Neural Network on Spark

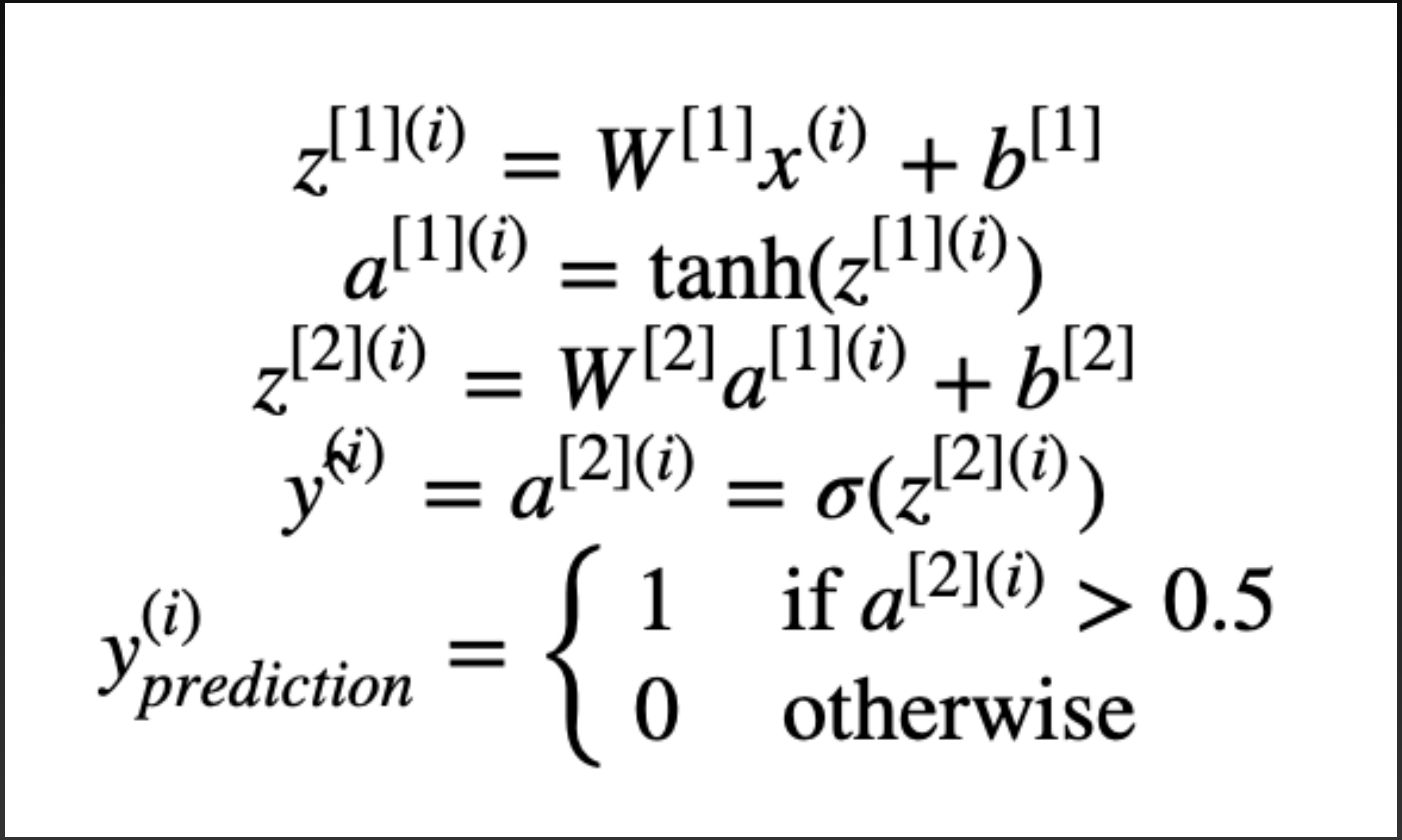
Zihao Li

Mon Aug 17

NN-Model

input layer of size 10,000
hidden layer of size n_h
activation: tanh
output layer of size 1
activation: sigmoid
probability & prediction
W1, b1
W2, b2
x_1
x_2
x_3
x_{10k}
a_1
a_2
a_3
a_{n_h}
sig(W2A1 + b2)
0.66
1

Forward-Propagation



$$z^{[1](i)} = W^{[1]}x^{(i)} + b^{[1]}$$
$$a^{[1](i)} = \tanh(z^{[1](i)})$$
$$z^{[2](i)} = W^{[2]}a^{[1](i)} + b^{[2]}$$
$$\hat{y}^{(i)} = a^{[2](i)} = \sigma(z^{[2](i)})$$
$$y^{(i)}_{prediction} = \begin{cases} 1 & \text{if } a^{[2](i)} > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Compute Cost

Binary Cross Entropy

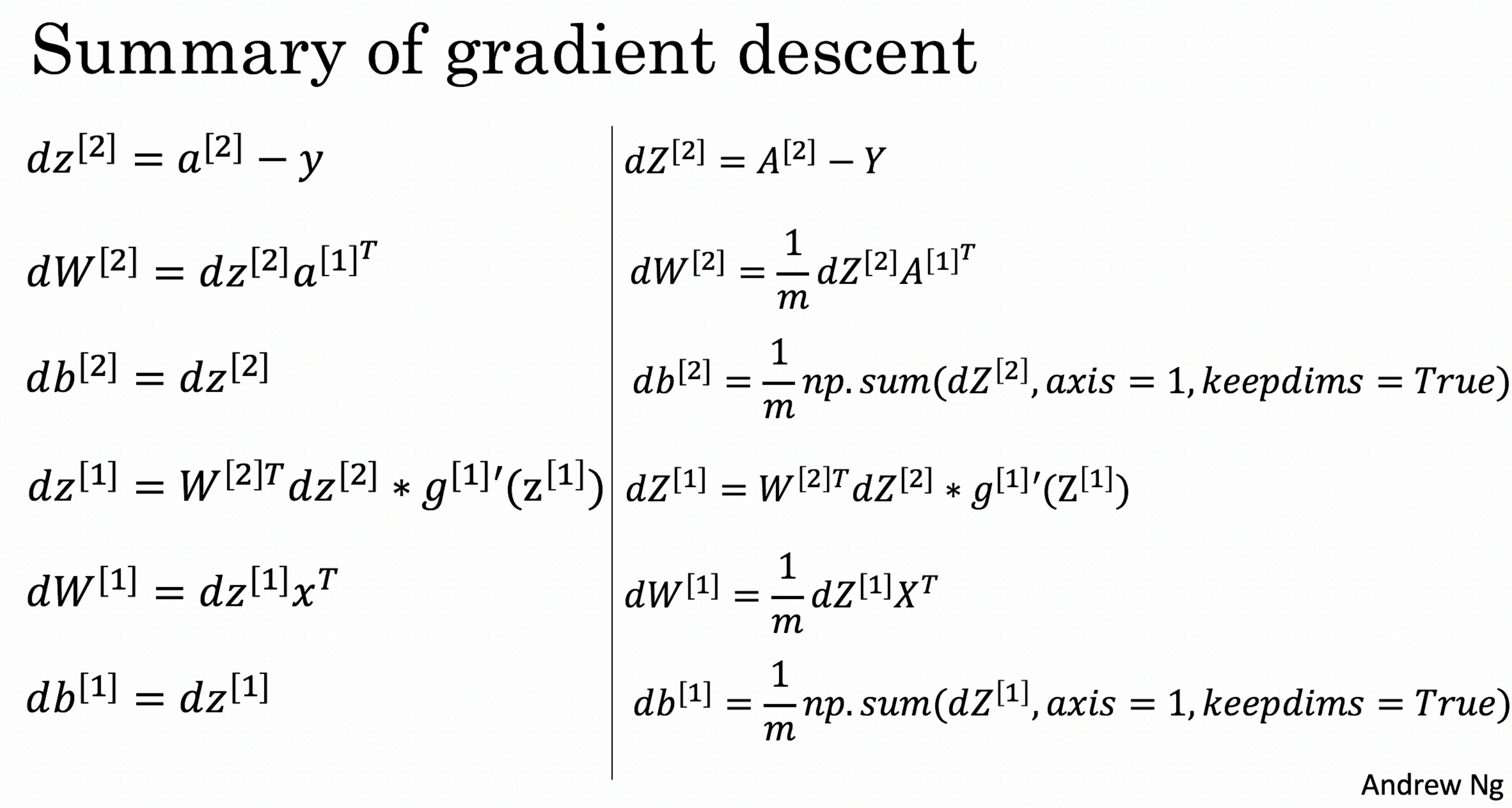
$$L_{bce} = -ylog(\widehat{y}) - (1 - y)log(1 - \widehat{y})$$

Weighted Binary Cross Entropy

$$L_{vbce} = -ylog(\widehat{y}) * NR - (1 - y)log(1 - \widehat{y}) * (1 - NR)$$

$$NR = \frac{numberofnegativesamples}{numberofallsamples}$$

Backward-Propagation



Summary of gradient descent

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

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

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

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

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

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

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

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

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

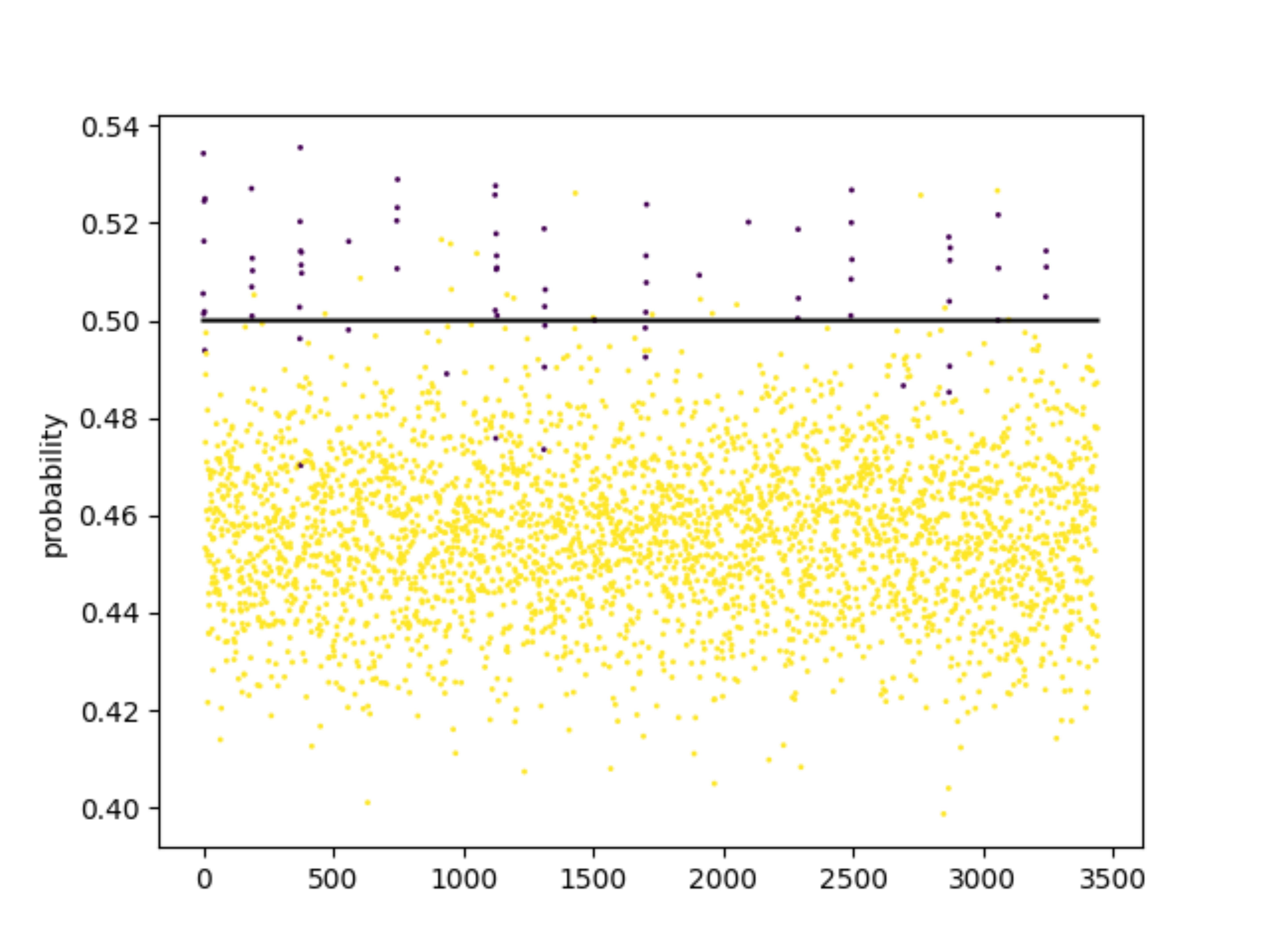
$$dZ^{[1]} = W^{[2]T} dZ^{[2]} * g^{[1]\prime}(\mathrm{Z}^{[1]})$$

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

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

Andrew Ng

Conclusion



0.54
0.52
0.50
0.48
0.46
0.44
0.42
0.40
probability
0
500
1000
1500
2000
2500
3000
3500

Thanks for Watching!

Zihao Li

Mon Aug 17