Curve fitting
2-dimensional function fitting

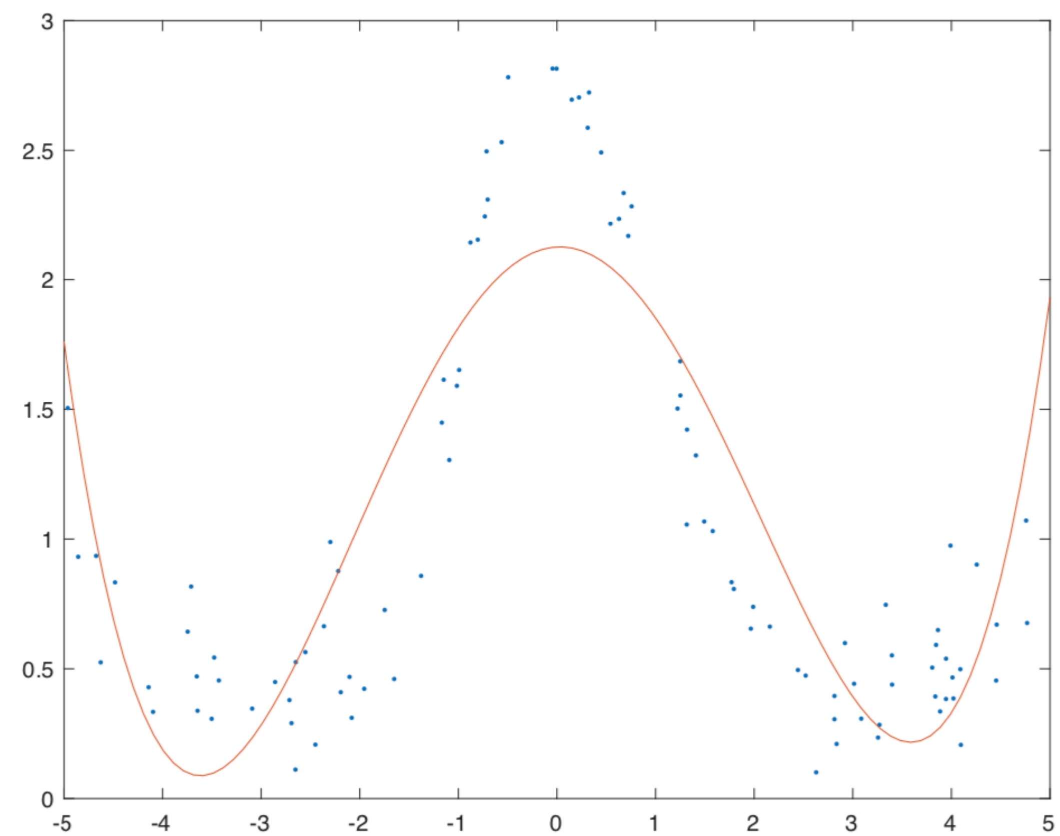
Approximate $\exp(\cos(x))$ using neural functions

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 = ' [redacted] ';
f = inline(str);
y = [redacted] + noise;
plot(x, y, '.'); hold on;

p = [redacted] ;
x_new = linspace(–5,5);
y_new = polyval(p, x_new);

plot(x_new, y_new)
```

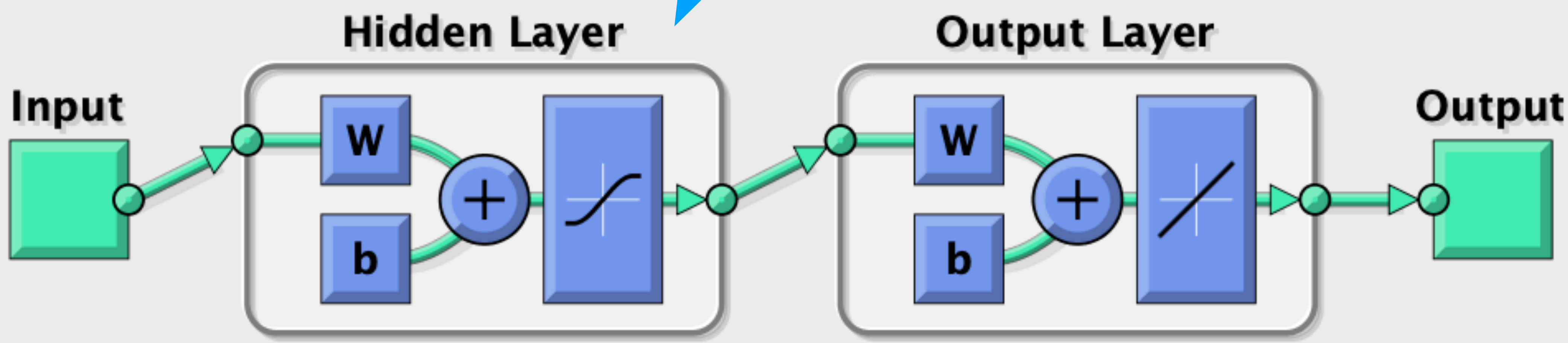


p =

0.0117 -0.0001 -0.3041 0.0189 2.1259

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

Neural Network



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.

$x \in R^d$

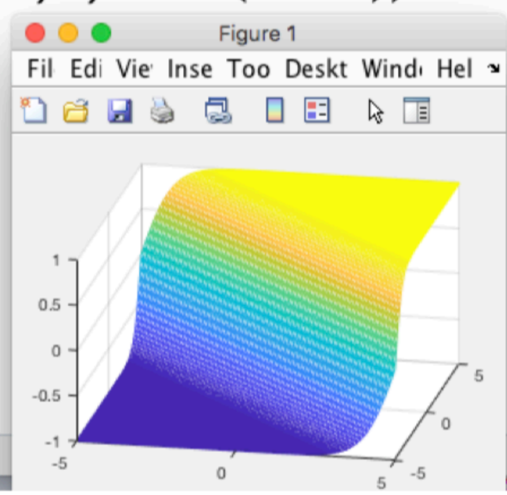
$$h = Wx + b, \quad W : m \times d, \quad b : m \times 1,$$

$$h : m \times 1$$

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

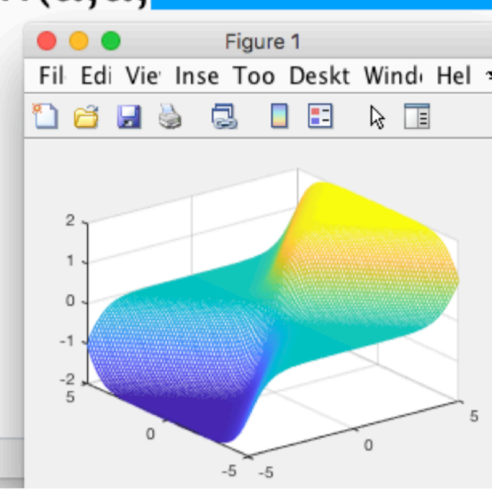
$$f(x_1, x_2) = \tanh(x_1 + 0.5 * x_2 - 1) + \tanh(0.5x_1 - 0.8x_2 + 0.5)$$

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

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

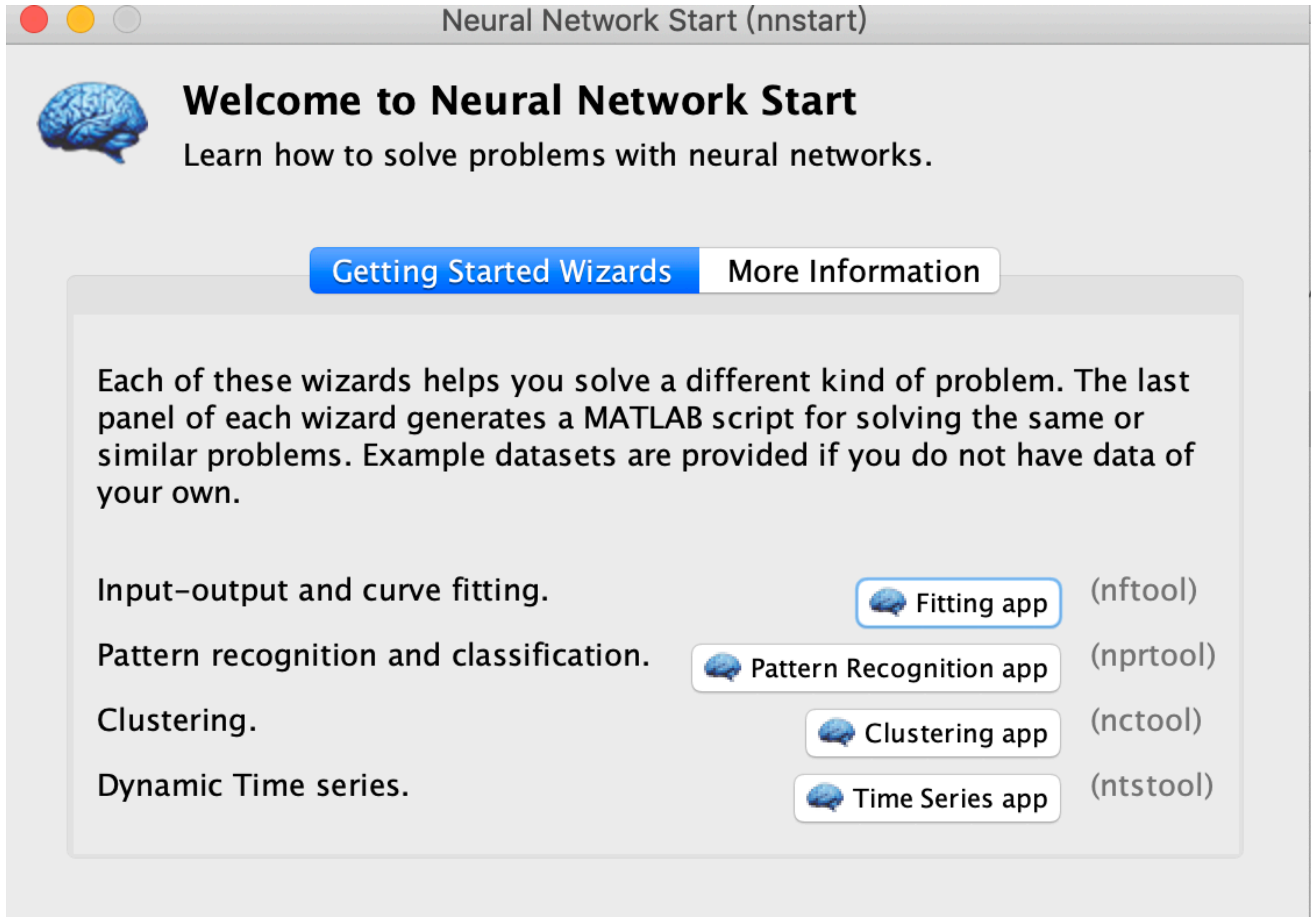
where each point $(x_1, x_2) \in [-\pi, \pi] \times [-\pi, \pi]$

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

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

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

How to use neural fitting



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

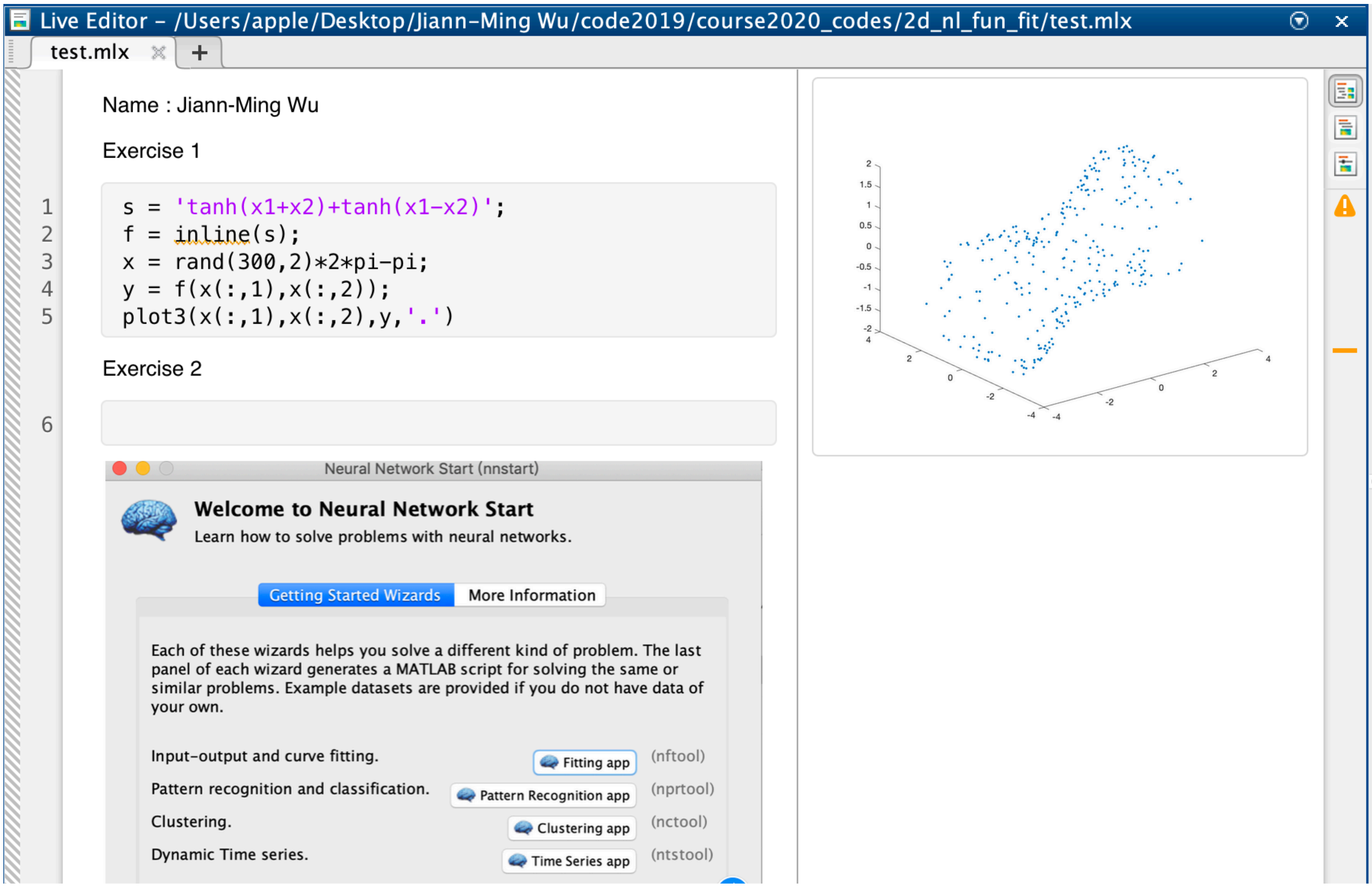
Exercise 3:

Problem A. Use the following code to train a neural network for mapping x to y, and calculate mean((y_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



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

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