ENGR 4350: Applied Deep Learning

Logistic Regression: Part 2

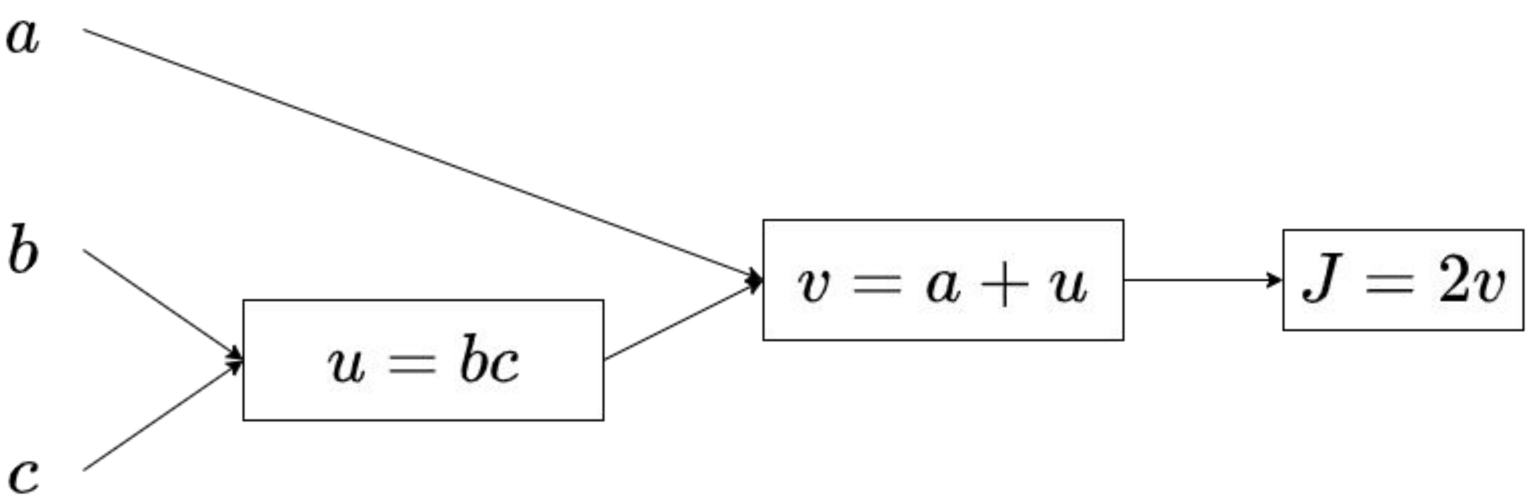
09/07/2022

Outline

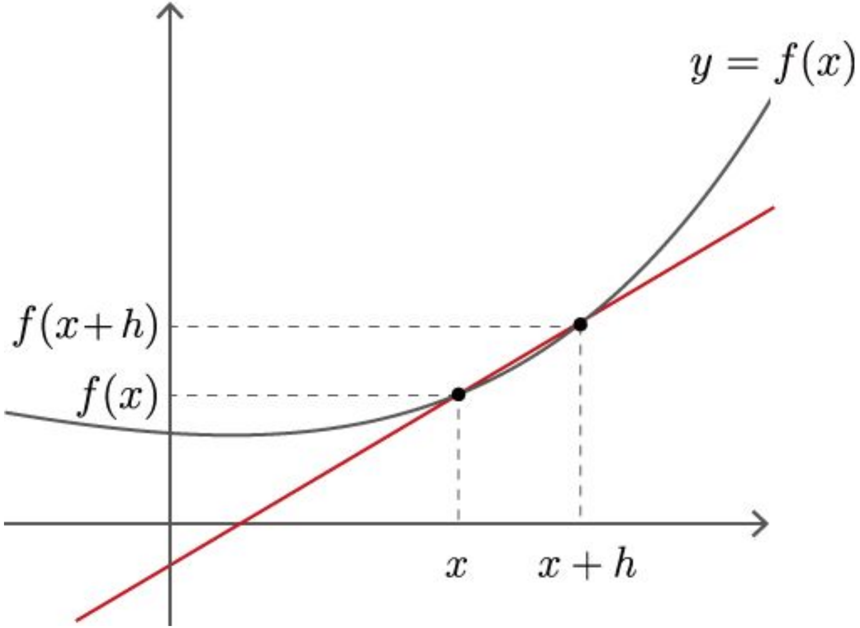
- Computation Graph
- Single Example Back Propagation
- Vectorization

Computation Graph: Forward Pass

$$J = 2(a + bc)$$



Derivatives



Analytic derivative: $f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

- Fast
- Accurate
- Error-Prone

Numerical derivative:

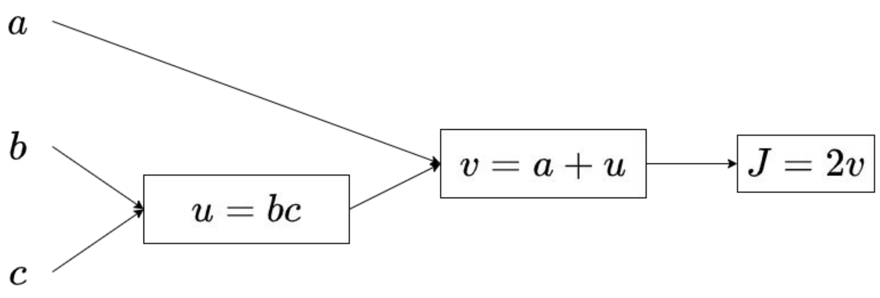
$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

- Slow
- Approximate
- Easy to code

Computation Graph: Backward Pass

$$J = 2(a + bc)$$



$$\frac{dJ}{dv} = 2$$

$$\frac{dJ}{da} = \frac{dJ}{dv} \cdot \frac{dv}{da} = 2$$

$$\frac{dJ}{du} = \frac{dJ}{dv} \cdot \frac{dv}{du} = 2$$

$$\frac{dJ}{db} = \frac{dJ}{dv} \cdot \frac{dv}{du} \cdot \frac{du}{db} = 2c$$

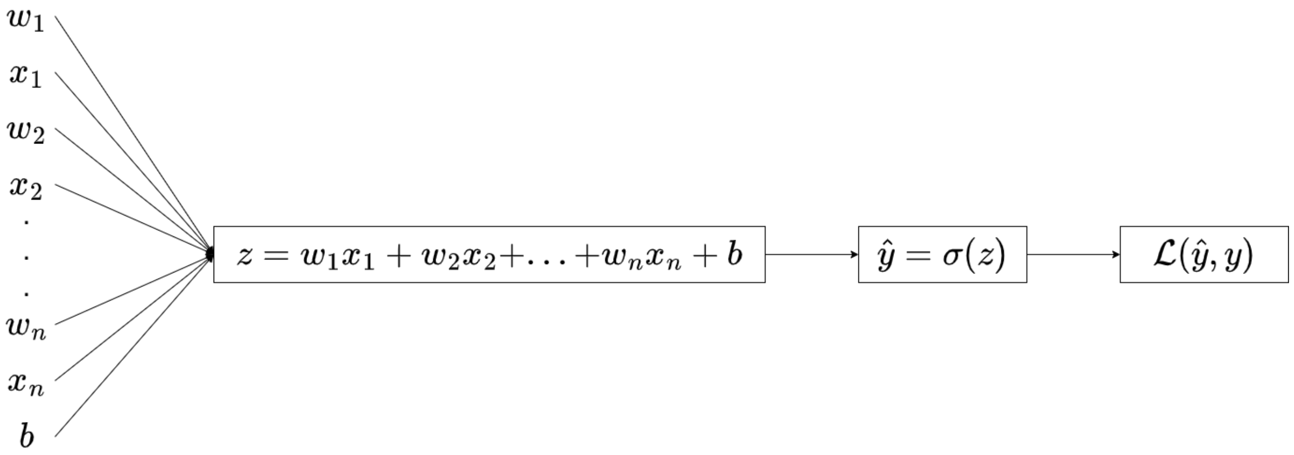
$$\frac{dJ}{dc} = \frac{dJ}{du} \cdot \frac{du}{dc} = 2b$$

Computation Graph of Logistic Regression

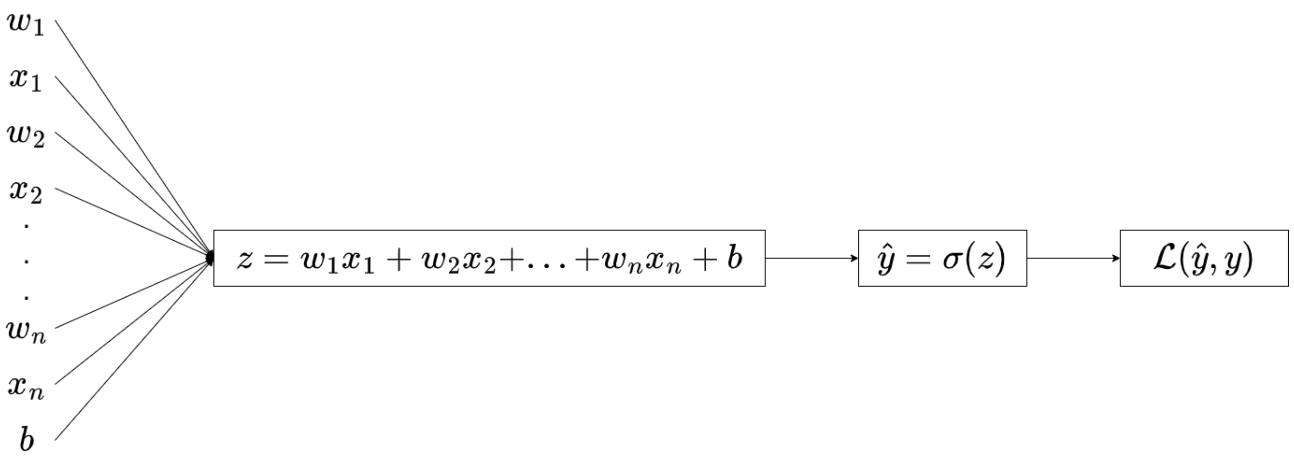
$$z = w_1x_1 + w_2x_2 + \ldots + w_\mathrm{n}x_\mathrm{n} + b$$

$$\hat{y} = \sigma(z) = \frac{1}{1 + e^{-z}}$$

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



Back Propagation of Logistic Regression



$$\frac{d\mathcal{L}}{dw_i} = \frac{d\mathcal{L}}{d\hat{y}} \cdot \frac{d\hat{y}}{dz} \cdot \frac{dz}{dw_i} = (\hat{y} - y)x_i$$

$$\frac{d\mathcal{L}}{db} = \frac{d\mathcal{L}}{d\hat{y}} \cdot \frac{d\hat{y}}{dz} \cdot \frac{dz}{db} = \hat{y} - y$$

$$\frac{d\mathcal{L}}{dz} = \frac{d\mathcal{L}}{d\hat{y}} \cdot \frac{d\hat{y}}{dz} = \hat{y} - y$$

$$\frac{d\mathcal{L}}{d\hat{y}} = -\frac{y}{\hat{y}} + \frac{1-y}{1-\hat{y}}$$

Back Propagation Loop

Initialize: $\frac{\partial J}{\partial w_1} = 0,\ \frac{\partial J}{\partial w_2} = 0,\ \ldots,\ \frac{\partial J}{\partial w_n} = 0,\ \frac{\partial J}{\partial b} = 0$

For i = 1 to m

For j = 1 to n

$$\frac{\partial J}{\partial w_j} = \frac{\partial J}{\partial w_j} + \left(\hat{y}^{(i)} - y^{(i)}\right) x_j^{(i)}$$

$$\frac{\partial J}{\partial b} = \frac{\partial J}{\partial b} + \left(\hat{y}^{(i)} - y^{(i)}\right)$$

For j = 1 to n

$$\frac{\partial J}{\partial w_j} = \frac{1}{m} \frac{\partial J}{\partial w_j}$$

$$w_j = w_j - \alpha \frac{\partial J}{\partial w_j}$$

$$\frac{\partial J}{\partial b} = \frac{1}{m} \frac{\partial J}{\partial b}$$

$$b = b - \alpha \frac{\partial J}{\partial b}$$

Vectorization

Initialize: $\frac{\partial J}{\partial w_1} = 0, \frac{\partial J}{\partial w_2} = 0, \ldots, \frac{\partial J}{\partial w_n} = 0, \frac{\partial J}{\partial b} = 0$

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{1}{m}(\hat{\mathbf{y}} - \mathbf{y})^T \cdot \mathbf{X}$$

`np.matmul()` or `np.dot()`

$$\frac{\partial J}{\partial b} = \frac{1}{m}\sum(\hat{\mathbf{y}} - \mathbf{y})$$

`np.sum()`

$$\mathbf{w} = \mathbf{w} - \alpha\frac{\partial J}{\partial \mathbf{w}}$$

$$b = b - \alpha\frac{\partial J}{\partial b}$$

Vectorized Gradient Descent

While $J > \varepsilon$

$$\hat{\mathbf{y}} = \sigma\left(\mathbf{X}\mathbf{w}^{\mathbf{T}} + b\right)$$

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

$$J(\mathbf{w}, b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$$

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{1}{m}(\hat{\mathbf{y}} - \mathbf{y})^{T} \cdot \mathbf{X}$$

$$\frac{\partial J}{\partial b} = \frac{1}{m}\sum(\hat{\mathbf{y}} - \mathbf{y})$$

$$\mathbf{w} = \mathbf{w} - \alpha\frac{\partial J}{\partial \mathbf{w}}$$

$$b = b - \alpha\frac{\partial J}{\partial b}$$