**TF3, C4, W3**

Quiz

**Question 1** **- multiple choice, shuffle**

For Variational AutoEncoders, which of the following are the correct operations performed in the *latent space*?

A: encoder mean \* encoder STDev + gaussian distribution

Feedback: Incorrect!

\*B: encoder mean + encoder STDev \* gaussian distribution

Feedback: Correct!

C: encoder mean + encoder STDev + gaussian distribution

Feedback: Incorrect!

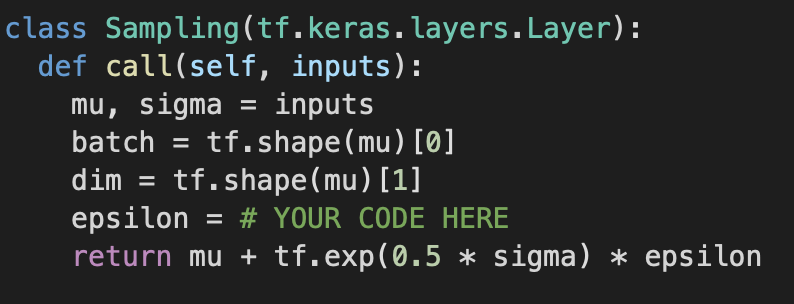
D: encoder mean \* encoder STDev \* gaussian distribution

Feedback: Incorrect!

**Question 2** **- text match**

Consider the following code, which is used in Variational AutoEncoder to represent the latent space. Fill in the missing piece of code.

(**Note:** Use shape as *shape=(batch, dim)* )



\*A: tf.keras.backend.random\_normal(shape=(batch, dim))

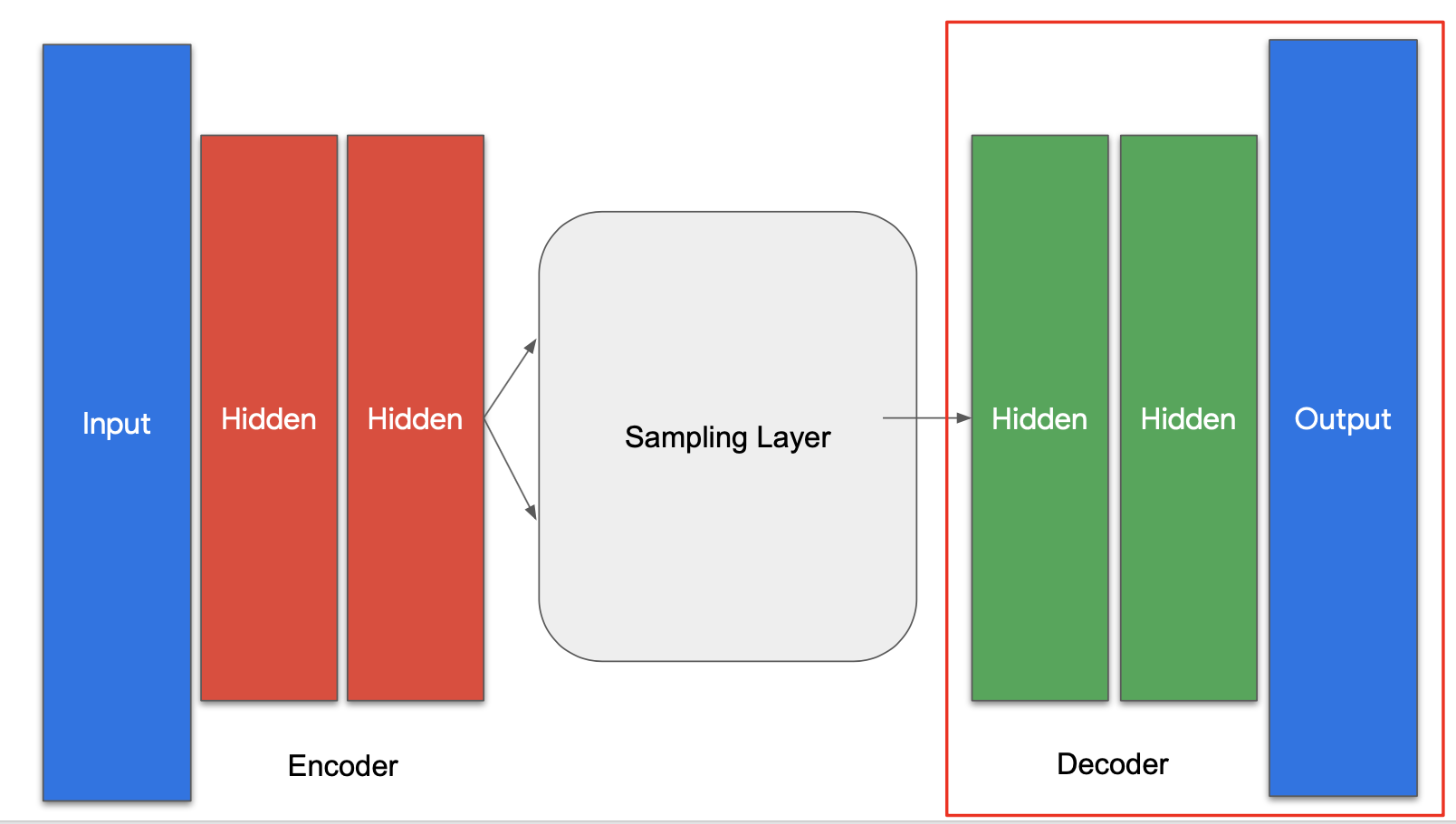
Feedback: Correct!

Default Feedback: Hints: Make sure your spelling is correct  
You have to use tf.*keras.backend*

You have to initialize epsilon with *random normal gaussian distribution*

**Question 3** **- multiple choice, shuffle**

When building the architecture for the decoder for a *convolutional Variational AutoEncoder*, what type of layers will you use ? Below is a screenshot of the code with # layer name # written in place of the actual layer that you would use. What goes in place of # layer name #?





\*A: Conv2DTranspose

Feedback: Correct! This will help you invert the convolutional filters applied during encoding.

B: Conv2D

Feedback: Incorrect! This layer type will not reverse the effects of the encoder

C: MaxPooling2D.

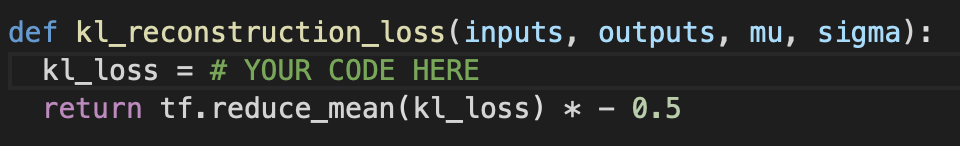
Feedback: Incorrect! This layer type will not reverse the effects of the encoder

D: Global AveragePooling2D

Feedback: Incorrect! This layer type will not reverse the effects of the encoder

**Question 4** **- multiple choice, shuffle**

Fill in the missing code for Kullback-Leibler cost function.



\*A: kl\_loss = 1 + sigma - tf.square(mu) - tf.math.exp(sigma)

Feedback: Correct!

B: kl\_loss = 1 + mu - tf.square(sigma) - tf.math.exp(mu)

Feedback: Incorrect!

C: kl\_loss = sigma - tf.square(mu) - tf.math.exp(sigma)

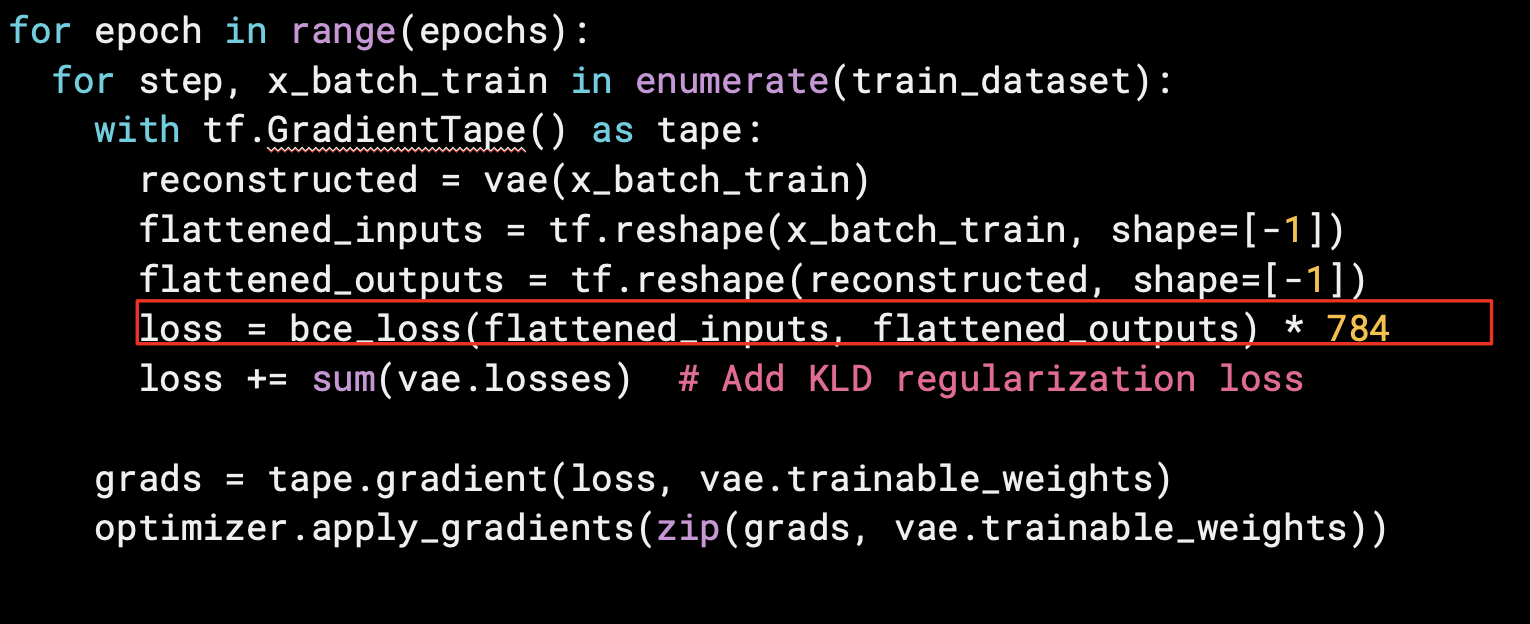
Feedback: Incorrect!

D: mu - tf.square(sigma) - tf.math.exp(mu)

Feedback: Incorrect!

**Question 5** **- multiple choice, shuffle**

Which of the following is true regarding *loss*?



\*A: The closer the values of binary cross entropy loss function are to 1, the closer is the output to expected results.

Feedback: Correct!

B: The closer the values of binary cross entropy loss function are to 0, the closer is the output to expected results.

Feedback: Incorrect! The closer they are to 0 the further we are from the expected results.