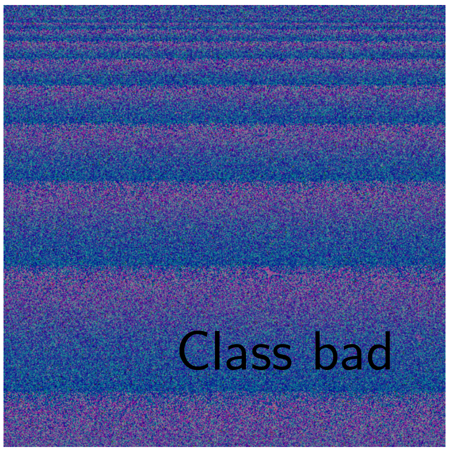
EXAMPLE OF ATTRIBUTE SELECTION 1



- Which attributes (features) are important in telling the difference between bad and good?
  1. Compute a set of potentially useful features for each image
  2. Perform feature selection using all weka methods
  3. The most frequently occurring features can be associated with aesthetic value.

EXAMPLE OF ATTRIBUTE SELECTION 2

Feature	Description
F02	Earth Mover Distance from unsaturated grey (Colourfulness)
F01, F03 - F07	Average hue, saturation, brightness on all pixels and the pixels in the centre of the image
F08 - F19	Various wavelet functions used to compute levels of smoothness on different scales
F20 - F21	Image dimensions (width+height, width/height)
F22	The number of contiguous regions based on colour similarity larger than 1/100th of the total number of pixels in the image
F23 - F37	Average hue, saturation and brightness for each of the 5 largest contiguous regions of similar colours
F38 - F42	Size in pixels of each of the 5 largest regions of similar contiguous colours divided by the total number of pixels in the image
F43 - F44	Two variations on the measure of complimentary colours
F45 - F49	The location in the image of the centre of each of the 5 largest contiguous regions of similar colours
F50 - F52	Depth of field effect (emulating telephoto lens zoom) on each of the hue, saturation and brightness channels

MOST IMPORTANT ATTRIBUTES

CFS	Gain Ratio	Info Gain	OneR	Relief	Symmetric Uncert	Wrapper
F01	F30	F25	F25	F04	F30	F04
F04	F29	F41	F41	F07	F45	F18
F07	F27	F40	F26	F41	F39	F21
F13	F34	F39	F39	F25	F29	F31
F16	F28	F42	F40	F24	F35	F36
F25	F45	F44	F38	F03	F42	F41
F30	F33	F37	F31	F06	F27	
F37	F12	F43	F01	F01	F34	
F39	F39	F45	F04	F13	F37	
F40		F38	F07	F16	F25	
F42						
F45						

Number of occurrences

5 F39 5 F25 4 F41 4 F04 3 F45 3 F42 3 F40 3 F37 3 F30 3 F07 3 F01

- Golden Nugget. Only colour features are being used, no texture features.

EXAMPLE OF ATTRIBUTE SELECT

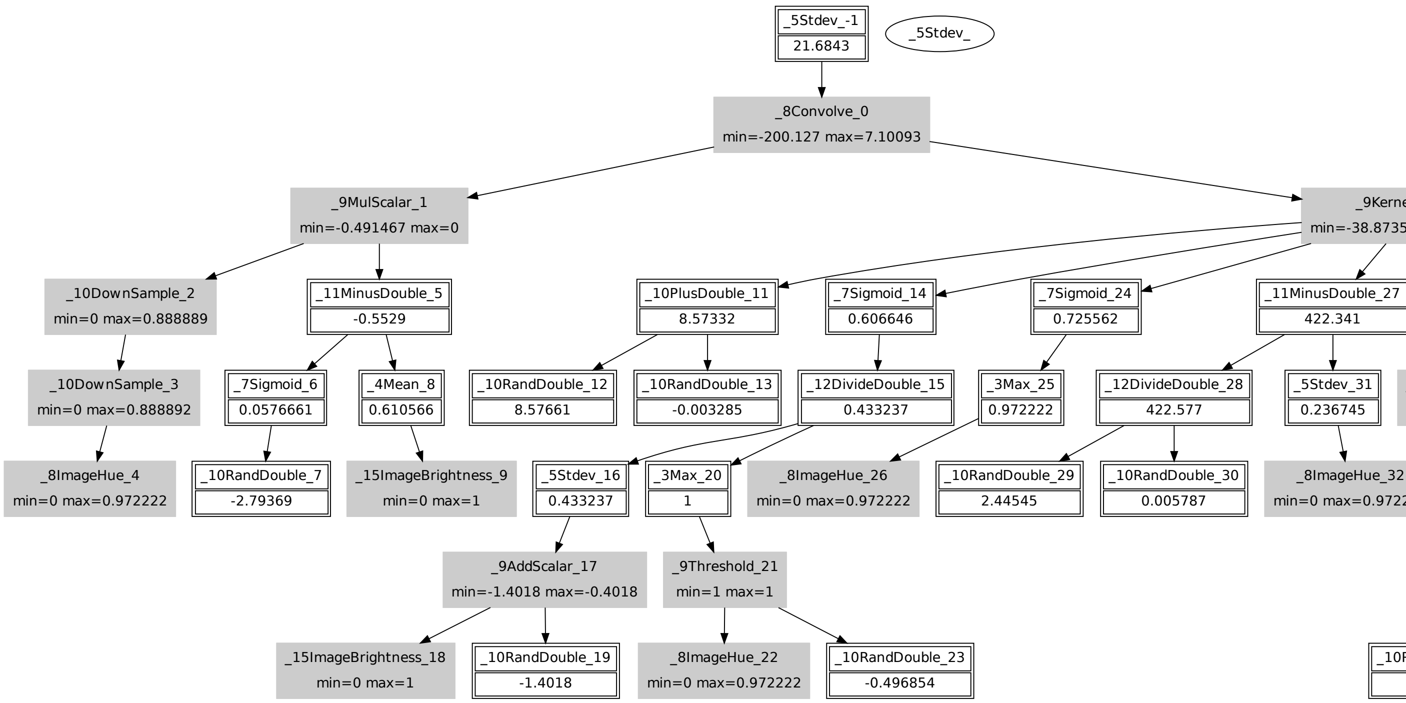
Classification accuracy of selected feature

Classifier	Full	CFS	GainRatio	InfoGain	OneR	Re
OneR	72	71	71	72	72	7
J48	87	88	75	88	87	8
Random Forest	91	92	77	88	88	9
SMO	89	83	55	79	79	8

Classification accuracy of EVOLVED featu

Classifier	Accuracy
OneR	80.3%
J48	89.7%
Random Forest	91.8%
SMO	89.9%

AN EVOLVED FEATURE



PREPARATION OF DATA FOR DATA MINING

- This lecture will be based on the files in:
  /KDrive/SEH/SCSIT/Students/Courses
  /COSC2111/DataMining/

  – data/parking_duration_of_parking_event_vs_street_ID.csv

    This is a file of 12,208,179 parking events in the city of Melbourne.

  – data/parking-small.csv

    This is a random subset of 10,000 events

  – code-and-scripts/parking-time.sh

    This is a script for taking the arrival date-time and generating useful features for data mining.

- Some typical data Arrival Time
  24/08/2012 11:34
  17/03/2012 13:07
  7/12/2011 19:50
  3/03/2012 14:36
  29/1/2012 12:26

PREPARATION OF DATA FOR DATA MINING

- Very bad
  - EXCEL or other spreadsheet
  - Your favourite editor
  - Interactive manual steps
- Why?
  - The procedure always needs to be done several times
  - Repeated manual steps introduce error
  - Little value in learning from erroneous data
- Very good
  - Data preparation script that can be executed repeatedly
    and independently verified for correctness
  - Uses mature utility programs

UNIX TOOLS FOR DATA MINERS

- Minimum requirement
  - cat, head, tail, cut, grep, pr, paste, sort, uniq, tr
  - Substitution with sed
  - Basic shell scripting
- To be an expert
  - Regular expressions
  - Advanced sed
  - Advanced shell scripting
  - awk or perl or python
- On a windows PC, install CYGWIN or equivalent

  Windows 10 has Ubuntu
  Mac has terminal

UNIX AND XWINDOWS ON UNIX SERVERS

1. Read the basic unix guide (Canvas week 6)

2. Use putty with X connection

3. On RMIT servers run xeyes to verify X connection

4. On RMIT servers start xclock to avoid timeout

5. putty demo

Important Unix tools

`cat file1 file2 file3`
Concatenate files to standard output

`head -n 100 file`
Send the first 100 lines to stdout

`tail -n 10 file`
Send the last 10 lines to stdout

`cut -d',' -f7 file`
Send col 7 to stdout

`sed -e's/from/to/'|`
Stream editor: replace first occurrence in a line $from$ with to

`sed -e's+from+to+g'`
replace all occurrences $from$ with to

`paste -d, col1 col2 col3`
merge lines of files col1 col2 col3

`fgrep str file`
Get regular [fixed] expression
Send only lines that contain str to stdout

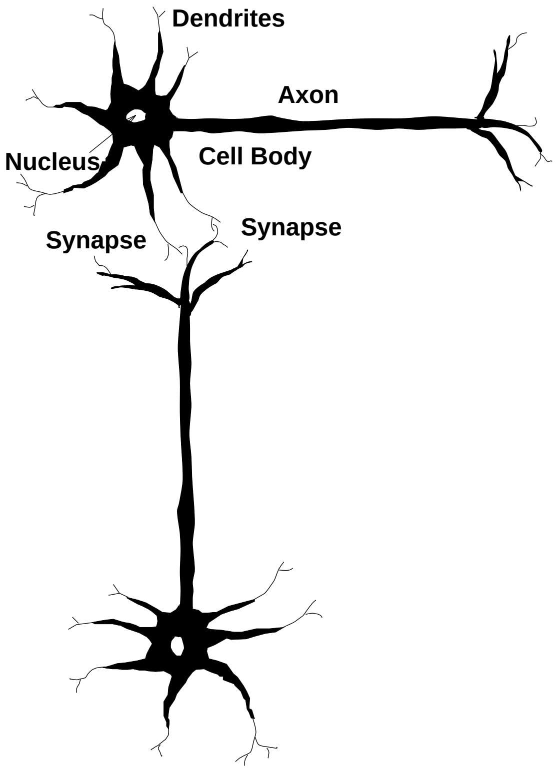
`egrep -e'RE' file`
Send only lines that contain the regular expressio
RE to stdout

```
tr ' ' ',' file
Translate characters
Change all occurrences of space to comma

tr -d'\r' file
Delete all occurrences of the return character

sort file
Sort

uniq file
Omit or count repeated lines

wc -l file
Word count, count lines (-l)
```

NEURAL NETWORKS SUMMARY

1. Introduction
2. Biological origins
3. Computational neuron
4. Overview of architectures
5. Feed forward networks
6. Training of networks (JavaNNS Package)
7. Data encoding/preparation

BIOLOGICAL NEURON



- The neuron receives impulses (signals) from other neurons via dendrites
- The neuron sends impulses to other neurons via the axon
- Input at dendrite, output from axon
- Synapse: dendrite of one neuron and axon of another
- Impulses cause neurotransmitters (chemicals) to diffuse across the synapse
- Enhance (excite) or inhibit
- Action adjusted (by learning?)

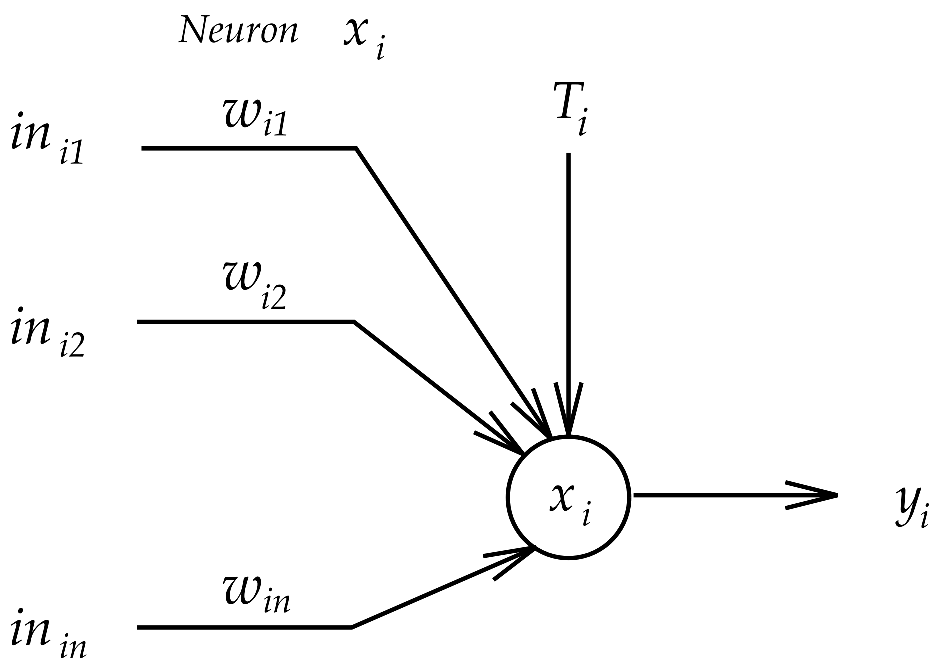
CEREBRAL CORTEX

- Contains 10^{11} neurons = no. stars in Milky Way
- Neurons massively connected
- Each neuron is connected to 10^3 to 10^4 other neurons
- Much more complex and dense than telephone network
- Brain contains 10^{14} to 10^{15} connections
- Brain message passing is 1,000,000 times slower than modern electronic circuits
- A complex decision like recognizing a face takes a few hundred milliseconds
- Operational speed of neurons is a few millisecon
- Thus computations cannot take more than 100 serial stages
- One hundred step rule

KINDS OF ARTIFICIAL NEURAL NETWORKS

- There are a very large number of network types
  - Hopfield
  - Hebbian
  - Recurrent
  - Radial Basis Functions
  - Kohonen self organizing map
  - ...
- We look only at the most frequently used types
  - Feed forward multi-layer perceptron with
  - with logistic OR linear threshold units
  - trained by backward error propagation

ARTIFICIAL NEURON



- Computation carried out by the neuron

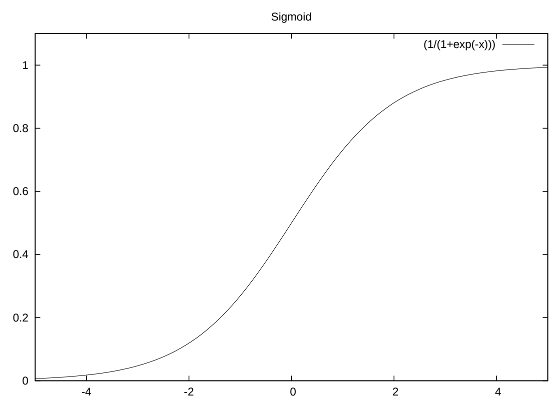
$$x_i = \sum_{j=1}^{n} w_{ij} in_{ij} + T_i$$

$$y_i = transferfunction(x_i)$$

- Transfer function is usually non-linear
- T_i
  - Threshold
  - also called the bias
  - sometimes written w_0

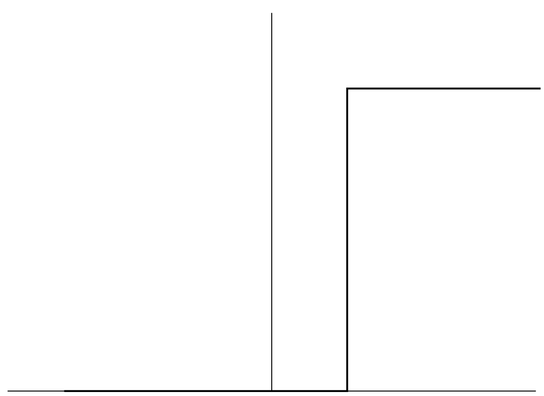
COMMON TRANSFER FUNCTIONS

Sigmoid/Logistic



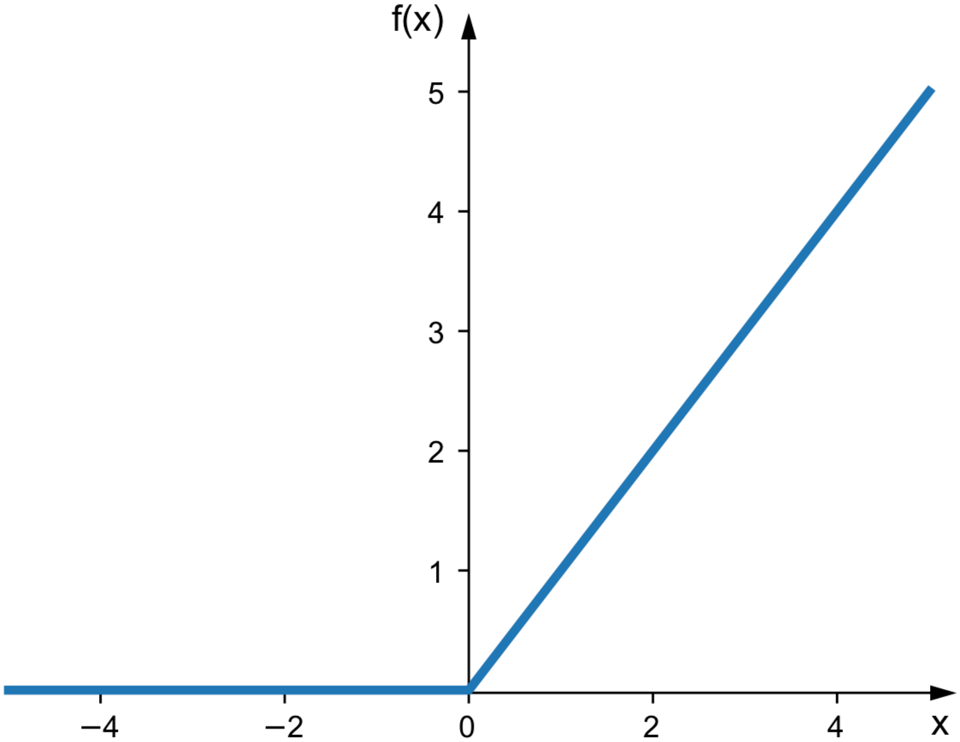
$$y_i = \frac{1}{1+e^{-kx_i}}$$

Threshold



$$if \;\; x < t \;\; then \;\; 0 \;\; else \;\; 1$$

DEEP NETWORK ACTIVATION FUNCTION



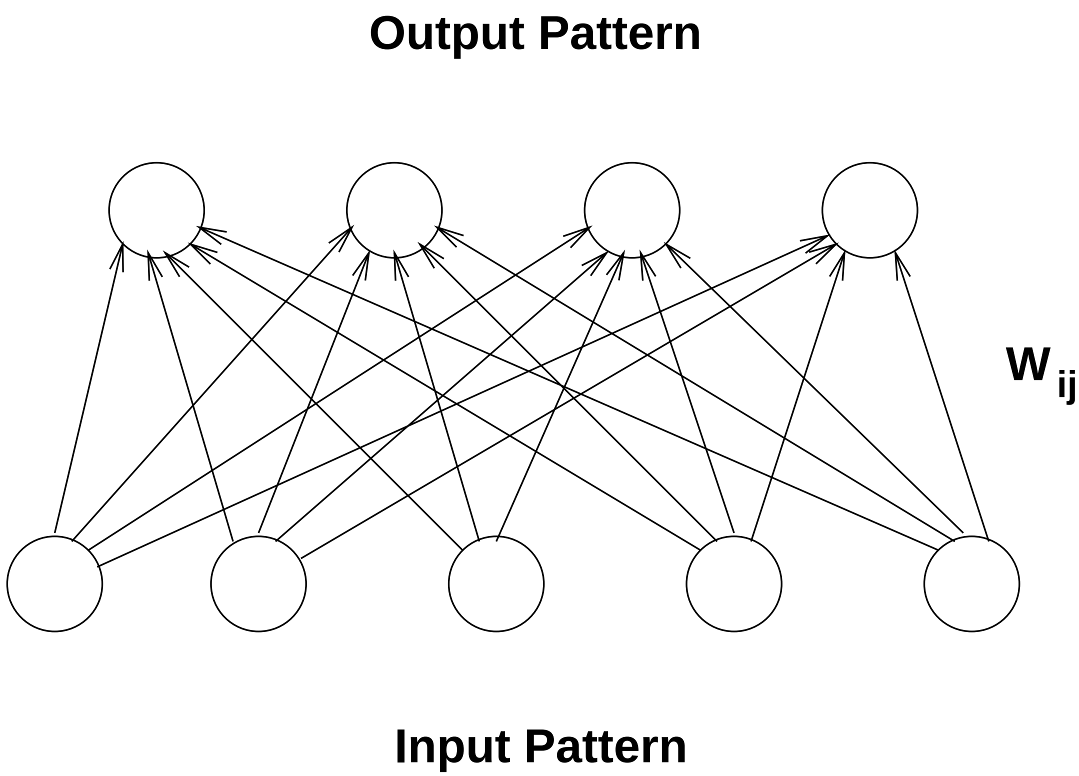
ReLU (Rectified linear Unit)

$$y_i = 0 \ for \ x \leq 0$$

$$y_i = x \ for \ x < 0$$

https://sebastianraschka.com/faq/docs/relu-derivative.html

ANN ARCHITECTURE 1
(Non data mining application)



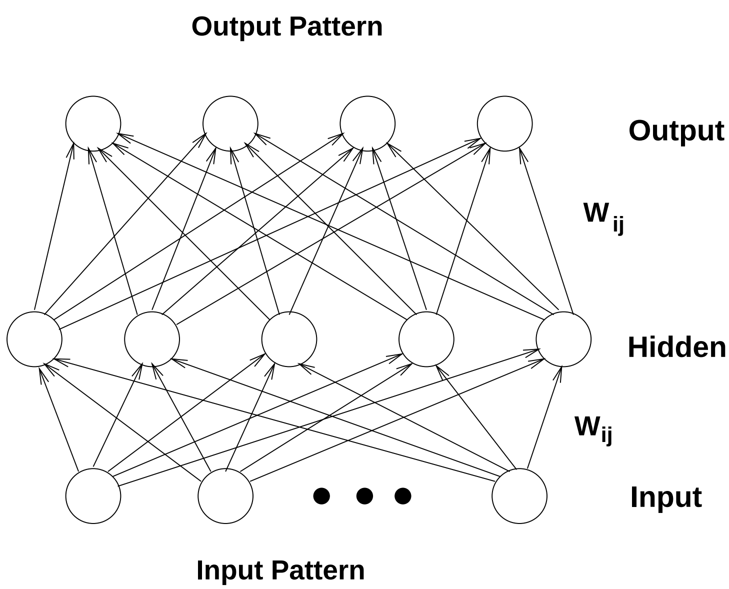
- Two layer network [One layer of weights]
- Could be used for associative memory
- Encodes ($(A_1, B_1), (A_2, B_2), ...(A_k, B_k)$
  - Put in a picture of a person, get out a name
  - Put in a partial/smudged picture, get out the full, clean picture
  - Put in a noisy audio signal, get out the clean sound
- Neural networks are particulary good at dealing with noisy, erroneous or incomplete patterns.

ANN ARCHITECTURE 2

- Feed Forward Network

- Can operate as a pattern classifier

Digits in Postcode	Ascii Text	Underwater Object	Weather Predicition

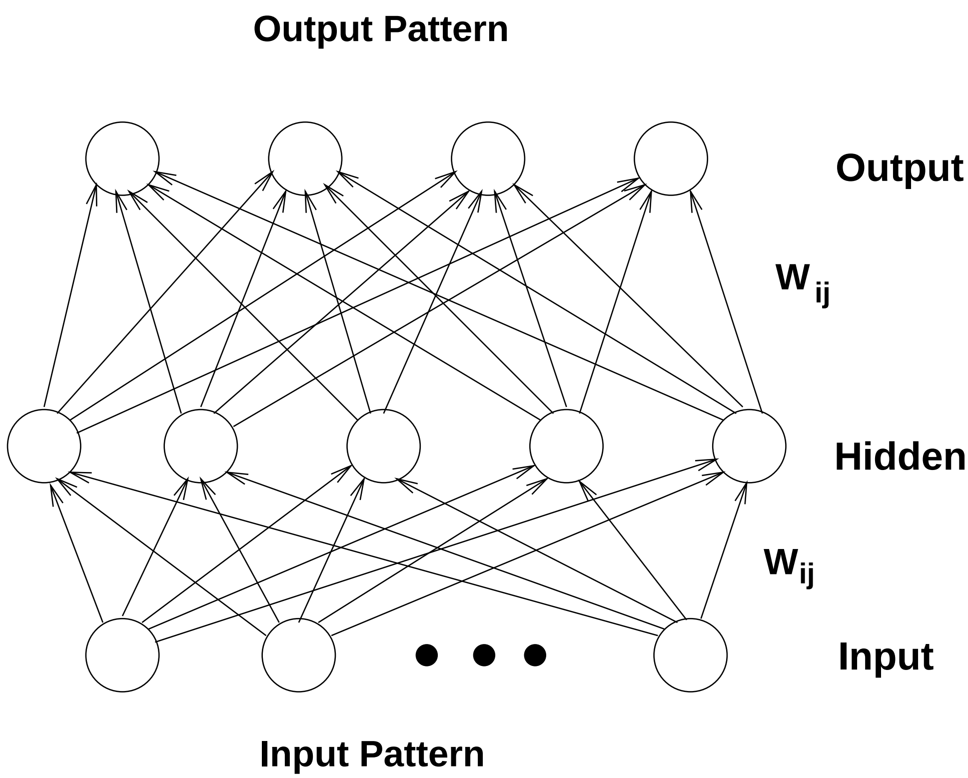


Picture of an Envelope	Speech Waveform	Sonar Signal	Weather Data

ANN ARCHITECTURE 3

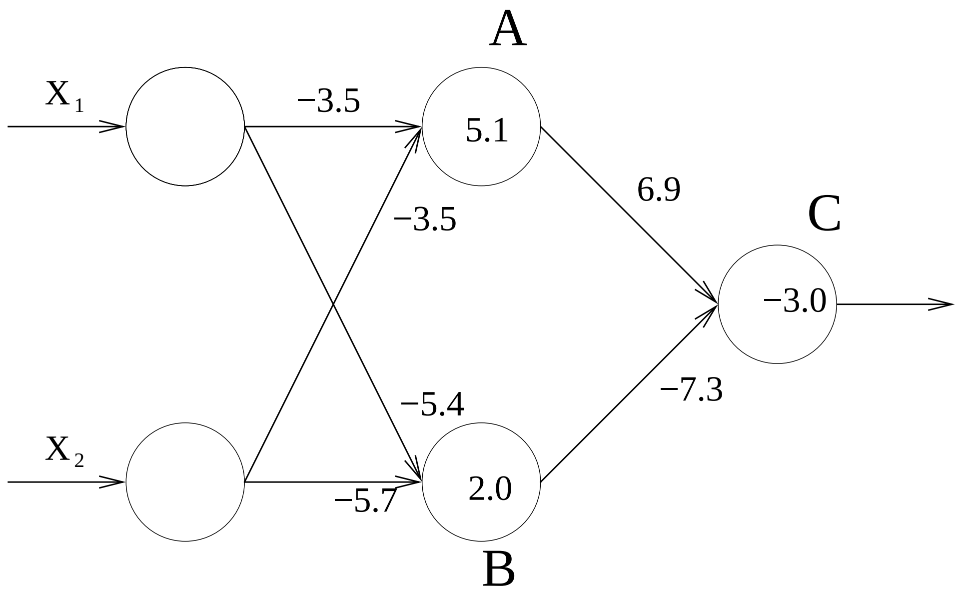
- Feed Forward Network
- Function Approximator/Time series predictor

Prediction of activity	Prediction of stock price	Survival months	Amount Rain



Sunspot Time Series	Stockmarket Time Series	Heart attack Data	Weather Data

NETWORK EXAMPLE



- Suppose the input is $X_1 = 0, X_2 = 0$
  - Output from node A:
    $0 \times (-3.5) + 0 \times (-3.5) + 5.1 = 5.1$
    $logistic(5.1) = 0.99$
  - Output from node B:
    $0 \times (-5.4) + 0 \times (-5.7) + 2.0 = 2.0$
    $logistic(2.0) = 0.88$
  - Output from node C:
    $0.99 \times 6.9 + 0.88 \times (-7.3) + (-3.0) =$
    $6.83 - 6.42 - 3.0 = -2.59$
    $logistic(-2.59) = 0.06$
- Output of network is 0.06

ANN FOR XOR

- Truth table for Exclusive OR (XOR)

X_1	X_2	Output
0	0	0
0	1	1
1	0	1
1	1	0

- If each of these examples/patterns is input to the network

X_1	X_2	Desired Output	Actual Output	Error	Squared Error
0	0	0	0.06	0.06	0.0036
0	1	1	0.92	0.08	0.0064
1	0	1	0.92	0.08	0.0064
1	1	0	0.10	0.10	0.01

- Total sum squared error (TSS) for n patterns

  $\sum_{i=1}^{n}(Desired_i - Actual_i)^2$

  $= 0.0264$

- Mean squared error (MSE) TSS/n

  $= 0.0264/4 = 0.066$

- TSS or MSE is plotted during training