HW6_abuitano

Problem 1

a) False: The activation functions involved won't let $h_t = h_{t-1}$ O_t has sigmoid activation, while tanh(ct) has tanh activation. f_t, i_t, $\bar{c}_t$ also have their own non linearities which will affect the value of h_t.

b) False: $C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$, so even if $f_t \approx 0$, error should still get backpropagated through o_t, i_t and $\tilde{c}_t$.

c) True, f_t, i_t and O_t have sigmoid activation, so they'll only have values between 0 and 1.

d) False. with sigmoid activation on f_t, O_t, i_t, all their entries will be non-negative but won't necesarily sum to 1. each entry has sigmoid applied independently.

e) f_t, i_t, O_t all should be the same dimension as h_t

f) $\boxed{h_1 = 0.21741}$ * Code attached for computation

$\boxed{h_2 = -0.18988}$

g) $MSE = \frac{1}{2}\left[(y_1 - h_1)^2 + (y_2 - h_2)^2\right] = \frac{1}{2}[0.07985 + 0.97986) = \boxed{0.5298}$

In [17]:

```
import numpy as np

wf = [1, 2]
wi = [-1, 0]
wc = [1, 2]
wo = [3, 0]

uf = [0.5]
ui = [2]
uc = [1.5]
uo = [-1]

bf = [0.2]
bi = [-0.1]
bc = [0.5]
bo = [0.8]

x1 = [[1],[0]]
x2 = [[0.5], [-1]]

y1 = 0.5
y2 = 0.8

h0 = 0
c0 = 0
```

In [18]:

```
def sigmoid(x):
    return 1/(1+np.exp(-x))
def tanh(x):
    return (np.exp(x)-np.exp(-x))/(np.exp(x)+np.exp(-x))
```

In [19]:

```
def foi(w, u, b, x, h):
    return sigmoid(np.dot(w,x) + np.dot(u,h) + b)
def c_hash(w, u, b, x, h):
    return tanh(np.dot(w,x) + np.dot(u,h) + b)

f1 = foi(wf, uf, bf, x1, h0)
i1 = foi(wi, ui, bi, x1, h0)
o1 = foi(wo, uo, bo, x1, h0)
chash1 = c_hash(wc, uc, bc, x1, h0)

c1 = f1*c0 + i1*chash1

h1 = o1*tanh(c1)
```

In [20]:

```
h1
```

Out[20]:

```
array([0.21741464])
```

In [23]:

```
f2 = foi(wf, uf, bf, x2, h1)
i2 = foi(wi, ui, bi, x2, h1)
o2 = foi(wo, uo, bo, x2, h1)
chash2 = c_hash(wc, uc, bc, x2, h1)

c2 = f2*c1 + i2*chash2

h2 = o2*tanh(c2)
```

```
In [24]: h2

Out[24]: array([-0.18988225])

In [25]: f2

Out[25]: array([0.23302782])

In [26]: i2

Out[26]: array([0.45880094])

In [27]: o2

Out[27]: array([0.88919901])

In [28]: chash2

Out[28]: array([-0.58752507])
```