Attention with a worked out example

We want the word `walk` to be more closely associated to `birds` in this specific text:

`Birds sometimes walk`

We want to do this because there are other examples in text:

`Sometimes for walking birds use floppy feet...`

`OR But birds don't just walk they fly...`

Lets say we have a poorly trained Value vector for **V**:

$$V = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ -0.8 & 0.3 & -0.1 \\ 0.5 & -0.1 & -0.4 \end{bmatrix}$$

where each token has a representation like:

$$v1(Birds) - > [0.1, 0.2, 0.3]$$
$$v2(sometimes) - > [-0.8, 0.3, -0.1]$$
$$v3(Walk) - > [0.5, -0.1, -0.4]$$

and say you have a reasonably good attention **A** matrix trained

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0.1 & 0.9 & 0 \\ 0.6 & 0.1 & 0.4 \end{bmatrix}$$

Initially the dot product of the representations is pretty low:

$$\vec{v3} \cdot \vec{v1} = 0.1 * 0.5 + 0.2 * -0.1 + 0.3 * -0.4 = 0.05 - 0.02 - 0.12 = \textbf{-0.09}$$

Now we want `walk` to pay more attention to `bird` and `sometimes` from a learnt **A**

$$\hat{v_3} = \begin{bmatrix} a_{3,1} * v_{11} + a_{3,2} * v_{21} + a_{3,3} * v_{31} \\ a_{3,1} * v_{12} + a_{3,2} * v_{22} + a_{3,3} * v_{32} \\ a_{3,1} * v_{13} + a_{3,2} * v_{23} + a_{3,3} * v_{33} \end{bmatrix}$$

$$\hat{v_3} = \begin{bmatrix} a_{31} * v_1 1 + a_{32} * v_{21} + a_{33} * v_{31} \quad \text{(Attention paid by `walk` to `bird` and `sometimes` *across* first colu} \\ a_{31} * v_1 2 + a_{32} * v_{22} + a_{33} * v_{32} \quad \text{(Same thing but across the second dimension)} \\ a_{31} * v_1 3 + a_{32} * v_{23} + a_{33} * v_{33} \quad \text{(... And the third dimension)} \end{bmatrix}$$

$$\hat{v_3} = \begin{bmatrix} 0.6 * 0.1 + 0.1 * -0.8 + 0.4 * 0.5 \\ 0.6 * 0.2 + 0.1 * 0.3 + 0.4 * -0.1 \\ 0.6 * 0.3 + 0.1 * -0.1 + 0.4 * -0.4 \end{bmatrix}$$

$$\hat{v_3} = \begin{bmatrix} 0.06 - 0.08 + 0.2 \\ 0.12 + 0.03 - 0.04 \\ 0.18 - 0.01 - 0.16 \end{bmatrix}$$

It would seem that if we have learnt `A` well, i.e. `bird` is paying more attention to `walk`, then it seems to be now closer to `walk` in the embedded space.

$$\hat{v_3} = \begin{bmatrix} 0.18 \\ 0.11 \\ 0.01 \end{bmatrix}$$

Now:

$$\hat{v_3} \cdot v_3 = 0.18 * 0.1 + 0.11 * 0.2 + 0.01 * 0.3 = 0.018 + 0.022 + 0.003 = \mathbf{0.043}$$

i.e. a better association between `walk` and `bird` in the embedded space.

Thoughts on the Mathematics of this

1. I *think*, we are saying that the word `walk` is a **weighted** combination of all the other word embeddings. And therefore I would conjecture that this may not be enough, linear functions are good but nonlinear are better so `V` will also be forward propagated to an MLP to learn any non linearity that's left.
2. Also notice that the final vector are three terms and over `n-grams` of differing sequence length. So perhaps the vector v_3 terms are these piece-wise, auto regressive collection of terms each is a linear function. This kind of reminds me of interaction terms in linear regression and how indicator variables are used to capture non-linear relationships. Is this thought too clever by half in how I think about these things?

Thoughts on how does `bird` get attented to by `flying` in the same text where its associated with `walking`?

We might have other examples in the text of `bird` being associated with the words `fly` too in the text. How could the network learn about it and predict it? I think the answer is `capacity` which loosely translates to the number of parameters in the network we can use. Lets say we have a sentence:

`But birds are good at ____.`

And the word we likely want learnt is `flying`.

Capacity here would mean the opportunities for the word `flying` to learn the word association with the word `bird` and possibly `but` and `good`. Well if we consider just one attention head

1. There are `n_head` parameters for each vector in **V** opportunities to learn these associations. Is there a "partition" of dimension of learn't `v_i` that is different for `walk` and `flying`?
2. Could it be that `walk` and `fly` i.e. the $a_{i,j}$ for ith token – which are learnt via Q,K – are different for their preceeding phrases `"birds sometimes ____"` and `"birds are good at ____"`?
3. In the Query matrix which has a `n_head` size vector Query for `flying`. Lets call this row `q_k`.
   - And Since we compute the attention row `a_i` as a product of `q_k` and `V` we have `n_head`*`n_head` parameters, which is typically `48,000` different parameters just for whatever `flying` needs to attend to! A big space indeed.
4. Not to mention more parameters in MLP and other heads that get learnt later.

Here is some matrix math for this stuff

$$A = \begin{bmatrix} . & . & . \\ . & a_{i,j} & . \\ . & . & . \end{bmatrix} V = \begin{bmatrix} . & v_0 & . \\ . & v_1 & . \\ . & v_2 & . \end{bmatrix} \hat{V} = A.V$$

where V are the old embeddings and A are the Attention dot product.

$$\hat{v_{i,j}} = \sum_{k=1}^{d_k} a_{i,k} v_{k,j}$$

$$\underline{\hat{v_i}} = \sum_{j=1}^{d_k} a_{i,j} * \underline{v_j}$$

So we are *interacting* previously independent $\underline{v_j}s$ across the embedding.
