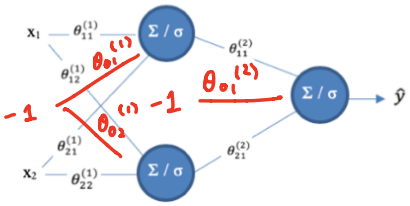
1. Consider the following multilayer perceptron.



The network should implement the XOR function. Perform one epoch of *backpropagation* as introduced in the lecture on multilayer perceptrons.

Notes:

- The activation function f for a perceptron is the *sigmoid function*:

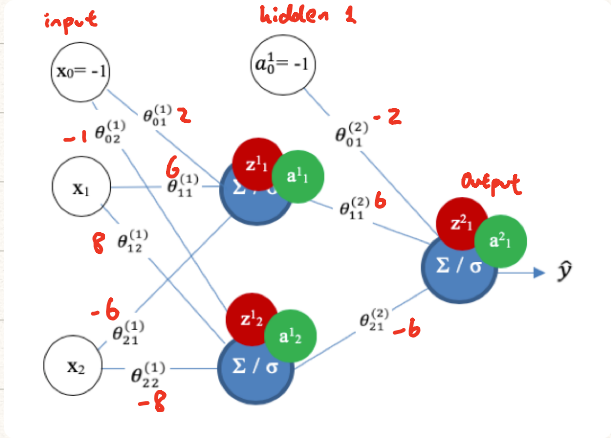
$$f(x) = \frac{1}{1 + e^{-x}}$$

- The thresholds are not shown in the network. The threshold nodes are set to -1.
- Use the following initial parameter values:

$$\theta_{01}^{(1)} = 2 \quad \theta_{02}^{(1)} = -1 \quad \theta_{01}^{(2)} = -2$$
$$\theta_{11}^{(1)} = 6 \quad \theta_{12}^{(1)} = 8 \quad \theta_{11}^{(2)} = 6$$
$$\theta_{21}^{(1)} = -6 \quad \theta_{22}^{(1)} = -8 \quad \theta_{21}^{(2)} = -6$$

- The learning rate is set to $\eta = 0.7$

i. Compute the activations of the hidden and output neurons.
ii. Compute the error of the network.
iii. Backpropagate the error to determine $\Delta\theta_{ij}$ for all weights θ_{ij} and updates the weight θ_{ij}.



(i) For level 1 :

$$a_1^{(1)} = \sigma(z_1^{(1)}) = \sigma(2(-1) + 6x_1 - 6x_2) = \sigma(-2 + 6x_1 - 6x_2)$$

$$a_2^{(1)} = \sigma(z_2^{(1)}) = \sigma(-1(-1) + 8x_1 - 8x_2) = \sigma(1 + 8x_1 - 8x_2)$$

For level 2 :

(Output $\hat{y}$ =) $a_1^{(2)} = \sigma(z_1^{(2)}) = \sigma(-2(-1) + 6a_1^{(1)} - 6a_2^{(1)}) = \sigma(2 + 6a_1^{(1)} - 6a_2^{(1)})$

(ii) Error:

$$E = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(y - a_1^{(2)})^2$$

(iii) Backpropagation (efficient way of computing partial derivatives of the error of MLP wrt. each weight $\theta_{jk}^{(l)}$. $\frac{\partial E}{\partial \theta_{jk}^{(l)}}$

1. Foward pass 2. Compute error 3. Backward pass (propagate error terms)

4. Update all θ (at once) $\Delta\theta_i$

GD: $\theta_i \leftarrow \theta_i - \eta \frac{\partial E}{\partial \theta_i}$ ← By back propagation

In MLP:

$$\theta_{jk}^{(l)} \leftarrow \theta_{jk}^{(l)} + \Delta\theta_{jk}^{(l)}$$

$$\Delta\theta_{jk}^{(l)} = -\eta \frac{\partial E}{\partial \theta_{jk}^{(l)}} = \eta\, \delta_k^{(l)} a_j^{(l-1)}$$ ← prev layer

error term: $\delta_k^{(l)} = (y - a_k^{(l)})\, \sigma'(z_k^{(l)})$ (lecture)

(last layer) $= (y - a_k^{(l)})\, \sigma(z_k^{(l)})(1 - \sigma(z_k^{(l)}))$

Chain Rule:

$$E = \frac{1}{2}(y - a_k^{(l)})^2$$

$$\frac{\partial E}{\partial \theta_{jk}^{(l)}} = \frac{\partial E}{\partial a_k^{(l)}} \cdot \frac{\partial a_k^{(l)}}{\partial z_k^{(l)}} \cdot \frac{\partial z_k^{(l)}}{\partial \theta_{jk}^{(l)}}$$

$$= -(y - a_k^{(l)}) \cdot \sigma'(z_k^{(l)}) \cdot a_j^{(l)}$$

$$= -\delta_k^{(l)} a_j^{(l-1)}$$

prev layer: $(l-1)$

✱ If $\frac{\partial E}{\partial \theta_{jk}^{(l-1)}} = \underbrace{\frac{\partial E}{\partial a_k^{(l)}} \cdot \frac{\partial a_k^{(l)}}{\partial z_k^{(l)}}}_{\delta_k^{(l)}} \cdot \underbrace{\frac{\partial z_k^{(l)}}{\partial a_k^{(l-1)}}}_{\theta_{jk}^{(l)}} \cdot \underbrace{\frac{\partial a_k^{(l-1)}}{\partial z_k^{(l-1)}}}_{\sigma'(z_k^{(l-1)})} \cdot \underbrace{\frac{\partial z_k^{(l-1)}}{\partial \theta_{jk}^{(l-1)}}}_{a_{jk}^{(l-2)}}$

new $\delta_k^{(l-1)}$ at prev layer

For training instance $\langle 1, 0 \rangle$: $y = 1$

$$a_1^{(1)} = \sigma(-2 + 6x_1 - 6x_2) = \sigma(4) = 0.982$$

$$a_2^{(1)} = \sigma(1 + 8x_1 - 8x_2) = \sigma(9) = 0.999$$

Output : $a_1^{(2)} = \sigma(2 + 6a_1^{(1)} - 6a_2^{(1)}) = \sigma(2 + 6 \times 0.982 - 6 \times 0.999) = \sigma(1.898) = 0.8691$

$$E = \frac{1}{2}(y - \hat{y})^2 = \frac{1}{2}(1 - 0.8691) = 0.0086$$

Back propagation : (right to left)

$$\delta_1^{(2)} = (y - a_1^{(2)})\, \sigma(z_1^{(2)})\, (1 - \sigma(z_1^{(2)}))$$

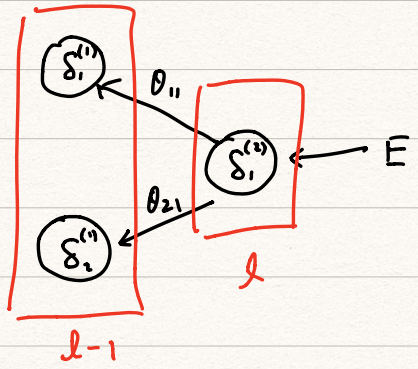
$$= (1 - 0.8691)\, \sigma(1.898)\, [1 - \sigma(1.898)]$$

$$= 0.0149$$

$$\delta_1^{(1)} = \theta_{11}^{(2)}\, \delta_1^{(2)}\, \sigma(z_1^{(1)})\, [1 - \sigma(z_1^{(1)})]$$

$$= 6 \times 0.0149 \times \sigma(4) \times [1 - \sigma(4)]$$

$$= 0.0016$$



$$\delta_2^{(1)} = \theta_{12}^{(2)}\, \delta_1^{(2)}\, \sigma(z_2^{(1)})\, [1 - \sigma(z_2^{(1)})]$$

$$= -6 \times 0.0149 \times \sigma(9) \times [1 - \sigma(9)]$$

$$= -0.00000103$$

$\eta = 0.7$, calculate $\Delta\theta$: bias (at once)

$$\Delta\theta_{01}^{(2)} = \eta\, \delta_1^{(2)}\, a_0^{(1)} = 0.7 \times 0.0149 \times (-1) = -0.0104$$

$$\Delta\theta_{11}^{(2)} = \eta\, \delta_1^{(2)}\, a_1^{(1)} = 0.7 \times 0.0149 \times 0.982 = 0.0102$$

...

...

Update θ:

$$\theta_{01}^{(2)} \leftarrow \theta_{01}^{(2)} + \Delta\theta_{01}^{(2)}$$

$$= -2 + (-0.0104) = -2.0104$$

$$\theta_{11}^{(2)} \leftarrow \theta_{11}^{(2)} + \Delta\theta_{11}^{(2)}$$

$$= 6 + 0.0102 = 6.0102$$

...

This is only one training instance ⇒ do this for all training instances to finish one epoch.

2. What is the difference between "model bias" and "model variance"?
   i. Why is a high bias, low variance classifier undesirable?
   ii. Why is a low bias, high variance classifier (usually) undesirable?

Bias: propensity of a classifier to systematically produce the same errors. $E[g(x) - f(x)]$ (average model approx error over all possible training sets)

If it doesn't produce error / produces different kinds of errors ⇒ unbiased (e.g. predict too many instances as the majority class)

Variance: propensity of a classifier to produce different classifications using different training set. (randomly sampled from same population)

Measure of the inconsistency of the classifier, from training set to training set.

$$E[\{f(x) - E(f(x))\}^2]$$

(i) High bias & low variance

⇒ Consistently wrong.

(ii) Low bias & high variance

low bias ⟶ may be correct predictions.

high variance ⟶ difficult to be certain about the performance of the classifier

If high variance, ER may be low on one set of data, and high on another set (not generalised)

3. Describe how validation set, and cross-validation can help reduce overfitting?

models usually have hyperparameter(s) ⟶ control model complexity

find best values ⟶ to achieve best predictive performance on new data.

(may also consider a range of different types of models ⇒ find best one)

performance on training data : not a good indicator on unseen data.
(may be overfitting)

2 ways :

① Validation set: we train models on training set, compare them on independent data (val set) ⇒ select best one.
evaluate the final model with test set.

② CV: If data is limited & want good models
⇒ use as much of the available data as possible for training.
⇒ small validation set ⇒ Use CV