One hidden layer
Neural Network

Backpropagation
intuition (Optional)

deeplearning.ai

Computing gradients

Logistic regression

x_1
x_2 → σ → $\hat{y} = a$
x_3

x_1
x_2 → $\circ$ → $\circ$ → $\hat{y}$
x_3

x
w →
b →
$$\boxed{z = w^T x + b}$$ → $$\boxed{a = \sigma(z)}$$ → $$\boxed{\mathcal{L}(a, y)}$$

$dw = dz \cdot x$

$db = dz$

$dz = a - y$

$dz = da \cdot g'(z)$

$g(z) = \sigma(z)$

$$\frac{\partial \mathcal{L}}{\partial z} = \frac{\partial \mathcal{L}}{\partial a} \cdot \boxed{\frac{da}{dz}}$$

"dz" = "da" ·

$da = \frac{d}{da} \mathcal{L}(a, y) = -y \log a - (1-y) \log(1-a)$

$= -\frac{y}{a} + \frac{1-y}{1-a}$

$\frac{d}{dz} g(z) = \underline{g'(z)}$

Andrew Ng
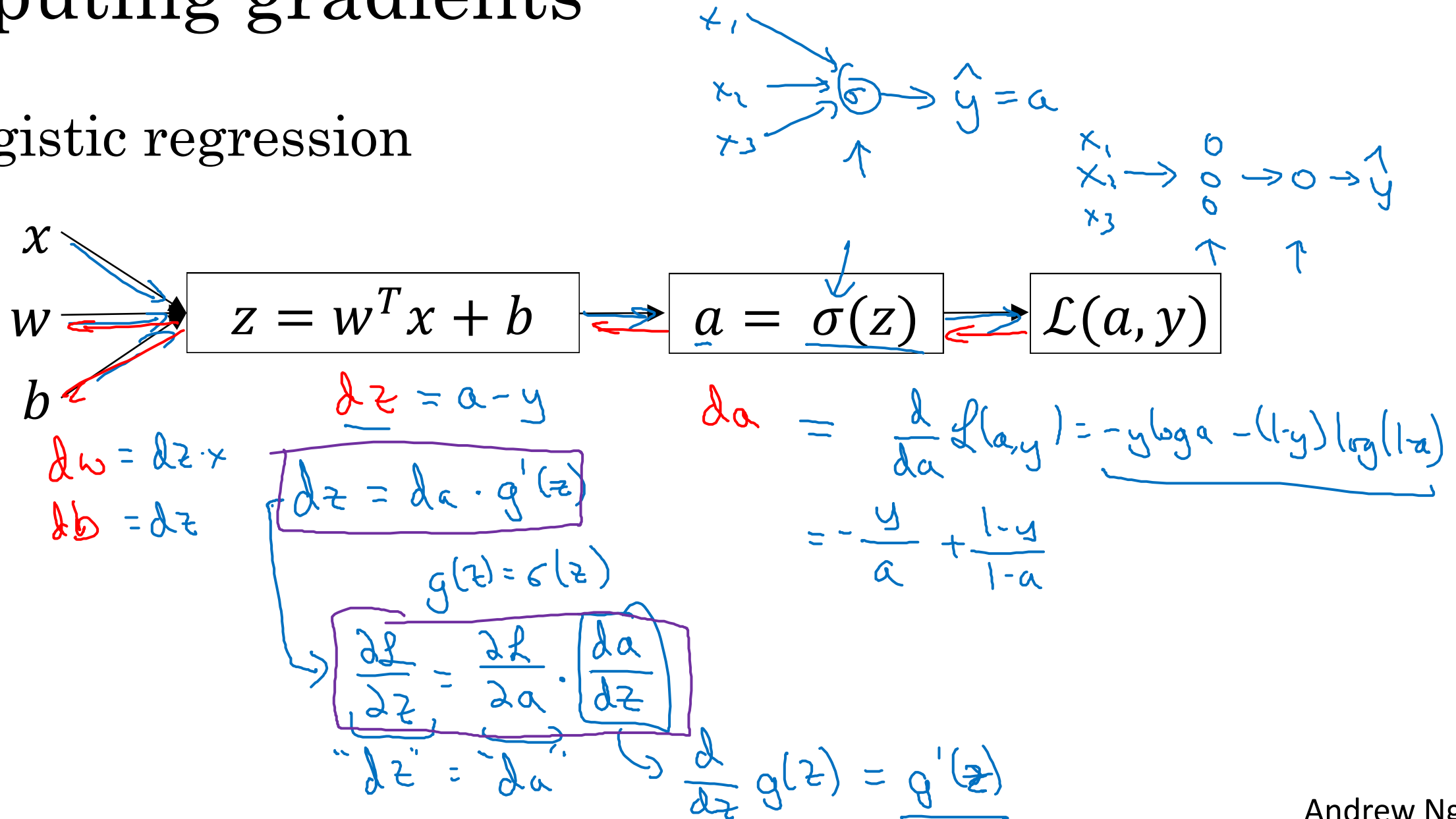
Neural network gradients

$x_1 \rightarrow O$
$x_2 \rightarrow O \rightarrow O \rightarrow \hat{y}$
$x_3 \rightarrow O$

$n_x = n^{[0]}$ $n^{[1]}$ $n^{[2]} = 1$

$W^{[2]}$ $dW^{[2]}$

$b^{[2]}$ $db^{[2]}$

x

$W^{[1]}$

$dw^{[1]}$

$b^{[1]}$

$db^{[1]}$

| $z^{[1]} = W^{[1]}x + b^{[1]}$ | $a^{[1]} = \sigma(z^{[1]})$ | $z^{[2]} = W^{[2]}x + b^{[2]}$ | $a^{[2]} = \sigma(z^{[2]})$ | $\mathcal{L}(a^{[2]}, y)$ |

$dz^{[1]} = W^{[1]T} dz^{[1]}$

$da^{[1]} = \cdots$

$dz^{[2]} = a^{[2]} - y$

$da^{[2]}$

$* g^{[1]'}(z^{[1]})$
$\curvearrowright$ element wise product

$W^{[2]} \quad (n^{[2]}, n^{[1]})$

$z^{[2]}, dz^{[2]} \quad (n^{[2]}, 1) \quad - (1,1)$

$z^{[1]}, dz^{[1]} \quad (n^{[1]}, 1)$

$dz^{[1]} = \boxed{W^{[2]T} \, dz^{[2]}} * \boxed{g^{[1]'}(z^{[1]})}$

$(n^{[1]}, 1) = (n^{[1]}, n^{[2]}) \, (n^{[2]}, 1) * (n^{[1]}, 1)$

$dW^{[2]} = dz^{[2]} a^{[1]T}$

$db^{[2]} = dz^{[2]}$

$\text{``} dw = dz \cdot x \text{''}$

$W^{[2]}: [\underline{\qquad}]$

foo $W^{[1]}$
$dfoo$ $dW^{[2]}$

$dW^{[1]} = dz^{[1]} \cdot x^T$

$db^{[1]} = dz^{[1]}$

$a^{[0]T}$
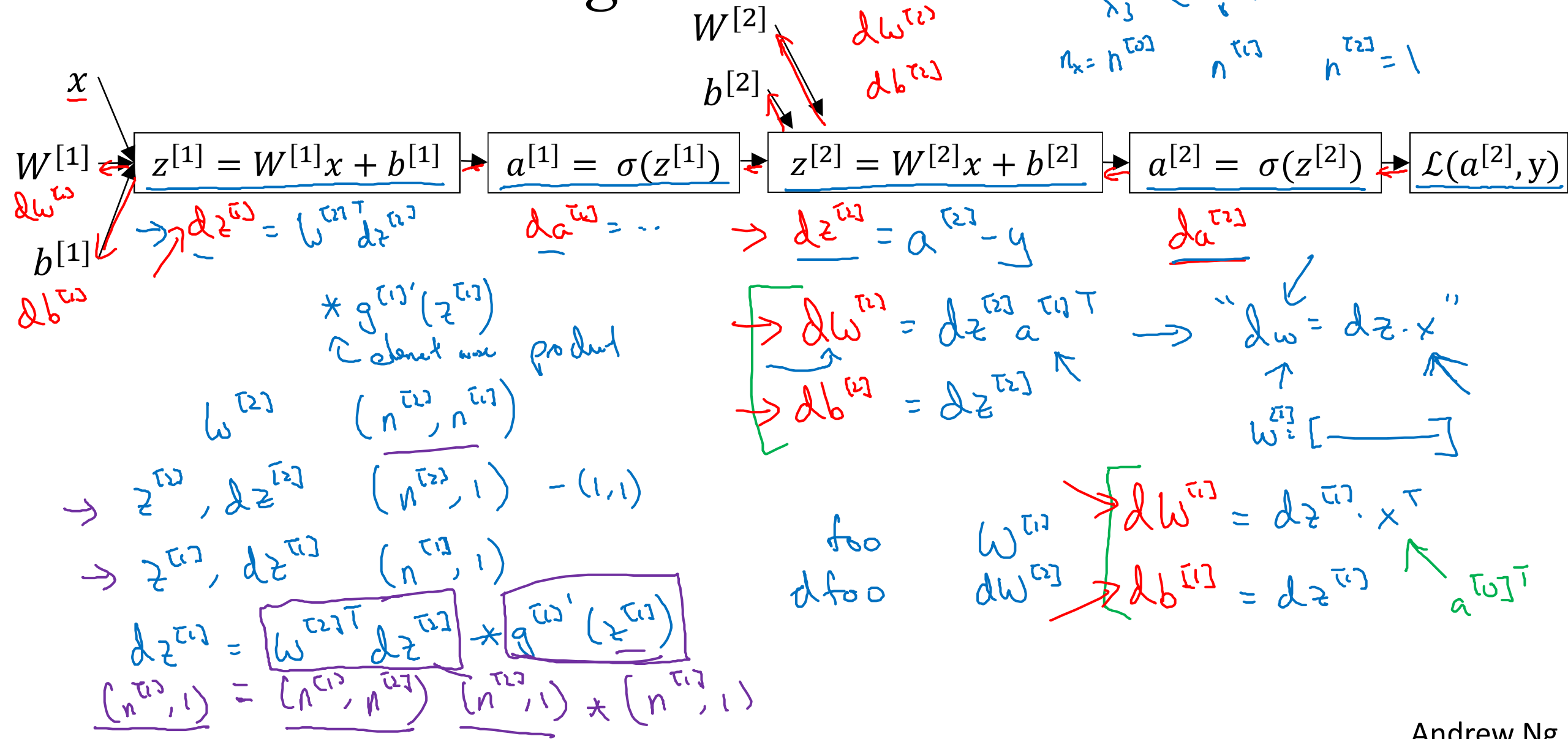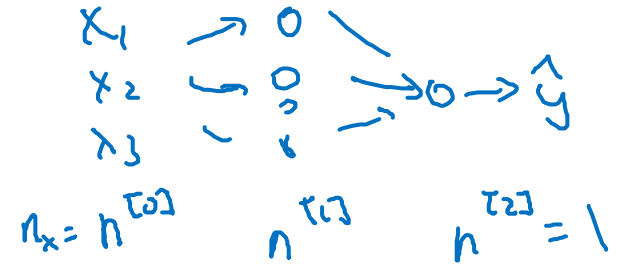
Summary of gradient descent

$$dz^{[2]} = a^{[2]} - y$$

$$dW^{[2]} = dz^{[2]} a^{[1]^T}$$

$$db^{[2]} = dz^{[2]}$$

$$dz^{[1]} = W^{[2]^T} dz^{[2]} * g^{[1]'}(z^{[1]})$$

$$dW^{[1]} = dz^{[1]} x^T$$

$$db^{[1]} = dz^{[1]}$$

Vectorized Implementation:

$$z^{[i]} = W^{[i]} x + b^{[i]}$$

$$a^{[i]} = g^{[i]}(z^{[i]})$$

$$Z^{[i]} = \begin{bmatrix} | & | & & | \\ z^{[i](1)} & z^{[i](2)} & \cdots & z^{[i](m)} \\ | & | & & | \end{bmatrix}$$

$$Z^{[i]} = W^{[i]} X + b^{[i]}$$

$$A^{[i]} = g^{[i]}(Z^{[i]})$$

Summary of gradient descent

$$dz^{[2]} = a^{[2]} - y$$

$$dW^{[2]} = dz^{[2]} a^{[1]T}$$

$$db^{[2]} = dz^{[2]}$$

$$dz^{[1]} = W^{[2]T} dz^{[2]} * g^{[1]\prime}(z^{[1]})$$

$(n^{[1]}, 1)$

$$dW^{[1]} = dz^{[1]} x^T$$

$$db^{[1]} = dz^{[1]}$$

$$dZ^{[2]} = A^{[2]} - Y$$

$$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}$$

$$db^{[2]} = \frac{1}{m} np.sum(dZ^{[2]}, axis = 1, keepdims = True)$$

elementwise product

$$dZ^{[1]} = \underbrace{W^{[2]T} dZ^{[2]}}_{(n^{[1]}, m)} * \underbrace{g^{[1]\prime}(Z^{[1]})}_{(n^{[1]}, m)}$$

$(n^{[1]}, m)$

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$$

$$db^{[1]} = \frac{1}{m} np.sum(dZ^{[1]}, axis = 1, keepdims = True)$$

$$J(\ldots) = \frac{1}{m} \sum_{i=1}^{n} \mathcal{L}(\hat{y}, y)$$

Andrew Ng