NAOlution: Using Convolutional neural network to navigate robot NAO towards an object

Martin Kostelansky

Faculty of Informaion Technology
Czech Technical University
Prague, Czechia
Email: kostema3@fit.cvut.cz

NAOlution rearches training Convolutional neural network to navigate robot NAO towards an object using image input. Convolutional Neural networks were trianed by genetic algorithm in real-time simulation. By this method were nets trianed to navigatge robot NAO towards "plant like object" in simulation.

1 Introduction

Convloutional neural networks have been widly used for many purposes such as classification or localization. In our research we tried to use ligtened CNN to navigate robot NAO in real-time simulation towards an object. The network's architecture was inspired by VGG [link]. To train such networks we used genetic algorithm. There were ranked by by the distance of the robot from the target. By this method we were able to train a model to come to close distance of the selected object.

2 Network architecture

The network's arcitechture was mainly inspired by deep convloutional neural nets used for image recognition such as VGG [link]. The architecture was lightened down to use this network for multiple frames per second.

2.1 Input

The image from NAO's camere were resized to $128x128$ pixels grayscale, so the input dimension was $128x128x1$, where were the values from interval $(0, 255)$.

2.2 Convolutional and pooling layers

The first layer was a maxpooling layer with receptor field $2x2$ pixels and the spatial resoultion of the output from this layer is thus $64x64x1$. This layer follows two convilutional layers both with four fearure maps with receptor field $4x4$ pixels and bias equals to one. The next sequence of convolutional layers is separated with maxpooling pooling layer with receptor field $4x4$ pixels. Second block of convolutional block is composed from two layers, both with eight fearure maps and receptor fields $4x4$ pixels. Last layer is again maxpooling layer with receptor field $2x2$ pixels.

Every convolutional layer was inicialized to random value from normal distribution with mean 0.0 and standard deviation 0.05.

2.3 Fully connected layers

Convolutional and pooling layers are followed by two fully connected layers, both of them contains eight neurons activated by RELU function and their weighst were randomly selected from normal distribution with mean 0.0 and standard deviation 0.05.

2.4 Output layer

Output layer has only three neurons, all of them use as activation function TANH function and were randomly selected from normal distribution with mean 0.0 and standard deviation 0.05.

Every output from these layer features robot's velocity towards/backward, left/right and left/right spin.

Layer	Dimension
Input	128x128x1
MaxPool1	2x2, stride 1
Conv1	4x4, 4 x 2
MaxPool2	2x2, stride 1
Conv2	4x4, 4 x 2
MaxPool3	2x2, stride 1
FC1	16 relu
FC2	16 relu
FC3	3 tanh

3 Evolution algorithm

During the evolution was created a polulation containing 50 individuals created by process described in [Link].

Each individual was evaluated by the distance from the target. Indivudual simulation run lasted until the robot commes to a target or until the timeout. After each epoch wes created a new generation by crossing layers of two individuals. Indivuduals for crossing were randomly selected from normal distribution with mean 0.0 and standard deviation equals to 30% of the populaiton size. Evolution ran for 30 generations. Graph [link] describes average, best and worst performance.
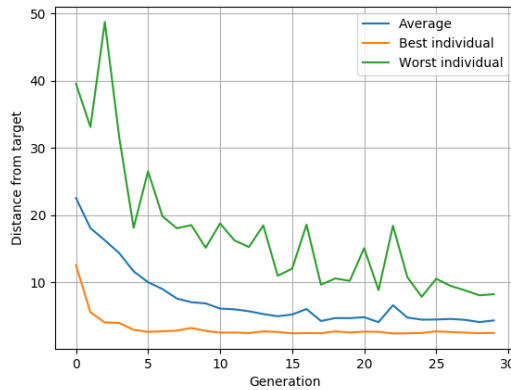

Fig. 1. Distance from the target during the evolution.

4 Conclusion

In this work we have shown that it's possible to use Convolutional neural networks to navigate robot towards an object in real time simulation. To use train CNN to navigate the robot NAO towards an object in multiple positions, or to generalize the target object would be needed more similations with diferent positions and diferent "similar objects" (e.g. in our case diferent "plant like objects"). Based on the research when was trained robotic arm to manipulate cubes [link] to highlite the target object would be also needed to change backround and floor to multiple patterns during the training.

References