

MIT  
Technology  
Review

Login / Register
Search Q

Topics+
The Daily
Magazine
Business Reports
More+

Subscribe

Computing

# Google Unveils Neural Network with “Superhuman” Ability to Determine the Location of Almost Any Image

Guessing the location of a randomly chosen Street View image is hard, even for well-traveled humans. But Google’s latest artificial-intelligence machine manages it with relative ease.

by Emerging Technology from the arXiv February 24, 2016

1

PiaNet - Photo Geolocation with Convolutional Neural Networks

Tobias Weyand  
Google  
weyand@google.com

Ilya Kostrikov  
RWTH Aachen University  
ilya.kostrikov@rwth-aachen.de

James Philbin  
Google  
philbin@google.com

## Abstract

It is possible to build a system to determine the location where a photo was taken using just the pixels? In general, the problem seems exceptionally difficult: it is trivial to construct situations where no location can be inferred. Yet images often contain informative cues such as landmarks, weather patterns, vegetation, road markings, and architectural details, which in combination may allow one to determine an approximate location and occasionally an exact location. Humans such as Google Maps and View from your Window suggest that humans are relatively good at inferring these cues in geotagged images, especially on maps. In computer vision, the photo geolocation problem is usually approached using image retrieval methods. In contrast, we pose the problem as one of classification by subdividing the surface of the earth into thousands of multi-scale geographic cells, and train a deep network using millions of geotagged images. While previous approaches only recognize landmarks or perform approximate matching using global image descriptors, our model is able to use and integrate multiple visible cues. We show that the resulting model, called PiaNet, outperforms previous approaches and even attains superhuman levels of accuracy in some cases. Moreover we extend our model to photo albums by combining it with a long short-term memory (LSTM) architecture. By learning to exploit temporal coherence in geotagged sequential photos, we demonstrate that this model achieves a 50% performance improvement over the single-image model.

Figure 1. Given a query photo (left), PiaNet outputs a probability distribution over the surface of the earth (right). Varying the task in a classification problem allows PiaNet to express its uncertainty about a photo. While the Eiffel Tower (a) is confidently assigned to Paris, the model believes that the first photo (b) could have been taken in either New Zealand or Hawaii. For the beach photo (c), PiaNet assigns the highest probability to southern California beaches, but some probability mass is also assigned to places with similar beaches, like Mexico and the Mediterranean. (For visualization purposes we use a model with much more rapid resolution than our full model.)

2

# CS 188: Artificial Intelligence

## Utilities

Instructor: Marco Alvarez  
University of Rhode Island

[These slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at <http://ai.berkeley.edu>.]

3

## Other Game Types

4

## Mixed Layer Types

- E.g. Backgammon
- Expectiminimax
- Environment is an extra “random agent” player that moves after each min/max agent
- Each node computes the appropriate combination of its children

5

## Expectiminimax Pseudocode

```

def value(state):
    if the state is a terminal state: return the state's utility
    if the next agent is MAX: return max-value(state)
    if the next agent is MIN: return min-value(state)
    if the next agent is EXP: return exp-value(state)

def max-value(state):
    initialize v = -∞
    for each successor of state:
        v = max(v, value(successor))
    return v

def min-value(state):
    initialize v = +∞
    for each successor of state:
        v = min(v, value(successor))
    return v

def exp-value(state):
    initialize v = 0
    for each successor of state:
        p = probability(successor)
        v += p * value(successor)
    return v

```

6

## Example: Backgammon

- Dice rolls increase b: 21 possible rolls with 2 dice
  - Backgammon ~ 20 legal moves
  - Depth  $2 = 20 \times (21 \times 20)^3 = 1.2 \times 10^9$
- As depth increases, probability of reaching a given search node shrinks
  - So usefulness of search is diminished
  - So limiting depth is less damaging
  - But pruning is trickier...
- Historic AI: TDGammon uses depth-2 search + very good evaluation function + reinforcement learning: world-champion level play
- 1<sup>st</sup> AI world champion in any game!

Image: Wikipedia

7

## Multi-Agent Utilities

- What if the game is not zero-sum, or has multiple players?
- Generalization of minimax:
  - Terminals have utility tuples
  - Node values are also utility tuples
  - Each player maximizes its own component
  - Can give rise to cooperation and competition dynamically...

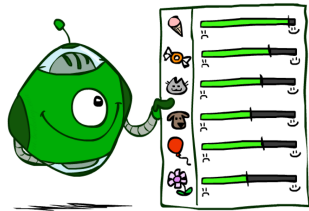
8

## Games Summary

- Games require decisions
  - optimality is impossible (for most games/problems)
  - bounded-depth search and evaluation functions
  - alpha-beta pruning
- Important advances (from game playing)
  - reinforcement learning
  - iterative deepening
  - monte carlo tree search
- Video games?
  - greater challenges

9

## Utilities



10

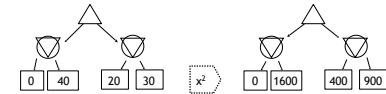
## Maximum Expected Utility

- Why should we average utilities? Why not minimax?
- Principle of maximum expected utility:
  - A rational agent should choose the action that maximizes its expected utility, given its knowledge
- Questions:
  - Where do utilities come from?
  - How do we know such utilities even exist?
  - How do we know that averaging even makes sense?
  - What if our behavior (preferences) can't be described by utilities?



11

## What Utilities to Use?



- For worst-case minimax reasoning, terminal function scale doesn't matter
  - We just want better states to have higher evaluations (get the ordering right)
  - We call this insensitivity to monotonic transformations
- For average-case expectimax reasoning, we need *magnitudes* to be meaningful

12

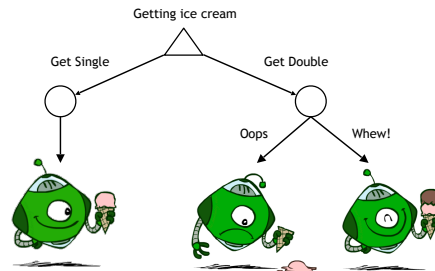
## Utilities

- Utilities are functions from outcomes (states of the world) to real numbers that describe an agent's preferences
- Where do utilities come from?
  - In a game, may be simple (+1/-1)
  - Utilities summarize the agent's goals
  - Theorem: any "rational" preferences can be summarized as a utility function
- We hard-wire utilities and let behaviors emerge
  - Why don't we let agents pick utilities?
  - Why don't we prescribe behaviors?



13

## Utilities: Uncertain Outcomes

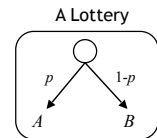


14

## Preferences

- An agent must have preferences among:
  - Prizes:  $A, B$ , etc.
  - Lotteries: situations with uncertain prizes

$$L = [p, A; (1-p), B]$$

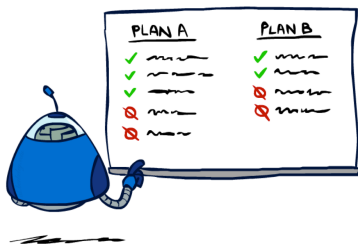


- Notation:
  - Preference:  $A \succ B$
  - Indifference:  $A \sim B$



15

## Rationality



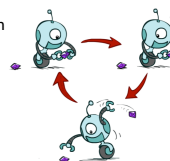
16

## Rational Preferences

- We want some constraints on preferences before we call them rational, such as:

$$\text{Axiom of Transitivity: } (A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$$

- For example: an agent with intransitive preferences can be induced to give away all of its money
  - If  $B \succ C$ , then an agent with  $C$  would pay (say) 1 cent to get  $B$
  - If  $A \succ B$ , then an agent with  $B$  would pay (say) 1 cent to get  $A$
  - If  $C \succ A$ , then an agent with  $A$  would pay (say) 1 cent to get  $C$



17

## Rational Preferences

### The Axioms of Rationality

- Orderability  
 $(A \succ B) \vee (B \succ A) \vee (A \sim B)$
- Transitivity  
 $(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$
- Continuity  
 $A \succ B \succ C \Rightarrow \exists p [p, A; 1-p, C] \sim B$
- Substitutability  
 $A \sim B \Rightarrow [p, A; 1-p, C] \sim [p, B; 1-p, C]$
- Monotonicity  
 $A \succ B \Rightarrow (p > q \Rightarrow [p, A; 1-p, B] \succ [q, A; 1-p, B])$



Theorem: Rational preferences imply behavior describable as maximization of expected utility

18

## MEU Principle

- Theorem [Ramsey, 1931; von Neumann & Morgenstern, 1944]
  - Given any preferences satisfying these constraints, there exists a real-valued function  $U$  such that:

$$U(A) \geq U(B) \Leftrightarrow A \succeq B$$

$$U([p_1, S_1; \dots; p_n, S_n]) = \sum_i p_i U(S_i)$$

- i.e. values assigned by  $U$  preserve preferences of both prizes and lotteries!

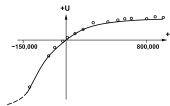
- Maximum expected utility (MEU) principle:
  - Choose the action that maximizes expected utility
  - Note: an agent can be entirely rational (consistent with MEU) without ever representing or manipulating utilities and probabilities
  - E.g., a lookup table for perfect tic-tac-toe, a reflex vacuum cleaner



19

## Money

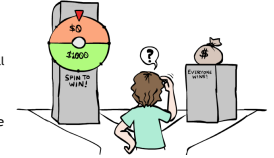
- Money does not behave as a utility function, but we can talk about the utility of having money (or being in debt)
- Given a lottery  $L = [p, \$X; (1-p), \$Y]$ 
  - The expected monetary value  $EMV(L)$  is  $p \cdot X + (1-p) \cdot Y$
  - $U(L) = p \cdot U(\$X) + (1-p) \cdot U(\$Y)$
  - Typically,  $U(L) < U(EMV(L))$
  - In this sense, people are risk-averse
  - When deep in debt, people are risk-prone



20

## Example: Insurance

- Consider the lottery  $[0.5, \$1000; 0.5, \$0]$ 
  - What is its expected monetary value? (\$500)
  - What is its certainty equivalent?
    - Monetary value acceptable in lieu of lottery
    - \$400 for most people
- Difference of \$100 is the insurance premium
  - There's an insurance industry because people will pay to reduce their risk
  - If everyone were risk-neutral, no insurance needed!
- It's win-win: you'd rather have the \$400 and the insurance company would rather have the lottery (their utility curve is flat and they have many lotteries)



21

## Example: Human Rationality?

- Famous example of Allais (1953)
  - A:  $[0.8, \$4k; 0.2, \$0]$   $\Leftarrow$
  - B:  $[1.0, \$3k; 0.0, \$0]$
  - C:  $[0.2, \$4k; 0.8, \$0]$
  - D:  $[0.25, \$3k; 0.75, \$0]$
- Most people prefer  $B > A, C > D$
- But if  $U(\$0) = 0$ , then
  - $B > A \Rightarrow U(\$3k) > 0.8 U(\$4k)$
  - $C > D \Rightarrow 0.8 U(\$4k) > U(\$3k)$



22