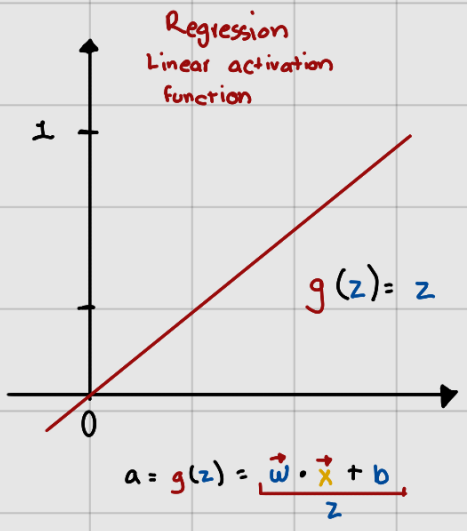
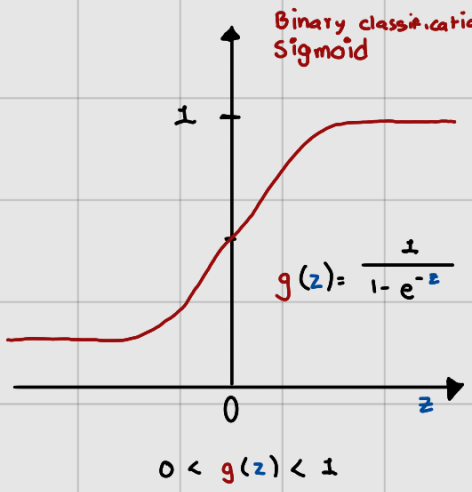
Activation Functions

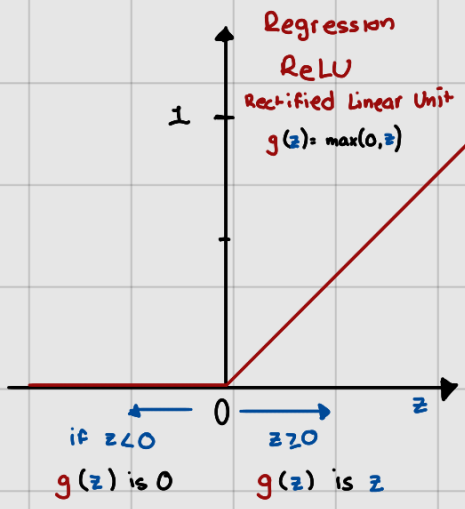
$$a_2^{[1]} = g(\underbrace{\vec{w}_2^{[1]} \cdot \vec{x} + b_2^{[1]}}_{z})$$



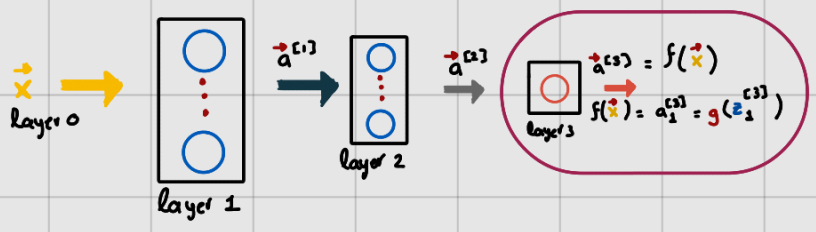
$$a = g(z) = \underbrace{\vec{w} \cdot \vec{x} + b}_{z}$$



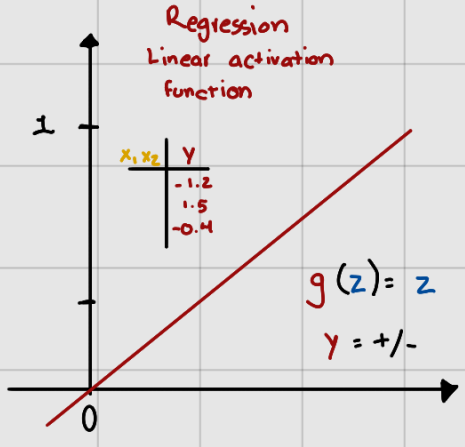
$0 < g(z) < 1$



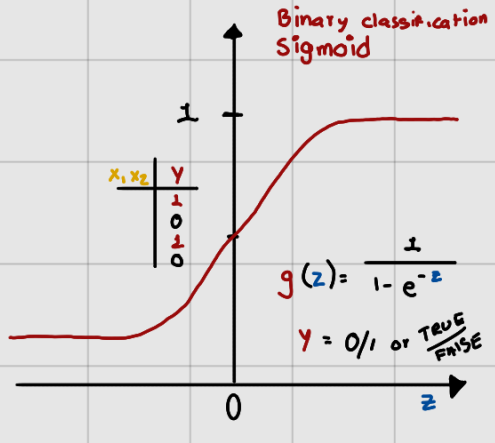
Output Layer



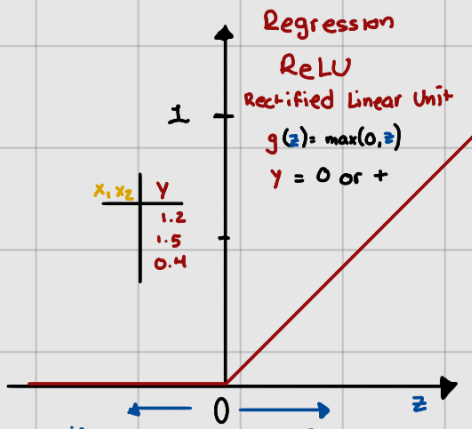
Choosing $g(z)$ for output layer



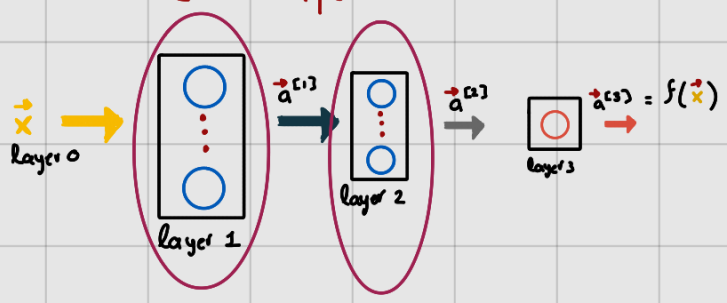
$$a = g(z) = \underbrace{\vec{w} \cdot \vec{x} + b}_{z}$$



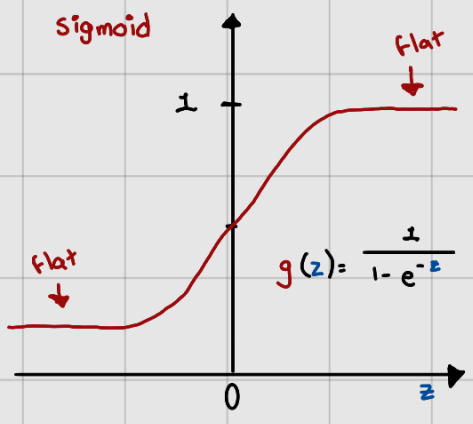
$0 < g(z) < 1$



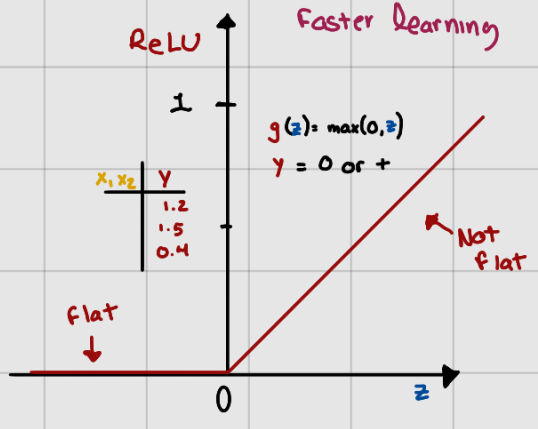
Hidden Layer



Choosing $g(z)$ for hidden layer



Most common choice
Faster learning

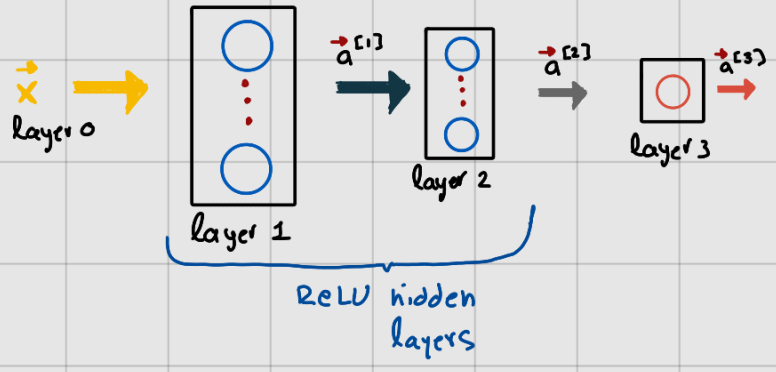


In code :

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'),     # layer 1
        Dense(units = 15, activation = 'relu'),     # layer 2
        Dense(units = 25, activation = 'linear')    # layer 3
                                  or
                                'sigmoid'
])
```

We use activation functions because without them, deep neural network would just compute a linear function.