- last time we had a weighted sum from the input layer activations + the weights on the connection from input layer to the one neuron that corresponds to an edge.

$$= w_1 a_1 + w_2 a_2 \cdots \cdots w_n + a_n \quad \rightarrow \text{ can be any number}$$

- but we want 0-1 (normalization)

Ex

a common way is sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$:

- now all numbers given as input ie a_6 will be a number 0-1 $\sigma(w_1 a_1 + w_2 a_2 \cdots w_n a_n)$

- So the activation of neuron in second layer

i) how positive or how large for (0-1) is the weighted sum after inputted to sigmoid func:

- BUT maybe you want that neuron to light up when sum is >10 and not >0.9 for ex then we need some bias for activation so we can just add some neg number before plugging into sigmoid

Ex $\sigma(w_1 a_1 + w_2 a_2 \cdots w_n a_n \underline{-10})$, bias

※ weights tell you what pixel patterns the neuron in second layer is picking up on

※ Bias tells you how high weighted sum needs to be before the neuron starts getting meaningfully active and can count as a edge