Hidden Units

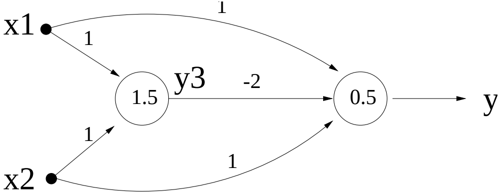
COMP4211

THE DEPARTMENT OF
COMPUTER SCIENCE & ENGINEERING
計算機科學及工程學系

Back to XOR

x_1	x_2	y_3	$y = XOR(x_1, x2)$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- recall that a single perceptron cannot solve the XOR problem
- but adding a hidden unit can solve the XOR problem



Multi-layer Feedforward Networks

- Generalization of simple perceptrons
- Multi-layer perceptrons (MLP)

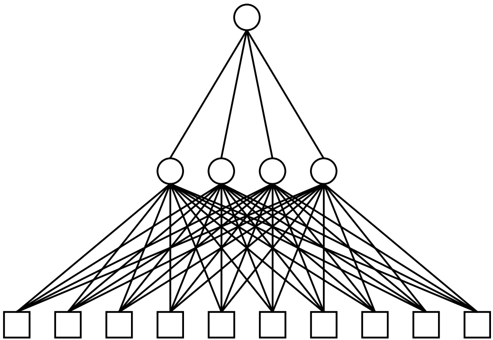
Output units O_i

$W_{j,i}$

Hidden units a_j

$W_{k,j}$

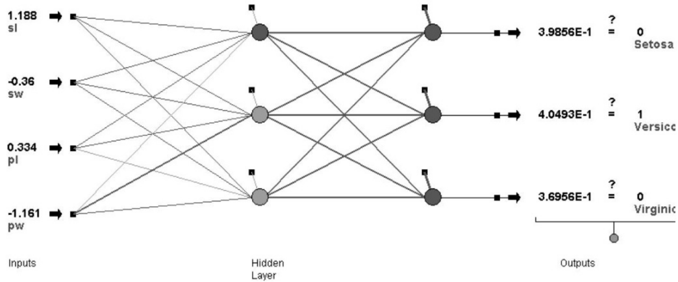
Input units I_k



ANN for Classification

Multiple classes

- one output for each class



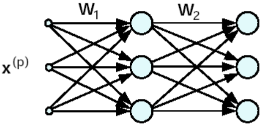
- assign object to the class $\arg\max_{i=1}^{m} y_i$

two classes

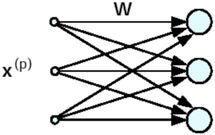
- treat like multiple classes, or
- only one output unit y:
  assign object into *yes* class if $y > 0$; *no* class if $y \leq 0$

Hidden Unit Transfer (Activation) Function

if hidden units were linear elements, then a single-layer neural network with appropriately chosen weights could exactly duplicate those calculations performed by any multi-layer network



$$
\begin{aligned}
y^{(p)} &= W_2 W_1 x^{(p)} \\
&= (W_2 W_1) x^{(p)}
\end{aligned}
$$



- the capabilities of MLP stem from the nonlinearities used within the hidden units

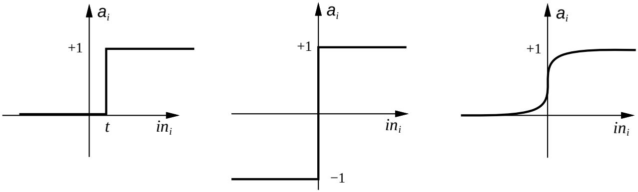
use the perceptron as the hidden unit?

- transfer function: step function

non-differentiable $\rightarrow$ unsuitable for gradient descent

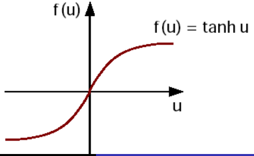
Sigmoid Unit

- a unit very much like a perceptron, but based on a smoothed, differentiable threshold function: $\boxed{\sigma(x) = \frac{1}{1+e^{-x}}}$

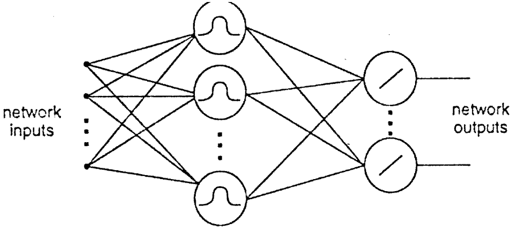


(a) Step function **(b) Sign function** **(c) Sigmoid function**

- nice property for sigmoid: $\boxed{\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x))}$
- the tanh is also sometimes used in place of the sigmoid function

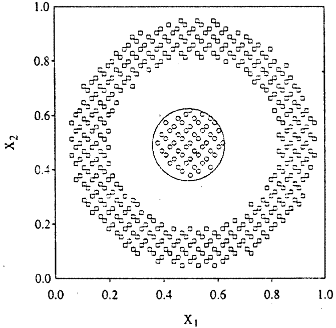


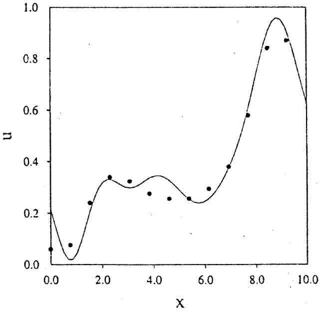
Radial Basis Functions (RBF) Network



- e.g. Gaussian: $\exp\left(-\frac{(\mathrm{x}-\mathrm{w}_j)^T(\mathrm{x}-\mathrm{w}_j)}{2\sigma_j^2}\right)$
  - radially symmetric $\Rightarrow$ radial basis function
- each hidden unit produces a localized response to the input
  - significant nonzero response only when input falls within a small localized region of the input
- cf sigmoid: nonzero over an infinitely large region of the input space

RBF Network...

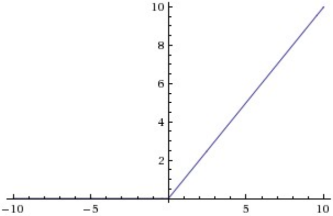




- some problems can be solved more efficiently with sigmoidal hidden units, other are more amenable to RBF units

Rectified Linear Unit (ReLU)

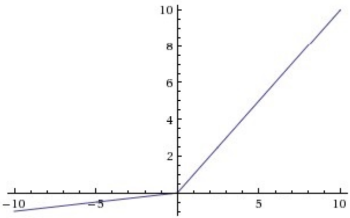
- $f(x) = \max(0, x)$



- the most popular activation function for deep networks
- efficient computation
- simple gradient
  - if > 0, gradient $= 1$
  - if ≤ 0, gradient $= 0$
- sparse activation (hidden units with non-zero outputs)

gradient can be 0!

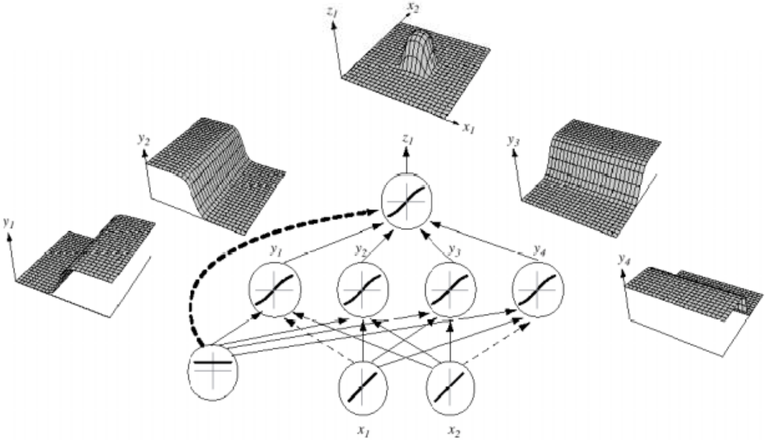
A Variant: Leaky ReLU



- as computationally efficient as standard ReLU
- but will not "die"

Universal Approximation

only one layer of sigmoid hidden units suffices to approximate any well-behaved function to arbitrary precision



Universal Approximation...

network with > 2 layers also have universal approximation property

why need networks with > 2 hidden layers?

- by using extra layers we might find a network with fewer weights in total while still achieving the same level of accuracy

How to determine network structure?