CW2

Aaron Song s267807

November 9, 2020

1 Task 1

1. An entry of data in this problem is defined as $(x, y, label)$, where x and y is the inputs and $label$ is the output. Define:

data Data: x:number, y:number, label: number

neural network A neural network in this task is an object with a function that takes a pair of numbers of outputs a number, so functionally define Apply: number-¿number-¿number and NeuralNetwork: Apply .

loss function Loss(nn: NeuralNetwork, d: Data) = ((Apply d.x d.y)-d.error)^2

fitness functtion The fitness of a neural network is the average loss on the specific dataset.

$$Ftiness(nn : NeuralNetwork, ds : Data[]) = \frac{\sum_{i=1}^{\text{ds.Length}} Loss(nn, ds[i])}{ds.Length}$$

2. A [-10, 10] is enough for this question.
3. Two hidden layers are used, with the first 5 7 neurons and the second 2 neurons. If only one hidden layer is permitted, 7 neurons are prefered to get a better result, as shown in 1.
4. The learning result is much better when non-linear input are involved. For this question specifically, a spiral data is generated by a formula $r = k\theta$ where $k > 0, \theta > 0$. When translating the polar coordinate into a Cartisean coordinate, it's

$$\sqrt{x^2 + y^2} = k(2n\pi + \text{atan2}(x, y)) \tag{1}$$

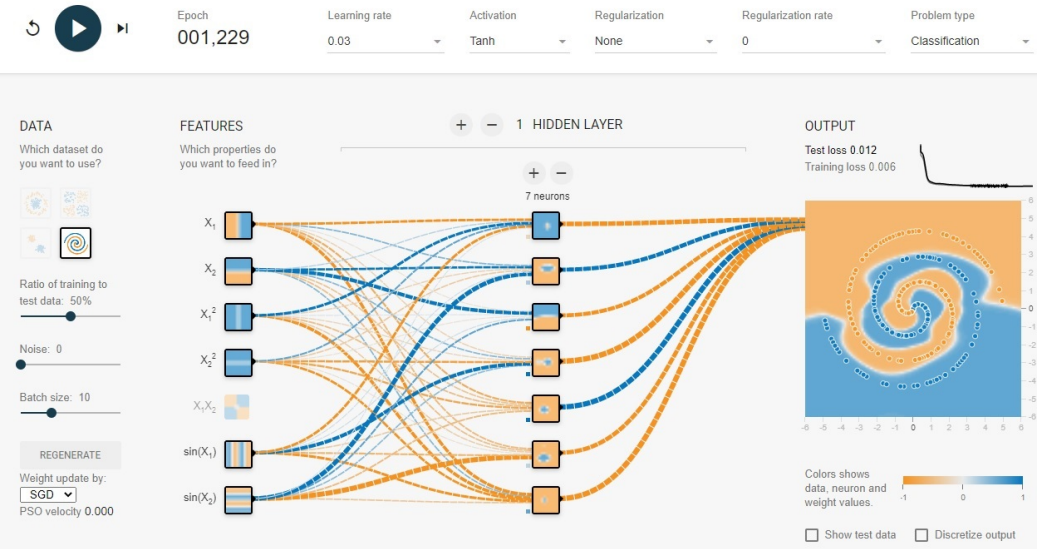

Figure 1: A hidden layer with 7 neurons

, where $n \in Z$ and atan2 is the arctan function defined by most common programming language where the range is $[0, 2\pi)$. In order to expand the aforementioned equation into a plain form, atan2 can be approximately translated into arctan, and according to Tylor expansion,

$$\frac{\mathrm{d}\arctan x}{\mathrm{d}x} = \frac{1}{x+1}\arctan x = x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \dots \tag{2}$$

. Curtailling high order terms,

$$\arctan x = x - \frac{1}{3}x^3 x^2 + y^2 = an + b(x - \frac{1}{3}x^3)^2$$

where $a = \frac{4k^2}{\pi^2}, b = k^2$. Curtailling high order terms again,

$$x^2 + y^2 = an + bx$$

We can see that x^2, y^2, terms are presented here. To fully simulate the spiral data generation function 1, $\sin x$ and $\sin y$ may also be included given the Tylor expansion:

$$\sin x = x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \dots$$

5. Many PSO parameters have an effect to this particular problem.

learning rate The learning rate decides how fast the weights and biases of the neural network changes. A moderate value (0.03 or 0.1) yields good output, otherwise if it's too high (≥ 1) the training loss and the test lost will fluctuate too fast and fail to converge to a proper local best. If too small (≤ 0.0003), the weights and the biases barely change. A global best may still present itself after enough long time, but that may take too many epochs.

activation function Activation function is used to discretse the floating-number output of the neural network into a number. The default activation tanh in this problem suits very well, and sigmoid also fits good despite longer training time may be required. RBF and ReLU are also acceptable. Sin doesn't fit well since the radius is not constant, and linear doesn't work at all.

regularisation

feature The features in this problem are the soul source of non-linear input and decide the form of the neural network; the number of layers and the number of neurons on each layer depend only the linear combination of the input features. As memtioned in the previous question, x_1^2 and $\sin x$ improve the learning result a lot compared to a purely linear model. I also tried x_1^3 and x_2^3, but the result is not satisfying.

nn shape Every neural node blends all inputs linearly and yields one single value. Mathematically, suppose there are x layers (including the input layer) of nodes in the neural network and on layer i there are $k[i]$ nodes ($0 \leq i < x$), then node b on layer $0 < a \leq x$ $n_{a,b}$ takes a vector $X = (x_0, x_1, \ldots, x_{k[a-1]})$ as input and output $A \cdot X + B$, where A stands for the weight between $a-1$ layer and node $n_{a,b}$ and B represents the bias of node $n_{a,b}$.

In other words, the input layer is defined by some common function $y = x$, $y = x^2$, $y = \sin x$, and nodes in a following layers form a linear combination of the last layer.

2 Task 2

1. There're seven input nodes so the encoding of the input layer would be a binary string with a length of 7; the maximum number of layers is 6 and on

each layer the limit of the number of neurons is 8, so a layer can be encoded in a binary string with a length of 3. In the first layer (also the input layer), every node is different because each one represents a different input function (it is however possible to have to input nodes with the same function, but it would be meaningless); for other layers, every node (neuron) is the same. So the neural network can be represented as:

$$i_0 i_1 \dots i_7,\ l_{1_0} l_{1_1} l_{1_2},\ l_{5_0} l_{5_1} l_{5_2},\ \dots,\ l_{6_0} l_{6_1} l_{6_2}$$

.