Neural Networks and their Application to Go

Learning Blackjack

Anne-Marie Bausch

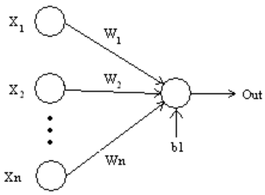
ETH, D-MATH

May 31, 2016

Table of Contents

Perceptron

A *perceptron* is the most basic artificial neuron (developed in the 1950s and 1960s).



The input $X \in \mathbb{R}^n$, $w_1, \dots, w_n \in \mathbb{R}$ are called *weights* and the output $Y \in \{0, 1\}$.
The output depends on some *treshold value* τ:

$$output = \begin{cases} 0, & \text{if} \quad W \cdot X = \sum_j w_j x_j \leq \tau, \\ 1, & \text{if} \quad W \cdot X = \sum_j w_j x_j > \tau. \end{cases}$$

Bias

Next, we introduce what is known as the perceptron's *bias B*,

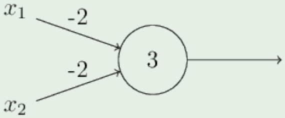
$$B := -\tau.$$

This gives us a new formula for the output,

$$output = \begin{cases} 0, & \text{if} \quad W \cdot X + B \leq 0, \\ 1, & \text{if} \quad W \cdot X + B > 0. \end{cases}$$

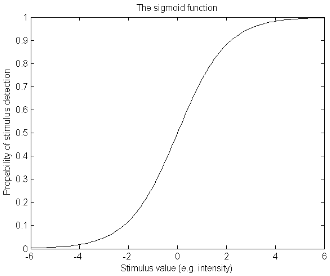
Example

NAND gate:



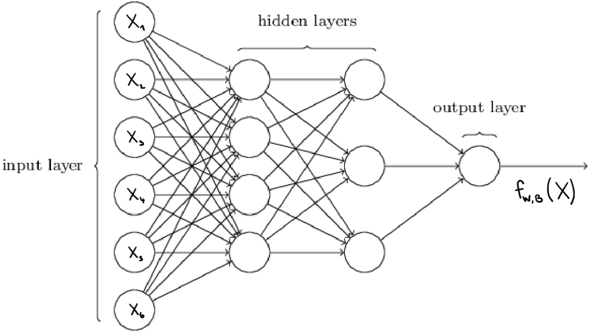
Sigmoid Neuron

- Problem: Small change in input can change output a lot $\rightarrow$ Solution: Sigmoid Neuron
- Input $X \in \mathbb{R}^n$
- Output $= \sigma(X \cdot W + B) = (1 + exp(-X \cdot W - B))^{-1} \in [0, 1]$, where $\sigma(z) := \frac{1}{1+exp(-z)}$ is called the *sigmoid function*.



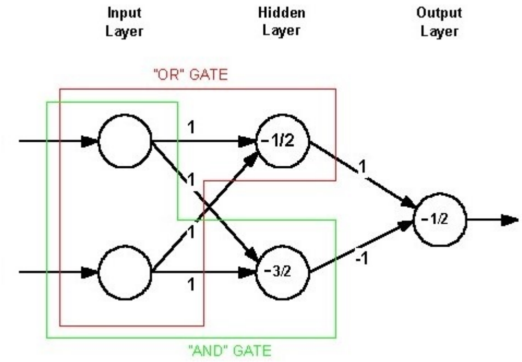
Neural Networks

Given an input X, as well as some training and testing data, we want to find a function $f_{W,B}$ such that $f_{W,B} : X \rightarrow Y$, where Y denotes the output.



How do we choose the weights and the bias?

Example: XOR Gate



Learning Algorithm

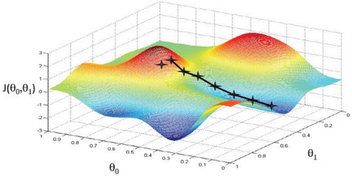
- A learning algorithm chooses weights and biases without interference of programmer.
- Smoothness in σ:

$$\Delta \text{output} \approx \sum_j \frac{\delta \text{output}}{\delta w_j} \Delta w_j + \frac{\delta \text{output}}{\delta B} \Delta B$$

How to update weights and bias

How does the learning algorithm update the weights (and the bias)?

$$\text{argmin}_{W,B} \| f_{W,B}(X) - Y \|_2$$



→ One method to do this is *gradient descent*

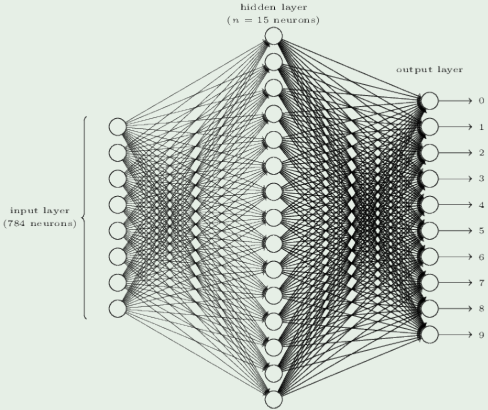
→ Choose appropriate learning rate!

Example

Digit Recognition (1990s) → Youtube Video

Example

One image consists of 28x28 pixels which explains why the input layer has 784 neurons



3 main types of learning

- Supervised Learning (SL)
  $\rightarrow$ Learning some mapping from inputs to outputs.
  Example: Classifying Digits
- Unsupervised Learning (UL)
  $\rightarrow$ Given input and no output, what kinds of patterns can you find? *Example*: Visual input is at first too complex, have to reduce number of dimensions
- Reinforcement Learning (RL)
  $\rightarrow$ Learning method interacts with its environment by producing actions $a_1, a_2, \ldots$ that produce rewards or punishments $r_1, r_2, \ldots$. *Example*: Human learning

Why was there a recent boost in the employment of neural networks?

The evolution of neural networks stagnated because networks with more than 2 hidden layers proved to be too difficult. The main problems and their solutions are:

- Huge amount of Data
  $\rightarrow$ Big Data
- Number of weights (capacity of computers)
  $\rightarrow$ capacity of computers improved (Parallelism, GPUs)
- Theoretical limits
  $\rightarrow$ Difficult ($\Rightarrow$ See next slide)

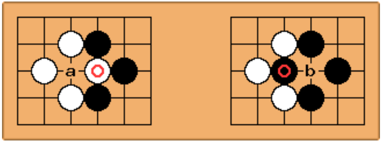
Theoretical Limits

- Back-propagated error signals either shrink rapidly (exponentially in the number of layers) or grow out of bounds
- 3 solutions:
  (a) *unsupervised pre-training* $\Rightarrow$ faciliates subsequent supervised credit assignment through back-propagation (1991).
  (b) LSTM-like networks (since 1997) avoid problem through special architecture.
  (c) Today, fast GPU-based computers allow for propagating errors a few layers further down within reasonable time

The Game of Go

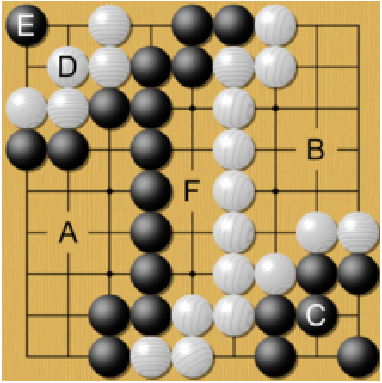
Main rules

- Origin: Ancient China more than 2500 years ago
- Goal: Gain the most points
- White gets 6.5 points for moving second
- Get points for territory at the end of game
- Get points for prisoners
  → Stone is captured if it has no more liberties (liberties are "supply chains")
- Not allowed to commit suicide
- Ko-Rule: Not allowed to play such that game is again as before



End of Game

The game is over when both players have passed consecutively
$\rightarrow$ Prisonners are removed and points are counted!



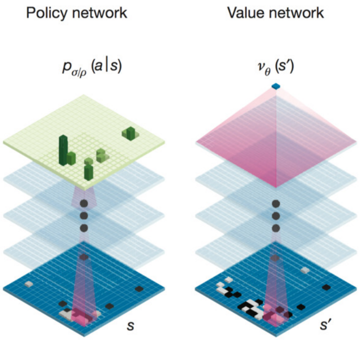
AlphaGo

- DeepMind was founded in 2010 as a startup in Cambridge
- Google bought DeepMind for $500M in 2014
- AlphaGo beat European Go champion Fan Hui (2-dan) in October 2015
- AlphaGo beat Lee Sedol (9-dan), one of the best players in the world in March 2016 (4 out of 5 games)
- Victory of AI in Go was thought to be 10 years into the future
- 1920 CPUs and 280 GPUs used during match against Lee Sedol
  $\rightarrow$ This equals around $1M without counting the electricity used for training and playing
- Next Game attacked by Google DeepMind: Starcraft

AlphaGo

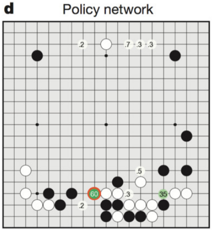
Difficulty: Search space of future Go moves is larger than the number of particles in the known universe

- Policy Network
- Value Network
- Monte Carlo Tree Search (MCTS)



Policy Network Part 1

- Multi-Layered Neural Network
- Supervised-learning (SL)
- Goal: Look at board position and choose next best move (does not care about winning, just about next move)
- is trained on millions of example moves made by strong human players on KGS (Kiseido Go Server)
- it matches strong human players about 57% of time (mismatches arenot necessarily mistakes)

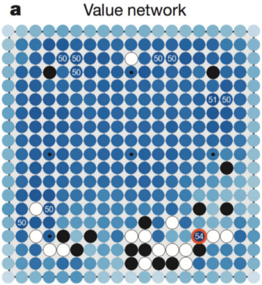


Policy Network Part 2

- 2 additional versions of policy networks: A stronger move picker and a faster move picker
- Stronger version uses RL
  $\rightarrow$ trained more intensively by playing game to the end (is trained by millions of training games against previous editions of itself, it does no reading, i.e., it does not try to simulate any future moves)
  $\rightarrow$ needed for creating enough training data for value network
- Faster version is called "rollout network" $\rightarrow$ does not look at entire board but at smaller window around previous move $\rightarrow$ about 1000 times faster!

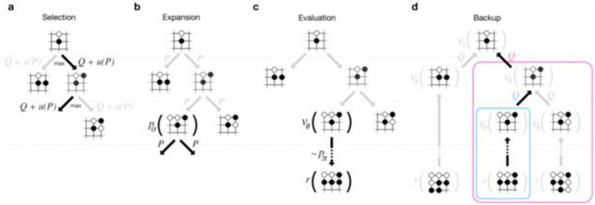
Value Network

- Multi-Layered Neural Network
- Estimates probability of each player winning the game
- Is useful for speeding up reading: If particular position is bad, can skip any more moves along that line of play
- Trained on millions of example board positions which were randomly picked between two copies of AlphaGo's strong move-picker

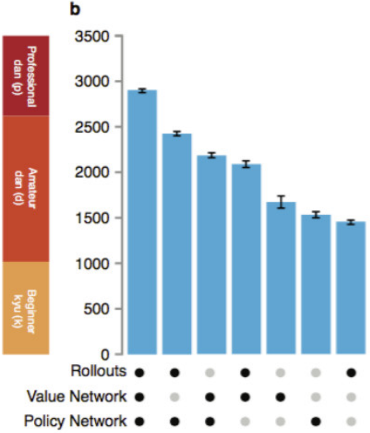


MCTS

- accomplishes reading and exploring
- Full-Power AlphaGo system then uses all of its "brains" in the following way:
  → Choose a few possible next moves using the basic move picker (stronger version made AlphaGo weaker!)
  → Evaluate each next move using value network and a deeper MC simulation (called "rollout", uses fast move picker) → Get 2 independent guesses → use parameter to combine 2 guesses (optimal parameter is 0.5)



How the strength of AlphaGo varies



References

- Mastering the game of Go with deep neural networks and tree search, Nature Volume 259, 2016
- http://neuralnetworksanddeeplearning.com/chap1.html
- https://www.dcine.com/2016/01/28/alphago/
- Wikipedia Go(game)