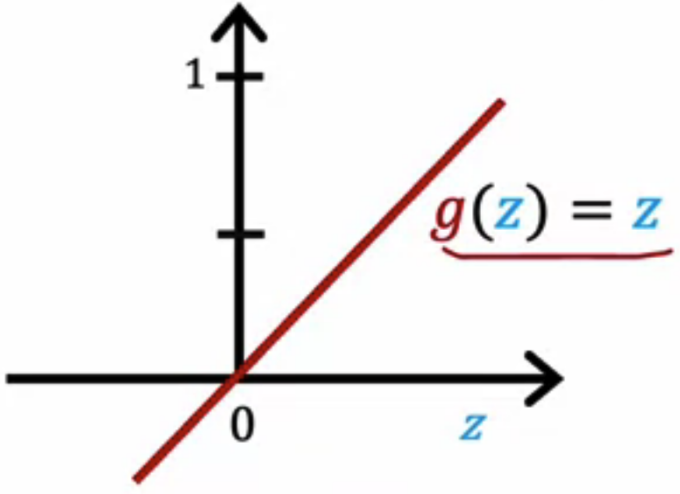
Examples of Activation Functions

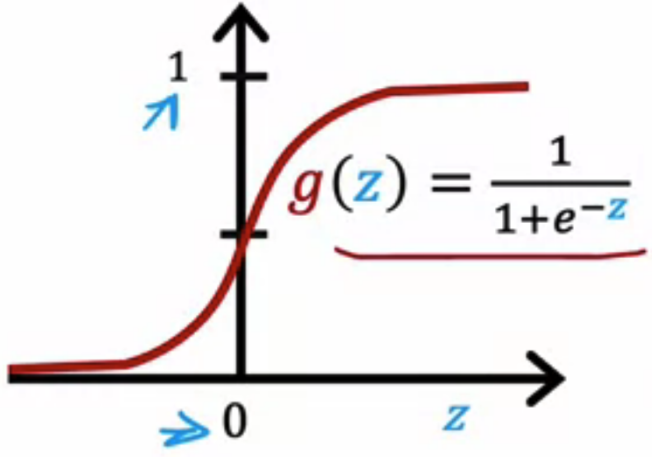
$$a_2^{[1]} = g(\underbrace{\vec{w}_2^{[1]} \cdot \vec{x} + b_2^{[1]}}_{z})$$

Linear activation function



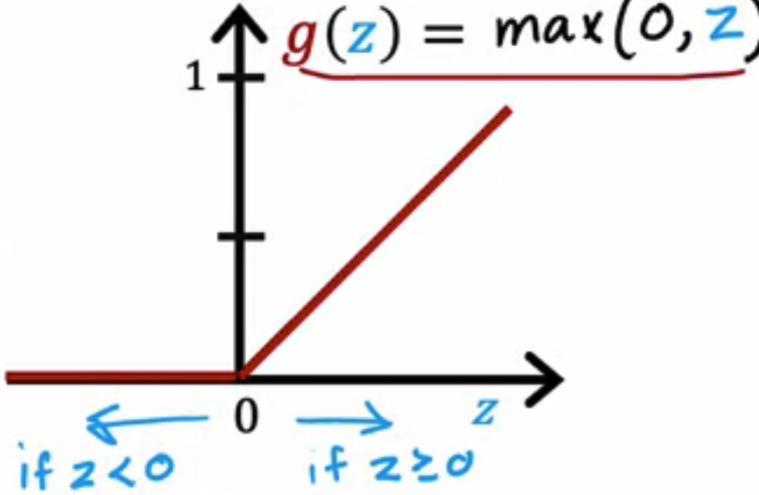
$a = g(z) = \underbrace{\vec{w} \cdot \vec{x} + b}_{z}$

Sigmoid



$0 < g(z) < 1$

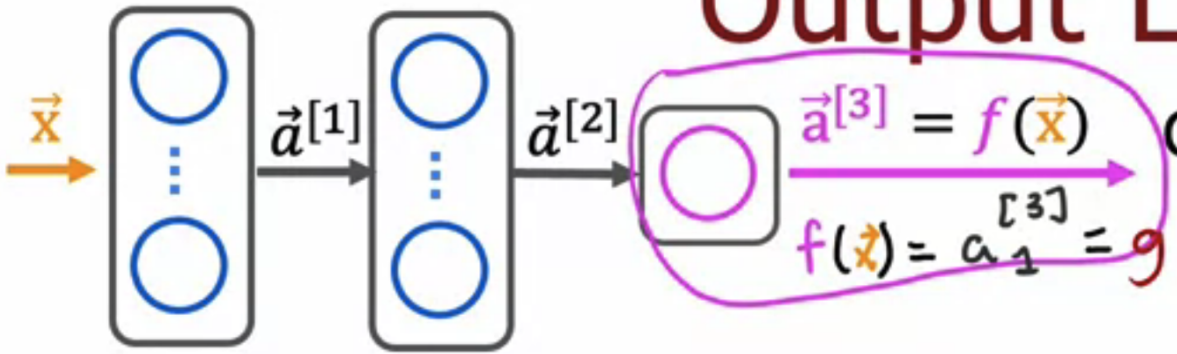
ReLU Rectified Linear Unit



if $z < 0$, $g(z)$ is 0

if $z \geq 0$, $g(z)$ is z

Output Layer

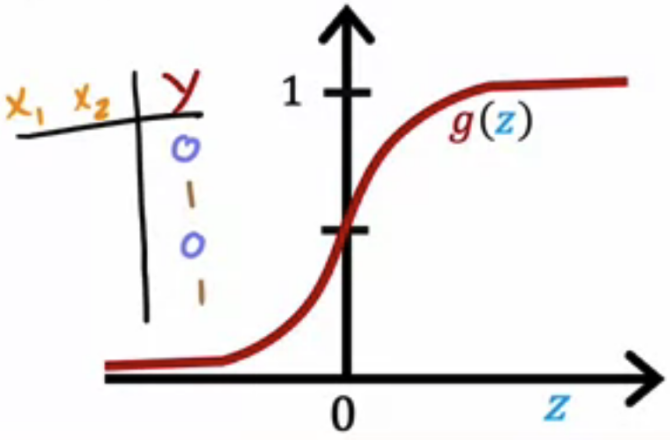


Choosing $g(z)$ for output layer?

Binary classification

Sigmoid

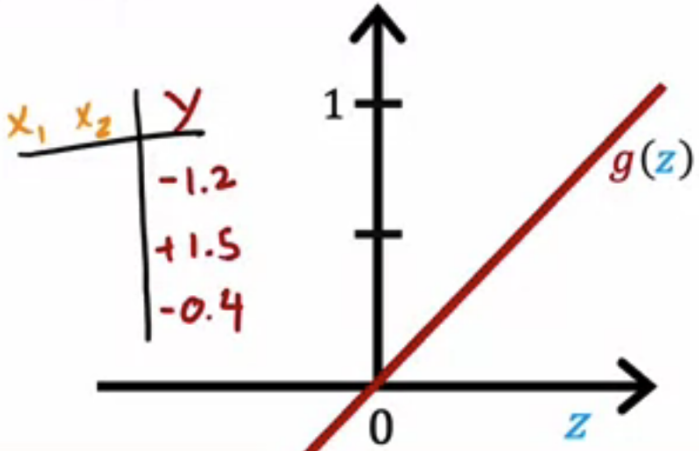
y=0/1



Regression

Linear activation function

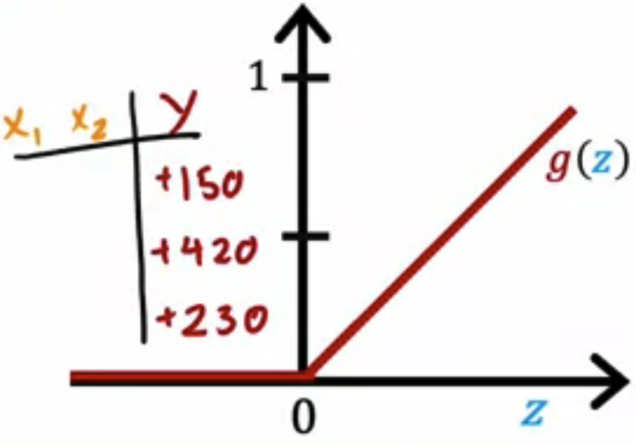
y = +/-



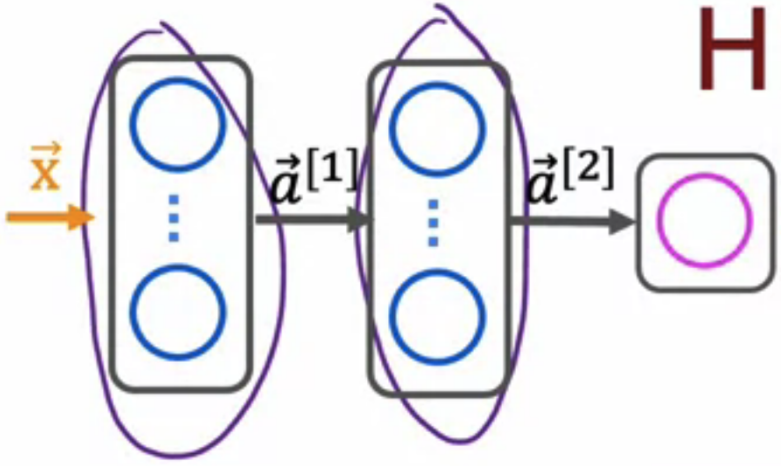
Regression

ReLU

y = 0 or +



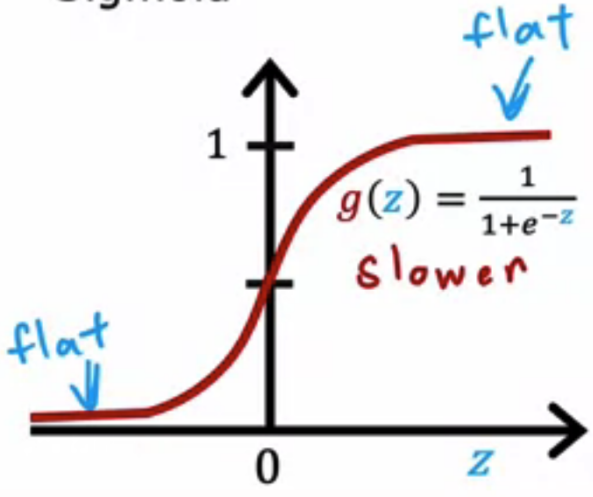
Hidden Layer

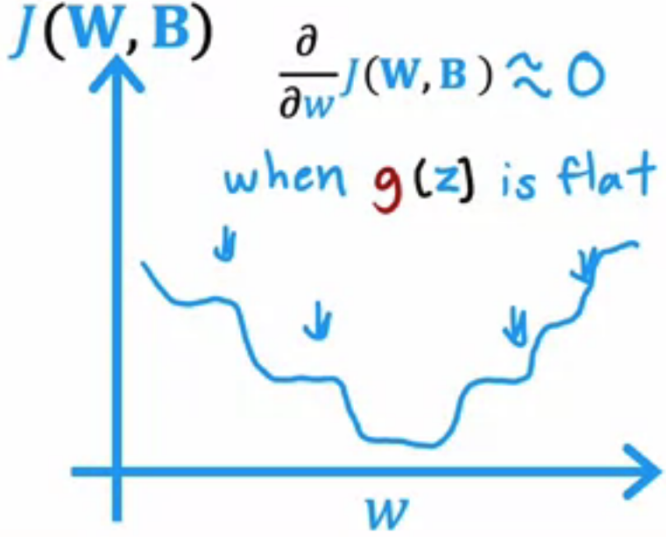


$\vec{a}^{[3]} = f(\vec{x})$

Choosing $g(z)$ for hidden layer

Sigmoid

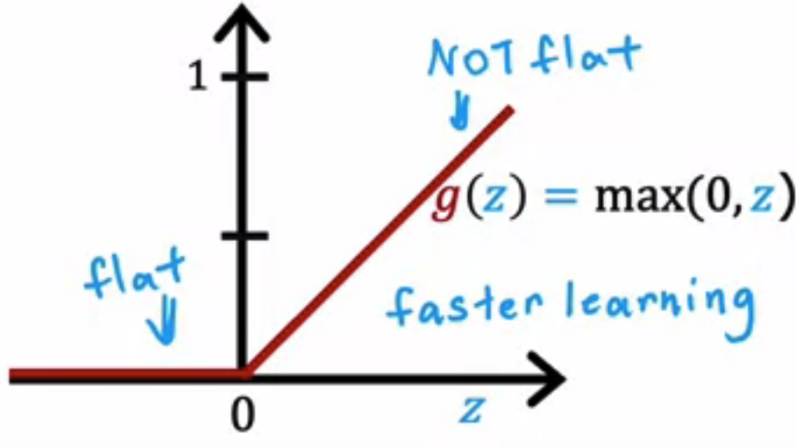




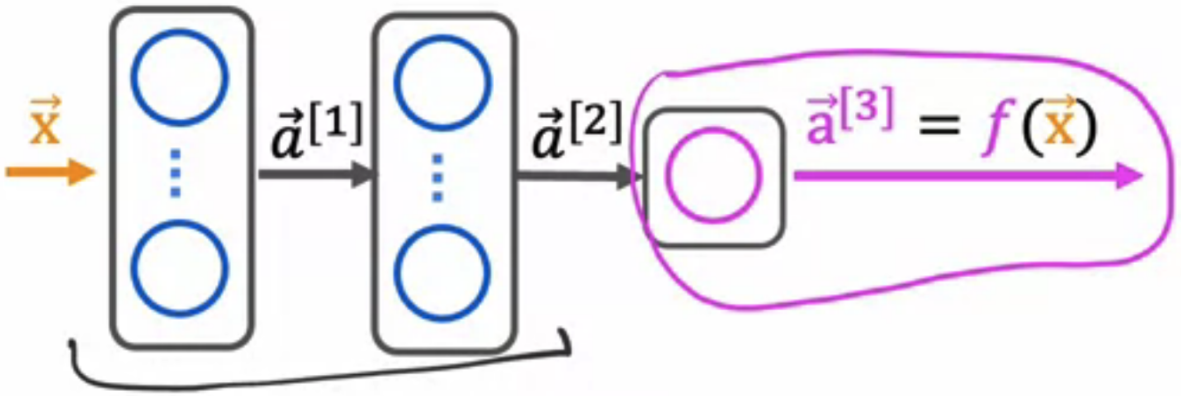
most common choice

ReLU

faster



Choosing Activation Summary



ReLU hidden layers

binary classification

activation='sigmoid'

regression y negative/positive

activation='linear'

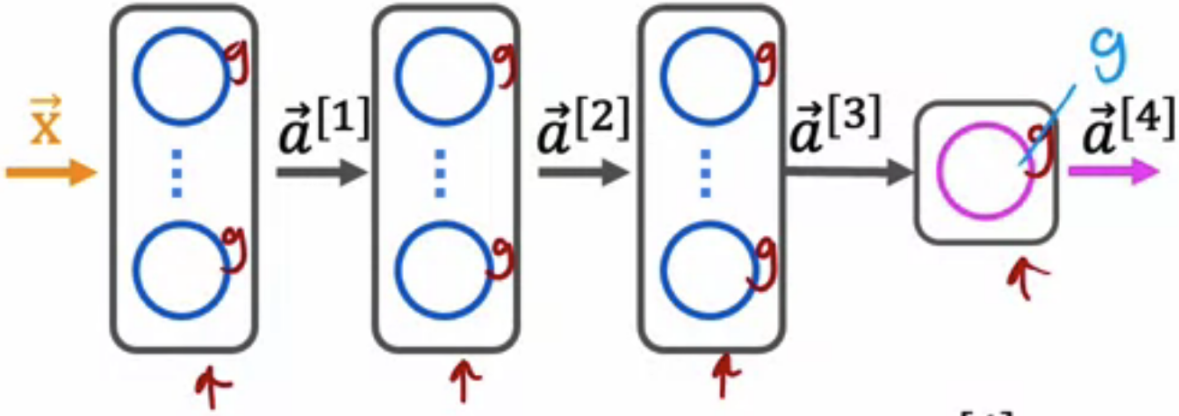
regression $y \geq 0$

activation='relu'

```
from tf.keras.layers import Dense
model = Sequential([
  Dense(units=25, activation='relu'),    layer1
  Dense(units=15, activation='relu'),    layer2
  Dense(units=1,  activation='sigmoid')  layer3
])
```

or 'linear'
or 'relu'

Example



$g(z) = z$

$$\vec{a}^{[4]} = \vec{w}_1^{[4]} \cdot \vec{a}^{[3]} + b_1^{[4]}$$

all linear (including output)
↳ equivalent to linear regression

$$\vec{a}^{[4]} = \frac{1}{1+e^{-(\vec{w}_1^{[4]} \cdot \vec{a}^{[3]} + b_1^{[4]})}}$$

output activation is sigmoid
(hidden layers still linear)
↳ equivalent to logistic regression

Don't use linear activations in hidden layers (use ReLU)