

# AI CHEAT SHEET

## TOPICS:

neural networks ✓  
 anns  
 knowledge representation  
 logic  
 theorem proving  
 rule-based reasoning  
 knowledge structures  
 planning

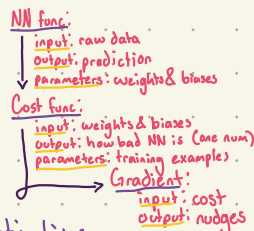
## neural networks

### types:

convolutional - great for image processing  
 long short-term memory - speech recognition  
 multi-layer perceptron - basic!

### neurons:

a neuron is a thing that holds data  
 this data is called "activation"  
 output num: confidence  
 activations from one layer determine activations  
 passed to next layer (hidden/output)



### training:

the "activation sequence" is known through experimentation  
 "weights" are set

### why layered?

components are processed individually  
 components of components...

ie:  $9 = 0 + 1$   
 ie:  $8 = 0 + 0$   
 ie:  $4 = 1 + +$

### training

activations · weights = "weighted sum"

weighted sums can be any number

we want 0 → 1

thus, we use a function to force weighted sums 0 → 1

**Sigmoid function**  $\sigma(x) = \frac{1}{1+e^{-x}}$

### bias

force neuron to only activate if weighted sum >  $\alpha$

thus  $\omega' = \omega - \alpha$  ( $\omega$  = weighted sum)

### learning

just finding the right weights & biases!

**0<sup>th</sup> neuron** **0<sup>th</sup> layer**

$$\sigma \left( \begin{bmatrix} \omega_{0,0} & \omega_{0,1} & \dots & \omega_{0,n} \\ \omega_{1,0} & \omega_{1,1} & \dots & \omega_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{k,0} & \omega_{k,1} & \dots & \omega_{k,n} \end{bmatrix} \begin{bmatrix} a_0^{(0)} \\ a_1^{(0)} \\ \vdots \\ a_n^{(0)} \end{bmatrix} + \begin{bmatrix} b_0 \\ \vdots \\ b_n \end{bmatrix} \right)$$

1. compute VC  
 2. step in - VC dir  
 3. repeat

**GRADIENT DESCENT**

& cost, or how bad the network is

encodes importance of change

Backpropagation  
 this then "nudges"  
 weights in the right  
 direction!

ball rolling  
 down a  
 hill

# Knowledge representation

modelling behavior of intelligent agent

knowledge:

objects

performance

meta-knowledge

events

facts

knowledge-base

five types of knowledge

declarative knowledge

concepts, facts, objects

structural knowledge

problem solving knowledge

relationships between concepts & objects

procedural knowledge

know how to do something

rules, strats, procedures

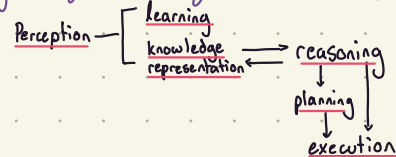
meta-knowledge

knowledge about knowledge

heuristic knowledge

expert knowledge in field

cycle of knowledge representation



knowledge vs intelligence

knowledge fuels intelligence

knowledge is the base of intelligent behavior

techniques

1) Logical representation

2) semantic network representation

3) frame representation

4) Production rules

**LOGICAL REPRESENTATION**

Lang w/ definite rules dealing w/ propositions

No ambiguity

concludes on conditions & cements communication rules

Syntax

\* how we construct

sentences/propositions

in logic

\* defines symbols

Logical reasoning, basis of programming langs

Semantics

\* rules governing

interpretation of

sentences

\* prescribes meaning

## SEMANTIC NETWORK REP

alternative of predicate logic

graphical networks

nodes = objs, edges = relationships

"is a" relation (inheritance)

"kind of" relation

Natural rep, simple

more comp. time

NOT intelligent

## FRAME REPRESENTATION

collection of attributes describing an entity in the world

stereotype situations

slots?

names & values

"facets"

groups related data, easily understandable

generalization

## PRODUCTION RULES

check if cond. exists

if yes, prod rule fires & action is executed

"recognize act cycle"

3 parts:

1) Set of Rules

2) Working Memory

3) Recognize-act cycle

natural language

no learning capabilities

& inefficient

## reasoning:

the ability to make decisions or inferences from facts

Automated

vs Symbolic

such as search

NN's = non-symbolic

encode knowledge in representation

apply inference mechanisms

↳ form new knowledge

## Theorem Proving

knowledge in logical formalisms

knowledge  $\Rightarrow$  new knowledge

## Rule-Based Reasoners

rep knowledge as if-then rules

apply rules  $\Rightarrow$  new knowledge

## Planners

rep knowledge as plan schemas, rules/logic, etc

use specialized planning techniques