GAGP Tutorial 7 (week 10)
Networks and Job Scheduling

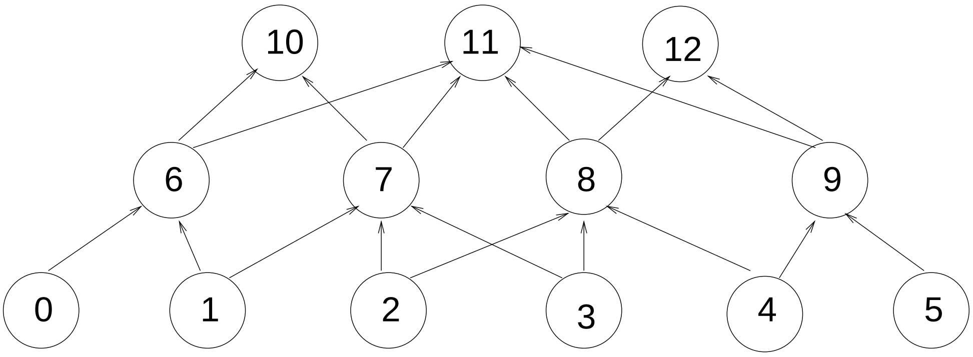
1. Follow the same procedure given in the lecture slides to construct the network given by the grammar below:

$$S \rightarrow \begin{matrix} A & B \\ C & D \end{matrix} \quad A \rightarrow \begin{matrix} c & p \\ a & c \end{matrix} \quad B \rightarrow \begin{matrix} c & a \\ g & e \end{matrix} \quad C \rightarrow \begin{matrix} a & a \\ b & a \end{matrix} \quad D \rightarrow \begin{matrix} f & e \\ a & b \end{matrix}$$

$$a \rightarrow \begin{matrix} 0 & 0 \\ 0 & 0 \end{matrix} \quad b \rightarrow \begin{matrix} 0 & 0 \\ 0 & 1 \end{matrix} \quad c \rightarrow \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \quad e \rightarrow \begin{matrix} 0 & 1 \\ 0 & 1 \end{matrix} \quad f \rightarrow \begin{matrix} 1 & 1 \\ 0 & 1 \end{matrix}$$

$$g \rightarrow \begin{matrix} 0 & 0 \\ 1 & 0 \end{matrix} \quad p \rightarrow \begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}$$

2. Design a grammar that will produce the network architecture given below.



3. Come up with six different ways of combining genetic algorithms and neural networks to produce solutions to problems such as classification problems.

4. Consider the following heterogeneous multiprocessor scheduling problem instance, courtesy of Graham Ritchie. The numbers in the table show the times taken to process job[i] on processor[j]:

	processor1	processor2
job1	2	3
job2	3	4
job3	4	5
job4	5	6

(a) Is this problem consistent or inconsistent? It is consistent if "whenever a processor p_j executes a job j_i faster than another processor p_k, then pj will execute all other jobs faster than p_k." (Ritchie and Levine 2003)

(b) Apply the Min-min algorithm to this problem, as specified below. What is the makespan (the total time taken to process all the jobs) of the resulting schedule?

(c) Apply the following local search: find the worst processor and try all possible swaps of jobs and transfers of single jobs, and then move to the solution which reduces the makespan most. Repeat until no further improvement is possible. What is the makespan of the resulting schedule? Is this result guaranteed to be optimal?

(d) How could you solve the heterogeneous MPS using a genetic algorithm?

(e) Does a GA seem like a potentially sensible solution to the heterogeneous MPS? What alternative methods might be competitive?

Min-min algorithm: compute the minimum completion time for every unscheduled job and assign the job with the minimum minimum completion time to the processor that will do that job in that minimum minimum time.