

Aplicação de redes neurais artificiais no balanceamento de carga em serviços de nuvem

KAROLINE KIMIKO FIGUEIREDO SETOUE

Orientador: Prof. Dr. Kelton Augusto Pontara da Costa

COMPUTAÇÃO EM NUVEM

“

Modelo que permite acesso a rede de forma ubíqua, conveniente, on-demand para compartilhar um grupo de recursos computacionais configuráveis (como por exemplo rede, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente fornecidos e gerados com mínimo esforço de gerenciamento ou interação com o provedor.

”

MELL; GRANCE, 2011

CONCEITOS

Sistemas distribuídos

Virtualização

Serviços

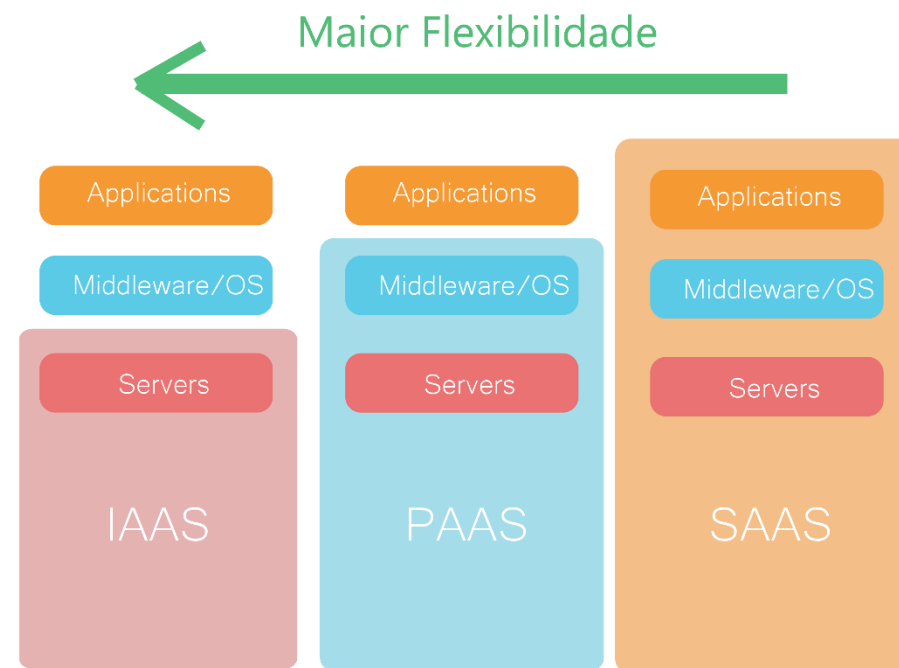
COMPUTAÇÃO EM NUVEM

Modelos

- ✓ Infrastructure-as-a-service
- ✓ Platform-as-a-service
- ✓ Software-as-a-service

Tipos

- ✓ Pública
- ✓ Privada
- ✓ Híbrida

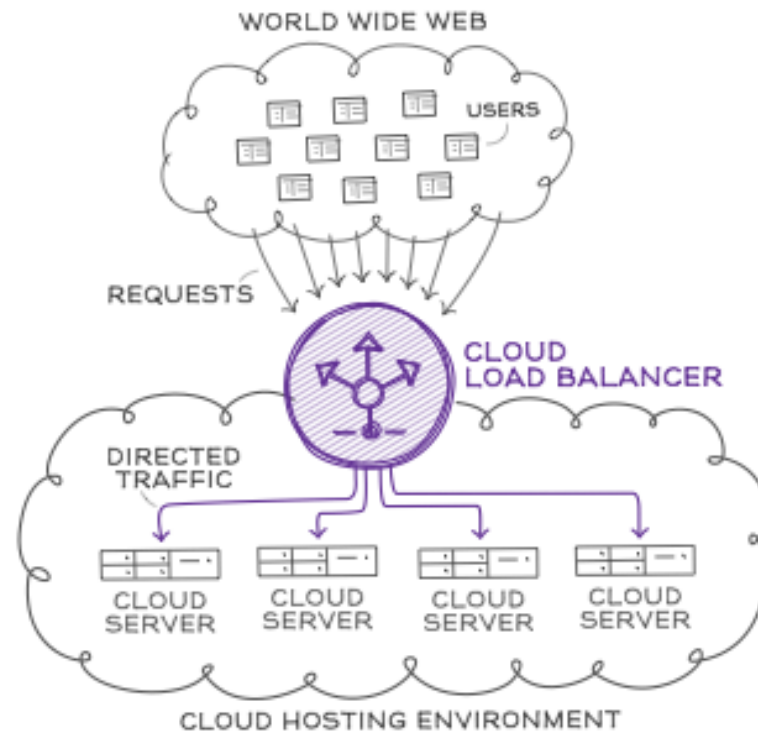


COMPUTAÇÃO EM NUVEM - FORNECEDORES



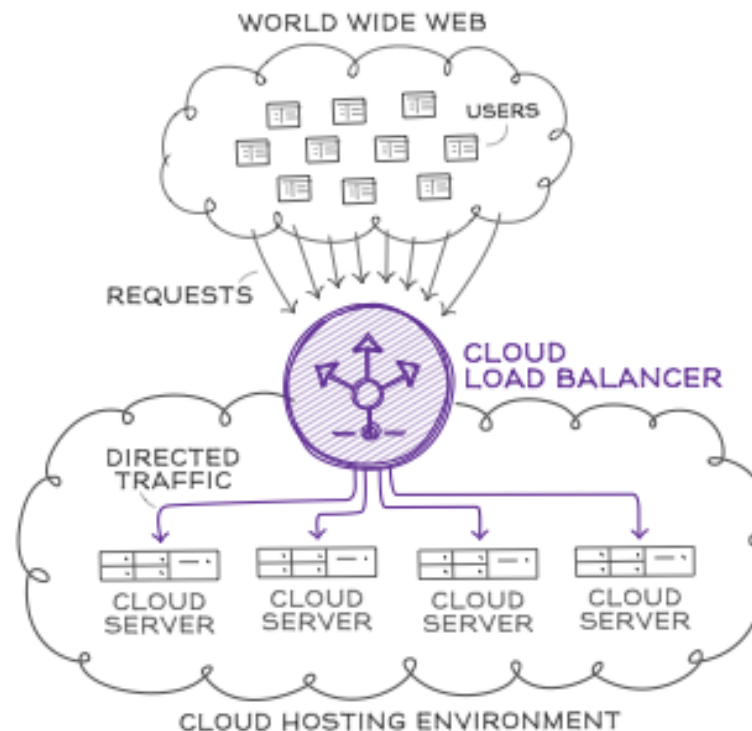
BALANCEAMENTO DE CARGA

- ✓ Divisão de cargas entre os nós da rede
- ✓ Distribuição equilibrada
- ✓ Transparência



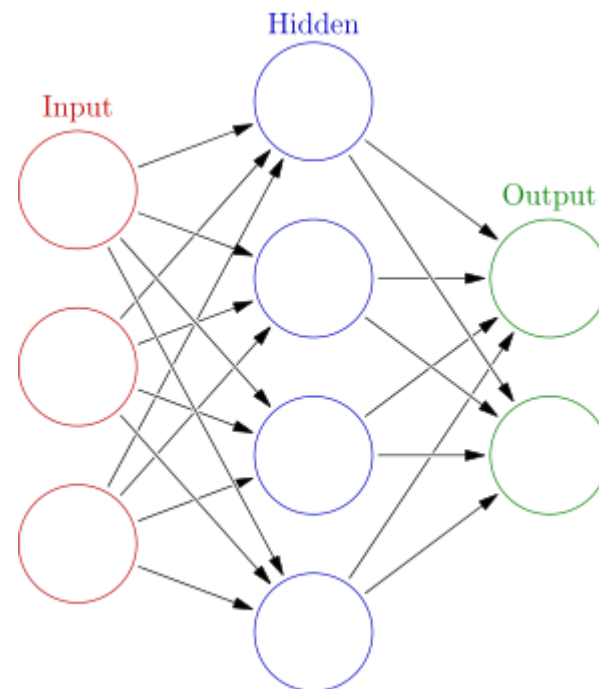
BALANCEAMENTO DE CARGA - CRITÉRIOS

- ✓ Taxa de transferência
- ✓ Custo de balanceamento
- ✓ Tolerância à falhas
- ✓ Tempo de resposta
- ✓ Escalabilidade
- ✓ Utilização de recursos
- ✓ Desempenho



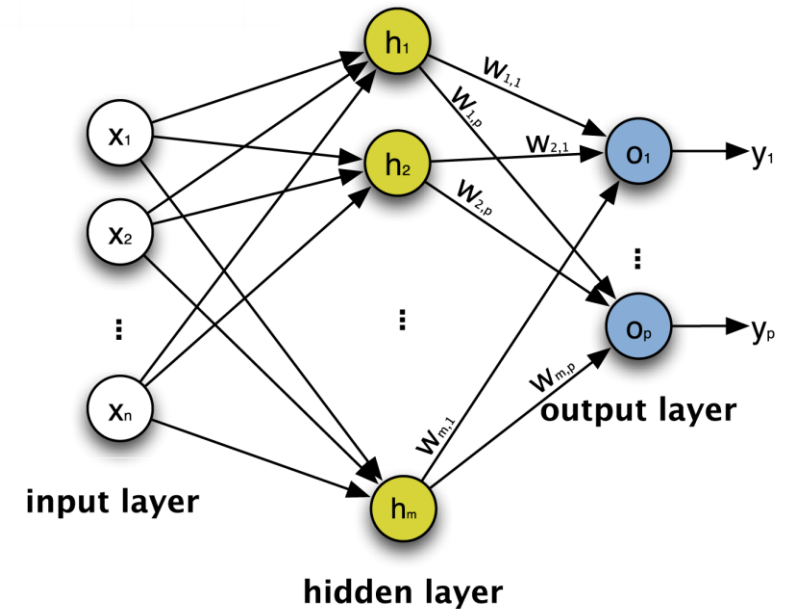
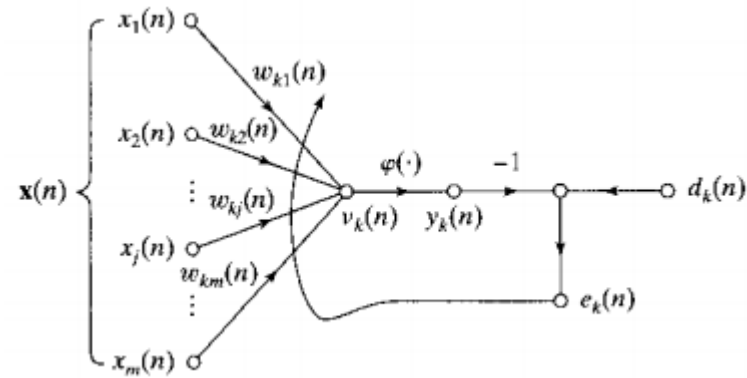
REDES NEURAIS ARTIFICIAIS

- ✓ Estrutura computacional inspirada na estrutura de neurônio para aquisição de conhecimento por meio de experiência
- ✓ Camadas e propagação de sinais



REDES NEURAIS ARTIFICIAIS

- ✓ Fluxo de aprendizado
- ✓ Ajustes
- ✓ Classificação



O PROJETO

Módulo de aplicação de balanceamento utilizando redes neurais artificiais

```
λ Cmder
*****
      Load Balancer
*****
Active servers = 3
-----
Servers:
http://ec2-52-26-51-200.us-west-2.compute.amazonaws.com:8000
http://ec2-35-167-36-190.us-west-2.compute.amazonaws.com:8000
http://ec2-35-167-32-77.us-west-2.compute.amazonaws.com:8000

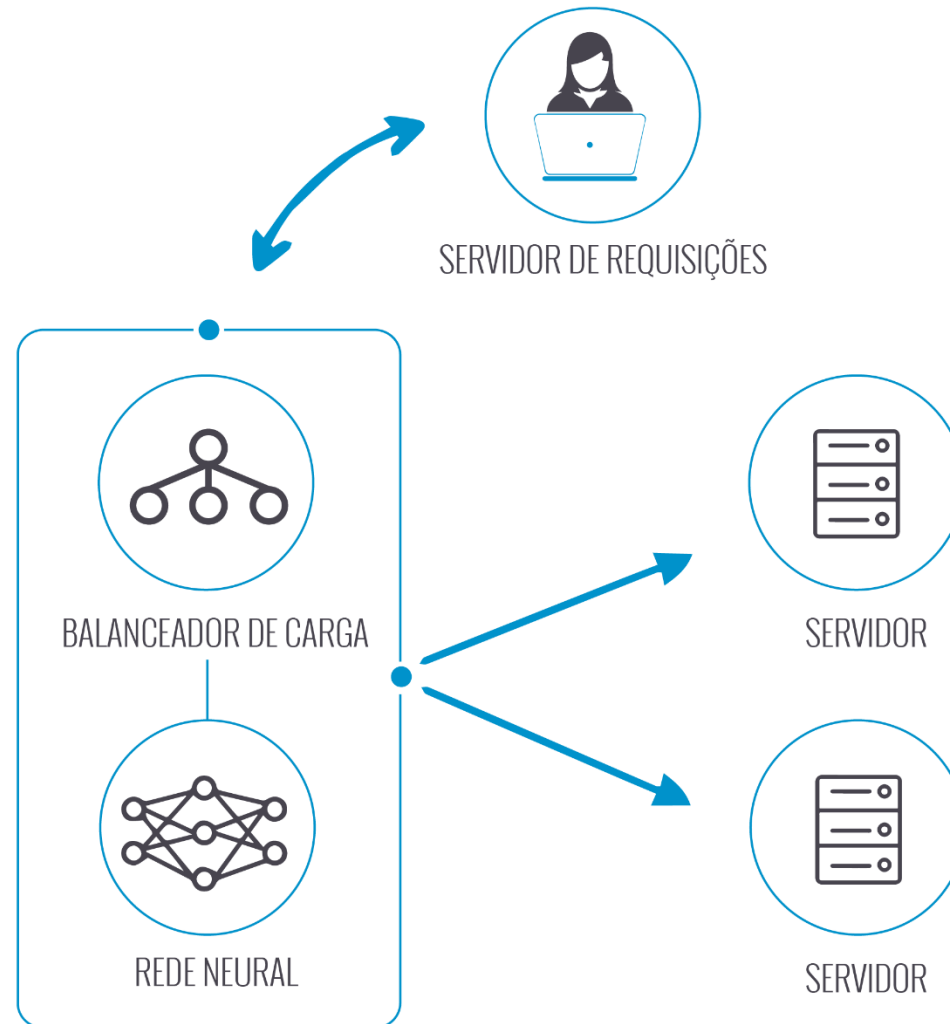
-----
Running load Balancer...
-----
Weights:
responseTime: 102
state: 1
type: 1
distance: 1
█

λ Cmder
*****
      Load Balancer Test
*****
Servers total requests:
http://ec2-35-167-36-190.us-west-2.compute.amazonaws.com:8000: 36
http://ec2-35-167-32-77.us-west-2.compute.amazonaws.com:8000: 35
http://ec2-52-26-51-200.us-west-2.compute.amazonaws.com:8000: 36
```

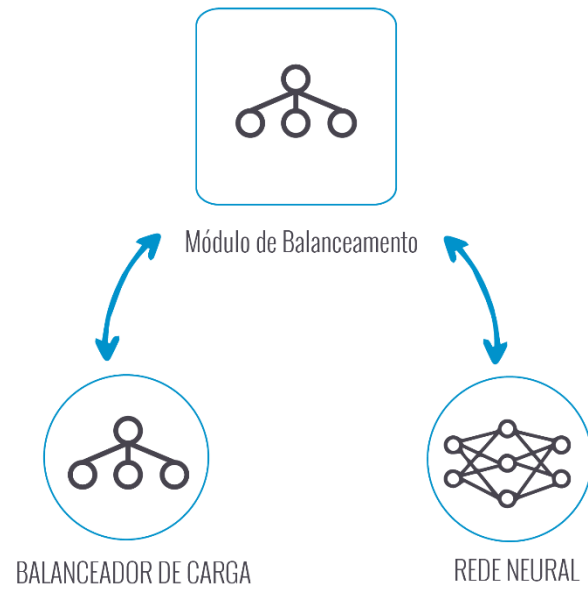
TECNOLOGIAS



ARQUITETURA



MÓDULOS

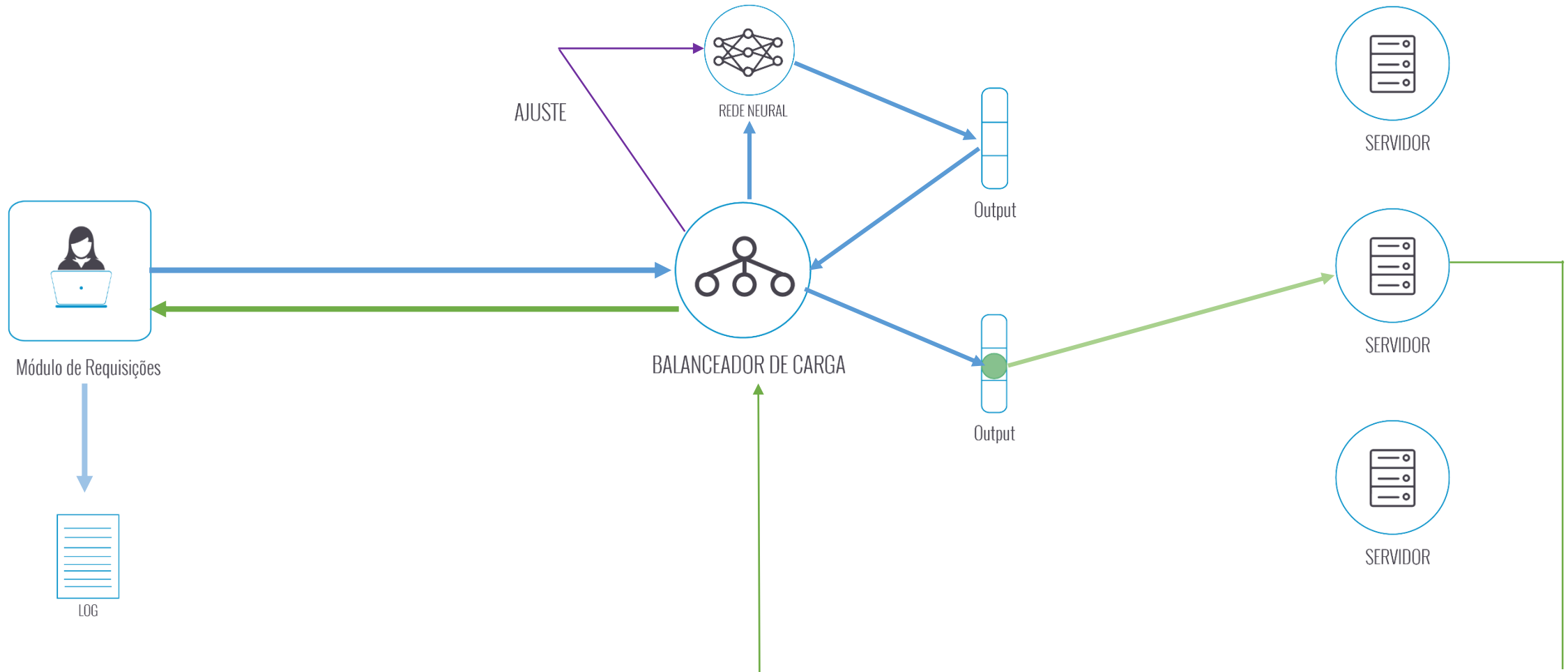


Módulo de Servidores



Módulo de Requisições

FUNCIONAMENTO



MÓDULO DE SERVERS



Módulo de Servidores

- ✓ Servidores do tipo t2.micro
 - ✓ 1 Gb RAM, 1 CPU
- ✓ Recebe requisições e retorna o servidor que respondeu a requisição
- ✓ Alocados em uma mesma região

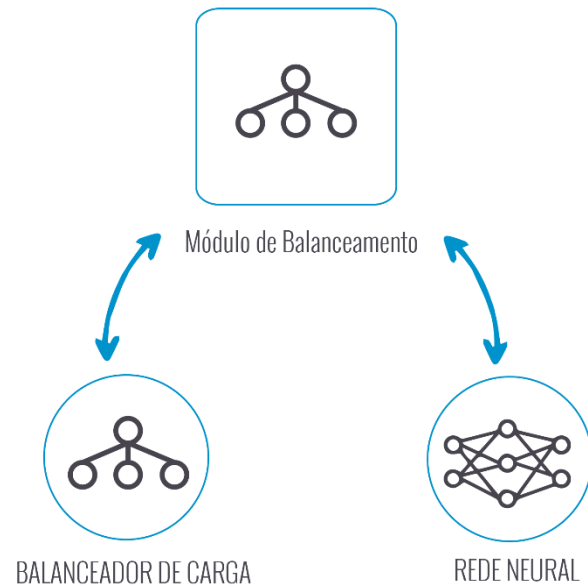
MÓDULO DE SERVIDOR DE REQUISIÇÕES



Módulo de Requisições

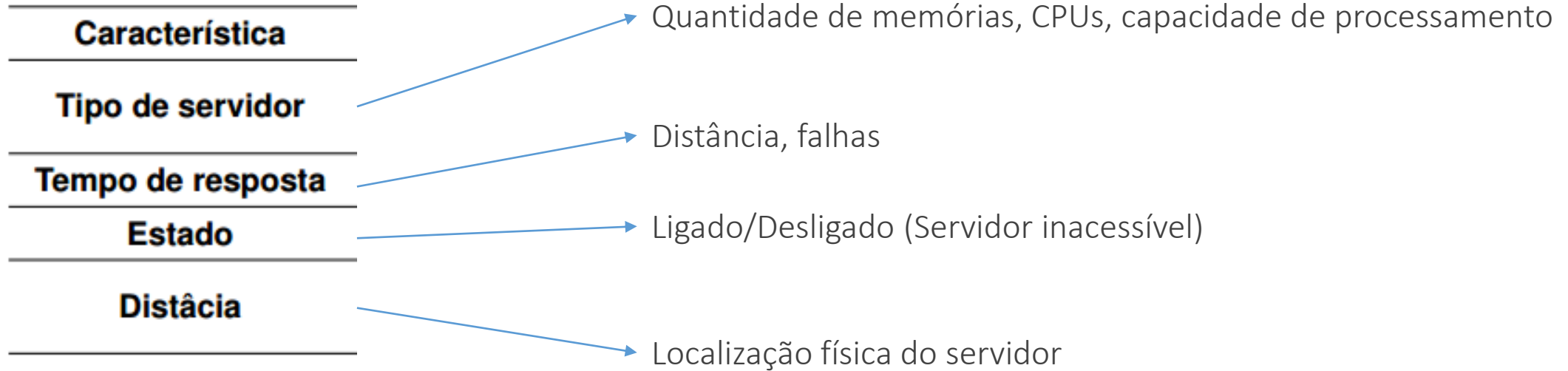
- ✓ Simula um cliente
- ✓ Gera requisições ao servidor de balanceamento
- ✓ Aguarda a resposta
- ✓ Salva um log das requisições e as repostas
- ✓ Utilizado para testes

MÓDULO DE BALANCEAMENTO



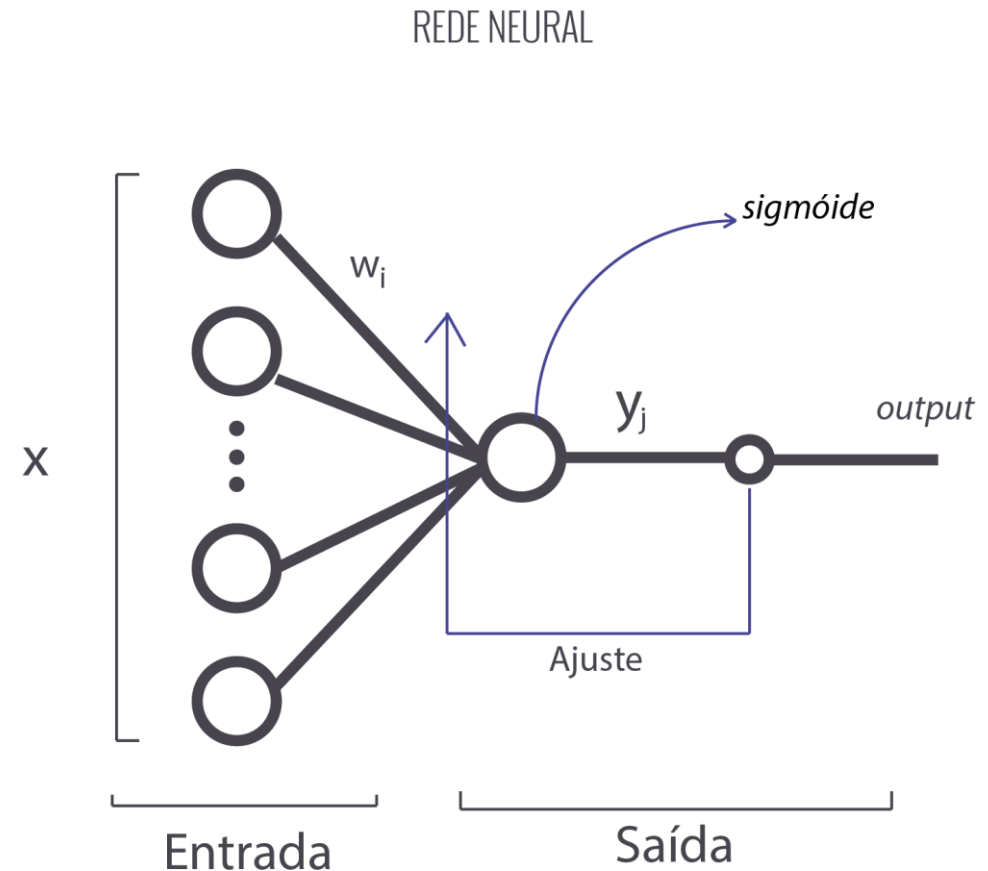
- ✓ Define a melhor rota até o servidor
- ✓ Distribui as requisições
- ✓ Duas aplicações:
 - ✓ Rede Neural
 - ✓ Round Robin

IMPLEMENTAÇÃO DA REDE NEURAL

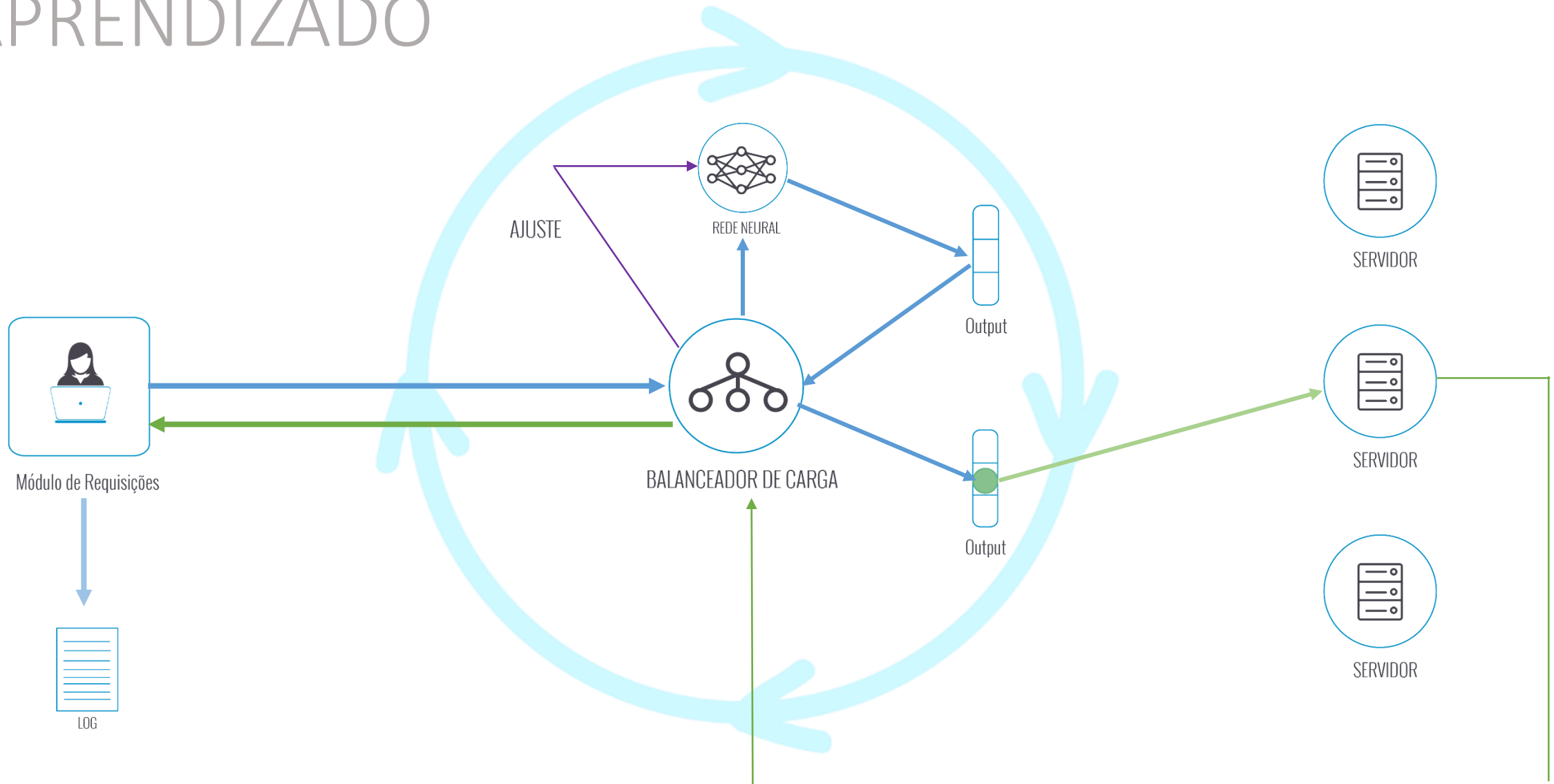


IMPLEMENTAÇÃO DA REDE NEURAL

- ✓ 2 camadas
- ✓ Aprendizado em relação ao servidor
- ✓ Implementação simples – Tempo de resposta
- ✓ Tolerância à falhas



APRENDIZADO



TESTES

```
*****
Load Balancer
*****
Active servers = 3
-----
Servers:
http://ec2-52-26-51-200.us-west-2.compute.amazonaws.com:8000
http://ec2-35-167-36-190.us-west-2.compute.amazonaws.com:8000
http://ec2-35-167-32-77.us-west-2.compute.amazonaws.com:8000

-----
Running load Balancer...
-----
Weights:
responseTime: 231.5
state: 1
type: 1
distance: 1
```

```
D:\_git\ann-load-balance\test (master)
λ node index.js -i 100 -u http://ec2-35-163-65-2.us-west-2.compute.amazonaws.com:8000
^C
D:\_git\ann-load-balance\test (master)
λ
D:\_git\ann-load-balance\test (master)
λ node index.js -i 100 -u http://ec2-35-163-65-2.us-west-2.compute.amazonaws.com:8000
*****
Load Balancer Test
*****
Servers total requests:
http://ec2-35-167-36-190.us-west-2.compute.amazonaws.com:8000: 37
http://ec2-35-167-32-77.us-west-2.compute.amazonaws.com:8000: 37
http://ec2-52-26-51-200.us-west-2.compute.amazonaws.com:8000: 32
|
```

```
D:\_git\ann-load-balance\test (master)
λ node index.js --help

Usage: index [options]

Options:
  -h, --help            output usage information
  -V, --version          output the version number
  -R, --roundrobin       Set Round Robin Output
  -P, --perceptron       Set perceptron output [DEFAULT]
  -O, --output [file]    Set a output log file[default: log.txt]
  -i, --iterations [iterations] Set a number of iterations [default: 1000]
  -u, --uri [uri]        Set a uri with port [default: http://localhost:8000]
```

é quando, ao vivo, a execução não é baseada em testes como nos demais modelos. O Node dispõe funções específicas para isso. Além disso, o Nodejs também possui um gerenciador de pacotes (denominado npm). Seu objetivo é a instalação e bibliotecas criadas pela comunidade e que tenham código fonte disponível para ser disponibilizado sua própria biblioteca.

testes criados da maneira descrita anteriormente têm como objetivo avaliar o desempenho do sistema em condições reais de uso. Para isso, é necessário utilizar ferramentas que possam simular o comportamento dos usuários e gerar uma carga de trabalho realista. O Nodejs oferece uma série de ferramentas para isso, como o npm e o Express.js.

TESTES – Resultados

- ✓ A quantidade de requisições na arquitetura utilizada foi a característica mais determinante
- ✓ Ao desligar um servidor, a rede demora até aprender que ele não deve mais ser considerado
- ✓ Em comparação com o Round Robin, não houve diferença significativa com a RNA

Tabela 3 – Comparação entre Distribuição de requisições entre servidores

| Servidores | Round Robin | RNA | RNA - treinada |
|------------|-------------|-----|----------------|
| Servidor 1 | 34 | 36 | 37 |
| Servidor 2 | 35 | 35 | 37 |
| Servidor 3 | 34 | 36 | 32 |

Elaborado pela autora.

CONCLUSÕES E TRABALHOS FUTUROS

- ✓ Módulo de aplicação funcional
- ✓ Não houve ganho significativo nos testes realizados (ambiente estável)
- ✓ Análise de outras características
- ✓ Testes em ambientes instáveis

REFERÊNCIAS

- ✓ MELL, P.; GRANCE, T. The nist definition of cloud computing. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg, 2011
- ✓ PBXL. SaaS, PaaS, IaaS? 2017. Disponível em: <http://pbxl.co.jp/en/saas-paas-iaas/> . Acesso em: 09 de Janeiro 2017.
- ✓ RACKSPACE. How Do Cloud Load Balancers Work. 2017. Disponível em: <http://es.rack.ly/cloud/load-balancers/how-it-works>. Acesso em: 09 de Janeiro 2017.
- ✓ NUAIMI N. MOHAMED, M. A. N. J. A.-J. K. A. A survey of load balancing in cloud computing: Challenges and algorithms. Second Symposium on Network Cloud Computing and Applications, IEEE Computer Society, p. 137 – 142, 2012.
- ✓ HAYKIN, S. Neural Networks: A Comprehensive Foundation (2nd Edition). [S.l.]: Prentice-Hall, Inc., 1999. 11 p.
- ✓ WIKIPEDIA. File Colored neural network. 2013. Disponível em: https://en.wikipedia.org/wiki/Artificial_neural_network#/media/File:Colored_neural_network.svg . Acesso em: 10 Janeiro 2017.
- ✓ HAYKIN, S. Neural Networks: A Comprehensive Foundation (2nd Edition). [S.l.]:Prentice-Hall, Inc., 1999. 52 p.
- ✓ NODEJS. About Node.js R . 2017. Disponível em:<https://nodejs.org/en/> . Acesso em: 11 Janeiro 2017.
- ✓ SERVICES, A. W. 2016. Disponível em:https://aws.amazon.com/pt/?nc2=h_lg . Acesso em: 07 de Fevereiro de 2017.
- ✓ STACK, A. C. Cloud Stack. 2016. Disponível em: <https://cloudstack.apache.org/> . Acesso em: 14 Maio 2016