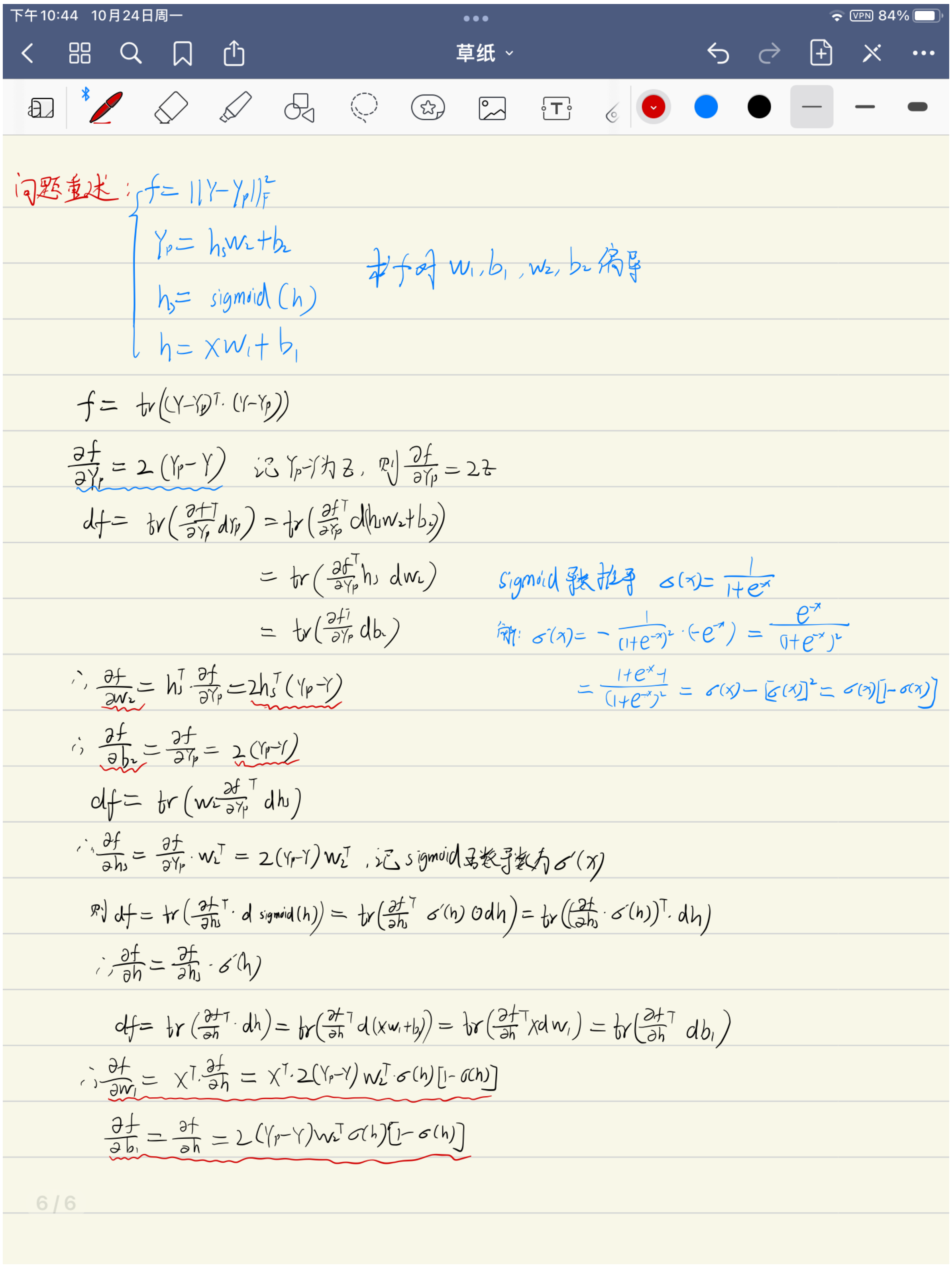
计算机视觉第五次作业

岳东旭 2201212864 指导教师：张健

1.手动推导



问题重述：
$$\begin{cases} f = \|Y - Y_p\|_F^2 \\ Y_p = h_s w_2 + b_2 \\ h_s = \mathrm{sigmoid}(h) \\ h = X w_1 + b_1 \end{cases}$$
求 f 对 w_1, b_1, w_2, b_2 偏导

$$f = tr\left((Y-Y_p)^T \cdot (Y-Y_p)\right)$$

$\underbrace{\frac{\partial f}{\partial Y_p} = 2(Y_p - Y)}$ 记 $Y_p - Y$ 为 Z，则 $\frac{\partial f}{\partial Y_p} = 2Z$

$$df = tr\left(\frac{\partial f}{\partial Y_p}^T dY_p\right) = tr\left(\frac{\partial f}{\partial Y_p}^T d(h_s w_2 + b_2)\right)$$
$$= tr\left(\frac{\partial f}{\partial Y_p}^T h_s\, dw_2\right)$$
$$= tr\left(\frac{\partial f}{\partial Y_p}^T db_2\right)$$

sigmoid 函数推导 $\sigma(x) = \frac{1}{1+e^{-x}}$

解: $\sigma'(x) = -\frac{1}{(1+e^{-x})^2} \cdot (-e^{-x}) = \frac{e^{-x}}{(1+e^{-x})^2}$
$$= \frac{1+e^{-x}-1}{(1+e^{-x})^2} = \sigma(x) - [\sigma(x)]^2 = \sigma(x)[1-\sigma(x)]$$

$\therefore \underbrace{\frac{\partial f}{\partial w_2}} = h_s^T \cdot \frac{\partial f}{\partial Y_p} = \underbrace{2h_s^T (Y_p - Y)}$

$\therefore \underbrace{\frac{\partial f}{\partial b_2}} = \frac{\partial f}{\partial Y_p} = \underbrace{2(Y_p - Y)}$

$$df = tr\left(w_2 \frac{\partial f}{\partial Y_p}^T dh_s\right)$$

$\therefore \frac{\partial f}{\partial h_s} = \frac{\partial f}{\partial Y_p} \cdot w_2^T = 2(Y_p - Y) w_2^T$ ，记 sigmoid 函数导数为 $\sigma'(x)$

则 $df = tr\left(\frac{\partial f}{\partial h_s}^T \cdot d\,\mathrm{sigmoid}(h)\right) = tr\left(\frac{\partial f}{\partial h_s}^T \sigma'(h) \odot dh\right) = tr\left(\left(\frac{\partial f}{\partial h_s} \cdot \sigma'(h)\right)^T \cdot dh\right)$

$\therefore \frac{\partial f}{\partial h} = \frac{\partial f}{\partial h_s} \cdot \sigma'(h)$

$$df = tr\left(\frac{\partial f}{\partial h}^T \cdot dh\right) = tr\left(\frac{\partial f}{\partial h}^T d(Xw_1 + b_1)\right) = tr\left(\frac{\partial f}{\partial h}^T X dw_1\right) = tr\left(\frac{\partial f}{\partial h}^T db_1\right)$$

$\therefore \underbrace{\frac{\partial f}{\partial w_1} = X^T \cdot \frac{\partial f}{\partial h} = X^T \cdot 2(Y_p - Y) w_2^T \cdot \sigma(h)[1-\sigma(h)]}$

$\underbrace{\frac{\partial f}{\partial b_1} = \frac{\partial f}{\partial h} = 2(Y_p - Y) w_2^T \sigma(h)[1-\sigma(h)]}$

2.搭建两层全连接神经网络

```
%matplotlib inline
import torch
import torch.nn.functional as F
import matplotlib.pyplot as plt
import torch.nn as nn

torch.manual_seed(1)    # reproducible

x = torch.unsqueeze(torch.linspace(-1, 1, 100), dim=1)  # x data (tensor),
shape=(100, 1)
y = x.pow(2) + 0.2*torch.rand(x.size())
```

```
x.shape
```

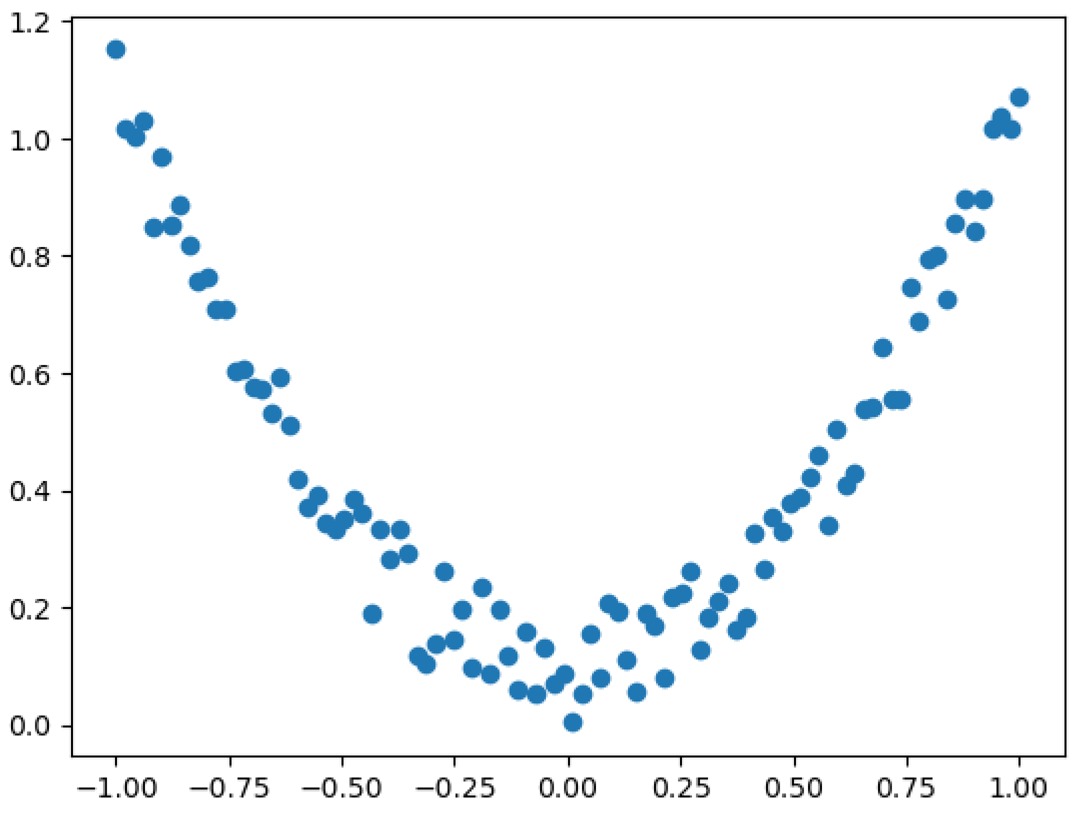
```
torch.Size([100, 1])
```

```
y.shape
```

```
torch.Size([100, 1])
```

```
plt.scatter(x.numpy(), y.numpy())
```

```
<matplotlib.collections.PathCollection at 0x1ce251747c0>
```



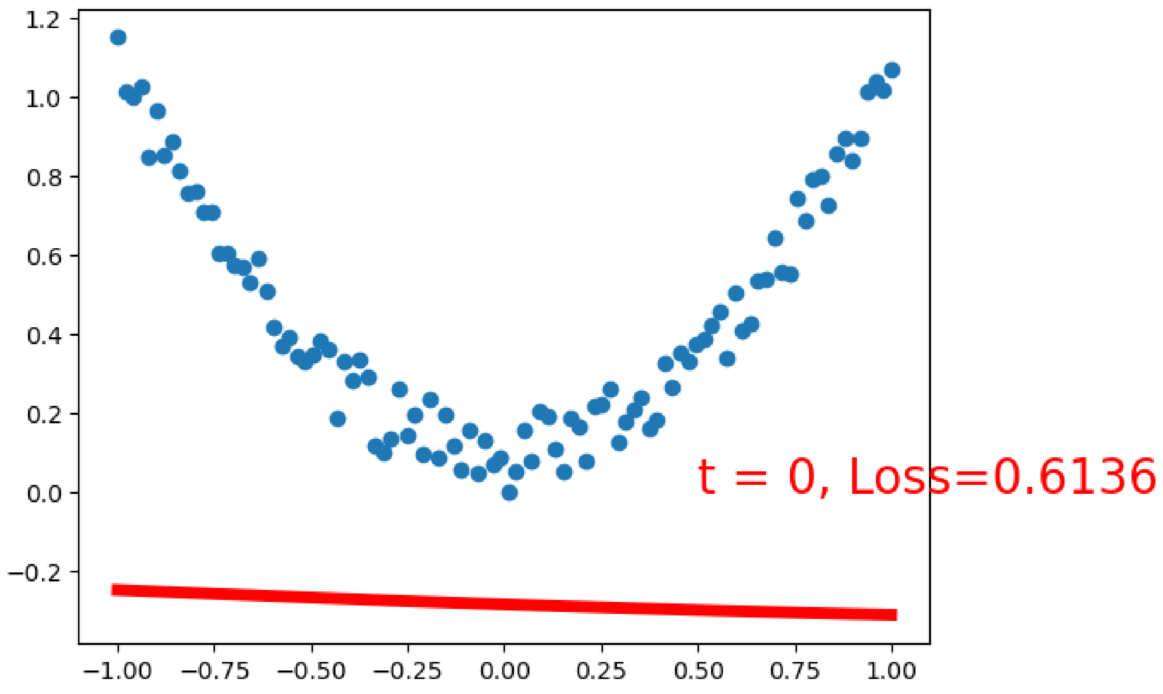
搭建两层含有bias的全连接网络，隐藏层输出个数为20，激活函数都用sigmoid()

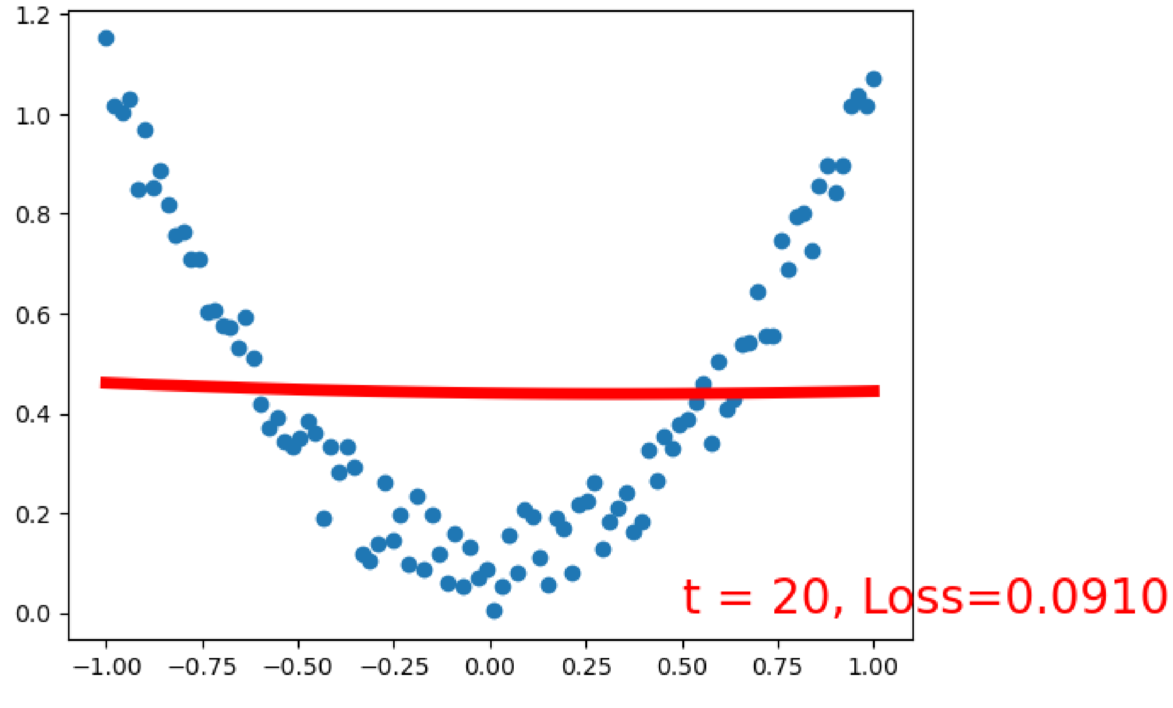
```
class Net(torch.nn.Module):
    def __init__(self, n_feature, n_hidden, n_output):
        super(Net, self).__init__()
        # self.net = nn.Sequential()
        self.linear_1 = nn.Linear(n_feature, n_hidden)
        self.linear_2 = nn.Linear(n_hidden, n_output)
    def forward(self, x):
        x = F.sigmoid(self.linear_1(x))
        x = self.linear_2(x)
        return x
```

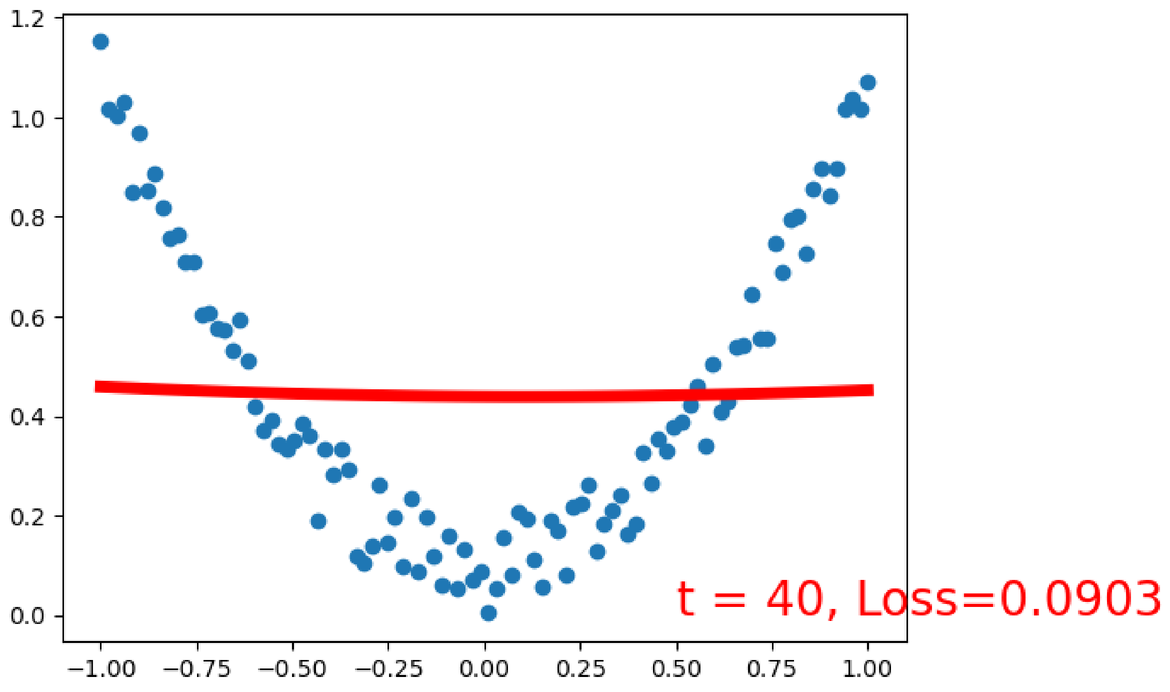
```
net = Net(n_feature=1, n_hidden=20, n_output=1)     # define the network
print(net)  # net architecture
optimizer = torch.optim.SGD(net.parameters(), lr=0.2)
loss_func = torch.nn.MSELoss()  # this is for regression mean squared loss

plt.ion()   # something about plotting

for t in range(2000):
    prediction = net(x)     # input x and predict based on x
    loss = loss_func(prediction, y)     # must be (1. nn output, 2. target)

    optimizer.zero_grad()   # clear gradients for next train
    loss.backward()         # backpropagation, compute gradients
    optimizer.step()        # apply gradients

    if t % 20 == 0:
        # plot and show learning process
        plt.cla()
        plt.scatter(x.numpy(), y.numpy())
        plt.plot(x.numpy(), prediction.data.numpy(), 'r-', lw=5)
        plt.text(0.5, 0, 't = %d, Loss=%.4f' % (t, loss.data.numpy()), fontdict=
{'size': 20, 'color':  'red'})
        plt.pause(0.1)
        plt.show()

plt.ioff()
# plt.show()
```

```
Net(
  (linear_1): Linear(in_features=1, out_features=20, bias=True)
  (linear_2): Linear(in_features=20, out_features=1, bias=True)
)
```

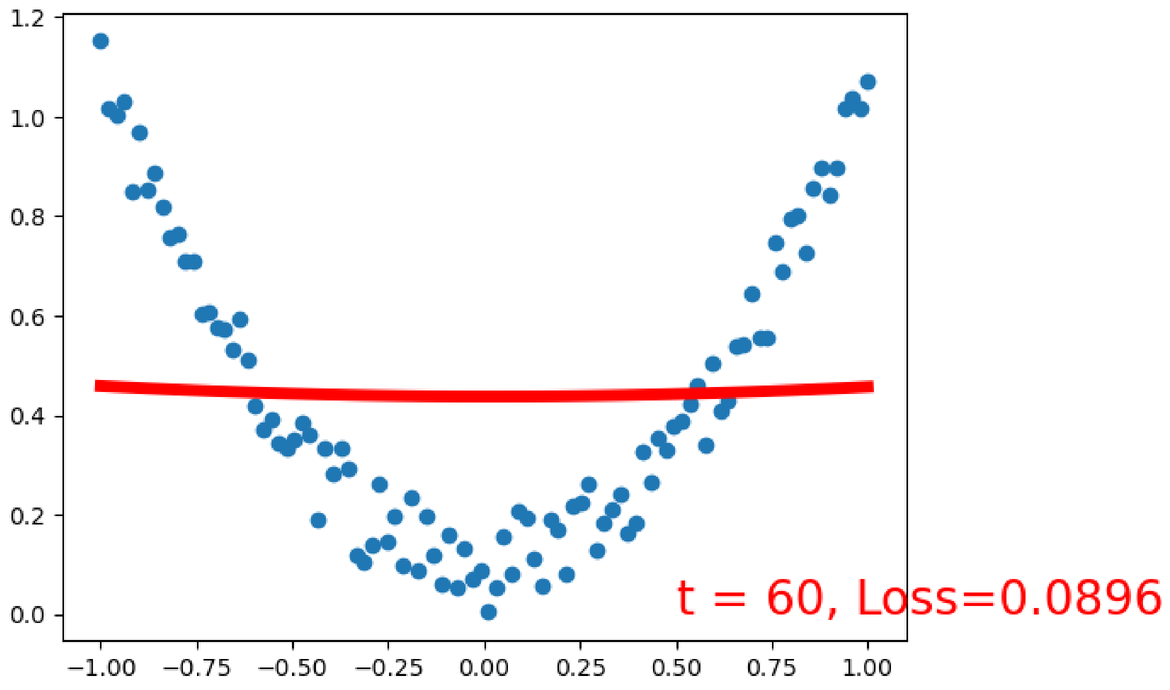
```
D:\anaconda\lib\site-packages\torch\nn\functional.py:1960: UserWarning:
nn.functional.sigmoid is deprecated. Use torch.sigmoid instead.
  warnings.warn("nn.functional.sigmoid is deprecated. Use torch.sigmoid
instead.")
```


t = 0, Loss=0.6136

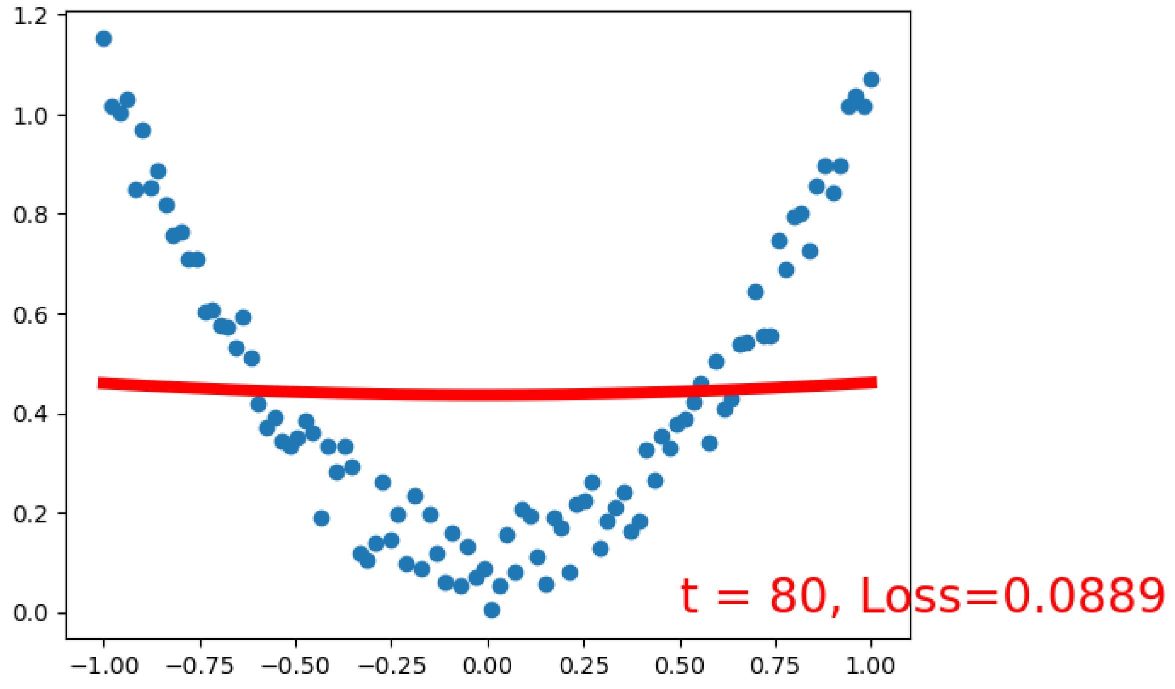

t = 20, Loss=0.0910

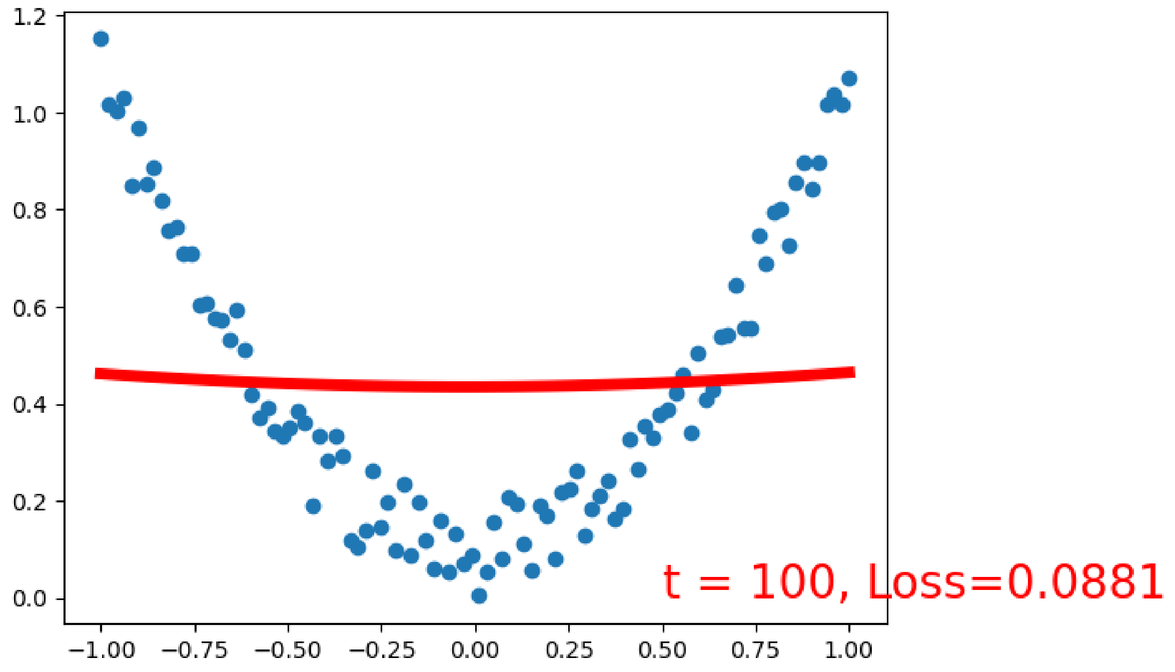


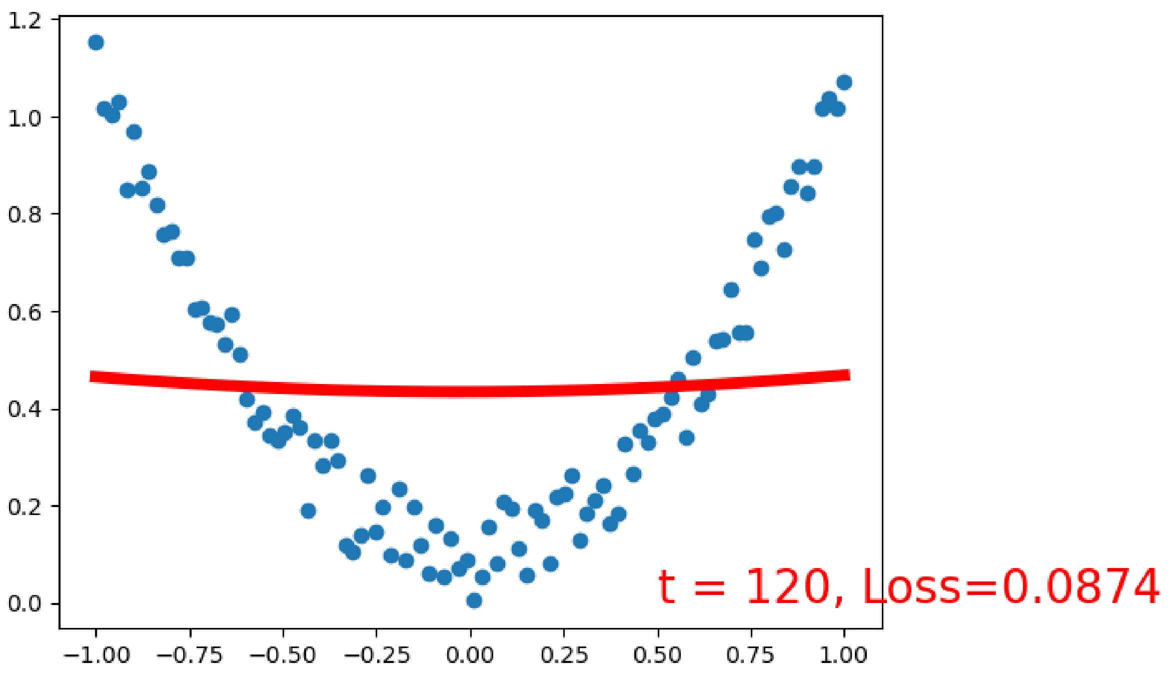
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 40, Loss=0.0903

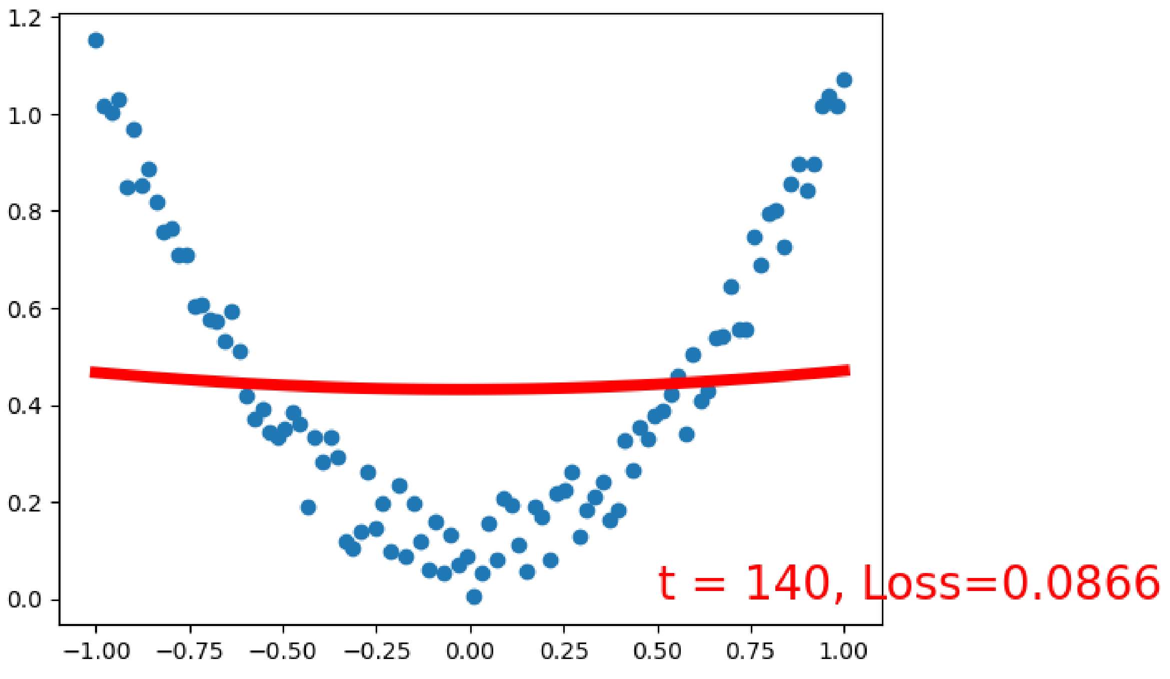


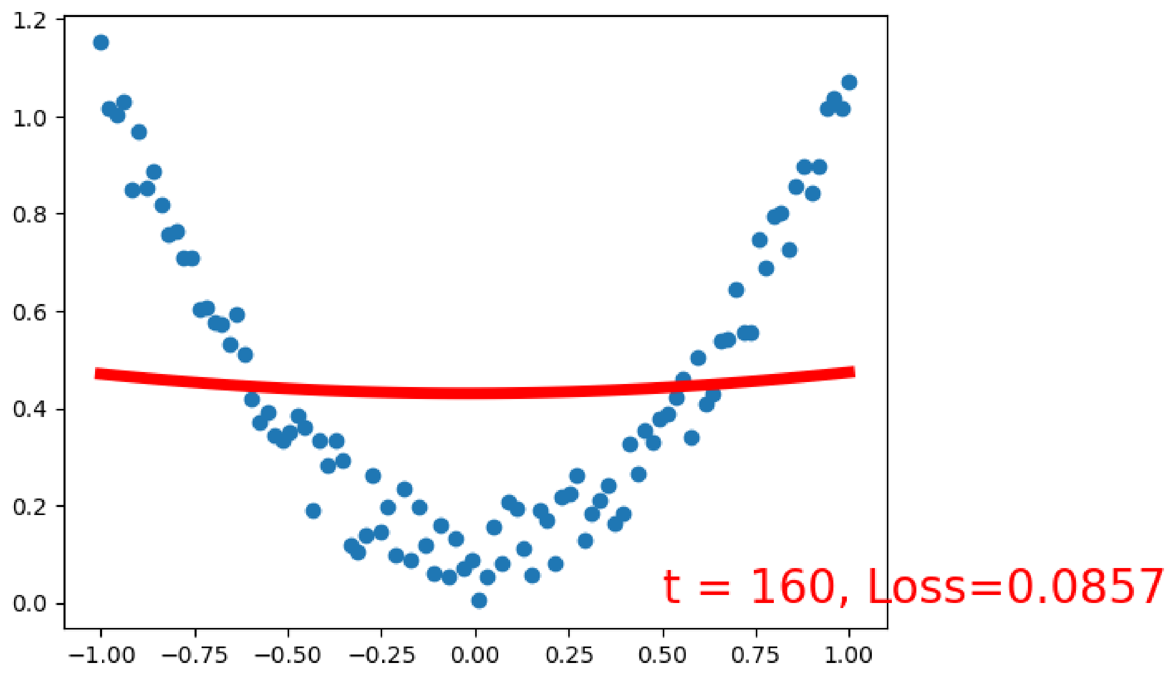
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 60, Loss=0.0896

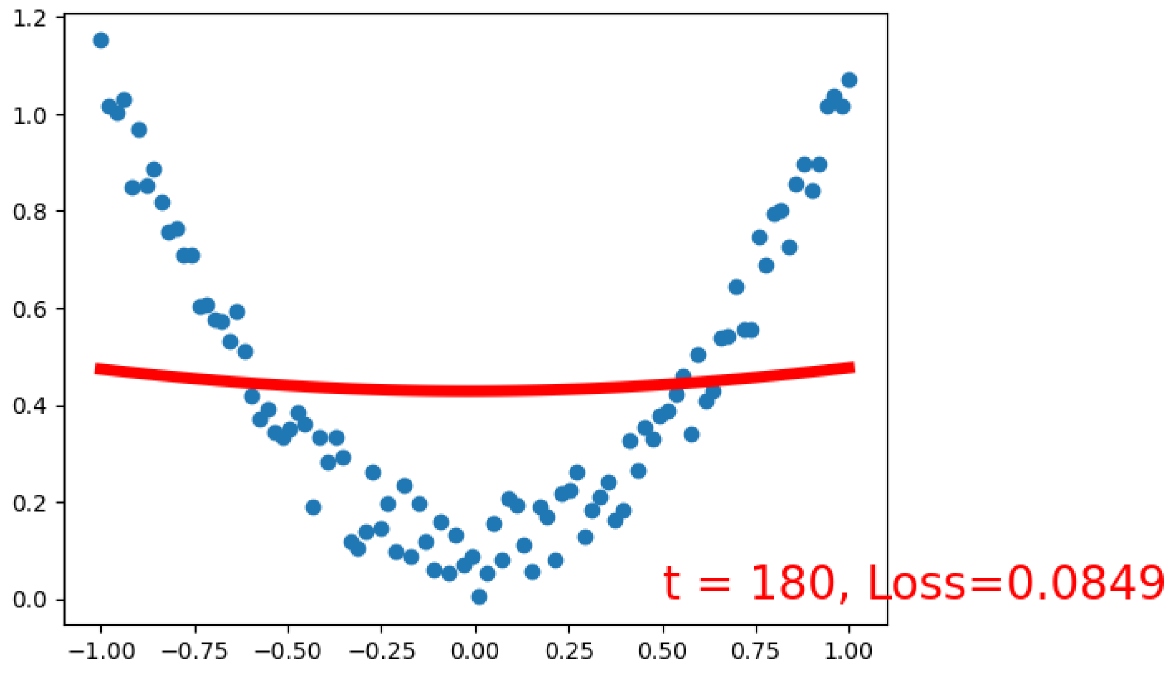

t = 80, Loss=0.0889

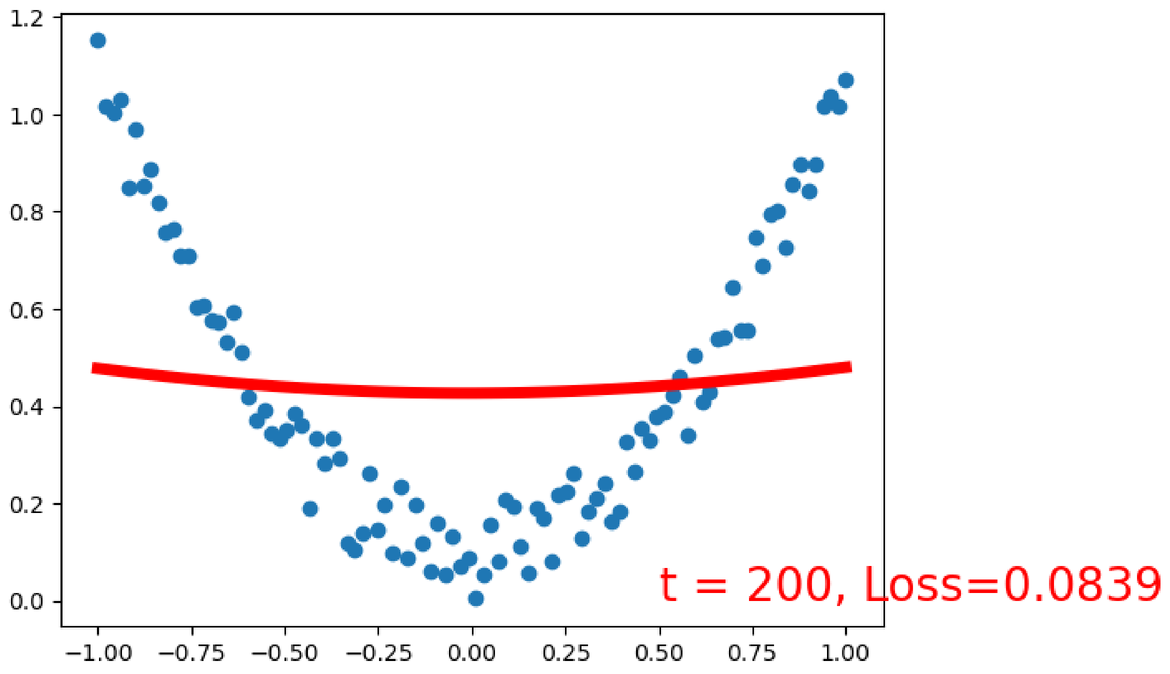

t = 100, Loss=0.0881

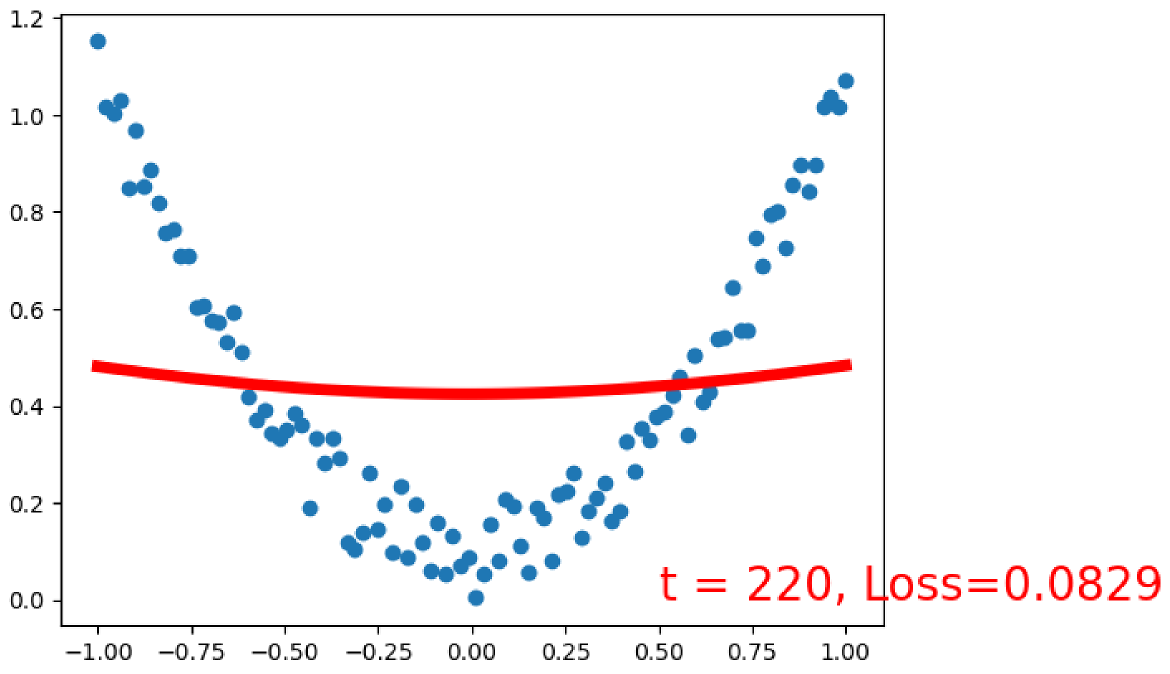

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 120, Loss=0.0874

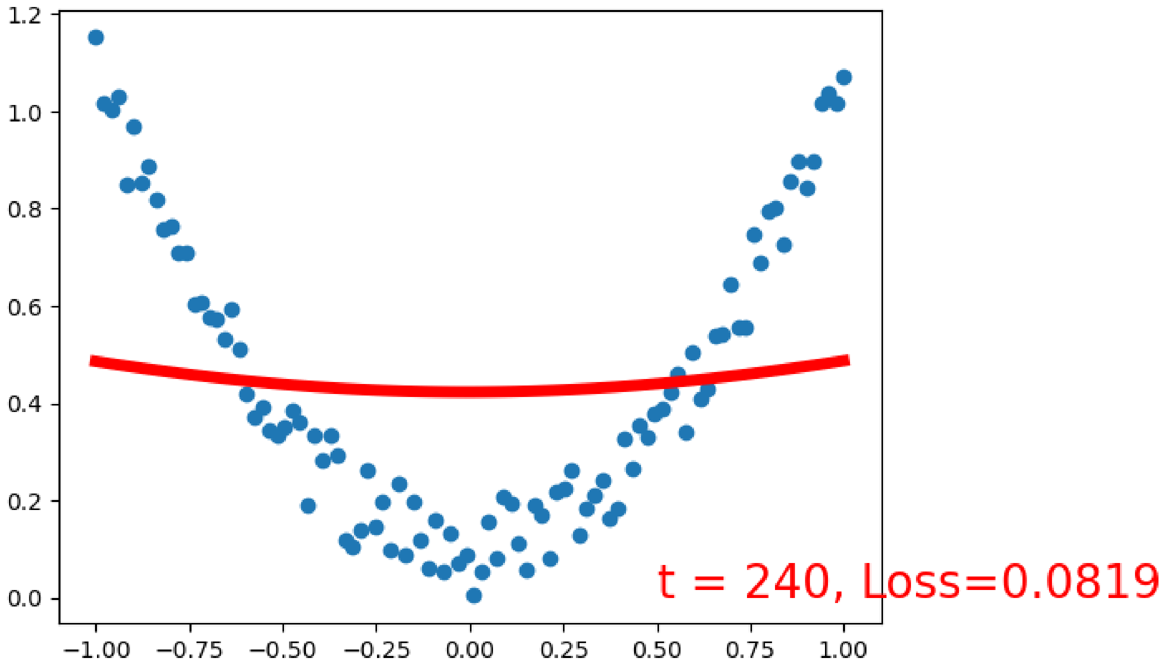

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 140, Loss=0.0866

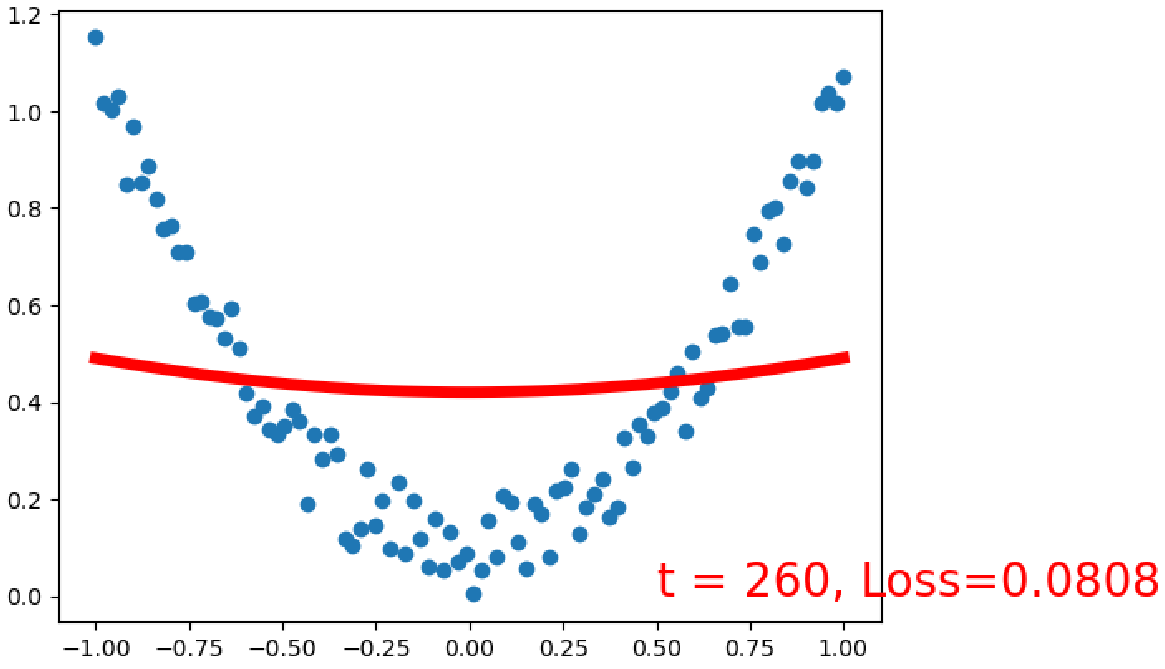

t = 160, Loss=0.0857

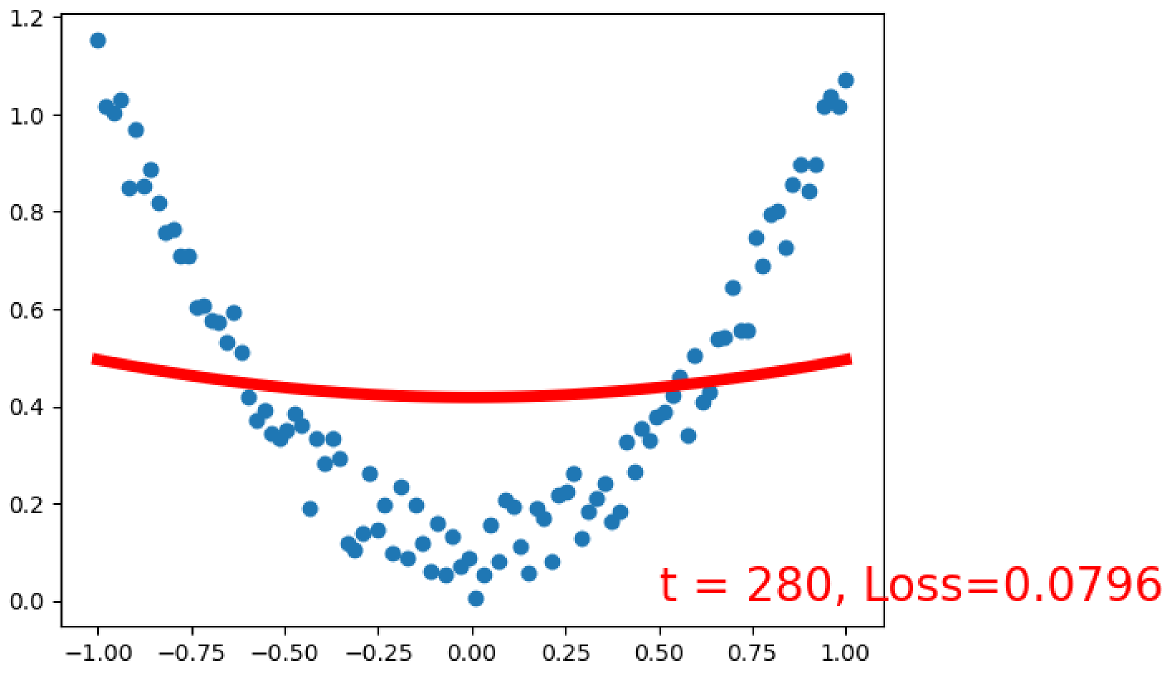

t = 180, Loss=0.0849

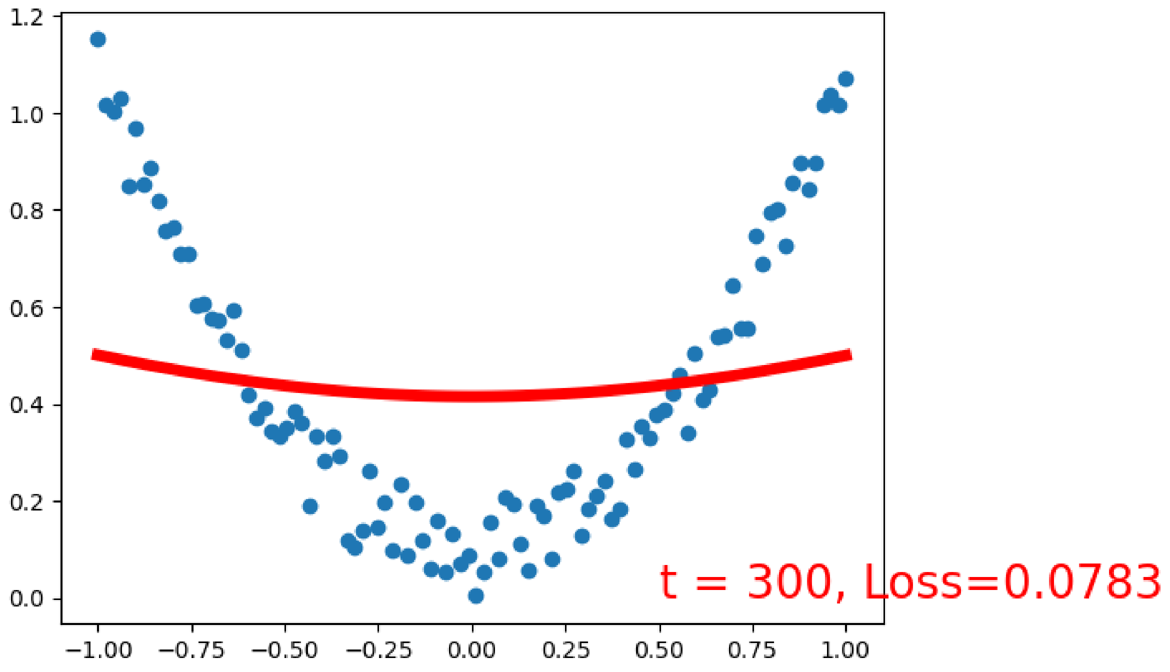

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 200, Loss=0.0839

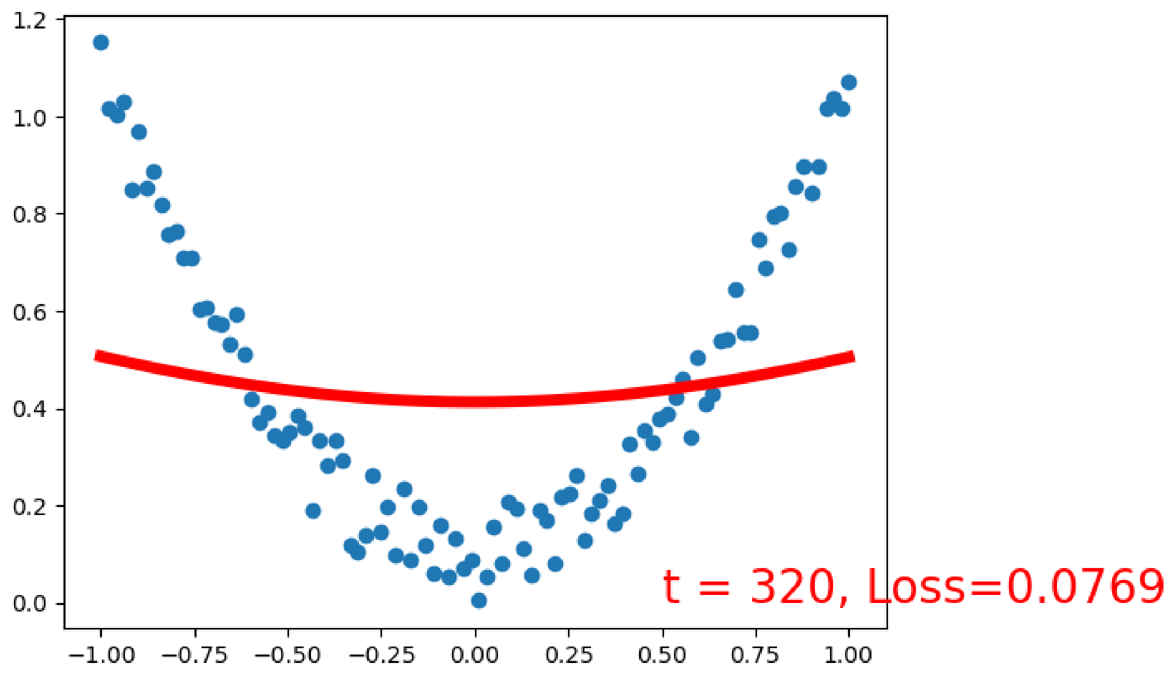

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 220, Loss=0.0829

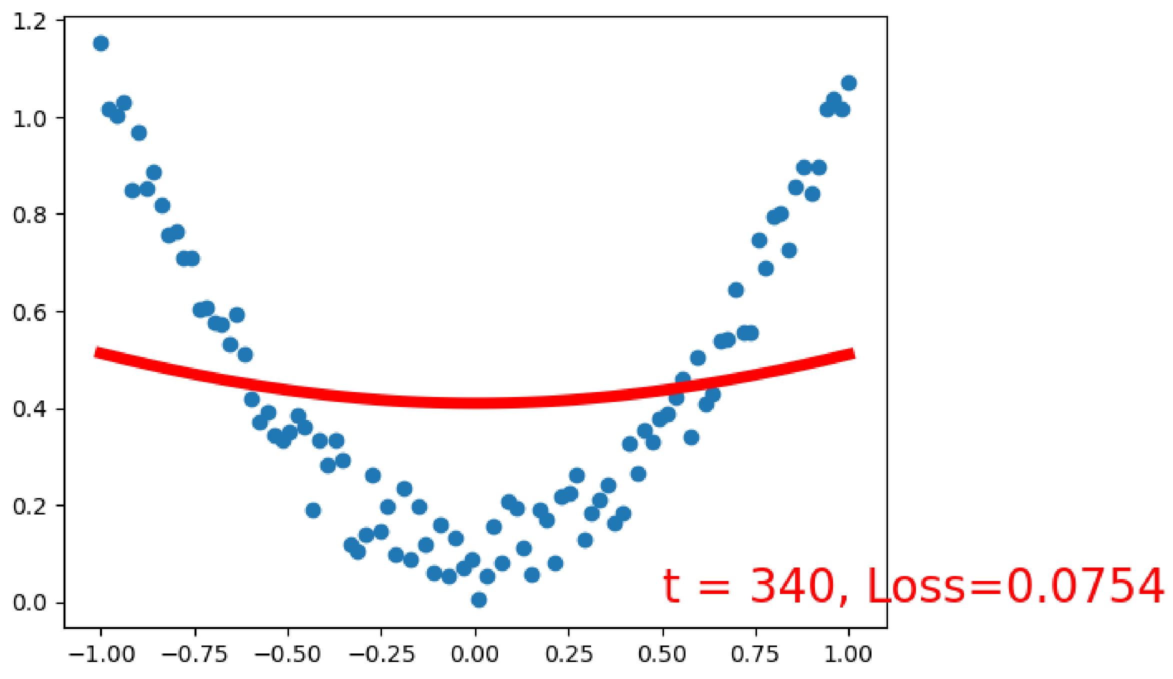

t = 240, Loss=0.0819

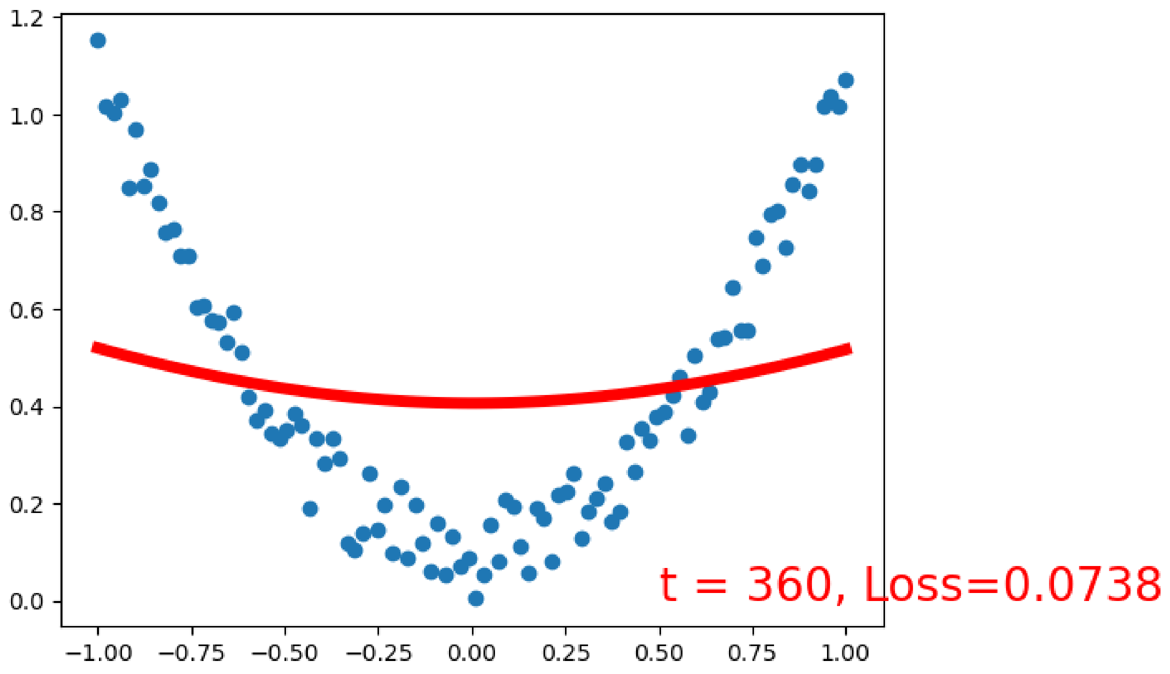

t = 260, Loss=0.0808

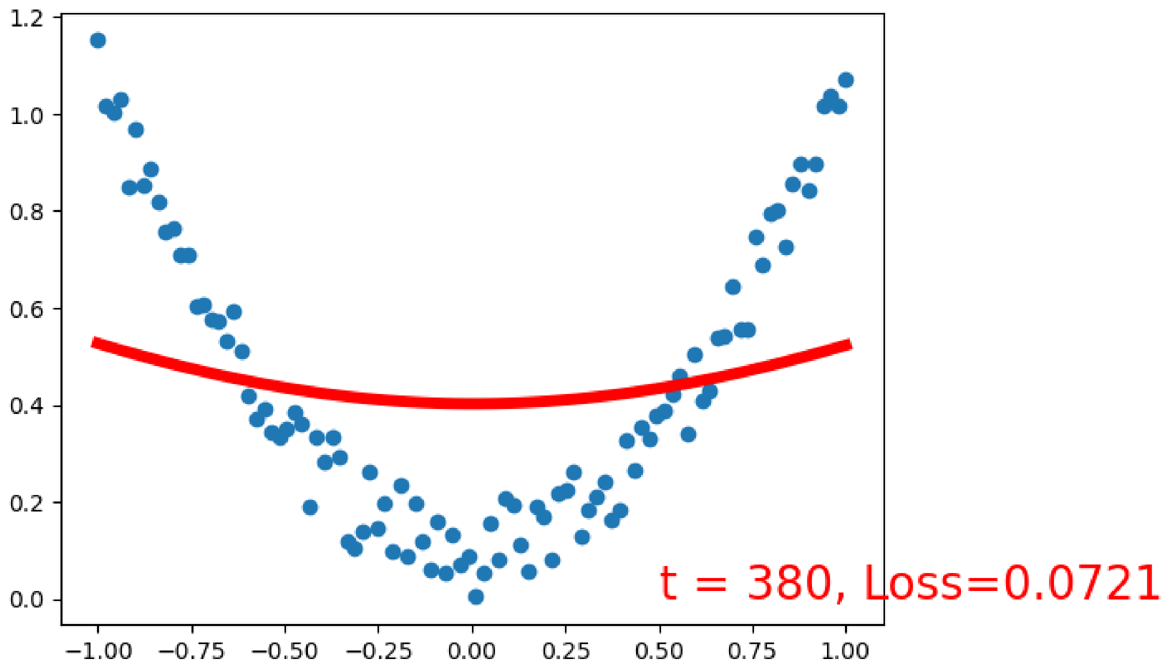

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 280, Loss=0.0796

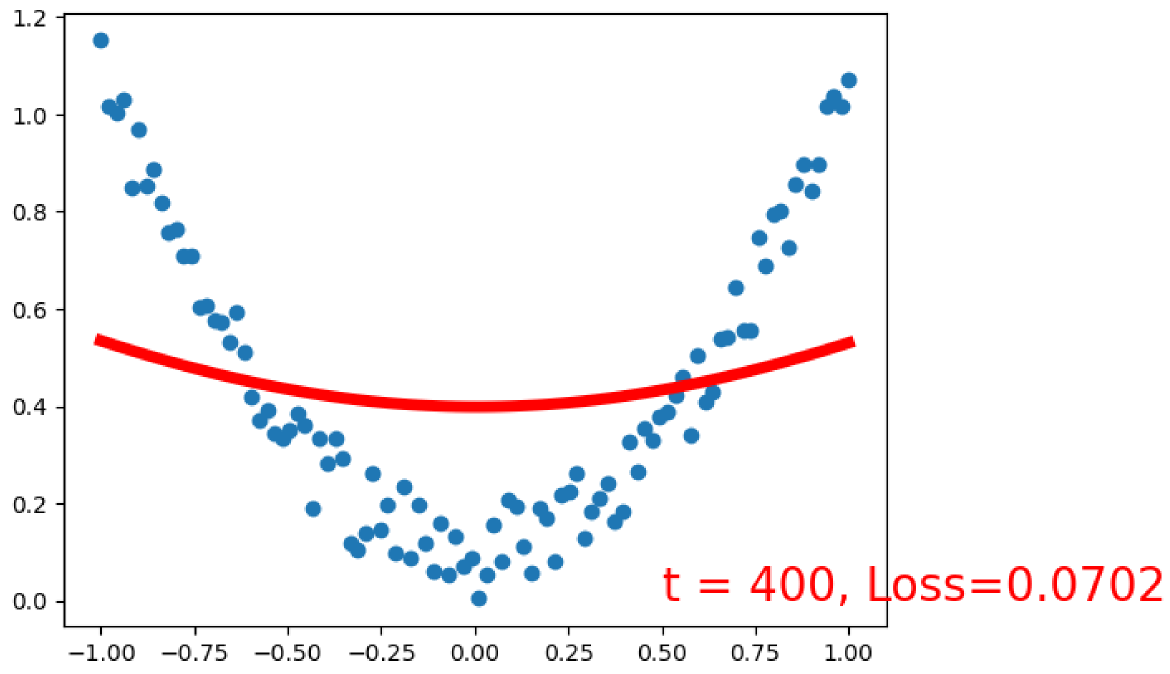

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 300, Loss=0.0783

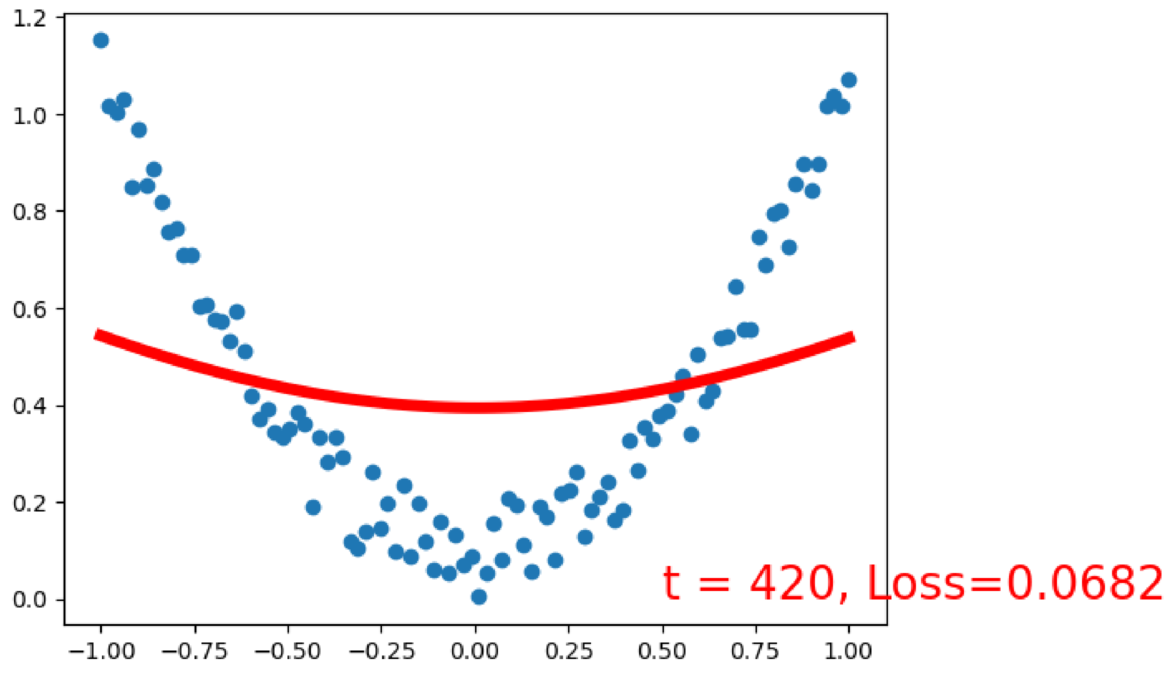

t = 320, Loss=0.0769

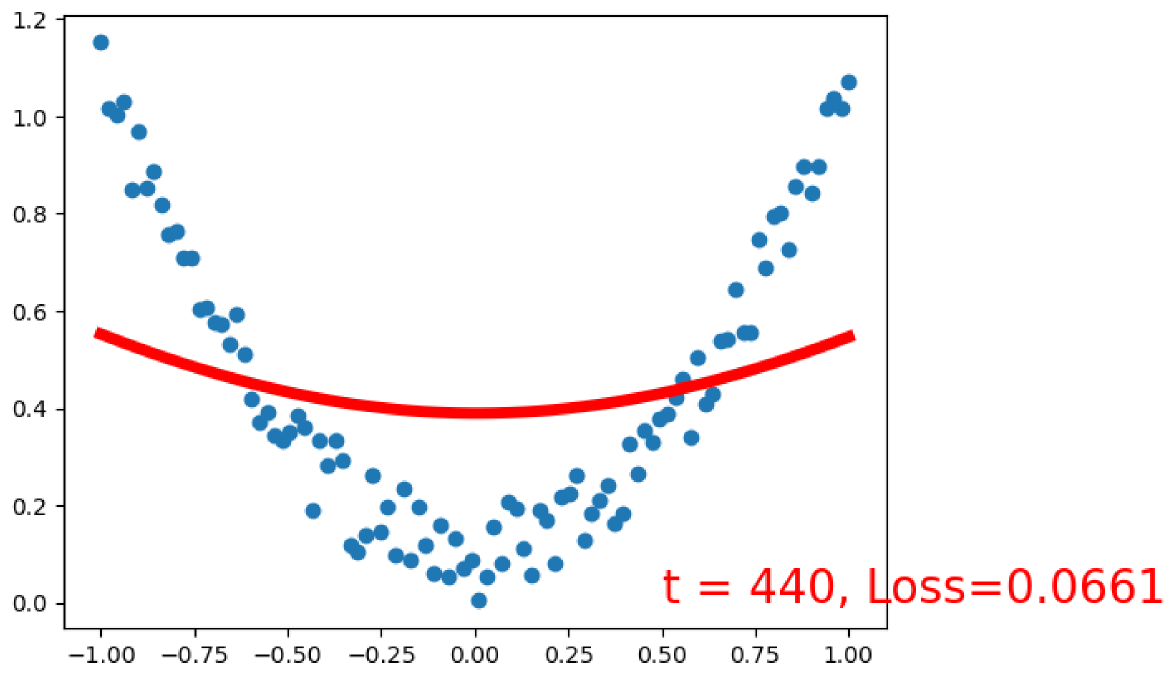

t = 340, Loss=0.0754


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 360, Loss=0.0738

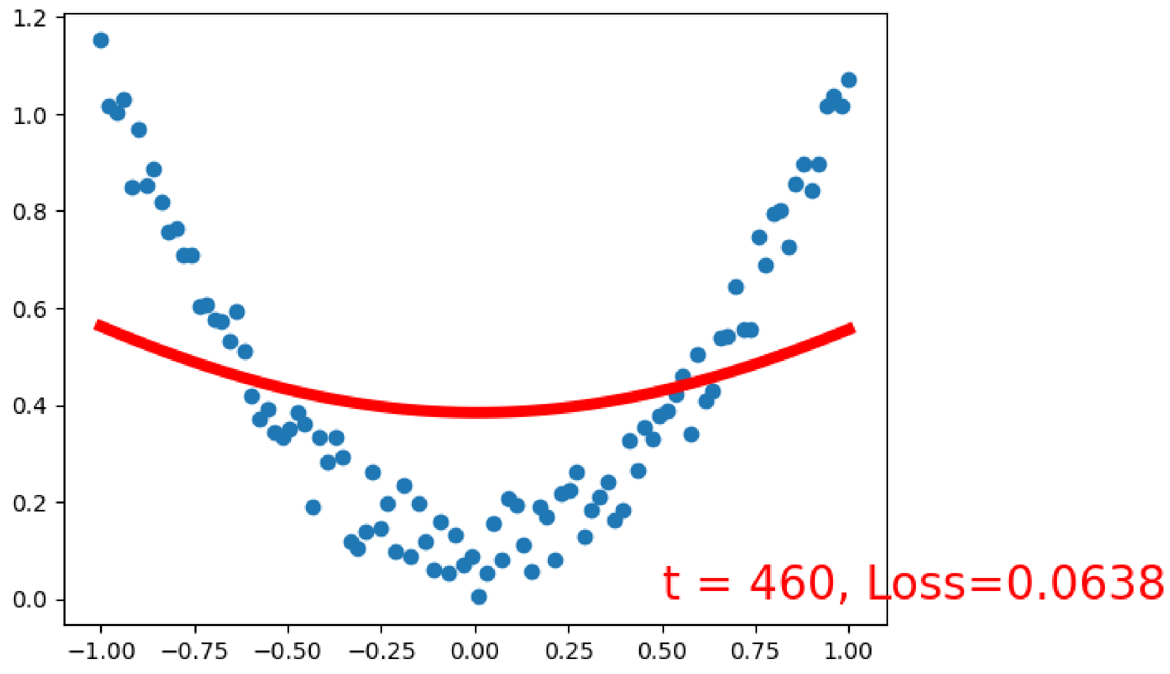

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 380, Loss=0.0721


t = 400, Loss=0.0702
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

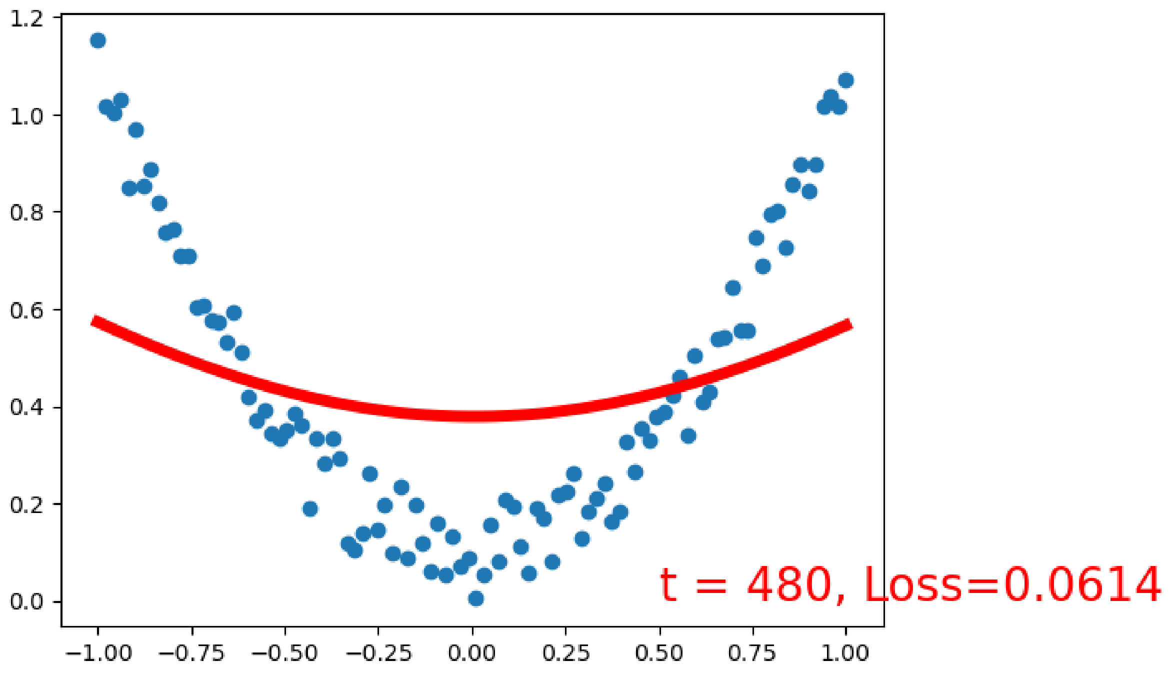

t = 420, Loss=0.0682
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

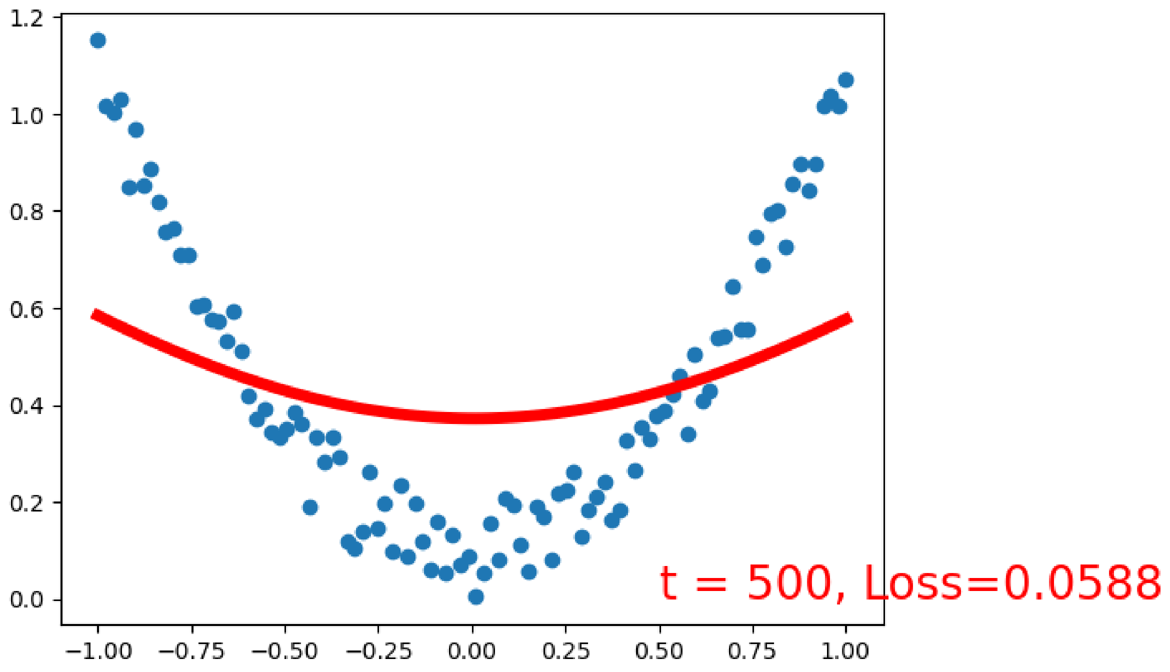


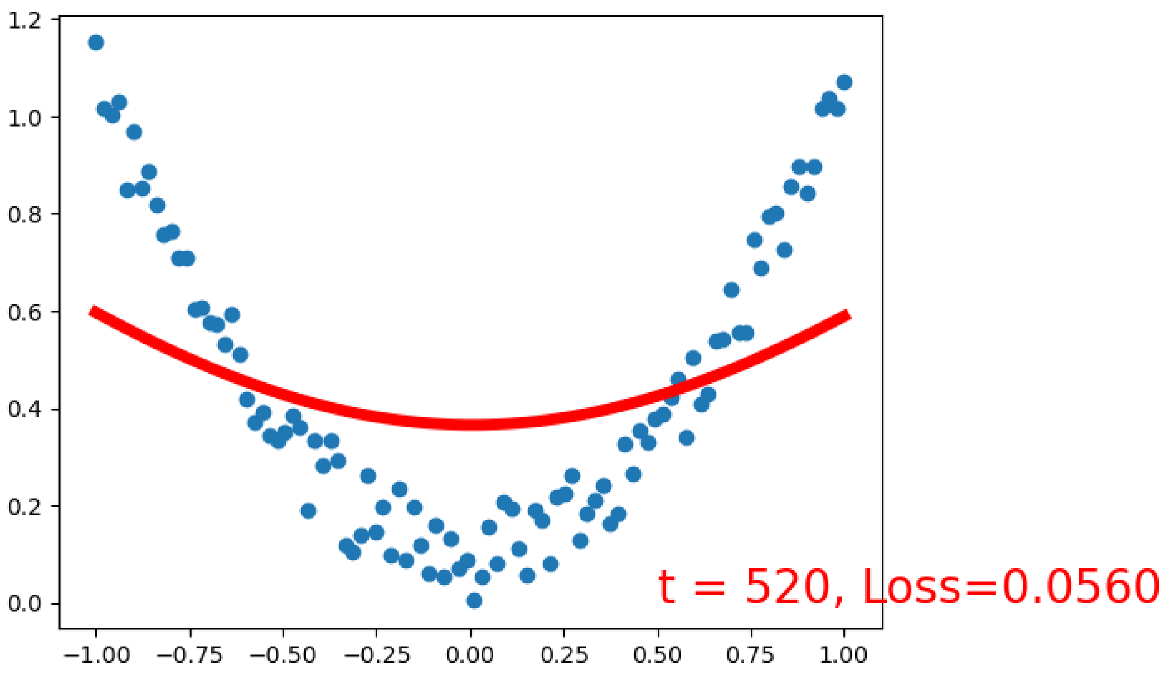
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 440, Loss=0.0661

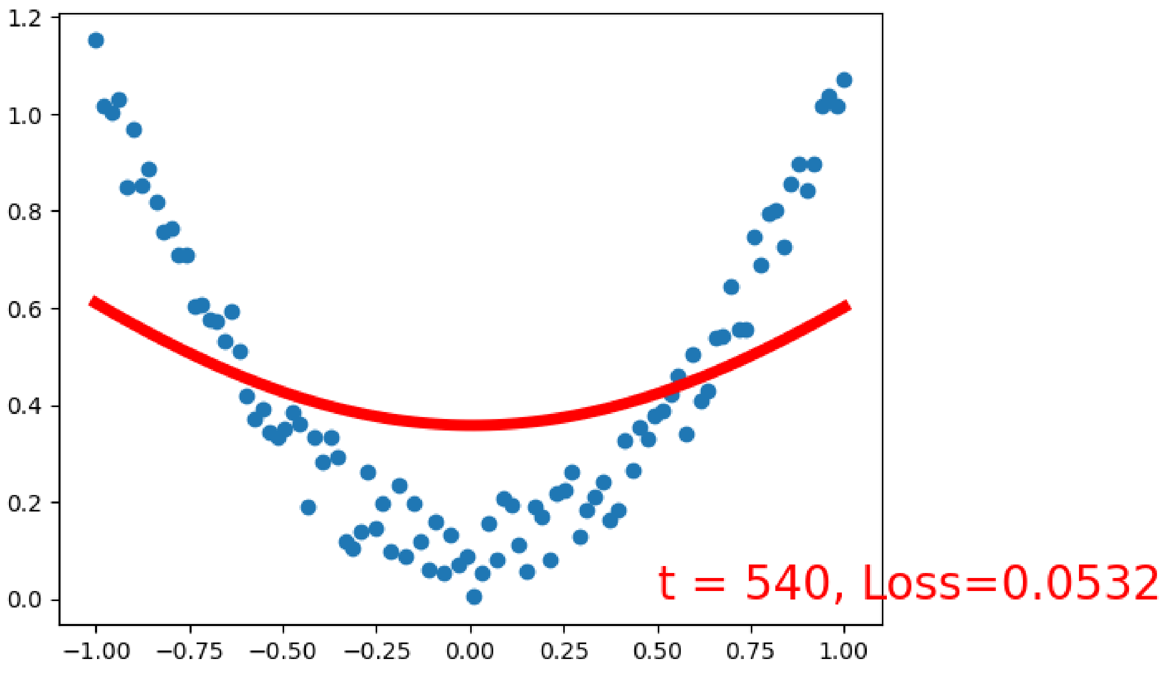


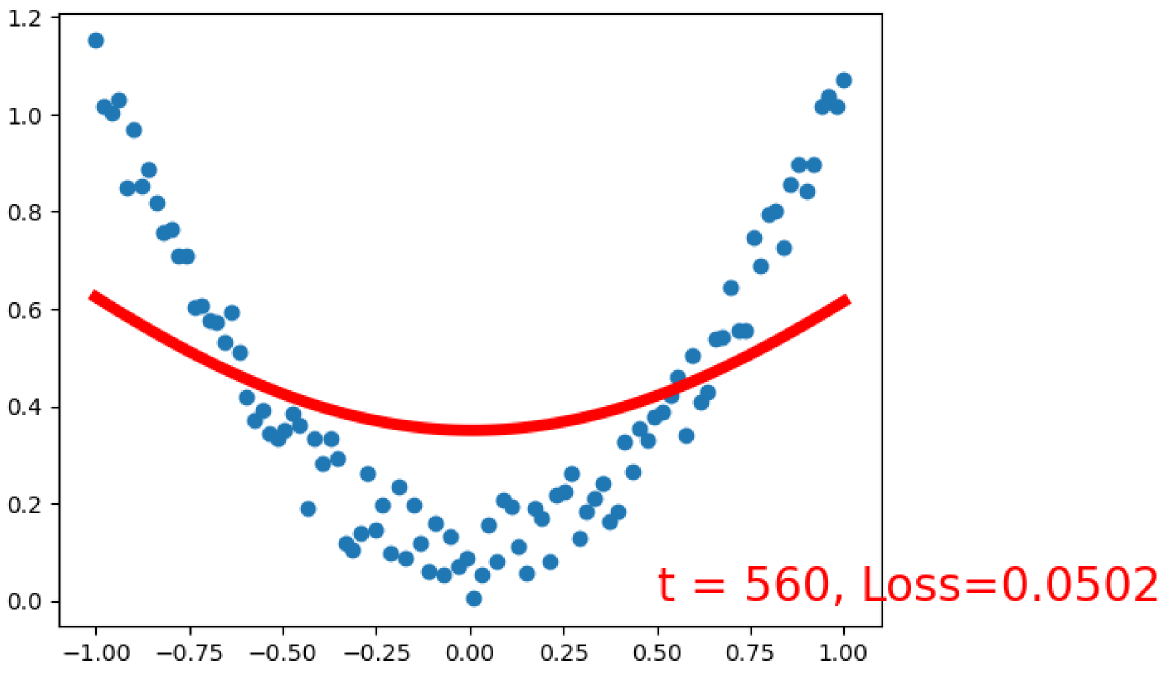
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 460, Loss=0.0638

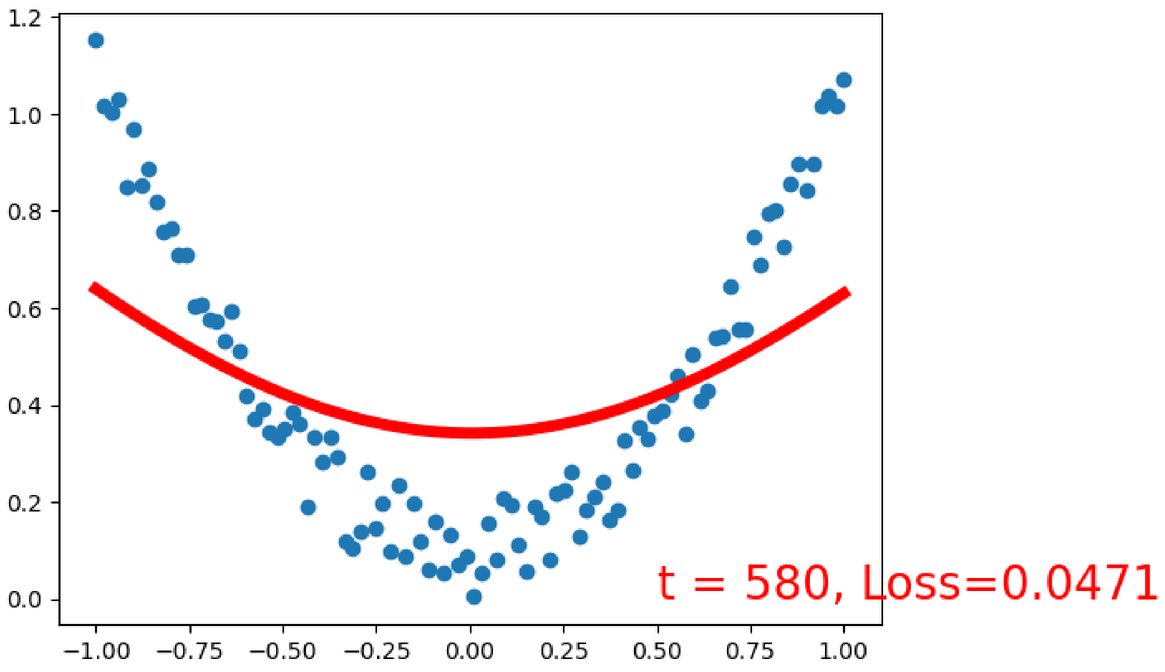

t = 480, Loss=0.0614

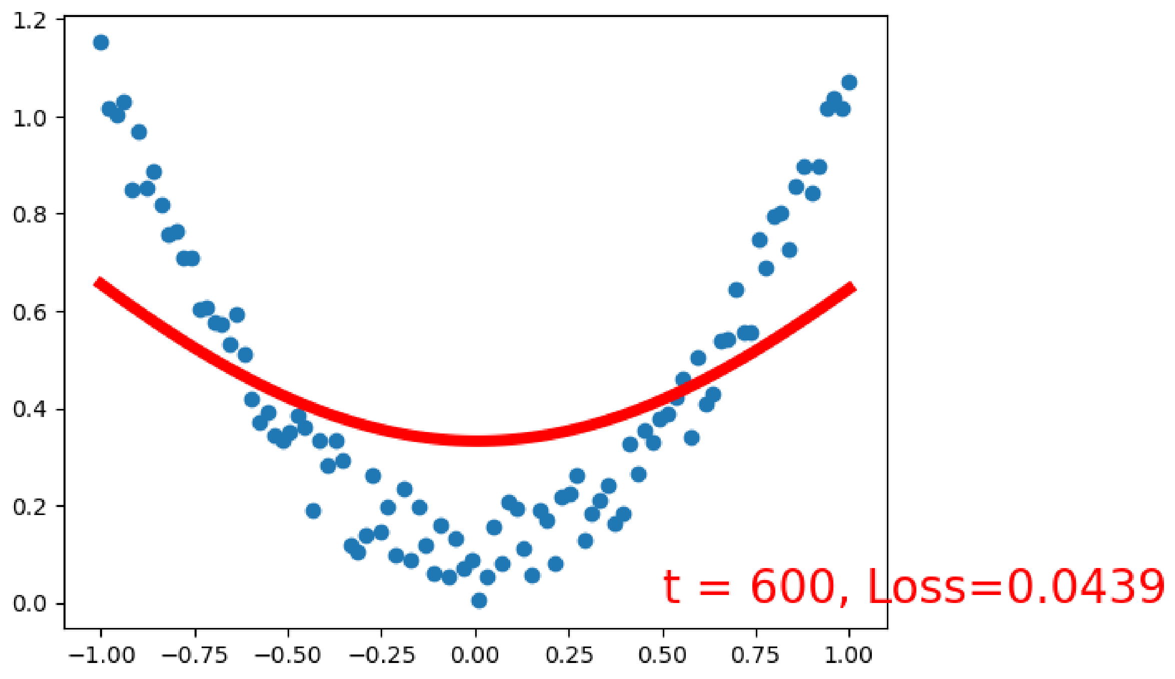

t = 500, Loss=0.0588


t = 520, Loss=0.0560

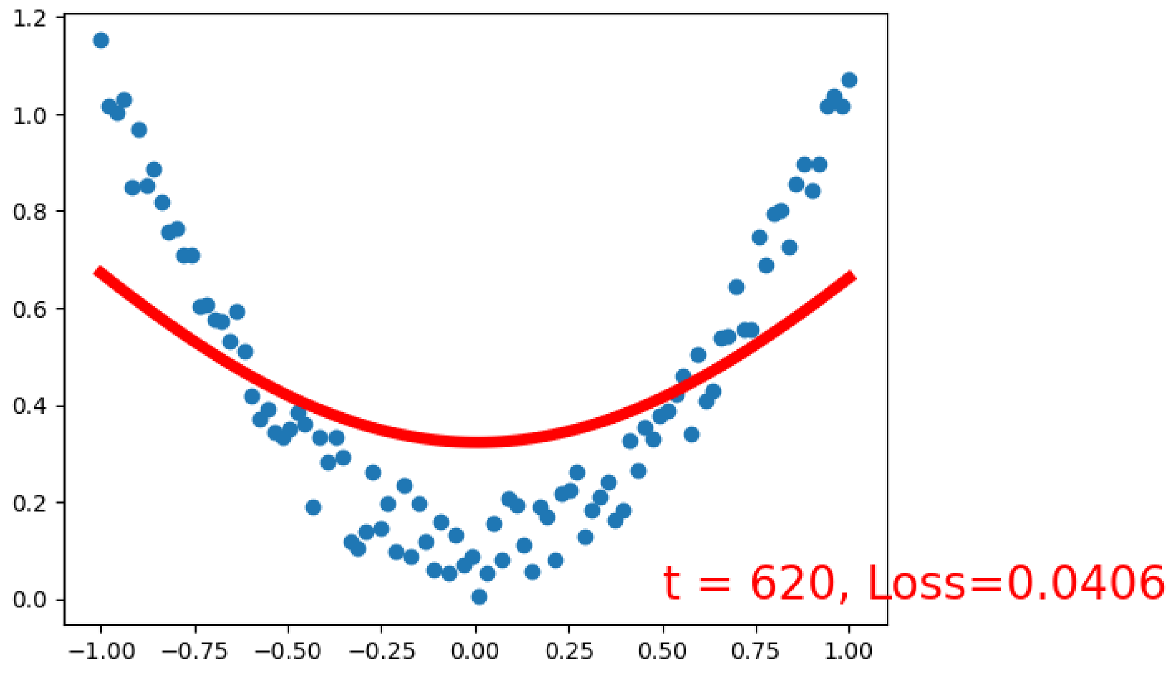

t = 540, Loss=0.0532


t = 560, Loss=0.0502

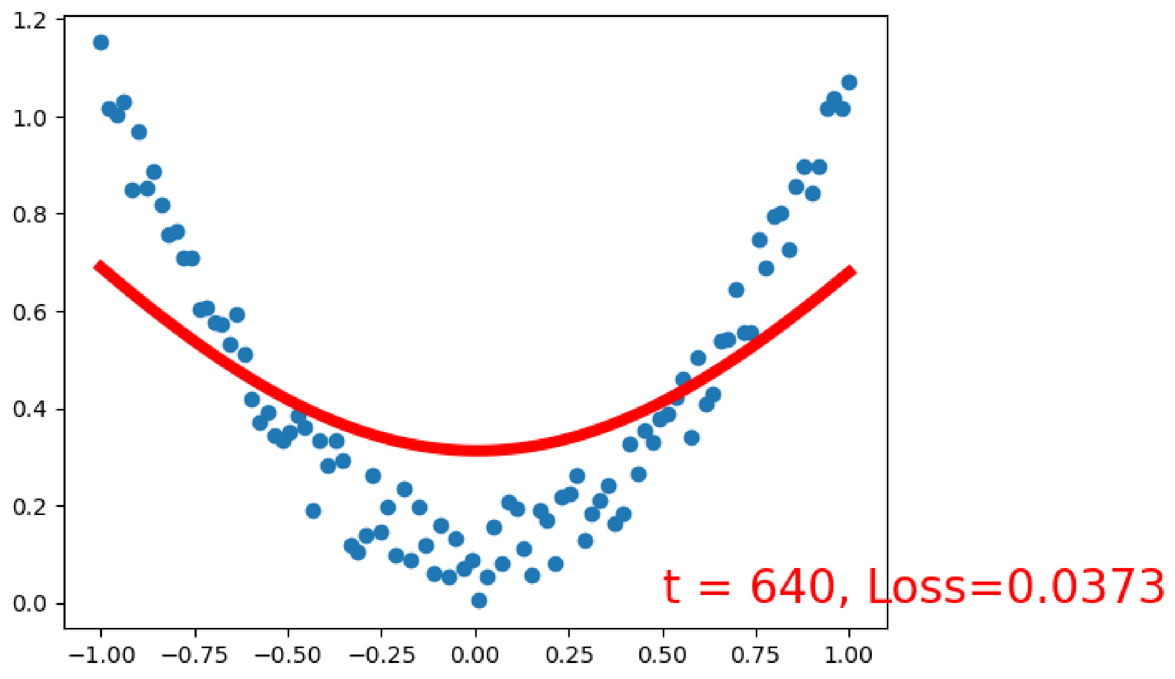

t = 580, Loss=0.0471

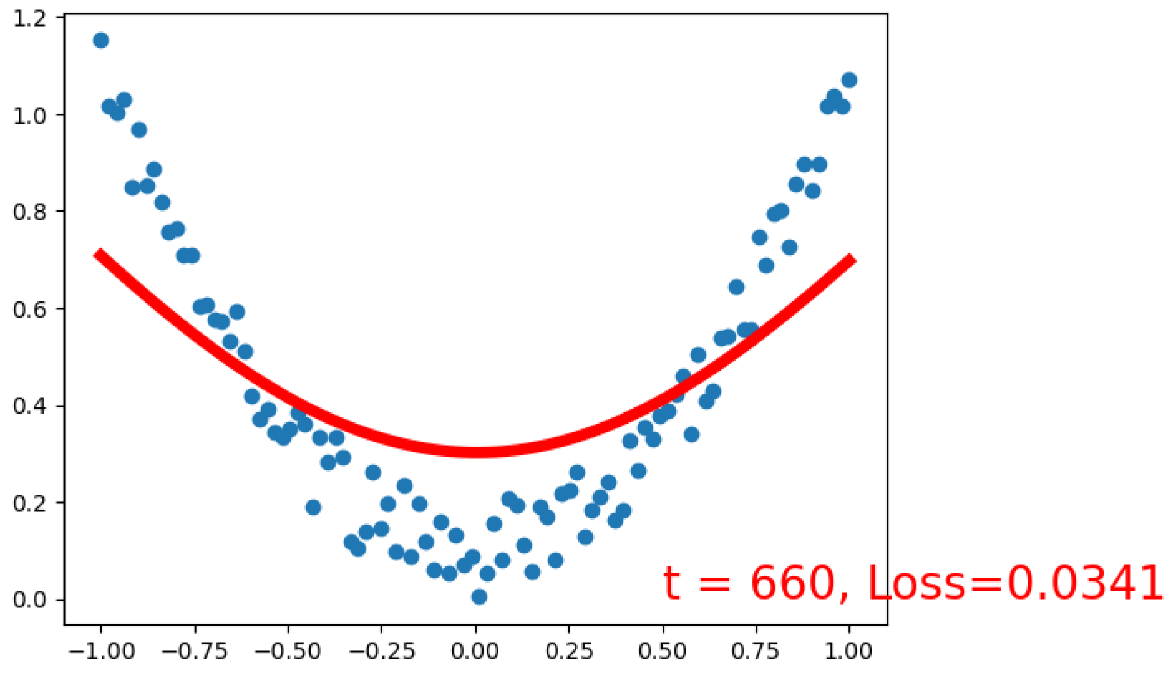


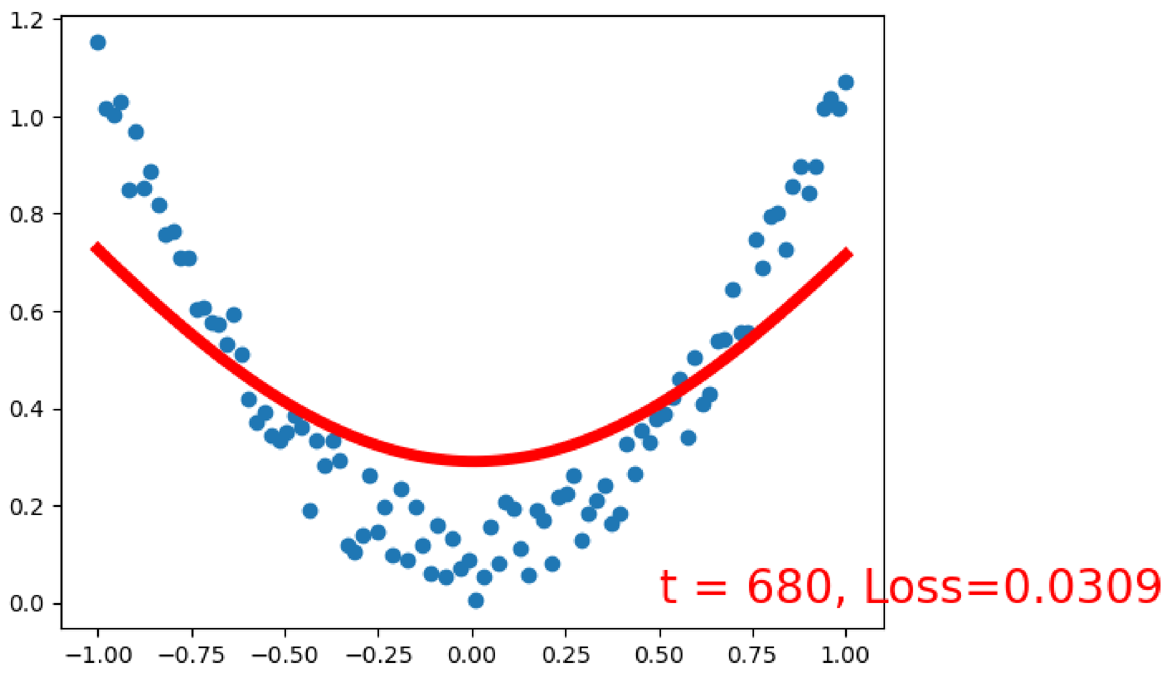
t = 600, Loss=0.0439



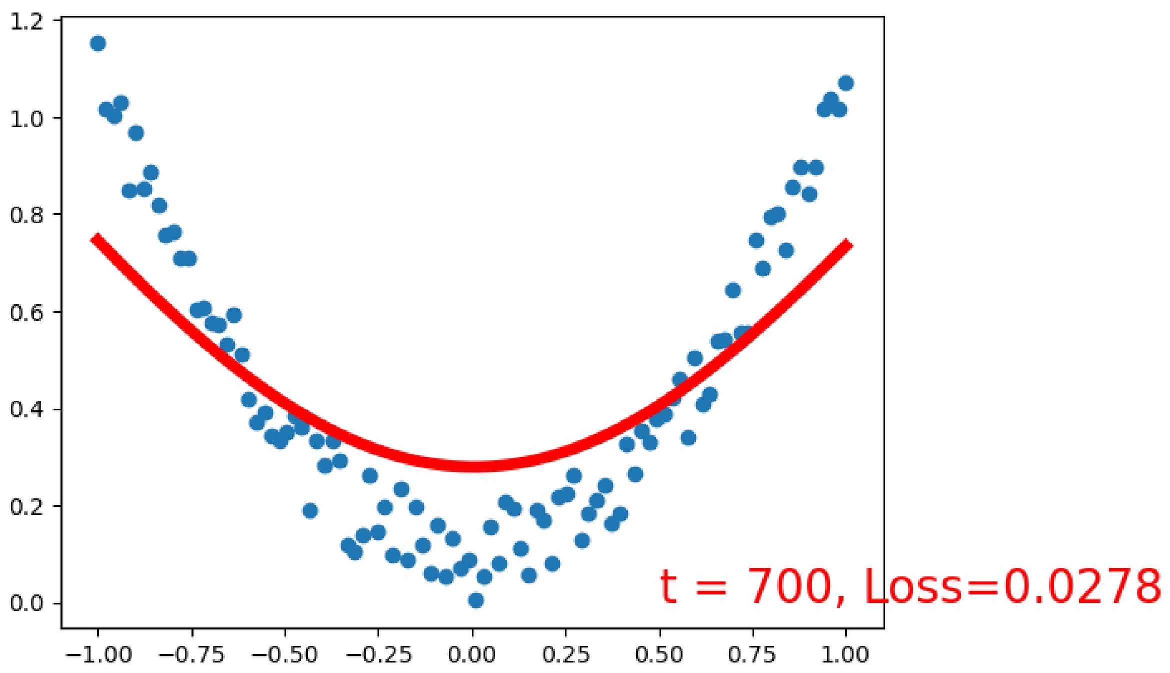
t = 620, Loss=0.0406


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 640, Loss=0.0373

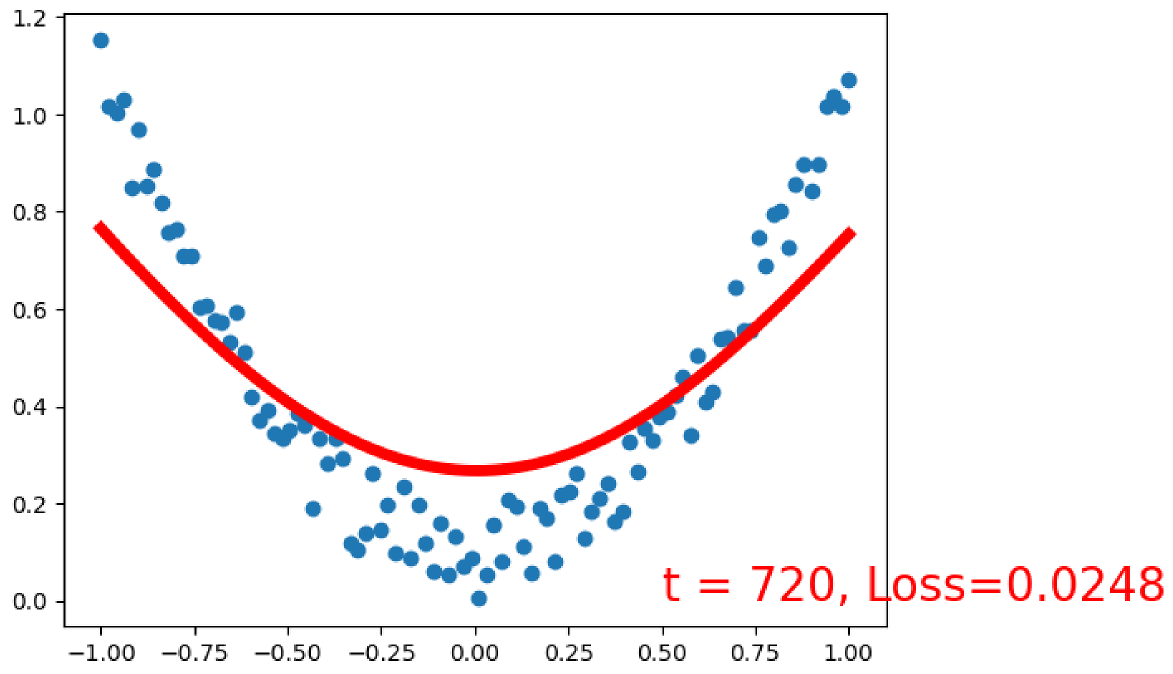

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 660, Loss=0.0341

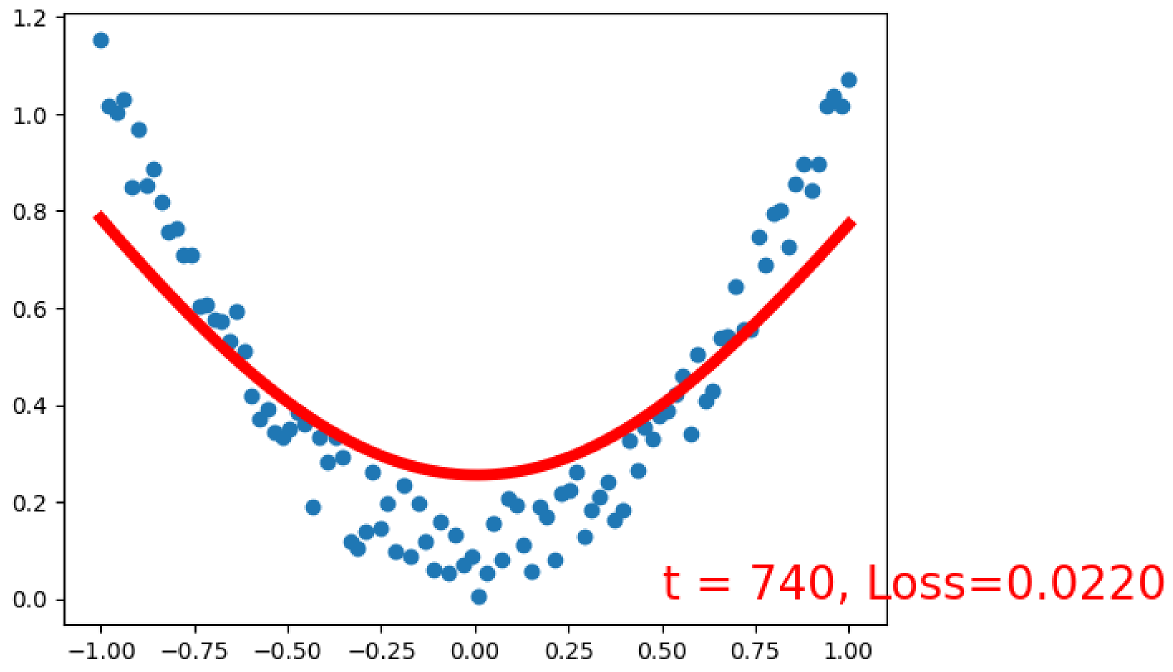


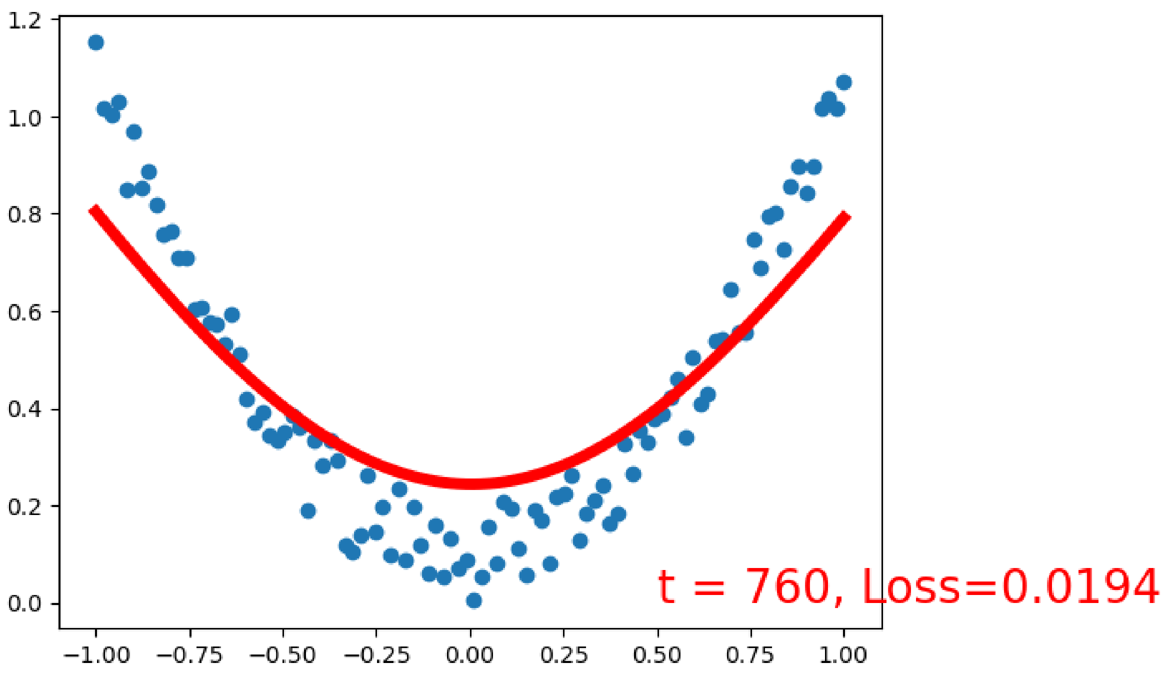
t = 680, Loss=0.0309

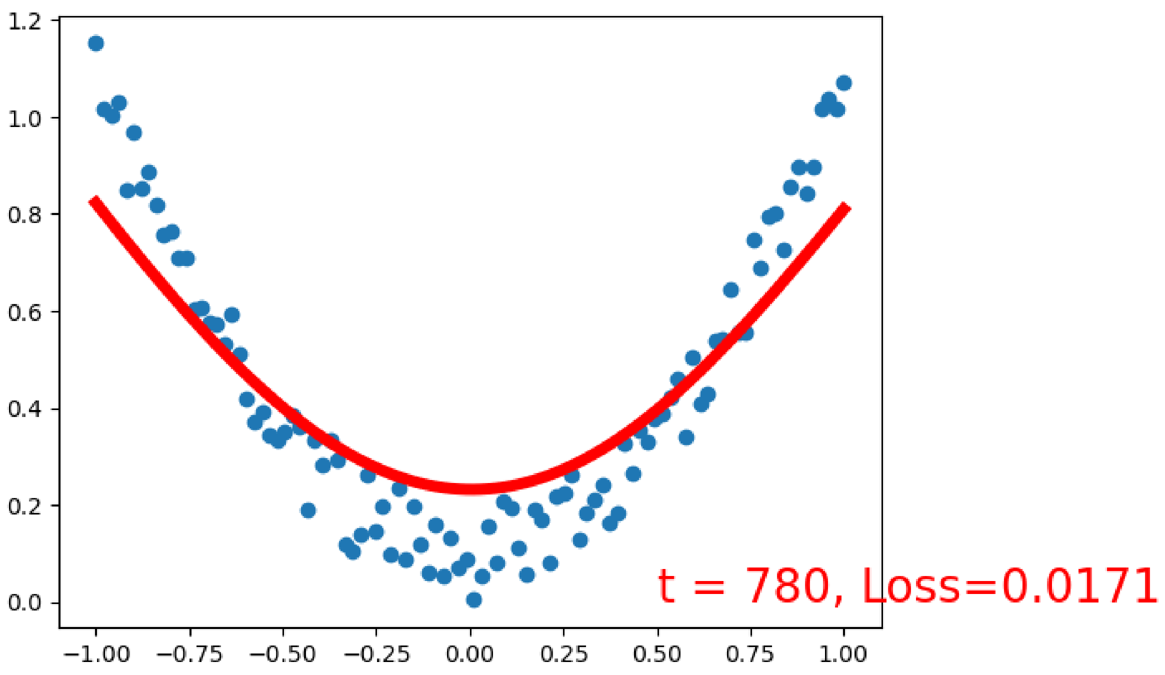


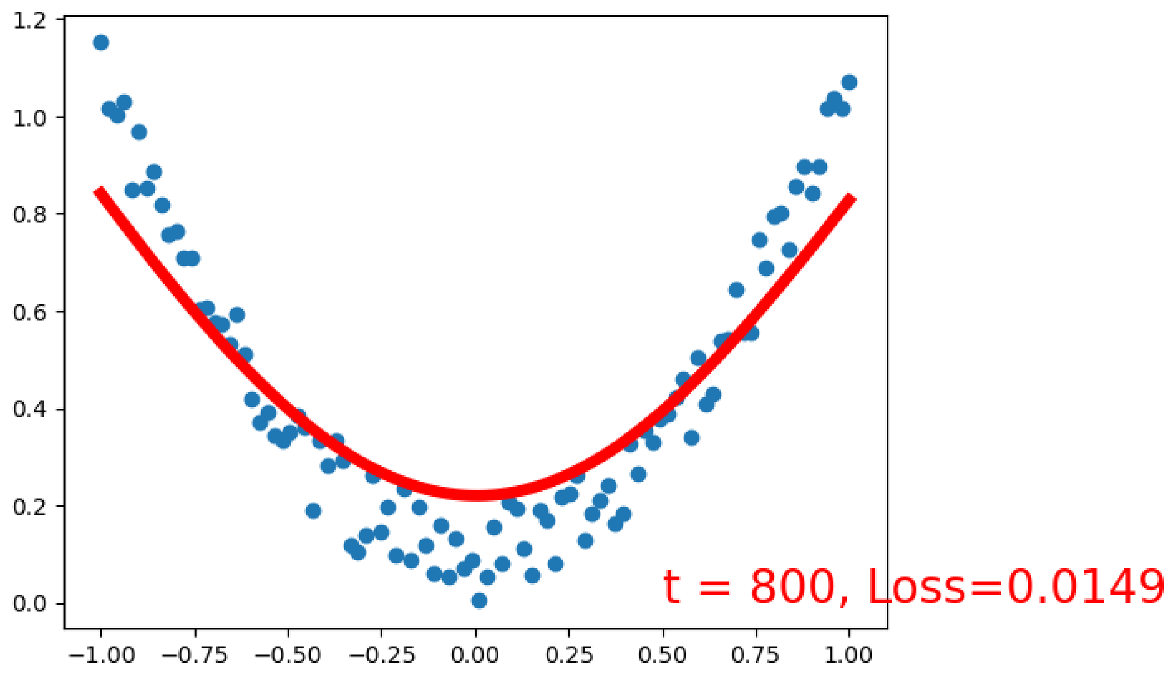
t = 700, Loss=0.0278


t = 720, Loss=0.0248

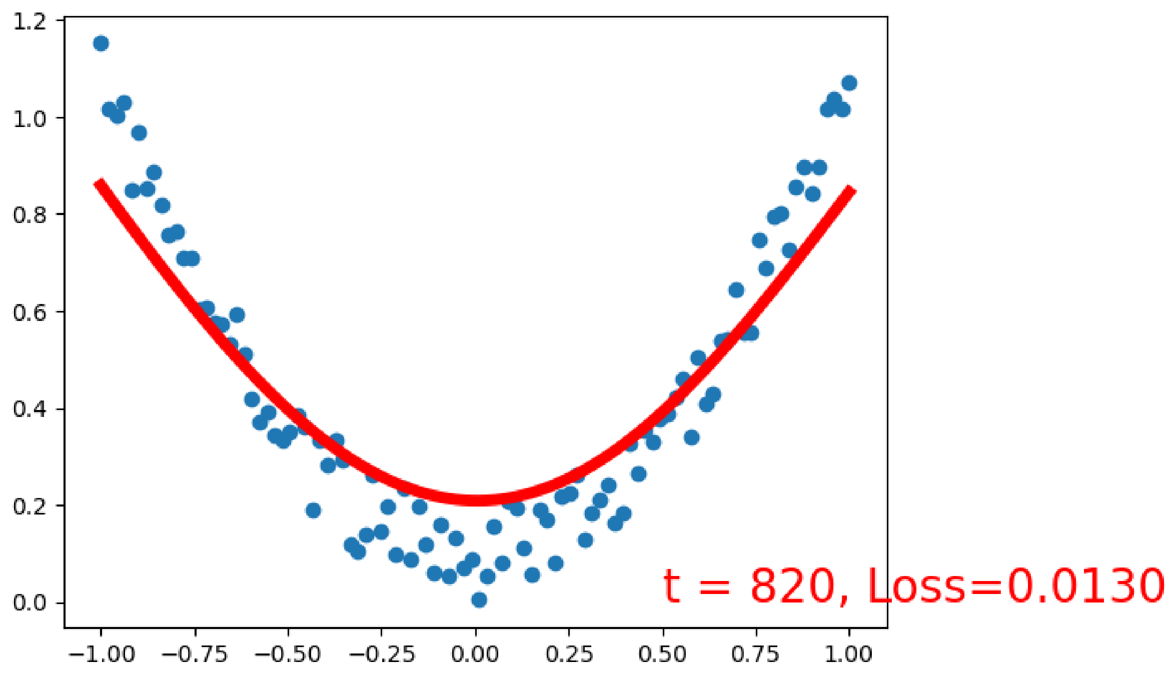

t = 740, Loss=0.0220


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 760, Loss=0.0194

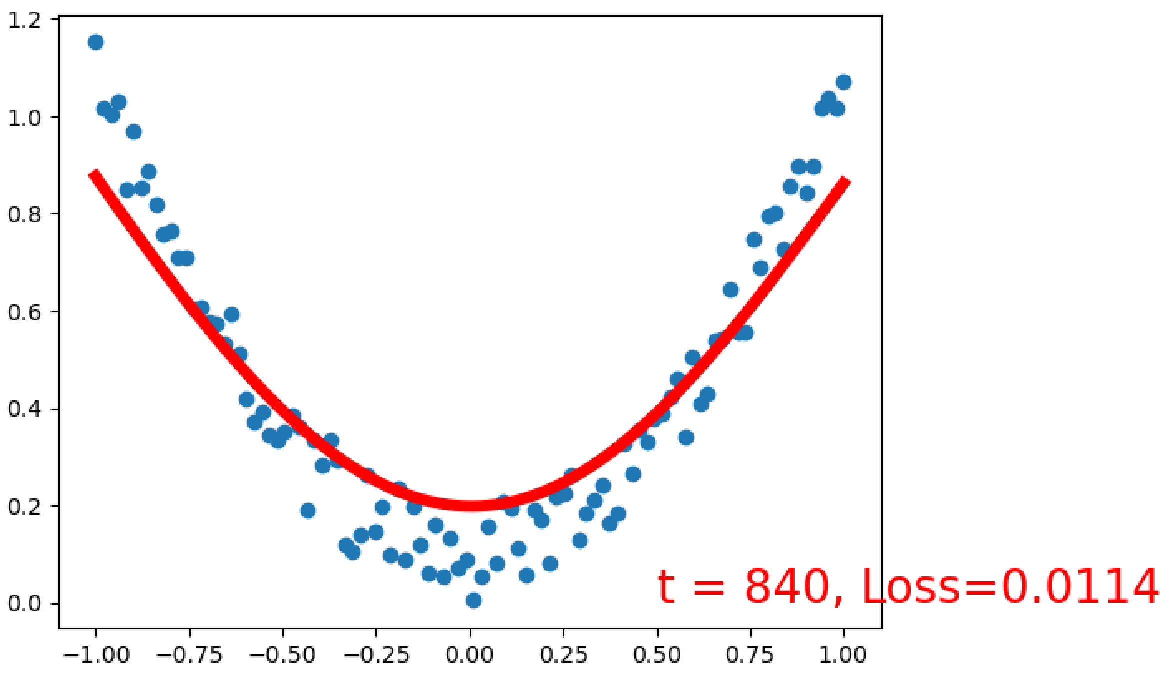

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 780, Loss=0.0171

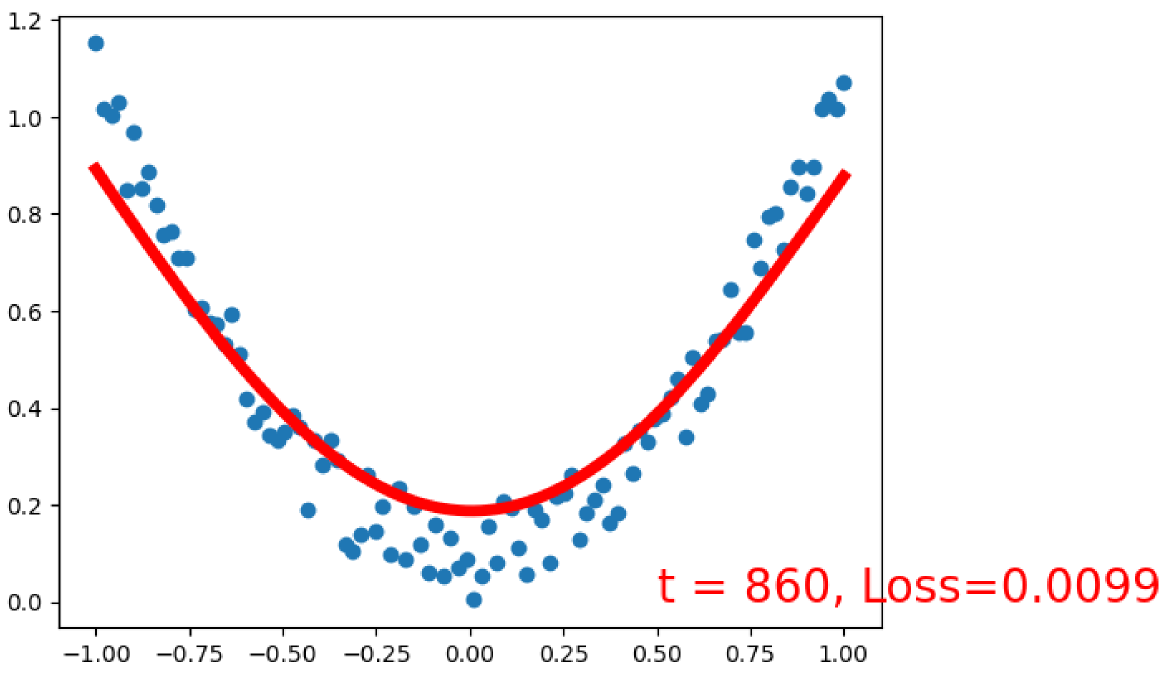


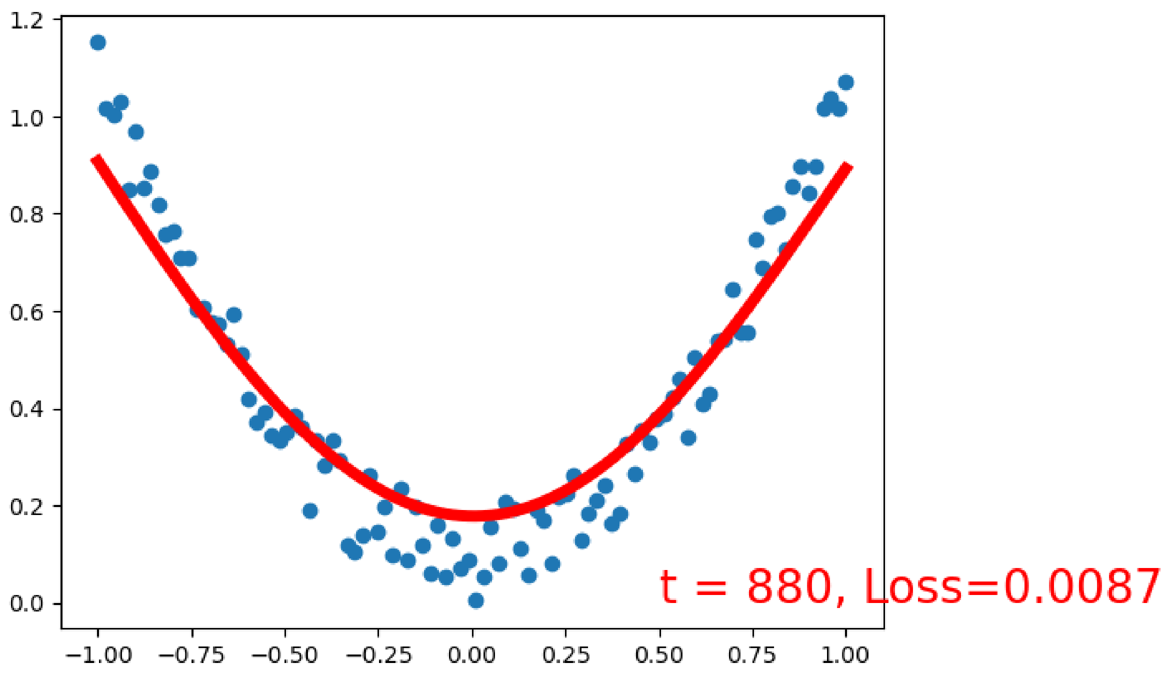
t = 800, Loss=0.0149



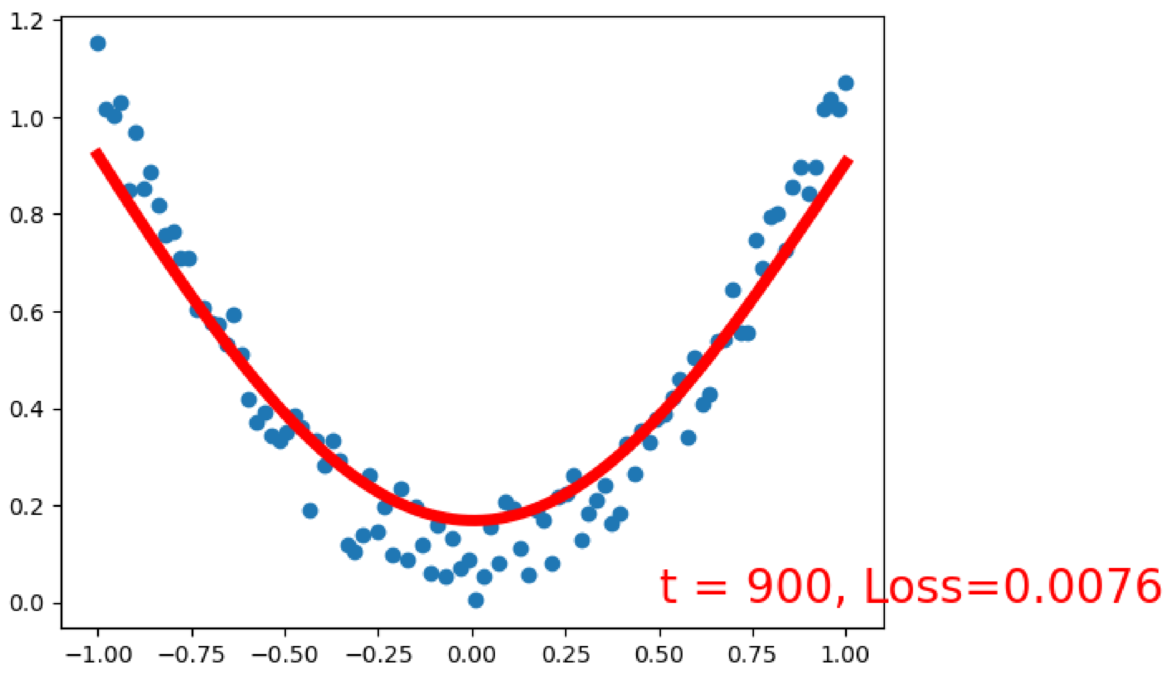
t = 820, Loss=0.0130


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 840, Loss=0.0114

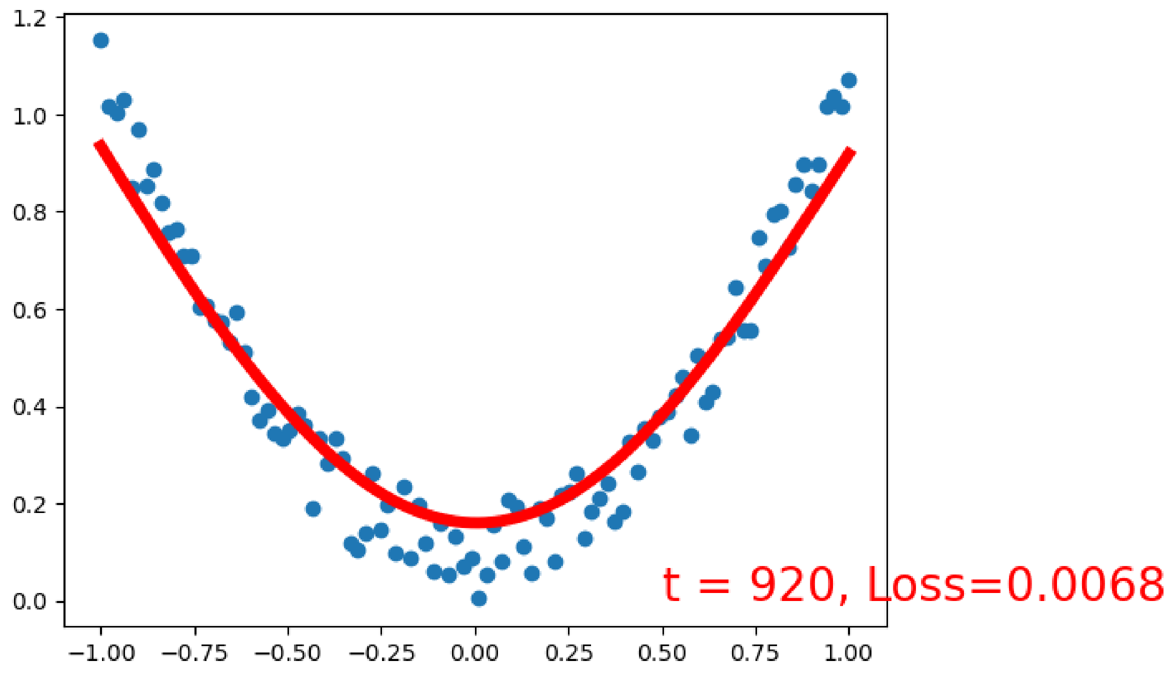

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 860, Loss=0.0099

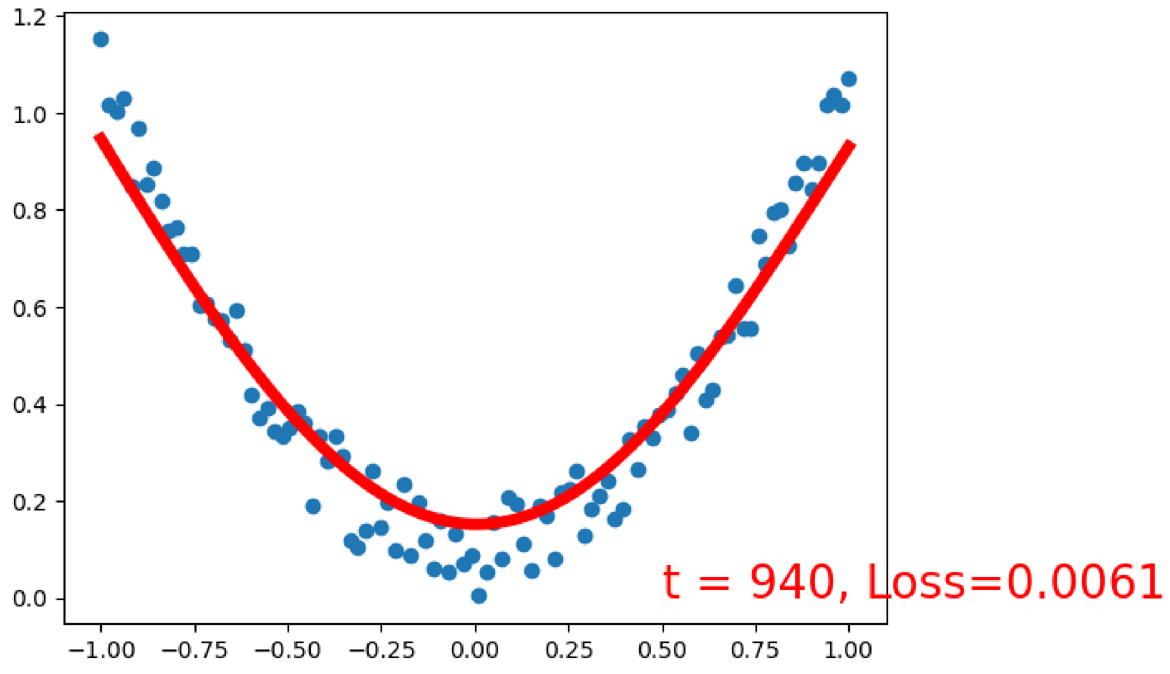


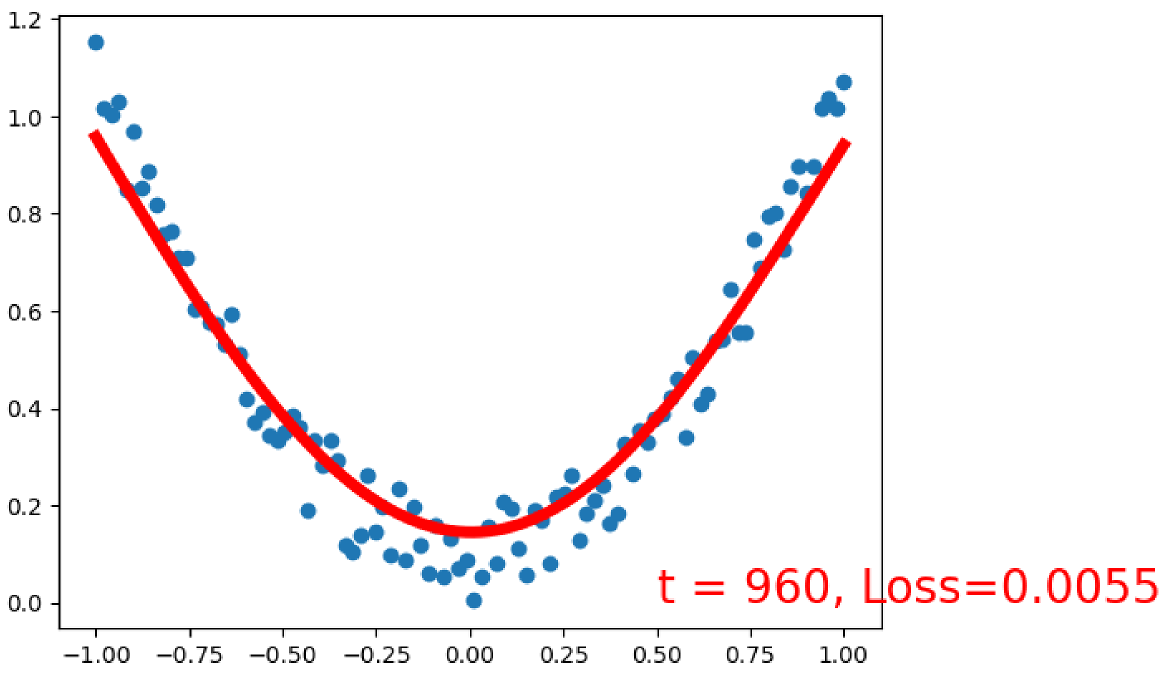
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 880, Loss=0.0087

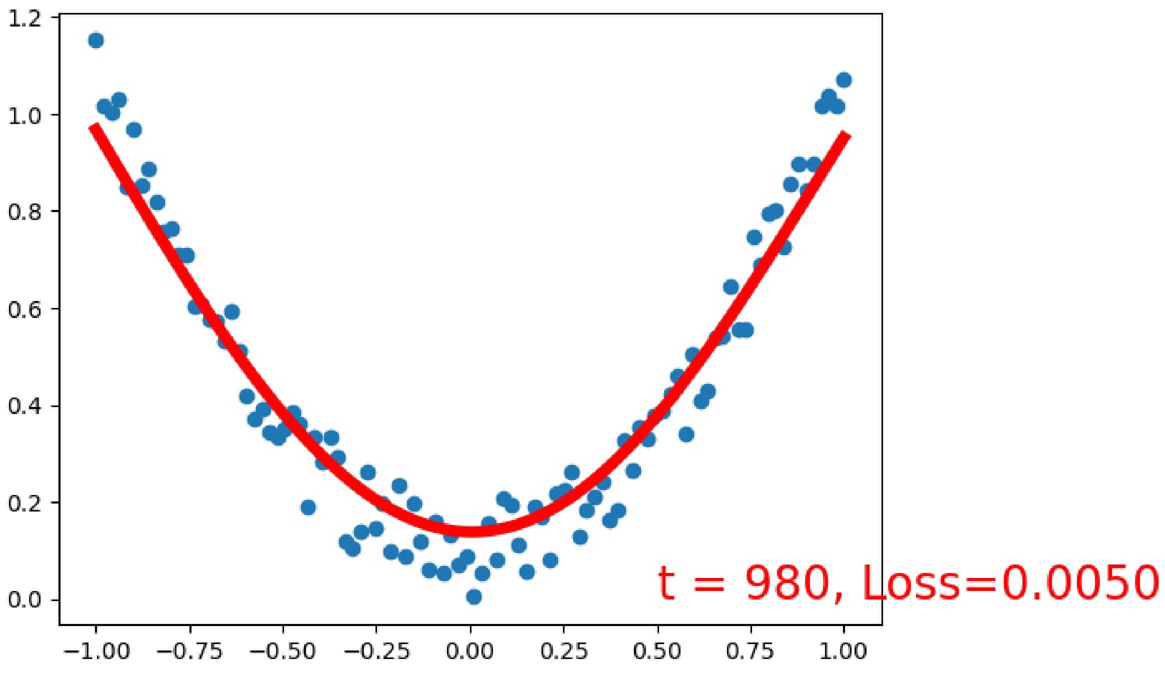


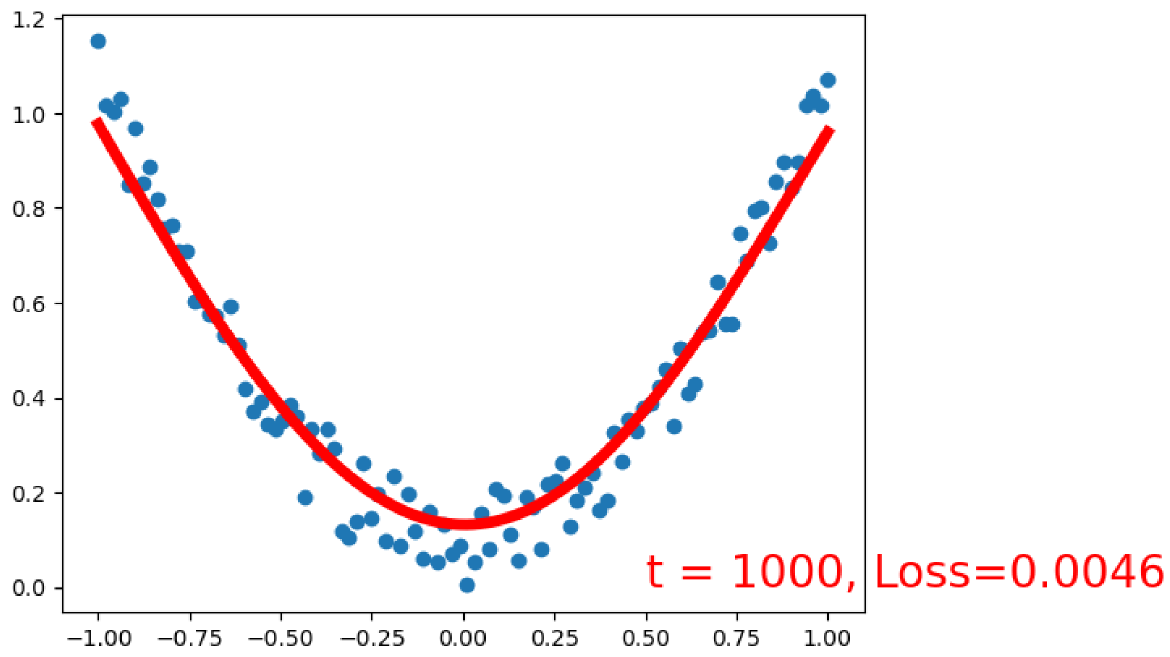
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 900, Loss=0.0076

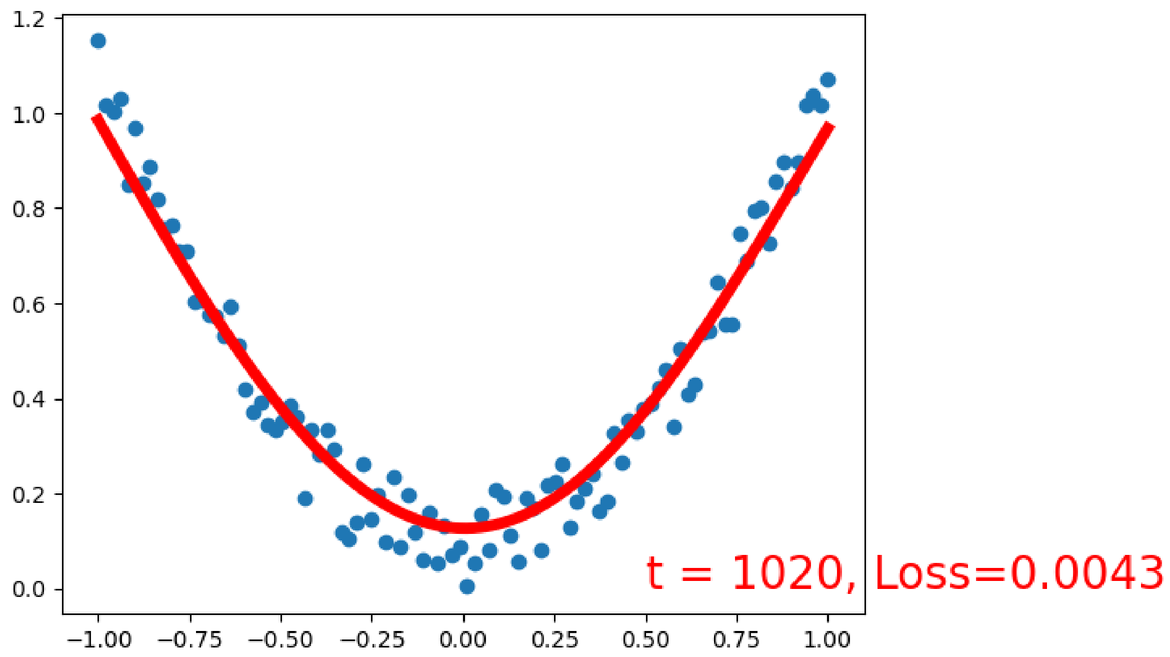

t = 920, Loss=0.0068

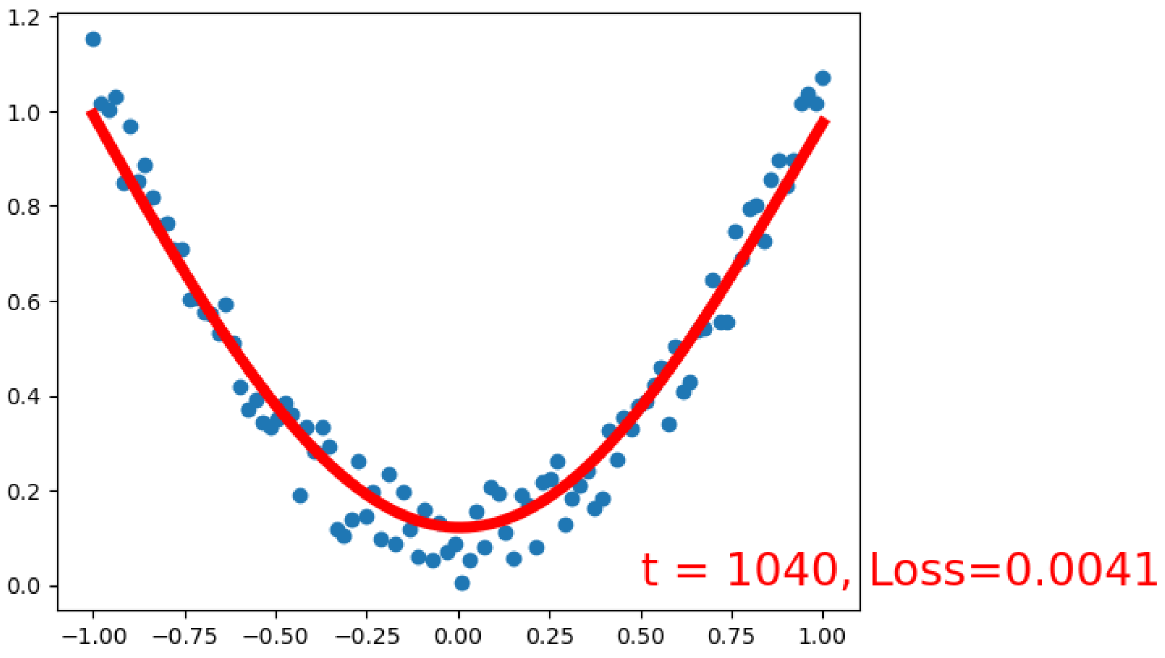

t = 940, Loss=0.0061

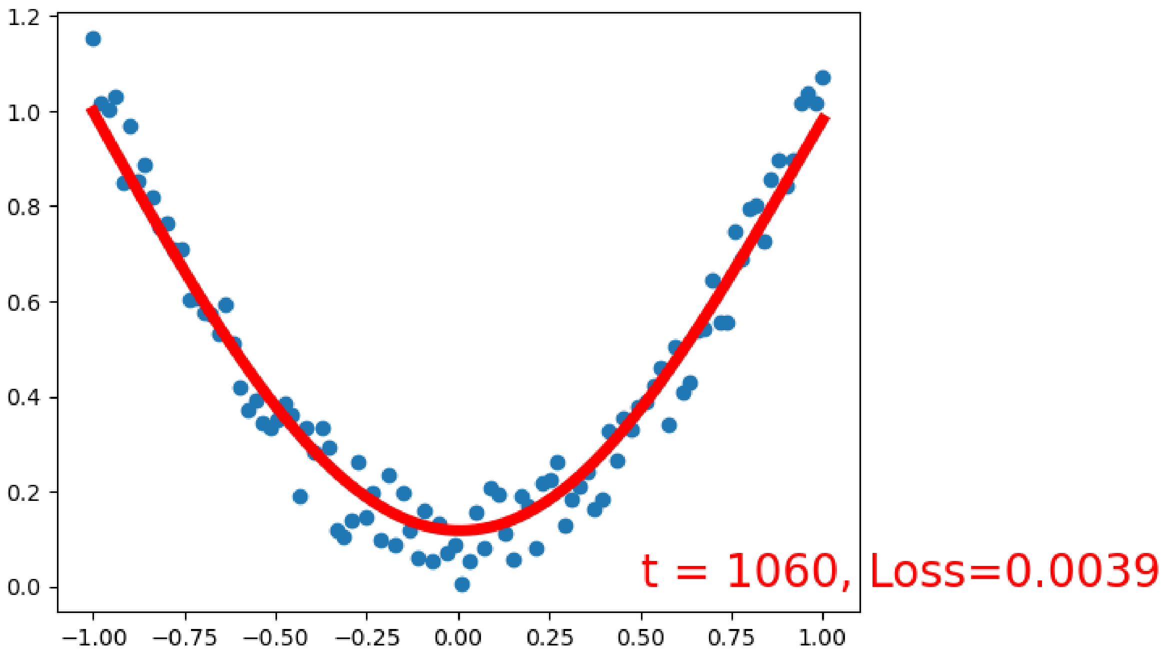

t = 960, Loss=0.0055
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

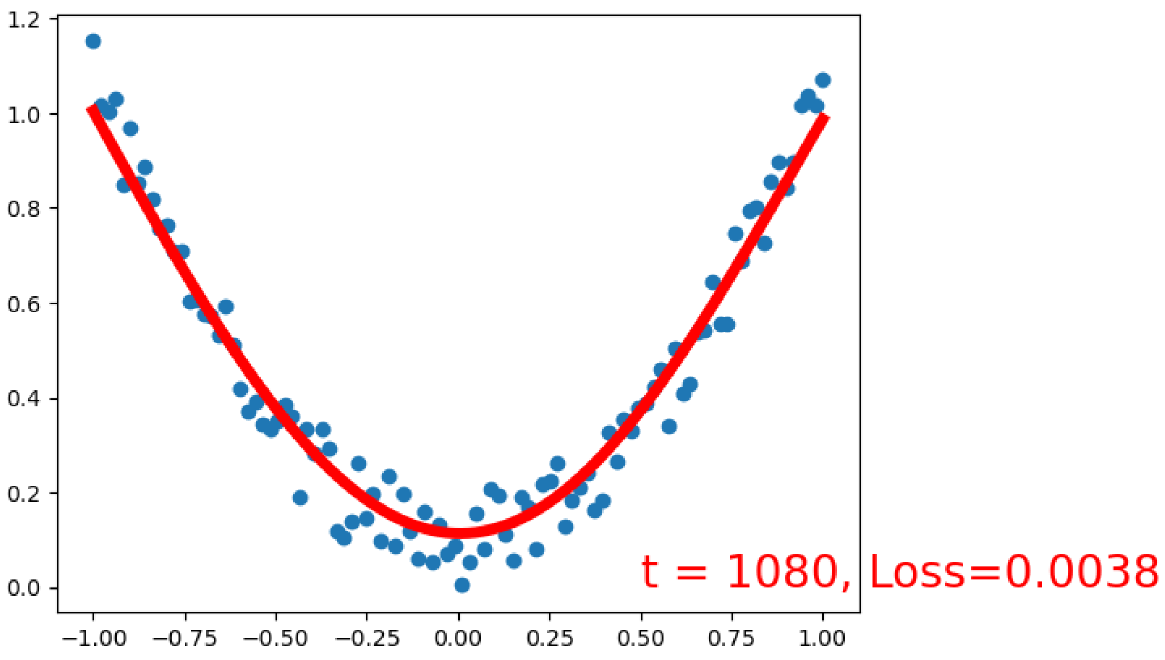

t = 980, Loss=0.0050
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1000, Loss=0.0046

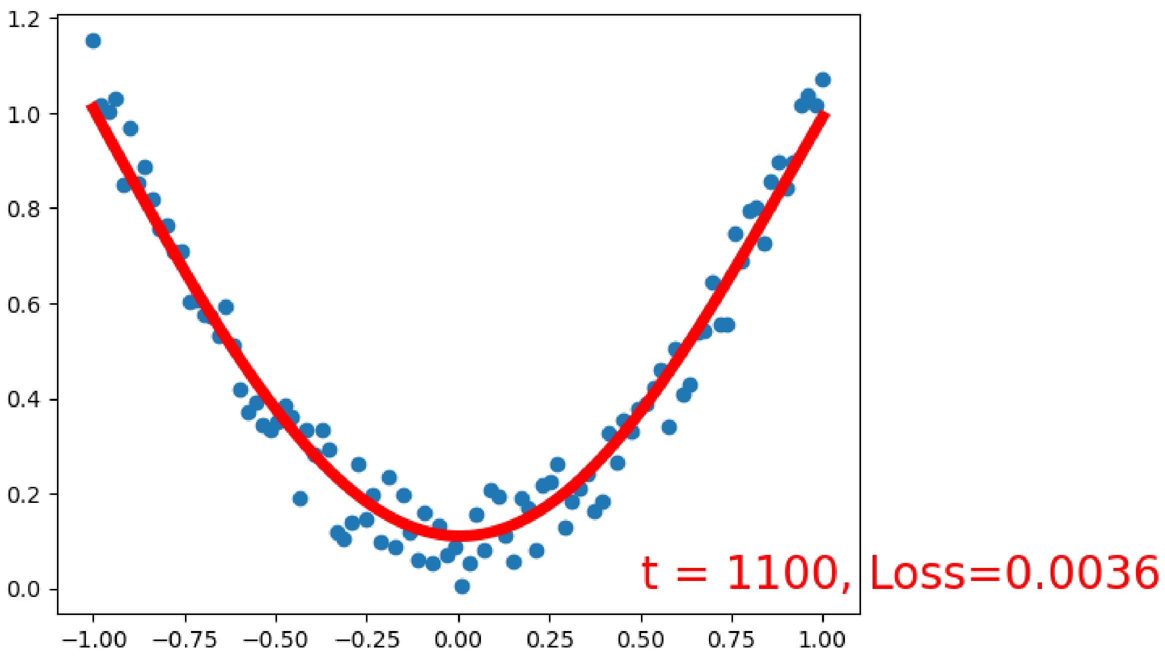

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1020, Loss=0.0043


t = 1040, Loss=0.0041
-1.00
-0.75
-0.50
-0.25
0.00
0.25
0.50
0.75
1.00
0.0
0.2
0.4
0.6
0.8
1.0
1.2

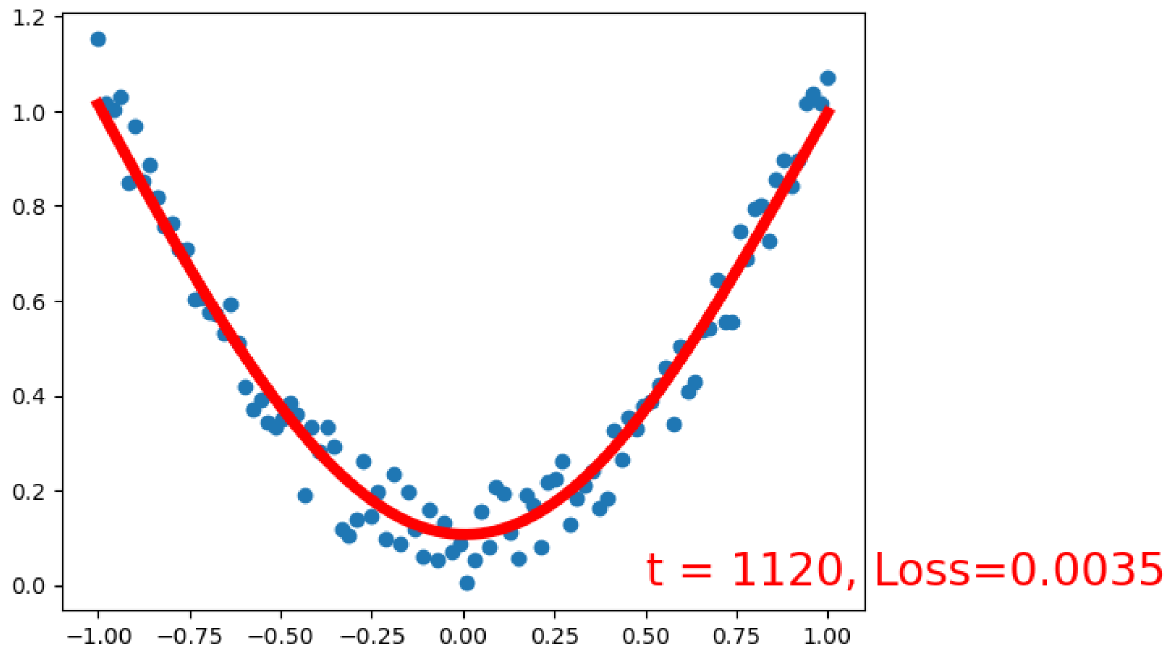

t = 1060, Loss=0.0039
-1.00
-0.75
-0.50
-0.25
0.00
0.25
0.50
0.75
1.00
0.0
0.2
0.4
0.6
0.8
1.0
1.2

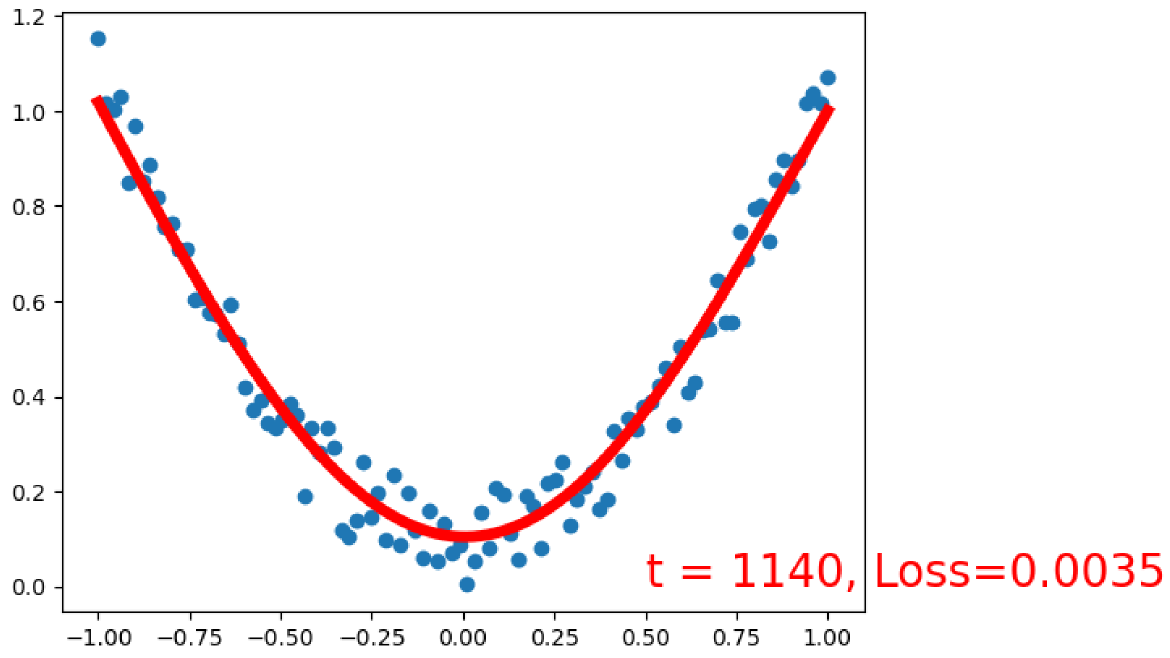


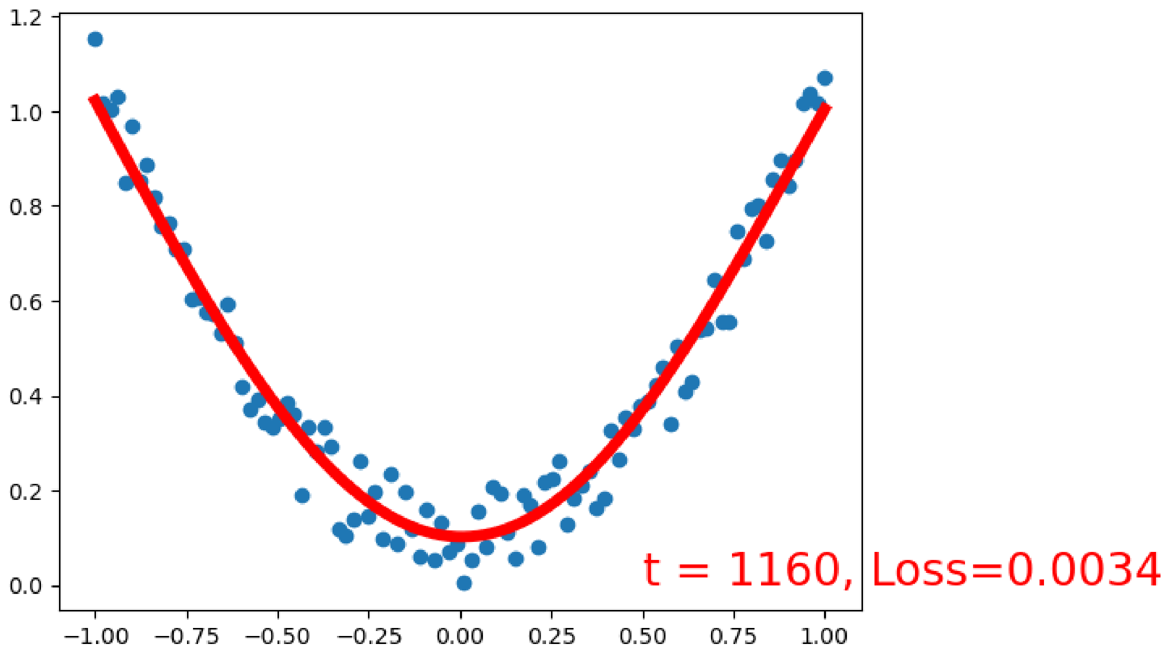
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1080, Loss=0.0038



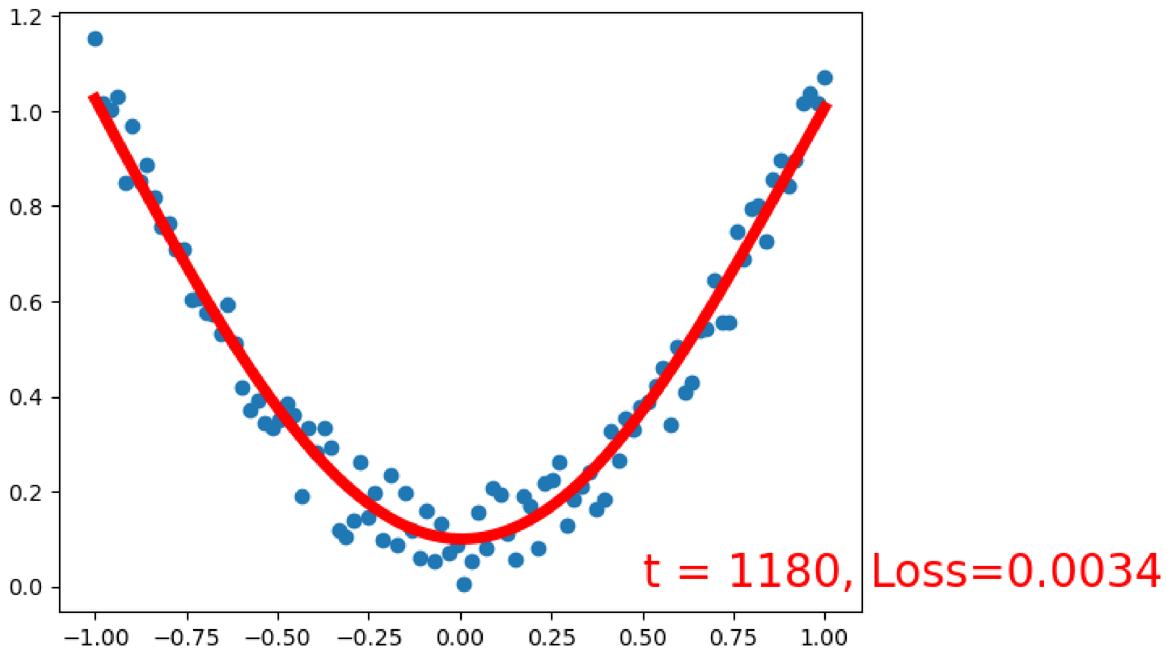
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1100, Loss=0.0036


t = 1120, Loss=0.0035

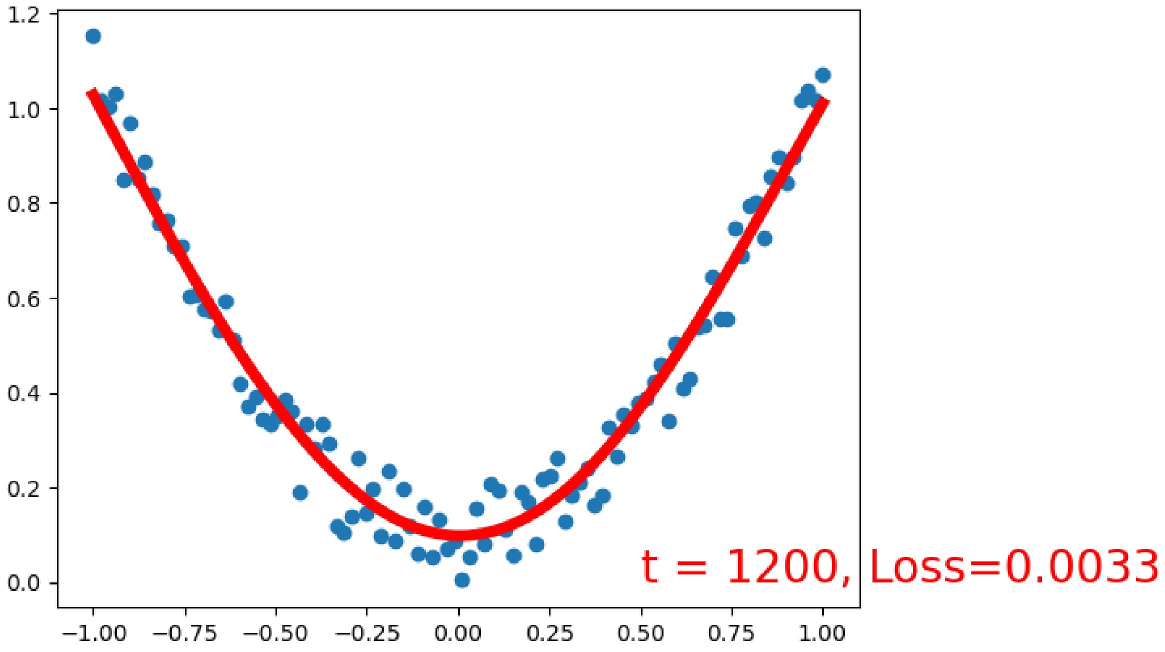

t = 1140, Loss=0.0035

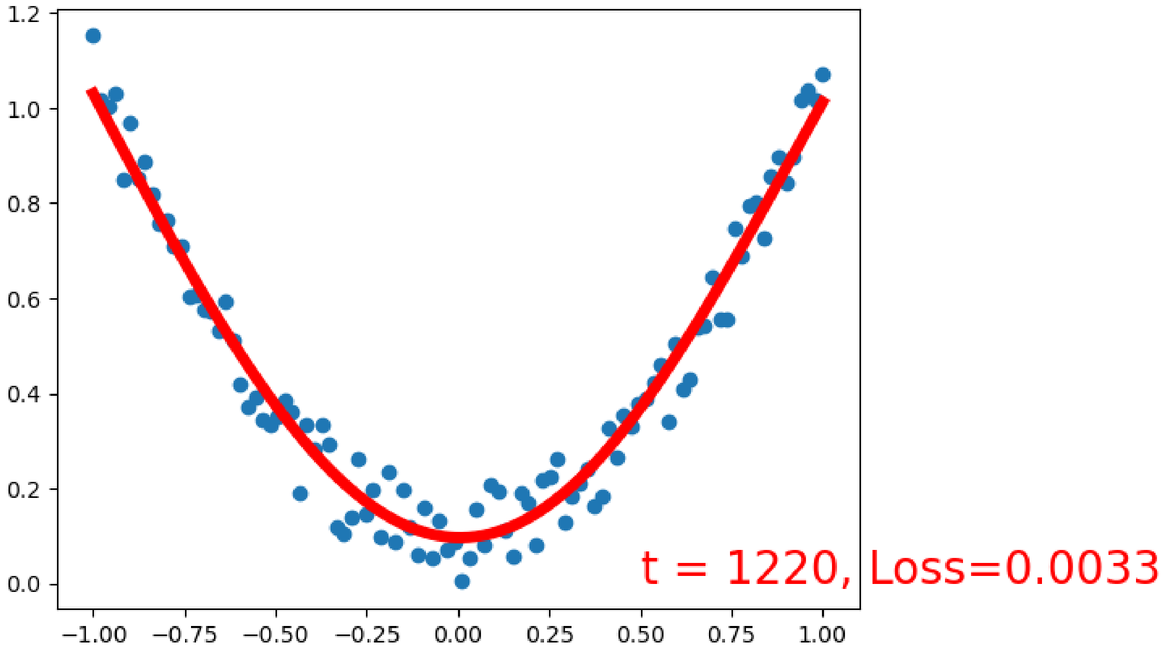


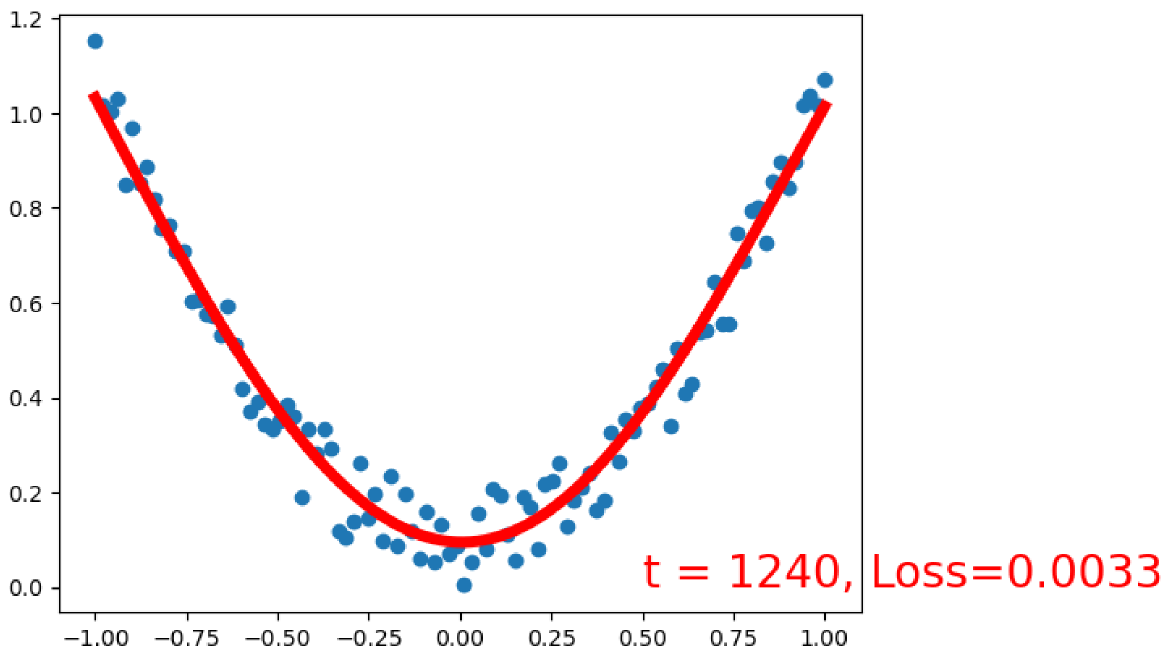
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1160, Loss=0.0034

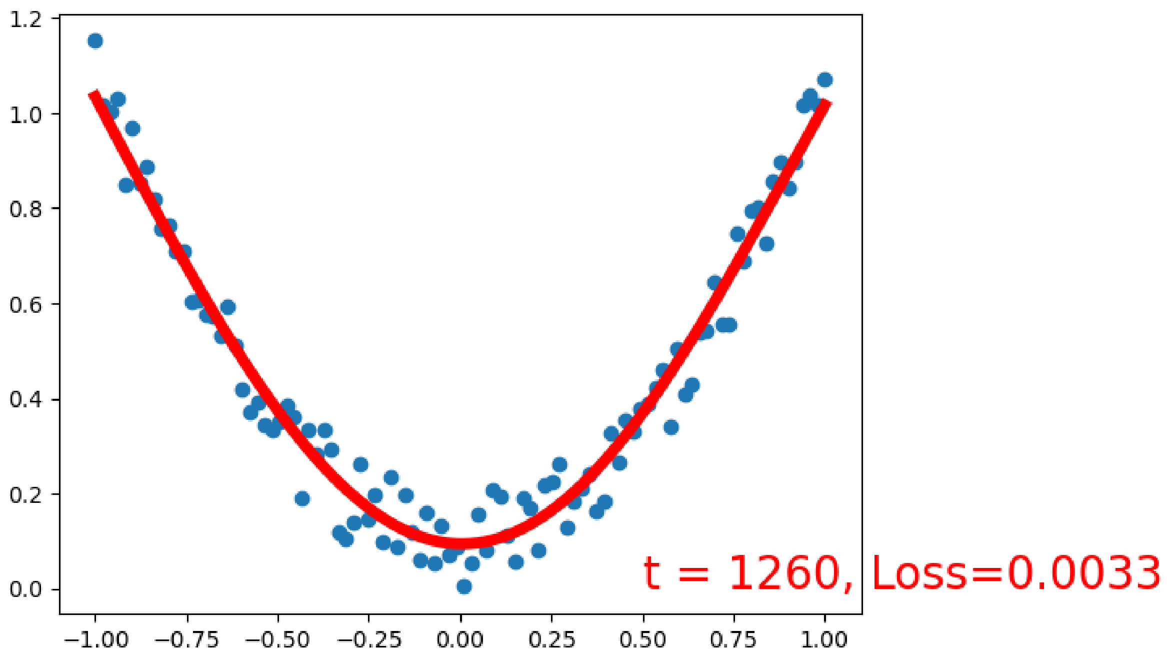


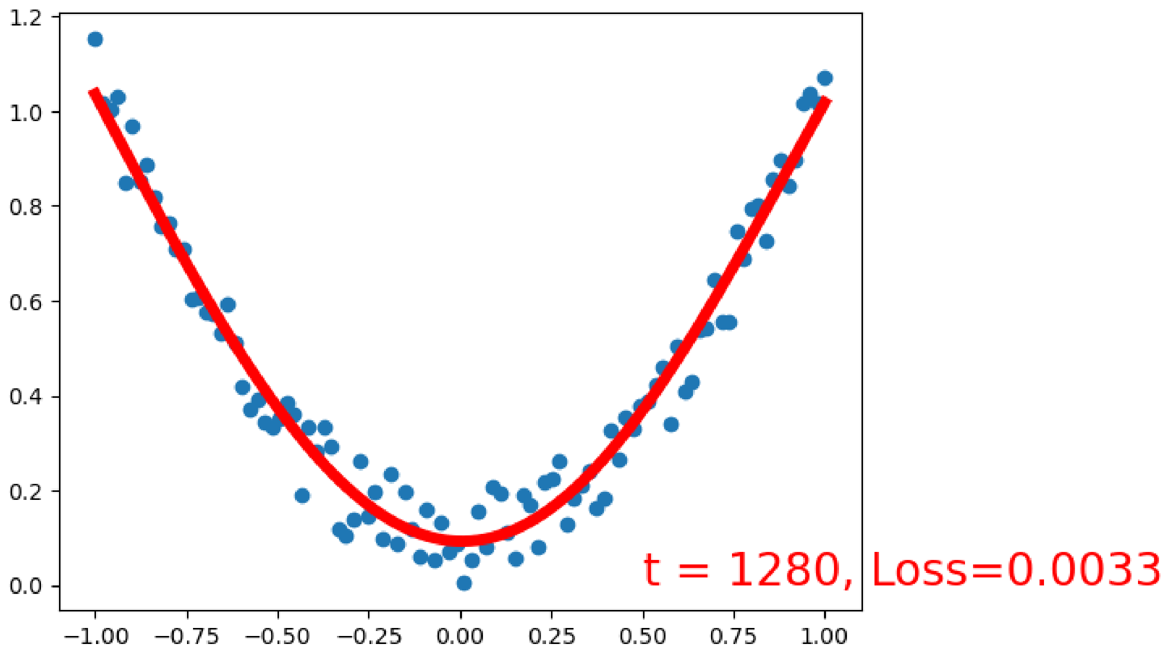
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1180, Loss=0.0034


t = 1200, Loss=0.0033
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

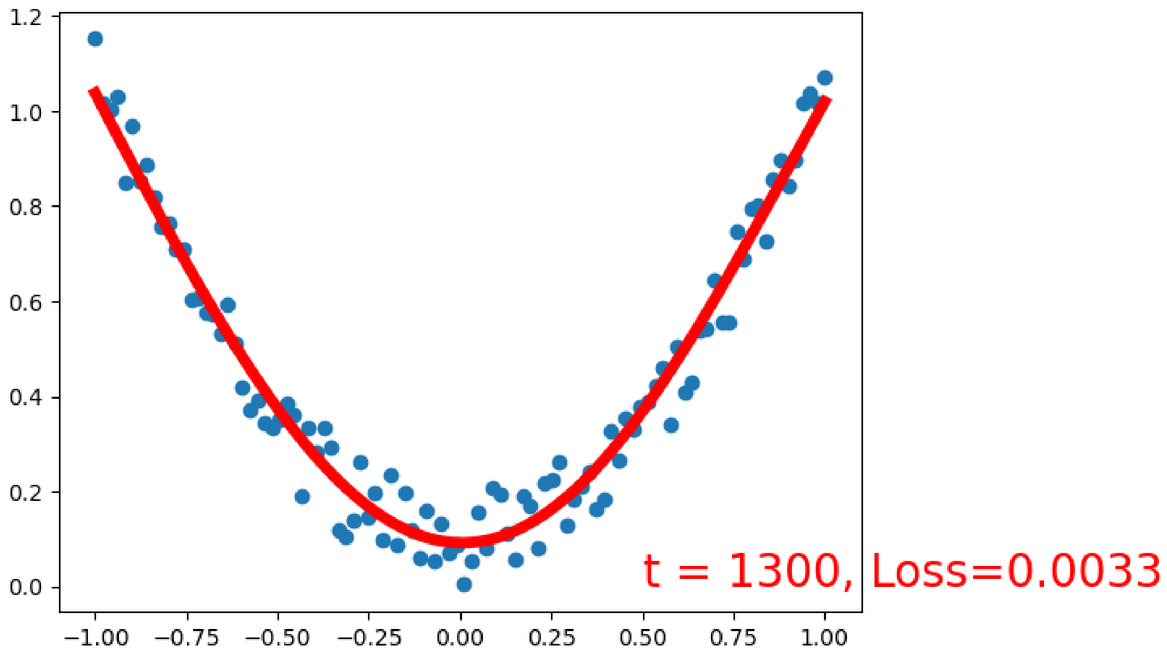

t = 1220, Loss=0.0033
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00


t = 1240, Loss=0.0033
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

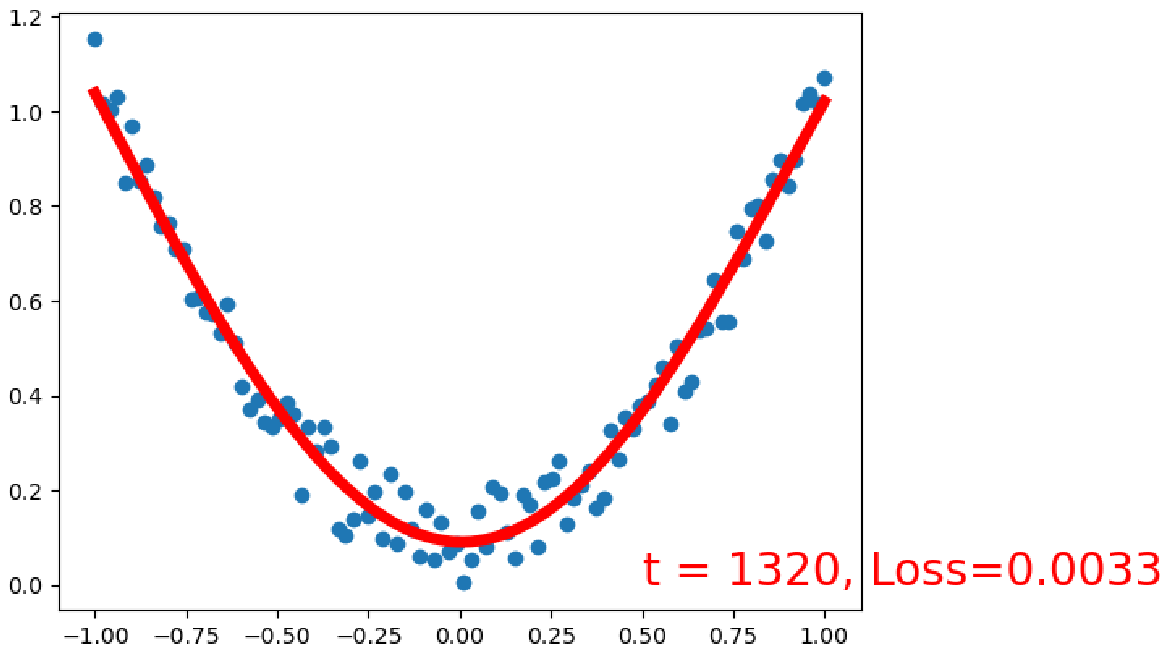

t = 1260, Loss=0.0033
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

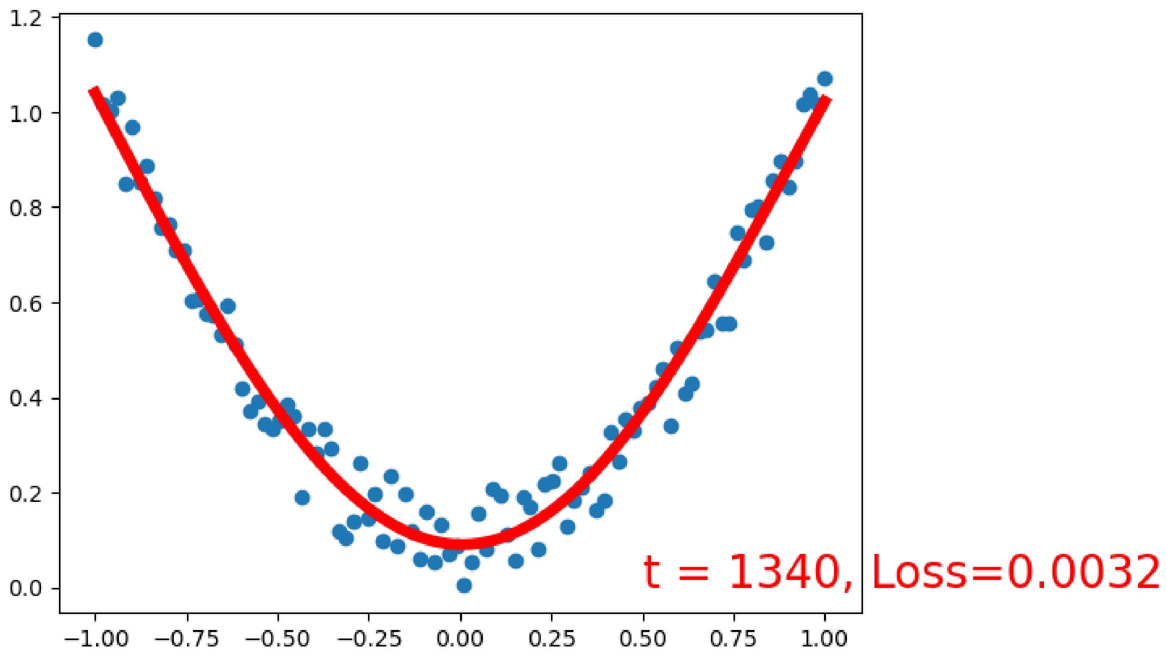


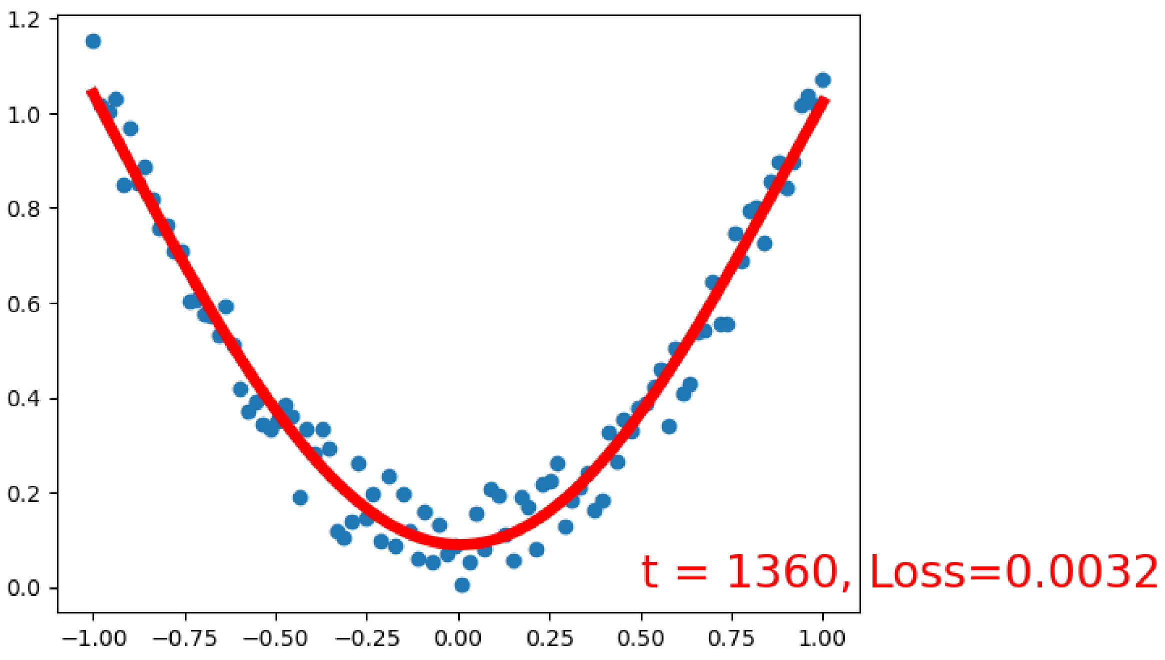
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1280, Loss=0.0033

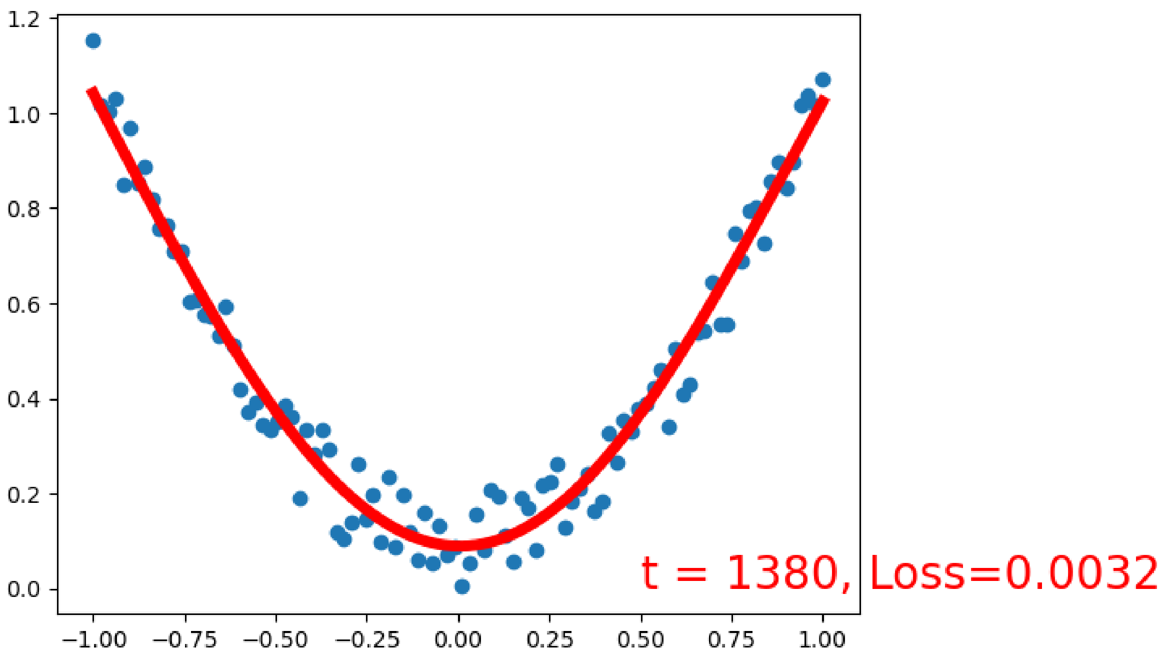


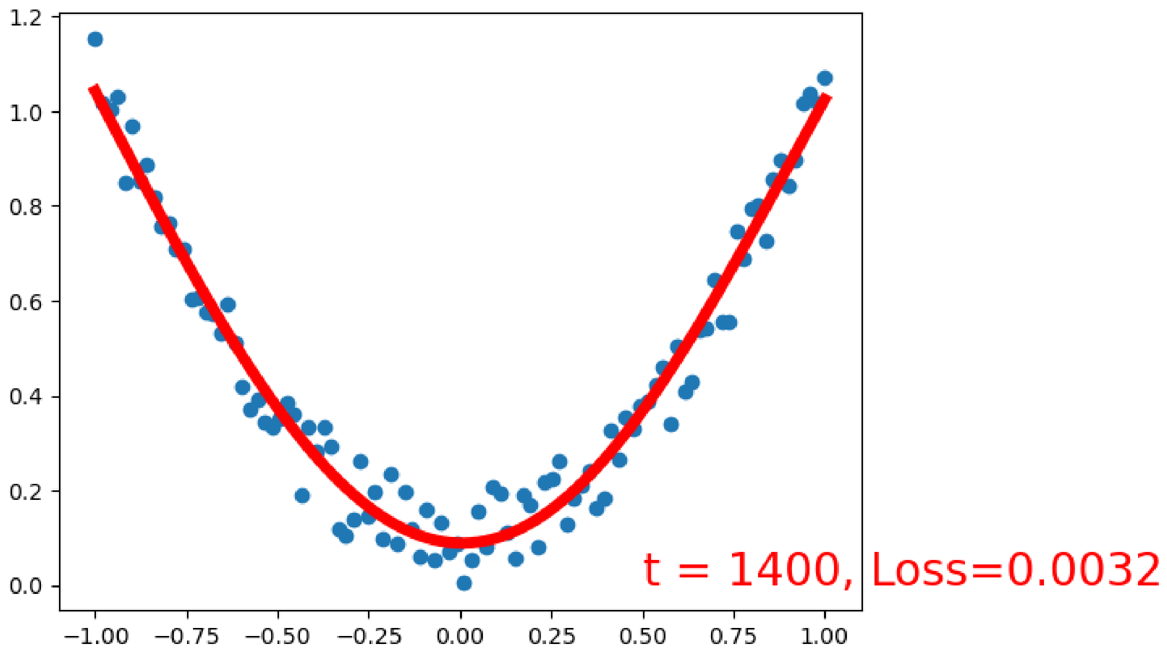
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1300, Loss=0.0033

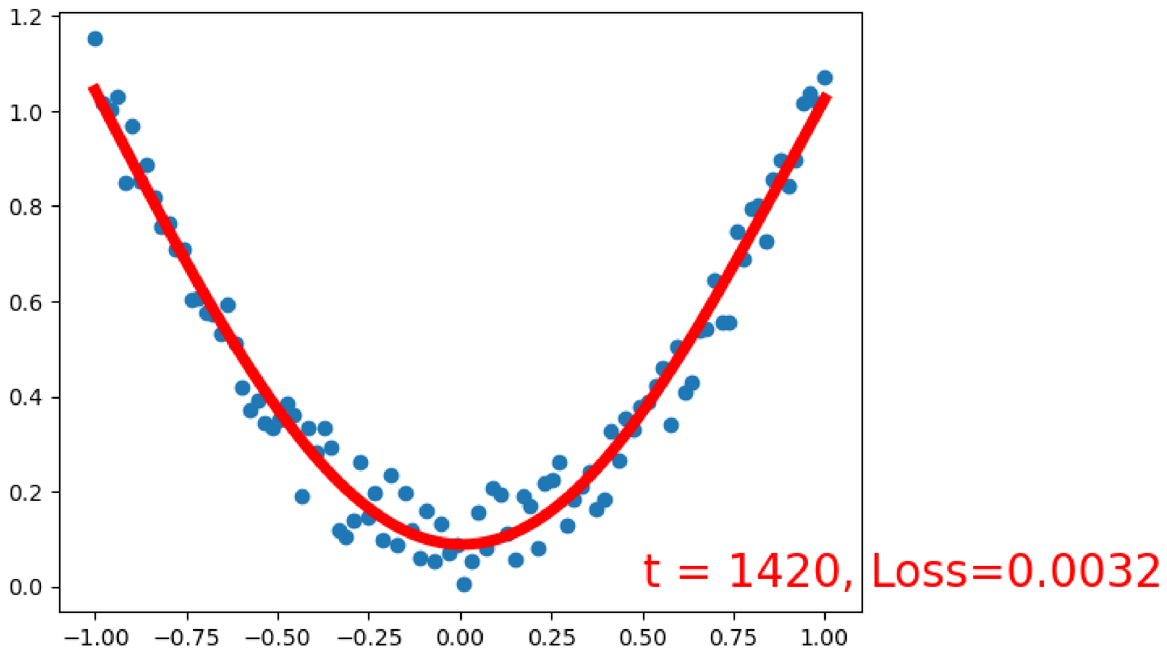

t = 1320, Loss=0.0033

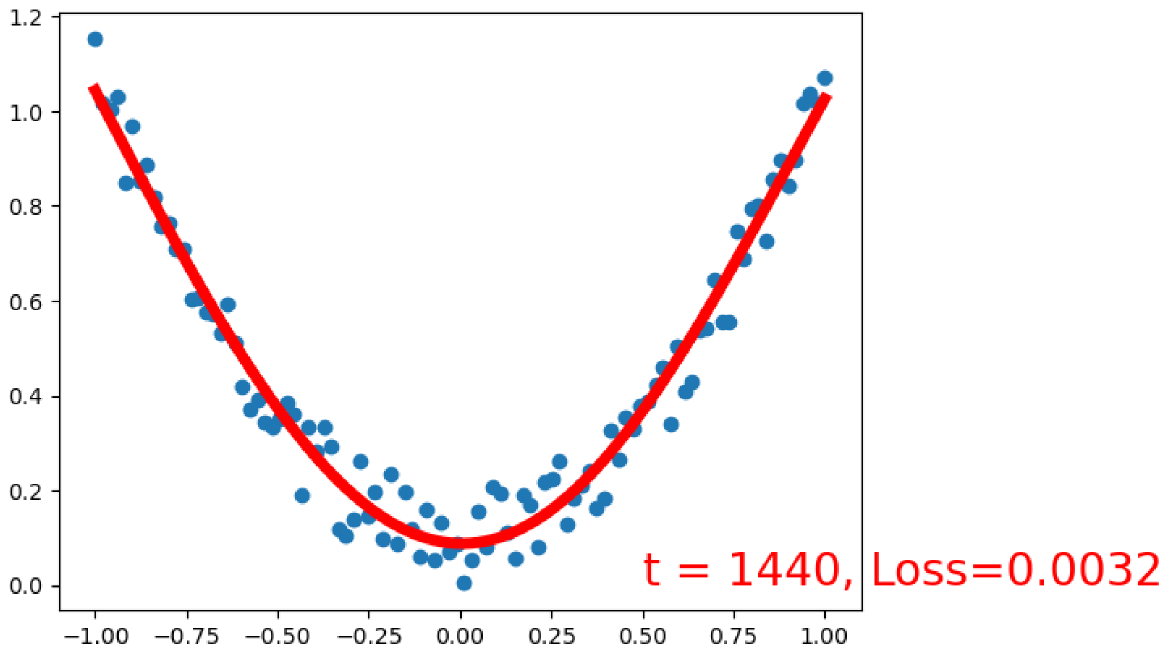

t = 1340, Loss=0.0032


t = 1360, Loss=0.0032

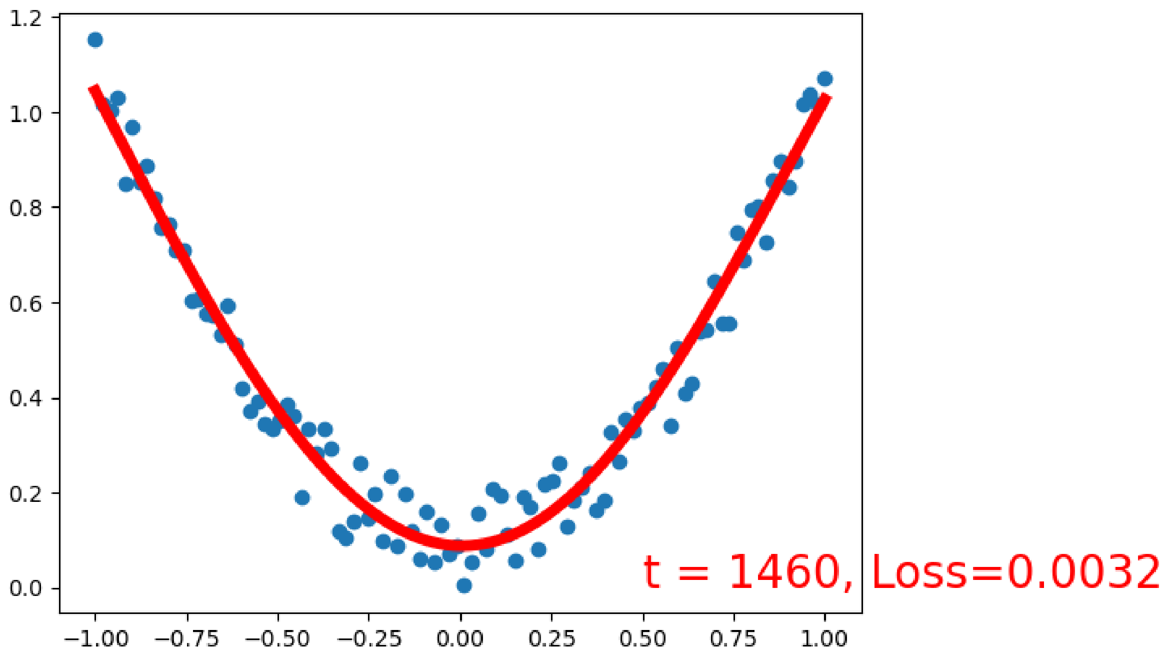

t = 1380, Loss=0.0032


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1400, Loss=0.0032

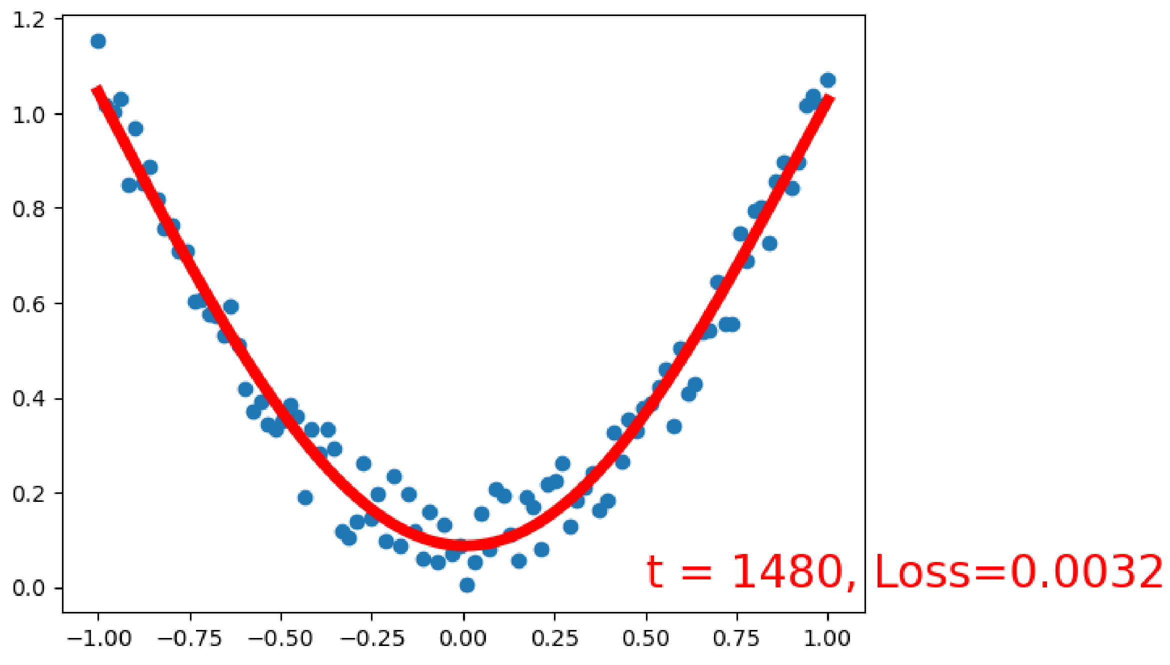

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1420, Loss=0.0032

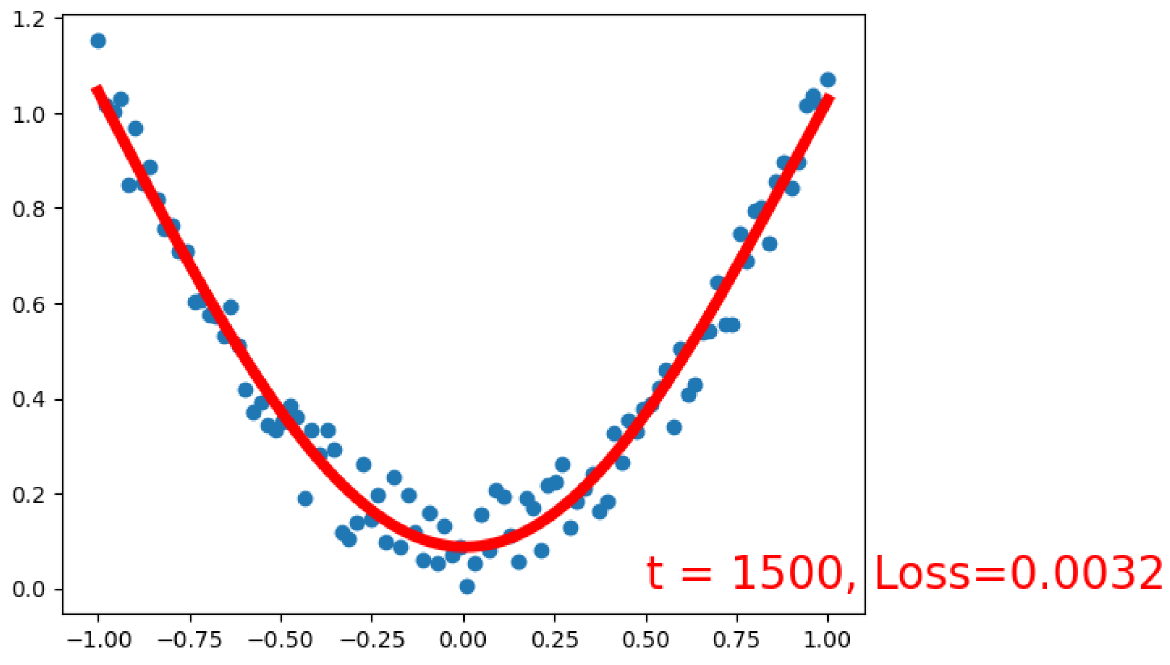


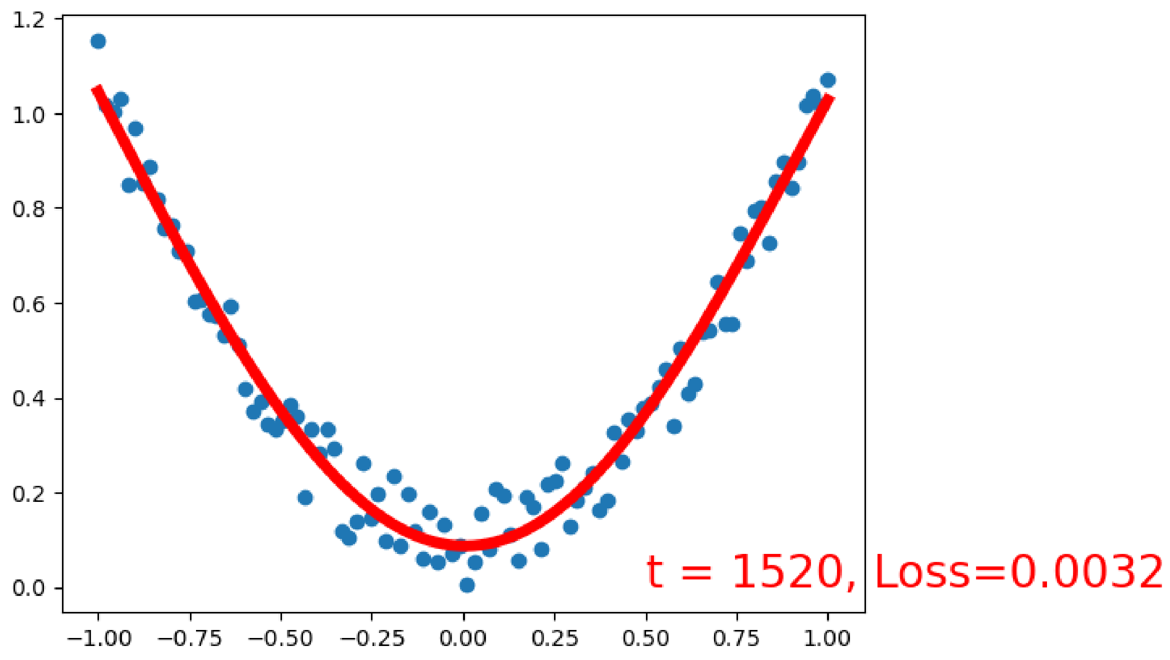
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1440, Loss=0.0032



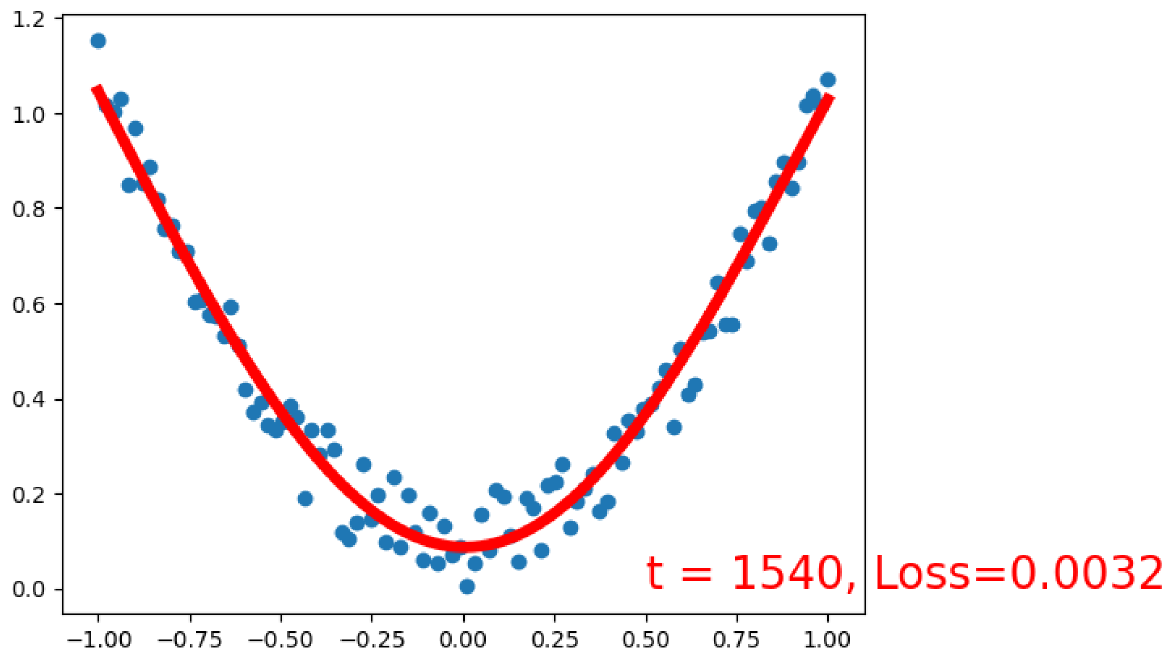
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1460, Loss=0.0032


t = 1480, Loss=0.0032

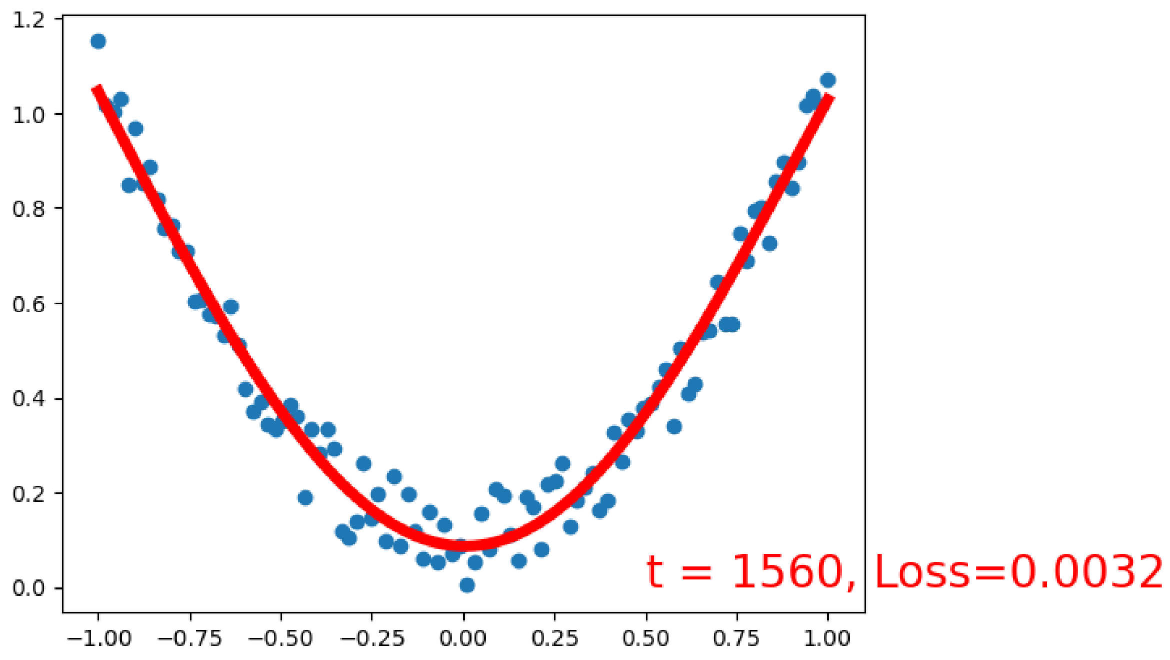

t = 1500, Loss=0.0032

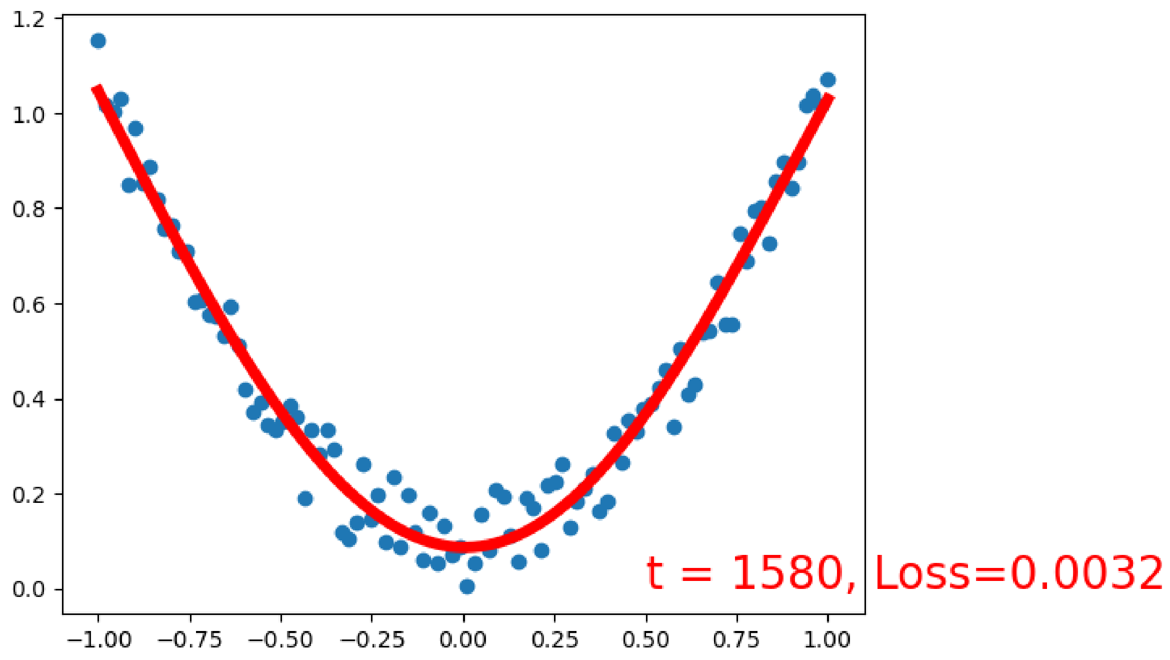


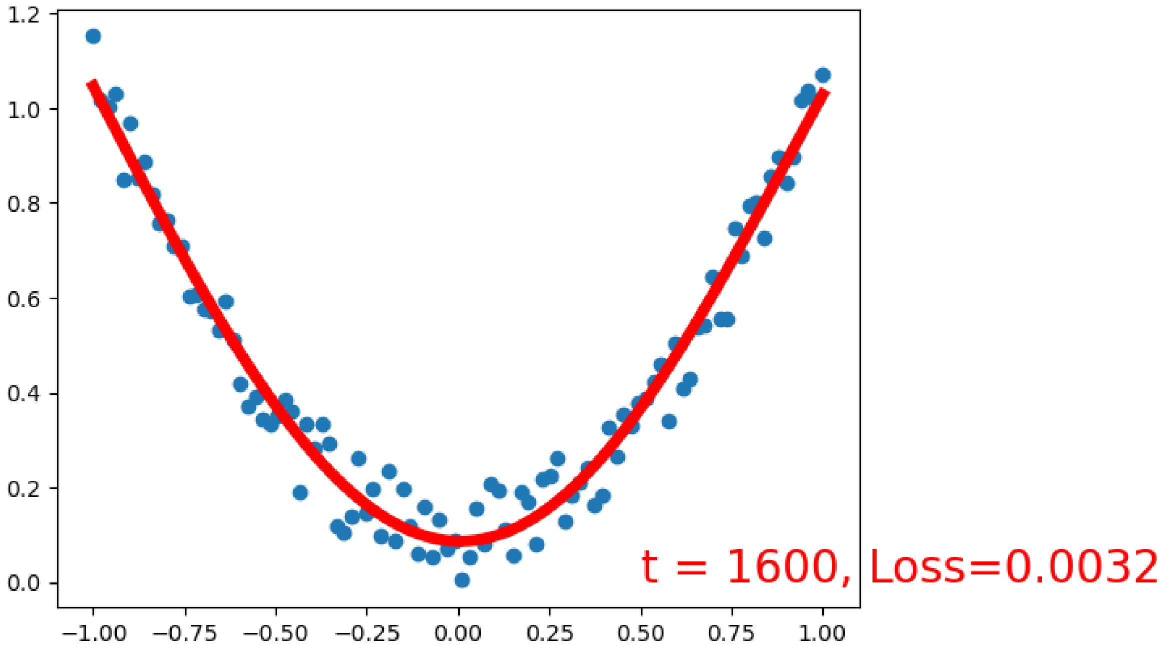
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1520, Loss=0.0032



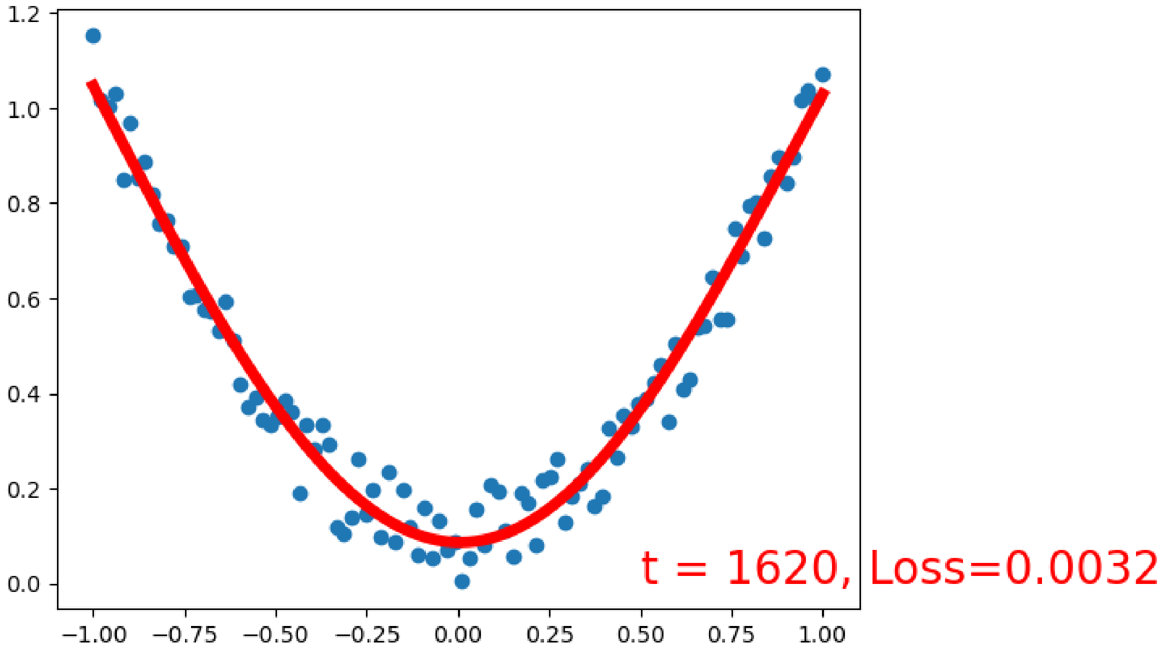
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1540, Loss=0.0032


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1560, Loss=0.0032

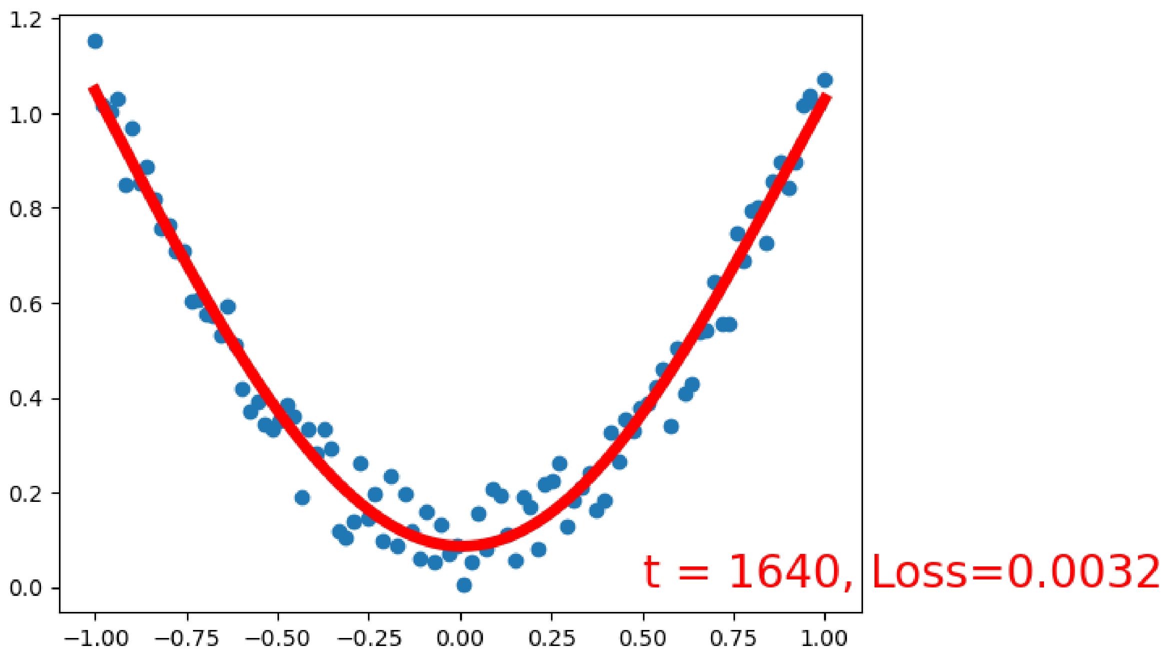

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1580, Loss=0.0032



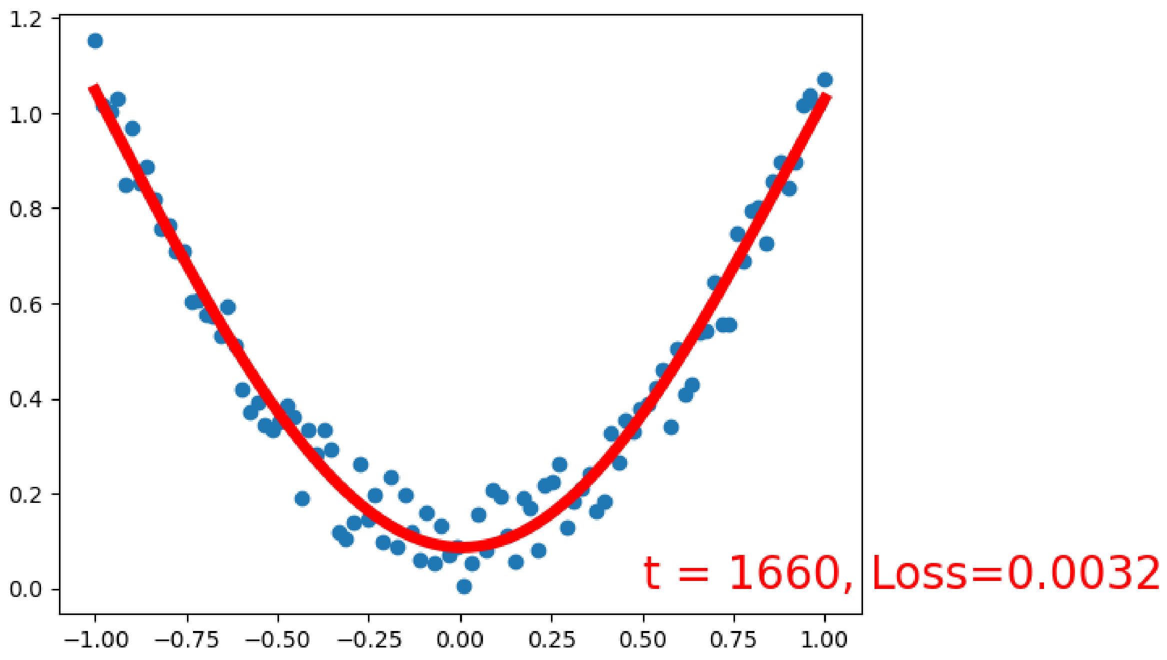
t = 1600, Loss=0.0032



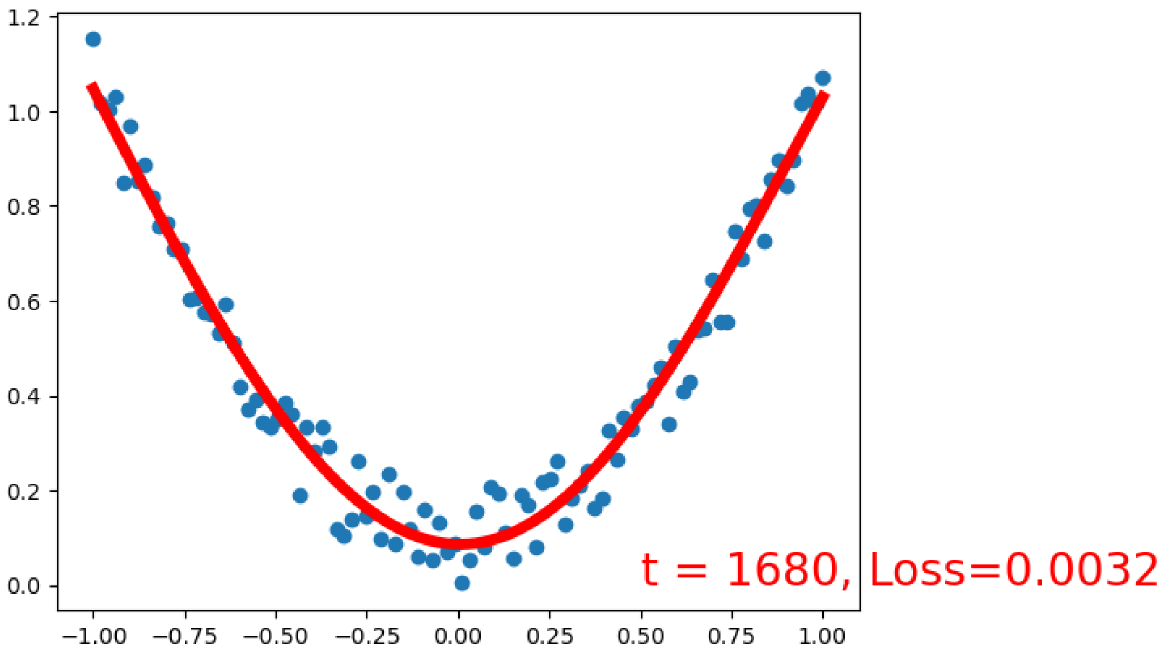
t = 1620, Loss=0.0032

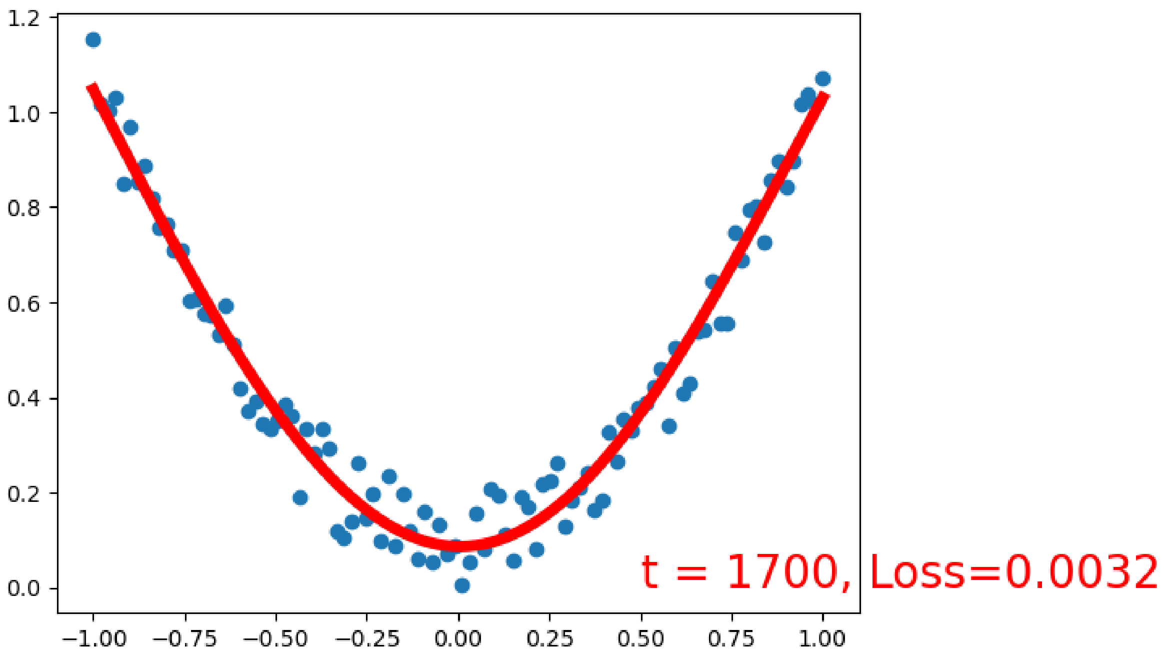


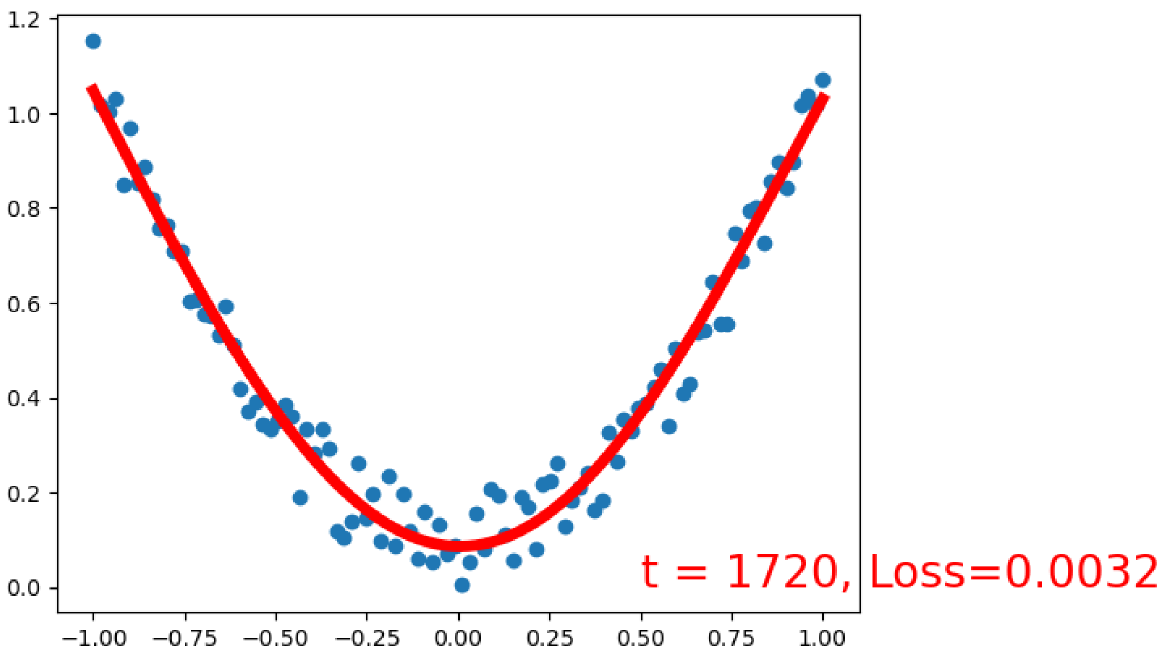
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1640, Loss=0.0032

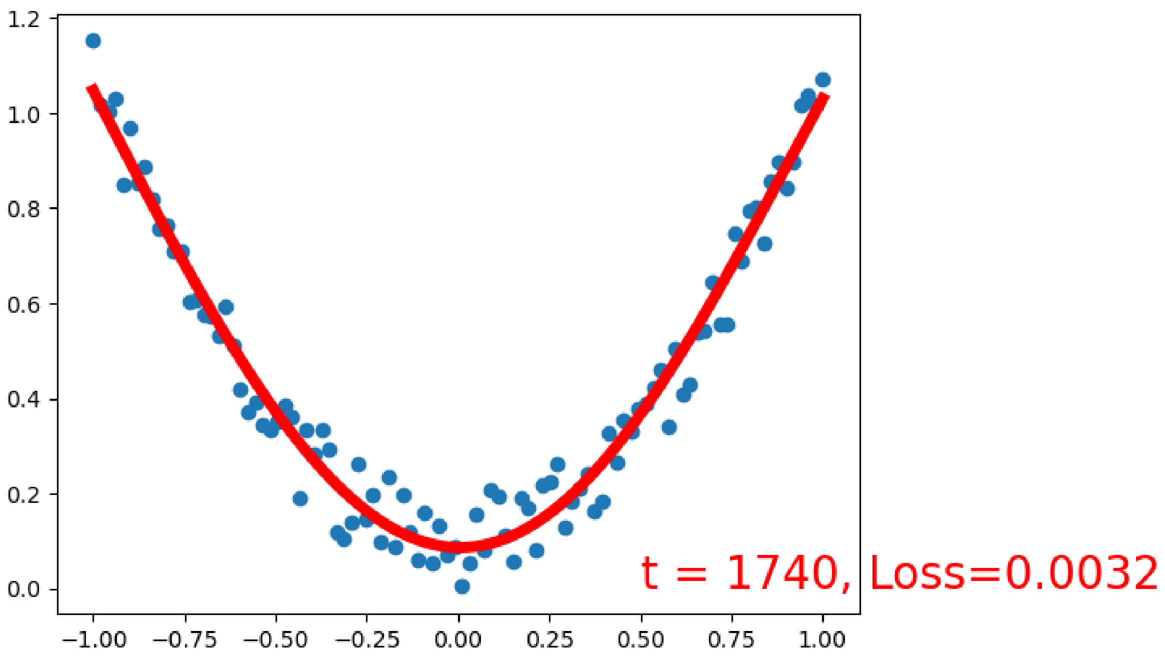


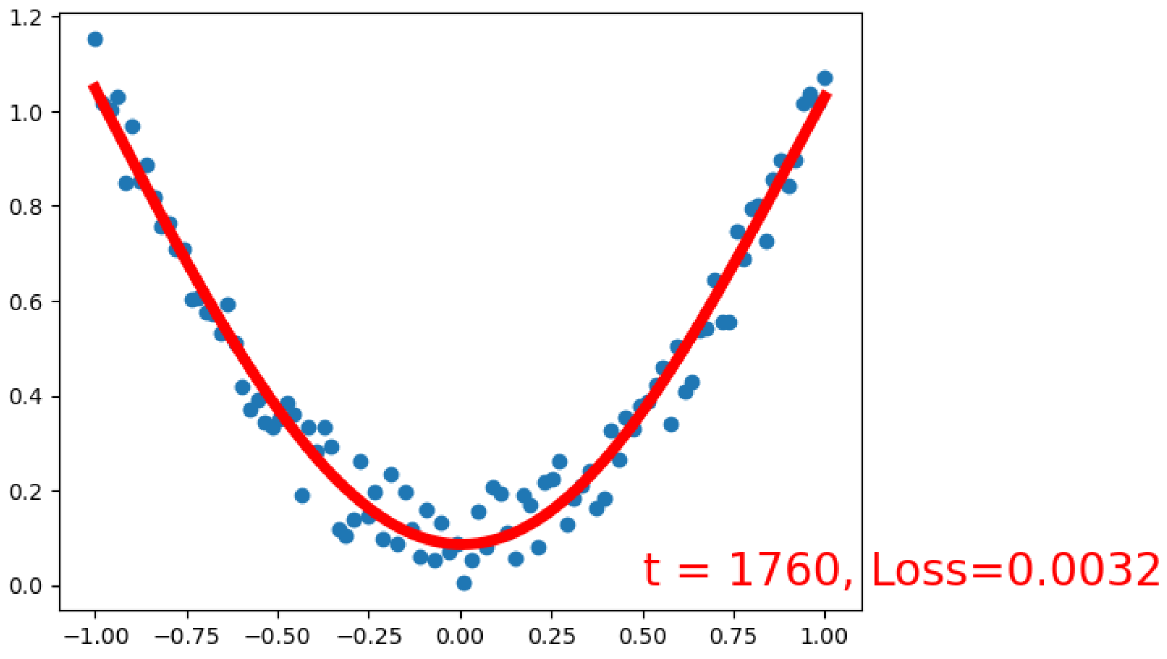
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1660, Loss=0.0032

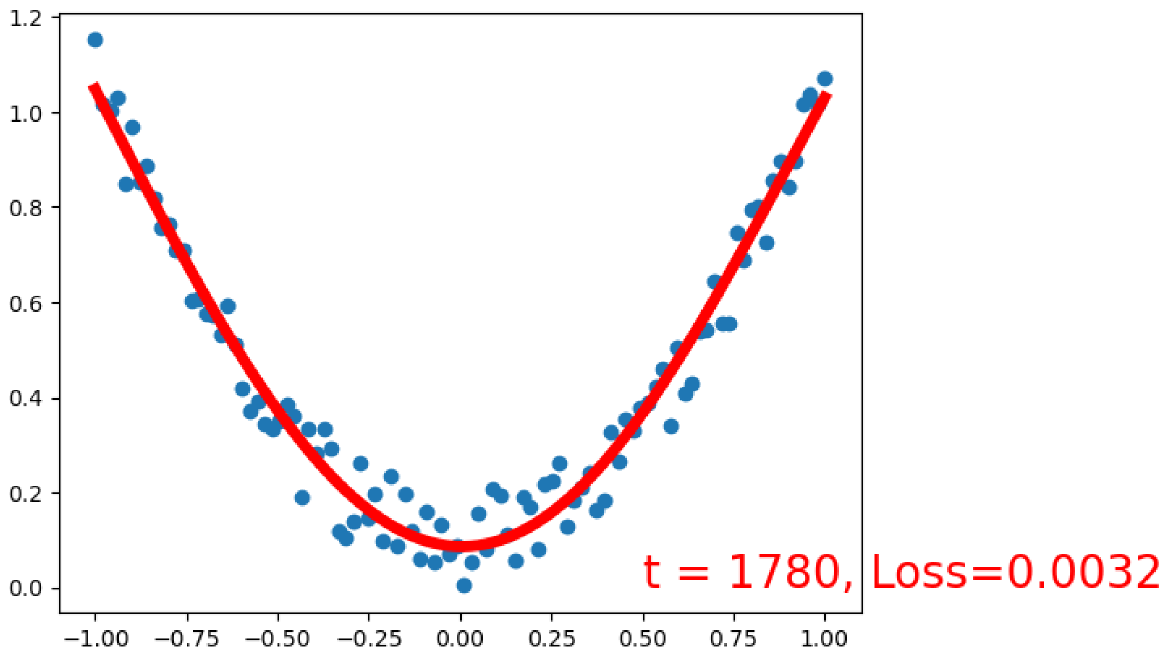

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1680, Loss=0.0032

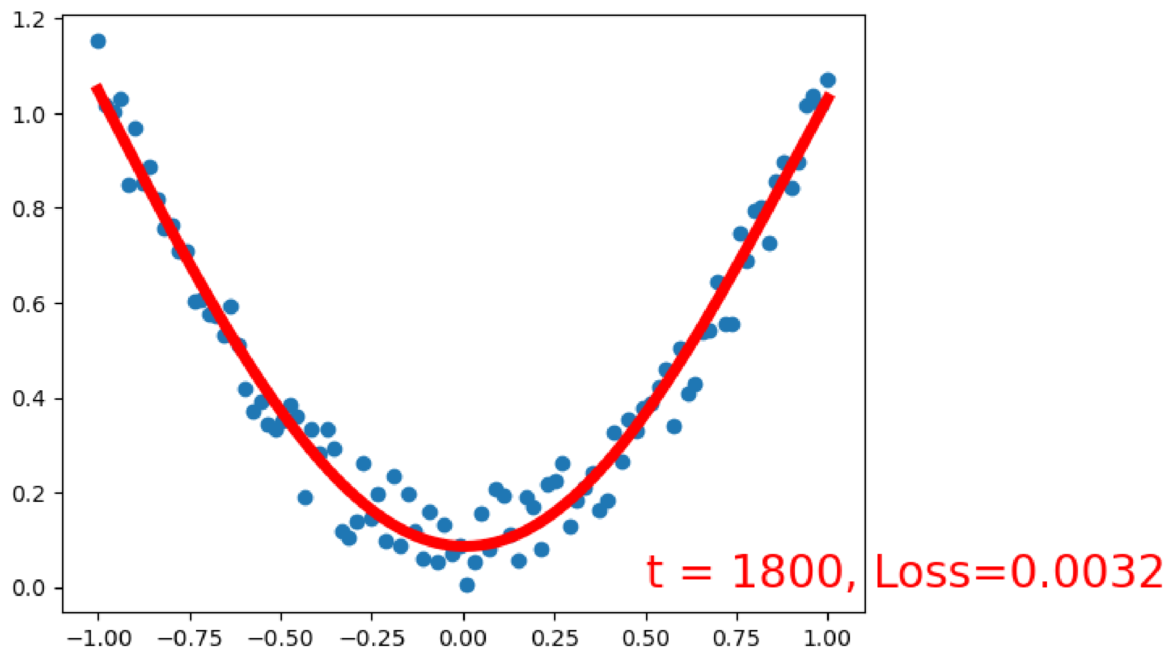

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1700, Loss=0.0032

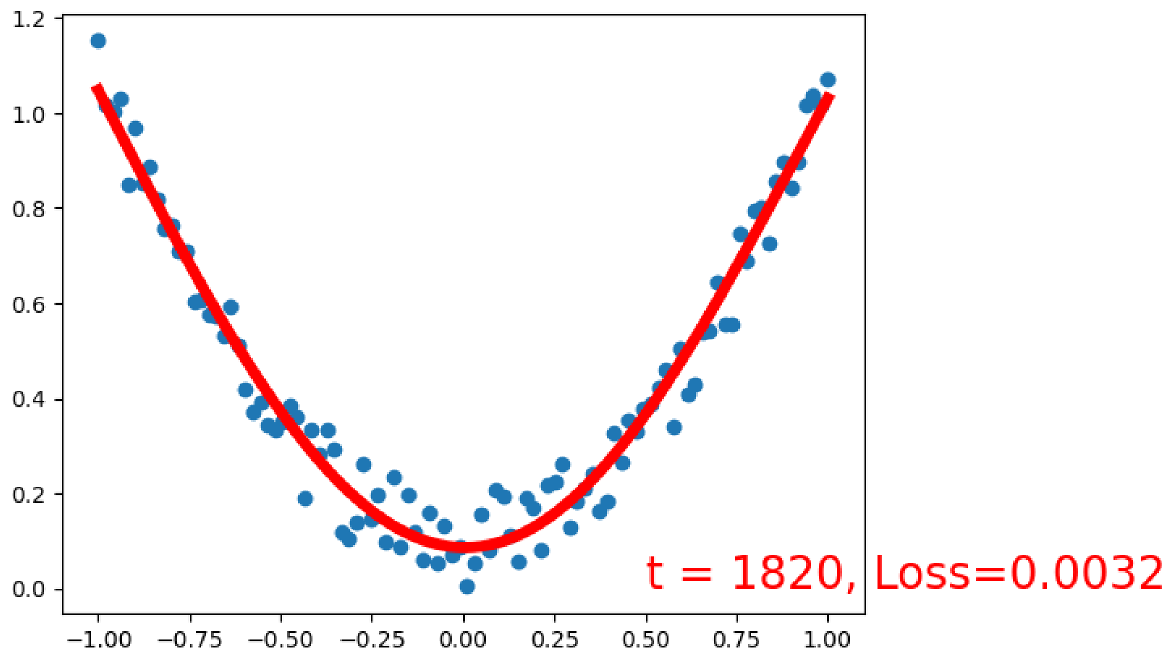

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1720, Loss=0.0032

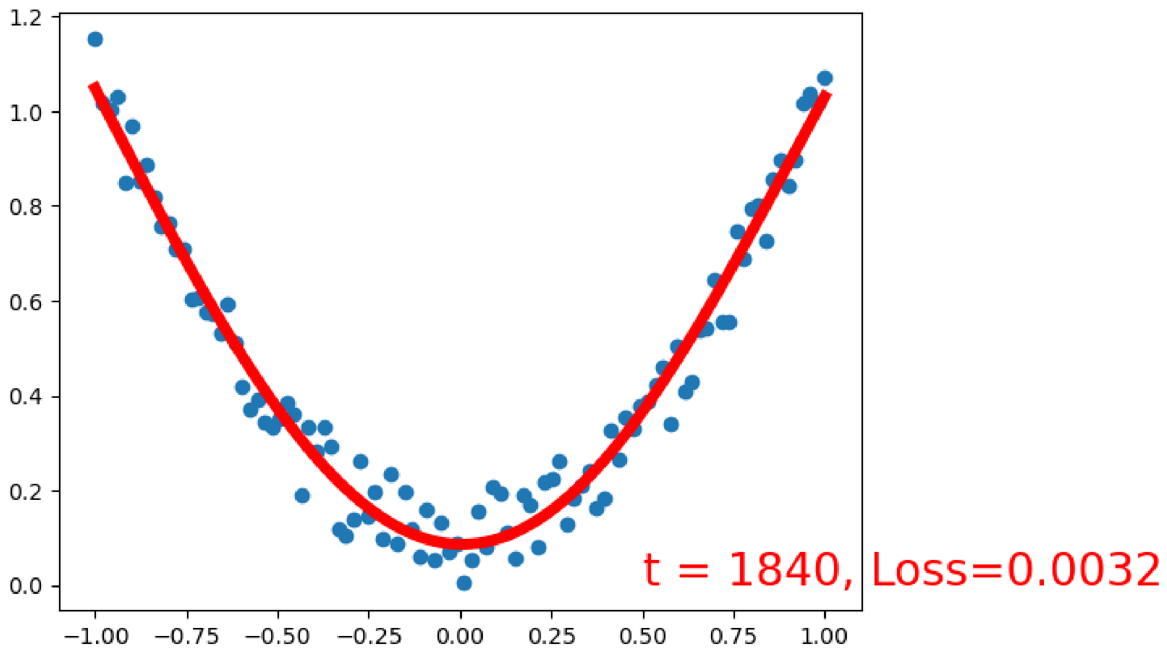

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1740, Loss=0.0032

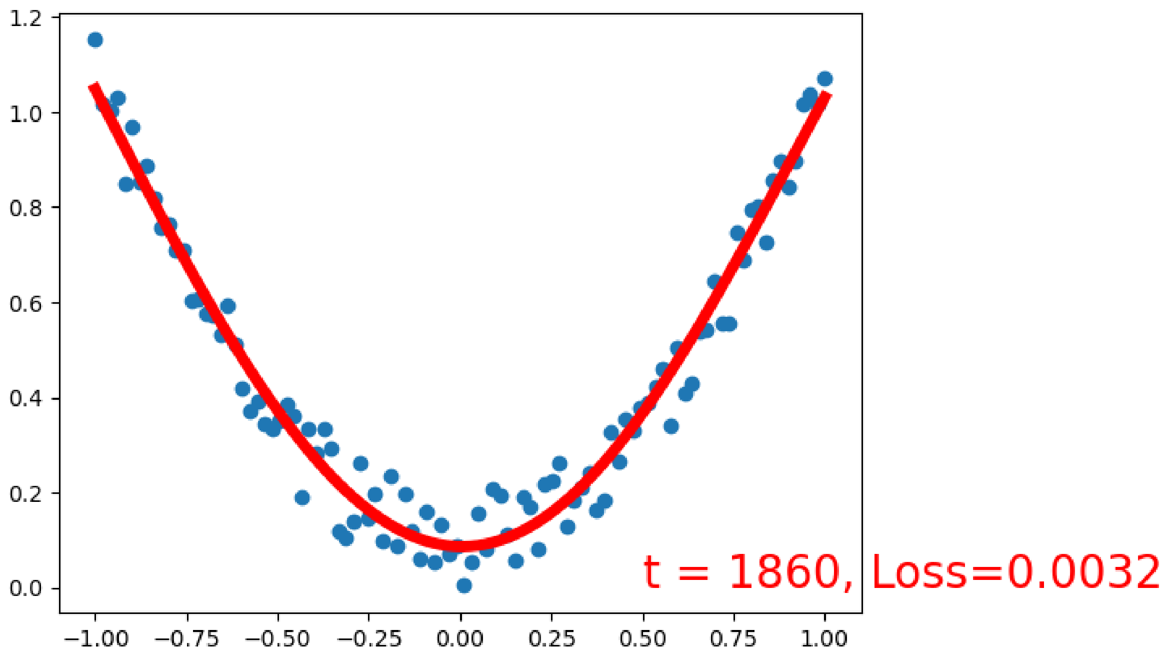

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1760, Loss=0.0032

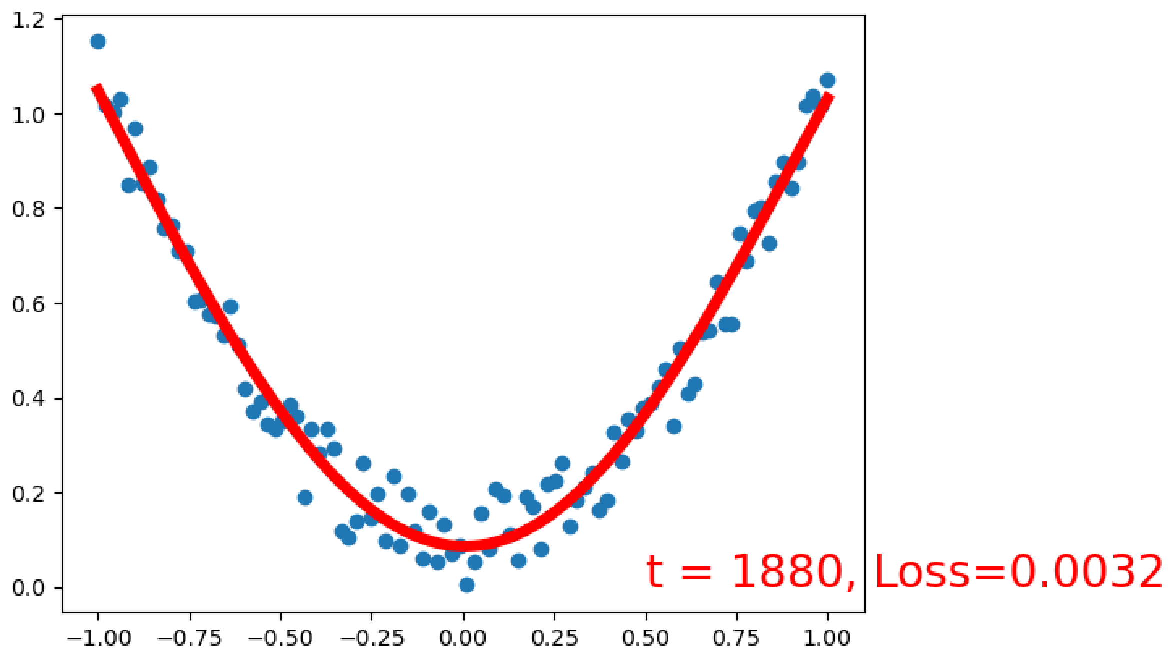

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1780, Loss=0.0032


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1800, Loss=0.0032

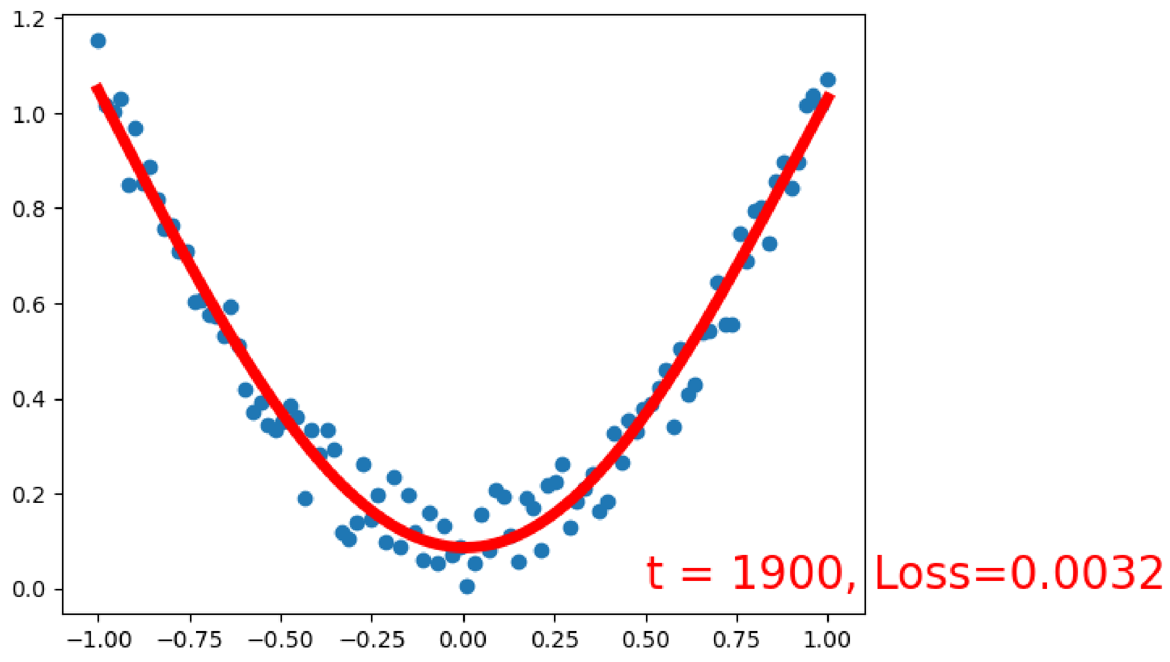

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1820, Loss=0.0032


t = 1840, Loss=0.0032
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

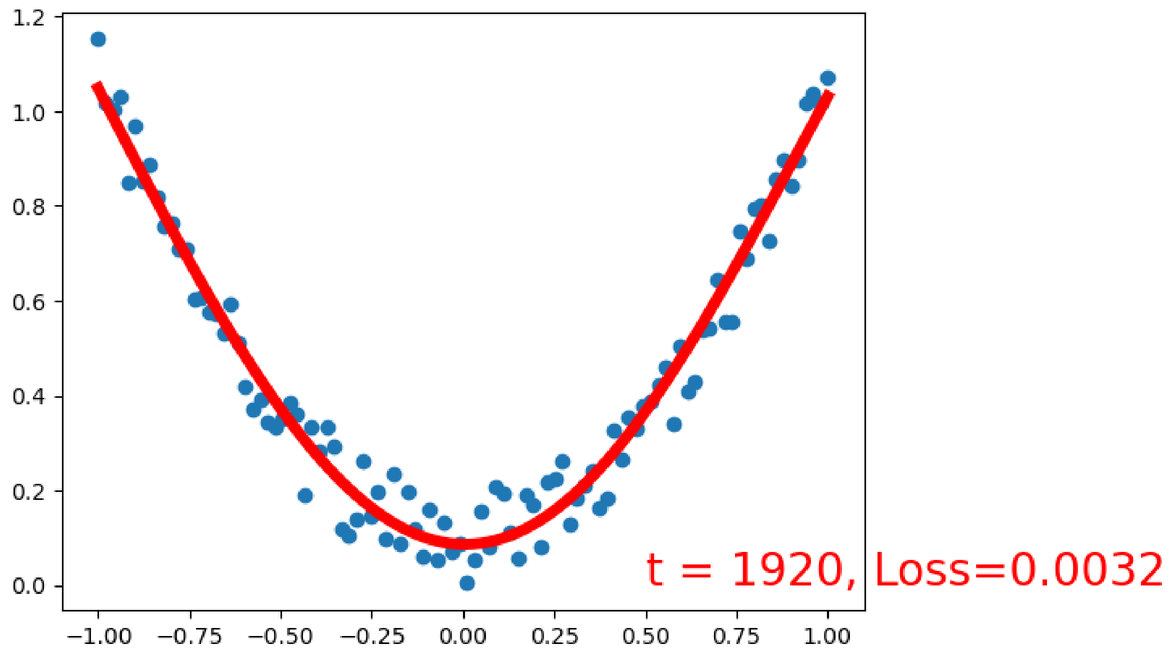

t = 1860, Loss=0.0032
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00

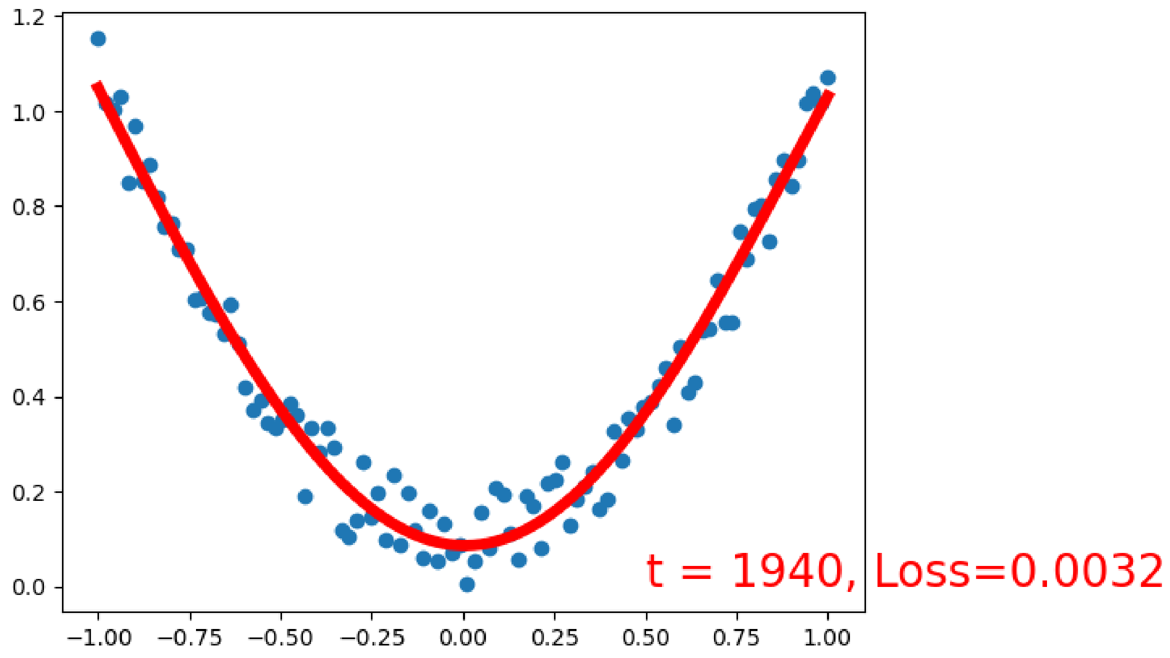


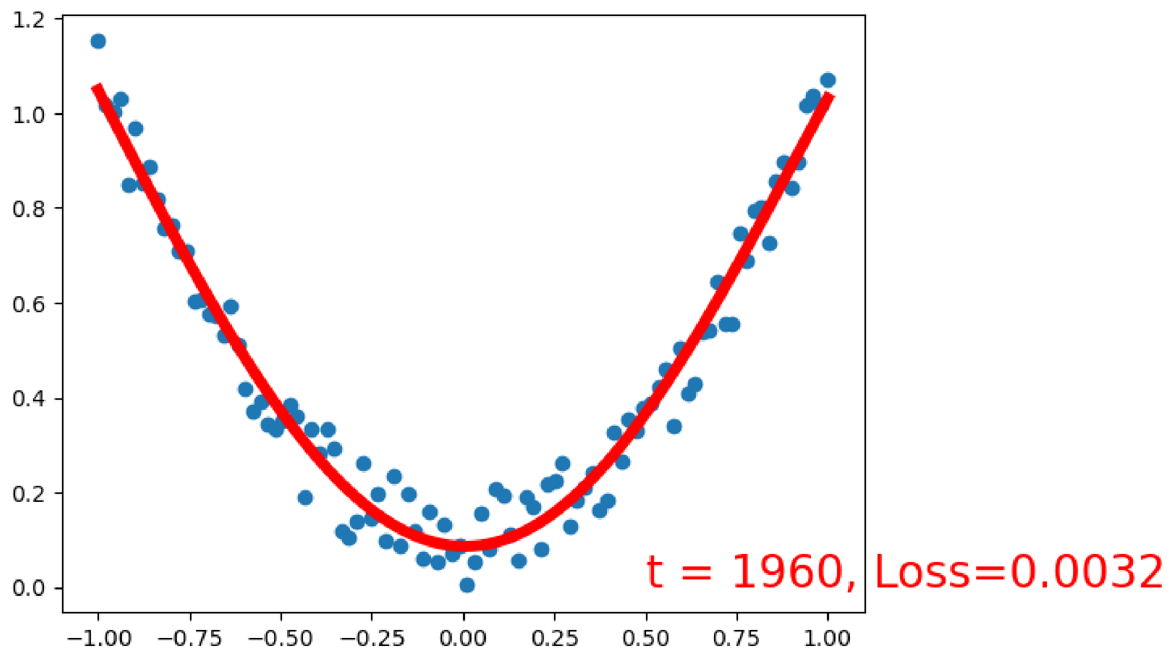
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1880, Loss=0.0032



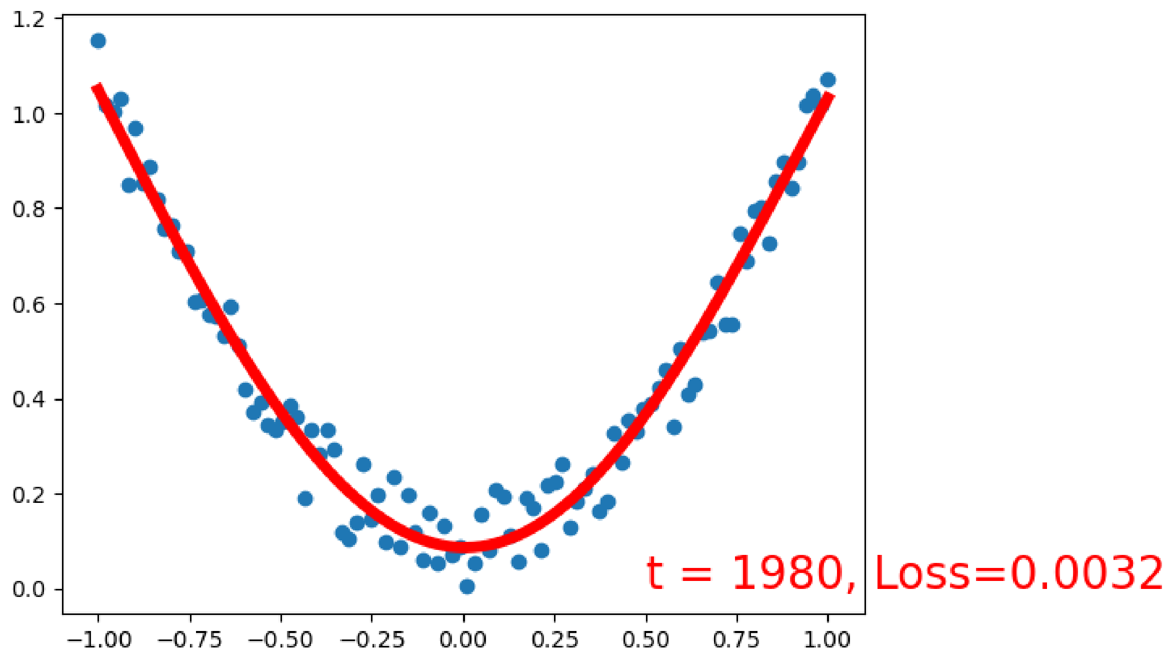
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1900, Loss=0.0032


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1920, Loss=0.0032


1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1940, Loss=0.0032



1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1960, Loss=0.0032



1.2
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
t = 1980, Loss=0.0032

```
<matplotlib.pyplot._IoffContext at 0x1ce1d35fb50>
```