

TurboMind

Table of contents

Review Documento	4
Introdução	4
O que é o TurboMind	4
Instalação e uso da DLL	5
Periféricos de intercomunicação	8
Soquete de entrada	8
SoqueteNormalizacaoEntrada	9
Configurando entradas	10
SoquetePadronizacaoEntrada	11
Métodos Comuns	12
Personalizando entrada	12
Soquete de saída	12
SoqueteNormalizacaoSaida	13
Configurando saídas	14
SaidaBitToIndex	14
SaidaDigital	15
SaidaPotencial	15
Métodos Comuns	16
Personalizando saída	16
Metodologias	17
Aprendizado	17
Aprendizado por reforço	17
Escolas	17
ARS	17
Setup de treinamento	19
HYDRA	19
Setup de treinamento	21
Ferramentas	21
Modulo de aprendizado por reforço	21
Métodos comuns	22
Inserir ambiente de treinamento	22
Aprendizado Supervisionado	22
Escolas	22
Deep Learning	23
Setup de treinamento	24
Funções de erro	25
Funcoes erro Enum	25
Inserir ambiente de treinamento	26
Converter CSV para treino	27
Tipo Mapeamento Enum	27
Rede Neural	28
Configurações gerais	29
Configurar BIAS	29
BiasOperation Enum	29
Funções de ativação	30
Active func enum	31
Gerador de Sementes	33

Seed Enum	34
VanishingExplodingConfiger	34
Comportament Enum	35
Habilitação	37
Desabilitação	37
Configurações específicas	37
noise	37
gama	38
Comandos Especial	38
Clone	38
Save Token	38
Agente	39
Instanciando	39
Inserções	40
Inserir metodologia de aprendizado	40
Inserir Rede neural	40
Inserir Soquete de entrada	41
Inserir Soquete de saída	41
Interagindo com ambiente	41
Remoções	42
Remover metodologia	42
Remover rede neural	42
Remover soquete entrada	42
Remover soquete saída	42
Clonagens	42
Clonar agente	43
Clonar metodologia	43
Clonar rede neural	43
Clonar soquete entrada	43
Clonar soquete saída	43
Treinamentos	44
AutoTreinamento	44
Avaliar Performance treinamento	44
Saves GetToken	45
Setups	45
DeepLearning	45
Lógica AND	45
Calibrador dinamico	46
Avaliação IMC	47
ARS	49
Inversor Logico	50
Logica AND	51
Chemical Dependency	52
Hydra	55
Contra Pressão	55

Review Documento

Autor: Gilberto Shimokawa Falcão.
Projeto: TurboMind
Data Criação documento: 29/05/2022.
Versão Framework: Beta 1.0.

Created with the Personal Edition of HelpNDoc: [Easily create EPub books](#)

Introdução

Created with the Personal Edition of HelpNDoc: [Create iPhone web-based documentation](#)

O que é o TurboMind

TurboMind é um framework de rede neural que visa prover apoio e agilidade para o desenvolvedor.

Totalmente desenvolvido em C#.

Possui diversas funcionalidades que podem ser úteis ao desenvolvedor, seja para resolver um problema ou para auto aprendizado de maneira dinâmica e satisfatória, já que podemos montar a rede neural e setar suas configurações em tempo de execução, bem como setar configurações dos módulos das metodologias, que são as escolas de aprendizado das redes neurais, etc.

O sistema do framework pode ou não trabalhar com o agente, onde ele amplifica as utilidades da rede neural já que além de receber:

RedeNeural
metodologia

Pode receber acoplamento de:

Soquete de entrada.
Soquete de saída.

Que uma vez instalados, se conectam automaticamente com a rede em seus periféricos, seja ela inserida antes ou depois.

Esses periféricos também são passivos de configuração e autoconfiguração, personalização, etc.

As metodologias são inseridas dentro do agente porque existe projeto para utiliza-las como Skill passivas das redes neurais, quer seja para treinamento em segundo plano usando o agente como hospedeiro da escola ou para apoio da rede neural, amplificando o poder de realização de trabalho.

Nós encorajamos que você já tenha um conhecimento prévio sobre redes neurais e suas ferramentas, tecnologias, embora teremos prévias explicações sobre elas pois além de nosso framework conter ferramentas tradicionais, estamos inserindo próprias ferramentas para aumentar a experiência nesse maravilhoso mundo das redes neurais.

A versão do sistema esta em beta, pois reconhecemos que ele possa ser passivo de

bugs ou faltar implementações que possam ser desejadas pelos usuários conforme a demanda de necessidade apareça.

Se você tem vontade de fazer uma doação ou colaborar com nosso projeto entre em contato através do Email:

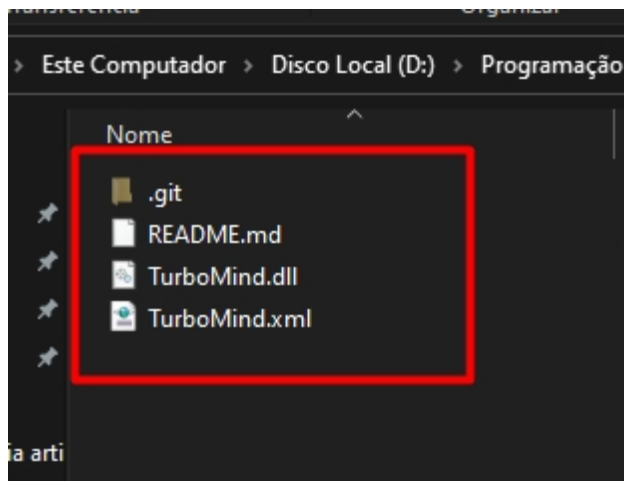
Aceitamos SHIBA INU COIN criptomoedas no endereço:
REDE BEP20 (Binance-Peg)
0xB2531931c42f0b9cbb032c3935250c08Ce7e77CC

Estaremos em breve com uma versão online deste documento para facilitar a consulta dos usuários.

Created with the Personal Edition of HelpNDoc: [Easily create PDF Help documents](#)

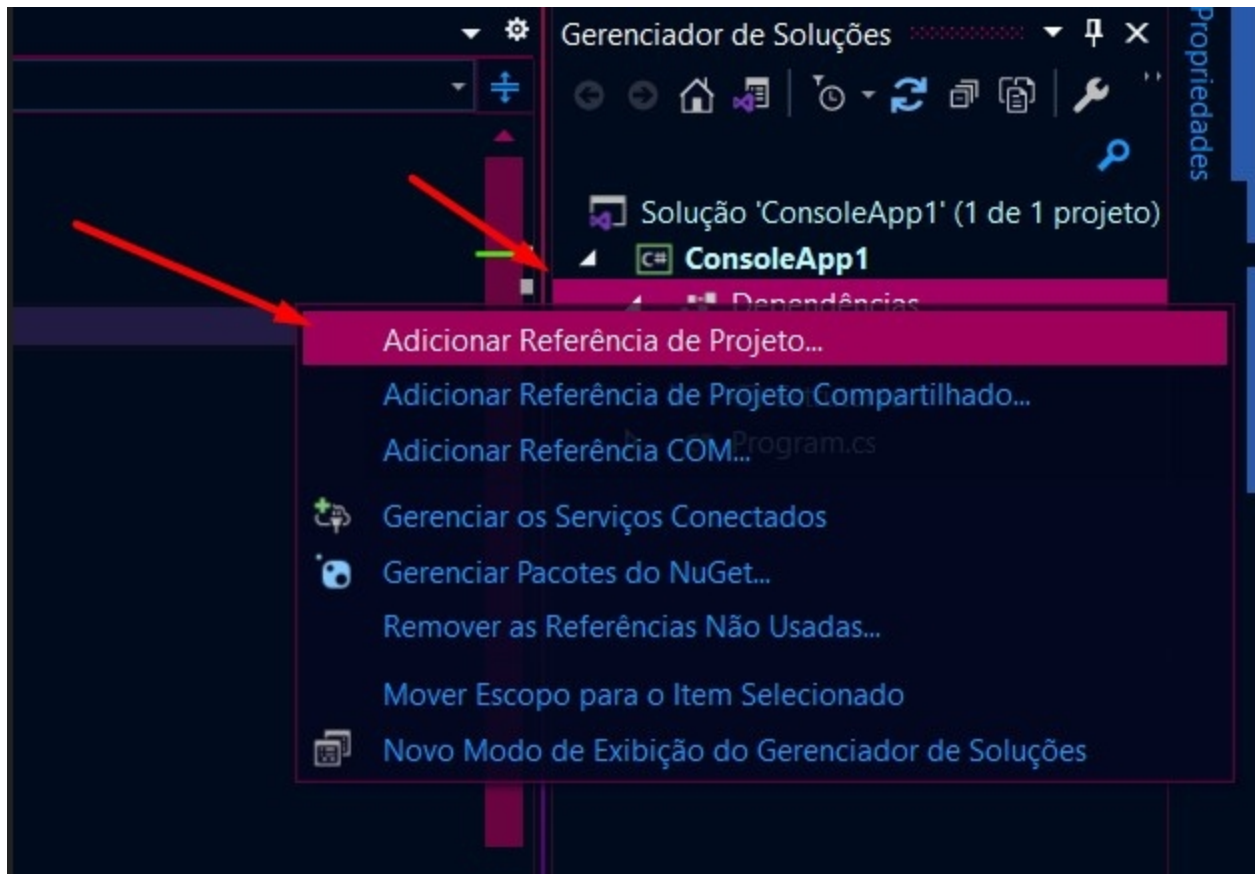
Instalação e uso da DLL

Localize a pasta em seu computador.

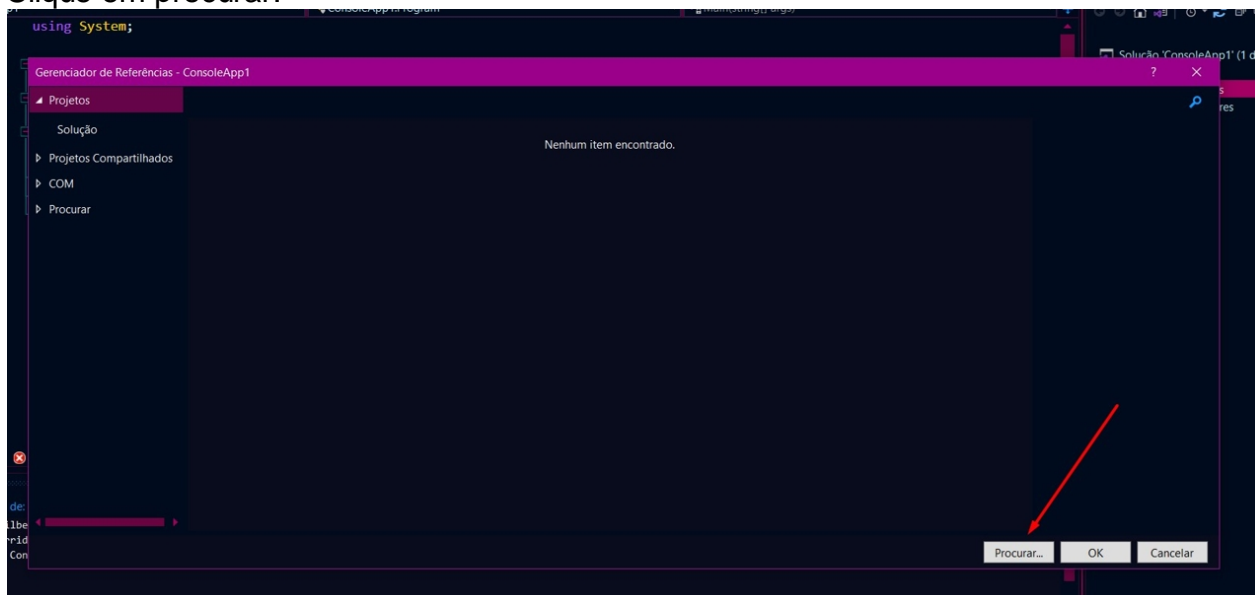


Se for salvar a DLL em uma pasta específica salve com todos os elementos que ela conter, em especial o Turbomind.xml pois ele traz a documentação da DLL.

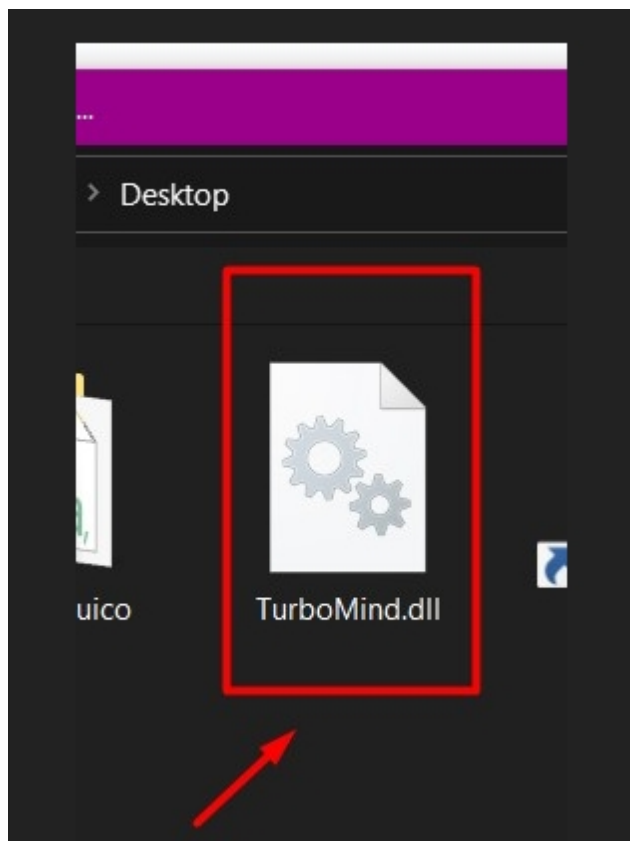
Abra seu projeto, vá na solution explorer e clique sobre a opção "dependências" com o botão direito do mouse e clique em: Adicionar referencia de projeto ou em adicionan referencia COM.



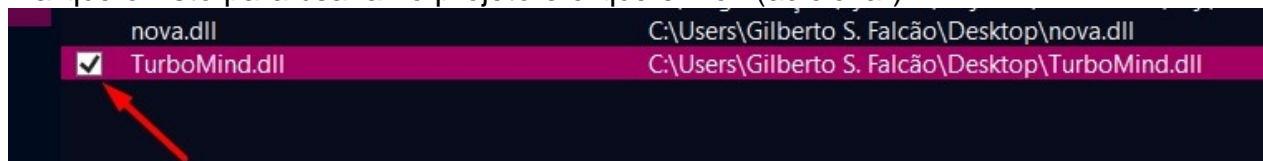
Clique em procurar:



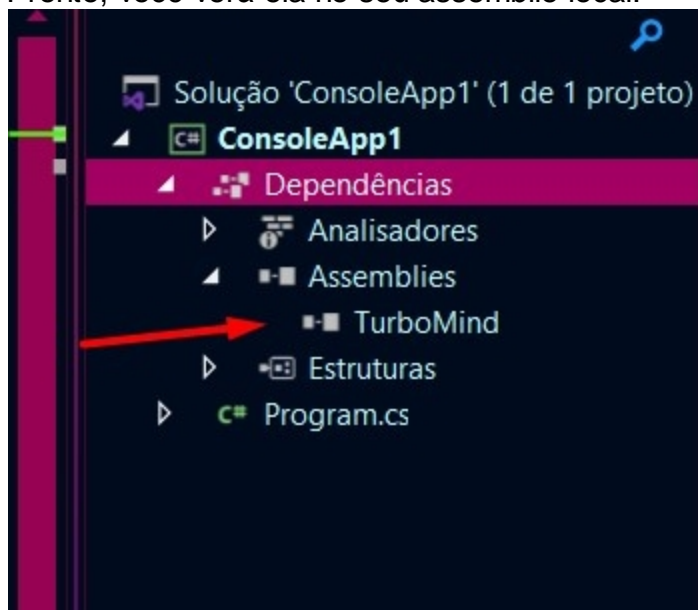
Selecione a DLL TurboMind



Marque o visto para usa-la no projeto e clique em ok (adicionar).

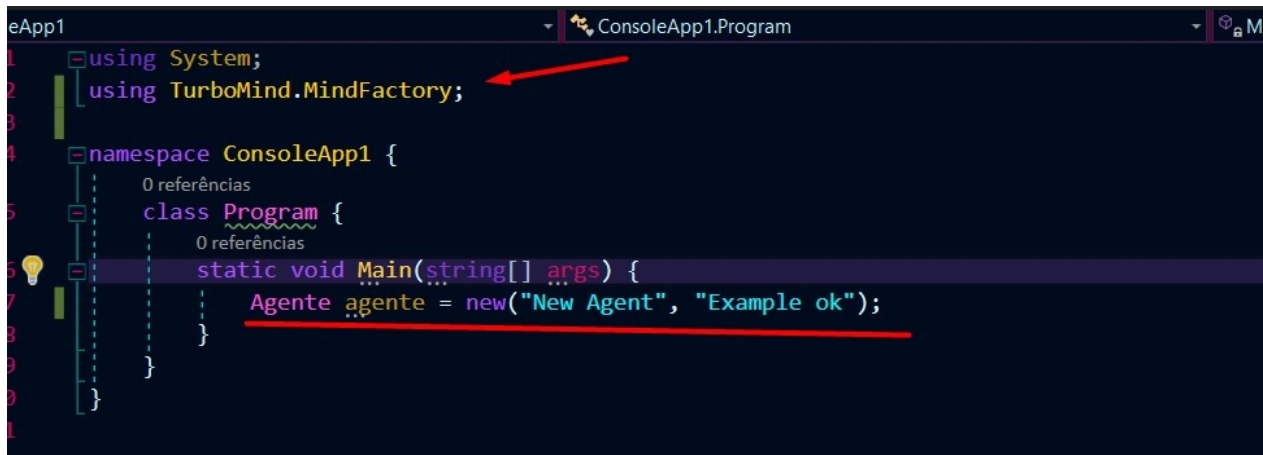


Pronto, você verá ela no seu assembleie local.



No projeto coloque a clausula de uso do pacote TurboMind, por exemplo:
`using Turbomind.MindFactory;`

Abaixo instanciamos o agente, definimos o nome para ele e uma descrição.



```
eApp1 ConsoleApp1.Program
1 using System;
2 using TurboMind.MindFactory;
3
4 namespace ConsoleApp1 {
5     class Program {
6         static void Main(string[] args) {
7             Agente agente = new("New Agent", "Example ok");
8         }
9     }
10 }
```

The screenshot shows a C# code editor with a dark theme. The code is for a console application named ConsoleApp1. It includes the System namespace and TurboMind.MindFactory. Inside the ConsoleApp1 namespace, there is a Program class with a static Main method. The Main method takes a string array args and creates a new Agente object with the name "New Agent" and description "Example ok". A red arrow points to the using statement for TurboMind.MindFactory, and a red underline is under the Agente agente = new line.

Created with the Personal Edition of HelpNDoc: [Easily create CHM Help documents](#)

Periféricos de intercomunicação

Nossos periférios de intercomunicação se limitam até o presente momento em:

Soquete de entrada.
Soquete de saída.

O uso dos soquetes é opcional e nada o impede de criar seu próprio interpretador de entrada ou de saída limitado a seu uso fora do agente e diretamente para a rede neural. Mas não se preocupe, estamos estudando formas mais sofisticadas de você poder herdar o nosso e criar suas próprias soluções.

Created with the Personal Edition of HelpNDoc: [Free CHM Help documentation generator](#)

Soquete de entrada

A missão do soquete de entrada é capturar o dado do vetor do mundo externo e poder transformar esta informação sem perder suas características intrínsecas da informação.



Um dado que tenha tamanho incompatível com o alcance de uma rede neural (pela própria limitação da natureza da rede).
Acaba sendo imperceptível ou se tornando um ruído cujas dimensões excedem a capacidade da rede de interpretar a informação que chega em seus primeiros neurônios.

Por isso o soquete de entrada se torna importante para traduzir essa informação e transcrever para a rede neural em dimensões reconhecíveis.

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

SoqueteNormalizacaoEntrada

Biblioteca:

```
using TurboMind.Intercommunication;
```

Sobre:

Converte valores do mundo real em escalas menores, reconhecíveis para a rede neural.

Instanciando:

```
SoqueteNormalizacaoEntrada CampoDeVisao = new(string NomeSoquete,
string descricaoSoquete, int quantidadeEntradas,
[limiteSaidaMinimaIA = 0], [limiteSaidaMaximaIA = 1]);
```

Parametros obrigatórios:

```
string NomeSoquete:
```

Faz-se interessante ter um nome para seu soquete de entrada.

```
string descricaoSoquete:
```

Permite inserir prévia descrição sobre o seu soquete de entrada.

```
int quantidadeEntradas:
```

Informa para a classe quantas entradas o soquete tem.

Parametros opcionais:

```
limiteSaidaMinimaIA = 0:
```

Define qual o valor mínimo que este soquete envia para a rede. (Padrão 0).

```
limiteSaidaMaximaIA = 1:
```

Define qual o valor máximo que este soquete envia para a rede. (Padrão 1).

A configuração das entradas deste soquete são obrigatórias.

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

Configurando entradas

As entradas do soquete de normalização de entrada devem ser configuradas. Isso porque elas são carentes da informação externa no quesito de maior valor ou menor valor.

Por exemplo, se uma entrada for de velocidade, o sistema precisa de saber qual a velocidade mínima do ambiente externo quanto sua velocidade máxima.

Com essas informações ele consegue normalizar esse dado e colocar no intervalo compreensível para a rede entre 0 a 1 ou até -1 a 1 de acordo com suas configurações.

MODOS VERBOSOS:

Configurando:

```
nomeSoqueteNormalizacaoEntrada.ConfigurarEntrada(int ordenacao,
double valorRealEntradaMinima, double valorRealEntradaMaxima,
[string nomeEntrada], [string descricaoEntrada]);
```

Parametros obrigatórios:

```
int ordenacao:
```

Informa para o soquete de entrada, sobre qual indexação de entrada está sendo configurada.

```
double valorRealEntradaMinima:
```

Informa para a entrada qual é o valor mínimo do ambiente externo que ela está apta a receber.

```
double valorRealEntradaMaxima:
```

Informa para a entrada qual é o valor máximo do ambiente externo que ela está apta a receber.

Parametros opcionais:

```
string nomeEntrada:
```

Permite inserir um nome para a sua entrada: Altura, peso, idade, etc.

`string descricaoEntrada:`

Permite inserir uma descrição para a sua entrada: "Esta entrada mensura as distancias entre corredores".

MODO RÁPIDO:

```
nomeSoqueteNormalizacaoEntrada.ConfigurarTodasEntradas(double
valorRealMinimoGenerico, double valorRealMaximoGenerico, [string
nomeGenerico], [string descricaoGenerica])
```

Parametros obrigatórios:

`double valorRealMinimoGenerico:`

Inserir o menor valor que este ambiente vai proporcionar todas suas entradas.

`double valorRealMaximoGenerico:`

Inserir o maior valor que este ambiente vai proporcionar todas suas entradas.

Parametros opcionais:

`string nomeGenerico:`

Todas entradas receberão nome genérico + indice.

`string descricaoEntrada:`

Todas entradas receberão uma descrição genérica + indice.

DICAS:

Você pode configurar todas entradas que são menos significativas e configurar apenas as que realmente são de maior significância.

Dependendo do modelo de negócio, com sistema que possui várias entradas, pode usar valores médios ou máximos e mínimos ou algoritmo que cria a primeira normalização.

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

SoquetePadronizacaoEntrada

Biblioteca:

```
using TurboMind.Intercommunication;
```

Sobre:

Muito utilizado em aprendizagem por reforço para situações onde não se tem parâmetro correto do ambiente externo onde o agente está se movendo pelo cenário.

Instanciando:

```
SoquetePadronizacaoEntrada meuSoquete = new(string NomeSoquete,
string descricaoSoquete, int quantidadeEntradas);
```

Parametros obrigatórios:

`string NomeSoquete:`

Faz-se interessante ter um nome para seu soquete de entrada.

`string descricaoSoquete:`

Permite inserir prévia descrição sobre o seu soquete de entrada.

`int` quantidadeEntradas:

Informa para a classe quantas entradas o soquete tem.

Configurações automáticas.

Created with the Personal Edition of HelpNDoc: [Full-featured Kindle eBooks generator](#)

Métodos Comuns

Os métodos aqui abordados são comuns para todos soquetes de entrada.

Created with the Personal Edition of HelpNDoc: [Full-featured EBook editor](#)

Personalizando entrada

Opcionalmente você pode personalizar o nome de suas entradas bem como descrições..

MODO VERBOSO:

Personalizando:

Usoquete.PessoalizarEntrada(`int` index, `string` nomeEntrada, `string` descricaoEntrada)

Parametros obrigatórios:

`int` index:

Informa para o soquete de entrada, sobre qual indexação de entrada está sendo personalizada.

`string` nomeEntrada

Insere o nome da entrada para esta indexação

`string` descricaoEntrada:

Insere descrição da entrada para esta indexação

MODO RÁPIDO:

Usoquete.PessoalizarTodasEntradas(`string` nomeGenerico, `string` descricaoGenerica);

Parametros obrigatórios:

`string` nomeGenerico

Insere o nome genérico para todas entradas + indice

`string` descricaoGenerica:

Insere descrição genérica para todas entradas + indice

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

Soquete de saída

Devolvem para o ambiente valores e sinais interpretáveis.



A rede neural processa valores em um intervalo que muita das vezes não é reconhecida no ambiente, precisando do mesmo ser:
Codificado, amplificado, etc.

Created with the Personal Edition of HelpNDoc: [Free HTML Help documentation generator](#)

SoqueteNormalizacaoSaida

Biblioteca:

```
using TurboMind.Intercommunication;
```

Sobre:

Recebe o valor da rede neural e converte em valor interpretável a saída do ambiente externo.

Requer informar ao sistema (via configuração) quais são os parametros interpretáveis para o ambiente externo.

Instanciando:

```
SoqueteNormalizacaoSaida meuSoquete = new(string
nomeSoqueteSaida, string descricaoSoqueteSaida, int
quantidadeSaidas, [double limiteSaidaMinimaDaRede], [double
limiteSaidaMaximaDaRede])
```

Parametros obrigatórios:

```
string nomeSoqueteSaida:
nome do soquete de saída.
```

```
string descricaoSoqueteSaida:
descrição do soquete de saída.
```

```
int quantidadeSaidas:
Quantidade de saídas que o soquete possui.
```

Parametros opcionais:

```
double limiteSaidaMinimaDaRede:
```

Informa para o soquete de saída qual é o valor mínimo que a inteligência artificial exhibe.
(Padrão 0)

`double` limiteSaidaMaximaDaRede:

Informa para o soquete de saída qual é o valor máximo que a inteligência artificial exhibe.
(Padrão 1)

Configuração das saídas são obrigatórias.

Created with the Personal Edition of HelpNDoc: [Full-featured EPub generator](#)

Configurando saídas

As Saidas do soquete de normalização de saída devem ser configuradas para poderem ofertar o ambiente externo, dados que estejam a altura.

MODO VERBOSO:

Configurando:

```
meuSoquete.ConfigurarSaida(int indexacaoSaida, double
menorValorSaida, double maiorValorSaida);
```

Parametros obrigatórios:

`int` indexacaoSaida:

Informa para o soquete qual indexação de saída está sendo configurada.

`double` menorValorSaida:

Informa para a saída qual é o valor mínimo do ambiente externo reconhece.

`double` maiorValorSaida:

Informa para a saída qual é o valor máximo do ambiente externo reconhece.

MODO RÁPIDO:

```
meuSoquete.ConfigurarTodasSaidas(double menorValorNeuralGenerico,
double maiorValorSaidaGenerico)
```

Parametros obrigatórios:

`double` menorValorNeuralGenerico:

Insere o menor valor genérico que este ambiente interpreta.

`double` maiorValorSaidaGenerico:

Insere o maior valor genérico que este ambiente interpreta.

Created with the Personal Edition of HelpNDoc: [Free CHM Help documentation generator](#)

SaidaBitToIndex

Biblioteca:

```
using TurboMind.Intercommunication;
```

Sobre:

Transforma o valor da saída indexada (Sendo esse valor maior que 0) no valor correspondente ao número da indexação daquela saída.
O contrário será -1 e deve ser desconsiderado.

Ex:

Se um valor for maior que 0, e este valor estiver na indexação 5, o valor da saída será um double 5.0.

Se o valor da indexação 5 for menor ou igual a 0, a saída terá um double -1.

Instanciando:

```
SaidaBitToIndex meuSoquete = new(string nomeSoqueteSaida, string
descricaoSoqueteSaida, int quantidadeSaidas);
```

Parametros obrigatórios:

string nomeSoqueteSaida:
nome do soquete de saída.

string descricaoSoqueteSaida:
descrição do soquete de saída.

int quantidadeSaidas:
Quantidade de saídas que o soquete possui.

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

SaidaDigital**Biblioteca:**

```
using TurboMind.Intercommunication;
```

Sobre:

Transforma todos valores de saídas que sejam maiores que zero em valores 1.
Valores iguais ou inferiores a zero ficam em 0.

Instanciando:

```
SaidaDigital meuSoquete = new(string nomeSoqueteSaida, string
descricaoSoqueteSaida, int quantidadeSaidas);
```

Parametros obrigatórios:

string nomeSoqueteSaida:
nome do soquete de saída.

string descricaoSoqueteSaida:
descrição do soquete de saída.

int quantidadeSaidas:
Quantidade de saídas que o soquete possui.

Created with the Personal Edition of HelpNDoc: [Write EPub books for the iPad](#)

SaidaPotencial**Biblioteca:**

```
using TurboMind.Intercommunication;
```

Sobre:

Seta a saída cujo o maior valor corresponder a ela , valor de saída é 1.

Se duas ou mais saídas possuírem mesmo valor, seta todas as saídas empatadas.

Todas saídas empatadas = 1.

Instanciando:

```
SaidaPotencial meuSoquete = new(string nomeSoqueteSaida, string
descricaoSoqueteSaida, int quantidadeSaidas);
```

Parametros obrigatórios:

`string nomeSoqueteSaida:`

nome do soquete de saída.

`string descricaoSoqueteSaida:`

descrição do soquete de saída.

`int quantidadeSaidas:`

Quantidade de saídas que o soquete possui.

Created with the Personal Edition of HelpNDoc: [Easy EPub and documentation editor](#)

Métodos Comuns

Os métodos aqui apresentados são comuns para todos soquetes de saída.

Created with the Personal Edition of HelpNDoc: [Easily create PDF Help documents](#)

Personalizando saída

Opcionalmente você pode personalizar o nome de suas saidas bem como descrições..

MODO VERBOSO:

Personalizando:

```
UsoqueteSaida.PersonalizarSaida(int index, string nomeSaida,
string descricaoSaida)
```

Parametros obrigatórios:

`int index:`

Informa para o soquete de saída, sobre qual indexação está sendo personalizada.

`string nomeSaida`

Insere o nome da saída para esta indexação

`string descricaoSaida:`

Insere descrição da saída para esta indexação

MODO RÁPIDO:

```
UsoqueteSaida.PersonalizarTodasSaidas(string nomeGenerico, string
descricaoGenerica);
```


Parametros obrigatórios:`string` nomeGenerico

Insere o nome genérico para todas saídas + indice

`string` descricaoGenerica:

Insere descrição genérica para todas saídas + indice

Created with the Personal Edition of HelpNDoc: [Full-featured EPub generator](#)

Metodologias

Metodologias de aprendizado.

Created with the Personal Edition of HelpNDoc: [iPhone web sites made easy](#)

Aprendizado

Created with the Personal Edition of HelpNDoc: [Qt Help documentation made easy](#)

Aprendizado por reforço

Escolas que o agente aprende sem um gabarito previamente existente.

Requer que o desenvolvedor crie um cenário para moldar como o agente irá aprender e ser punido no mesmo.

Created with the Personal Edition of HelpNDoc: [Write EPub books for the iPad](#)

Escolas

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

ARS**Sobre:**

SIGLA - Augmented Random Search (ARS)

Escola de aprendizado por reforço (conhecida como aprendizado SHALLOW "Aprendizado superficial").

Uma das redes neurais mais poderosas criadas no ano de 2018 pela univervdade de oxford.

Utilizado para IAs que aprende por reforço.

OBS: A rede deve conter apenas 1 camada de neuronios.**Sintaxe:**`using TurboMind.Methodology.Reinforcement;``ARS escolaARS = new(string nomeEscola, string descricaoEscola);`**Parametros obrigatórios:**`string` nomeEscola

Nome da escola - Escola de motoristas bebados!

`string` descricaoEscola

Insere uma descrição para a escola, motivo de existencia - Educar bebados durante o transito, relação ao pensamento coerente e cognitivo!

Created with the Personal Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

Setup de treinamento

Sobre:

Programa o setup de como a escola irá atuar na educação da rede.

Sintaxe:

```
escolaARS.SetupDeTreinamento(int
qtdMaximaMatrizesAleatoriasGeradas, int
qtdMelhoresMatrizesSelecionaveis, int qtdMaximaDuracaoSessao, int
qtdEpochs, [double? stopPoint = null]);
```

Parametros obrigatórios:

`int qtdMaximaMatrizesAleatoriasGeradas`

Informa para o sistema quantas matrizes aleatórias serão geradas, lembrando que, para cada 1 matriz criada, é gerada uma com valores opostos a esta em relação ao noise da rede.

`int qtdMelhoresMatrizesSelecionaveis`

Critério de fusão, ou seja, dada N matrizes que foram criadas e competidas entre si, resgatar e fundir apenas As melhores matrizes geradas para o novo modelo.

OBS: A quantidade desejável deve ser menor ou igual a quantidade de matrizes produzidas aleatoriamente.

`int qtdMaximaDuracaoSessao`

Informa para a escola que (Dentro de uma época) quanto tempo que ela deve permanecer na sessão de treinamento.

Essa sessão é terminada ou com o fim da época ou com o proprio treinamento chegando no fim ou gerando o game over para esta.

`int qtdEpochs`

Informa para a escola quantas epocas que a rede neural principal irá atualizar.

Lembrando que num cenário por exemplo com 5 redes neurais geradas, você terá +5 opostas, totalizando 10 redes. Então numa atualização de 1000 epocas e 100 sessoes, voce tera $10 * 100 * 1000$ de propagação do tempo e processamento.

Parametros opcionais:

`double? stopPoint = null`

Se inserido um valor, a escola terminará o treinamento quando a rede atingir no minimo a pontuação indicada.

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

HYDRA

Sobre:

Cortesia da Fibborbit Digital Solutions, criada por Salvington em 2021.

Se trata de uma rede neural com algoritmo especial com 8 cabeças que se competem entre si.

Criando um tipo de rede neural altamente devoradora de conhecimento e canibal entre elas, aumentando mais a disputa para não serem destronadas sem ser levada a juri popular e criada uma nova rebelião contra o grupo dos mais fracos a fim de mudar a politica quando sentem que a evolução esta muito parada, então elas derrubam a ditadora.

OBS: A rede deve conter apenas 1 camada de neuronios.

Sintaxe:

```
Hydra hydraSchool = new(string nomeEscola, string  
descricaoEscola);
```

Parametros obrigatórios:

`string` nomeEscola

Nome da escola - Esola de avioes de papel!

`string` descricaoEscola

Insere uma descrição para a escola, motivo de existencia - Ensinar como fazer avioes de papel e identificar aviao de papel e avião de madeira...

Created with the Personal Edition of HelpNDoc: [Full-featured EPub generator](#)

Setup de treinamento

Sobre:

Programa o setup de como a escola irá atuar na educação da rede.

Sintaxe:

```
hydraSchool.SetupDeTreinamento(int
qtdMaximaMatrizesAleatoriasGeradas, int qtdMaximaDuracaoSessao,
int qtdEpochs, [double? stopPoint = null]);
```

Parametros obrigatórios:

[int](#) qtdMaximaMatrizesAleatoriasGeradas

Informa para o sistema quantas matrizes aleatórias serão geradas, lembrando que, para cada 1 matriz criada, é gerada 7 com valores diferenciados e correlacionados a esta.

[int](#) qtdMaximaDuracaoSessao

Informa para a escola que (Dentro de uma época) quanto tempo que ela deve permanecer na sessão de treinamento.

Essa sessão é terminada ou com o fim da época ou com o proprio treinamento chegando no fim ou gerando o game over para esta.

[int](#) qtdEpochs

Informa para a escola quantas epocas que a rede neural principal irá atualizar.

Lembrando que num cenário por exemplo com 5 redes neurais geradas, você terá +35 redes, totalizando 40 redes. Então numa atualização de 1000 epocas e 100 sessoes, voce tera $40 * 100 * 1000$ de propagação do tempo e processamento.

Parametros opcionais:

[double](#)? stopPoint = [null](#)

Se inserido um valor, a escola terminará o treinamento quando a rede atingir no minimo a pontuação indicada.

Created with the Personal Edition of HelpNDoc: [Free Kindle producer](#)

Ferramentas

Created with the Personal Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

Modulo de aprendizado por reforço

Sobre:

Um modulo que conversa com as escolas de aprendizado por reforço de nossa base (Por enquanto) ;).

Com ele voce desenvolverá sua arquitetura e cenário para aprendizado de sua rede, recompensando-a ou punindo-a.

Sintaxe:

```
[Serializable]
class minhaClasse: IModuloAprendizadoReforcoAdapter{

    //Reset do seu projeto, padronização de pontos etc..
```

```

public double[] Reset(){
}

//Recebe estímulos da rede neural, converge essas ações e
devolve a sequência de:
//[Novos Dados Ambiente] ; Pontuação ; Fim de jogo(true) ou
não (false).
public(double[], double, bool)Action(double[]
EstimuloNeural){
}
}

```

Temos em nossos setups alguns exemplos!

Created with the Personal Edition of HelpNDoc: [Produce online help for Qt applications](#)

Métodos comuns

Métodos comuns para as escolas de reforço.

Created with the Personal Edition of HelpNDoc: [Free EPub producer](#)

Inserir ambiente de treinamento

Sobre:

Por enquanto, nossas redes de aprendizado por reforço recebem o `IModuloAprendizadoReforcoAdapter`.
Para que você possa construir seus cenários.

Sintaxe:

`EscolaAprendizadoReforco.InserirAmbienteTreinamento(IModuloAprendizadoReforcoAdapter moduloAprendizadoReforcoAdapter);`

Parâmetros obrigatórios:

[IModuloAprendizadoReforcoAdapter](#) moduloAprendizadoReforcoAdapter
Nosso módulo de aprendizado por reforço, após codar nele caso houver necessidade de clonagens, assinar a notação `[Serializable]` sobre a classe do módulo assinado.

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

Aprendizado Supervisionado

Escolas onde o agente precisa da situação problema e de seu gabarito para poder aprender.
São as redes mais clássicas.

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

Escolas

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

Deep Learning

Sobre:

Rede mais classica de todas e base para as demais.
Suporta rede neurais de 1 a N camadas.

Sintaxe:

```
using TurboMind.Methodology.Supervised;
```

```
DeepLeaning deepSchool = new(string nomeEscola, string  
descricaoEscola);
```

Parametros obrigatórios:

`string` nomeEscola

Nome da escola - Escola de artes.

`string` descricaoEscola

Insere uma descrição para a escola, motivo de existencia - Esta escola ensina como agentes devem subir em carteiras e colar em provas!

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

Setup de treinamento

Sobre:

Programa o setup de como a escola irá atuar na educação da rede.

Sintaxe:

```
escolaARS.SetupDeTreinamento(int qtdEpochs, [bool  
acompanharTreinamento = false], [int apartirDaEpoch = 1]);
```

Parametros obrigatórios:

int qtdEpochs

Informa para a escola quantas epocas que a rede será submetida a treino.

Parametros opcionais:

bool acompanharTreinamento

O padrão vem desligado, mas se colocado como **true**, é possível acompanhar o treinamento no prompt de comando, recomendamos colocar números altos para esta opção, pois o acompanhamento de treino onera a velocidade e desempenho.

int apartirDaEpoch

Vem com o padrão 1, mas é altamente recomendado colocar valores próximos da quantidade de epochs.

Se acionado, o treinamento no prompt é exibido quando a rede chegar na epoch que coincide com o valor inserido aqui.

Funções de erro

Sobre

A escola DeepLearning suporta funções de erro.
Elas moldam como o agente deve aprender com seus erros.

Sintaxe:

```
using TurboMind.Methodology;

deepSchool.ConfigurarFuncaoErro(Enum FuncaoErro);
```

Parametros obrigatórios:

Enum FuncaoErro

Tipo da função de erro, vide Enum para mais detalhes.

OBS: Os valores de Alfa e da constante c, por enquanto são impraticáveis.

Created with the Personal Edition of HelpNDoc: [Free PDF documentation generator](#)

Funcoes erro Enum

SingleLoss

Subtrai O Valor desejado pelo obtivo apenas.

AbsoluteLoss

Permite suporte a regressão vetorial, mas pode criar uma aprendizagem ruim. Ele não é otimizado através da decida do gradiente.
Mas é otimizado nos cálculos de subgradientes.

SquaredLoss

Subtrai o desejado pelo obtido e eleva ao quadrado, isso hipersensibiliza a rede.
Não é interessante se houver valores muito discrepantes.

Created with the Personal Edition of HelpNDoc: [Create HTML Help, DOC, PDF and print manuals from 1 single source](#)

Inserir ambiente de treinamento

Sobre:

Nosso suporte é de duas listas, sendo uma da situação problema e a outra de gabarito no formato `List<double[]>` duas dimensões.

Sintaxe:

```
escolaAprendizadoSupervisionado.InserirAmbienteTreinamento(List<double[]> situacaoProblema, List<double[]> gabarito);
```

Parametros obrigatórios:

`List<double[]>` situacaoProblema

Lista com as situações de problemas (Temos exemplo no setup deste documento).

`List<double[]>` gabarito

Gabarito relacionado a situação problema.

O sistema relaciona cada indexação de problema com o gabarito.

Created with the Personal Edition of HelpNDoc: [Create cross-platform Qt Help files](#)

Converter CSV para treino

Sobre:

É claro que não iríamos deixar nossos queridos entusiastas sem alguma coisinha para brincar!

Por isso a escola Possui função para converter e gerar programa de treinamento!
Transformando CSV em treinamento!

Sintaxe:

```
escolaAprendizadoSupervisionado.GerarPriogramaTreinamento(string
atalhoTabela, TipoMapeamento modoMapeamentoTabela, [string
prefixoEntrada = "In:"], [string prefixoSaida = "Out:"]);
```

Parametros obrigatórios:

string atalhoTabela

É o path! Caminho de onde seu arquivo.csv se encontra!

TipoMapeamento modoMapeamentoTabela

Informa para a escola qual lógica de mapeamento da tabela será usada para gerar o programa de treino!

Vide Enum para mais detalhes!

O sistema relaciona cada indexação de problema com o gabarito.

Parametros opcionais:

string prefixoEntrada = "In:"

Dependendo do treinamento selecionado, precisará informar e marcar na tabela quais serão os campos relacionados a situação problema!

Nosso Default está como In: isso significa que em seu CSV, qualquer campo que for considerado dado de entrada, precisa de por em seu cabeçalho a palavra In:
(Considerar os dois pontos).

string prefixoSaida = "Out:"

Dependendo do treinamento selecionado, precisará informar e marcar na tabela quais serão os campos relacionados ao gabarito!

Nosso Default está como Out: isso significa que em seu CSV, qualquer campo que for considerado dado de resposta, precisa de por em seu cabeçalho a palavra Out:
(Considerar os dois pontos).

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

Tipo Mapeamento Enum

ENTRADA_SAIDA

Mapeia cabeçalhos das colunas timbradas com prefixos de entrada e saída fornecidos. Nesse caso, na tabela previamente deve existir com prefixos de entrada e saída obedecendo a sequencia:

NomePrefixo: atributo.

MIRROR

Considera todas colunas de uma tabela como argumento de entrada e se saída. Faz uma cópia da situação problema para o Gabarito.

Rede Neural

A rede neural irá processar as informações tratadas e devolver array de dados pertinentes como resposta para o meio.



Instanciando:

Biblioteca:

```
using TurboMind.MindFactory;
```

MODO 1 - NOVA INSTANCIA

```
RedeNeural rede = new(double taxaAprendizado, int  
quantidadeEntradas, params int[] quantidadesNeuroniosPorLayer)
```

Parametros Obrigatórios:

`double taxaAprendizado`

Informa qual é a taxa de aprendizado da rede neural.

`int quantidadeEntradas`

Informa qual a quantidade de entradas esta rede irá possuir, isso significa que todos os neuronios da primeira layer irão possuir tantos sensores quanto o número de entradas.

`params int[] quantidadesNeuroniosPorLayer`

Informa quantos neuronios cada camada (layer) irá ter, as camadas são separadas por virgula, sendo assim o número de cada valor representa a quantidade de neuronios naquele espaço.

Ex: [2,3,2] => 2 neuronios na entrada, 3 na camada oculta, 2 na camada de saída.

MODO 2 - RESGATE

```
RedeNeural redeAnterior = new(string neuroToken, [string secret],
```

```
[bool useNewId = false])
```

Parametros Obrigatórios:

```
string neuroToken
```

Token da rede neural, gerado anteriormente, onde contem a criptografia do modelo.

Parametros Opcionais:

```
string secret
```

Insere a chave ou segredo para descriptografar o token, a chave deve ser guardada para ser utilizada posteriormente para descriptografar a rede.

```
bool useNewId
```

Se false, traz o id anterior para a rede, se true, gera um novo id para esta rede, mantendo todos os dados anteriores, independente se true ou false.

Created with the Personal Edition of HelpNDoc: [Easy to use tool to create HTML Help files and Help web sites](#)

Configurações gerais

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

Configurar BIAS

Sobre:

Permite configurar uso de BIAS na rede neural bem como range de trabalho de forma opcional.

Sintaxe:

```
redeNeural.ConfigurarBIAS([enum BiasOperation],[double  
startBiasVal],[double finishBiasVal]);
```

Parametros opcionais:

```
enum BiasOperation
```

Informa para a rede neural o comportamento do BIAS.

Utilizamos como default a operação em que a calibração dos pesos dele varia entre -1 e +1 Independente se startBiasVal ou finishBiasVal possuem valores diferentes.

Consulte nosso manual para ver outros comportamentos.

```
double startBiasVal
```

Informa para a rede neural qual o menor valor que o BIAS pode atingir ou valor iniciado no sistema, dependendo da operação Enum selecionada.

```
double finishBiasVal
```

Informa para a rede neural qual o maior valor que o BIAS pode atingir ou limite de restrição de acordo com a operação do Enum selecionado.

Created with the Personal Edition of HelpNDoc: [Create iPhone web-based documentation](#)

BiasOperation Enum

DEFAULT

Operação padrão, permite que o BIAS varie em entre 1 e -1 automaticamente, qualquer valor que o sistema gerar que possa exceder esses valores, é impedido que o faça mantendo até o seu limite.

NOT

BIAS não será utilizado no modelo.

OP1

BIAS varia de 0 a 1.

ON1

BIAS varia de 0 a -1.

JAILBREAK

BIAS Livre para operar em qualquer valor (Excedendo inclusive +1 ou -1) ao infinito se necessário. startBIASVal e finishBIASVal são desconsiderados.

RANGE

BIAS Opera em valores pré estabelecidos pelo usuário variando nos valores do intervalo de startBIASVal e finishBIASVal. **OBS:** Valor de startBIASVal deve ser Menor que finishBIASVal.

STARTER

Todos os BIAS da rede neural é iniciado com o valor desejado, use startBIASVal para colocar esse valor, o campo finishBIASVal é ignorado.

CONSTANT

O Valor do BIAS será constante e decidido pelo desenvolvedor na hora que configura o mesmo.

Use startBIASVal para colocar este valor. Campo finishBIASVal é ignorado.

Created with the Personal Edition of HelpNDoc: [Free EPub and documentation generator](#)

Funções de ativação

Sobre:

Permite que a rede neural use funções de ativação personalizada por layer.

OBS: A função de ativação padrão é a identity (identidade).

Ela é uma função neutra que não distorce e nem influencia o aprendizado.

Sintaxes:**MODO**

VERBOSO:-----

```
redeNeural.InserirEditarFuncaoAtivacao(int indexLayer, ActiveFunc
tipoFuncao, [double alpha]);
```

Parametros obrigatórios:

`int indexLayer`

Informa para a rede neural qual indexação de layer receberá a função de ativação.

ActiveFunc tipoFuncao

Enum do tipo de função de ativação que será utilizado. Vide Active Func Enum para mais detalhes.

Parametros opcionais:

double alpha

Valor opcional e serve para ajudar como constantes para algumas funções de ativação, no Enum colocamos detalhes de quais função de ativação o uso do alpha se faz presente.

MODO RÁPIDO - Todas layers com mesma função de ativação.

```
redeNeural.InserirEditarTodasFuncoesAtivacao(ActiveFunc
tipoFuncao, [double alpha]);
```

Parametros obrigatórios:

ActiveFunc tipoFuncao

Enum do tipo de função de ativação que será utilizado. Vide Active Func Enum para mais detalhes.

Parametros opcionais:

double alpha

Valor opcional e serve para ajudar como constantes para algumas funções de ativação, no Enum colocamos detalhes de quais função de ativação o uso do alpha se faz presente.

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

Active func enum

ArcTan:

ArcoTangente

NÃO UTILIZA alpha.

BentIdentity

Função que hipersensibilizada de valores. Gerando um ganho maior na sua saída e calibrações maiores no sistema.

NÃO UTILIZA alpha.

Degrau

Função simples e digital, define saídas 1 ou 0 de acordo com o limite estabelecido.

NÃO UTILIZA alpha.

ELU

FAZ USO DE Alpha. Valores recomendados a partir de 0.01.

Função de ativação mais utilizada pelas escolas modernas.

Referencias:

https://en.wikipedia.org/wiki/Activation_function#Comparison_of_activation_functions

Gaussian:

Ref: https://en.wikipedia.org/wiki/Activation_function#Comparison_of_activation_functions
 NÃO UTILIZA alpha.

GELU:

Gaussian Error Linear Unit

Ref: https://en.wikipedia.org/wiki/Activation_function#Comparison_of_activation_functions
 NÃO UTILIZA alpha.

Identity:

Função padrão! Não altera saída do neurônio. Utilizada em camadas de saídas de redes neurais de regressão.

NÃO UTILIZA alpha.

LeakyReLU:

FAZ USO DE Alpha. Valores recomendados a partir de 0.01.

ref: https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html

Mish:

Ref: https://en.wikipedia.org/wiki/Activation_function

NÃO UTILIZA alpha.

PReLU:

Parametric Rectified Linear Unit

FAZ USO DE Alpha. Valores recomendados a partir de 0.01.

Ref: https://en.wikipedia.org/wiki/Activation_function#Comparison_of_activation_functions

ReLU

Ativação Linear Retificada

Ref: https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html

NÃO UTILIZA alpha.

SELU

ref: <https://paperswithcode.com/method/selu#:~:text=Scaled%20Exponential%20Linear%20Units%2C%20or,1.6733%20and%20%CE%BB%20%E2%89%88%201.0507%20.>

NÃO UTILIZA alpha EXTERNO. Seu alpha interno vale:

1.6732632423543772848170429916717 e Lambda:

1.0507009873554804934193349852946;

Sigmoide:

Função classica, alguns a chamam de OLD SCHOOL, comumente utilizada em redes neurais de propagação positiva (FeedForWard).

Tem saidas classificatórias e positivas em redes neurais multi camadas e em outras de sinais contínuo.

NÃO UTILIZA alpha.

SILU:

Sigmoid linear unit - SiLU Sigmoid shrinkage SiL or Swish- 1

Ref: https://en.wikipedia.org/wiki/Activation_function#Comparison_of_activation_functions

NÃO UTILIZA alpha.

Sinc:

Ref: <https://sefiks.com/2018/01/07/sinc-as-a-neural-networks-activation-function/>
NÃO UTILIZA alpha.

Sinusoidal

ref: https://www.gabormelli.com/RKB/Sinusoidal_Activation_Function
NÃO UTILIZA alpha.

SoftExponential

FAZ USO DE Alpha. Valores recomendados a partir de 0.01.

Ref: https://www.gabormelli.com/RKB/Soft_Exponential_Activation_Function

SoftPlus:

Ref: https://en.wikipedia.org/wiki/Activation_function#Comparison_of_activation_functions
NÃO UTILIZA alpha.

SoftSing

Activation functions play pivotal role in neural networks.

As an alternative to hyperbolic tangent, softsign is an activation function for neural networks.

Even though tanh and softsign functions are closely related, tanh converges exponentially whereas softsign converges polynomially.

Even though softsign appears in literature, it would not be adopted in practice as much as tanh.

Ref: <https://towardsdatascience.com/soft-sign-activation-function-with-tensorflow-manual-back-prop-with-tf-5a04f3c8e9c1>

NÃO UTILIZA alpha.

TanH:

Sigla de Tangente Hiperbólica possui saída muito comum em redes neurais, é melhor escolha do que a sigmóide, pois explora tanto saídas positivas +1 até negativas -1.

ref: <https://towardsdatascience.com/soft-sign-activation-function-with-tensorflow-manual-back-prop-with-tf-5a04f3c8e9c1>

NÃO UTILIZA alpha.

MaxOut:

NÃO UTILIZA alpha.

SoftMax:

Utilizada em redes neurais de classificação (Principalmente de imagens), Forçando saída da rede neural a representar a probabilidade dos dados de uma das classes definidas. Sem elas as saídas dos neurônios são simplesmente valores numéricos onde o maior indica probabilidade vencedora

Ref: https://www.gabormelli.com/RKB/Soft_Exponential_Activation_Function

NÃO UTILIZA alpha.

OBS: Não otimizamos esta função (ainda).

Created with the Personal Edition of HelpNDoc: [Free PDF documentation generator](#)

Gerador de Sementes

Sobre:

Personaliza a qualidade dos pesos aleatórios gerados para nova rede neural para as escolas de aprendizado.

Sintaxe:

```
rede.ConfigurarSeed(enum seed);
```

Parametros opcionais:

enum seed

Informa o tipo de semente que será utilizada.

Nosso padrão é o GAUSSIAN_NORMAL_DISTRIBUTION.

Vide Enum para mais detalhes.

Created with the Personal Edition of HelpNDoc: [Free HTML Help documentation generator](#)

Seed Enum

GAUSSIAN_NORMAL_DISTRIBUTION

Gera valores randômicos levando em consideração a dispersão gaussiana com valores comuns entre -1 e 1,

Raros entre -2 e 2 e "Legendariamente" raros entre -3 e 3.

RANDOMIC_DOUBLE

Gera valores aleatórios diversos que variam de 0 a 9 sem obedecer padrões.

Em breve teremos outros tipos de sementes.

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

VanishingExplodingConfiger**Sobre:**

Uso opcional.

Um dos piores desafios das redes neurais é:

Vanishig gradient, ou vazamento do gradiente, quando os pesos da rede neural se tornam tão ínfimos que impedem da rede aprender ou

Exploding gradient, é a explosão, quando as saídas da rede neural setam todas as saídas com valores enormes, generalizando tudo e impedindo também o aprendizado.

Muito se tem feito para corrigir este problema. Inclusive as funções de ativação são utilizadas para "contornar este problema".

Mas que tal poder usar um pouco das duas coisas?

Trazemos uma solução que em alguns casos, fará ser desnecessário até o uso da função de ativação, embora deixaremos aqui uma frase que será muito repetida, que esta ferramenta não deve ser usada sozinha sem a função de ativação, mas sim como um complemento para ela.

Todavia, nem em todos os projetos o uso dela será uma solução, assim como da função de ativação poderá não ser e vice verso. Cabendo ao Desenvolvedor tomar suas próprias decisões.

Sintaxe:

```
redeNeural.VanishingExplodingPolicyConfiger(enum Comportament],
```

```
[int creditUpdate],[double valExploding], [double valVanish],[int
precisao]);
```

Observação: Por default, o VanishingExplodingPolicyConfigurer vem desabilitado. Requer assinatura mínima no código para ativar o booleano.

```
redeNeural.VanishingExplodingPolicyConfigurer();
```

Parametros opcionais:

```
enum Comportament
```

Influência o comportamento do policiamento da rede neural em relação as saídas de seus neurônios.

Nosso default é o Relative on faz correções mínimas nos pesos e bias para calibrar a saída até desse neurônio dentro dos limites permitidos e determinado pelo desenvolvedor nos valores de valExploding e valVanish.

```
int creditUpdate
```

Quantidade de tentativas que a rede neural pode tentar corrigir o valor por neurônio, pode acontecer da rede ter que processar mais vezes algoritmos específicos com valores muito extremos.

```
double valExploding
```

Informa para a rede qual maior valor de saída é permitido (antes de passar pela função de ativação).

Se o valor exceder o permitido, ativa o comportamento determinado para correção desse peso.

```
double valVanish
```

Informa para a rede qual menor valor de saída é permitido (antes de passar pela função de ativação).

Se o valor exceder o permitido, ativa o comportamento determinado para correção desse peso.

```
int precisao
```

Critério de casas decimais a considerar e comparar na hora de efetivamente ativar o algoritmo de correção.

Quanto menor o número de casas decimais, menor o processamento exigido, quanto maior, mais processamento envolvido.

Created with the Personal Edition of HelpNDoc: [Generate Kindle eBooks with ease](#)

Comportament Enum

AMNESIA:

Após ultrapassar limites estabelecidos, o neurônio perde a identidade, gerando novos valores de pesos e BIAS aleatórios, podendo gerar nova energia de saída que pode ou não ter bom resultado.

AUDACIOUS:

Após exceder seus limites de explosão ou vazão do gradiente, automaticamente ajustará seus pesos e BIAS para se manter no maior limite do valor permitido.

"Ousadia é uma questão de sorte ou azar, basta se arriscar."

BEASTE :

Se a saída do neurônio explodir, os pesos se ajustam para ficar na extremidade do Vanish, se a saída do neurônio Vazar, ajusta pesos e BIAS para ser o extremo limite da explosao. Arranque de pontos muito altos, indice de acertividade mediano e acidental no inicio.

"Após um tempo de fúria, precede um aquietamento envolto de mistério, o sistema começa a controlar ela e os resultados começam a aparecer. Nesse momento, fúria e precisão começam a conversar. Se conseguir dominar a besta, não haverá limites para o aprendizado. Com o tempo a besta se cansa e começa a seder com o treino."

LUCK :

Escolhe um tipo de comportamento na sorte. E este se torna o padrão (Pra sempre).

METAMORFO :

Quando a saída excede o valor, aplica um comportamento aleatório (Diferente de METAMORFO e LUCK).

NEUTRONIO :

Ao exceder seus limites estabelecidos, neutraliza seus pesos e BIAS para que a energia seja o valor mediano de seus limites.

RANDOMIC :

Quando o valor é excedido, gera um valor aleatório nos limites estabelecidos e calibra seus BIAS e pesos para que este novo valor de saída aceitável seja realidade. "Deixe que uma pitada de aleatoriedade converja o destino da rede neural a teu favor."

RELATIVE :

Regula a saída da rede neural de tal forma a achatar a onda dentro do seu extremo limite permitido excedido. O achatamento possui uma tolerancia para mais ou para menos dentro desse limite máximo ou mínimo de acordo de onde a saída tocar.

CONSERVATIVE :

Após exceder seus limites para explosao ou vazão do gradiente, automaticamente ajustará seus pesos e BIAS para se manter no menor limite dessa rede neural,

"Criando uma personalidade mais conservadora e segura de avançar somente com muitas "tatadas" de certeza. Mas sempre apelando para a acertividade garantida. Mistura de sagacidade com certeza, tudo depende do orquestramento dos pontos. Pode ter inicios ganhando pontuação aos poucos mas com muita dose de precisão e certeza. Só irá avançar a cada passo que tiver certeza do que está sendo feito."

PING_PONG :

Após exceder os limites do gradiente, contra ataca a parede desse gradiente com mesma energia sinética na direção oposta, e se, ainda assim essa energia for maior na contra direção, vai bater com mesma intensidade, criando efeito de uma bolinha de ping pong. Esta é uma opção um pouco onerosa mas com bons resultados.

INFINITY:

Ao chegar em uma das extremidades permitidas, a energia será considerada como se estivesse na contra ponta dos limites do gradiente, gerando efeito de loop infinito até que a inercia diga o ponto de saída da energia.

CENTRONIUM:

Ao exceder os limites da extremidade, o sistema considera que esta energia partiu do centro para mesma direção, amortecendo o excesso com inercia.

DESCENTRONIUM:

Ao exceder os limites da extremidade, o sistema considera que a energia partiu do centro para a direção oposta amortecendo o excesso com inercia.

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

Habilitação**Sobre:**

Habilita a função de VanishingExplodingPolicyConfigurer com as configurações anteriores programadas, se não houver, usa o padrão.

Sintaxe:

```
rede.HabilitarVanishingExplodingConfigurer();
```

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

Desabilitação**Sobre:**

Desabilita a função de VanishingExplodingPolicyConfigurer em tempo de execução.

Sintaxe:

```
rede.DesabilitarVanishingExplodingConfigurer();
```

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

Configurações específicas

Configurações relacionadas a atributos específicos da rede.

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

noise**Sobre:**

O noise funciona como constante para números que são gerados de forma aleatória. No momento estamos considerando ela apenas para aprendizado por reforço, porém estamos desenvolvendo métodos que permitirão fazer uso dela de forma mais abrangente junto com a geração de semente.

Dependendo da escola, o noise será obrigatório.

Sintaxe:

```
rede.InserirEditarNoise(double noise);
```

Parametros Obrigatórios:`double noise`

Insere o valor que influencia como constante em pesos gerados aleatoriamente.

Created with the Personal Edition of HelpNDoc: [Easily create Web Help sites](#)

gama**Sobre:**

O gama é uma constante utilizada em aprendizado por reforço profundo. Deep Q Learning por exemplo.

No momento, não estamos utilizando o Gama em nossas escolas.

Sintaxe:

```
rede.InserirEditarGama(double gama);
```

Parametros Obrigatórios:`double gama`

Insere constante de aprendizagem por reforço.

Created with the Personal Edition of HelpNDoc: [Free CHM Help documentation generator](#)

Comandos Especial

Created with the Personal Edition of HelpNDoc: [Easily create Web Help sites](#)

Clone**Sobre:**

Clona a rede neural de forma profunda em tempo de execução, gerando uma nova rede rede clonada com um novo id.

Sintaxe:

```
var redeClonada = redeNeural.ClonarRedeNeural();
```

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

Save Token**Sobre:**

Você pode salvar sua rede neural, com todos os dados que ela adquiriu, pesos e bias, configurações, etc.

Este comando gera um token que pode ser inserido posteriormente para resgatar a rede em uma nova instânciação.

Sintaxe:

```
string token = redeNeural.GetToken([string secret = "TurboMind"]);
```

Paramtro opcional:`string secret`

Chave de codificação, deve ser guardada para posteriormente resgatar a rede.

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

Agente

O Agente permite juntar diversas ferramentas como os soquetes de entradas, saídas, metodologia e a própria rede neural e fazer com que tudo isso trabalhe da forma mais harmoniosa a teu favor.



Estamos implementando diversas funcionalidades em nosso agente para entregar um "Robo" com grande poder de processamento neural. Nosso agente é preparado para autoprogramar a rede neural de acordo com a escola que estiver sido inserida em seu corpo.

Ele também fica com o encargo de interpretar a entrada através do soquete de entrada. Gerenciar essas informações para a rede neural. Processar a saída através do soquete de saída e entregar esta saída ao ambiente.

Estamos inserindo aos poucos novas funções.

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

Instanciando

Biblioteca:

```
using TurboMind.MindFactory;
```

MOD0 1 - NOVA INSTANCIA

```
Agente agente = new(string nomeAgente, string propositoAgente)
```

Parametros Obrigatórios:

```
string nomeAgente
```

Nome do novo agente!

`string` `propositoAgente`

Propósito do agente, destino - Salvar o mundo contra os ataques de marcianos em 2027!

MODO 2 - RESGATE

```
RedeNeural redeAnterior = new(string tokenAgent, [string secret],
[bool useNewId = false])
```

Parametros Obrigatórios:

`string` `tokenAgent`

Token do agente, gerado anteriormente, onde contem a criptografia do modelo.

Parametros Opcionais:

`string` `secret`

Insere a chave ou segredo para descriptografar o token, a chave deve ser guardada para ser utilizada posteriormente para descriptografar o agente.

`bool` `useNewId`

Se false, traz o id anterior, se true, gera um novo id para este agente, mantendo todos os dados anteriores, independente se true ou false.

Created with the Personal Edition of HelpNDoc: [Create cross-platform Qt Help files](#)

Inserções

Created with the Personal Edition of HelpNDoc: [Free EBook and documentation generator](#)

Inserir metodologia de aprendizado

Sobre:

Insere a metodologia de aprendizado seja ela de aprendizado por reforço ou supervisionado.

Sintaxe:

```
agente.InserirMetodologiaAprendizado(_Metodologia metodologia);
```

Parametros Obrigatórios:

`_Metodologia` `metodologia`

Metodologia de aprendizado, é importante calibrar os parametros dela antes de inseri-la no agente.

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

Inserir Rede neural

Sobre:

Insere a rede neural no agente.

Sintaxe:

```
agente.InserirRedeNeural(RedeNeural redeNeural);
```


Parametros obrigatórios:

RedeNeural redeNeural

Rede neural, que esteja de preferencia previamente configurada.

Se houver soquete de entrada ou de saída, é necessário que seus terminais estejam com contatos equivalentes, ou gerará excessão.

Created with the Personal Edition of HelpNDoc: [Easily create EBooks](#)

Inserir Soquete de entrada**Sobre:**

Insere Soquete de entrada no agente.

Sintaxe:

```
agente.InserirSoqueteEntrada(_SoqueteEntrada soqueteEntrada);
```

Parametros Obrigatórios:

_SoqueteEntrada soqueteEntrada

Soquete de entrada, requer que o mesmo esteja previamente configurado se existir obrigatoriedade.

Created with the Personal Edition of HelpNDoc: [Easy EBook and documentation generator](#)

Inserir Soquete de saída**Sobre:**

Insere Soquete de saída no agente.

Sintaxe:

```
agente.InserirSoqueteDeSaida(_SoqueteSaida soqueteSaida);
```

Parametros Obrigatórios:

_SoqueteSaida soqueteSaida

Soquete de saída, requer que o mesmo esteja previamente configurado se existir obrigatoriedade.

Created with the Personal Edition of HelpNDoc: [Easily create Web Help sites](#)

Interagindo com ambiente**Sobre:**

Permite que os dados do ambiente externo possa entrar no agente, passando pelo circuito do soquete de entrada (Se houver, bem como pela rede neural).

O Agente processa um pulso de entradas de cada vez por toda a estrutura da rede neural, se houver soquete de saída a informação também será transformada antes de ser devolvida.

Sintaxe:

```
agente.InteragirComAmbiente(double[] entradas);
```

Parametros Obrigatórios:

double[] entradas

Entradas do mundo externo em formado de array com seus dados brutos, sendo cada entrada relacionada a um neurônio.

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

Remoções

Created with the Personal Edition of HelpNDoc: [Free HTML Help documentation generator](#)

Remover metodologia

Sobre:

Remove metodologia de aprendizado se ela existir.

Sintaxe:

```
agente.RemoverMetodologia();
```

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

Remover rede neural

Sobre:

Remove rede neural se ela existir.

Sintaxe:

```
agente.RemoverRedeNeural();
```

Created with the Personal Edition of HelpNDoc: [Free iPhone documentation generator](#)

Remover soquete entrada

Sobre:

Remove soquete de entrada se existir.

Sintaxe:

```
agente.RemoverSoqueteEntrada();
```

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

Remover soquete saída

Sobre:

Remove soquete de saída se existir.

Sintaxe:

```
agente.RemoverSoqueteSaida();
```

Created with the Personal Edition of HelpNDoc: [Full-featured Documentation generator](#)

Clonagens

Created with the Personal Edition of HelpNDoc: [Create help files for the Qt Help Framework](#)

Clonar agente

Sobre:

Clonagem profunda do agente, com todos os seus objetos devidamente configurados.

Sintaxe:

```
Agente agenteClonado = agente.ClonarAgente();
```

Created with the Personal Edition of HelpNDoc: [Full-featured EPub generator](#)

Clonar metodologia

Sobre:

Clonagem profunda da metodologia.

Importante.

Se a metodologia estiver carregada com o adaptador de aprendizagem por reforço, requer colocar a notation [Serializable] na classe que herda a interface adaptadora.

Sintaxe:

```
var metodologiaClonada = agente.ClonarMetodologia();
```

Created with the Personal Edition of HelpNDoc: [What is a Help Authoring tool?](#)

Clonar rede neural

Sobre:

Clonagem profunda da rede neural, com todos os seus objetos devidamente configurados.

Sintaxe:

```
RedeNeural redeClone = agente.ClonarRedeNeural();
```

Created with the Personal Edition of HelpNDoc: [Easily create Qt Help files](#)

Clonar soquete entrada

Sobre:

Clonagem profunda do soquete de entrada, com todos os seus objetos devidamente configurados.

Sintaxe:

```
var soqueteClone = agente.ClonarSoqueteEntrada();
```

Created with the Personal Edition of HelpNDoc: [Easily create Help documents](#)

Clonar soquete saída

Sobre:

Clonagem profunda do soquete de saída, com todos os seus objetos devidamente

configurados.

Sintaxe:

```
var soqueteClone = agente.ClonarSoqueteSaida();
```

Created with the Personal Edition of HelpNDoc: [Free PDF documentation generator](#)

Treinamentos

Created with the Personal Edition of HelpNDoc: [Produce electronic books easily](#)

AutoTreinamento

Sobre:

Permite que o agente auto inicie o treinamento.

Requer ter no mínimo Escola de aprendizado configurada e rede neural configurada.

Sintaxe:

```
agente.AutoTreinamento();
```

Created with the Personal Edition of HelpNDoc: [News and information about help authoring tools and software](#)

Avaliar Performance treinamento

Sobre:

Permite que o agente teste o treinamento da rede neural e devolva relatório do desempenho.

Sintaxe:

```
var dadosTreinamento = agente.AvaliarPerformanceTreinamento(int
qtdEpochs, [bool exibirRelatorioExecucao = false], [double
tolerancia = 0.1]);
```

Parametros Obrigatórios:

`int qtdEpochs`

Informa ao agente por quantas epocas o treinamento será submetido.

Parametros Opcionais:

`bool exibirRelatorioExecucao = false`

Se colocado como true, exibirá em prompt de comando (CMD) o desempenho do treinamento.

`double tolerancia = 0.1`

Margem de tolerancia para considerar que foi um acerto da rede sobre o valor de cada neurônio para mais ou para menos, o padrão colocado é de 0.1 mas pode ser alterado.

RETORNOS:

```
double totalPontosAcumulados ->
dadosTreinamento.totalPontosAcumulados
Somatória de pontos que a rede acertou durante o treino.
```

```
double mediaPonderadaPontios ->
dadosTreinamento.mediaPonderadaPontos
```

Média de pontos feita por ciclo de tentativa e erro.

```
double mediaIndiceAcertividade ->
dadosTreinamento.mediaIndiceAcertividade
```

Porcentagem de acertos da rede neural no decorrer de todo o processo das epochs.
(Varia de 0% a 100%).

Created with the Personal Edition of HelpNDoc: [Full-featured multi-format Help generator](#)

Saves GetToken

Sobre:

Assim como a rede neural, é possível salvar o agente com todos seus dados juntamente com os objetos que o compõe!

Sintaxe:

```
string token = agente.getToken([string secret = "TurboMind"]);
```

Paramtro opcional:

```
string secret
```

Chave de codificação, deve ser guardada para posteriormente resgatar o agente.

Created with the Personal Edition of HelpNDoc: [Easy EPub and documentation editor](#)

Setups

Created with the Personal Edition of HelpNDoc: [Easily create HTML Help documents](#)

DeepLearning

Created with the Personal Edition of HelpNDoc: [Easily create Help documents](#)

Lógica AND

```
/// <summary>
/// Nesse desafio temos um neurônio com duas entradas e
uma saída. E Obedece a lógica AND.
/// </summary>
public void Deep_Learning_LogicaAND() {
    Agente agenteAND = new("AND Agente", "Testar Novo
Modelo De Inteligencia Artificial");
    RedeNeural redeAND = new(0.06, 2, 1);
    redeAND.InserirEditarFuncaoAtivacao(0,
ActiveFunc.Degrau);

    agenteAND.InserirRedeNeural(redeAND);
    DeepLearning deepLearningSchool = new("Escola AND",
"Treina agente");
    LogicaANDSupervisionada AND = new();
    deepLearningSchool.InserirAmbienteTreinamento(AND.Sit
```

```

uacaoProblema(), AND.Gabarito());
        deepLearningSchool.SetupDeTreinamento(50, true, 50);
        deepLearningSchool.ConfigurarFuncaoErro(FuncaoErro.Si
ngleLoss);
        agenteAND.InserirMetodologiaDeAprendizado(deepLearnin
gSchool);

        do agenteAND.AutoTreinamento();
        while (agenteAND.AvaliarPerformanceTreinamento(10,
true).mediaIndiceAcertividade < 100);
    }

```

USE PARA A CLASSE AND:

```

public class LogicaANDSupervisionada {
    public List<double[]> SituacaoProblema() {
        List<double[]> TabelaPadrao = new();
        TabelaPadrao.Add(new double[] { 0, 0 });
        TabelaPadrao.Add(new double[] { 0, 1 });
        TabelaPadrao.Add(new double[] { 1, 0 });
        TabelaPadrao.Add(new double[] { 1, 1 });
        return TabelaPadrao;
    }

    //-----
    public List<double[]> Gabarito() {
        List<double[]> Gabarito = new();
        Gabarito.Add(new double[] { 0 });
        Gabarito.Add(new double[] { 0 });
        Gabarito.Add(new double[] { 0 });
        Gabarito.Add(new double[] { 1 });
        return Gabarito;
    }
}

```

Created with the Personal Edition of HelpNDoc: [Free Kindle producer](#)

Calibrador dinamico

```

public void Deep_Learning_DynamicCalibrator() {
    Agente caliberAgent = new("Caliber Agent", "Dynamic
agent of test of numbers");

    RedeNeural rede = new(0.06, 1, 1, 2,4,2, 1);
    caliberAgent.InserirRedeNeural(rede);
    rede.VanishingExplodingPolicyConfiger();

    OpenCalibrator openCalibrator = new();
    openCalibrator.InjetarVetorProblema(24, 32, 10, 44);
    openCalibrator.InjetarVetorSolucao(0.24, 0.32, 0.1,
0.4);

    SoqueteNormalizacaoEntrada soqueteEntrada =
new("Entrada", "A", 1);

```

```

        soqueteEntrada.ConfigurarEntrada(0, 0, 100, "E1",
"Entrada 1");

        caliberAgent.InserirSoqueteEntrada(soqueteEntrada);

        DeepLearning deepLearning = new("Avaliador IMC",
"Testa a capacidade do agente em avaliar IMC");
        deepLearning.ConfigurarFuncaoErro(FuncaoErro.SingleLoss);
        deepLearning.InserirAmbienteTreinamento(openCalibrato
r.SituacaoProblema(), openCalibrato.Gabarito());
        deepLearning.SetupDeTreinamento(5000, true, 5000);

        caliberAgent.InserirMetodologiaDeAprendizado(deepLear
ning);

        do caliberAgent.AutoTreinamento();
        while
(caliberAgent.AvaliarPerformanceTreinamento(100,
true).mediaIndiceAcertividade < 100);
    }

```

USE NA CLASSE:

```

public class OpenCalibrator {

    List<double[]> RegistroDeSituacao = new();
    List<double[]> Resposta = new();

    //-----
    public void InjetarVetorProblema(params double[] problem)
=>
        problem.ToList().ForEach(_ =>
RegistroDeSituacao.Add(new double[] { _ }));

    //-----
    public void InjetarVetorSolucao(params double[] gabarito)
=>
        gabarito.ToList().ForEach(_ => Resposta.Add(new
double[] { _ }));

    //-----
    public List<double[]> SituacaoProblema() =>
RegistroDeSituacao;

    //-----
    public List<double[]> Gabarito() => Resposta;
}

```

Created with the Personal Edition of HelpNDoc: [Easily create iPhone documentation](#)

Avaliação IMC

```

/// <summary>

```

```

    /// Nesse desafio é entregue dois valores na entrada da
    IA, esta IA possui 2 neurônios de entrada, 10 neuronios na camada
    oculta e
    /// 7 neurônios na camada de saída. Ela basicamente
    precisa de comparar os valores de entrada e setar apenas uma
    saída e com
    /// isso avaliar o IMC de uma pessoa, de acordo com os
    dados entregues de uma tabela.
    /// </summary>
    public void Deep_Learning_Avaliacao_IMC() {
        Agente AgenteOrigem = new("Function", "Testar funções
de ativação");

        RedeNeural rede = new(0.001, 2, 10, 7);
        rede.VanishingExplodingPolicyConfigurer(Comportament.RE
LATIVE, 50 ,1, -1);

        rede.InserirEditarTodasFuncoesAtivacao(ActiveFunc.ELU
, 0.02);

        AgenteOrigem.InserirRedeNeural(rede);

        SoqueteNormalizacaoEntrada soqueteEntrada =
new("Entradas", "A e B", 2, 0, 1);
        soqueteEntrada.ConfigurarEntrada(0, 16, 100, "E1",
"Entrada 1");
        soqueteEntrada.ConfigurarEntrada(1, 16, 100, "E2",
"Entrada 2");
        AgenteOrigem.InserirSoqueteEntrada(soqueteEntrada);

        SaidaPotencial soqueteSaidaPot = new("Soquete",
"Desc", 7);
        AgenteOrigem.InserirSoqueteDeSaida(soqueteSaidaPot);

        AvaliacaoIMC IMC = new AvaliacaoIMC();
        DeepLearning deepLearning = new("Avaliador IMC",
"Testa a capacidade do agente em avaliar IMC");
        deepLearning.ConfigurarFuncaoErro(FuncaoErro.SingleLo
ss);
        deepLearning.InserirAmbienteTreinamento(IMC.SituacaoP
roblema(), IMC.Gabarito());
        deepLearning.SetupDeTreinamento(1000000, true,
1000000);

        AgenteOrigem.InserirMetodologiaDeAprendizado(deepLear
ning);

        do AgenteOrigem.AutoTreinamento();
        while
        (AgenteOrigem.AvaliarPerformanceTreinamento(100,
true).mediaIndiceAcertividade < 100);
    }

```


CLASSE IMC:

```

internal class AvaliacaoIMC {
    public List<double[]> SituacaoProblema() {
        List<double[]> RegistroDeSituacao = new();

        RegistroDeSituacao.Add(new double[] { 16,
16.9 }); //Queda de cabelo, infertilidade, ausência menstrual
        RegistroDeSituacao.Add(new double[] { 17,
18.4 }); //Fadiga, stress, ansiedade
        RegistroDeSituacao.Add(new double[] { 18.5,
24.9 }); //Menor risco de doenças cardíacas e vasculares
        RegistroDeSituacao.Add(new double[] { 25,
29.9 }); //Fadiga, má circulação, varizes
        RegistroDeSituacao.Add(new double[] { 30,
34.9 }); //Diabetes, angina, infarto, aterosclerose
        RegistroDeSituacao.Add(new double[] { 35,
39.9 }); //Apneia do sono, falta de ar
        RegistroDeSituacao.Add(new double[] { 40,
100 }); //Refluxo, dificuldade para se mover, escaras, diabetes,
        infarto, AVC

        return RegistroDeSituacao;
    }

    //-----
    public List<double[]> Gabarito() {
        List<double[]> Gabarito = new();
        Gabarito.Add(new double[] { 1, 0, 0, 0, 0, 0,
0 }); //Queda de cabelo, infertilidade, ausência menstrual
        Gabarito.Add(new double[] { 0, 1, 0, 0, 0, 0,
0 }); //Fadiga, stress, ansiedade
        Gabarito.Add(new double[] { 0, 0, 1, 0, 0, 0,
0 }); //Menor risco de doenças cardíacas e vasculares
        Gabarito.Add(new double[] { 0, 0, 0, 1, 0, 0,
0 }); //Fadiga, má circulação, varizes
        Gabarito.Add(new double[] { 0, 0, 0, 0, 1, 0,
0 }); //Diabetes, angina, infarto, aterosclerose
        Gabarito.Add(new double[] { 0, 0, 0, 0, 0, 1,
0 }); //Apneia do sono, falta de ar
        Gabarito.Add(new double[] { 0, 0, 0, 0, 0, 0,
1 }); //Refluxo, dificuldade para se mover, escaras, diabetes,
        infarto, AVC

        return Gabarito;
    }
}

```

Created with the Personal Edition of HelpNDoc: [Free PDF documentation generator](#)

ARS

Created with the Personal Edition of HelpNDoc: [Free Qt Help documentation generator](#)

Inversor Logico

```

        public void
ARS_Logica_Inversora_Um_Neuronio_Uma_Entrada() {
    Agente agentInversorARS = new("Inversor", "Lógica
Inversa");

    RedeNeural redeInversora = new(0.065, 1, 1);
    redeInversora.InserirEditarNoise(0.05);
    redeInversora.InserirEditarTodasFuncoesAtivacao(Activ
eFunc.Degrau);
    agentInversorARS.InserirRedeNeural(redeInversora);

    SoquetePadronizacaoEntrada padrao = new("Entrada",
"teste", 1);
    padrao.PersonalizarEntrada(0, "teste", "desc");
    agentInversorARS.InserirSoqueteEntrada(padrao);

    ARS ARSInversorSchool = new("Inversor School",
"Escola de lógica Inversora");
    ARSInversorSchool.SetupDeTreinamento(10,10,10,100);

    ARSInversorSchool.InserirAmbienteTreinamento(new
LogicaInversora());

    agentInversorARS.InserirMetodologiaDeAprendizado(ARSI
nversorSchool);

    do agentInversorARS.AutoTreinamento();
    while
(agentInversorARS.AvaliarPerformanceTreinamento(10,
true).mediaIndiceAcertividade < 100);
}

```

CLASSE:

```

        internal class LogicaInversora:
IModuloAprendizadoReforcoAdapter {
    double i = 0;
    double ValorEntrada;
    double pontos;
    double[] Teste = new double[1];

    //-----
    public (double[], double, bool) Action(double[]
EstimuloNeural) {
        if (ValorEntrada == 1 && EstimuloNeural[0] == 0) {
            pontos++;
        } else if (ValorEntrada == 0 && EstimuloNeural[0] ==
1) {
            pontos++;
        } else { pontos--; }
        ValorEntrada = new Random().Next(0, 2);
    }
}

```

```

        Teste[0] = ValorEntrada;
        return (Teste, pontos, false);
    }

    //-----
    public double[] Reset() {
        ValorEntrada = new Random().Next(0, 2);
        pontos = 0;
        return new double[] { ValorEntrada };
    }
}

```

Created with the Personal Edition of HelpNDoc: [Easy EBook and documentation generator](#)

Logica AND

```

    public void ARS_Logica_AND_Um_Neuronio_Duas_Entradas() {
        Agente agentARSLogic = new("Agente Logico And",
        "Exemplo de teste de porta AND");

        RedeNeural rede = new(0.005, 2, 1);
        rede.InserirEditarNoise(0.04);
        rede.InserirEditarTodasFuncoesAtivacao(ActiveFunc.Deg
rau);

        agentARSLogic.InserirRedeNeural(rede);

        SoquetePadronizacaoEntrada padroni = new("Teste",
        "Soquete de Teste", 2);
        padroni.PersonalizarTodasEntradas("Teste", "Teste");
        agentARSLogic.InserirSoqueteEntrada(padroni);

        ARS ARSANDSchool = new("Logic And", "Test Logica
And");

        ARSANDSchool.InserirAmbienteTreinamento(new
LogicaAnd());
        ARSANDSchool.SetupDeTreinamento(8, 8, 100, 5000);
        agentARSLogic.InserirMetodologiaDeAprendizado(ARSANDS
chool);

        agentARSLogic.AutoTreinamento();
        agentARSLogic.AvaliarPerformanceTreinamento(10,
true);
    }

    CLASE:
    internal class LogicaAnd : IModuloAprendizadoReforcoAdapter {
        int StackMemoryA;
        int StackMemoryB;

        //-----
        public (double[] estimuloAmbiente, double pontos, bool
fim) Action(double[] EstimuloNeural) {

            double pontos = 0;

```

```

        if (StackMemoryA == 0 && StackMemoryB == 0 &&
EstimuloNeural[0] == 0) {
            pontos = 1;
        } else if (StackMemoryA == 0 && StackMemoryB == 1 &&
EstimuloNeural[0] == 0) {
            pontos = 1;
        } else if (StackMemoryA == 1 && StackMemoryB == 0 &&
EstimuloNeural[0] == 0) {
            pontos = 1;
        } else if (StackMemoryA == 1 && StackMemoryB == 1 &&
EstimuloNeural[0] == 1) {
            pontos = 1;
        } else { pontos = -1; }

        StackMemoryA = new Random().Next(0, 2);
        StackMemoryB = new Random().Next(0, 2);
        return (new double[] { StackMemoryA, StackMemoryB },
pontos, false);
    }

    //-----
    public double[] Reset() {
        StackMemoryA = new Random().Next(0, 2);
        StackMemoryB = new Random().Next(0, 2);
        return new double[] { StackMemoryA, StackMemoryB };
    }
}

```

Created with the Personal Edition of HelpNDoc: [Write eBooks for the Kindle](#)

Chemical Dependency

```

    public void ARS_Chemical_Dependency() {
        Agente agentChemicalARS = new("Chemical Dependent",
"Dependente Químico");

        RedeNeural rede = new(0.03, 3, 3);
        rede.InserirEditarNoise(0.05);
        rede.ConfigurarBIAS(BiasOperation.JAILBREAK);
        rede.VanishingExplodingPolicyConfiger();
        rede.InserirEditarTodasFuncoesAtivacao(ActiveFunc.Deg
rau);

        agentChemicalARS.InserirRedeNeural(rede);

        SoquetePadronizacaoEntrada visao = new("Visao",
"Noção periférica", 3);
        agentChemicalARS.InserirSoqueteEntrada(visao);

        ARS ARSChemicalSchool = new("ChemicalShcool", "Test
Chemical Agent");
        ARSChemicalSchool.SetupDeTreinamento(10, 10, 200,
1000);
        ARSChemicalSchool.InserirAmbienteTreinamento(new
ChemicalDependency());
    }

```

```

        agentChemicalARS.InserirMetodologiaDeAprendizado(ARSC
hemicalSchool);

        var clonador = agentChemicalARS.ClonarMetodologia();

        do agentChemicalARS.AutoTreinamento();
        while
(agentChemicalARS.AvaliarPerformanceTreinamento(200,
true).mediaIndiceAcertividade < 100);
    }

```

CLASSE:

```

    [Serializable]
    internal class ChemicalDependency :
IModuloAprendizadoReforcoAdapter {
        //Dados da Gertruza
        -----

        int abstinencia;
        double Grana;
        double pontosInterno;

        //Dados do Cenario
        -----

        int Traficante;
        int Civil;

        //-----
        public double[] Reset() {
            ResetarGertruza();
            var RandTrafico = TraficoRandomico();
            return new double[] { RandTrafico.Item1,
RandTrafico.Item2, 0 };
        }

        //-----
        public void ResetarGertruza() {
            abstinencia = 0;
            Grana = 0;
            pontosInterno = 0;
        }

        //-----
        public (double[], double, bool) Action(double[]
EstimuloNeural) {
            bool Traficante = EstimuloNeural[0] == 1 ? true :
false;

            bool Civil = EstimuloNeural[1] == 1 ? true : false;
            bool PedeGrana = EstimuloNeural[2] == 1 ? true :
false;

            abstinencia++;
            double pontos = 0;

            if (Traficante == false && Civil == false &&

```

```

PedeGrana == false) {
    pontos = 0.1;
    pontosInterno += pontos;
} else if (Traficante == false && Civil == false &&
PedeGrana == true) {
    pontos = -1;
    pontosInterno += pontos;
} else if (Traficante == false && Civil == true &&
PedeGrana == false) {
    pontos = 0.1;
    pontosInterno += pontos;
} else if (Traficante == false && Civil == true &&
PedeGrana == true) {
    pontos = 0.2;
    pontosInterno += pontos;
    double GranaLocal = new Random().NextDouble() *
10;

    Grana += GranaLocal;
} else if (Traficante == true && Civil == false &&
PedeGrana == false) {
    pontos = -1;
    pontosInterno += pontos;
} else if (Traficante == true && Civil == false &&
PedeGrana == true && Grana > 0) {
    pontos = 1;
    pontosInterno += pontos;
    abstinencia = 0;
    Grana--;
} else if (Traficante == true && Civil == false &&
PedeGrana == true && Grana <= 0) {
    pontos = 0.2;
    pontosInterno += pontos;
    abstinencia--;
} else if (Traficante == true && Civil == true &&
PedeGrana == false) {
    pontos = 0.1;
    pontosInterno += pontos;
    abstinencia--;
} else if (Traficante == true && Civil == true &&
PedeGrana == true) {
    pontos = -1;
    abstinencia--;
    pontosInterno += pontos;
} else {
    pontos = -0.4;
    pontosInterno += pontos;
    abstinencia--;
}

}

TraficoRandomico();

int Trafic = new Random().Next(0, 2);
int Civ = new Random().Next(0, 2);
if (abstinencia >= 10) {

```

```

        pontos = -1;
        pontosInterno += pontos;
    }

    if (Grana < 0) Grana = 0;
    return (new double[] { Trafic, Civ, Grana }, pontos,
pontosInterno >= 100 || pontosInterno <= -10 ? true : false);
}

//-----
public (double, double) TraficoRandomico() {
    Traficante = new Random().Next(0, 2);
    Civil = new Random().Next(0, 2);
    return (Traficante, Civil);
}
}

```

Created with the Personal Edition of HelpNDoc: [Full-featured Help generator](#)

Hydra

Created with the Personal Edition of HelpNDoc: [Free EPub producer](#)

Contra Pressão

```

    public void
Hydra_Equilibrador_de_Pressao_Um_Neuronio_Uma_Entrada() {
    Agente agentPression = new("PressionAgent", "IA
apta");

    RedeNeural rede = new(0.005, 1, 1);
    rede.InserirEditarNoise(0.04);
    rede.ConfigurarBIAS(BiasOperation.JAILBREAK);
    rede.VanishingExplodingPolicyConfigurer(Comportament.AU
DACIOUS);
    agentPression.InserirRedeNeural(rede);

    Hydra pressionHydraScholl = new("Preassure Scholl",
"Preassure example.");
    pressionHydraScholl.InserirAmbienteTreinamento(new
EquilibradorDePressao());
    pressionHydraScholl.SetupDeTreinamento(20, 200,
1000);

    agentPression.InserirMetodologiaDeAprendizado(pressio
nHydraScholl);

    do agentPression.AutoTreinamento();
    while
(agentPression.AvaliarPerformanceTreinamento(200,
true).mediaIndiceAcertividade < 100);
}

```

CLASSE:

```

        internal class EquilibradorDePressao :
IModuloAprendizadoReforcoAdapter {
    double pressao = 0;

    //-----
    public (double[], double, bool) Action(double[]
EstimuloNeural) {
        double pontos = 0;
        bool termino = false;

        double ContraPressao = EstimuloNeural[0] * 100;
        if (ContraPressao < 0 || ContraPressao > 100) {
            pontos = -1;
        } else if
            ((Convert.ToDouble(pressao) - ContraPressao) <=
0.25 && (Convert.ToDouble(pressao) - ContraPressao) >= -0.25) {
            pontos = 0.5;
        } else {
            pontos = -0.5;
        }
        pressao = new Random().Next(0, 101);
        return (new double[] { pressao }, pontos, termino);
    }

    //-----
    public double[] Reset() {
        pressao = new Random().Next(0, 101);
        return new double[] { pressao };
    }
}

```