

ตัวอย่างการคำนวณ Word2Vec

มี 2 วิธีที่สามารถใช้เทรน Word2vec embeddings

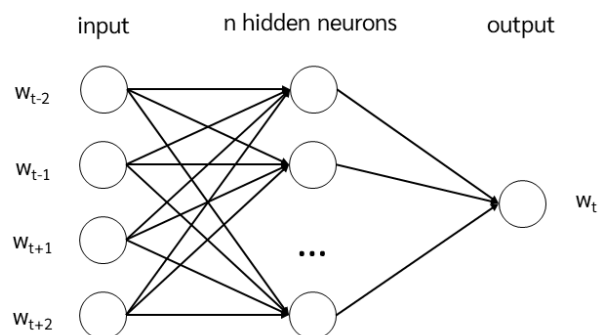
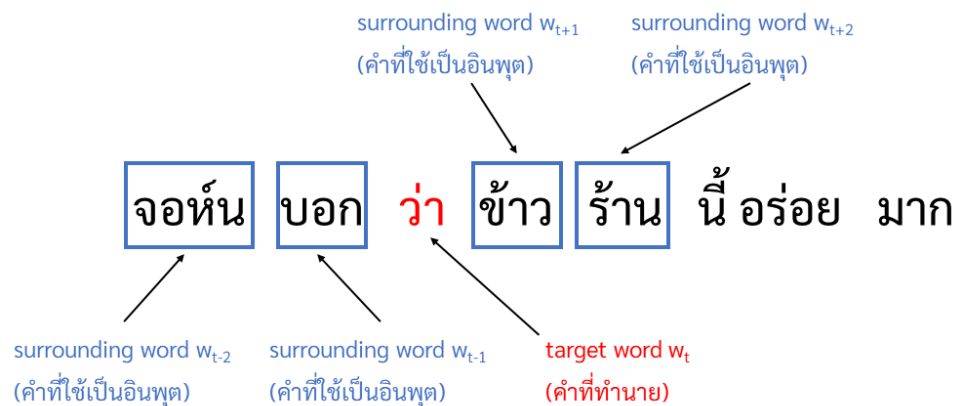
1. continuous bag-of-words (CBOW)
2. skip-gram

1. Continuous bag-of-words (CBOW)

ให้ w_t เป็น target word หรือ center word

ให้ w_{t-2} , w_{t-1} , w_{t+1} , w_{t+2} เป็น context หรือ surrounding words

CBOW ใช้ context หรือ surrounding words เป็น input ตัวอย่างถ้าให้ C (context window) กำหนดให้ $C=5$ จะได้ input เป็นคำในตำแหน่งที่ w_{t-2} , w_{t-1} , w_{t+1} และ w_{t+2} จากนั้นทำนาย target word



“จอห์น บอก ว่า ข้าว ร้าน นี้ อร่อย มาก” กำหนดให้ $C=5$ ก็จะ slide จากซ้ายไปขวา ตามรูปด้านล่าง

สีแดง เป็น target word

สีฟ้า เป็น context words

	w_t	w_{t+1}	w_{t+2}					
#1	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#2	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#3	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#4	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#5	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#6	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#7	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#8	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
					w_{t-2}	w_{t-1}	w_t	

เนื่องจากเราไม่สามารถแปลงข้อความเป็น matrix ได้โดยตรง เราจึงต้องแปลงข้อความด้วย one hot encoding ก่อน

ข้าว	1	0	0	0	0	0	0	0
จอห์น	0	1	0	0	0	0	0	0
นี้	0	0	1	0	0	0	0	0
บอก	0	0	0	1	0	0	0	0
มาก	0	0	0	0	1	0	0	0
ร้าน	0	0	0	0	0	1	0	0
ว่า	0	0	0	0	0	0	1	0
อร่อย	0	0	0	0	0	0	0	1

เมื่อเรา encoding ข้อความ เป็น one hot แล้ว ต่อมาสร้าง feature และ target matrices

#	Feature			
1	บอก	ว่า		
2	จอห์น	ว่า	ข้าว	
3	จอห์น	บอก	ข้าว	ร้าน
4	บอก	ว่า	ร้าน	นี้
5	ว่า	ข้าว	นี้	อร่อย
6	ข้าว	ร้าน	อร่อย	มาก
7	ร้าน	นี้	มาก	
8	นี้	อร่อย		

Target
จอห์น
บอก
ว่า
ข้าว
ร้าน
นี้
อร่อย
มาก

#	Feature							
	ข้าว	จอห์น	นี้	บอก	มาก	ร้าน	ว่า	อร่อย
1	0	0	0	1	0	0	1	0
2	1	1	0	0	0	0	1	0
3	1	1	0	1	0	1	0	0
4	0	0	1	1	0	1	1	0
5	1	0	1	0	0	0	1	1
6	1	0	0	0	1	1	0	1
7	0	0	1	0	1	1	0	0
8	0	0	1	0	0	0	0	1

Target							
ข้าว	จอห์น	นี้	บอก	มาก	ร้าน	ว่า	อร่อย
0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0
1	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0

เมื่อได้ feature และ target matrices มาแล้ว ก็นำไปเทรนด้วย Neural network (1 hidden layers, 3 nodes) หลังจากเทรนไป 1000 iterations ได้ผลลัพธ์ดังนี้

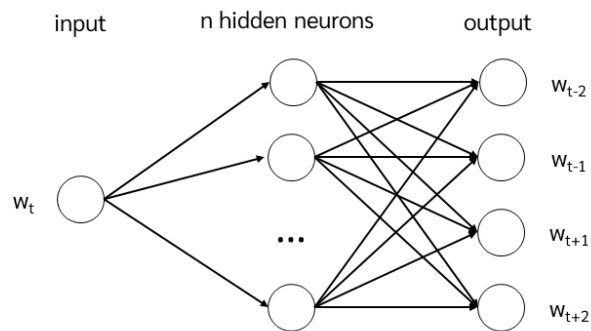
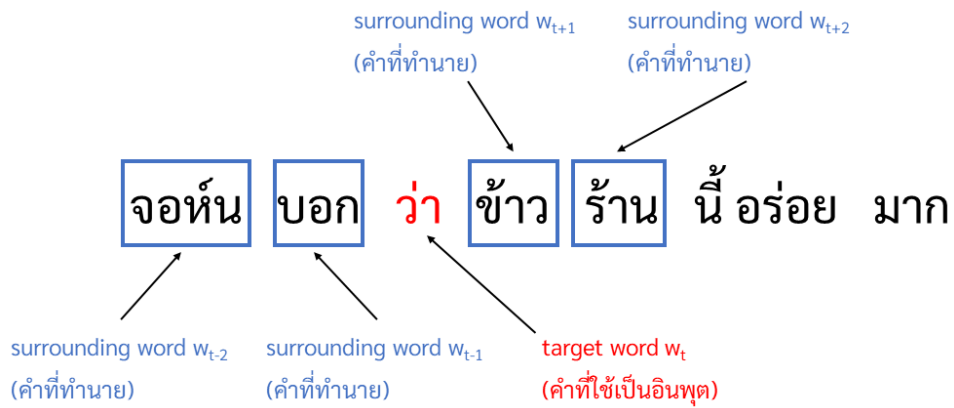
	ข้าว	จอห์น	นี้	บอก	มาก	ร้าน	ว่า	อร่อย
จอห์น	0.0392	0.0961	0.2108	0.0479	0.0429	0.1844	0.2653	0.1133
บอก	0.3193	0.2013	0.0806	0.0063	0.1769	0.0909	0.1035	0.0212
ว่า	0.1661	0.0184	0.1168	0.0032	0.1944	0.1512	0.0127	0.3371
ข้าว	0.0345	0.1903	0.0623	0.112	0.2348	0.0965	0.0482	0.2214
ร้าน	0.0277	0.2135	0.1863	0.007	0.2748	0.0044	0.0723	0.214
นี้	0.1085	0.074	0.1391	0.0612	0.0832	0.0963	0.4125	0.0252
อร่อย	0.1727	0.2398	0.3143	0.0664	0.0015	0.1396	0.0034	0.0623
มาก	0.0989	0.1725	0.0412	0.0753	0.1354	0.285	0.0084	0.1833

2. Skip-gram

ให้ w_t เป็น target word หรือ center word

ให้ w_{t-2} , w_{t-1} , w_{t+1} , w_{t+2} เป็น context หรือ surrounding words

skip-gram จะตรงข้ามกับ CBOW ก็คือจะใช้ target word เป็น input และทำนาย context words



“จอห์น บอกว่า ข้าวร้านนี้ อร่อยมาก” กำหนดให้ $C=5$ ก็จะ slide จากซ้ายไปขวา ตามรูปด้านล่าง

สีแดง เป็น target word

สีฟ้า เป็น context words

	w_t	w_{t+1}	w_{t+2}					
#1	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#2	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#3	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#4	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#5	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#6	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#7	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก
#8	จอห์น	บอก	ว่า	ข้าว	ร้าน	นี้	อร่อย	มาก

เนื่องจากเราไม่สามารถแปลงข้อความเป็น matrix ได้โดยตรง เราจึงต้องแปลงข้อความด้วย one hot encoding ก่อน

ข้าว	1	0	0	0	0	0	0	0
จอห์น	0	1	0	0	0	0	0	0
นี้	0	0	1	0	0	0	0	0
บอก	0	0	0	1	0	0	0	0
มาก	0	0	0	0	1	0	0	0
ร้าน	0	0	0	0	0	1	0	0
ว่า	0	0	0	0	0	0	1	0
อร่อย	0	0	0	0	0	0	0	1

เมื่อเรา encoding ข้อความ เป็น one hot แล้ว ต่อมาสร้าง feature และ target matrices

#	Feature
1	จอห์น
2	บอก
3	ว่า
4	ข้าว
5	ร้าน
6	นี้
7	อร่อย
8	มาก

Target			
บอก	ว่า		
จอห์น	ว่า	ข้าว	
จอห์น	บอก	ข้าว	ร้าน
บอก	ว่า	ร้าน	นี้
ว่า	ข้าว	นี้	อร่อย
ข้าว	ร้าน	อร่อย	มาก
ร้าน	นี้	มาก	
นี้	อร่อย		

#	Feature							
	ข้าว	จอห์น	นี้	บอก	มาก	ร้าน	ว่า	อร่อย
1	0	1	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0
3	0	0	0	0	0	0	1	0
4	1	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0
6	0	0	1	0	0	0	0	0
7	0	0	0	0	0	0	0	1
8	0	0	0	0	1	0	0	0

Target							
ข้าว	จอห์น	นี้	บอก	มาก	ร้าน	ว่า	อร่อย
0	0	0	1	0	0	1	0
1	1	0	0	0	0	1	0
1	1	0	1	0	1	0	0
0	0	1	1	0	1	1	0
1	0	1	0	0	0	1	1
1	0	0	0	1	1	0	1
0	0	1	0	1	1	0	0
0	0	1	0	0	0	0	1

เมื่อได้ feature และ target matrices มาแล้ว ก็นำไปเทรนด้วย Neural network (1 hidden layers, 3 nodes) หลังจากเทรนไป 1000 iterations ได้ผลลัพธ์ดังนี้

	ข้าว	จอห์น	นี้	บอก	มาก	ร้าน	ว่า	อร่อย
จอห์น	0.0587	0.0983	0.279	0.0701	0.3777	0.0385	0.0058	0.072
บอก	0.1617	0.147	0.1355	0.1749	0.0107	0.1864	0.0042	0.1796
ว่า	0.533	0.176	0.0334	0.0986	0.0313	0.0598	0.0076	0.0603
ข้าว	0.1073	0.2535	0.1043	0.1201	0.0673	0.1214	0.1312	0.0948
ร้าน	0.0055	0.0098	0.1339	0.5338	0.0743	0.0655	0.1053	0.0719
นี้	0.0859	0.0309	0.1603	0.0222	0.5098	0.089	0.0692	0.0327
อร่อย	0.0735	0.032	0.2	0.0703	0.1315	0.1055	0.2547	0.1325
มาก	0.0265	0.3224	0.1247	0.085	0.0597	0.0803	0.187	0.1143