Deep Learning Models for Games

Bachelor Thesis Session – September 2015

Florentina-Ştefania 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

- Regression with complex model
- Regression with simple model

4 Future work

5 Conclusions

6 QA

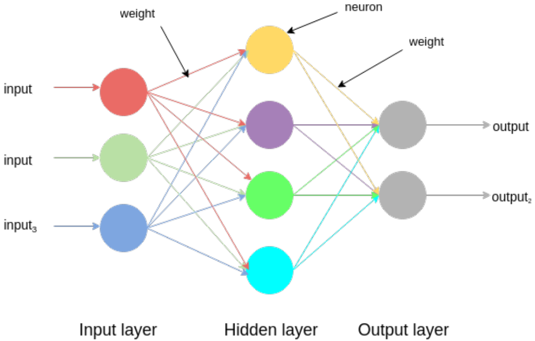
Motivation

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

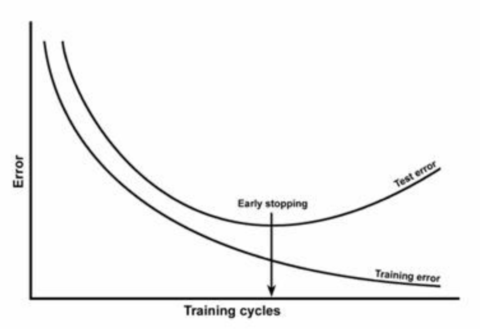


Preprocessing, Model, Loss function

- 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

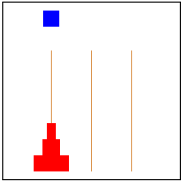
Train and test

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

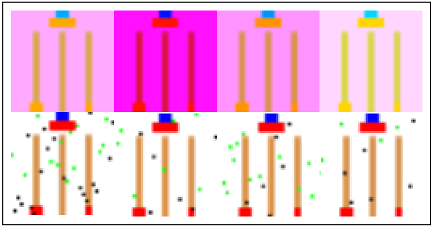


Source:
http://documentation.statsoft.com/statisticahelp.aspx?path=sann/overview/sannoverviewsnetworkgeneralization

Once upon a time...



Tower of Hanoi (first state)

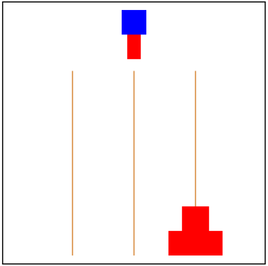


Dataset with noise added and color changed

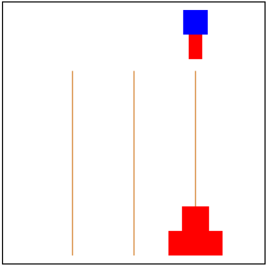
- game: Tower of Hanoi
- reinforcement learning: Q-Learning
- deep neural networks: predict values from Q-Learning
- machine learning framework: Torch based on Lua

Q-Learning

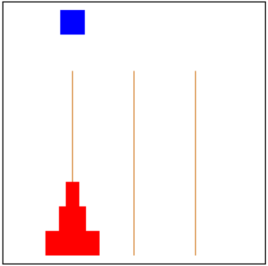
- finds optimal policy for action-value function
- $\mathrm{Q(s,a)} = \mathrm{Q(s,a)} + \alpha\cdot(\mathrm{r} + \gamma \cdot \mathrm{max}_a\, Q(s', a') - Q(s, a))$



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



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

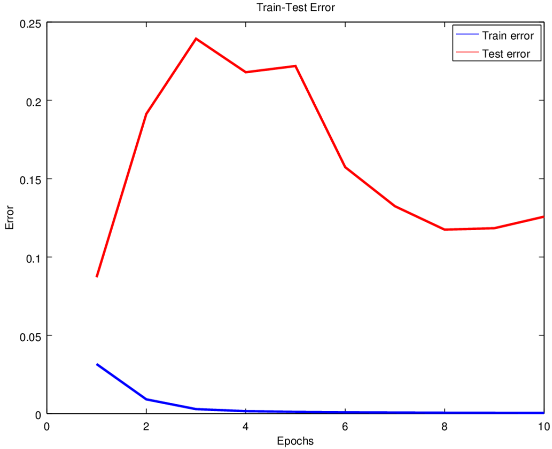


UP = 26,3520
DOWN = 23,8452
LEFT = 23,8897
RIGHT = 22,8827

Model

```
(1):  nn.SpatialConvolutionMM(3 -→ 8, 5x5)
(2):  nn.Tanh
(3):  nn.SpatialSubSampling
(4):  nn.SpatialConvolutionMM(8 → 20, 5x5)
(5):  nn.Tanh
(6):  nn.SpatialSubSampling
(7):  nn.SpatialConvolutionMM(20 → 120, 5x5)
(8):  nn.Reshape(120)
(9):  nn.Linear(120 → 100)
(10): nn.Tanh
(11): nn.Linear(100 → 4)
(12): nn.Sigmoid
```

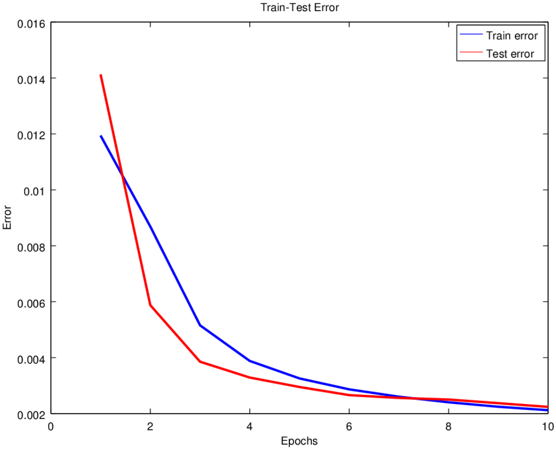
Results


Train-Test Error
Error
Epochs
0
0.05
0.1
0.15
0.2
0.25
0
2
4
6
8
10
Train error
Test error

Model

(1): nn.SpatialConvolutionMM(3 → 4, 5x5)
(2): nn.Tanh
(3): nn.SpatialSubSampling
(4): nn.SpatialConvolutionMM(4 → 6, 5x5)
(5): nn.Tanh
(6): nn.SpatialSubSampling
(7): nn.Reshape(150)
(8): nn.Linear(150 → 4)

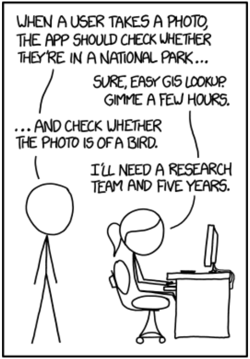
Results


Train-Test Error
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
Error
0
2
4
6
8
10
Epochs
Train error
Test error

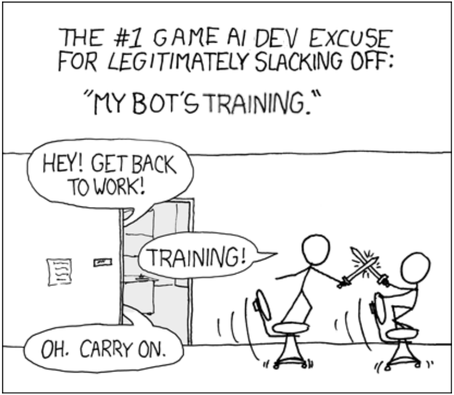
Future work

- 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!