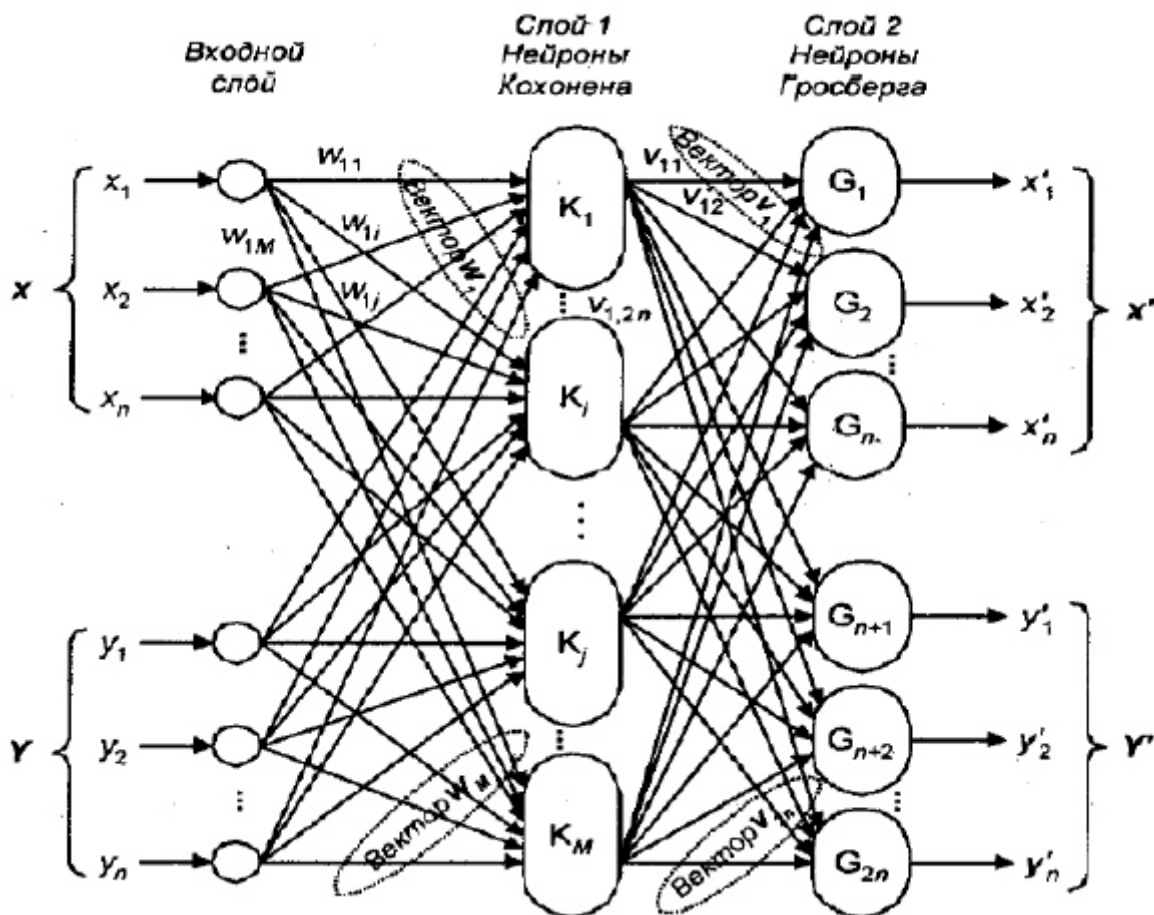


Искусственные нейронные сети, ЛР №6

Что такое слой Кохонена?

Сеть Кохонена представляет собой специальный тип нейронной сети для решения задачи кластеризации.



Слой Кохонена (1) работает таким образом, что для данного входного вектора один и только один нейрон Кохонена выдаёт на выходе логическую единицу, все остальные - ноль. Ассоциированное с каждым нейроном Кохонена множество весов соединяет его с каждым входом. На (1) нейрон Кохонена K_1 имеет веса $w_{11}, w_{21}, \dots, w_{m1}$, составляющие весовой вектор W_1 . Они соединяются с входными сигналами x_1, x_2, \dots, x_m , составляющих входной вектор X . Выход каждого нейрона является суммой взвешенных входов:

$$N_j = w_{1j}x_1 + w_{2j}x_2 + \dots + w_{mj}x_m$$

Нейрон с максимальным значением N_j является победителем. Его выход равен единице, у остальных он равен нулю.

Слой Кохонена классифицирует входные векторы в группы схожих векторов, что достигается с помощью такой подстройки весов слоя Кохонена, что близкие входные

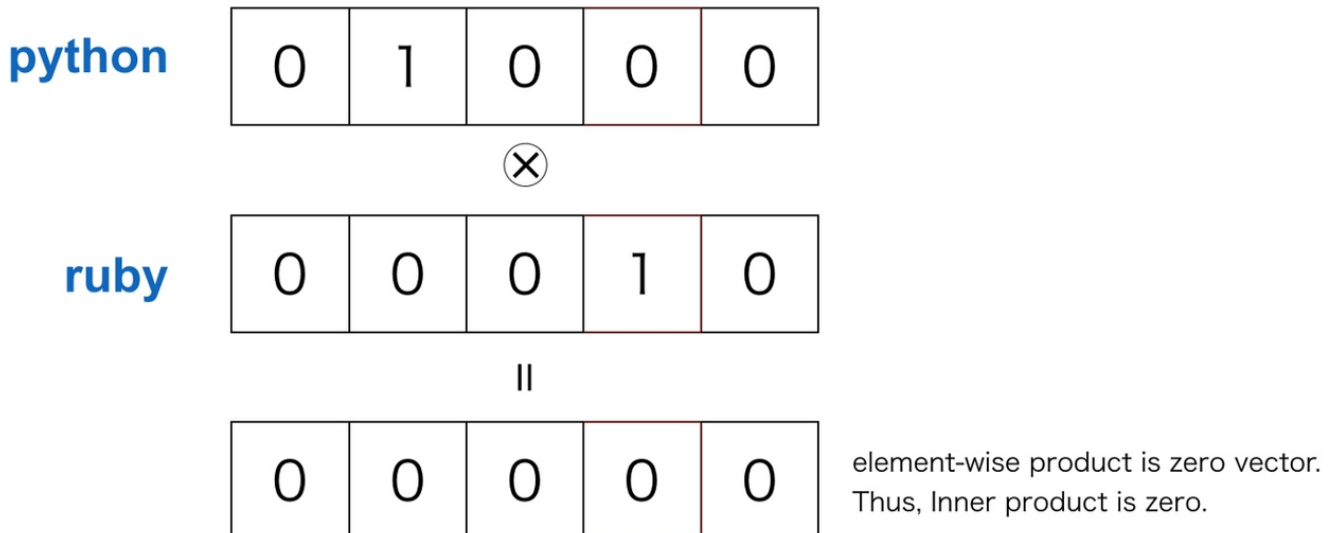
вектора активируют один и тот же нейрон данного слоя. Обучение Кохонена является самообучением, протекающим без учителя.

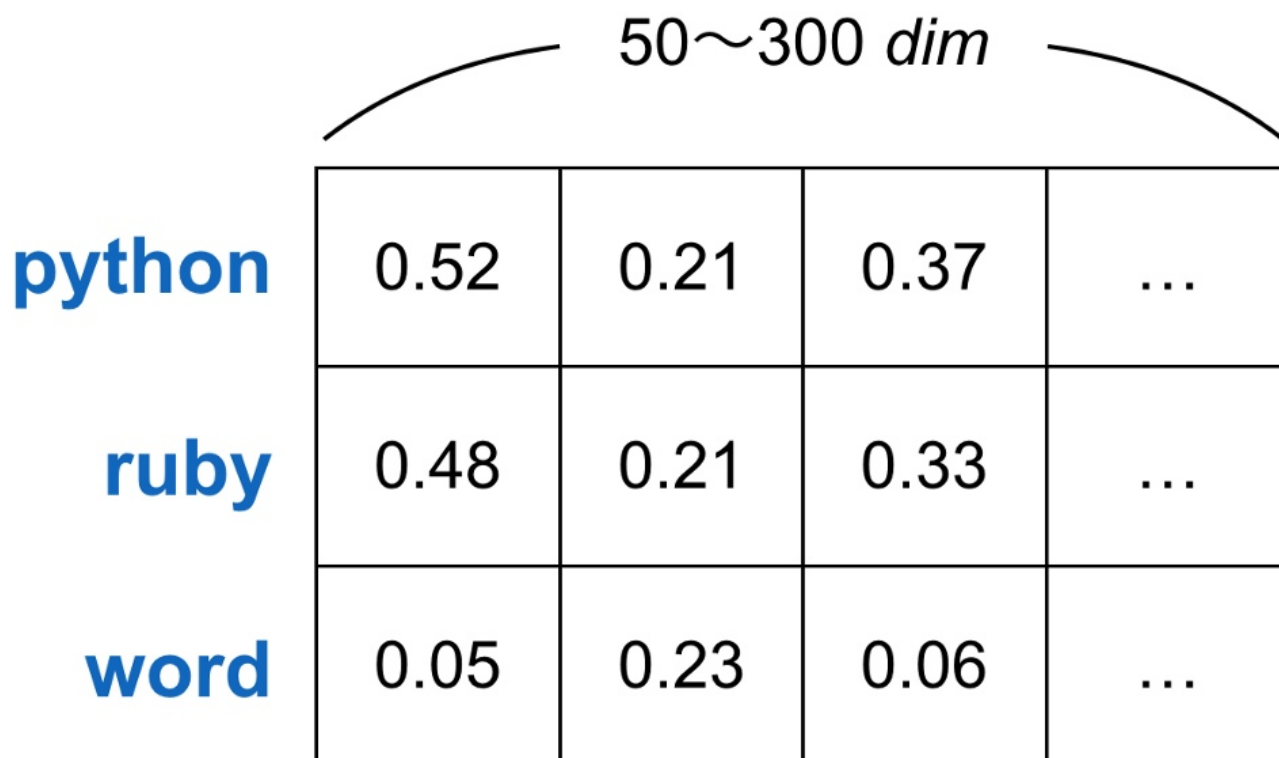
Уравнение описывающее процесс обучения w_n - новое значение веса, w_c - старое, α - коэффициент скорости обучения, варьирующийся в процессе обучения:

$$w_n = w_c + \alpha(x - w_c)$$

Для чего нужен слой Embedding?

Он предоставляет распределенное представление слов (Distributed Representation of Words), которое может быть значительно больше, чем one-hot representation на картинке 1. Например, можно вычислить сходство между словами по векторной операции. Глядя на вектор на картинке 2, сходство между «python» и «ruby» кажется выше, чем сходство между «python» и «word». Также, даже если количество слов увеличивается, мы не увеличиваем количество измерений каждого слова.





	50~300 <i>dim</i>			
python	0.52	0.21	0.37	...
ruby	0.48	0.21	0.33	...
word	0.05	0.23	0.06	...

То есть Embedding кодирует информацию, уменьшая влияние шума или несущественных элементов на входе. Это также известно как сжатие при обработке сигналов. Поскольку всегда будут присутствовать нерелевантные детали, сжатие или внедрения помогают отфильтровывать такие нерелевантные представления.

Почему выбран размер пакета 400?

Просто пробовал разные варианты, потому что меньшие размеры батча имеют шум, предоставляя эффект регуляризации и меньшую ошибку обобщения, таким образом пытался избавиться от переобучения в дефолтной конфигурации сети. Особого профита в итоге не получил - оставил последнее использованное значение.