

1. Create the Model

→ mention the way you will be going to compute the output using given input x and parameters \vec{w}, b .

traditionally

$$z = \text{np.dot}(w, x) + b$$
$$f_x = 1 / (1 + \text{np.exp}(-z))$$

tensorflow

```
model = Sequential([
    Dense(...),
    Dense(...),
    Dense(...),
])
```

2. Loss and cost functions

Loss function needs to be specified.
Tensorflow will automatically average using loss function.

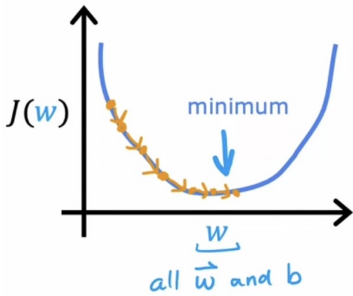
$$L(f(\vec{x}), y) = -y \log(f(\vec{x})) - (1-y) \log(1-f(\vec{x}))$$

↓
binary_crossentropy

```
model.compile(loss = BinaryCrossentropy())
```

↓
model.compile(loss = MeanSquaredError())
Regression

3. Gradient Descent



repeat {

$$w_j^{[l]} = w_j^{[l]} - \alpha \frac{\partial}{\partial w_j} J(\vec{w}, b)$$

$$b_j^{[l]} = b_j^{[l]} - \alpha \frac{\partial}{\partial b_j} J(\vec{w}, b)$$

}

using \swarrow backprop.
to compute derivatives

\Rightarrow In tensorflow: `model.fit(x, y, epochs = 100)`