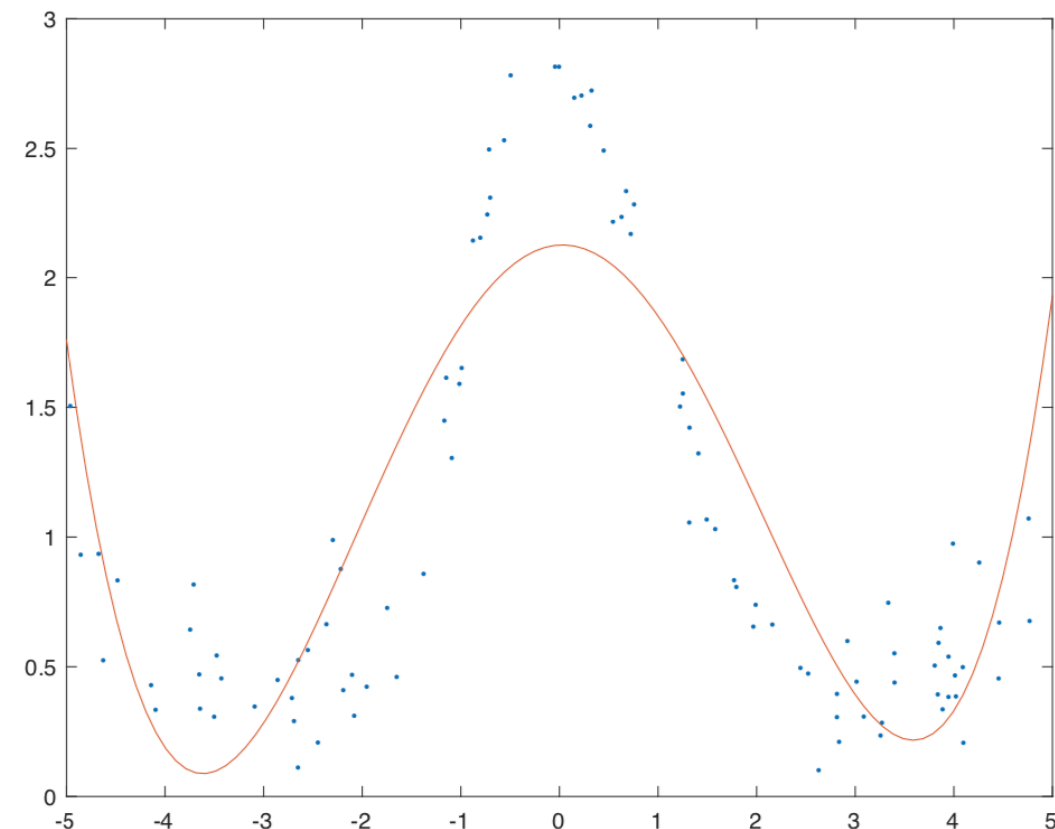


Curve fitting
2-dimensional function
fitting

Approximate $\exp(\cos(x))$ using neural functions

Apply polyfit to estimate coefficients of a 4-degree polynomial subject to given data for approximating $\exp(\cos(x))$

```
n = 100;  
x = rand(1, n) * 10 - 5;  
noise = randn(1,n) * 0.2;  
str = ' exp(cos(x)) ';  
f = inline(str);  
y = f(x) + noise;  
plot(x, y, '.'); hold on;  
  
p = polyfit(x, y, 4);  
x_new = linspace(-5,5);  
y_new = polyval(p, x_new);  
plot(x_new, y_new)
```

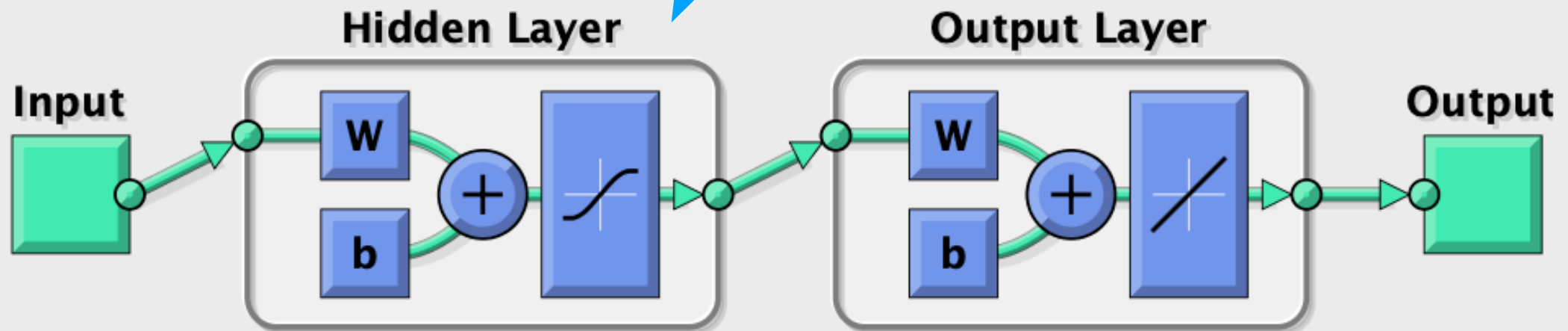


p =

0.0117 -0.0001 -0.3041 0.0189 2.1259

Neural Network

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



A two-layer feed-forward network with sigmoid hidden neurons and linear output neurons (`fitnet`), can fit multi-dimensional mapping problems arbitrarily well, given consistent data and enough neurons in its hidden layer.

The network will be trained with Levenberg-Marquardt backpropagation algorithm (`trainlm`), unless there is not enough memory, in which case scaled conjugate gradient backpropagation (`trainscg`) will be used.

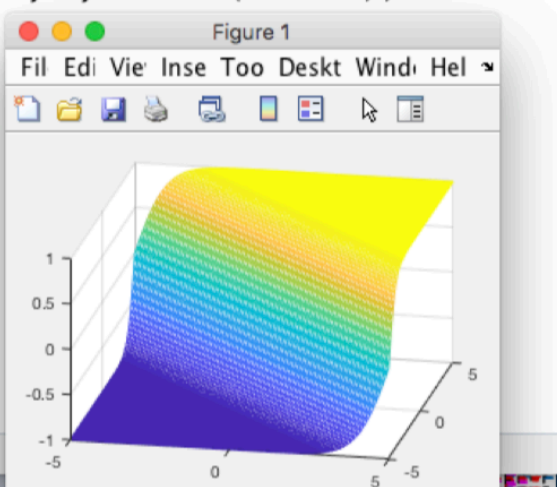
$$x \in R^d \quad h = Wx + b, \quad W : m \times d, \\ h : m \times 1 \quad b : m \times 1,$$



請完成mesh指令，繪製 $f(x, y) = \tanh(x + y)$ 的立體圖

```
n = 100;  
a = linspace(-5,5,n);  
X = _____;  
Y = _____;  
mesh(a,a, _____)  
>>  
>>  
>>  
>>  
>>  
>>  
fx >>
```

本題答題



Problem A. Draw the following function in a figure

$$f(x_1, x_2) = \tanh(x_1 + 0.5 * x_2 - 1) + \tanh(0.5x_1 - 0.8x_2 + 0.5)$$

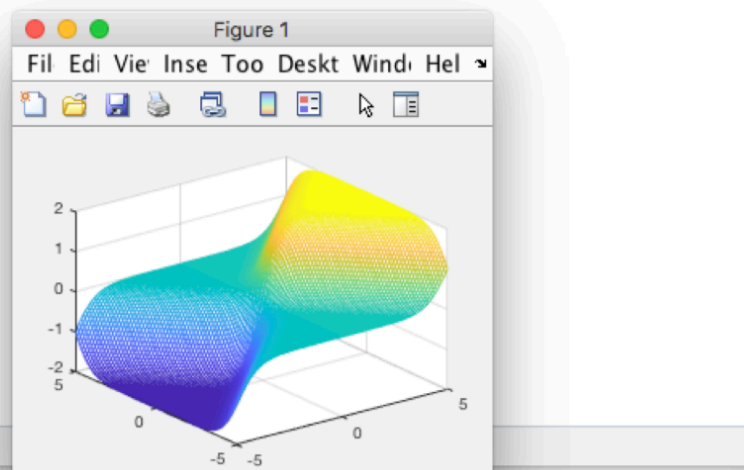
where each point

$$(x_1, x_2) \in [-\pi, \pi] \times [-\pi, \pi]$$

請完成mesh指令，繪製 $f(x, y) = \tanh(x + y) + \tanh(x - y)$ 的立體圖

```
n = 100;  
a = linspace(-5,5,n);  
X = _____;  
Y = _____;  
mesh(a,a, _____)  
>>  
>>  
>>  
>>  
>>  
>>  
fx >>
```

本題答題



Exercise 1: Write a script for tasks in problems A, B and C

Problem A. Draw the following function in a figure

$$f(x_1, x_2) = \tanh(x_1 + 0.5 * x_2 - 1) + \tanh(0.5x_1 - 0.8x_2 + 0.5)$$

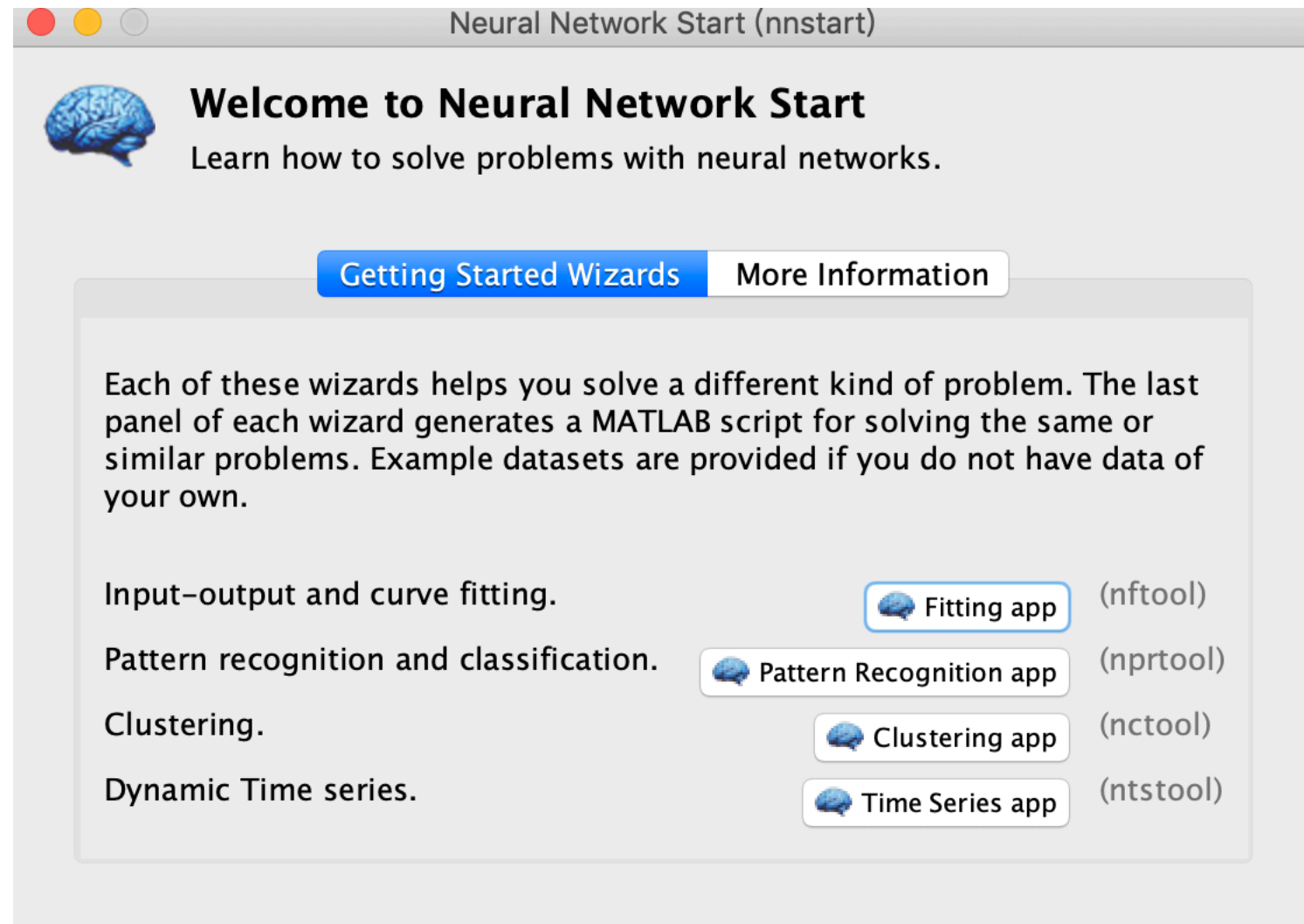
where each point $(x_1, x_2) \in [-\pi, \pi] \times [-\pi, \pi]$

Problem B. Data Sampling: Let $n = 300$. Generate an $n \times 2$ matrix x , where each row denotes the coordinate of a random point in $[-\pi, \pi]^2$, and an $n \times 1$ vector y such that


$$y[i] = f(x[i,1], x[i,2]), \quad i = 1 \dots n$$

Problem C. Draw n points, represented by x and y , in a figure

How to use neural fitting







Neural Network Start (nnstart)

 **Welcome to Neural Network Start**
Learn how to solve problems with neural networks.

Getting Started Wizards | More Information

Each of these wizards helps you solve a different kind of problem. The last panel of each wizard generates a MATLAB script for solving the same or similar problems. Example datasets are provided if you do not have data of your own.

Input-output and curve fitting.	 Fitting app	(nftool)
Pattern recognition and classification.	 Pattern Recognition app	(nprtool)
Clustering.	 Clustering app	(nctool)
Dynamic Time series.	 Time Series app	(ntstool)

Exercise 2: Use the toolbox of neural fitting for mapping x to y

Problem A. Import x and y for neural fitting

Problem B. Train a neural network for function approximation subject to x and y

Problem C. Try to print the final screen of neural fitting

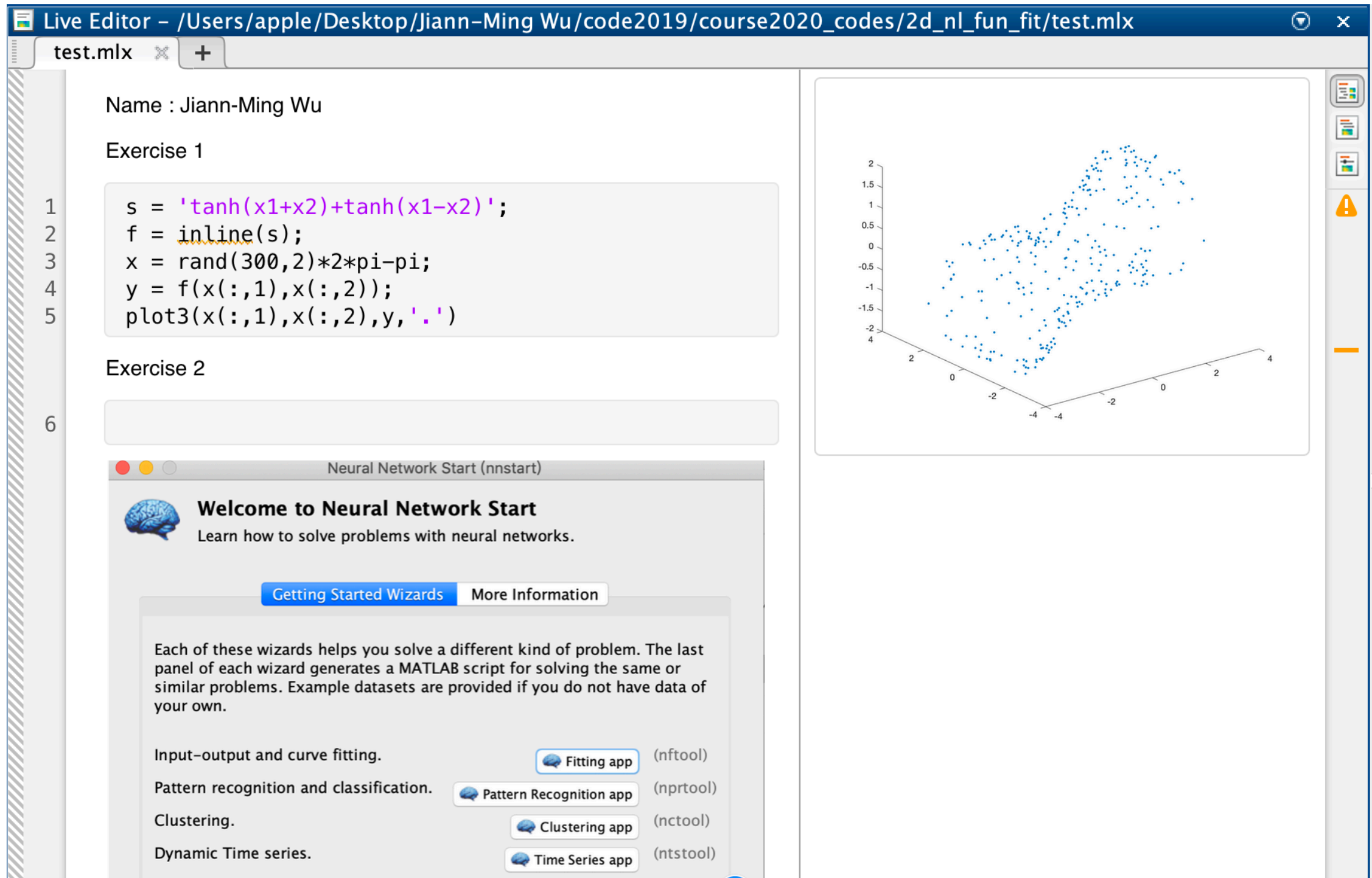
Exercise 3:

Problem A. Use the following code to train a neural network for mapping x to y , and calculate $\text{mean}((y_{\text{hat}}-y').^2)$

```
net = feedforwardnet(20, 'trainlm');  
net = train(net, x', y');  
y_hat = net(x');
```

Problem B. Write a script to draw the neural mapping, which is represented by `net`, in a figure

How to edit an mlx file



The screenshot displays the MATLAB Live Editor interface. The top window title is "Live Editor - /Users/apple/Desktop/Jiann-Ming Wu/code2019/course2020_codes/2d_nl_fun_fit/test.mlx". The editor shows the following code:

```
1 s = 'tanh(x1+x2)+tanh(x1-x2)';  
2 f = inline(s);  
3 x = rand(300,2)*2*pi-pi;  
4 y = f(x(:,1),x(:,2));  
5 plot3(x(:,1),x(:,2),y, 'b.')
```

Below the code, there is a section for "Exercise 2" which is currently empty. At the bottom of the editor, a window titled "Neural Network Start (nnstart)" is open. It features a brain icon and the text "Welcome to Neural Network Start" and "Learn how to solve problems with neural networks." There are two tabs: "Getting Started Wizards" (selected) and "More Information". The "Getting Started Wizards" tab contains the following text: "Each of these wizards helps you solve a different kind of problem. The last panel of each wizard generates a MATLAB script for solving the same or similar problems. Example datasets are provided if you do not have data of your own." Below this text are four categories of wizards, each with a button and a tool name in parentheses:

- Input-output and curve fitting. [Fitting app](#) (nftool)
- Pattern recognition and classification. [Pattern Recognition app](#) (nprtool)
- Clustering. [Clustering app](#) (nctool)
- Dynamic Time series. [Time Series app](#) (ntstool)

On the right side of the Live Editor, a 3D scatter plot is displayed. The plot shows a cloud of blue dots in a 3D coordinate system. The vertical axis (z-axis) ranges from -2 to 2, and the horizontal axes (x and y) range from -4 to 4. The dots are scattered in a roughly spherical pattern.

Exercise 4: Write a script for tasks in problems A, B and C

Problem A. Draw the following function in a figure

$$f(x) = \exp(\cos(x))$$

where each point $(x) \in [-5,5]$

Problem B. Data Sampling: Let $n = 300$. Generate a $n \times 1$ matrix x , where each row denotes a point in the square $[-5,5]$, and a $n \times 1$ vector y such that $y[i] = f(x[i])$, $i = 1..n$

Problem C. Draw sampled points, represented by x and y , in a figure

Exercise 5: Neural fitting for approximating $\exp(\cos(x))$

Problem A. Import x and y for neural fitting

Problem B. Train a neural network for function approximation subject to x and y

Problem C. Try to print the final screen of neural fitting

Problem D. Try to draw the neural function