Answers to questions about part1

1. ***Assuming that you number the tiles in the natural way, the tiles in the first tiling will run from 0 to 120, and the tiles in the second tiling will run from 121 to 241 (why?)***

   Since each tiling is an 11x11 grid, each tiling has actually 11x11=121 blocks. Notice that the index starts from 0, thus the first tiling will run from 0 to 120. Thus, the second tiling starts from 0+121*1=121 to 120+121*1=241, that is run from 121 to 241.

2. ***For example, the point from the first example in the training set above, in1=0.1 and in2=0.1, or 0.1,0.1, will be in the first tile of the first seven tilings, that is, in tiles 0, 121, 242, 363, 484, 605, 726 (why?)***

   First of all, each grid of the tiling is a square in width of 0.6 and height of 0.6. After every time of shifting, in the next tiling, the relative coordinate of one given point will be offset by 0.6/8=0.075 in both directions.
   Therefore, for the given point 0.1,0.1:
   a) the coordinate of point, in the 1st tiling, is (0.1, 0.1) is inside of the first grid of the first tiling, the index of this grid is 0.
   b) the coordinate of the point, in the 2nd tiling, is (0.1+0.075,0.1+0.075)=(0.175, 0.175), is inside of the first grid of the second tiling, the corresponding index is 121. .
   c) the coordinate of the point, in the 3rd tiling, is (0.1+0.075*2,0.1+0.075*2)=(0.25, 0.25), is inside of the first grid of the third tiling, the corresponding index is 242.
   d) the coordinate of the point, in the 4th tiling, is (0.1+0.075*3,0.1+0.075*3)=(0.325, 0.325), is inside of the first grid of the fourth tiling, the corresponding index is 363.
   e) the coordinate of the point, in the 5th tiling, is (0.1+0.075*4,0.1+0.075*4)=(0.4, 0.4), is inside of the first grid of the fifth tiling, the corresponding index is 484.
   f) the coordinate of the point, in the 6th tiling, is (0.1+0.075*5,0.1+0.075*5)=(0.475, 0.475), is inside of the first grid of the sixth tiling, the corresponding index is 605.
   g) the coordinate of the point, in the 7th tiling, is (0.1+0.075*6,0.1+0.075*6)=(0.55, 0.55), is inside of the first grid of the seventh tiling, the corresponding index is 726.

3. ***In the eighth tiling this point will be in the 13th tile (why?)***
   According to what has been computed in question 2 above, in the 8th tiling, the coordinate of the point, with respect to the tiling, is (0.1+0.075*7,0.1+0.075*7)=(0.625, 0.625). Since in the tiling system, each grid is a square of width 0.6 and height 0.6. Therefore, the tiling is no longer located in the first tile of the 8th tiling(which is of range(0-0.6))again. Instead, it is now in the tile to the top-right of the first tile – the 13th tile.

4. ***which is tile 859 (why?)***
   From question 3, we have that the point will be in the 13th tile of the 8th tiling. Because

the indexing of tiles are in an order from 0-967. The first tile of the 8th tiling is 847, and the 13th tile is of index 859, accordingly.

5. ***If you call tilecode(0.1,0.1,tileIndices), then afterwards tileIndices will contain exactly these eight tile indices. The largest possible tile index is 967 (why?)***
   Since we have 8 tiling in total, and each tiling is a size of 11x11=121. When the top-left of the input space is the same as the 11x11 gird, we have the largest possible index 11*11*8-1=967.

6. ***Finally, the second and fourth examples should produce very similar sets of indices (they should have many tiles in common) (why?)***
   The second example has input (4,2) and the forth has (4,2.1). These two are close to each other, so are potentially to be in the same tiling. Therefore, they should produce very similar sets of indices.

Explanation of part 2

1. ***After only 20 examples, your learned function will not yet look like the target function. Explain in a paragraph why it looks the way it does. If your learned function involves many peaks and valleys, then be sure to explain both their number, their height, and their width.***
   The learned function is not yet look like the target function because we only have 20 examples here which is not large enough to explore all the states. In the graph, we have most place to be flat because they are not explored, but we do have several peaks and valleys.
   In the plot, we have 2 peaks and 2 valleys that are significantly obvious. The peaks are in the point(5.38, 0.157) with a height of 0.137 and (1.5, 3.8) with a height of 0.14. The valleys are in the point(1.7, 0.36) with a height of -0.08 and (4.19, 2.16) with a height of -0.06. Their heights, representing the value, correspond to the target value. Their width is caused by the noise exerted on the target function. The indexes of the peaks and valleys are just alike the ones from the F10000 graph. Therefore, the values of the peaks and valleys are within the correct range with respect to the true value which is shown in the F10000 graph. Since the points are visited by the algorithm, they learned their corresponding values.

2. ***Suppose that, instead of tiling the input space into an 11x11 grid of squares, you had divided into an 11x21 grid of rectangles, with the in1 dimension being divided twice as finely as the in2 dimension. Explain how you would expect the function learned after 20 examples to change if this alternative tiling were used.***
   Since the in1 dimension will be divided twice as finely as the in2 dimension, we will have more accurate approximation and smoother graph than the former one, although it may take us longer to explore.

Answers to questions about part 2

1. *The before value of the fourth point should be nonzero (why?)*
   As we seen before, the forth example (4,2.1) is very close to the second example (4,2), so they produce very similar sets of indices. As a result, we learn the second example (4,2) first and update the weight. When we learn the forth example (4,2) afterwards, we have an updated weight for it instead of zero. Therefore, we will get a nonzero before value of the forth point.

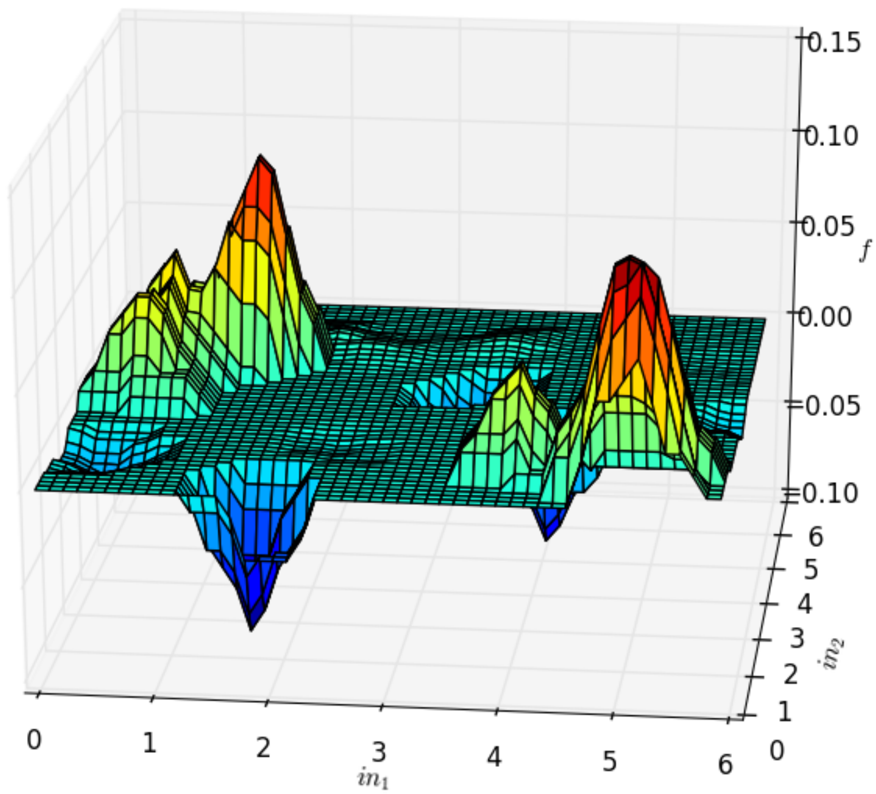
2. *You should see the MSE coming down smoothly from about 0.25 to almost 0.01 and staying there (why does it not decrease further towards zero?)*
   Since in the target function we have introduce N(0,1) -- a normally distributed random number with mean 0 and standard deviation 0.1, which cause noise in the target function that can not be reduced by the learning process. Therefore, we will always have MSE≥0.01

MSEs

printout from step 3:
Example (0.1 , 0.1 , 3.0):   f before learning: 0.0   f after learning : 0.30000000000000004
Example (4.0 , 2.0 , -1.0):   f before learning: 0.0   f after learning : -0.09999999999999999
Example (5.99 , 5.99 , 2.0):   f before learning: 0.0   f after learning : 0.19999999999999998
Example (4.0 , 2.1 , -1.0):   f before learning: -0.075 f after learning : -0.16749999999999998

the MSEs printed from step 4:
The estimated MSE: 0.252139017816
The estimated MSE: 0.0575318556046
The estimated MSE: 0.0212079824436
The estimated MSE: 0.0141191057181
The estimated MSE: 0.0121817554761
The estimated MSE: 0.011842712039
The estimated MSE: 0.0116831759436
The estimated MSE: 0.0112201263821
The estimated MSE: 0.0112019183338
The estimated MSE: 0.011330711048
The estimated MSE: 0.0112135975801

F20



F10000