**Neural Network – Non-linear Hypothesis**

To get interesting, complex and non-linear hypothesis

|  |
| --- |
| **Model Representation: logistic unit** |
| where x0 = bias unit = 1  Sigmoid (logistic) activation function g(z)  Theta = parameters = weights |
| **Input layer 🡪 hidden layer (Learned feature) 🡪 output layer** |
| = ‘activation’ of unit i in layer j  = matrix of weights/parameters controlling function mapping from layer j to layer j+1        = size(3\*4) = #features in layer j \* # activation/hypothesis in layer j+1  **Intuition:**  If network has units in layer j, units in layer j+1, then will be \* |
| **Vectorized Computation** |
| Set x = a(1)  **H(x) = z(j+1) = g(z(j+1)) = g( theta(j)** (( \* ) \* **a(j)** (, 1)  is number of feature  is the number of hypothesis  a is a vector of size #feature |
| **Intuition** |
| Exact same process in logistic regression |

**Example of Neural Network**

|  |
| --- |
| **Computing Logical Operation with Neural Network** |
| **AND**        **OR**    **Negation**    **XNOR** |

Multi-classification with neural network

|  |
| --- |
| **Example: Computer Vision** |
| https://d3c33hcgiwev3.cloudfront.net/imageAssetProxy.v1/9Aeo6bGtEea4MxKdJPaTxA_4febc7ec9ac9dd0e4309bd1778171d36_Screenshot-2016-11-23-10.49.05.png?expiry=1535414400000&hmac=l9qw5YI2lZayVL_wBa81CqixX9mgewHLZDMHhJ1VIeo |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |