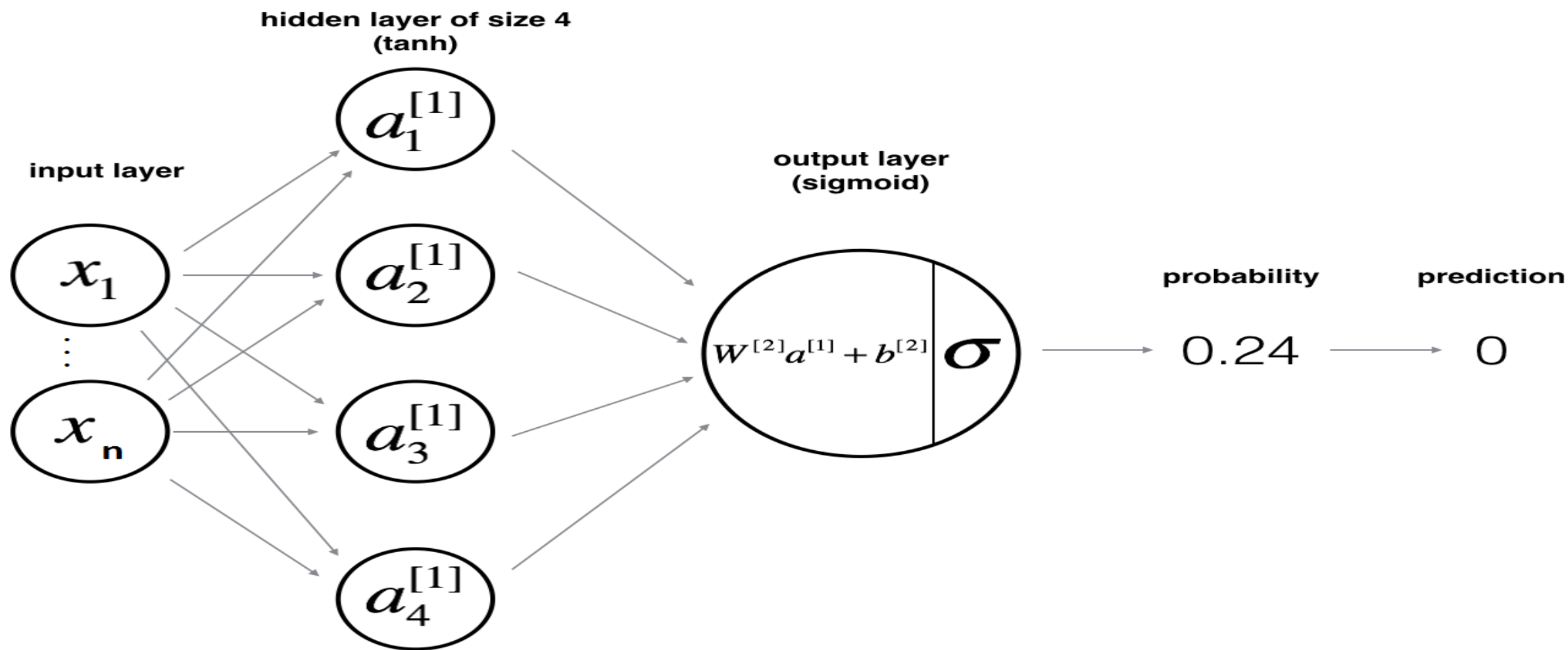


Shallow Neural Network lab5

Eng: Samar Shaban

- Problem: sonar Mines VS Rocks
- Model : neural network
- Performance matrix: Accuracy



1. Define the neural network structure (# of input units, # of hidden units, etc).
2. Initialize the model's parameters
3. Loop:
 - Implement forward propagation
 - Compute loss
 - Implement backward propagation to get the gradients
 - Update parameters (gradient descent)

→ merge steps from 1-3 into one function we call nn_model()

Mathematically:

For one example $x^{(i)}$:

$$z^{[1](i)} = W^{[1]}x^{(i)} + b^{[1]} \quad (1)$$

$$a^{[1](i)} = \tanh(z^{[1](i)}) \quad (2)$$

$$z^{[2](i)} = W^{[2]}a^{[1](i)} + b^{[2]} \quad (3)$$

$$\hat{y}^{(i)} = a^{[2](i)} = \sigma(z^{[2](i)}) \quad (4)$$

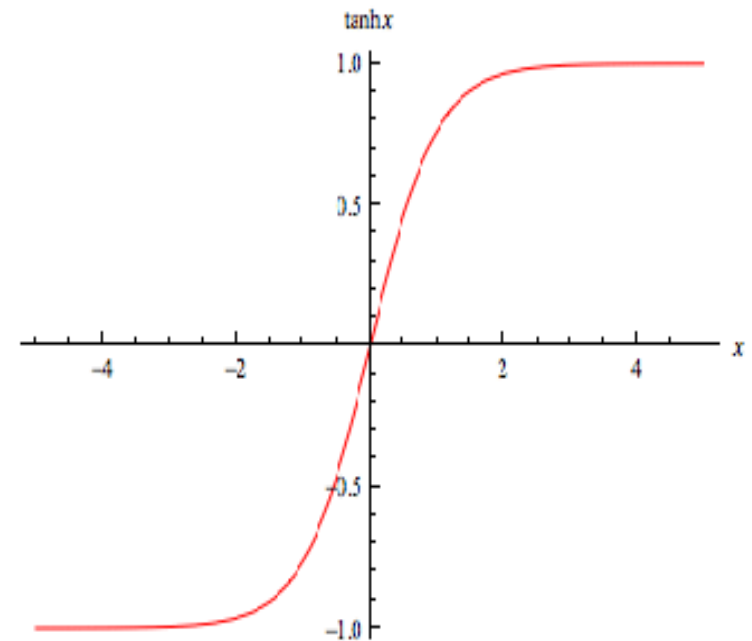
$$y_{prediction}^{(i)} = \begin{cases} 1 & \text{if } a^{[2](i)} > 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

Given the predictions on all the examples, you can also compute the cost J as follows:

$$J = -\frac{1}{m} \sum_{i=0}^m \left(y^{(i)} \log(a^{[2](i)}) + (1 - y^{(i)}) \log(1 - a^{[2](i)}) \right) \quad (6)$$

- Hyperbolic Tangent: tanh

$$\text{Tanh}(x) = (e^x - e^{-x}) / (e^x + e^{-x})$$



Gradient descent

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

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

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

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

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

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

Andrew Ng