AI & Deep Learning 2024 (Solution)

1. (20%) (Optimization)
2. (10%) Explain the SGD (Stochastic Gradient **Descent**) updating policy. You need to explain SGD in terms of the gradient of loss function 𝛻(𝐽(𝑤)), parameters (**w**), and the learning rateε. (10%). (Note that a ‘+’ or a ‘-‘ sign in the equation has a different meaning. Please don’t mix them up!)

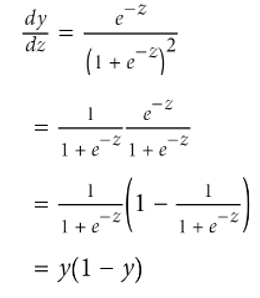
The new weight (vector) equals the original weight minus Epsilon multiplied by the gradient of the loss function with weights as parameters.

1. Since simple SGD may be stuck in a local optimum or a saddle point, list at least two optimization algorithms that prevent the problems.

Adam, Adagrad

1. (20%)(Activation functions)

(a) (10%) Define the sigmoid activation function and derive its derivative. (導出它的微分公式)



(b) (10%) The softmax activation function can be written as:

(1) Where do we use a softmax function in a neural network?

It is used to transform the output of a neural net layer into a probability distribution.

(2) Derive the derivative of , and show that it equals

\*(1-)

Use the formula of (f\*g)’=f’\*g+f\*g’)

=

1. (20%) The following is a Keras code segment for a convolutional Neural network. The inputs are two-dimensional 32x32 CIFAR-10 color pictures. Answer the following questions:
2. (10%) What is the total number of parameters in **the first Conv2D layer**?

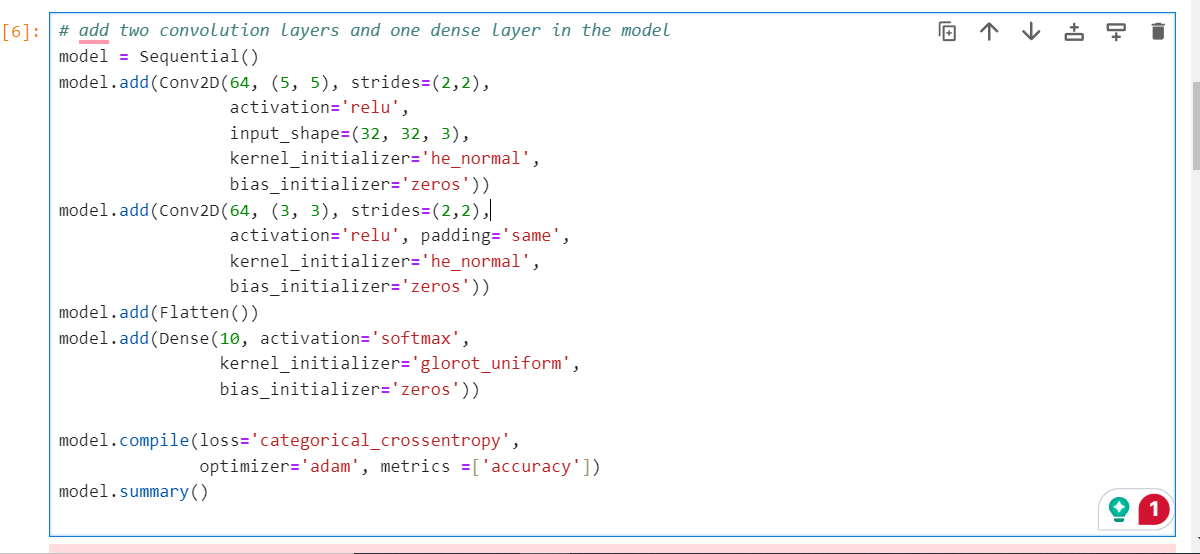
(5\*5\*3)\*64 + 64 =

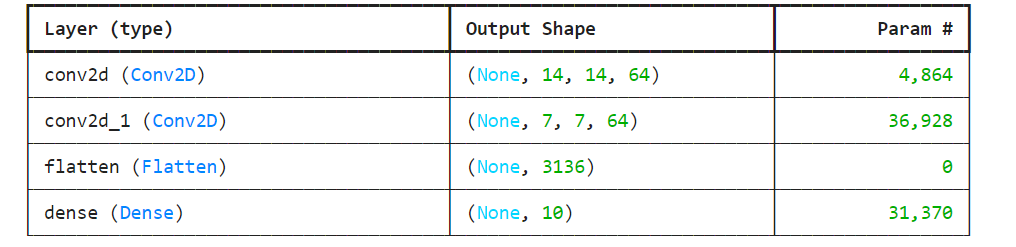
4864

1. (10%) What is **the output shape of the first Conv2D layer**?

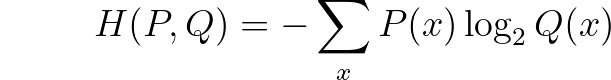
dim=(32-5)/2 +1 =14, with 64 filters=>

(none, 14, 14, 64)





1. (Cross-Entropy) (20%) Calculate the cross-entropy of distribution P(X) on distribution Q(x) using the following equation:



P(x)

Q(x)

Python code:

import numpy as np

P=[0.4, 0.2, 0.2, 0.2]

Q=[0.2, 0.2, 0.4, 0.2]

cross=0

for i in range(len(P)):

cross+=-(P[i]\*np.log2(Q[i]))

print("Cross-entropy of P on Q is ", cross)

Ans. The cross-entropy of P on Q is 2.1219280948873624

1. (20%) (Gradient Descent) Suppose we want to fit an equation of two parameters for the following dataset:

x\_data = [1.0, 2.0, 3.0, 4.0]

y\_data = [3.0, 10.0, 21.0, 36.0] ,

where x\_data contains the input values of x, and y\_data contains the corresponding outputs.

The equation is

are parameters to be estimated. Furthermore, are initialized to 1 and 2, respectively. . Answer the following questions with the loss function:

(a) Find the gradient equations for w1 and w2, respectively.

Loss= =(2

2 \* x \* ((x\*\*2 )\* w2 +x\*w1- y)

2\*(x\*\*2)\*((x\*\*2 )\* w2 +x\*w1- y)

(b) Find the gradients of w1 and w2, respectively with the initial weights w2= 1 and w1 =2 for the first input x=1.0 and the output y=3.0.

(0, 0)

I will show the Python code in class to demonstrate the gradient exploding problem in SGD.