AI Course

Chapter 8. Quiz

For instructors (with answers)

ⓒ2022 SAMSUNG. All rights reserved.

Samsung Electronics Corporate Citizenship Office holds the copyright of this document.

This document is a literary property protected by copyright law so reprint and reproduction without permission are prohibited.

To use this document other than the curriculum of Samsung Innovation Campus, you must receive written consent from copyright holder.

1. What is the problem that occurs when the number of nodes is small when analyzing artificial neural networks?
2. If the number of nodes is small, the number of operations to be processed is small, so it is performed quickly.
3. Complex decision-making boundaries cannot be created when performing an analysis model.
4. Backpropagation algorithm that adjusts weights and thresholds cannot be performed.
5. The number of nodes has nothing to do with the analysis model.

Answer. 2

1. What is the concept described below?

|  |
| --- |
| ***Multidimensional matrix with 3 components: Rank, Shape, Type*** |

Answer. Tensors

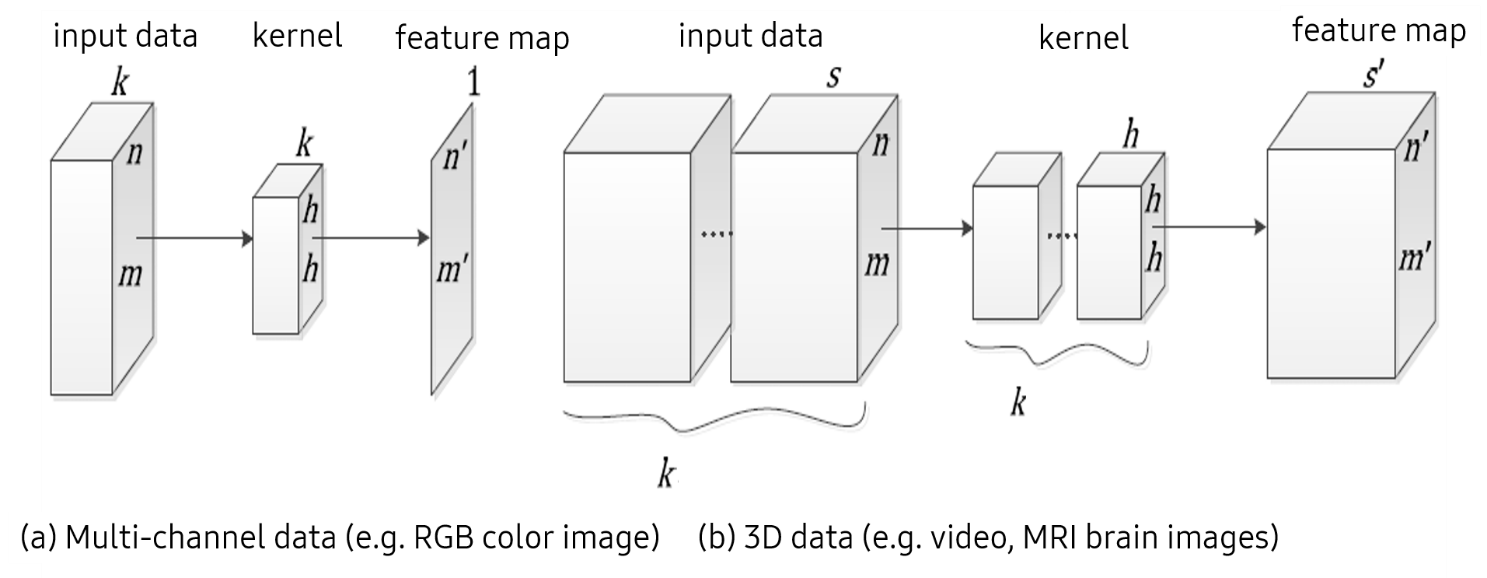
1. What happens if the verification error is constantly rising when we graph the verification error for each epoch using the batch slope descent method? Also, how can you solve this problem?

Answer.

If verification errors continue to rise for each epoch, one possibility may be that the learning rate is too high and algorithm divergence might have been occurring.

If training errors increase, this problem should be clear and the learning rate should be lowered. However, if the training error does not go up, the model is overfitting the training set and the training should be stopped.

1. The figure below is data with a three-dimensional structure.
2. Present the convolution equations for (a).
3. Present the convolution equations for (b).



Answer 1)

Answer 2)

1. Apply the softmax function when the output of the neural network is and write the result.

Answer.

|  |
| --- |
| import numpy as np  def softmax(x):  e\_x = np.exp(x - np.max(x))  return e\_x / e\_x.sum()  x = np.array([0.4,2.0,0.001,0.32])  y = softmax(x)  print(y) |

[0.13250053 0.65627943 0.08890663 0.12231341]