Task:

Week 5 Exercise - Neural Networks by hand

In week 5 we created a simple toy neuron by hand

This should have left you with enough information to create a single later of neurons

If you managed to do this, you may submit this as one of your in-class assignments.

The matplotlib.pyplot.plot(\*args, \*\*kwargs) method strictly speaking can draw line graphs or sample markers. Among them, \*args allows to enter a single yy value or x, yx, y value.

import numpy as np # Load numerical calculation module

Generate 1000 values equally spaced between -2PI and 2PI, which is the X coordinate

Calculate the y coordinate

Enter X, y coordinates into the method `\*args`

图表

描述已自动生成

np.linspace() Generate 1000 uniform number types in the specified interval -2\*np.pi to 2\*np.pi is numpy. Ndarray's data is the x coordinatenp.sin(x) The sin value is calculated based on the value of X. Is the y coordinate

plt.plot(x,y) is to pass X and Y as a list of X-axis coordinates and a list of Y-axis coordinates, and then display the image.

Generate grid matrix

#Generate 500 uniform numbers in the specified interval -5 to 5 The type is numpy. Ndarray data

x = np.linspace(-5, 5, 500)

#Generate 500 uniform numbers in the specified interval -5 to 5 The type is numpy. Ndarray data

y = np.linspace(-5, 5, 500)

#Generate grid point coordinate matrix X, Y according to x, y

X, Y = np.meshgrid(x, y)

Contour calculation formula

Z = (1-X / 2 + X \*\* 3 + Y \*\* 4) \* np.exp(-X \*\* 2-Y \*\* 2)

Display the image according to the coordinate matrix X, Y and the contour value Z

plt.contourf(X, Y, Z)图表

描述已自动生成

# Generate 1000 values equally spaced between -2PI and 2PI, which is the X coordinate

X = np.linspace(-2 \* np.pi, 2 \* np.pi, 1000)

# Calculate the ordinate corresponding to sin()

y1 = np.sin(X)

# Calculate the ordinate corresponding to cos()

y2 = np.cos(X)

# Enter the X and y coordinates into the method `\*args` to display the picture

plt.plot(X, y1, color='r', linestyle='--', linewidth=2, alpha=0.8)

plt.plot(X, y2, color='b', linestyle='-', linewidth=2)图形用户界面, 图表

描述已自动生成

import numpy as np

from mpl\_toolkits.mplot3d import Axes3D

import matplotlib.pyplot as plt

%matplotlib inline

# x, y, z are 100 random numbers between 0 and 1 random.normal is a random number that generates a normal distribution

x = np.random.normal(0, 1, 100)

y = np.random.normal(0, 1, 100)

z = np.random.normal(0, 1, 100)

#Initial words a drawing tool

fig = plt.figure()

Initial words receipt 3D graphics axes3D

ax = Axes3D(fig)

Pass the random normal distribution xyz into the scatter method to draw 3D graphics

ax.scatter(x, y, z)图表, 散点图

描述已自动生成

# Generate data from -6\*np.pi to a thousand evenly spaced numbers before 6\*np.pi to generate x coordinates

x = np.linspace(-6 \* np.pi, 6 \* np.pi, 1000)

#Calculate the sin value of x Generate the y coordinate

y = np.sin(x)

#Calculate the con value of x Generate the z coordinate

z = np.cos(x)

# Create 3D graphics objects

fig = plt.figure()

#Initial wordsAxes3D

ax = Axes3D(fig)

#Display image

ax.plot(x, y, z)图示

描述已自动生成

# Create 3D graphics objects

fig = plt.figure()

ax = Axes3D(fig)

# Generate data and plot

x = [0, 1, 2, 3, 4, 5, 6]

for i in x:

y = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

# The absolute value of the normal distribution between 1 and 10

z = abs(np.random.normal(1, 10, 10))

#barDraw 3D histogram

ax.bar(y, z, i, zdir='y', color=['r','g','b','y'])手机屏幕截图

中度可信度描述已自动生成

# Create 3D graphics objects

fig = plt.figure()

ax = Axes3D(fig)

# Generate data

#From -2 to 2 interval 0.1 to generate data X axis

X = np.arange(-2, 2, 0.1)

#From -2 to 2 interval 0.1 to generate data Y axis

Y = np.arange(-2, 2, 0.1)

#Generate grid point coordinate matrix X, Y according to x, y

X, Y = np.meshgrid(X, Y)

#Find the square root of X plus the square root of Y

Z = np.sqrt(X \*\* 2 + Y \*\* 2)

# Draw a surface map and use cmap to color it

ax.plot\_surface(X, Y, Z, cmap=plt.cm.winter)图表, 表面图

描述已自动生成