# 1, Activations (Basic Properties)

## Outline of this video

- What are activations
  - ▼ Reasoning behind non-linear differential activations

    `differential activations` >> take derivative of them >>means, they have gradient, take the gradient
  - ▼ Why non linear?

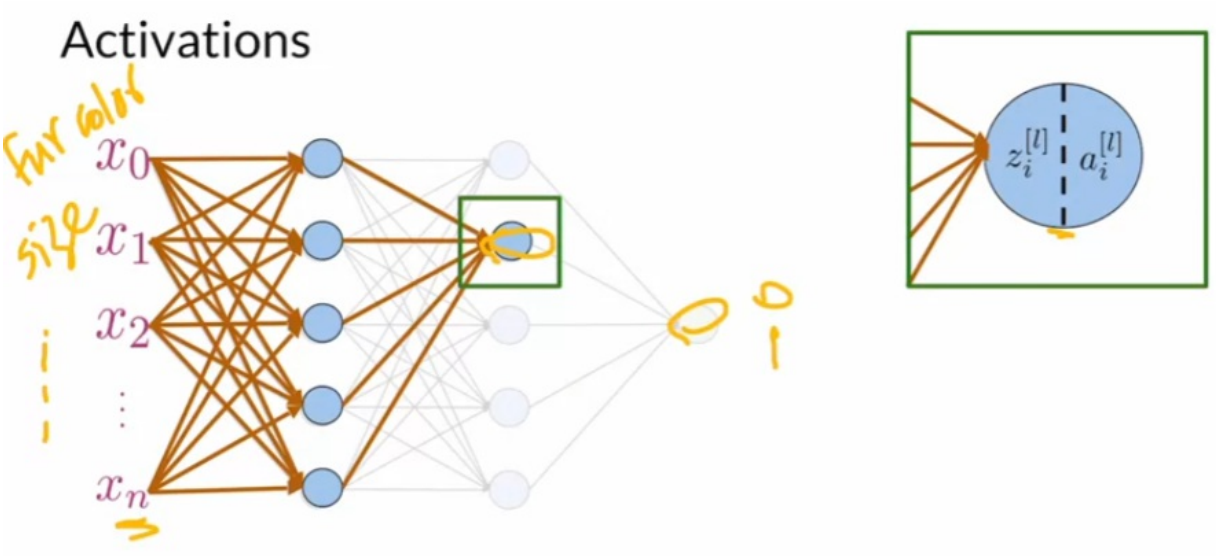
    bcz your NN will be complex, if you just take linear layer, then it will collasp

## Notes

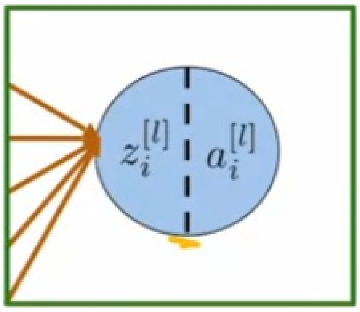
- ▼ what is activation (activation function)?

  > Activations are functions that take any real number as input, also known as its domain, and outputs a number in a certain range using a non-linear differentiable function.

- ▼ why need? if not?
- ▼ what is those nodes in the NN ?
  - ▼ overview

    there are 5 nodes here and map to 1 node at next layer
  - ▼ zoom in (just individual single node)

  - ▼ zilz^{l}_izil

$$z_i^{[l]} = \sum_{i=0} W_i^{[l]} a_i^{[l-1]}$$

> Z is the outputs of the previous layer (ail−1a^{l-1}_iail−1), weighted by these values W, and this is typically called a linear layer.

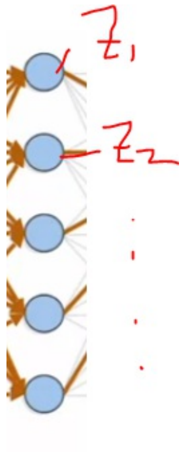
>> bbb included in in this WWW

- typically is written like this (with bias (bbb))

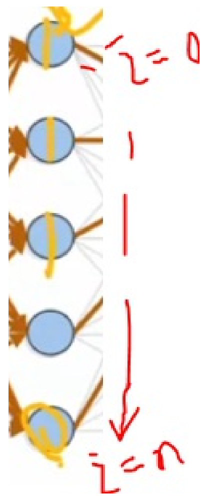
$$s = \sum_{i=1}^{n} w_i a_i + b$$

>> but since bbb can be included in www, so this equation for the node is not used.

- iii >> represents which node it is

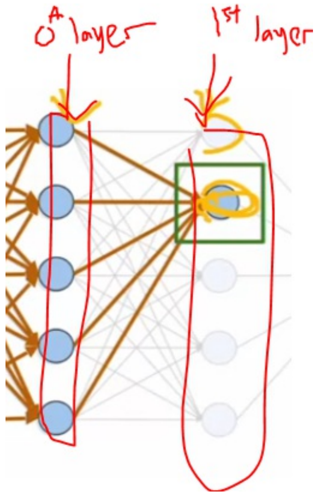



- ∑i=0\sum_{i=0}∑i=0




> It looks at all of those different values, where i equals zero all the way up to i equals n down here.

- lll >> layer




- ▼ ziz_izi

  node with info of sequence of node of particular layer, but without info of which layer it belongs

- ▼ ail−1a^{l-1}_iail−1 >> output from the previous layer

- ▼ aila^{l}_iail >> output (current layer)

$$a_i^{[l]} = g^{[l]}(z_i^{[l]})$$

  ggg>> activation function that take in the value of zee (zilz^{l}_izil)

aia_iai

> A node here takes in information from the previous layer and predicts two things, which I'm going to split with this dotted line. The first here is z_i and the other one is a_i. The bracket l notation up here just says which layer it is. Here maybe it's the second layer. Bracket l would be the second layer like that, and the i here is just which node. That looks like it's 0, 1, it's the first node. First, it computes z, where i represents which node it is. Here it's 0, 1, it's the first node, so i equals one and l represents the layer. Here is the zeroth layer, and here it's in the first layer.

## Vocab

- ▼ differentiable>>take derivative of them

  means, they have gradient, take the gradient

## Summary

- ▼ Activation functions are non-linear and differentiable

  > You could use any custom function as an activation for your deep learning models as long as

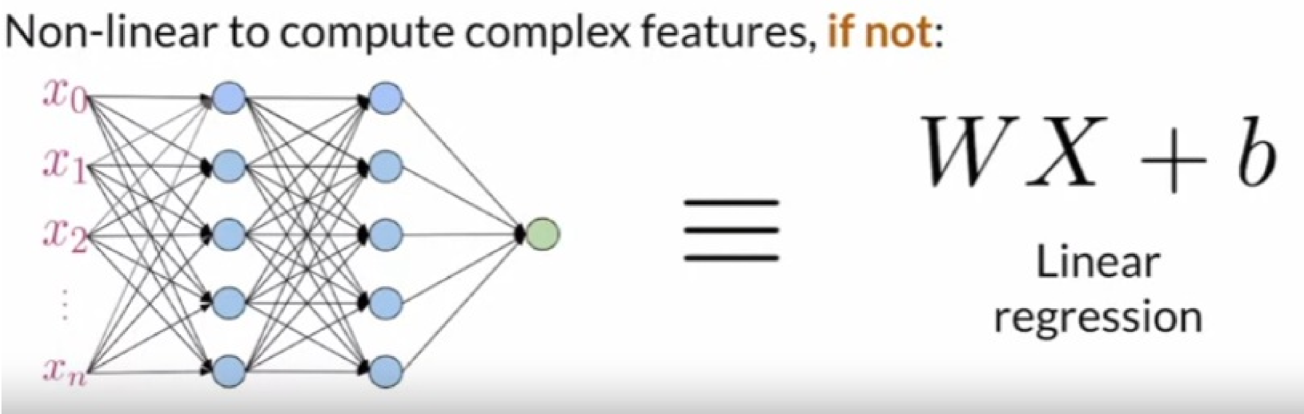
> you make sure that it's non-linear and differentiable.

$$a_i^{[l]} = \boxed{g^{[l]}}(z_i^{[l]})$$

Differentiable non-linear function

- Differentiable for backpropagation
  - ▼ Non-linear to approximate complex functions

> It also needs to be non-linear. The features you compute within the neural networks can be complex. **If you didn't use non-linear activations**, a neural network like this one with multiple hidden layers and neurons could actually be collapsed into a simple linear regression.




linear regression, just stack on each other, not complex

>> which we dont want this to happen

▼ zilz^{l}_izil

$$z_i^{[l]} = \sum_{i=0} W_i^{[l]} a_i^{[l-1]}$$

> Z is the outputs of the previous layer (ail−1a^{l-1}_iail−1), weighted by these values W, and this is typically called a linear layer.

>> bbb included in in this WWW

▼ typically is written like this (with bias (bbb))

$$s = \sum_{i=1}^{n} w_i a_i + b$$

>> but since bbb can be included in www, so this equation for the node is not used.

▼ aila^{l}_iail >> output (current layer)

$$a_i^{[l]} = g^{[l]}(z_i^{[l]})$$

ggg>> activation function that take in the value of zee (zilz^{l}_izil)

## URL

Activation Functions Explained - GELU, SELU, ELU, ReLU and more

During the calculations of the values for activations in each layer, we use an activation function right before deciding what exactly the activation value should be. From the previous activations, weights and biases in each layer, we calculate a value for every activation in the next layer.

https://mlfromscratch.com/activation-functions-explained/#/