

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CAMPUS CURITIBA

ALLAN PATRICK  
BRUNO MABA  
TOMÁS ABRIL

## RELATÓRIO

Desenvolvimento de um Sistema de Controle por rede neural  
multinível para o desvio de obstáculos por um robô móvel  
autônomo simulado no ambiente V-REP

CURITIBA  
2017

ALLAN PATRICK  
BRUNO WANDREY  
TOMÁS ABRIL

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

**Desenvolvimento de um Sistema de  
Controle por rede neural multinível  
para o desvio de obstáculos por um  
robô móvel autônomo simulado no am-  
biente V-REP**

Curitiba, junho de 2017

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>5</b>
<b>1.1</b>	<b>Fundamentação teórica</b>	<b>5</b>
<b>1.2</b>	<b>Ambiente de simulação V-REP</b>	<b>6</b>
<b>2</b>	<b>DESENVOLVIMENTO</b>	<b>7</b>
<b>2.1</b>	<b>Sensores</b>	<b>7</b>
<b>2.2</b>	<b>Treinamento da Rede Neural</b>	<b>7</b>
2.2.1	Métodos de geração de dados para treinamento	7
2.2.2	Dados de Treinamento	7
2.2.3	Configuração da Rede Neural	8
<b>2.3</b>	<b>Integrando Rede Neural ao Kjunior</b>	<b>8</b>
<b>3</b>	<b>CÓDIGO FONTE</b>	<b>9</b>
<b>4</b>	<b>CONCLUSÃO</b>	<b>9</b>



# 1 Introdução

O presente relatório tem com objetivo detalhar funcionamento e a implementação de um sistema de rede neural multinível aplicado ao ambiente de simulação V-REP. Além proporcionar uma discussão sobre os métodos utilizados para obtenção dos melhores resultados. Conforme será apresentado a seguir.

A seguir apresenta-se uma breve fundamentação teórica sobre redes neurais e o desenvolvimento da implementação para controle no ambiente simulado V-REP.

## 1.1 Fundamentação teórica

Rede neural artificial é um sistema de computação construído por um numero de elementos de processamento simples e muito interconectados. O sistema processa informações com a resposta desses elementos a entradas externas. Redes neurais artificiais são dispositivos modelados vagamente na estrutura do córtex cerebral dos mamíferos, mas em escala muito menor. A Figura 1 mostra uma representação de uma rede neural multinível.

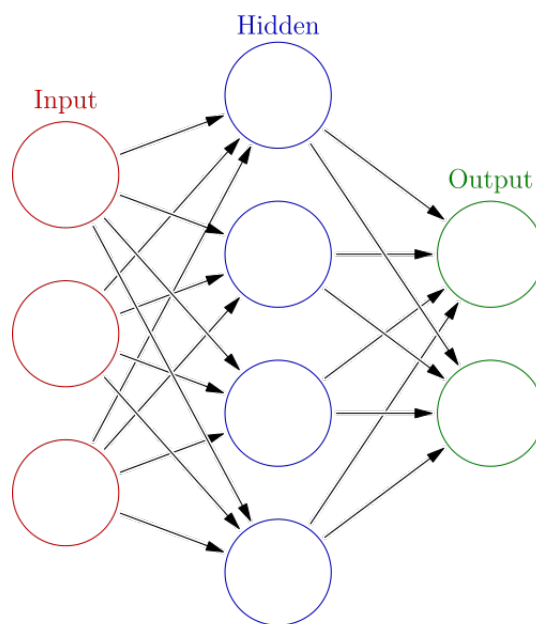


Figura 1 – Modelo de RN utilizado na implementação

## 1.2 Ambiente de simulação V-REP

V-REP(Virtual Robot Experimentation Plataform) é um ambiente de simulação de robôs, onde os atuadores e sensores podem ser acessados por api's em diversa linguagens, dentre elas C++, a utilizada para o desenvolvimento. O robô que programado foi o KJunior o qual nos foi fornecido o código fonte pelo professor.

## 2 Desenvolvimento

As seções a seguir irão detalhar os passos aplicados no sistema de controle por redes neurais no robô Kjunior.

### 2.1 Sensores

O robô K-junior possui um total de 10 sensores. Foram utilizados apenas 5 cujas direções são esquerda, esquerda/frente, frente, direita/frente, direita.

### 2.2 Treinamento da Rede Neural

A biblioteca de rede neural utilizada foi a disponibilizada pelo professor João Fabro.

#### 2.2.1 Métodos de geração de dados para treinamento

Foram utilizadas diversas técnicas para a geração dos dados de treinamento.

Entre elas a geração de dados a partir da resposta de um sistema de controle com logica fuzzy. Neste método foram testados uma quantidade de dados entre 500 e 10000.

Outro método foi a criação manual de 34 conjuntos de entrada/saída visando cobrir a maioria dos casos em que o robô encontra durante o seu percurso no labirinto simulado.

O último método apresentou o melhor resultado onde foram criados a partir de quatro conjuntos de dados simples apenas.

Todos os dados de treinamento encontram-se no repositório do Github.

#### 2.2.2 Dados de Treinamento

A Tabela 1 contém os dados de treinamento que apresentaram o melhor resultado. Eles estão inseridos no arquivo training.junin

Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Motor 1	Motor 2
1	1	1	1	1	1	1
0	1	1	1	1	-1	1
1	1	1	1	0	1	-1
0	1	1	1	0	0.9	1

Tabela 1 – Dados do treinamento

Valores 1 nos sensores representam distancia muito longe, valores 0 são a leitura mais próxima possível. Nos motores valor de 1 representa velocidade máxima para frente, valor 0 parado e valor -1 velocidade máxima para trás.

### 2.2.3 Configuração da Rede Neural

No arquivo `config.junin` foi configurada a rede da seguinte forma:

- 6 - Number of inputs
- 2 - Number of outputs
- 2 - Number of layers in the neural net
- 6 - Number of neurons in the first hidden layer
- 2 - Number of neurons in the second hidden layer
- 0.099 - Parameter momentum
- 0.2 - Learning rate
- 0.0025 - Maximum error

A quantidade de neurônios intermediários, o momento e a taxa de aprendizagem foram definidos empiricamente. Com essa configuração foi possível obter uma taxa de erro bem pequena com pouco tempo de treinamento. A quantidade de entradas são 6 porque o robô está enviando o valor de 6 de seus sensores para a rede, entretanto o sensor traseiro está desativado e envia apenas o valor 0. As saídas são 2 porque seus valores são passados para cada um dos dois motores das rodas.

## 2.3 Integrando Rede Neural ao Kjunior

A compilação e o treinamento da rede pode ser feito executando o arquivo `compilar.sh`.

Após obter sucesso no passo anterior é necessário copiar os arquivos `juninClient` e `neuralnet.junin` para a pasta principal do V-REP. Dentro do V-REP a cena `puquejunin.ttt` que encontra-se no repositório do Github deve ser carregada.



## 3 Código Fonte

O código pode ser acessado em <<https://github.com/tomasabril/vrep-fuzzy>>.

## 4 Conclusão

O projeto foi concluído satisfatoriamente, conseguimos implementar um sistema de rede neural na prática, o robô consegue desviar de obstáculos com boa margem de segurança ao risco de bater na parede, apresentando ainda movimentos suaves na realização de curvas demonstrando-se um controle muito eficaz nas tomadas de decisões para desvios de obstáculos.

Um video do robô Kjunior em execução com o nosso treinamento pode ser visto no youtube pelo link a seguir.

<<https://youtu.be/U46yRWEapMk>>

Os conhecimentos adquiridos em sala de aula foram revistos e aprofundados. O desenvolvimento na plataforma VREP foi relativamente trabalhoso, visto que houveram diversas dificuldades em apropriar o sistema de base que contém dezenas de arquivos mas não muita documentação. Existiu a necessidade de fazer pequenas mudanças em vários arquivos para conseguir compilar, treinar e executar a rede neural e o robô com sucesso. Também foram encontradas dificuldades em utilizar a biblioteca de redes neurais fornecida. Em alguns casos ela apresenta erros e Segmentation Faults sem indicar a causa.

Para trabalhos futuros seria interessante explorar a possibilidade de usar uma biblioteca mais moderna e simples como scikit-learn - Machine Learning in Python.