COMP 9417 – Machine Learning

Homework 7: Neural Networks

Wanqing Yang – z5325987

Question 1

Let $f(x) = \frac{1}{1+e^{-x}},$ then $f'^{(x)} = f(x)\big(1 - f(x)\big)$

$$\frac{dp_i}{dw} = \frac{\partial p_i}{\partial w^T x_i}\frac{\partial w^T x_i}{\partial w} = p_i(1 - p_i)x_i$$

Question 2

$$\frac{dlnp_i}{dw} = \frac{dlnp_i}{dp_i}\frac{dp_i}{dw} = \frac{dlnp_i}{dp_i}\frac{dp_i}{dw^T x_i}\frac{dw^T x_i}{dw} = \frac{1}{p_i}p_i(1 - p_i)x_i = (1 - p_i)x_i$$

Question 3

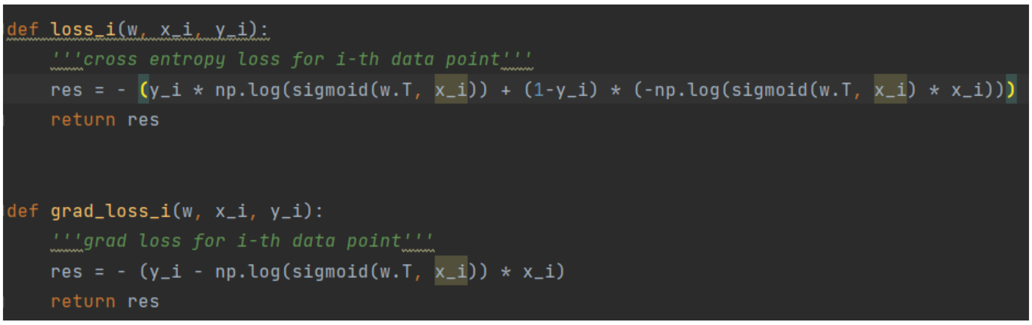
$$L(w) = -\sum_{i=1}^{n} y_i lnp_i + (1 - y_i)ln(1 - p_i)$$

Then $\frac{\partial L(w)}{\partial w} = -\sum_{i=1}^{n} y_i(1 - p_i)x_i + (1 - y_i)(-p_i x_i) = -\sum_{i=1}^{n} y_i x_i - x_i y_i p_i - p_i x_i + x_i y_i p_i$

$$= -\sum_{i=1}^{n}(y_i - p_i)x_i$$

Then $\Delta w = -\eta\frac{\partial L(w)}{\partial w} = \eta\sum_{i=1}^{n}(y_i - p_i)x_i$ $\qquad w \leftarrow w + \Delta w$

Question 4



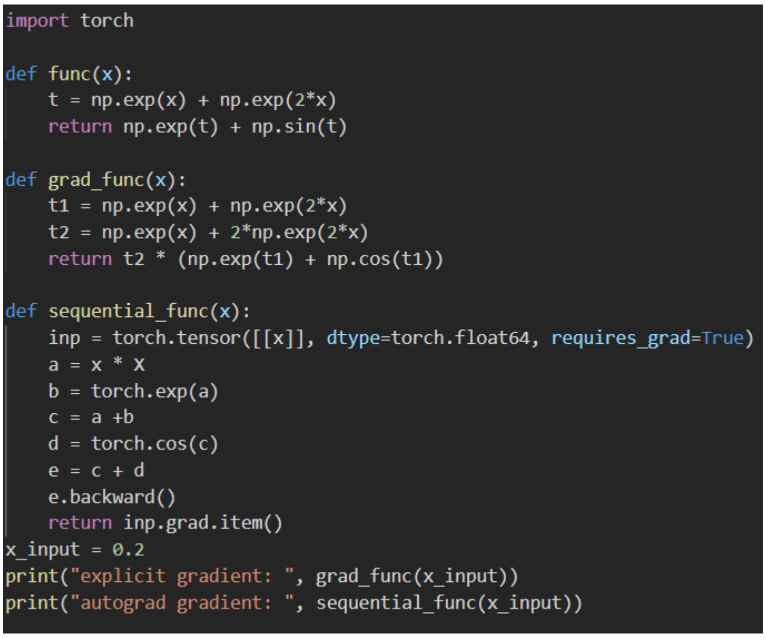
```
def loss_i(w, x_i, y_i):
    '''cross entropy loss for i-th data point'''
    res = - (y_i * np.log(sigmoid(w.T, x_i)) + (1-y_i) * (-np.log(sigmoid(w.T, x_i) * x_i)))
    return res


def grad_loss_i(w, x_i, y_i):
    '''grad loss for i-th data point'''
    res = - (y_i - np.log(sigmoid(w.T, x_i)) * x_i)
    return res
```

Question 8

$$\boldsymbol{f(x) = x^2 + e^{x^2} + cos\big(x^2 + e^{x^2}\big)}$$

$$\boldsymbol{f'(x) = 2x + 2xe^{x^2} - sin\big(x^2 + e^{x^2}\big)2xe^{x^2}}$$



```python
import torch

def func(x):
    t = np.exp(x) + np.exp(2*x)
    return np.exp(t) + np.sin(t)

def grad_func(x):
    t1 = np.exp(x) + np.exp(2*x)
    t2 = np.exp(x) + 2*np.exp(2*x)
    return t2 * (np.exp(t1) + np.cos(t1))

def sequential_func(x):
    inp = torch.tensor([[x]], dtype=torch.float64, requires_grad=True)
    a = x * X
    b = torch.exp(a)
    c = a +b
    d = torch.cos(c)
    e = c + d
    e.backward()
    return inp.grad.item()
x_input = 0.2
print("explicit gradient: ", grad_func(x_input))
print("autograd gradient: ", sequential_func(x_input))
```