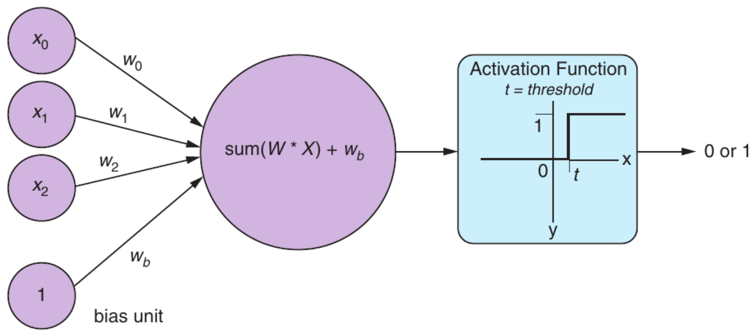
COMP 474/6741 Intelligent Systems (Winter 2021)

Worksheet #9: Neural Networks & Word Embeddings

Task 1. Word analogy questions often appear on standardized tests, like the SSAT, to test language aptitude and reasoning. Here's a simple one (fill in the blank): *Japan is to Sushi what Germany is to* .... Can we solve this type of question with an AI? Stay tuned for the answer!

Task 2. Calculate your first neuron activation for the *Perceptron* (only 100 billion−1 more to go!):



Your input vector $\vec{x} = [0, 1, 1]$ and your weights are $\vec{w} = [0.25, 0.5, 0.75]$. Activation function:

$$f(\vec{x}) = \begin{cases} 1, & \text{if } \vec{x} \cdot \vec{w} \geq \text{threshold} \\ 0, & \text{otherwise} \end{cases}$$

(use a threshold of 0.5):
$f(\vec{x}) =$ ....

Task 3. Let's train our Perceptron to learn the logical *and* function. Here, we have a two-dimensional input vector and four labeled training examples $l_0, \ldots, l_3$:

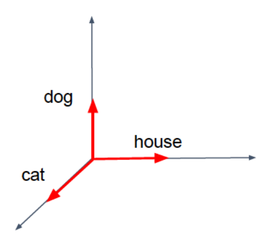
	x_0	x_1	$x_0 \wedge x_1$
l_0	1	1	1
l_1	1	0	0
l_2	0	1	0
l_3	0	0	0

Epoch	Input	w_0	w_1	w_2	$f(\vec{x})$	ok?
0	l_0	0	0	0		
	l_1					
	l_2					
	l_3					
1	l_0					
	l_1					
	l_2					
	l_3					

Note that x_2 is our *bias* (input always 1). Use a threshold for the activation function of 0.5 and a learning rate $\eta = 0.1$. Train the Perceptron by checking the output for each training sample. Update the weights if there is an error: $w'_i = w_i + \eta \cdot (label - predicted) \cdot x_i$.

Task 4. Here are three words in one-hot vector representation (three words, so three dimensions):

'cat' = [0, 0, 1]
'dog' = [0, 1, 0]
'house' = [1, 0, 0]



What is the *distance* between the one-hot word vectors for (cat, dog) and (cat, house):

....

Using the Euclidian distance, $d(\vec{p}, \vec{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$

Task 5. Ok, now re-write the question from Task 1 in form of a word vector calculation:

....

Task 6. Compute the *softmax* function σ on the vector v below:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

$$v = \begin{bmatrix} 0.5 \\ 0.9 \\ 0.2 \end{bmatrix} \qquad \sigma(v) = \begin{bmatrix} \ldots\ldots \\ \ldots\ldots \\ \ldots\ldots \end{bmatrix}$$