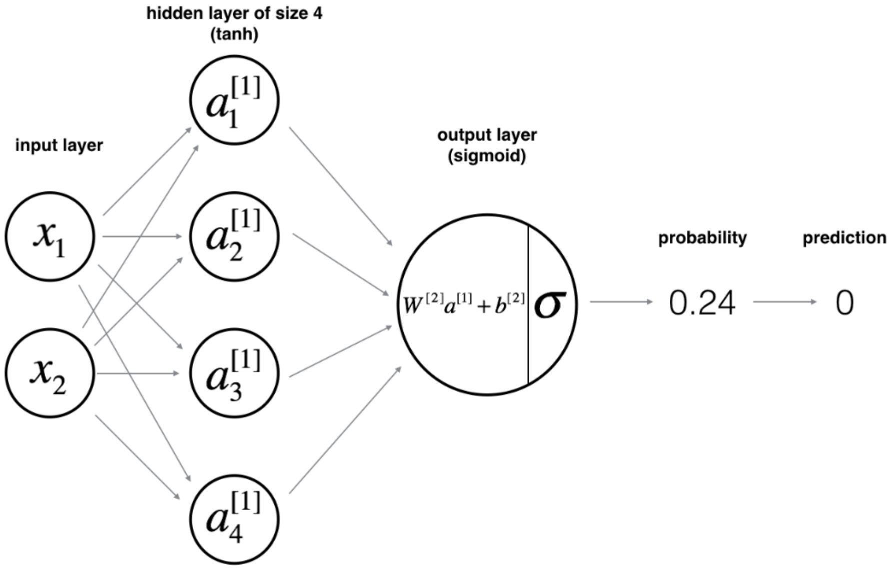
Neural Network model for week 3 – Planar data classification with one hidden layer

The model:



The general methodology:

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

Helper functions compute steps 1-3, and then merged into one function (nn_model)

Mathematically:

Forward propagation:

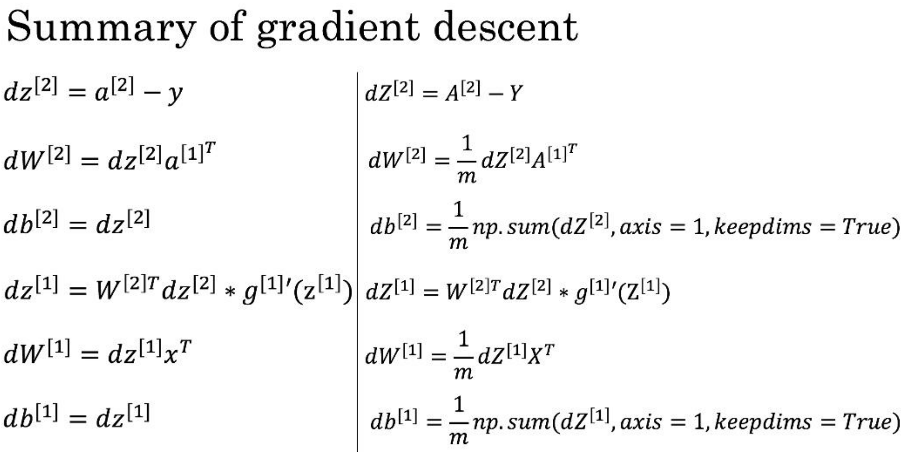
For one training example $x^{(i)}$:

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

Given the prediction to all examples, compute the cost J:

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

backward propagation:



Summary of gradient descent

$dz^{[2]} = a^{[2]} - y$	$dZ^{[2]} = A^{[2]} - Y$
$dW^{[2]} = dz^{[2]} a^{[1]^T}$	$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]^T}$
$db^{[2]} = dz^{[2]}$	$db^{[2]} = \frac{1}{m} np.sum(dZ^{[2]}, axis = 1, keepdims = True)$
$dz^{[1]} = W^{[2]T} dz^{[2]} * g^{[1]\prime}(\mathrm{z}^{[1]})$	$dZ^{[1]} = W^{[2]T} dZ^{[2]} * g^{[1]\prime}(\mathrm{Z}^{[1]})$
$dW^{[1]} = dz^{[1]} x^T$	$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$
$db^{[1]} = dz^{[1]}$	$db^{[1]} = \frac{1}{m} np.sum(dZ^{[1]}, axis = 1, keepdims = True)$

$NOTE$: $g^{[1]} = activation\ function\ of\ layer\ 1 = \tanh()$

$$g^{[1]'} = 1 - \left(g^{[1]}\right)^2$$

Update parameters:

$$W1 \coloneqq W1 - training\ rate * dW1$$
$$b1 \coloneqq b1 - training\ rate * db1$$
$$W2 \coloneqq W2 - training\ rate * dW2$$
$$b2 \coloneqq b2 - training\ rate * db2$$

Predictions:

$$y_{prediction} = \begin{cases} 1 & activation > 0.5 \\ 0 & otherwise \end{cases}$$