Neural Network Forward Pass

1. **Univariate Logistic Regression Calculation**

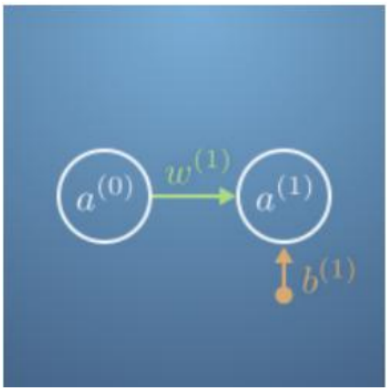


image: Mathematics for Machine Learning

Input layer: $a^{[0]} = x$

Output layer: (first layer)

$$Z^{[1]} = W^{[1]} a^{[0]} + W_0{}^{[1]}$$

$$a^{[1]} = \sigma\left(Z^{[1]}\right)$$

Activation function: Sigmoid

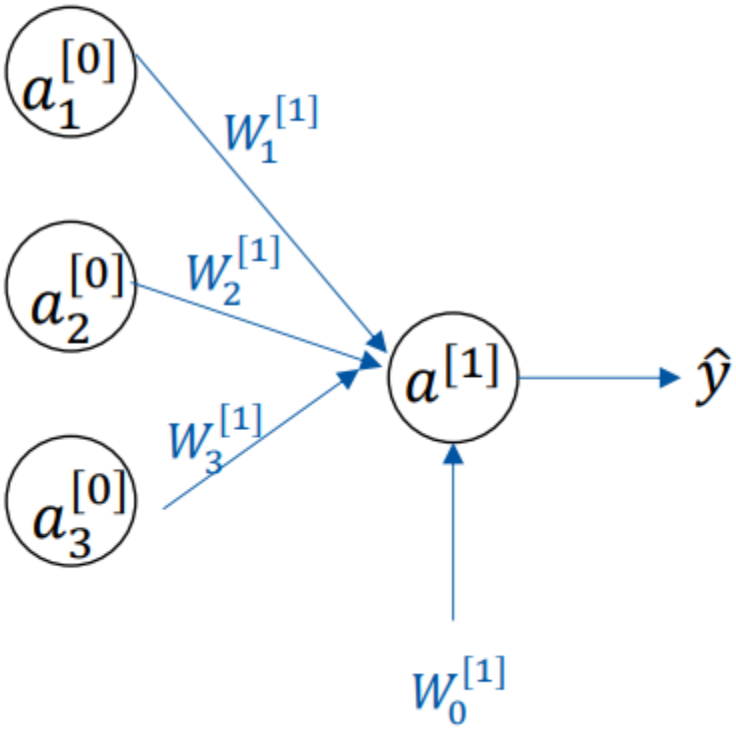
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

Given a single neuron where the input $x = 2$, the weight $W^{[1]} = 0.5$, and the bias $b^{[1]} = -0.5$ (or $W_0^{[1]} = -0.5$ with $a^{[0]} = x$).

1. Calculate the linear output $Z^{[1]}$.
2. Calculate the activation $a^{[1]}$ using the Sigmoid function.

1.

Multivariate Logistic Regression



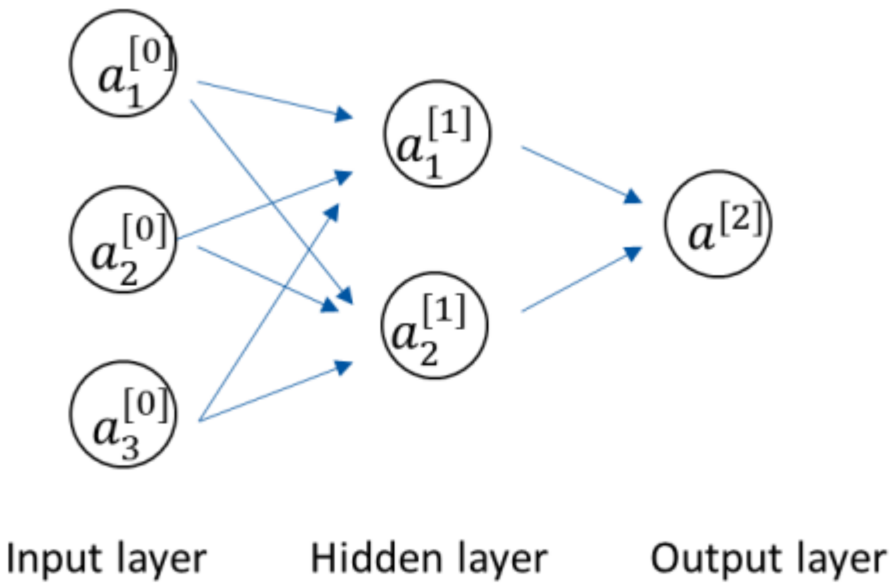
Consider a multivariate case with three inputs. Given:

- Weights: $W_0^{[1]} = -1$ (Bias), $W_1^{[1]} = 2$, $W_2^{[1]} = 0.5$, $W_3^{[1]} = -2$
- Inputs: $a_1^{[0]} = 1$, $a_2^{[0]} = 4$, $a_3^{[0]} = 0.5$
- Note: The bias term corresponds to an implicit input of 1.

Calculate $Z^{[1]}$ and determine the predicted class $\hat{y}$ (threshold = 0.5).

1.

Shallow Neural Network: Hidden Layer Calculation



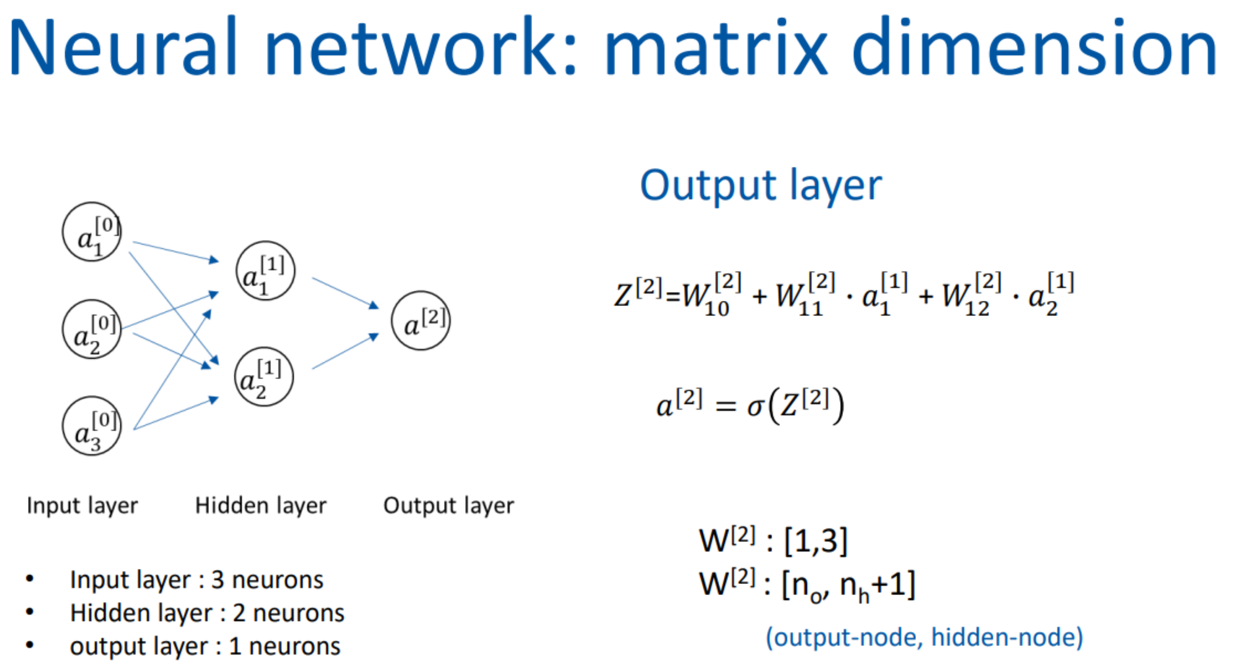
We are calculating the first neuron in the hidden layer, $Z_1^{[1]}$. Inputs ($a^{[0]}$): $[10, 20, 30]$ (for neurons 1, 2, and 3). Weights for the first hidden neuron ($W_{1...}^{[1]}$):

- $W_{10}^{[1]} = 0$ (Bias)
- $W_{11}^{[1]} = 0.1$
- $W_{12}^{[1]} = 0.2$
- $W_{13}^{[1]} = -0.1$

Calculate $Z_1^{[1]}$.

1.

Matrix Dimensions in Neural Networks



Consider a Shallow Neural Network with the following structure:

- **Input layer:**

  5 neurons ($n_x = 5$)
- **Hidden layer:**

  4 neurons ($n_h = 4$)
- **Output layer:**

  2 neurons ($n_o = 2$) (e.g., a multiclass classifier).

Using the notation from the slides (where the weight matrix includes the bias column), what are the dimensions of $W^{[1]}$ and $W^{[2]}$?

1.

The Symmetry Problem (Zero Initialization)

Assume we initialize a network with 2 hidden neurons ($a_1^{[1]}, a_2^{[1]}$) and all weights in $W^{[1]}$ are set to exactly

zero

. Explain mathematically what happens to $Z_1^{[1]}$ and $Z_2^{[1]}$ during the forward pass, and why this is problematic for learning.

Solutions

1.

1. Linear Output $Z^{[1]}$ Using the formula $Z^{[1]} = W^{[1]} \cdot x + b^{[1]}$:

$$Z^{[1]} = (0.5 \cdot 2) + (-0.5) = 1.0 - 0.5 = 0.5$$

2. Activation $a^{[1]}$ Using the formula $a^{[1]} = \sigma(Z^{[1]}) = \frac{1}{1+e^{-Z^{[1]}}}$:

$$a^{[1]} = \frac{1}{1+e^{-0.5}} \approx \frac{1}{1+0.6065} \approx \frac{1}{1.6065} \approx 0.622$$

2.

Step 1: Calculate $Z^{[1]}$ According to Slide 3:

$$Z^{[1]} = W_0^{[1]} \cdot 1 + W_1^{[1]} \cdot a_1^{[0]} + W_2^{[1]} \cdot a_2^{[0]} + W_3^{[1]} \cdot a_3^{[0]}$$

$$Z^{[1]} = (-1 \cdot 1) + (2 \cdot 1) + (0.5 \cdot 4) + (-2 \cdot 0.5)$$

$$Z^{[1]} = -1 + 2 + 2 - 1 = 2$$

Step 2: Calculate Activation and Prediction

$$a^{[1]} = \sigma(2) = \frac{1}{1+e^{-2}} \approx \frac{1}{1.135} \approx 0.88$$

Since $a^{[1]} = 0.88 \geq 0.5$, the prediction is:

$$\hat{y} = 1$$

3.

$$Z_1^{[1]} = W_{10}^{[1]} + W_{11}^{[1]} \cdot a_1^{[0]} + W_{12}^{[1]} \cdot a_2^{[0]} + W_{13}^{[1]} \cdot a_3^{[0]}$$

$$Z_1^{[1]} = 0 + (0.1 \cdot 10) + (0.2 \cdot 20) + (-0.1 \cdot 30)$$

$$Z_1^{[1]} = 0 + 1 + 4 - 3 = 2$$

The activation for this neuron would be $a_1^{[1]} = \sigma(2) \approx 0.88$.

4.

1. Matrix $W^{[1]}$ (Hidden Layer Weights) The slides define the dimensions as $[n_h, n_x + 1]$.

$$W^{[1]} \text{ dimensions: } [4, 5+1] = [4, 6]$$

(4 rows for the hidden neurons, 6 columns for the 5 inputs + 1 bias)

2. Matrix $W^{[2]}$ (Output Layer Weights) The slides define the dimensions as $[n_o, n_h + 1]$.

$$W^{[2]} \text{ dimensions: } [2, 4+1] = [2, 5]$$

(2 rows for the output neurons, 5 columns for the 4 hidden outputs + 1 bias)

5.

The Calculation: If $W^{[1]} = 0$, then for any input vector x:

$$Z_1^{[1]} = \sum (0 \cdot x_i) = 0$$

$$Z_2^{[1]} = \sum (0 \cdot x_i) = 0$$

Consequently, the activations are identical:

$$a_1^{[1]} = \sigma(0) = 0.5$$

$$a_2^{[1]} = \sigma(0) = 0.5$$

The Problem: Because both neurons output the exact same value, they will receive the exact same gradient update (dW) during backpropagation. In the next iteration, weights will change by the same amount, remaining identical. The neurons will fail to learn distinct features (symmetry), effectively acting as a single neuron. This is why we initialize with `0.01 * np.random.randn`.