# COMP9444 Neural Networks and Deep Learning
# Term 3, 2020

## Solutions to Exercise 6: Word Vectors

**This page was last updated: 10/26/2020 08:19:34**

---



1. Consider the sentence

   "two flowers grew tall on two tall towers"



   a. Write the co-occurrence matrix X for this sentence, using a 4-word context window (i.e. two context words on either side of the central word)



| | flowers | grew | on | tall | towers | two |
|---|---|---|---|---|---|---|
| flowers | 0 | 1 | 0 | 1 | 0 | 1 |
| grew | 1 | 0 | 1 | 1 | 0 | 1 |
| on | 0 | 1 | 0 | 2 | 0 | 1 |
| tall | 1 | 1 | 2 | 0 | 1 | 2 |
| towers | 0 | 0 | 0 | 1 | 0 | 1 |
| two | 1 | 1 | 1 | 2 | 1 | 0 |

   b. Use `torch.svd()` to compute the singular value decompositon of this matrix $X = USV^T$

```
import torch

M = torch.Tensor(
[[0,1,0,1,0,1],
[1,0,1,1,0,1],
[0,1,0,2,0,1],
[1,1,2,0,1,2],
[0,0,0,1,0,1],
[1,1,1,2,1,0]]);
```

```
U, S, V = torch.svd(M)

torch.set_printoptions(precision=2)

print(U)
print(S)
print(V)
```

```
tensor([[-0.30,  0.24,  0.38, -0.36,  0.41,  0.64],
        [-0.37, -0.11, -0.03,  0.80,  0.47,  0.04],
        [-0.41,  0.53,  0.29, -0.12,  0.08, -0.67],
        [-0.56, -0.74,  0.16, -0.27, -0.13, -0.14],
        [-0.32,  0.19,  0.37,  0.36, -0.75,  0.29],
        [-0.50,  0.25, -0.78, -0.13, -0.17,  0.17]])
tensor([4.83, 2.53, 1.70, 1.10, 0.40, 0.11])
tensor([[-0.30, -0.24, -0.38,  0.36,  0.41,  0.64],
        [-0.37,  0.11,  0.03, -0.80,  0.47,  0.04],
        [-0.41, -0.53, -0.29,  0.12,  0.08, -0.67],
        [-0.56,  0.74, -0.16,  0.27, -0.13, -0.14],
        [-0.32,  0.19,  0.37,  0.36,  0.75,  0.29],
        [-0.50, -0.25,  0.78,  0.13, -0.17,  0.17]])
```



(Note: replacing U and V with -U and -V would preserve $X = USV^T$)

c. Extract a word representation from the first two columns of U and use `matplotlib` to plot the words on a 2-dimensional graph.



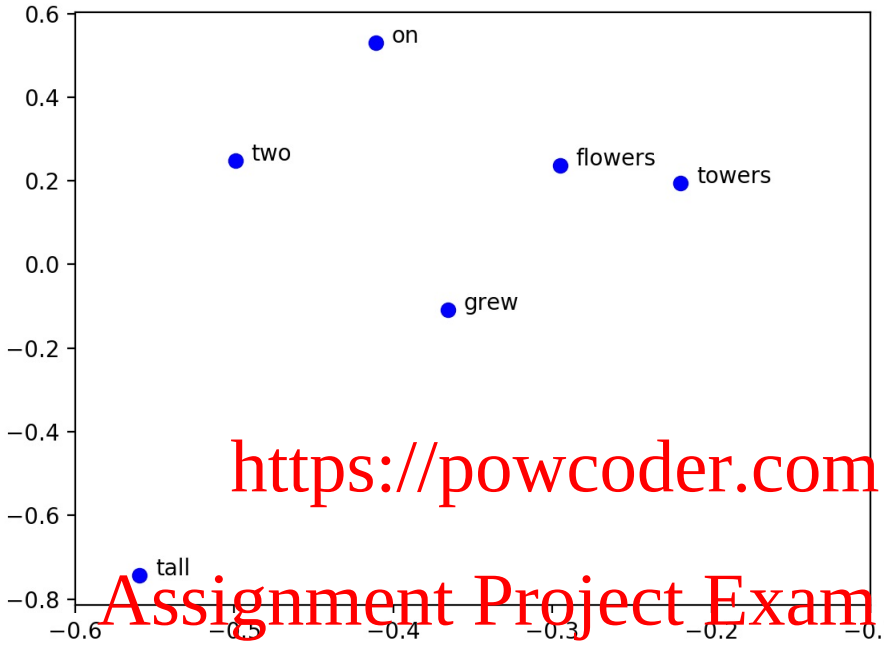
```
import matplotlib.pyplot as plt

Lex = ['flowers','grew','on','tall','towers','two']

plt.scatter(U[:,0],U[:,1],c='B')
plt.xlim([-0.6,-0.1])

for a in range(U.size()[0]):
    plt.text(0.01+U[a,0],U[a,1],Lex[a])

plt.savefig('vectors.png')
plt.show()
```






(Note: the image may be rotated, depending on the sign of U)

---