compute the effective gradient component.



Rationale: Efficiently sums products over all elements using fewer rotations, reducing computational overhead and maintaining precision.

* eval.right\_rotate: Rotates to align elements for summation, leveraging the structure of homomorphic encryption.
* msg\_mask: Mask for selective summation, ensuring only relevant components are combined.
* eval.mult: Applies the mask, isolating the necessary components for summation.
* eval.add: Accumulates results, efficiently combining elements to finalize the gradient computation.



Rationale: Updates the model parameters using the computed gradient, completing the gradient descent step.

* eval.add: Updates the model parameters with the computed gradient, ensuring the model learns from the data.

텍스트, 스크린샷, 폰트, 그래픽이(가) 표시된 사진

자동 생성된 설명

Rationale: Returns the updated model parameters for the next iteration of gradient descent.

* ctxt\_d: Updated model parameters.

Enhanced Encryption Function Explanation

The enhanced\_encrypt function is designed to encrypt a message using homomorphic encryption and add noise to the ciphertext for differential privacy. Below is a detailed explanation of each part of the function, including the rationale for the choices made.

스크린샷, 텍스트, 폰트, 그래픽이(가) 표시된 사진

자동 생성된 설명

* msg: The plaintext message to be encrypted.
* enc: The encryptor object used for encryption operations.
* eval: The evaluator object used for performing homomorphic operations.
* sk: The secret key used for encryption.
* noise\_level: The standard deviation of the Gaussian noise to be added to the ciphertext (default is 0.1).

스크린샷, 텍스트, 폰트, 그래픽이(가) 표시된 사진

자동 생성된 설명

* ctxt: Initializes an empty ciphertext object to hold the encrypted message.

텍스트, 스크린샷, 폰트, 그래픽이(가) 표시된 사진

자동 생성된 설명

* enc.encrypt: Encrypts the plaintext message msg using the secret key sk. This ensures that the message is securely encrypted and can only be decrypted by someone with the corresponding secret key.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Rationale: Adding noise to the encrypted message helps to protect the privacy of the data by making it harder to infer the original plaintext, even if some aspects of the encryption are compromised.

* noise: Creates a message with Gaussian noise. The noise level is determined by the noise\_level parameter, with a default standard deviation of 0.1.
* np.random.normal(0, noise\_level): Generates Gaussian noise with a mean of 0 and a standard deviation specified by noise\_level.
* ctxt\_noise: Initializes an empty ciphertext object to hold the encrypted noise.
* enc.encrypt: Encrypts the noise message using the secret key sk, ensuring that the noise is also securely encrypted.

텍스트, 스크린샷, 폰트, 그래픽이(가) 표시된 사진

자동 생성된 설명

* eval.add: Adds the encrypted noise to the encrypted message. This operation ensures that the resulting ciphertext contains both the original message and the noise, providing an additional layer of security.

텍스트, 스크린샷, 폰트, 그래픽이(가) 표시된 사진

자동 생성된 설명

* ctxt: Returns the final encrypted ciphertext that includes the original message and the added noise.

‘generate\_qr\_code’ Function Explanation

The generate\_qr\_code function generates a QR code that includes encrypted data and a digital signature. This function uses homomorphic encryption to secure the data and then encodes the encrypted data into a QR code. Below is a detailed explanation of each part of the function, including the rationale for the choices made.

스크린샷, 텍스트, 폰트, 그래픽이(가) 표시된 사진

자동 생성된 설명

* data: The data to be encoded in the QR code.
* authorized: A flag indicating whether the QR code generation is authorized. Defaults to True.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* parsed\_url: Parses the input URL to extract its components.
* domain\_name: Extracts the domain name from the parsed URL.
* changedDomainName: Sanitizes the domain name to make it a valid filename by replacing non-alphanumeric characters with underscores.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* msg: Initializes a message with random values. This message will be encrypted and used as a part of the QR code data.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* enhanced\_encrypt: Encrypts the message using the enhanced\_encrypt function, which adds noise for differential privacy.
* ctxt\_X, ctxt\_Y, ctxt\_beta: Encrypted ciphertexts representing features, labels, and model parameters, respectively.
* n: Number of elements to process. Set to 10 in this case.

스크린샷, 텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명

* step: Executes the gradient descent step function to compute an encrypted signature.
* encrypted\_signature: Resulting ciphertext after performing the gradient descent step.
* signature\_b64: Base64-encoded string of the encrypted signature, converted for embedding in the QR code.

스크린샷, 텍스트, 폰트이(가) 표시된 사진

자동 생성된 설명

* combined\_data: Combines the original data and the encrypted signature into a JSON string, preparing it for encoding in the QR code.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* qrcode.QRCode: Initializes the QR code object.
* version: Controls the size of the QR code.
* error\_correction: Level of error correction. ERROR\_CORRECT\_L allows for 7% error correction.
* box\_size: Size of each box in the QR code.
* border: Border size around the QR code.
* add\_data: Adds the combined data to the QR code.
* make\_image: Generates the QR code image with specified fill and background colors.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* image\_width, image\_height: Defines the dimensions of the final image.
* img: Creates a blank image with the specified dimensions and a white background.
* qr\_width, qr\_height: Dimensions of the QR code image.
* qr\_position: Calculates the position to paste the QR code onto the blank image, centering it.
* paste: Pastes the QR code onto the blank image at the calculated position.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* text: Determines the text to be displayed based on the authorized flag.
* font\_size: Size of the font for the text.
* font: Font object for drawing text.
* draw: Draw object to draw on the image.
* text\_width, text\_height: Dimensions of the text to be drawn.
* text\_position: Calculates the position to center the text horizontally within the image.
* draw.text: Draws the text on the image at the calculated position.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* img.save: Saves the final image with the QR code and text to a file named after the sanitized domain name.
* print: Outputs a message indicating that the QR code has been generated and saved.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

* try-except block: Catches and handles any exceptions that occur during the QR code generation process, printing an error message if needed.