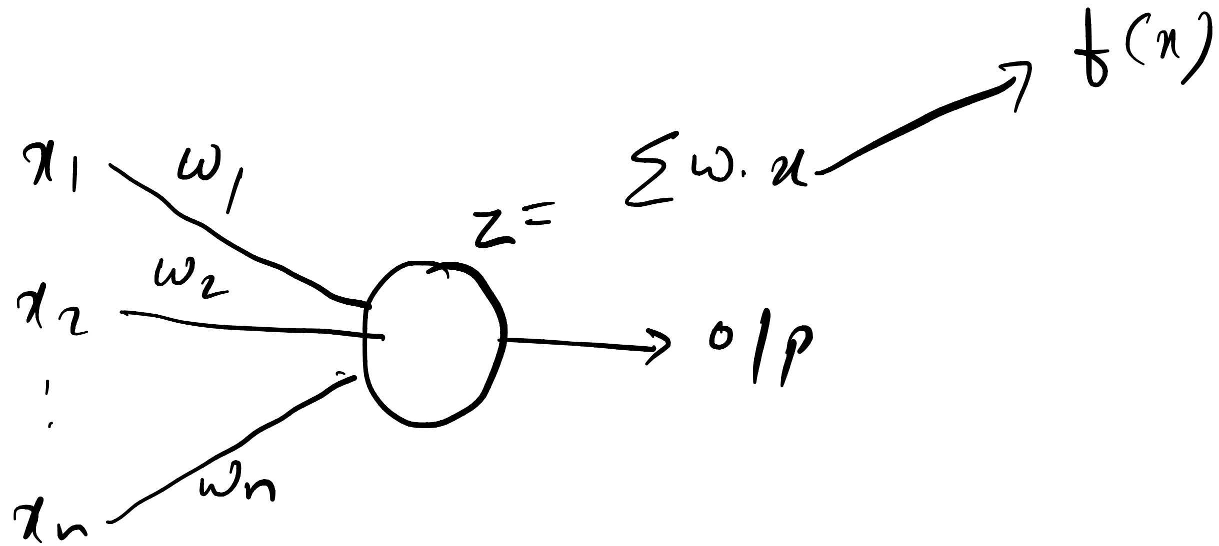
Neural Networions

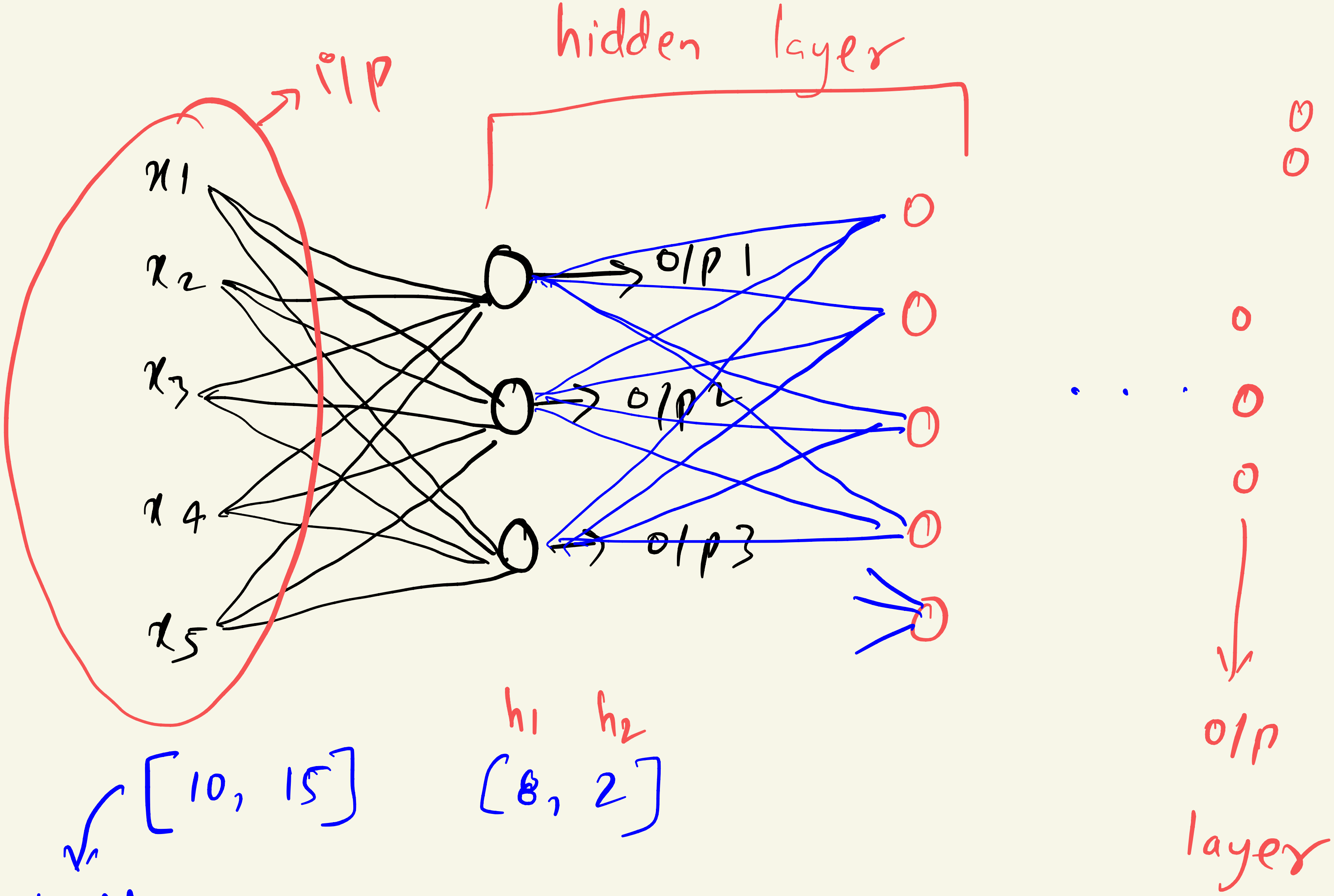
Percep, Logis res.

Need

more than one perceptron

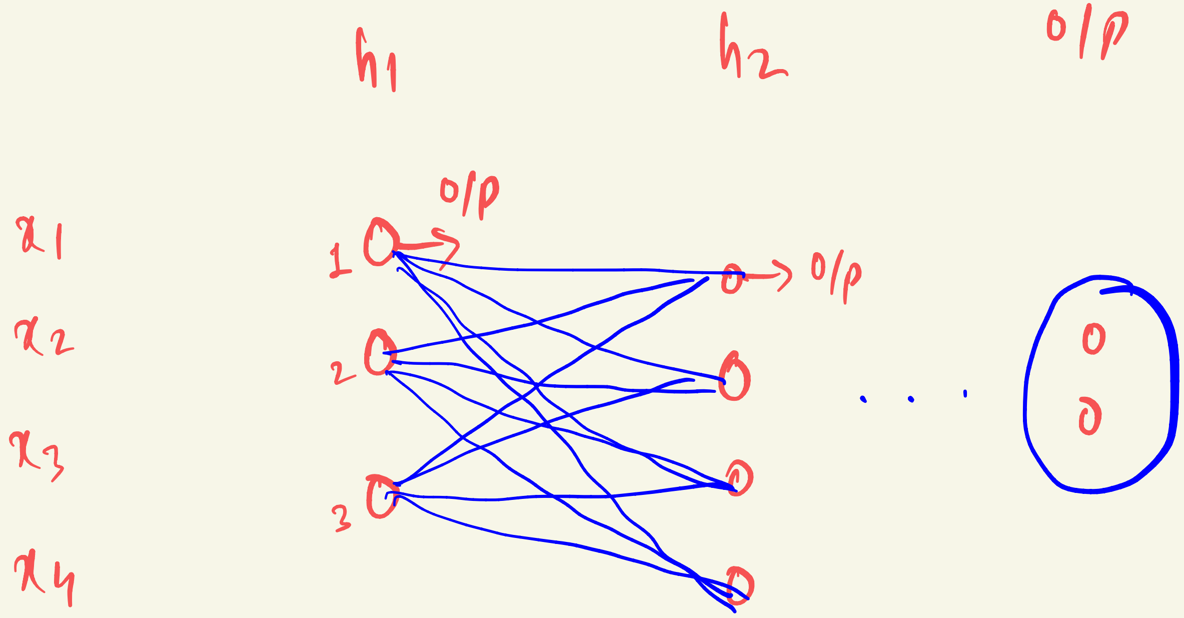
Single percept





hidden

i/p layer
[Hidden layer]
o/p layer



$$o/p\ h_{1_1} = A^{h_1}\left(\sum x_1 w_{h_1}^{1}\right)$$

$$o/p\ h_{1_2} = A^{h_1}\left(\sum x_1 w_{h_1}^{2}\right)$$

$$o/p\ h_{1_3} = A^{h_1}\left(\sum x_1 w_{h_1}^{3}\right)$$

$$o/p\ h_{2_1} = A^{h_2}\left(\sum o/p\ h_1 \cdot w_{h_2}^{1}\right)$$

$$o/p\ h_{2_2} = A^{h_2}\left(\sum o/p\ h_1 \cdot w_{h_2}^{2}\right)$$

$$\vdots$$

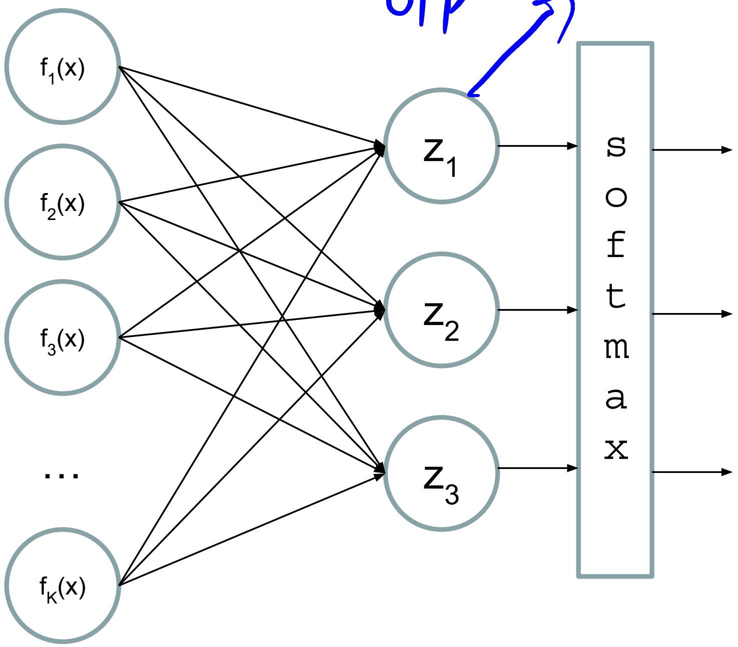
Multi-class Logistic Regression

No hidden layer

- = special case of neural network

$z = \Sigma f(x) \cdot w$

O/P



$$P(y_1|x;w) = \frac{e^{z_1}}{e^{z_1}+e^{z_2}+e^{z_3}}$$ ← multiclass

$$P(y_2|x;w) = \frac{e^{z_2}}{e^{z_1}+e^{z_2}+e^{z_3}}$$

$$P(y_3|x;w) = \frac{e^{z_3}}{e^{z_1}+e^{z_2}+e^{z_3}}$$

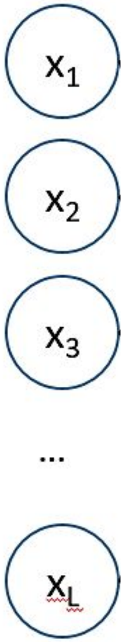
softmax

I/P

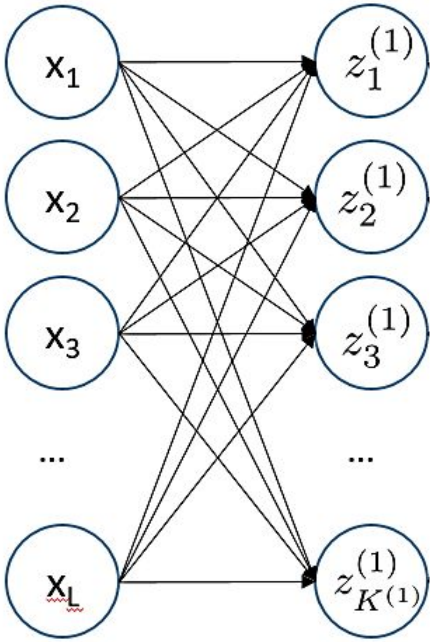
binary ⇒ Sigmoid

Deep Neural Network = Also learn the features!

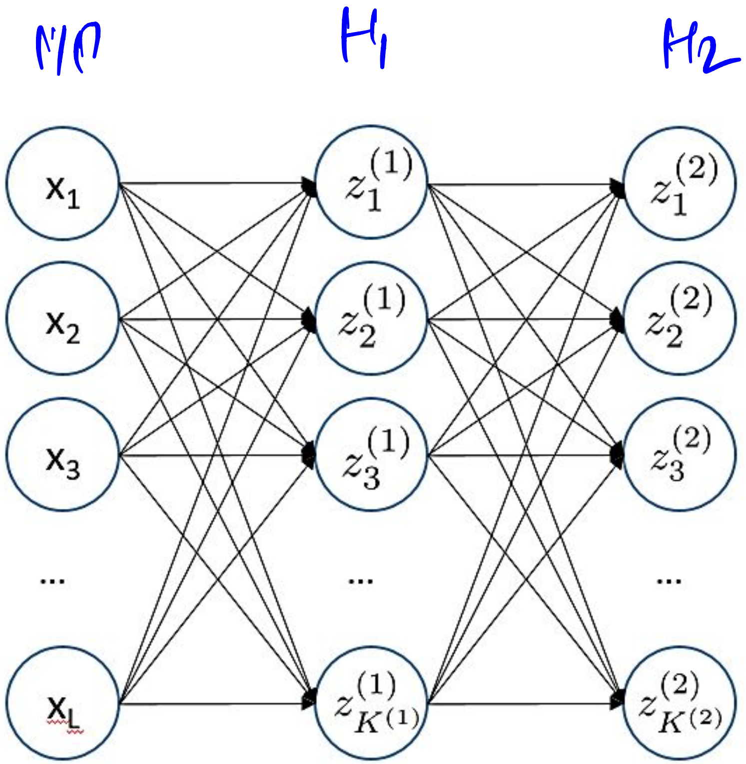
i/p Layer



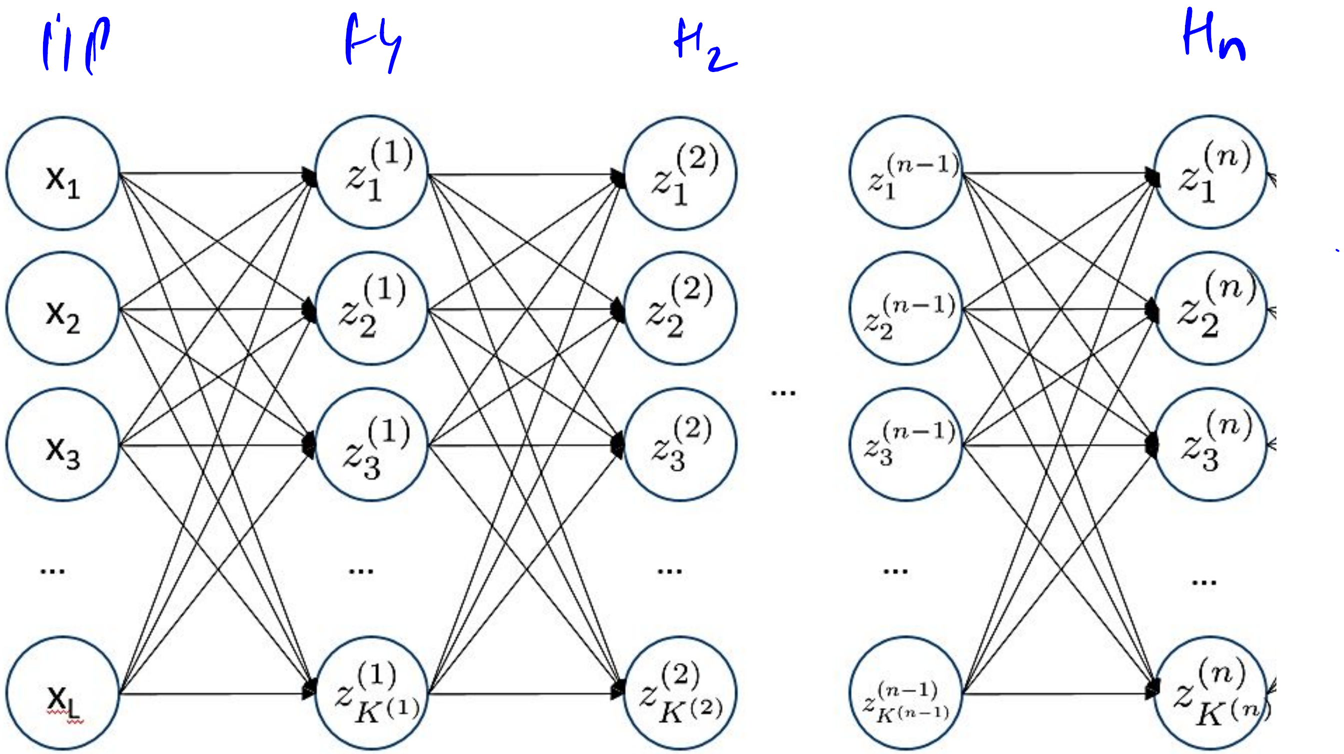
Deep Neural Network = Also learn the features!



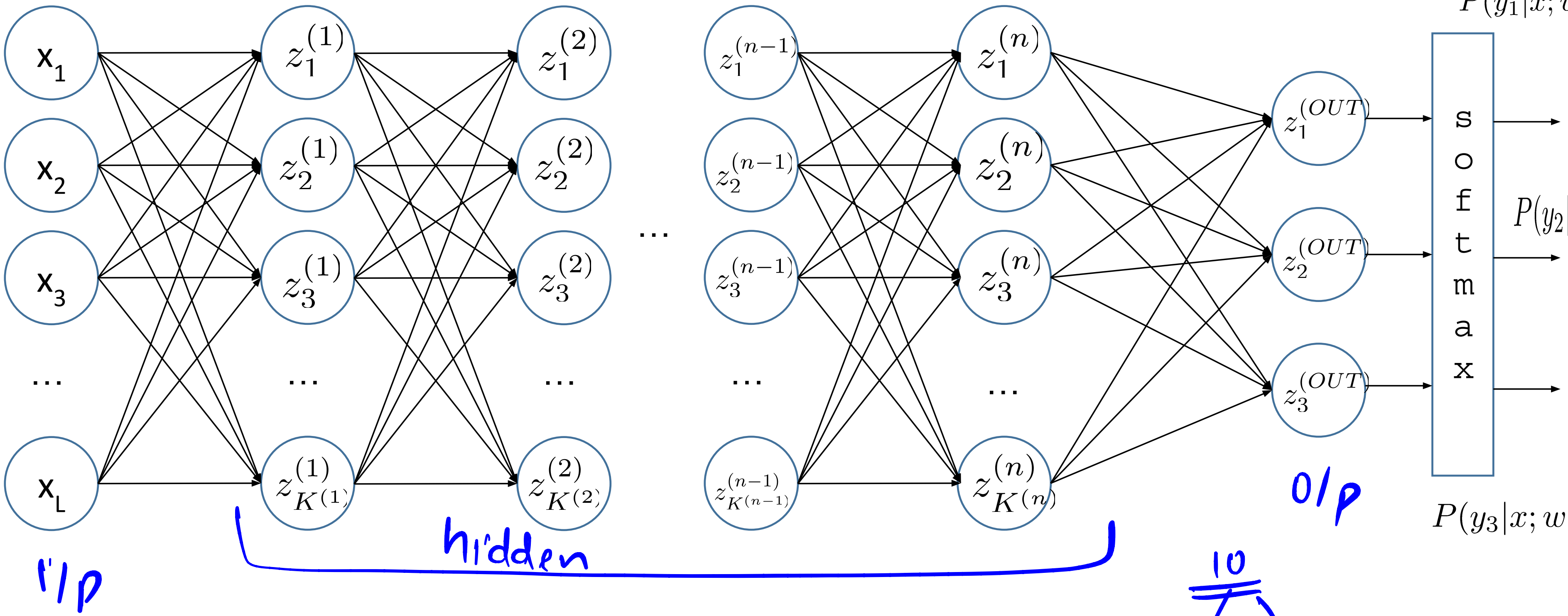
Deep Neural Network = Also learn the features!



Deep Neural Network = Also learn the features!



Deep Neural Network = Also learn the features!



$$P(y_1|x;w) = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$P(y_2|x;w) = \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$P(y_3|x;w) = \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

10

8 2

$$z_i^{(k)} = g\left(\sum_j W_{i,j}^{(k-1,k)} z_j^{(k-1)}\right)$$

g = nonlinear activation function

20x20 = 400

→ forward propagatⁿ

→ backward propagatⁿ

→ overfitting

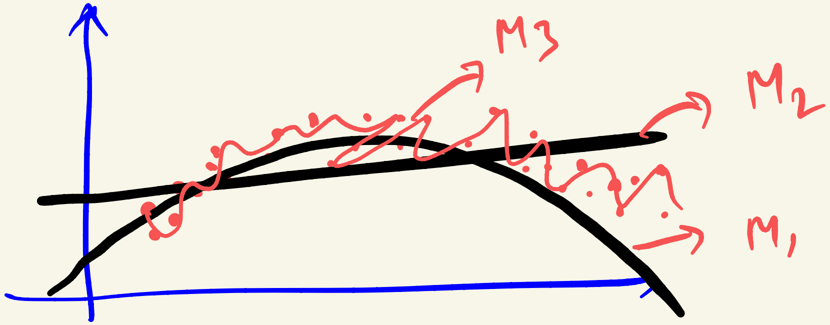
→ underfitting

→ Training data

→ testing data

→ epoch

→ mini-batch



overfi → M3

underfi → M2

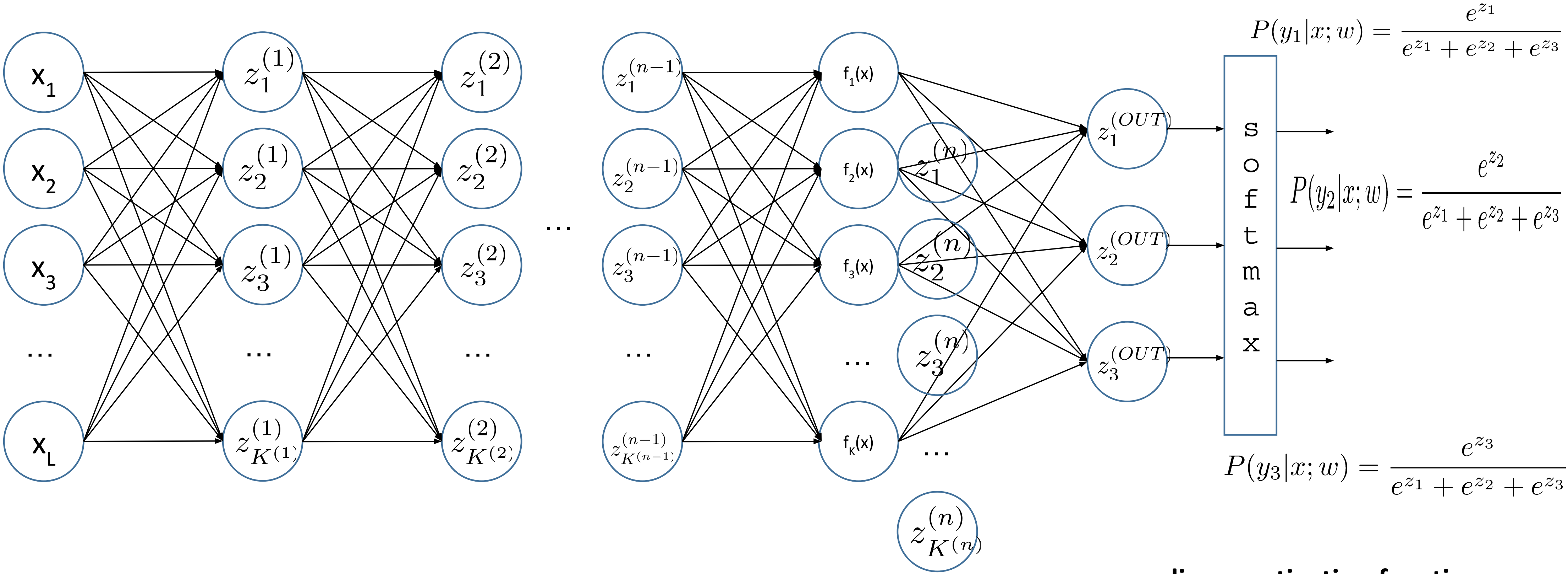
Acc
Tr ⇒ high ⇒ 80

Acc
Ts ⇒ low ⇒ 40

Acc
Tr ⇒ low

Test ⇒ low

Deep Neural Network = Also learn the features!

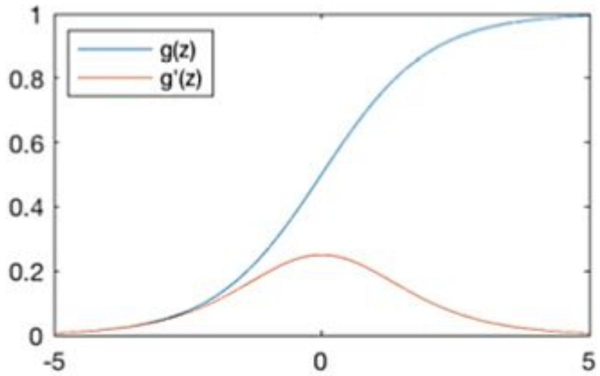


g = nonlinear activation function

$$z_i^{(k)} = g\Big(\sum_j W_{i,j}^{(k-1,k)} z_j^{(k-1)}\Big)$$

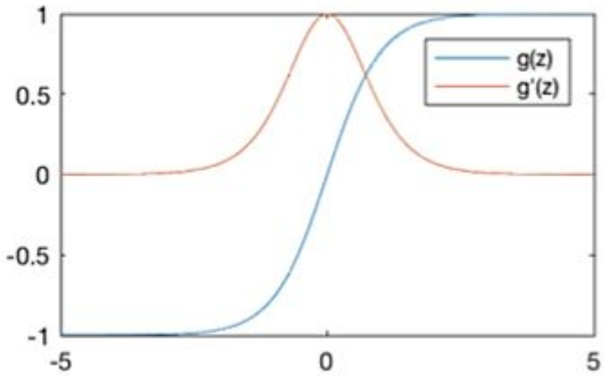
Common Activation Functions

Sigmoid Function



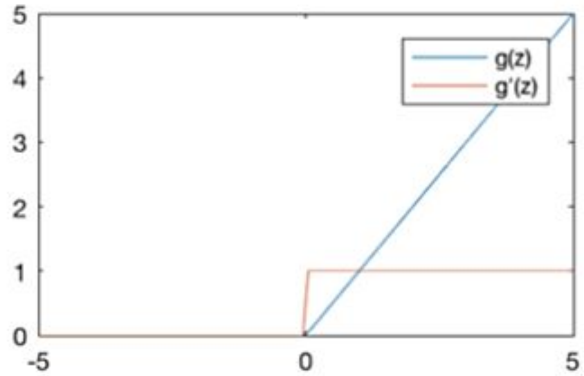
$$g(z) = \frac{1}{1 + e^{-z}}$$

Hyperbolic Tangent



$$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

Rectified Linear Unit (ReLU)



$$g(z) = \max(0, z)$$

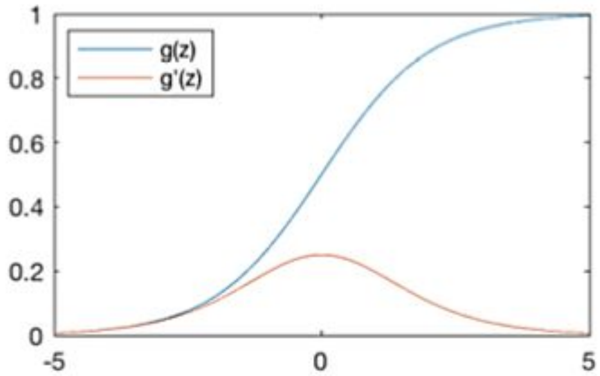
Find derivative.

exploding gradi

vanishing gradi

Common Activation Functions

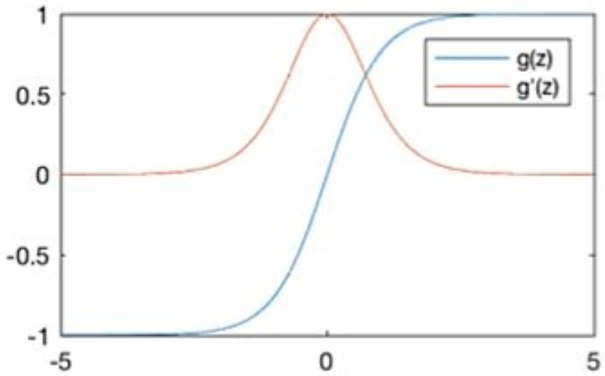
Sigmoid Function



$$g(z) = \frac{1}{1 + e^{-z}}$$

$$g'(z) = g(z)(1 - g(z))$$

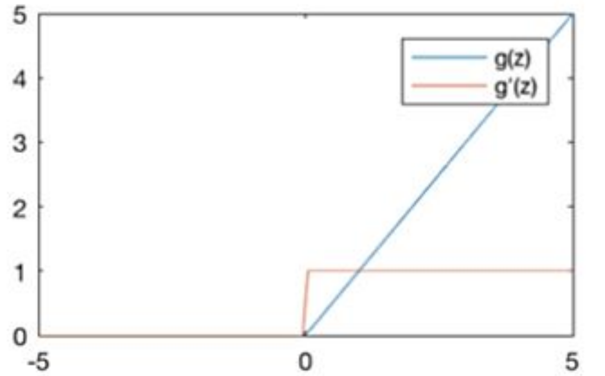
Hyperbolic Tangent



$$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

$$g'(z) = 1 - g(z)^2$$

Rectified Linear Unit (ReLU)



$$g(z) = \max(0, z)$$

$$g'(z) = \begin{cases} 1, & z > 0 \\ 0, & \text{otherwise} \end{cases}$$

Deep Neural Network: Also Learn the Features!

- Training the deep neural network is just like logistic regression:

$$\max_{w} \quad ll(w) = \max_{w} \quad \sum_{i} \log P(y^{(i)} | x^{(i)}; w)$$

just w tends to be a much, much larger vector ☺

 just run gradient ascent

Neural Networks Properties

- Theorem (Universal Function Approximators). A two-layer neural network with a sufficient number of neurons can approximate any continuous function to any desired accuracy.

- Practical considerations
  - Can be seen as learning the features

  - Large number of neurons
    - Danger for overfitting
    - (hence early stopping!)

Neural Networks

https://cs.stanford.edu/people/karpathy/convnetjs/

Neural Networks