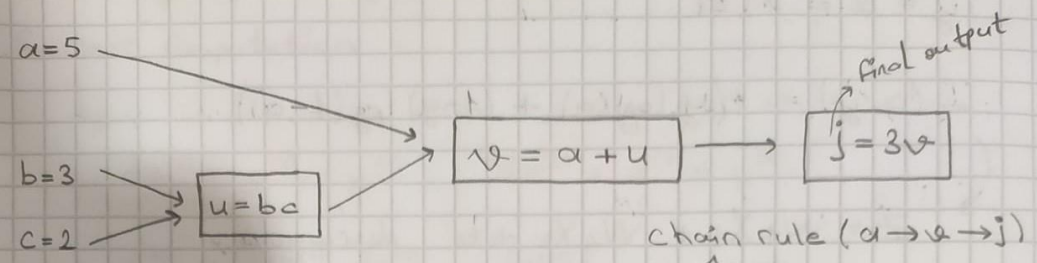
2) Computation Graph



chain rule ($a \to v \to j$)

$$\frac{dj}{dv} = 3 \qquad \frac{dj}{da} = 3 = \underbrace{\frac{dj}{dv}}_{3} \underbrace{\frac{dv}{da}}_{1}$$

↳ 1 br v'de ne kadar j değişir

↳ a = 5 iken 5.001 olsun

v = 11 ⟶ 11.001

j = 33 ⟶ 33.003

$\frac{dj}{da}$ ↷ da şeklinde de yazılabilir (Derivative of j with respect to a)

$$\rightarrow \frac{dj}{db} = \frac{dj}{du} \cdot \underbrace{\frac{du}{db}}_{2}$$

$$\frac{dj}{dv}\frac{dv}{du} = 3$$

b = 3 iken 3.001 olsun

u = b·c = 6 → 6.002 ; c = 2

= 2

Forward propagation → * Computing the cost function.
* To get the output and compare it with the real value to find error.

Backward (Backpropagation → * Use a form of Gradient Descant to update new weight values.

↳ After finding, network error, you back propagate and use a form of gradient descant to update new values of weights. Then, you will again forward propagate to see how well those weights are performing.

Logistic Regression Derivatives

$$z = w^T x + b$$

$$\hat{y} = a = \sigma(z)$$

$$\mathcal{L}(a, y) = -(y \log(a) + (1-y) \log(1-a))$$

$z = w_1 x_1 + w_2 x_2 + b$ durumda;

$$\frac{\partial \mathcal{L}}{\partial w_1} = dw_1 = x_1 \cdot dz \quad ; \quad dw_2 = x_2 \cdot dz \quad ; \quad db = dz$$

(average of all m training example)

$$w_1 := w_1 - \alpha \, dw_1 \rightarrow dw_1/m$$

$$w_2 := w_2 - \alpha \, dw_2$$

$$b := b - \alpha \, db \rightarrow db/m$$

(α: learning rate)