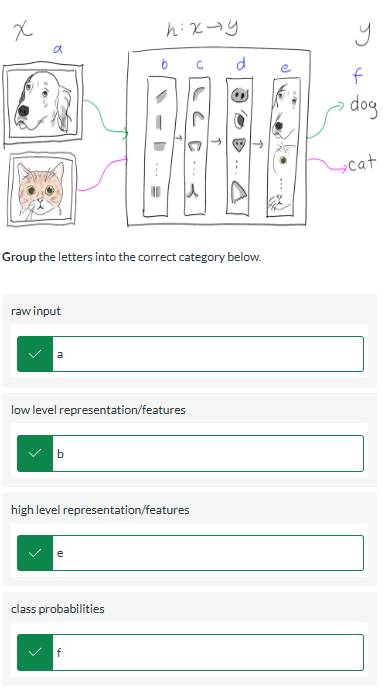
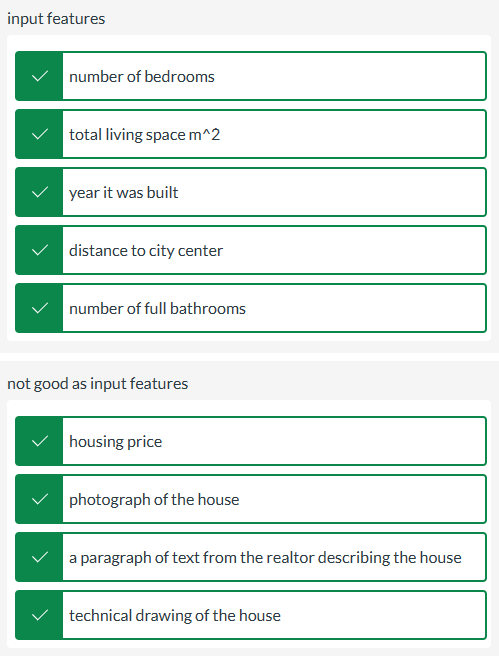
**4.1 & 4.2**

1. Neural networks are **composition** of simple non-linear functions that perform **representation learning**
2. Consider the following idealized diagram of a neural network.

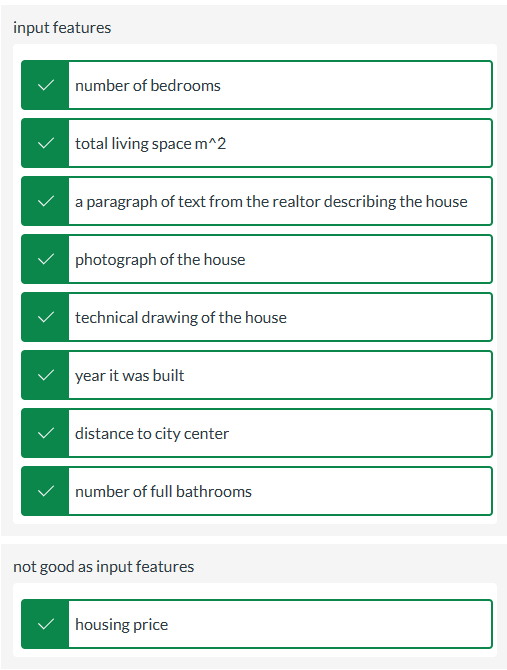


1. You are tasked with training a ML model to predict housing prices in Västerås. Perform the job of a *feature engineer* and **sort** the following according to whether or not they should be used as input features for a 'classic' ML model lik**e *linear regression* or *SVM****. （传统的ML方法）*

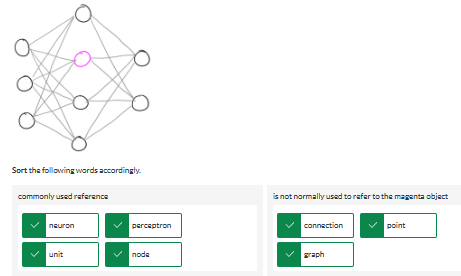
由物体的属性作为可以衡量的input，例如衡量价格，那么价格就是output，影响价格的客观属性就是input

**

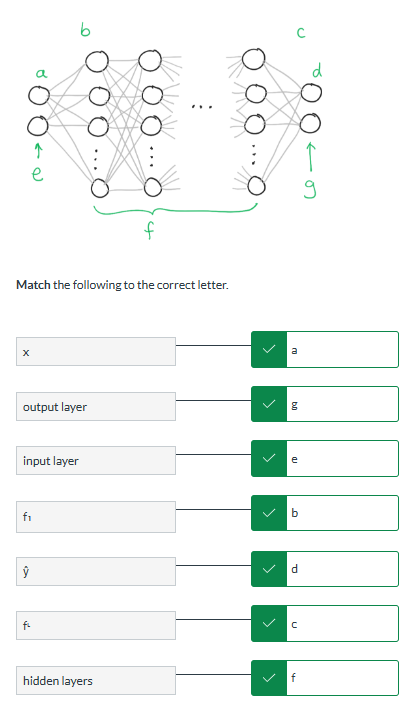
1. You are tasked with training a ML model to predict housing prices in Västerås. **Sort** the following according to whether or not they should be used as input features for a representation learning model like a deep neural network.（神经网络的方法）除了outpu以外都可以是input

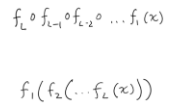


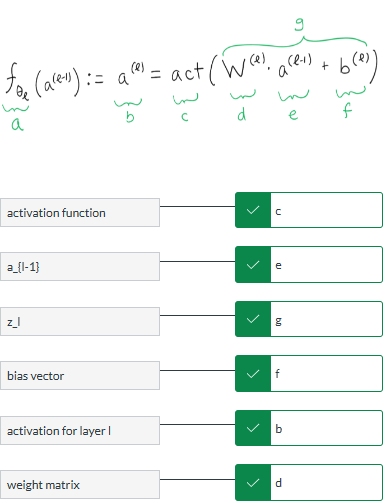
1. One of the fundamental components of a neural network, depicted in magenta below goes by many names.



1. Consider the following diagram of a general feedforward neural network



1. TRUE- In practice, deep learning usually refers to deep neural networks.
2. FALSE - The following expressions are equivalent
3. **Match** the following terms of the function of a layer to the description below.

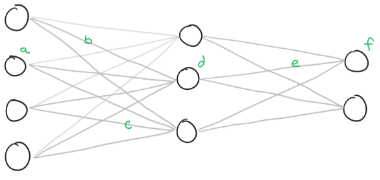


1. **Arrange** the following expressions below to be equivalent to the following

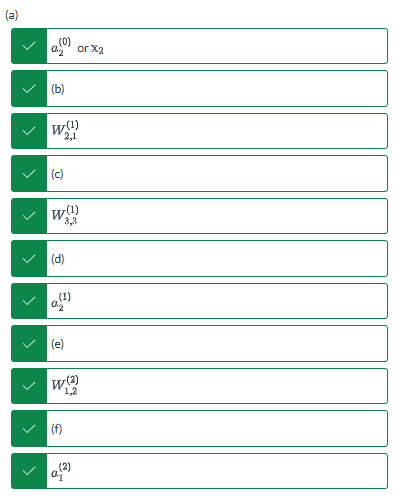
but in terms of activations.   
Hint: use fullscreen mode if you have trouble manipulating the terms.



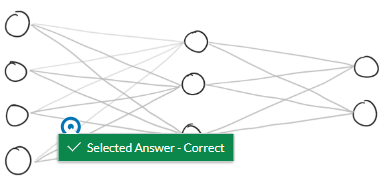
1. Consider this diagram of a neural network



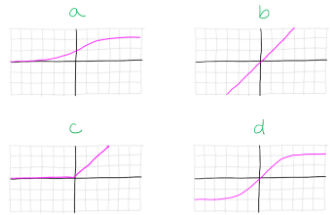
**Arrange** the following *alphabetically starting with (a)* along with the appropriate labels.

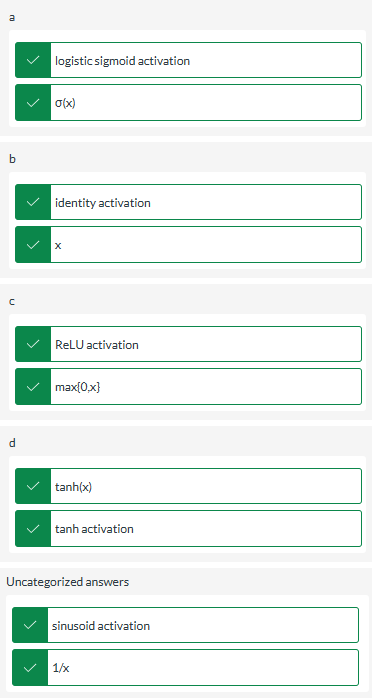


1. **Click** on the diagram to indicate where the following quantity is depicted

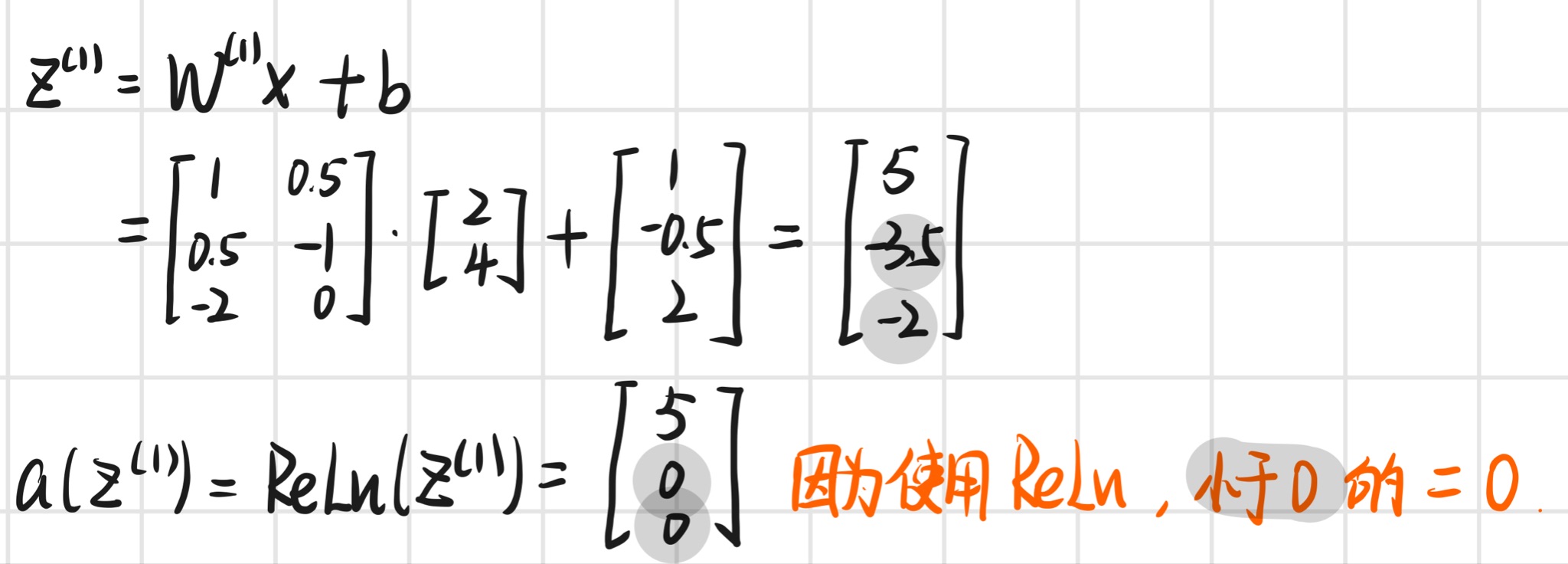
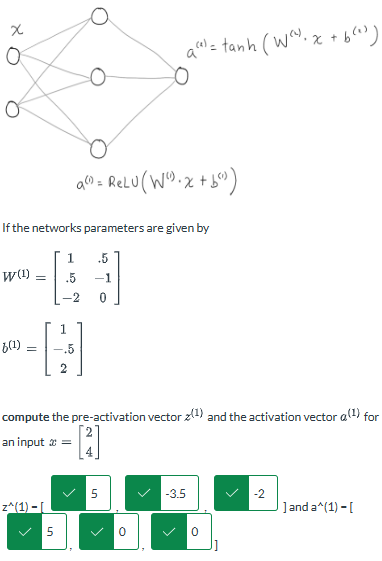


1. **Match** the following correctly.

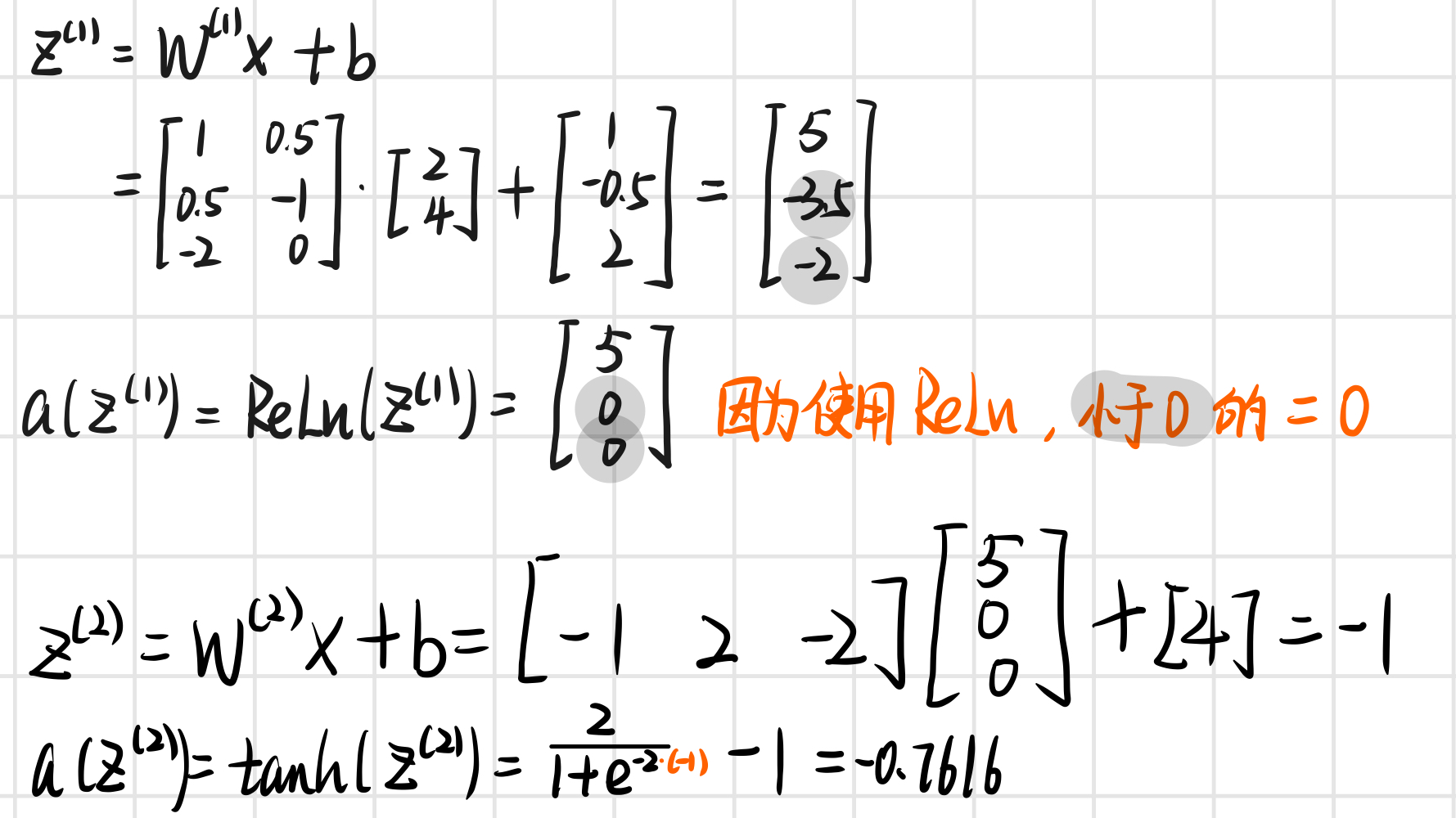
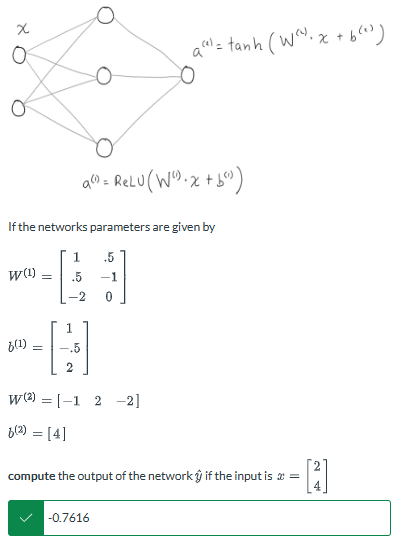




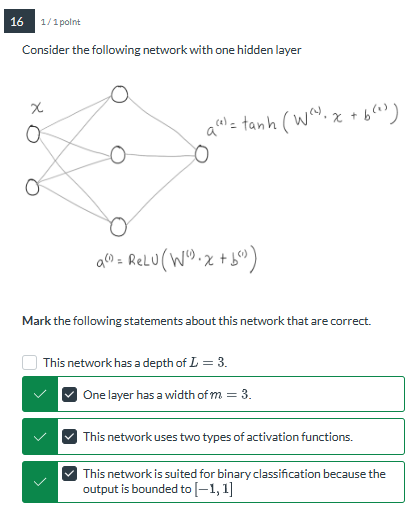
1. Consider the following network with one hidden layer



1. Consider the following network with one hidden layer

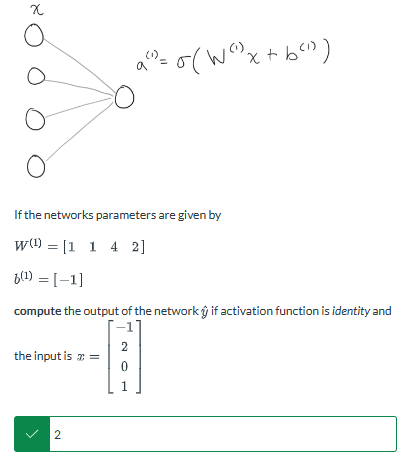


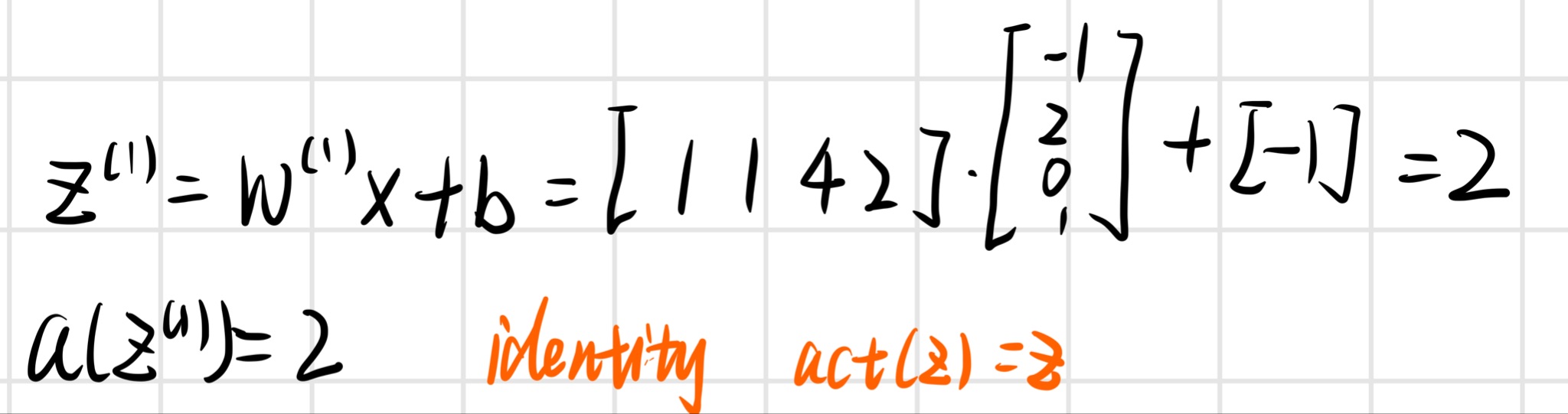
1. Consider the following network



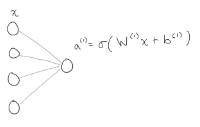
* This network has a depth of L = 3 这里L=2
* One layer has a width of m = 3 因为L1处 width为3
* This network uses two types of activation functions 2个不同的激活函数 一个是ReLu一个是tanh
* This network is suited for binary classification because the output is bounded to [-1, 1] 因为tanh的区间为[-1, 1]

1. Consider the following network





1. Consider the following network



**Select** which of the following methods this network is equivalent to ( is a sigmoid).

* SVM
* perceptron
* linear regression
* logistic regression

1. True - End-to-end learning (or representation learning) methods map directly from raw input to the desired output. This stands in contrast to classic machine learning methods which often requires feature engineering or expert domain knowledge to provide curated useful features as input.
2. Neural networks build new **representations** on top of previous representations in a process known as **compositional learning**. By doing so, they can combine **simple features** like an edge or contour into **complex features**
3. True - The *universal approximation theorem* (1989) states that a sufficiently wide neural network can approximate *any* continuous function.

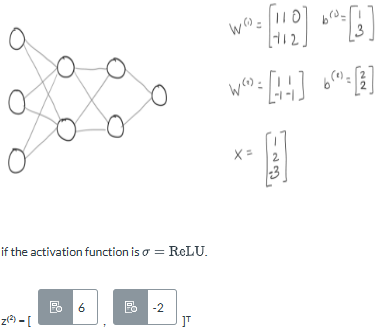
* Neural networks become more expressive as you increase the depth of the network L
* Dropout is a type of regularization for neural networks.
* Regularization methods can help us balance between expressive models and overfitting.
* Batch normalization helps regularize neural networks.
* Neural networks become more expressive as you increase the width of a layer of the network m

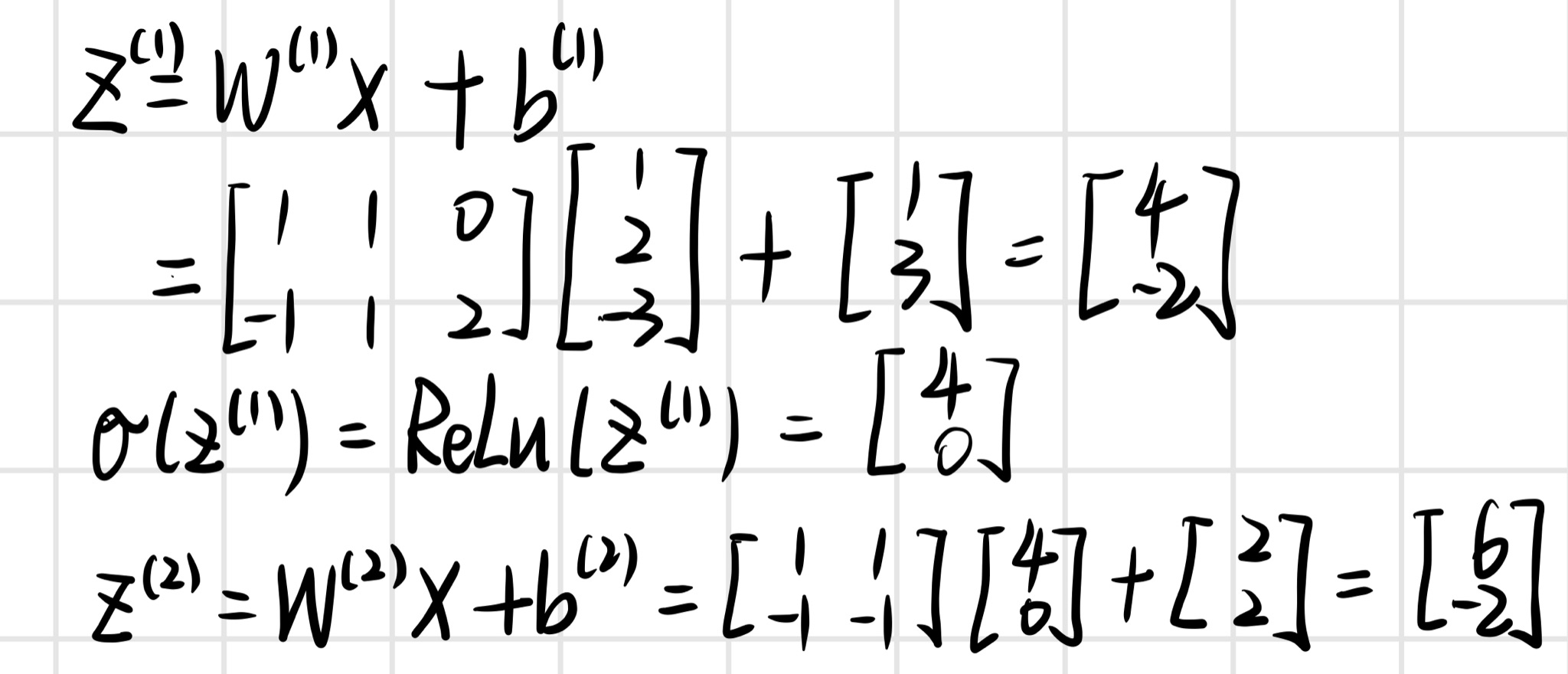
**4.3**

1. What enables backpropagation to efficiently optimize neural networks with very large numbers of parameters?

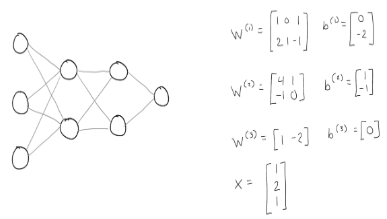
* It updates parameters in *stochastic gradient descent*.
* It iteratively computes gradients starting from the last layer, reusing stored calculations.
* Activations and pre-activations are computed and stored *once* for each node in the forward pass.
* It uses the chain rule to compute gradients.

1. **Compute** the pre-activations for layer ℓ=2 for the following network



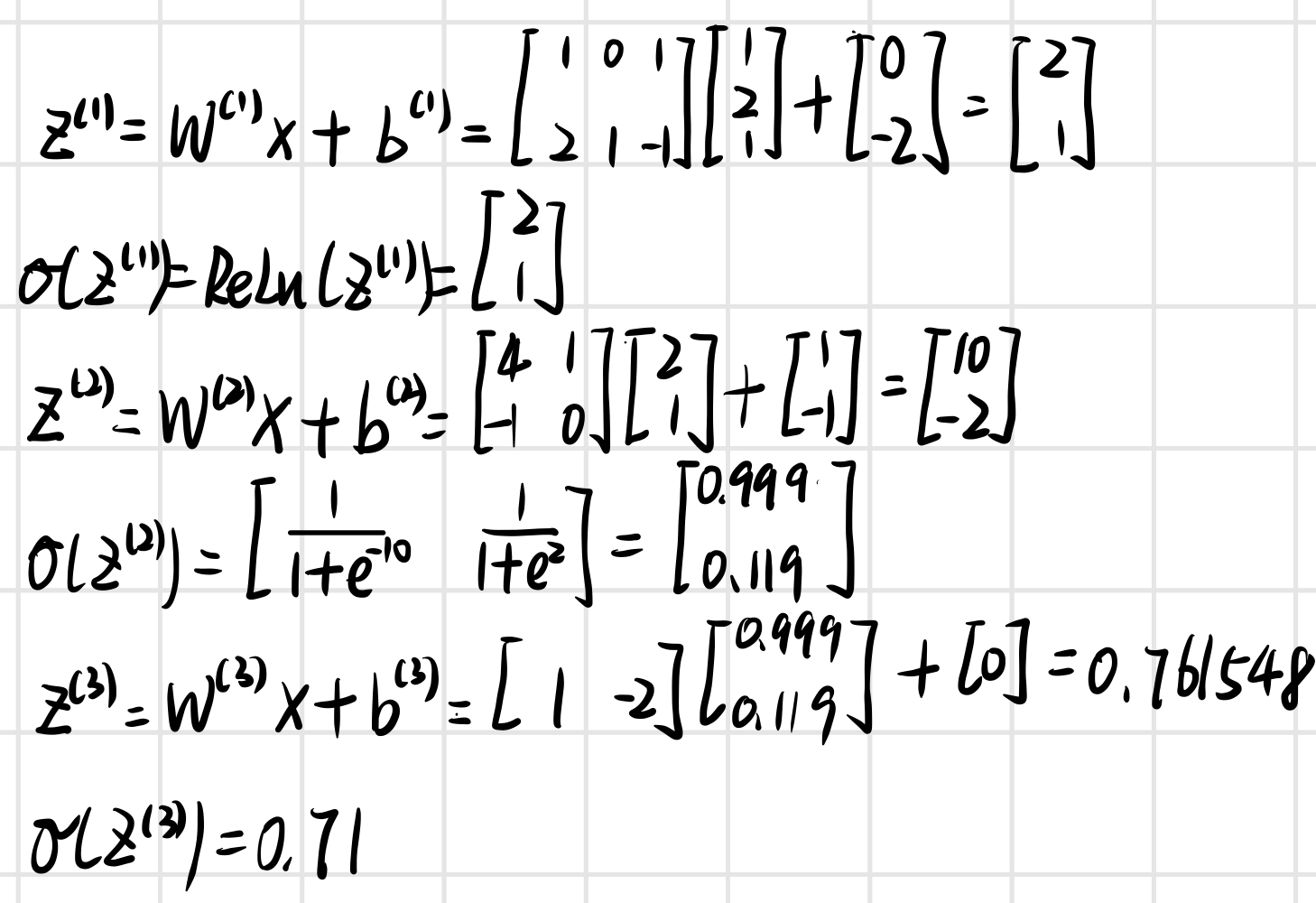


1. **Compute** the output of the network for the following

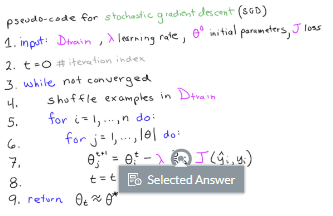


if the activation function for layers 2 and 3 is the *logistic sigmoid* (the activation for layer 1 is *ReLU* as in the last question. Your answer should be correct to 2 decimal places.   
*Hint: you computed the pre-activations for layer 2 in the previous question.*

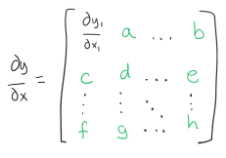
**The network output = 0.71**

****

1. Which step of stochastic gradient descent is backpropagation concerned with?



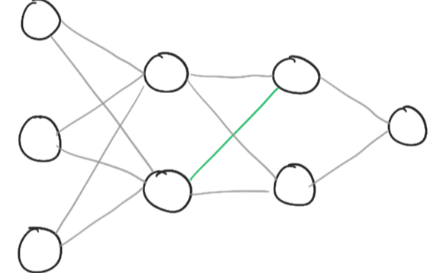
1. **Select** the correct matches below to express the first partial derivative of y or the Jacobian that we use in this course



1. **Mark** the following statements that are correct regarding gradients of common activation functions below.

* The gradient of the *ReLU* function is
* The gradient of the *identity* function is
* The gradient of the *tanh* function is
* The gradient of the *logistic sigmoid* function is

1. Consider the following network

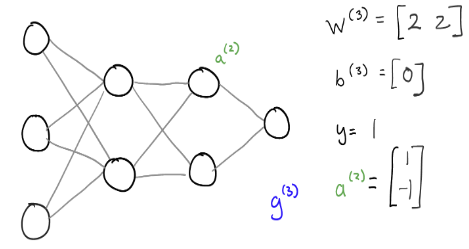


**Arrange** the following terms to form an expression for the derivative

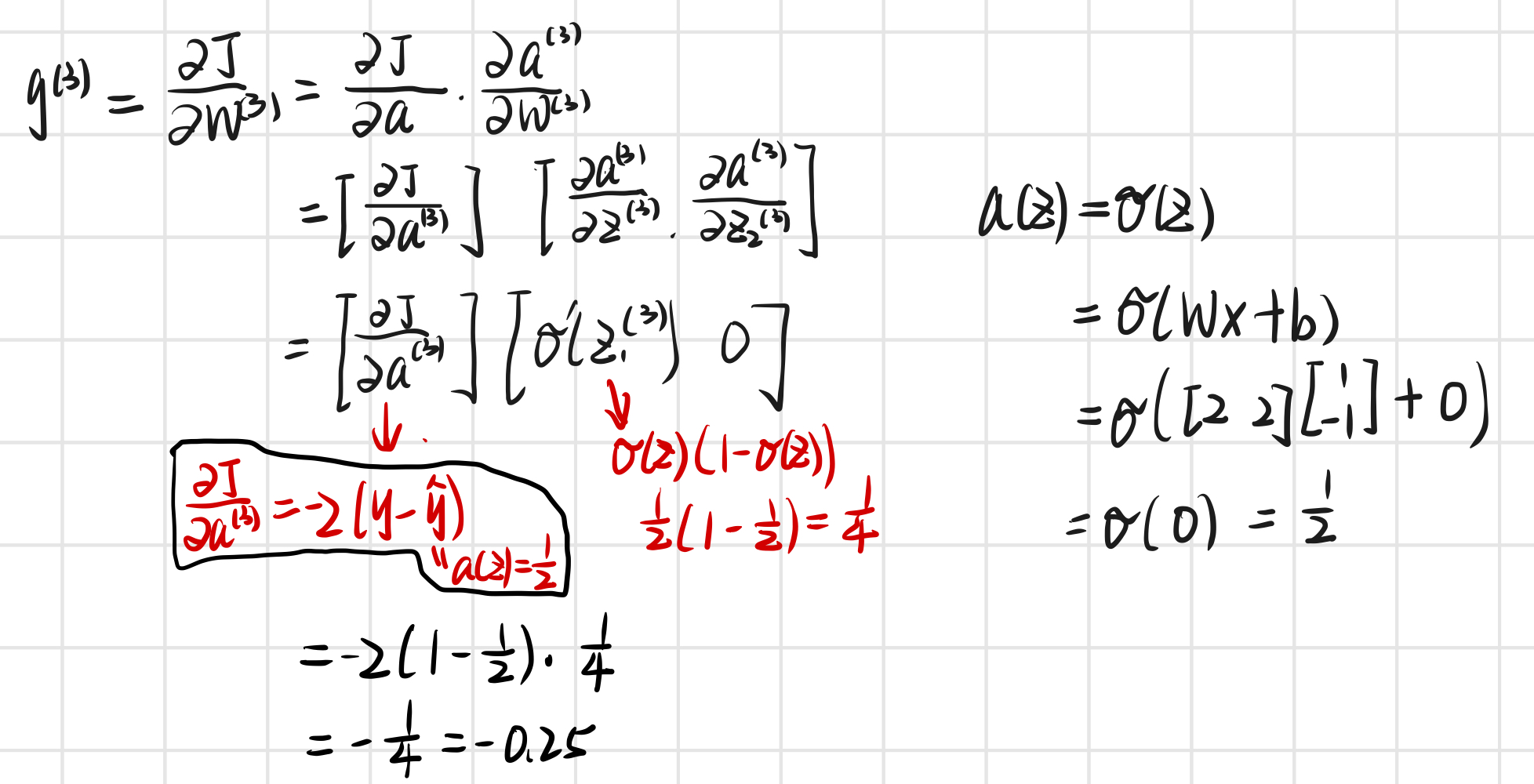
Hint: use fullscreen mode if you have difficulty manipulating the terms.



1. **Compute** the gradient w.r.t. the pre-activations of the final layer for the network below

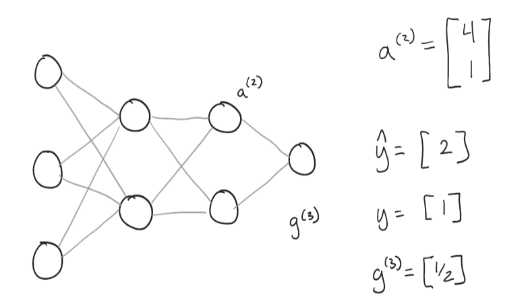
the gradient forif the activation function is a logistic sigmoid and the loss is the square loss

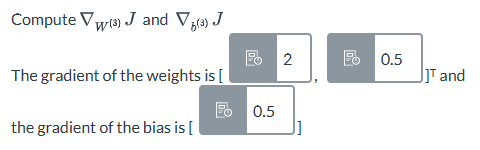
**The gradient for layer 3 is: -0.25**

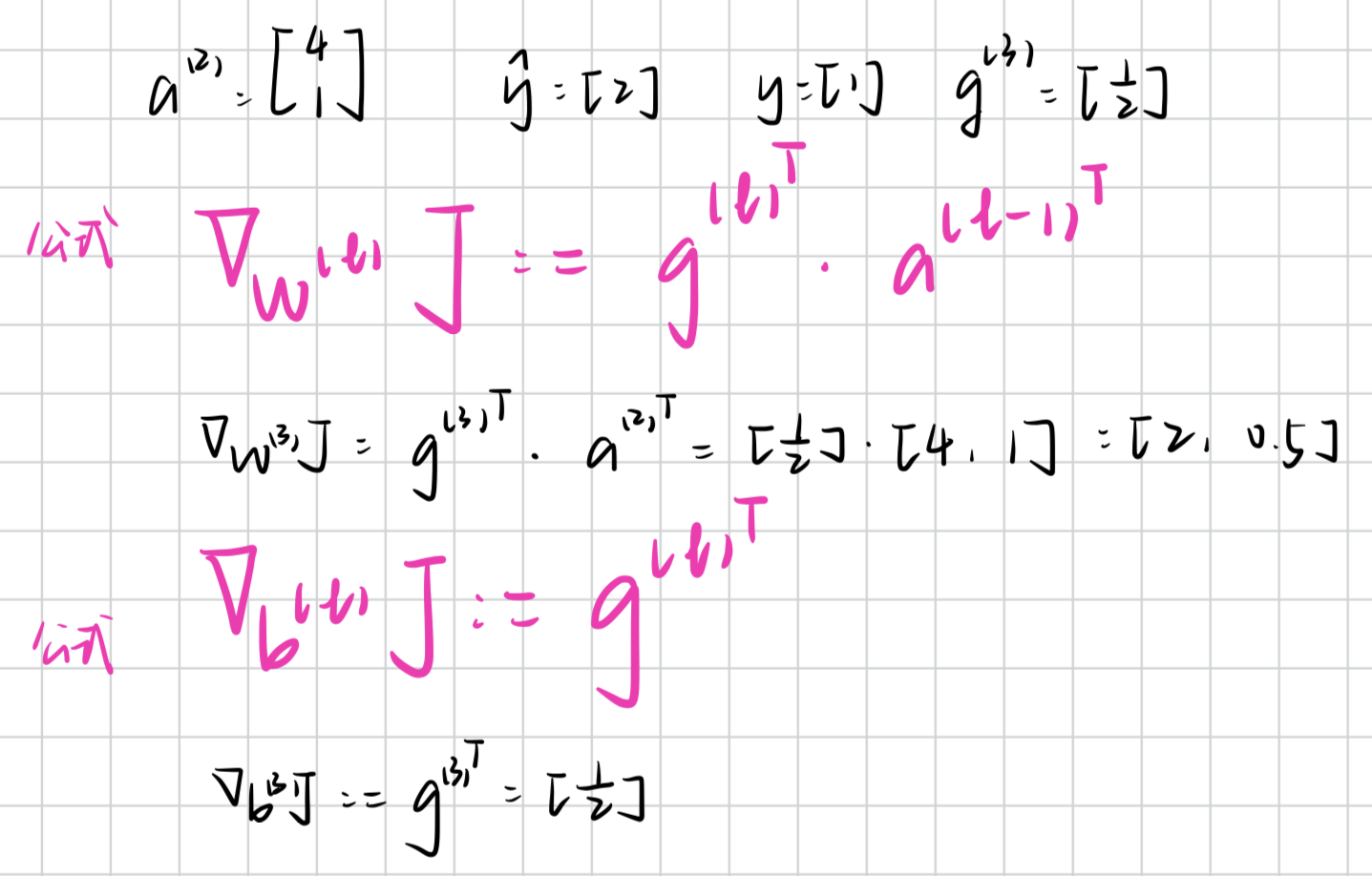
****

1. Compute the gradient w.r.t. the weights and biases of layer ℓ=3

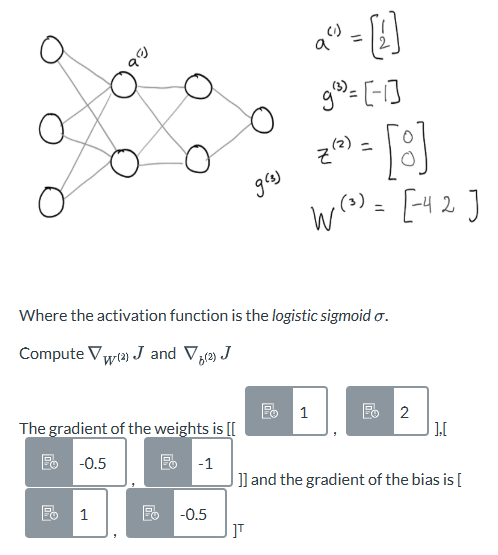
for the following network

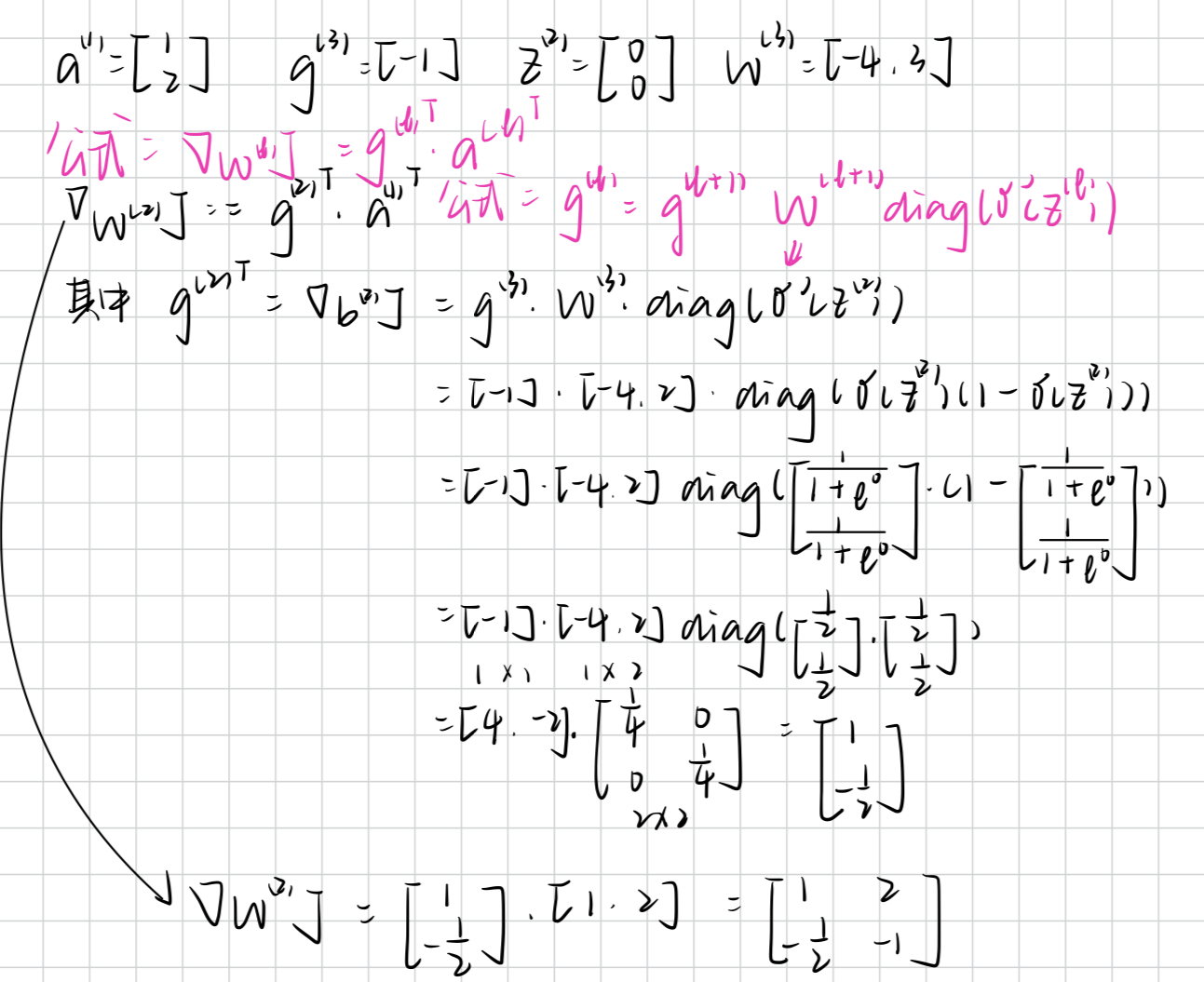
****

****

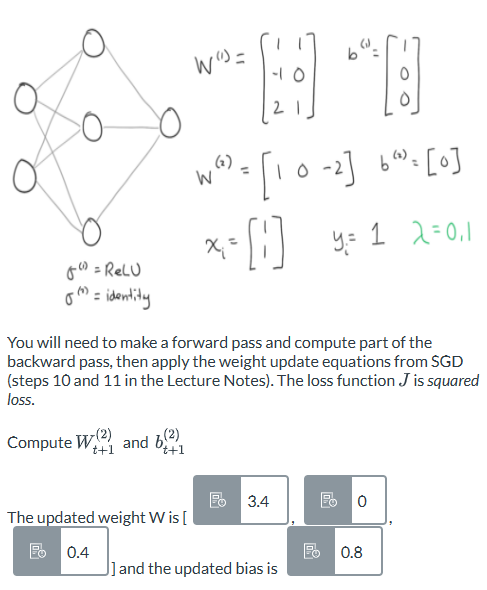
****

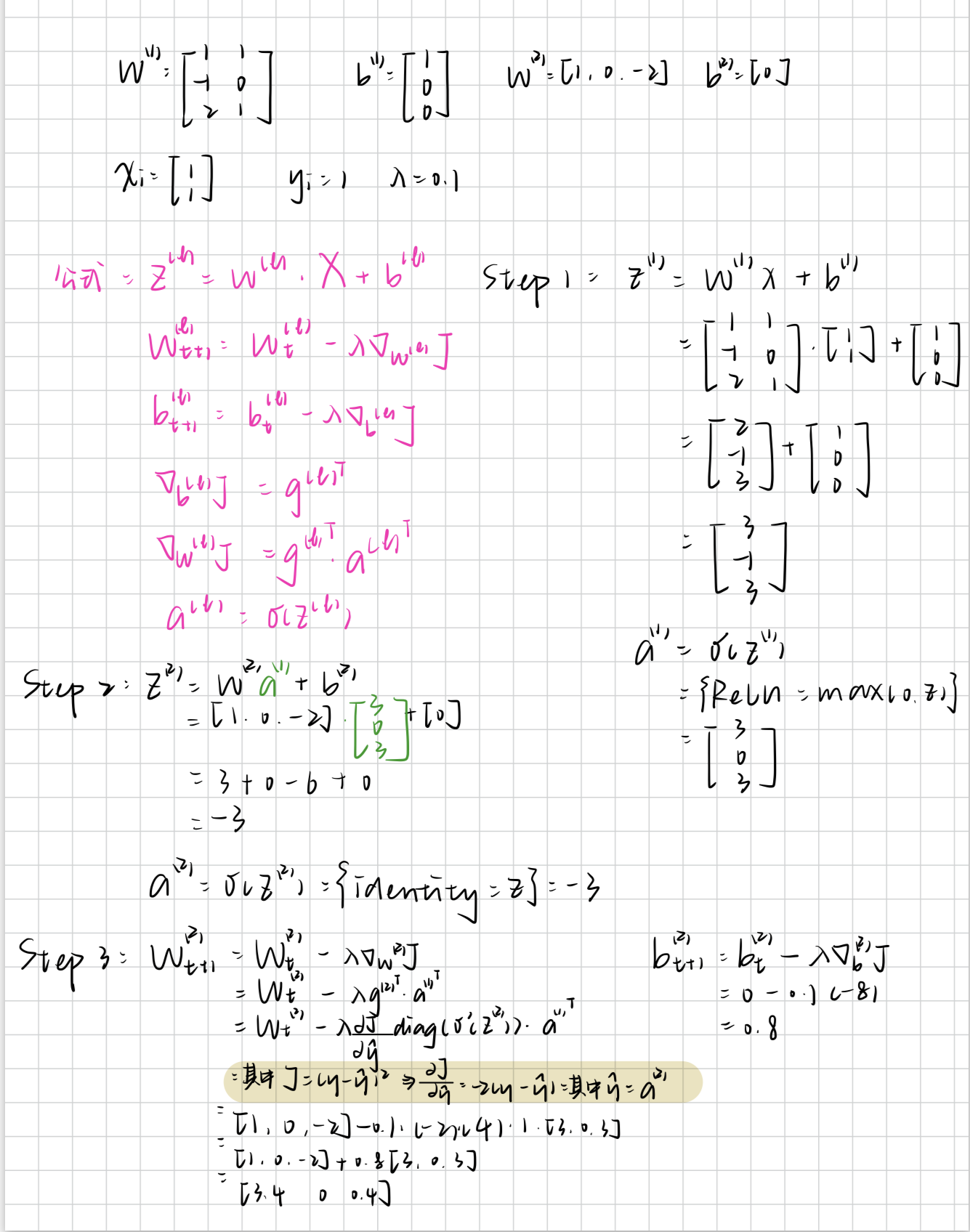
1. **Compute the gradient w.r.t. the weights and biases for layer ℓ=2 given the following information:**

****

****

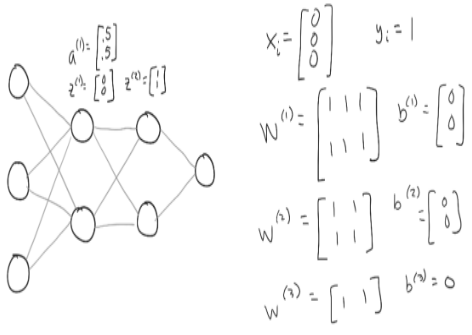
1. **Compute the updated weights and biases for layer ℓ=2 given the following information:**

****

****

**4.4 & 4.5**

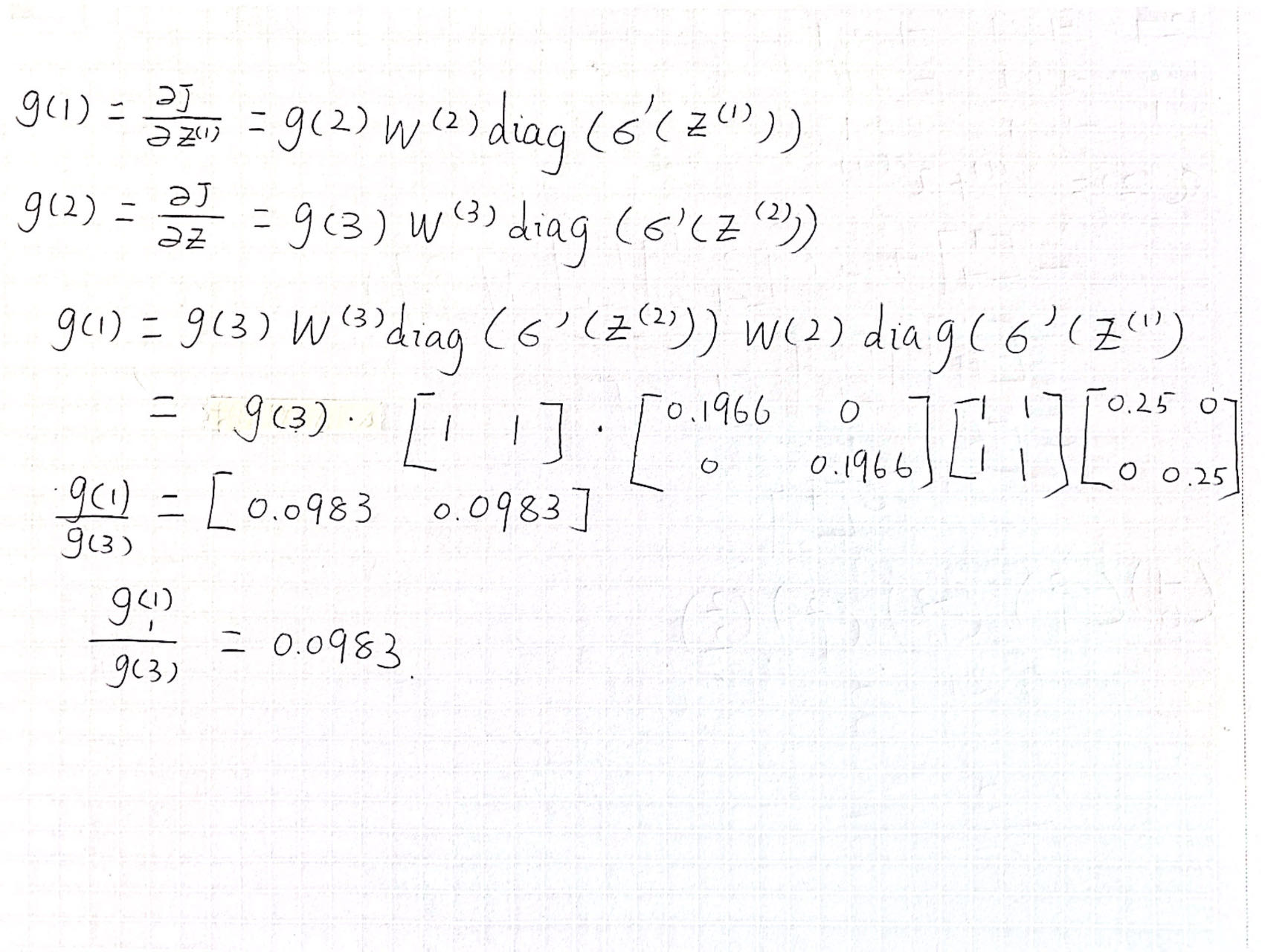
1. Consider the following network with sigmoid activations and *very* uniform weights.



Check how much weaker the gradient becomes at layer ℓ=1 by computing the ratio . We provide the gradient w.r.t. the pre-activations .

**Compute** the ratio of the gradients .



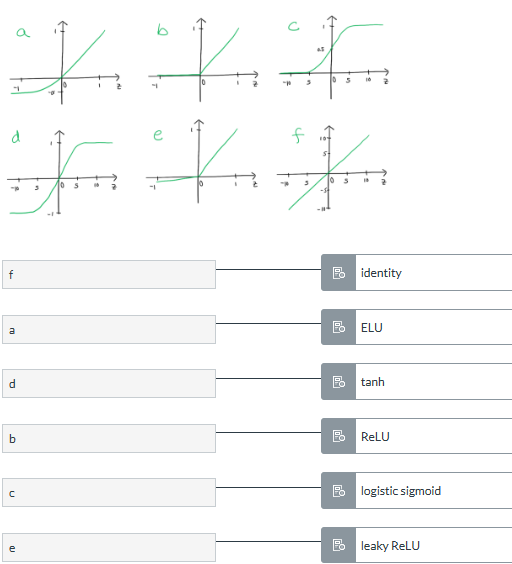


1. What causes the *vanishing gradient problem?*

**Mark** all that apply.

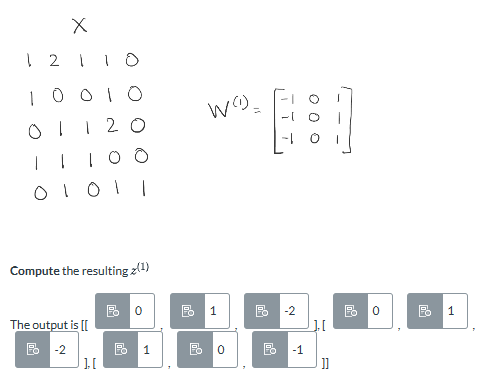
* The derivative of certain activation functions like logistic sigmoid and *tanh* have small values for strong activations.
* The compounding effect of the chain rule over many layers.
* Some activation functions are not differentiable at 0.
* The ReLU has a derivative equal to 1.

1. **Match** the following activations functions correctly.



1. Given the following input and the convolution weights

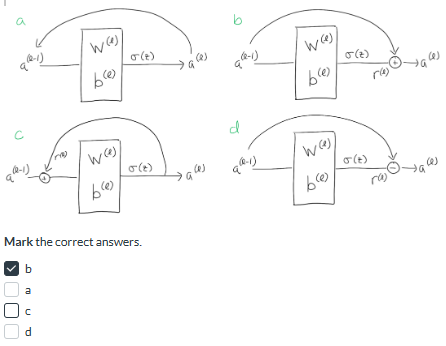
below, compute the resulting pre-activation (you may ignore the bias term).



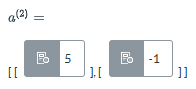
1. **Mark** the following statements that are correct.

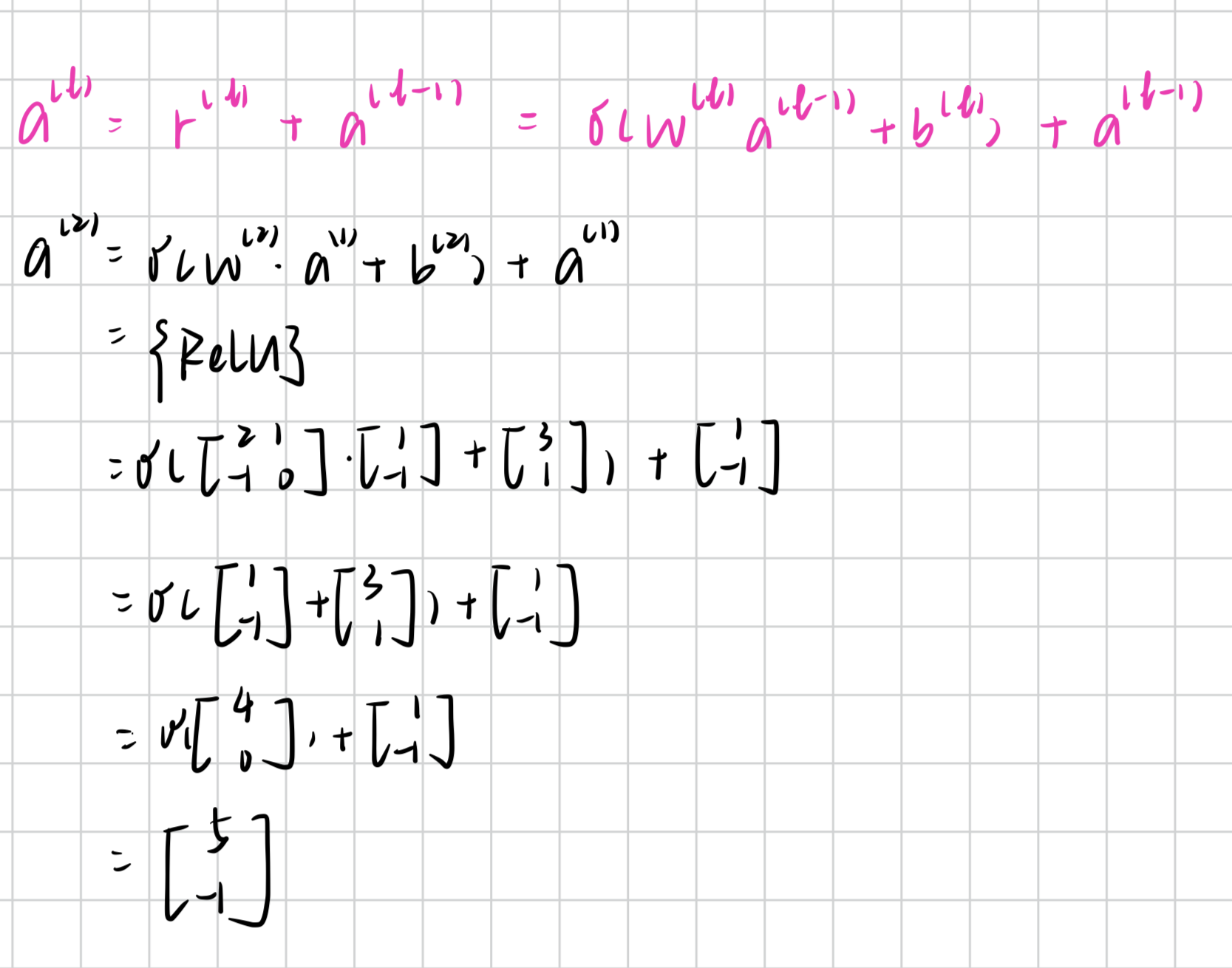
* A convolutional layer can combine information from distant regions of the input.
* A network consisting of convolutional layers is known as an *image processing network.*
* Convolution layers are more computationally efficient at processing large inputs than fully connected layers.
* Convolutions are well suited to image data because they act locally and are translationally equivariant.

1. The **degradation problem** was observed when it was discovered that **increasing** network depth, after a certain point, **degrades** performance. It was initially thought that this was due to **overfitting**, but experiments disproved this. Not only did the **test error** get worse, but the **training error** did as well. A thought experiment demonstrated that, in principle, this should not be the case. If one were to train a **shallow network** achieving good performance, and then append **identity layers** to that network, it would achieve the same performance. But in practice, optimization algorithms could not find such good solutions. This led to the insight that **identity mappings** may be difficult for deep neural networks to learn, and led to the invention of residual layers
2. Which of the following correctly depicts a *residual connection?*

******

1. Consider the output of layer 1 given by  and the parameters of layer 2 given by and the activation function is a . **Compute** the output of layer ℓ=2 that has a *residual connection*.





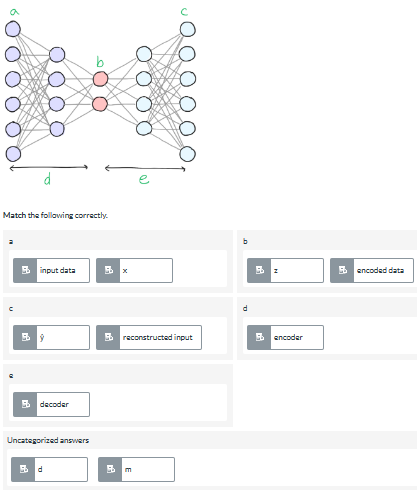
1. **Mark** all of the correct statements below regarding *pretraining*.

* Using a pretrained network (trained on a large, relevant dataset) often leads to better performance than training from scratch on a small dataset.
* Using a *pretrained* network is a kind of *transfer learning* where knowledge in the form of learned weights is transferred to a new problem/domain.
* A small network trained on a small dataset is a good candidate for using as pretrained weights.
* Initializing with a pretrained network often speeds up training

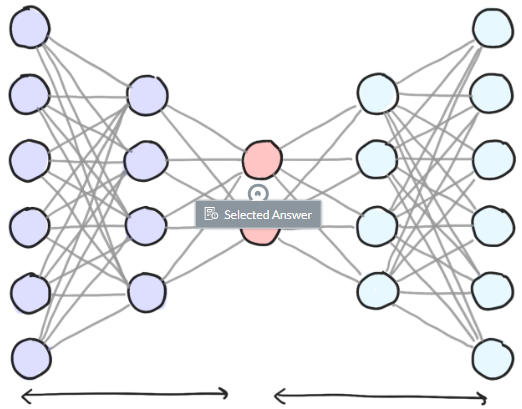
1. **Mark** all of the following statements about *data augmentation* below that are correct.

* Data augmentation is a form of regularization
* Inappropriate data augmentations can worsen model performance
* Data augmentation is used to increase the size of the training data.
* Data augmentations can substantially increase training time
* Data augmentation can help generalization performance

1. dfgdfgd



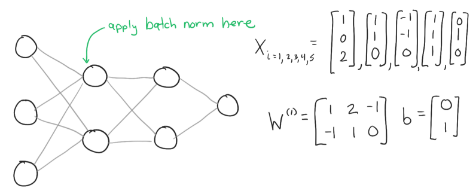
1. **Click** on the location in the diagram below corresponding to the autoencoder's *latent space*.

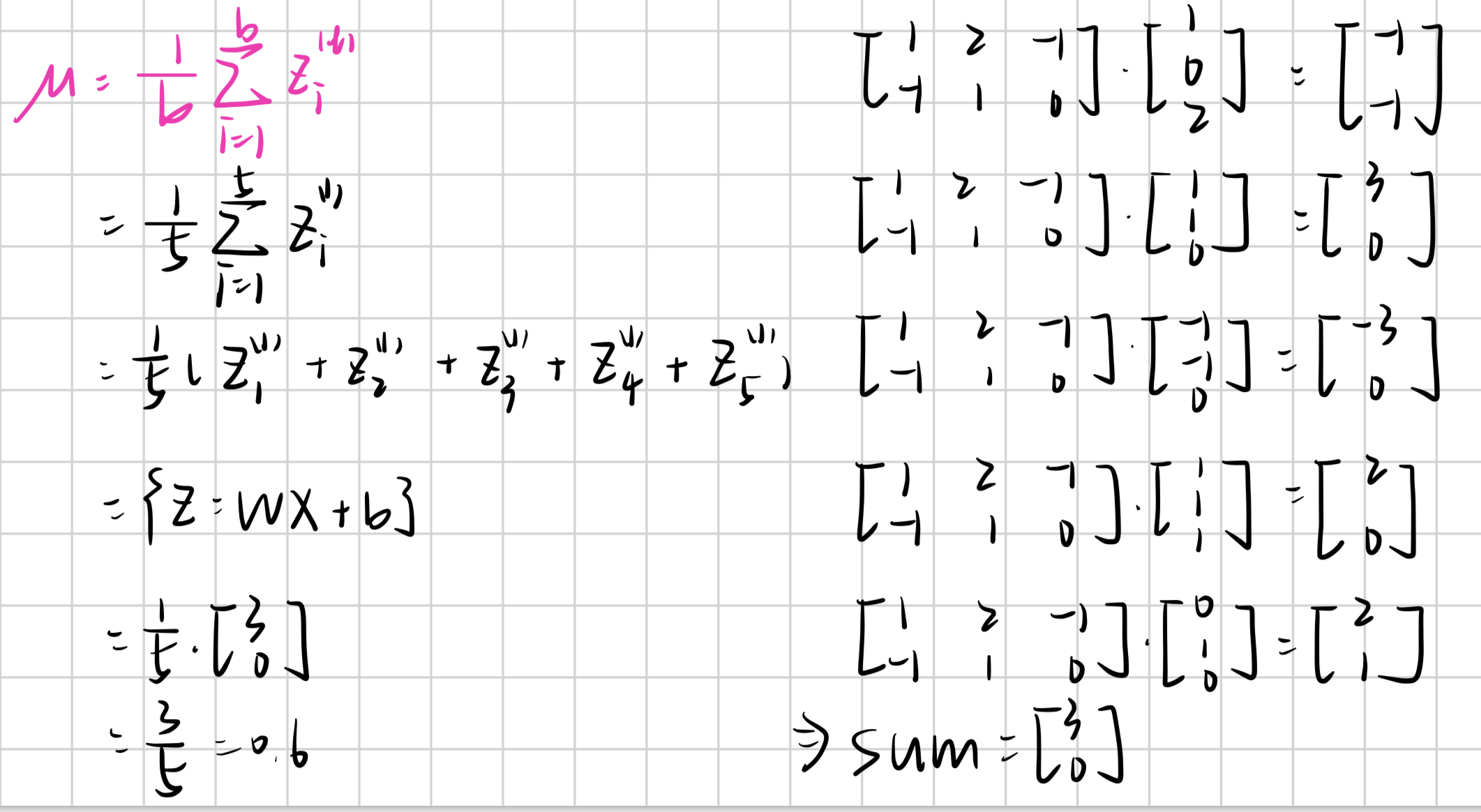


1. **Fill in** the statement below correctly.

An **autoencoder** is a neural network architecture that uses an **encoder** to compress input data into a **latent space**. The **decoder** tries to **reconstruct** the original input from the latent vector z. To accomplish this, the neural network must learn to preserve the most important parts of the information into the latent code . This makes the autoencoder useful for **unsupervised learning** since it can extract features from unlabeled data.

1. Consider the following neural network.

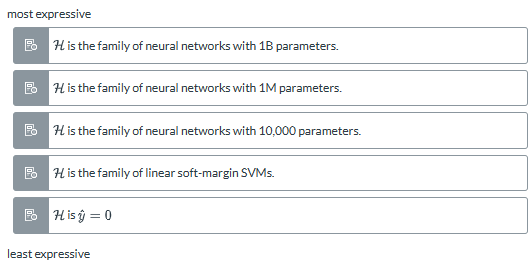
**Compute** the mean pre-activations of layer ℓ=1 that would be necessary to apply *batch normalization* to the layer given the batch of b=5 inputs above.**The mean of is 0.6**

****

1. False - The purpose of batch normalization is to introduce *covariate shift* to the network so it generalizes better.
2. **Mark** all the statements about *dropout* below that are correct.

* Dropout removes nodes from the network randomly during the backwards pass of each training step.
* The *dropout rate* is a hyperparameter introduced when using dropout which controls the probability a node will be 'dropped'.
* Dropout helps reduce overfitting and improves generalization.
* One explanation for why dropout works is that it forces each node to be more co-adapted with the others
* The connections (weights) of randomly dropped neurons are also dropped.

1. **Sort** the following in terms of model complexity (or expressivity) on a dataset with 50,000 training examples (assuming similar architectures).

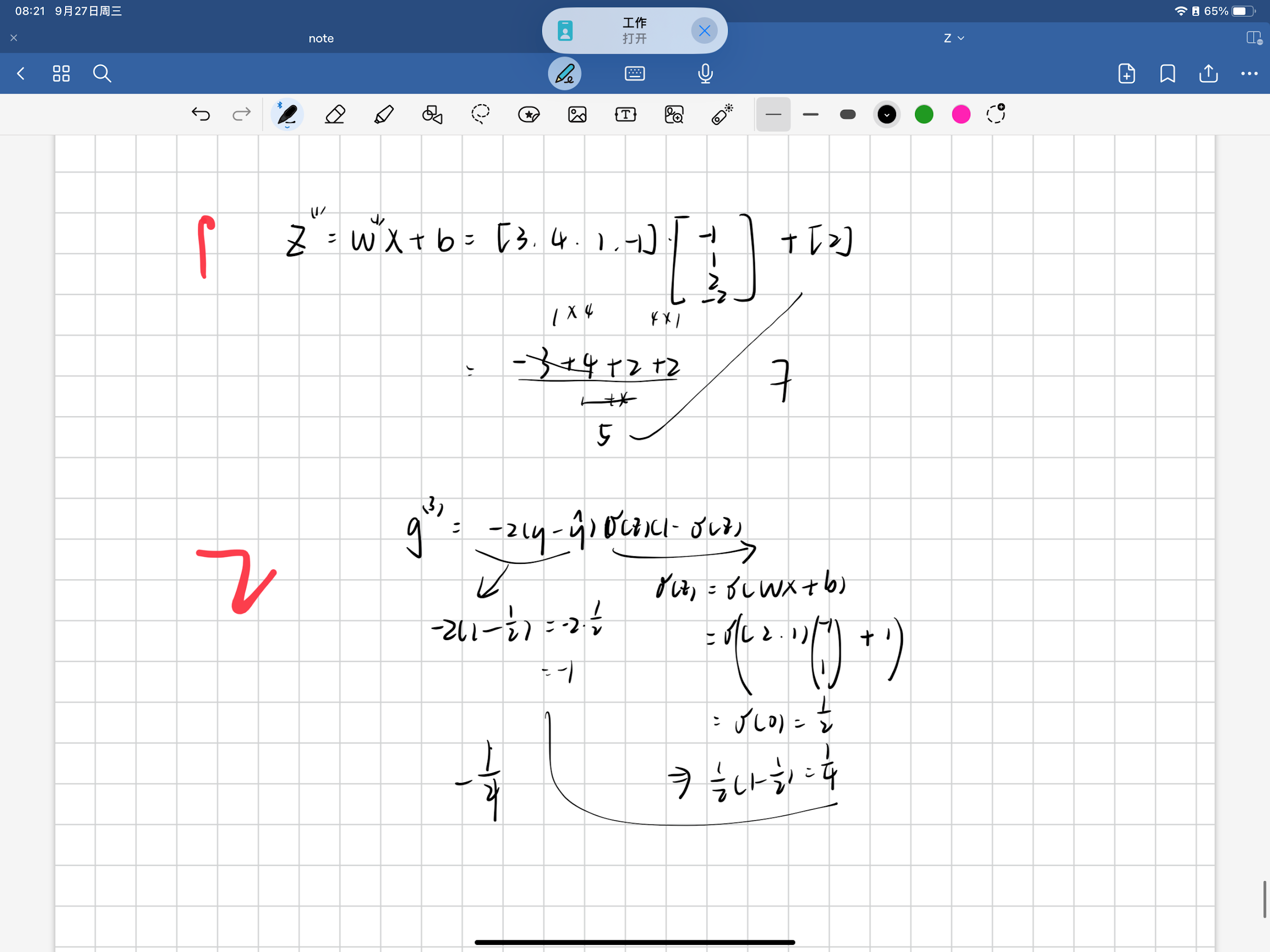


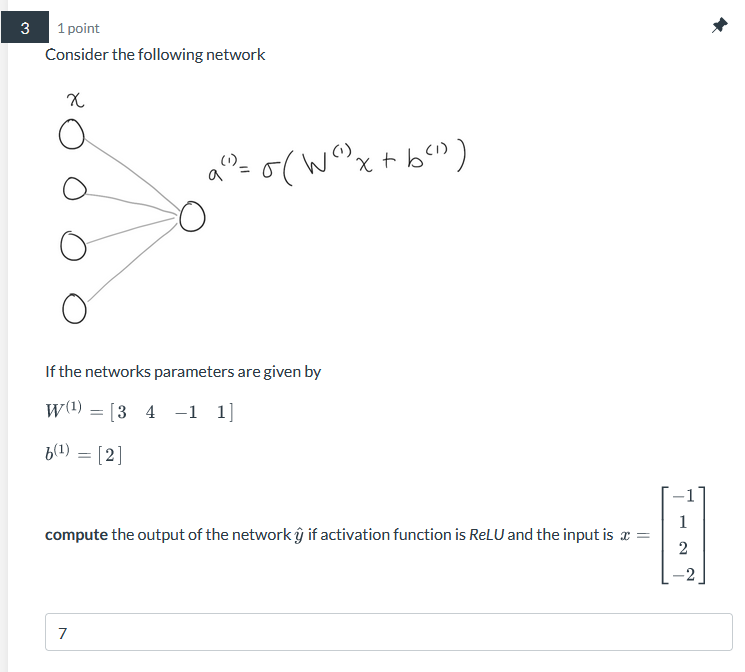
1. **Fill in** the statement below correctly.

**Neural networks** tend to see more performance benefit as more **training data** is added, unlike **traditional learning methods** which tend to plateau earlier. **Larger** neural networks tend to benefit more from big data than **smaller** networks.

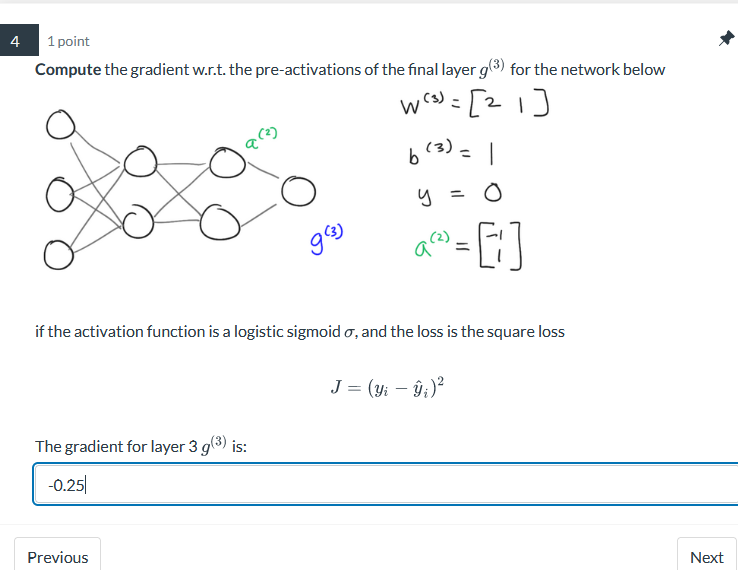
1. True - The *bitter lesson*tells us that our attempts to leverage human knowledge into artificial intelligence are doomed to be outperformed by general purpose learning methods (such as neural networks) that scale with compute and data.

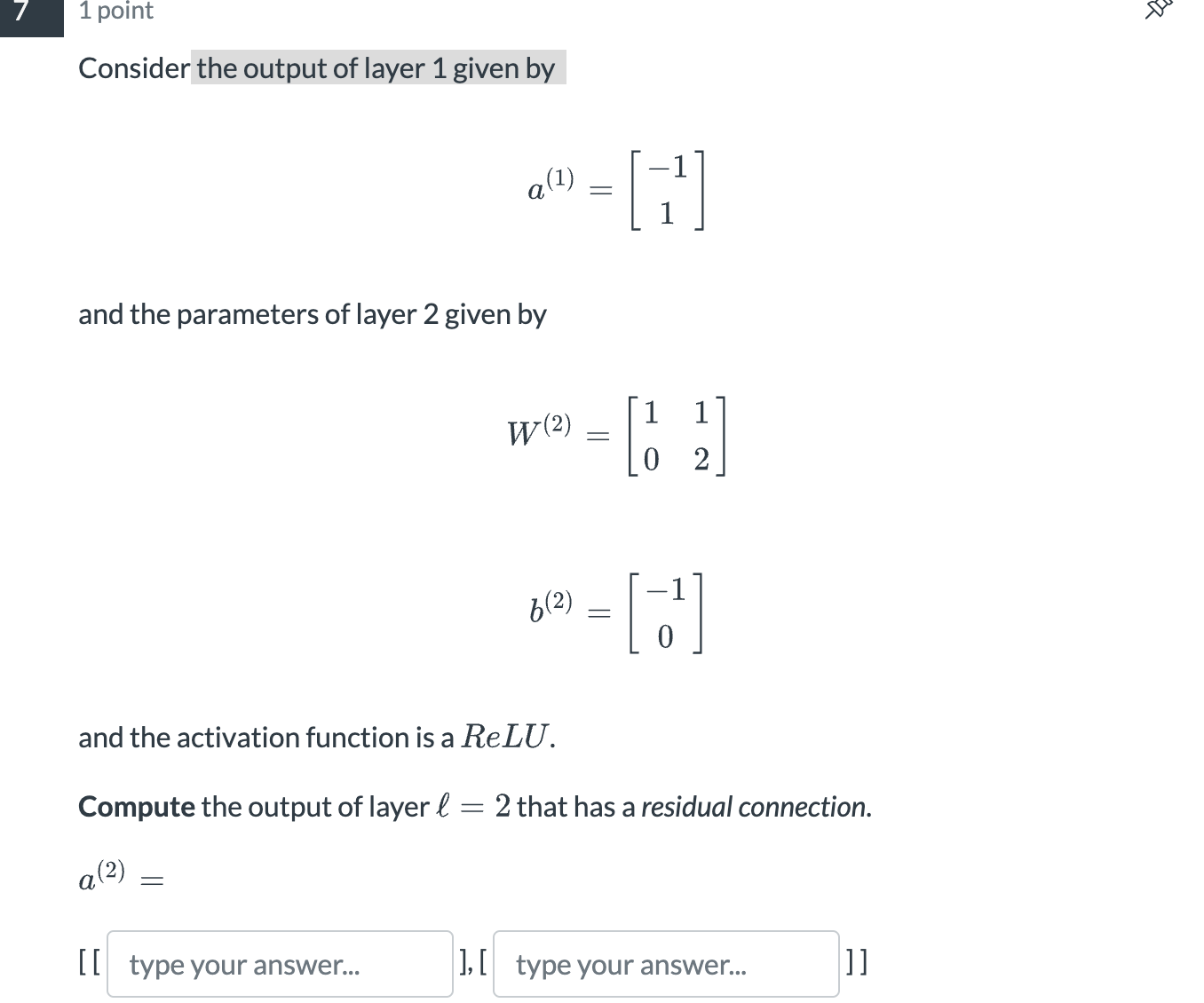
—----



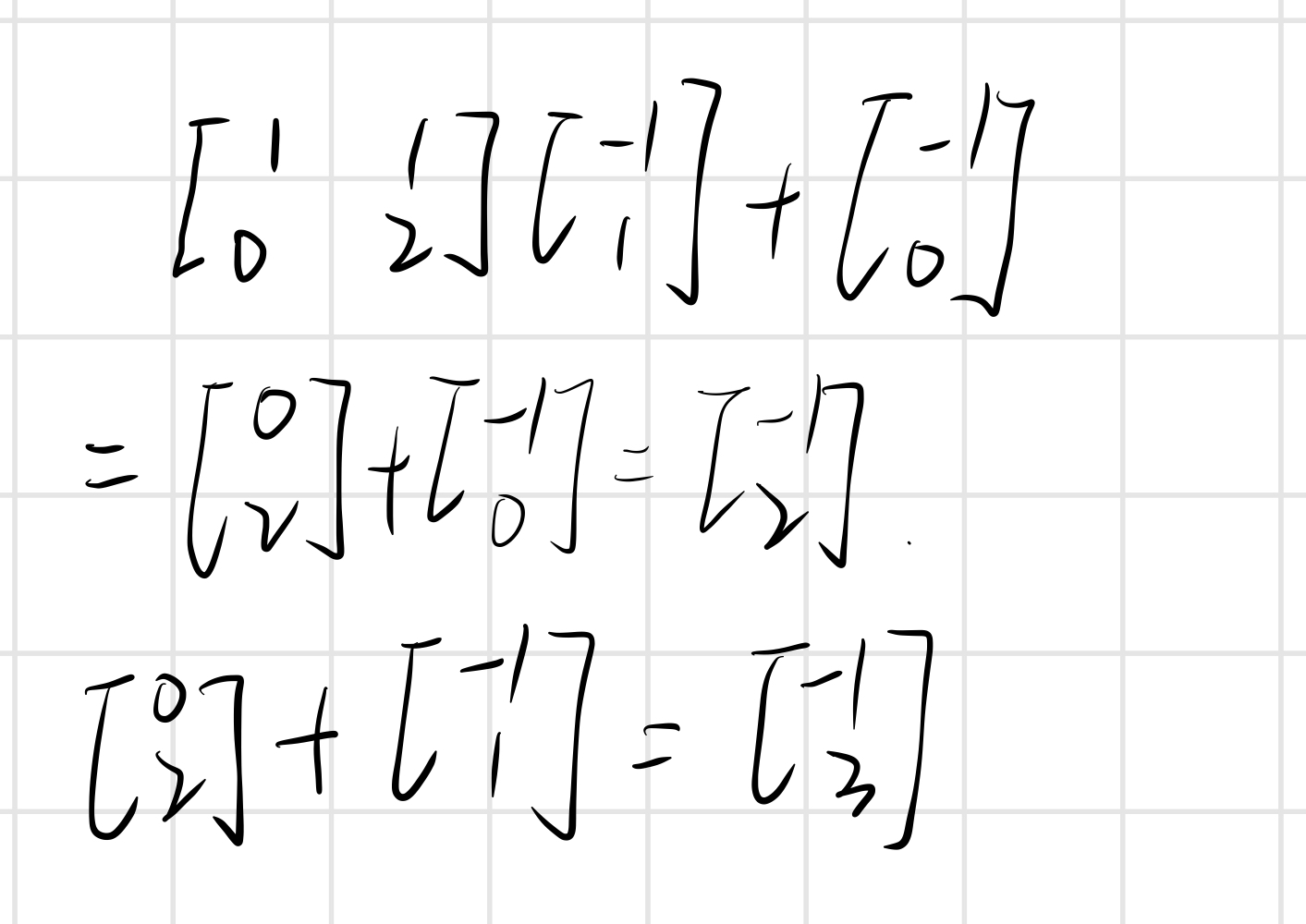


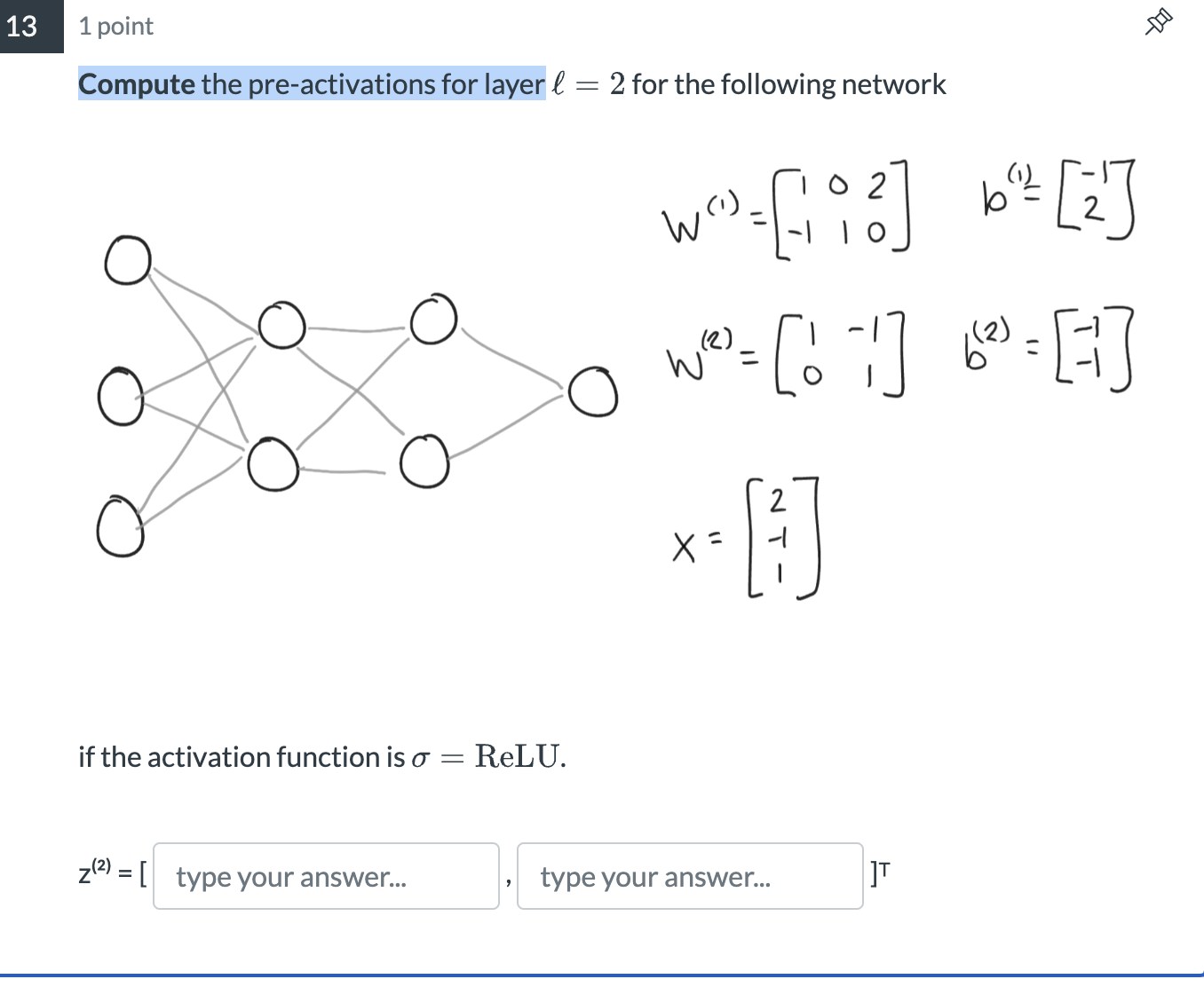
0—





[-1 3]





[2 -1]

