

## Solution to Assignment 1, Problem 1(a)

*Akash Rana*

**Softmax** Prove that *softmax* is invariant to constant offsets in the input, that is, for any input vector  $x$  and any constant  $c$ ,

$$\text{softmax}(\mathbf{x}) = \text{softmax}(\mathbf{x} + c), \quad (1)$$

where  $(\mathbf{x} + c)$  means adding the constant  $c$  to every dimension of  $\mathbf{x}$ .

---

$$\text{softmax}(\mathbf{x})_j = \frac{e^{x_j}}{\sum_{k=1} e^{x_k}} \quad (2)$$

Solution:

$$\text{softmax}(\mathbf{x} + c)_j = \frac{e^{(x_j+c)}}{\sum_{k=1} e^{(x_k+c)}} \quad (3)$$

$$= \frac{e^c}{e^c} \frac{e^{(x_j)}}{\sum_{k=1} e^{(x_k)}} \quad (4)$$

$$= \text{softmax}(\mathbf{x})_j \quad (5)$$

## Solution to Assignment 1, Problem 1(b)

Akash Rana

Given an input matrix of N-rows and d-columns, compute the softmax prediction for each row. Write your implementation in `q1_softmax.py`. You may test by executing `python q1_softmax.py`.

**Note:** The provided tests are not exhaustive. Later parts of the assignment will reference this code so it is important to have a correct implementation. Your implementation should also be efficient and vectorized whenever possible. A non-vectorized implementation will not receive full credit!

---

```
import numpy as np

def softmax(x):
    #x.shape by default gives column value
    if len(x.shape) > 1:
        # Matrix
        ### YOUR CODE HERE
        c = np.max(x, axis=1).reshape(-1,1) #-1 here means, internally
        #numpy is just calculating, to get the missing dimension.
        x = np.exp(x-c) / np.sum(np.exp(x-c), axis=1).reshape(-1,1)
        print (np.shape(x))
        ### END YOUR CODE
    else:
        # Vector
        ### YOUR CODE HERE
        c = np.max(x)
        x = np.exp(x-c) / np.sum(np.exp(x-c))
        ### END YOUR CODE

    assert x.shape == orig_shape
    return x
```

---

**Solution to Assignment 1, Problem 2(a)***Akash Rana*

Derive the gradients of the sigmoid function and show that it can be rewritten as a function of the function value (i.e. in some expression where only  $\sigma(x)$ , but not  $x$ , is present). Assume that the input  $x$  is a scalar for this question.

---

Denote the sigmoid function as  $\sigma(z)$ ,

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \tag{6}$$

Using chain rule,

$$\begin{aligned} \sigma'(z) &= \frac{-1}{(1 + e^{-z})^2} \times (-e^{-z}) \\ &= \frac{1}{1 + e^{-z}} \left( \frac{e^{-z}}{1 + e^{-z}} \right) \\ &= \left( \frac{1}{1 + e^{-z}} \right) (1 + e^{-z}) \\ &= \sigma(z) - \sigma^2(z) \\ &= \sigma(z)(1 - \sigma(z)) \end{aligned}$$