

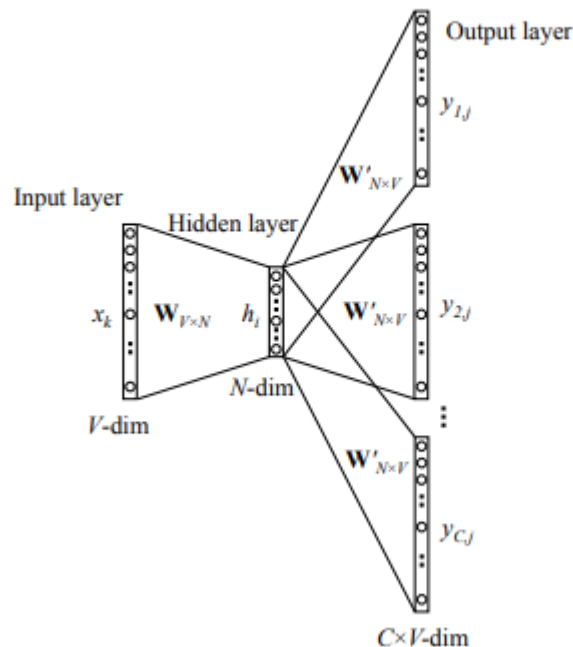
به نام خدا

Word2Vec یک تکنیک معمول است که برای پردازش زبان طبیعی استفاده میشود. در این تکنیک کلمات مشابه word embedding های مشابه دارند.

برای پیاده سازی Word2Vec 2 الگوریتم اصلی وجود دارد: Skip-Gram و Continuous Bag of Words. این الگوریتم ها از مدل های شبکه عصبی استفاده میکنند تا به بردار کلمات دست یابند. هر دو الگوریتم از کلمات نزدیک بهم استفاده میکنند. معنی های کلمات را استخراج کنند و در embedding شان قرار دهند. اگر گروهی از کلمات همیشه در نزدیک کلمات یکسان پیدا شوند embedding مشابه دارند. برای چگونگی نزدیکی یک window-size تنظیم میکنیم. با این window-size مشخص میشود کدام کلمات نزدیک بهم را برمیداریم. برای مثال اگر مقدار آن 2 باشد، برای هر کلمه ما 2 کلمه قبل و 2 کلمه بعد از آن را برمیداریم. در جدول پایین جفت کلمه هایی که ساخته شده اند را میتوانیم مشاهده کنیم. کلمه های لایت شده همانی است که ما برای آن جفت پیدا کرده ایم.

حالا که میدانیم چگونه جفت کلمات را پیدا میکنیم به سراغ چگونگی کار الگوریتم ها میرویم.

### Skip-Gram:

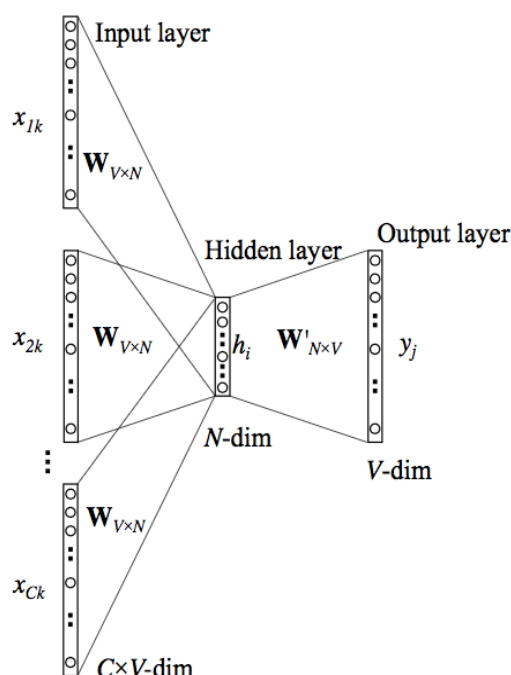


در Skip-Gram تلاش میکنیم context word ها را با استفاده از کلمه اصلی پیش بینی کنیم. برای مثال از جمله the pink horse is eating horse embedding می‌خواهیم کلمه horse را به دست آوریم. اول کلمات در مجموعه نوشته ها را به فرمت one-hot انکود میکنیم. جفت کلمات را از کلمه ای که می‌خواهیم embedding اش را پیدا کنیم برمی‌داریم: (horse, the), (horse, pink), (horse, is), (horse, eating). حالا برای هر یک از این ها از مدل شبکه عصبی بالا استفاده میکنیم. ورودی همان کلمه اصلی با فرمت one-hot است. ماتریس وزن  $W_{V \times N}$  ورودی را به لایه مخفی تبدیل میکند و ماتریس وزن  $W'_{N \times V}$  لایه مخفی را به لایه خروجی تبدیل میکند. خروجی word context به فرمت one-hot میباشد. ما مدل را به ازای هر context word یکبار اجرا خواهیم کرد و مدل اینگونه با تلاش برای پیش بینی context word ها آموزش می‌بیند. زمانی که آموزش کل لغات تمام شد ماتریس  $W_{V \times N}$  همان embedding مورد نظر است.

$V$ : تعداد کلمات در مجموعه نوشته ها

$N$ : اندازه مطلوب برای word embedding

**CBOW:**



در CBOW الگوریتم مشابه الگوریتم قبلی است ولی عمل مخالف Skip-Gram را انجام میدهد. در اینجا می‌خواهیم مدلمان کلمه اصلی را از context word ها پیش بینی کند. لایه ورودی context word با فرمت

one-hot میباشد. برای هر context word نتیجه لایه مخفی از ماتریس وزن  $W_{V \times N}$  را میگیریم سپس از تمام این نتیجه ها میانگین میگیریم و آن را در یک لایه مخفی قرار میدهیم که این لایه مخفی به لایه خروجی پاس داده میشود. مانند قبل بعد از آموزش، ماتریس وزن  $W_{V \times N}$  همان word embedding مورد نظر است.

مزایا و معایب:

1- مزیت cbow و

عیب skip-gram:

برای cbow همگرایی سریعتری از skip-gram داریم. برای حالت های مختلف در مقاله [Mikolov et al. 2013](#) تعداد روز های مورد نیاز برای آموزش هر دو مدل نشان داده شده است.

Model	Vector Dimensionality	Training words	Accuracy [%]			Training time [days]
			Semantic	Syntactic	Total	
3 epoch CBOW	300	783M	15.5	53.1	36.1	1
3 epoch Skip-gram	300	783M	50.0	55.9	53.3	3
1 epoch CBOW	300	783M	13.8	49.9	33.6	0.3
1 epoch CBOW	300	1.6B	16.1	52.6	36.1	0.6
1 epoch CBOW	600	783M	15.4	53.3	36.2	0.7
1 epoch Skip-gram	300	783M	45.6	52.2	49.2	1
1 epoch Skip-gram	300	1.6B	52.2	55.1	53.8	2
1 epoch Skip-gram	600	783M	56.7	54.5	55.5	2.5

2- Cbow روابط نحوی بین کلمات را بهتر از skip-gram یاد میگیرد درحالیکه skip-gram روابط معنایی را بهتر از cbow یاد میگیرد. برای مثال cbow برای دو کلمه "cat" و "cats" دو بردار word embedding مشابه میسازد درحالیکه skip-gram دو کلمه "cat" و "dog" را نزدیک بهم میبیند.

در این مثال مزیت و عیب هر دو با مقایسه هم گفته شد

Model Architecture	Semantic-Syntactic Word Relationship test set		MSR Word Relatedness Test Set [20]
	Semantic Accuracy [%]	Syntactic Accuracy [%]	
RNNLM	9	36	35
NNLM	23	53	47
CBOW	24	64	61
Skip-gram	55	59	56

3- Skip-gram کلمات و عبارات نادر (آن هایی که به دفعات کم در نوشته ها ظاهر شده اند) را به خوبی بازنمایی میکند درحالیکه cbow دقت بهتری در کلمات مکرر (آن هایی که به دفعات زیاد در نوشته ظاهر شده اند) دارد. چرا؟ همانطور که میدانیم cbow احتمال کلمه هدف را با نگاه کردن به context ماکسیمم میکند و این یک مشکل برای کلمات نادر است.

برای مثال با context: yesterday was a really [...] day مدل cbow به ما میگوید این کلمه با بیشترین احتمال beautiful یا nice میباشد. کلماتی مانند delightful توجه کمتری از مدل را به خود جلب میکنند چون این مدل برای پیش بینی محتمل ترین کلمه طراحی شده است. کلمه delightful با تعداد زیادی مثال با کلمات مکرر محو خواهد شد.

از سوی دیگر، skip-gram برای پیش بینی context طراحی شده است. کلمه delightful داده میشود و مدل باید با احتمال بزرگی بگوید context جمله yesterday was a really [...] day یا جملات مرتبط دیگر میباشد. در skip-gram کلمه delightful تلاش نمیکند با کلمه beautiful رقابت کند و به جای آن با delightful+context به عنوان یک مشاهده جدید برخورد میشود.

در این مثال مزیت و عیب هر دو با مقایسه هم گفته شد