prep-exercises

March 5, 2021

[4]:

```
import numpy as np
```

[6]:

```
# 3.1
z = 1.0
h = lambda x: np.exp(x)/(1 + np.exp(x))

p_1 = h(z)
p_2 = 1 - p_1

print('p(y = 1|x) = ' + str(p_1))
print('p(y =-1|x) = ' + str(p_2))
```

```
p(y = 1|x) = 0.7310585786300049
p(y =-1|x) = 0.2689414213699951
```

[7]:

```
# 3.2
z = np.array([0,-1,1])
softmax = lambda z: np.exp(z)/sum(np.exp(z))

p_softmax = softmax(z)
p_1 = p_softmax[0]
p_2 = p_softmax[1]
p_3 = p_softmax[2]

print('p(y = 1|x) = ' + str(p_1))
print('p(y = 2|x) = ' + str(p_2))
print('p(y = 3|x) = ' + str(p_3))
```

```
p(y = 1|x) = 0.24472847105479767
p(y = 2|x) = 0.09003057317038046
p(y = 3|x) = 0.6652409557748219
```

[13]:

```
# 3.3
y = np.eye(3)
def cross_entropy(z,y):
    return -(y@np.log(softmax(z)))

print(cross_entropy(z,y))
```

```
[1.40760596 2.40760596 0.40760596]
```

[14]:

```
# 3.4
# W^1: 30 x 144
# b^1: 30 x 1

# W^2: 4 x 30
# b^2: 4 x 1

print('The network has ' + str(30*144 + 30 + 4*30 + 4) + ' parameters')
```

```
The network has 4474 parameters
```

[ ]:

```
# 3.5
# W^1:  5 x  5 x  1 x  4
# b^1:  1 x  1 x  1 x  4
# Q^1: 12 x 12 x  4
```

[ ]:

```
# 3.6
# W^2:  3 x  3 x  4 x  8
# b^2:  1 x  1 x  1 x  8
# Q^1:  6 x  6 x  8
```

[ ]:

```
# 3.7
# W^3: 60 x 288
# b^3: 60 x 1

# W^4:  4 x 60
# b^4:  4 x 1
```