Deep Learning Models for Games

Bachelor Thesis Session – September 2015

Florentina-Ștefania Bratiloveanu
Supervisor: As. Drd. Ing Tudor Berariu

Faculty of Automatic Control and Computers,
University POLITEHNICA of Bucharest

September 14, 2015

1 Motivation

2 State of the art

3 Architecture, Design, Results

4 Future work

5 Conclusions

6 QA

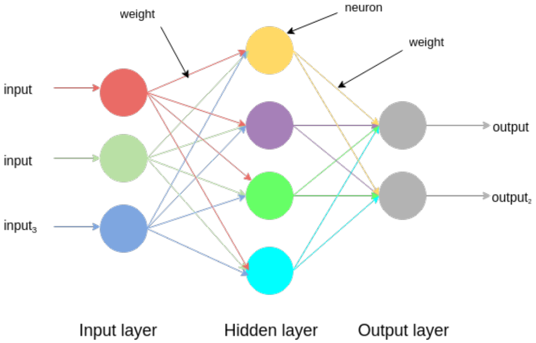
Motivation

Deep Learning

- find models capable of generalization
- extract from low-level(edges,colors) to high-level(combination of rudimentary features) features
- reduce programming burden
- applicability: cancer classification, autonomous cars, object recognition from images

Once upon a time...

- reinforcement learning: Q-Learning
- neural networks **vs** deep neural networks

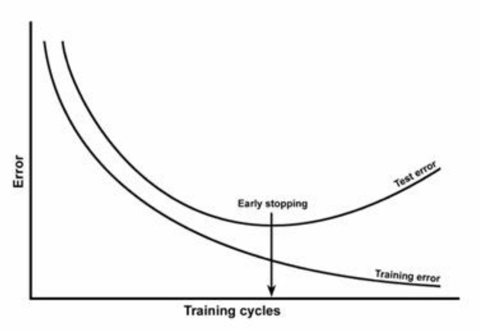


Preprocessing, Model, Loss function

- color space: RGB, YUV, grayscale
- data normalization 0..1, contrast normalization
- activation functions
    - hidden layer vs output layer
    - tanh, sigmoid, ReLU
- how many layers/features, what type of layers
- loss function: classification(binary/multi-class) or regression?
- gradient descent **vs** stochastic gradient descent

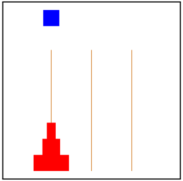
Train and test

- split dataset for training and testing
- when to stop training?

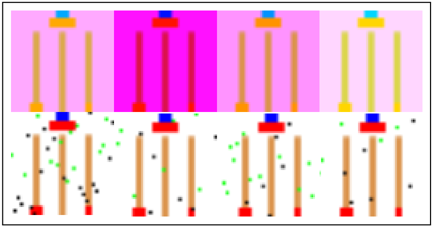


Source:
http://documentation.statsoft.com/statisticahelp.aspx?path=sann/overview/sannoverviewsnetworkgeneralization

Once upon a time...



Tower of Hanoi (first state)

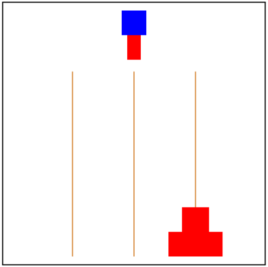


Dataset with noise added and color changed

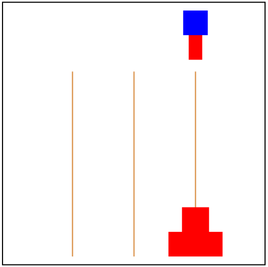
- game: Tower of Hanoi
- reinforcement learning: Q-Learning
- deep neural networks: predict values from Q-Learning
- machine learning framework: Torch based on Lua

Q-Learning

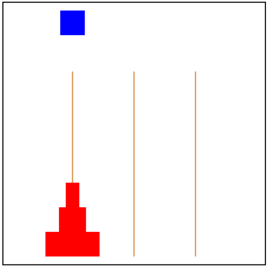
- finds optimal policy for action-value function
- $Q(s,a) = Q(s,a) + \alpha \cdot (r + \gamma \cdot \max_a Q(s', a') - Q(s, a))$



UP = 90,6534
DOWN = 86,8787
LEFT = 89,1867
RIGHT = 94,2824



UP = 97,6530
DOWN = 100,0000
LEFT = 93,8538
RIGHT = 92,5261



UP = 26,3520
DOWN = 23,8452
LEFT = 23,8897
RIGHT = 22,8827

Regression with complex model

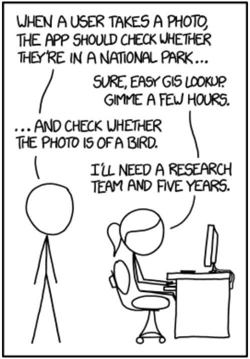
nn

- predict values obtained from Q-Learning with convnet
- failure (the dnn was too complex)

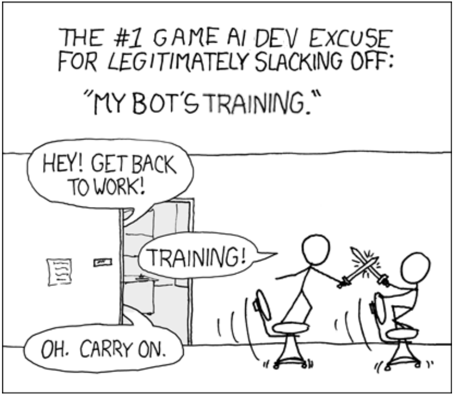
Future work

- implement Q-Network
- test algorithm on dynamic environments or games where the state of the universe is not fully observed
- make Nao capable of playing Tic-Tac-Toe
- after all tasks mentioned above are done, use all the information gathered for cancer classification, etc.

Conclusions


WHEN A USER TAKES A PHOTO, THE APP SHOULD CHECK WHETHER THEY'RE IN A NATIONAL PARK...
SURE, EASY GIS LOOKUP. GIMME A FEW HOURS.
...AND CHECK WHETHER THE PHOTO IS OF A BIRD.
I'LL NEED A RESEARCH TEAM AND FIVE YEARS.

IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.



Source: http://xkcd.com/

QA

Questions and Answers

Thank you for your attention!