**Paper Link**

[GloVe: Global Vectors for Word Representation](https://nlp.stanford.edu/pubs/glove.pdf)

**Video Summary**

[GloVe](https://www.youtube.com/watch?v=InCWrgrUJT8)

Global vectors = Glove

**Pre-requisite papers (Roots)**

[Word2Vec](https://docs.google.com/document/d/1VVZN0tSOfzoSJrJvK-9jpBdnnzaJeK-wdX-7UnpGbWQ/edit)

[LSA](https://www.youtube.com/watch?v=bzNch-dBCN8)

LSA reconstructs word co-occurences using singular value decomposition

Matrix factorization methods for embeddings are statistically efficient, but have an inoptimal sub-vector representation space

Skip-gram models (using a sliding window) poorly utilize the corpus statistics since they train on separate local context windows instead of global co-occurence count

Global co-occurence means that word relationships can be captured across different examples (e.g human and computer in doc 1, computer and user in doc 2 ⇒ relationship between human and user)

* LSA does this through SVD and slicing some of the columns, so that words which occur in the same context (which may happen in different examples) are considered similar by the model

In skip gram models, the word's context is predicted given the word itself

Skip-gram model showed that linguistic patterns could be captured by linear transformations of vector representations

Window-based methods scan context windows across the corpus, failing to account for repetition in data

**Notes**

The global corpus statistics are captured directly by GloVe

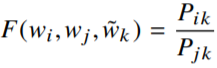
Let Xij be # of times word j appears in context of word i,

be the number of times any word appears in the context of word i, and Pij = P(j | i) = Xij / Xi be the probability that word j appears in the context of word i

Consider 2 words i and j; for words k related to i, but not j, the ratio Pik / Pjk should be large and Pjk / Pik should be small

If both i and j are not related to a word k, then the ratio should be 1 ⇒ these facts suggest using ratios of co-occurence probabilities rather than the probabilities themselves

The general model takes on the form:



We want to impose some conditions on F: encode the info present in the ratio in the word vector space, so we can do this with vector differences since vector spaces are linear

We restrict f to rely only on the difference between two target words:



Since the inputs to F are vectors, and F must produce a scalar, in order to maintain our property of linearity, we instead input the dot product to F

which prevents F from using nonlinear/undesirable ways to transform the vector dimensions

Note that F is a parametrized neural network so what inputs we feed in is importnat

The distinction between a context word and a word is arbitrary, and the two may be interchanged, so we must make our function invariant under relabeling which produces:



This would exhibit exchange symmetry if not for the Xi, but since this term is independent of k it can be expressed as a bias bi corresponding wi ; now we just add another bias bk to restore symmetry (so that the output is the same regardless of input position)

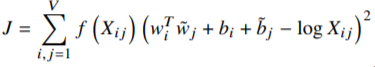


Perform an additive shift such that the right hand side becomes log(1 + Xik) since log diverges when the input approaches 0

This maintains the sparsity of X since its entries can be 0

Factorizing the log of the co-occurence matrix is similar to LSA; one issue is that the model weights co-occurences unequally, even the rare and noisy ones

To address this, a new weighted least squares problem with a weighting function f(Xij) yields:

where V is the vocab size

The weighting function must adhere to the following:

* f(0) = 0
* f(x) should be non-decreasing so co-occurences are not overweighted
* f(x) should be small for large values of x so that frequent co-occurences are not overweighted
* Basically, it should change the relative weighting of how much different co-occurences count towards the loss

Notice that we have created a function F (from earlier) which is invariant under relabeling, so that if you swap i and j, the same loss occurs (since Xij = Xji and same with wi^twj + bi + bj when i and j subscripts are swapped)



**Branches**