

Jen Ada (jennifer-ada@outlook.com)

School of Applied Computing, Mount Pleasant, Swansea, Swansea SA1 6ED

Objective

Identify which parts of an artificial neural network (ANN) can perform faster by running aspects of the ANN in parallel.



Leduc Hold 'Em?

- A smaller version of hold 'em (poker)[1], which is more manageable to simulate.
- The deck has two suits with three cards from both suits.
- There are two rounds:
 - Round 1: A private card is dealt to each player.
 - Round 2: A board card is revealed.

Artificial Neural Networks?

- Much like the human brain, an artificial neural network (ANN) is made of neurons which are connected by synapses.
- Input represents imperfect knowledge of the problem (I.e. The AI's hand, cards on the board, NOT the opponent's cards, etc.)
- This input is sent to a layer of neurons and each neuron performs a function which has a weight associated with it.

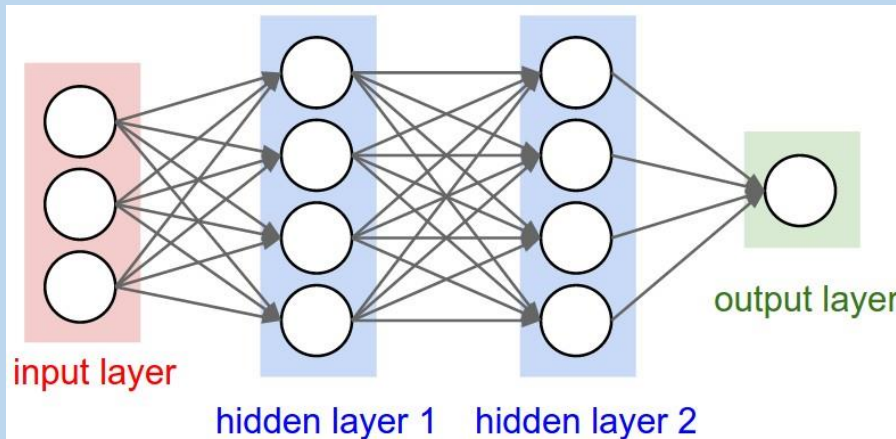


Fig.1: "A 3-layer neural network with three inputs, two hidden layers of 4 neurons each and one output layer." [2]

- These neurons then send their output to the next layer of neurons and so on until they get to the last layer which generates the final output layer.
- If the output is desirable, the connections between the neurons are weighted so that next time the behaviour is more likely to occur again (training).

Parallelisation?

- ANN's can take a large amount of time to train.
- Different parts of the ANN or whole ANN's can be run at the same time on separate pieces of hardware to make the training faster.
- The smallest scope encompasses the individual artificial neuron, the largest being whole ANN's being run at the same time.

Methods

- Initially an ANN will be constructed in the C++ programming language via test driven development.
- Instances of the ANN will be trained in parallel.
- The trained instances will be unified into a single cohesive ANN.
- The performance time of training the parallelised ANN will be compared to a non-parallelised ANN along with accuracy of output.

References

- [1] F. Southey et al., "Bayes' Bluff: Opponent Modelling in Poker," Dept. Comp. Sci., Univ. Alberta, 2012.
- [2] CS231n Convolutional Neural Networks for Visual Recognition, "ANN diagram", 2017. [Online]. Available: <http://cs231n.github.io/neural-networks-1/>