



COORDENADORIA DE ENGENHARIA DA COMPUTAÇÃO

GUSTAVO ARAUJO BORGES

LUCAS PAGEL DE FARIA

LUIZ ANTONIO BUFFOLO

**APLICAÇÃO DE APRENDIZAGEM POR REFORÇO EM CARROS
AUTÔNOMOS ATRAVÉS DE TOPOLOGIAS AUMENTANTES**

Sorocaba/SP

2022

Gustavo Araujo Borges

Lucas Pagel de Faria

Luiz Antonio Buffolo

APLICAÇÃO DE APRENDIZAGEM POR REFORÇO EM CARROS AUTÔNOMOS ATRAVÉS DE TOPOLOGIAS AUMENTANTES

Trabalho de conclusão de curso apresentado
ao Centro Universitário Facens como exigência
parcial para obtenção do diploma de graduação
em Engenharia da Computação.
Orientador: Prof. Johannes von Lochter

Sorocaba/SP

2022

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| Figura 1 – Gráfico de evolução de investimentos em robôs industriais. | 10 |
| Figura 2 – Modelos diversos de AGVs adaptados para suas aplicações. | 11 |
| Figura 3 – NavLab 1. | 14 |
| Figura 4 – Tesla Model S. | 15 |
| Figura 5 – Diagrama de Euler da relação entre AI, ML e DL. | 19 |
| Figura 6 – Tabela com visualizações de diferentes métodos de aprendizado de máquina. | 22 |
| Figura 7 – Gene, Cromossomo e População. | 24 |
| Figura 8 – Crossover entre dois cromossomos. | 26 |
| Figura 9 – Interface gráfica utilizada para a simulação, com o modelo de carro autônomo, trajeto de treino a ser percorrido e botões de controle de <i>layout</i> | 30 |
| Figura 10 – Canto inferior esquerdo definido como meta a ser alcançada para obtenção de resultado. | 31 |
| Figura 11 – Organização da rede, com seus nós e conexões. | 32 |
| Figura 12 – Rede neural da Iteração 3 de treinos do algoritmo NEAT, com uma conexão excluída e um nó adicional na camada oculta. | 33 |
| Figura 13 – Rede neural da Iteração 2 de treinos do algoritmo NEAT, com duas conexões excluídas e uma desativada. | 34 |
| Figura 14 – Rede neural da Iteração 1 de treinos do algoritmo de Neuroevolução. | 35 |
| Figura 15 – Rede neural da Iteração 2 após 10 gerações de treinos do algoritmo de Neuroevolução. | 36 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Resultados obtidos pelo NEAT até a meta de Fitness ser alcançada. | 33 |
| Tabela 2 – Resultados obtidos pelo NEAT após a execução de 10 gerações. . . | 34 |
| Tabela 3 – Resultados obtidos em suas respectivas gerações após execução do algoritmo de Neuroevolução. | 35 |
| Tabela 4 – Resultados obtidos até a décima geração após execução do algo- ritmo de Neuroevolução. | 36 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|---|
| AI | Artificial Intelligence |
| AGV | Automated Guided Vehicle |
| AMR | Autonomous Mobile Robot |
| DL | Deep Learning |
| GPS | Global Position System |
| IA | Inteligência Artificial |
| ML | Machile Learning |
| NEAT | Neuroevolution of Augmentating Topologies |
| RL | Reinforcement Learning |

SUMÁRIO

| | |
|---|-----------|
| 1 INTRODUÇÃO | 6 |
| 2 VEÍCULOS AUTÔNOMOS | 9 |
| 2.1 VEÍCULO GUIADO AUTOMATICAMENTE | 9 |
| 2.1.1 Tipos de AGVs | 10 |
| 2.1.2 Métodos de navegação de AGVs | 11 |
| 2.2 CARROS AUTÔNOMOS | 13 |
| 2.2.1 História | 13 |
| 2.2.2 Veículos autônomos atuais | 14 |
| 2.2.2.1 Tesla | 15 |
| 2.2.2.2 Visão Computacional | 16 |
| 3 INTELIGÊNCIA ARTIFICIAL | 17 |
| 3.1 ESTRUTURANDO UM PROBLEMA | 17 |
| 3.2 UTILIZANDO INTELIGÊNCIAS ARTIFICIAIS | 18 |
| 3.3 APRENDIZADO DE MÁQUINA | 19 |
| 3.4 APRENDIZADO POR REFORÇO | 21 |
| 4 TECNOLOGIAS | 23 |
| 4.1 ALGORÍTMO GENÉTICO | 23 |
| 4.1.1 Indivíduo e população | 24 |
| 4.1.2 Reprodução | 24 |
| 4.1.2.1 Seleção | 25 |
| 4.1.2.2 Crossover | 25 |
| 4.1.3 Mutação | 26 |
| 4.2 REDES NEURAIS | 26 |
| 4.3 NEUROEVOLUÇÃO DE TOPOLOGIAS AUMENTANTES | 27 |
| 5 DESENVOLVIMENTO | 29 |
| 5.1 METODOLOGIA | 29 |
| 5.2 SIMULAÇÃO | 29 |
| 6 RESULTADOS | 31 |
| 6.1 NEAT | 32 |
| 6.2 NEUROEVOLUÇÃO | 34 |
| 7 CONSIDERAÇÕES FINAIS | 37 |
| REFERÊNCIAS | 38 |

1 INTRODUÇÃO

De forma geral, esta monografia visa apresentar a implementação de um ambiente simulado capaz de treinar um veículo autônomo que, através dos casos de teste, consegue se adaptar ao ambiente em que se situa buscando chegar a determinado objetivo, através de um algoritmo de neuroevolução que possui uma topologia aumentante. Desta forma, simulações de possíveis trajetos e adaptações de percurso podem ser rapidamente criadas e testadas sem a necessidade de interação com um ambiente físico.

Tratando-se de veículos autônomos, é possível citar os carros sem motorista, segundo [Arruda \(2015\)](#), são veículos capazes de realizar tarefas [...] sem a intervenção de operadores humanos, sua demanda vem crescendo ultimamente por conta da acessibilidade que fornecem e por permitirem uma produtividade maior ao motorista. Considerados também veículos autônomos, existem aqueles voltados para sistemas logísticos, onde, através de mínima interação com operadores, salvo manutenções e reprogramações, são capazes de realizar transporte de objetos por uma linha de produção.

A solução geralmente aplicada para carros autônomos se fundamenta no princípio da visão computacional. Esta técnica se baseia em utilizar de câmeras para a captura de imagens que virão a servir de entrada de dados, para que o algoritmo de direção tome as devidas decisões quanto à melhor decisão a ser tomada para aquela situação.

O problema desta técnica se encontra justamente na lentidão de colocar os sistemas em prática, pois é necessário que se criem diversas situações de teste com muitas variáveis aplicáveis, como iluminação ou resolução da foto, e considerando que a aplicação é direcionada a carros que navegarão pelas ruas, é necessário que estes testes sejam prolongados ainda mais para que se haja uma garantia de segurança a todos ao redor do veículo.

Ao observar veículos autônomos dentro do ambiente fabril, é possível encontrar máquinas chamadas de Veículos Automaticamente Guiados (AGVs), seu uso para o transporte de contêineres data de 1993 em um terminal de contêineres de Rotterdam ([SAPUTRA; RIJANTO, 2021](#)). Estes possuem a finalidade de transportar materiais e produtos pela fábrica, através de sensores, são capazes de identificar a distância necessária para a movimentação do garfo quando aplicados a uma empilhadeira ou a distância de segurança para evitar acidentes. A sua trajetória desses veículos é geralmente predefinida, baseando-se na planta da fábrica e o caminho que o veículo deve percorrer, como através de faixas magnéticas, por exemplo. Sendo assim, estes dispositivos falham quando encontram obstáculos em seu caminho, como um pallet ou

caixa que manterão sua movimentação impedida.

Tratando-se de aprendizado de máquina, existem diversas técnicas que podem ser empregadas para se chegar a resultados simulados, estas, focadas na modelagem da estrutura de aprendizado podem ser divididas entre, supervisionado, não supervisionado e por reforço. A aplicação de cada método deve ser indicada de acordo com a necessidade da aplicação, e, analisando a proposta do projeto presente nesta monografia, é possível analisar qual solução possui a aplicação mais justificável. Em uma observação imediata, o aprendizado por máquina supervisionado (aquele em que se sabe quais são as possíveis entradas e saídas equivalentes a elas) se torna inviável, tal que a aplicação exige que o algoritmo calcule em tempo real a intermissão de possíveis objetos em seu trajeto, sendo assim impossível definir uma saída previamente.

Este projeto procura explorar um tipo diferente de aprendizado, chamado de aprendizado por reforço. Esse tipo de aprendizado tem a característica de não necessitar de supervisão para o treinamento do modelo. Ou seja, não se faz necessário treinamento em campo ou uma requisição de grande quantidade de dados para a etapa de treinamento. Como entradas para o modelo serão utilizados sensores de distância, em vez dos habituais sistemas de visão computacional. O experimento será feito dentro de um ambiente virtual, com a representação do carro e trajetos gerados aleatoriamente para treinamento.

O método de aprendizado por reforço faz necessário menos investimentos antecipados, pois os dados de treinamento serão obtidos por um quociente de performance ao invés de dados rotulados para retroalimentação. Além disso, a utilização de sensores diminui significativamente o requerimento de poder computacional para o sistema comparado com processamento de imagens, e o mesmo pode ser dito quanto ao algoritmo de aprendizado na etapa de testes.

O aprendizado se dará pelo algoritmo NEAT (Neuroevolution of Augmenting Topologies). Este algoritmo se inicia com uma rede neural com apenas as entradas e saídas definidas. Por meio de um Algoritmo Genético, o algoritmo evoluirá a topologia da rede neural de forma a melhorar o desempenho do carro (neste caso, conseguir dirigir por mais tempo sem colisões).

Por exemplo, a princípio o algoritmo gera diversos carros com topologias aleatórias da rede neural. Os carros que percorrerem a maior distância sem colisões serão selecionados para reprodução - seus genes serão combinados com outros carros de performance alta, além de também ocorrerem mutações para o descobrimento de novos comportamentos. Este processo criará uma nova geração de carros, que passará pelos mesmos testes, e o processo se repetirá.

Deste modo, espera-se atingir um estado de otimização em que se faz possível

o alcance da trajetória especificada sem colisões, e preferivelmente em um caminho próximo do mais otimizado.

Apesar da elevada facilidade introduzida com esse método, a perda de flexibilidade na utilização pode ser algo que geraria necessidade de um sistema mais robusto como o de visão computacional. Um outro contraponto seria a necessidade de um ambiente mais controlado, para certificar um funcionamento correto dos sensores no ambiente e nos possíveis obstáculos. Mesmo com tais observações, existe margem para crer que o método proposto se faz uma alternativa interessante em aplicações em que as condições de funcionamento podem ser mapeadas previamente.

2 VEÍCULOS AUTÔNOMOS

Veículos autônomos podem ser definidos como transportes terrestres capazes de se movimentarem de um ponto a outro por conta própria de maneira a não necessitarem de um condutor para tomada de decisões. Esses podem ser divididos em duas categorias: internos e externos, com os internos focando em aplicações industriais e os externos para ambientes urbanos e consumidores finais.

Segundo [SAE \(2018\)](#) Existem níveis de automação veicular divididos em 5 categorias, que possuem o intuito de facilitar o entendimento da função do veículo no ambiente para qual foi desenvolvido. Esses níveis são:

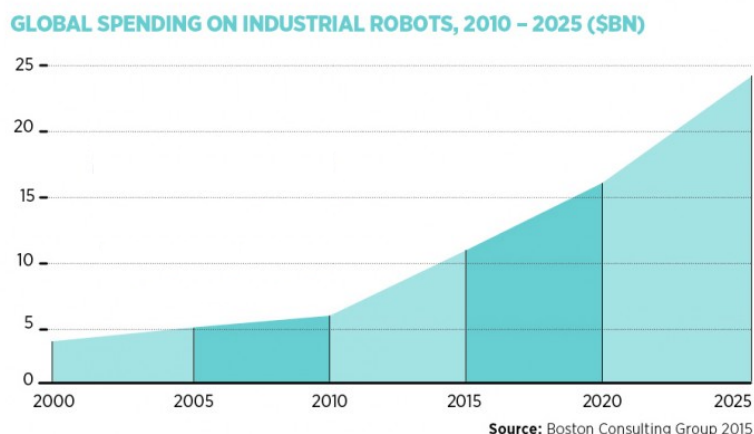
0. Não automação: Autonomia zero; O motorista realiza todas as tarefas;
1. Direção Assistida: O veículo é controlado pelo motorista mas algumas assistências à condução podem ser incluídas no veículo;
2. Automação Parcial: O veículo possui algumas funções automatizadas de controle de maneira combinada, como aceleração e direção, mas o motorista precisa permanecer preparado para tomar o controle e monitorar o ambiente o tempo todo;
3. Automação Condicional: O veículo é capaz de realizar as tarefas de direção sozinho em certas circunstancias, porém o motorista precisa permanecer preparado o tempo todo;
4. Automação Elevada: O veículo é capaz de realizar as tarefas de direção sozinho em certas circunstancias. O motorista pode ter a opção de controlar o veículo;
5. Automação Completa: O veículo é capaz de realizar as tarefas de direção sozinho sob qualquer circunstancia. O motorista pode ter a opção de controlar o veículo.

2.1 VEÍCULO GUIADO AUTOMATICAMENTE

Com o avanço tecnológico estando presente em todas as frentes da economia global atual, não há outra alternativa para fábricas dos mais diversos setores a não ser se manter atualizadas a estas tendências, considerando que se desejam manter relevantes no mercado. O fenômeno que mais vem se alastrando neste ambiente no passado recente é o da Indústria 4.0, o qual define o uso de diversas tecnologias na área de inteligência artificial, computação em nuvem, big data e internet das coisas a fim de se automatizar processos que viriam a demandar recursos da empresa,

e conseqüentemente, afetando sua produtividade. A Figura 1 ilustra o avanço dos investimentos na área ao longo dos últimos anos.

Figura 1 – Gráfico de evolução de investimentos em robôs industriais.



¹Disponível em: <https://www.raconteur.net>. Acesso em: 3 Mai. 2022.

A automação de máquinas visa permitir que os operários da produção sejam capazes de as programar e deixar a executar funções sozinhas, com ou sem uma supervisão externa, mas sempre sem a necessidade de um operador humano, o avanço da quarta revolução industrial vem permitindo que estas máquinas executem suas funções de forma mais rápida e precisa com o passar do tempo. Uma função básica e de extrema importância para o ambiente industrial, é a de manuseio e movimentação de materiais pela planta da fábrica, principalmente no transporte de carga.

Os veículos guiados automaticamente, conhecidos como AGVs, são veículos móveis e programáveis que cumprem a função de transporte de carga dentro de um espaço de linha de produção. Para serem de fato automatizados, devem possuir algum meio de seleção de trajetória da qual não necessitem de intervenção manual, assim sendo capazes de cumprir suas funções com um baixo custo operacional (DAS; PASAN, 2016). Estes devem ser adaptados ao ambiente em que serão aplicados, como a inclusão de sensores necessários ou capacidade para carga a ser carregada.

2.1.1 Tipos de AGVs

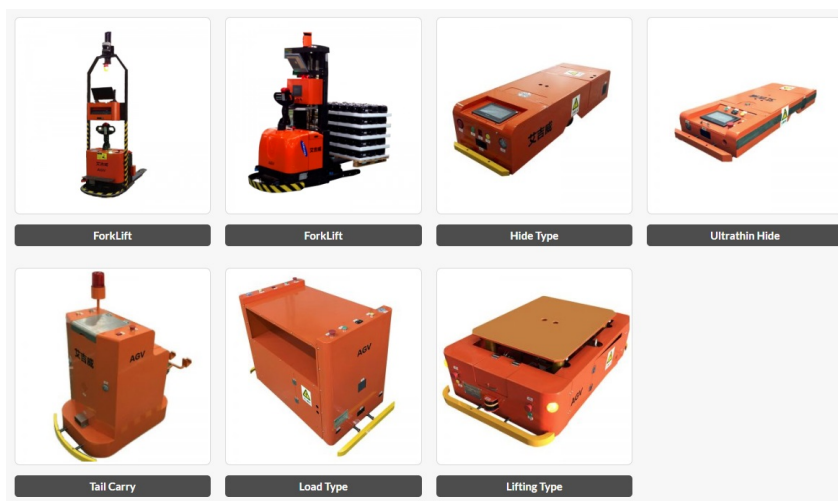
Antes de aplicar um veículo guiado automaticamente na linha de produção, se faz necessário entender qual o problema que ele visa solucionar, existe uma série de veículos desde os de uso geral até mais específicos, dentre os mais popularmente conhecidos é possível citar:

¹ Imagem retirada de: <https://www.raconteur.net/industry-4-0-smart-machines-are-new-industrial-revolution/>

- *Under Ride*, também chamados de *Automated Guided Carts*, são carros que carregam plataformas móveis e se encaixam por baixo da carga, sendo geralmente um dos modelos mais baratos e fáceis de se aplicar a linha de montagem;
- *Tugger AVG*, traduzidos como AGV Rebocador, são os veículos que acoplam suas cargas como um comboio e tem como principal vantagem a quantidade de peso elevada que conseguem puxar;
- *Pallet Mover AVG*, ou AGV Empilhadeira é aquele que, assim como empilhadeiras convencionais, conseguem movimentar *pallets* e bobinas das linhas de produção até armazéns, podendo levantar sua carga de acordo com a altura máxima do equipamento;
- *Unit Load AVG*, traduzidos como AVG com Mesa de Transferência, são capazes de transferir uma carga que carregam sobre eles para outras plataformas através de um sistema de esteira rolantes que possuem.

Além dos citados, existem outros tipos de modelos empregados na indústria que podem vir a ser utilizados por empresas em situações específicas de sua necessidade, adaptando os AGVs de acordo com sua linha de produção. É possível verificar alguns exemplos na Figura 2.

Figura 2 – Modelos diversos de AGVs adaptados para suas aplicações.



²Disponível em: <https://www.agvnetwork.com>. Acesso em: 3 Mai. 2022.

2.1.2 Métodos de navegação de AGVs

Existem diversas técnicas aplicáveis a fim de se obter uma navegação automatizada dos AVGs, metodologias mais antigas se baseiam em sensoramento indutivo,

²Imagem retirada de: <https://www.agvnetwork.com/automated-guided-vehicles-manufacturers>

ótico ou magnético e soluções mais modernas empregam o uso de visão computacional e GPS, a melhor solução é encontrada justamente na mais bem aplicável ao problema proposto.

- Navegação de caminho físico exige que, de certa forma, o AVG seja capaz de seguir um trajeto definido na superfície de movimento e caminhe por ele até ser necessário tomar alguma ação baseando-se em seu algoritmo, mesmo sem saber onde exatamente ele se localiza geograficamente. Esta movimentação pode ocorrer graças a diferentes tipos de sensoriamento:
 - Ótico: navegação baseada em sensores que identificam uma fita que contraste em cor com o restante do chão da fábrica, permitindo que o AVG se movimente enquanto a guia estiver presente e a seguindo, ajustando sua rota quando necessário, esta checagem de trajeto também pode ser feita através de códigos QR presentes no chão;
 - Indutivo: navegação baseada em sensores indutivos instalados no AVG que se mantém em movimento enquanto estiver no alcance de fitas magnéticas sobre ou sob o chão da fábrica, ajustando seu trajeto por onde a fita o guiar.
- Navegação de caminho virtual é onde o mapa do trajeto aplicável ao AGV está presente dentro de sua memória ou em um computador central (transmitido pela rede), de modo que a navegação se torne mais facilmente adaptável, em contrapartida, se faz necessário saber a posição do veículo para que o mesmo possa interpretar o melhor trajeto a se percorrer. Esta movimentação também possui algumas variações de navegação sujeitas a implementação:
 - Laser: raios são emitidos partindo do AGV e com base na resposta de refletores montados ao seu redor na fábrica, ele consegue identificar suas coordenadas e se basear nelas para encontrar sua rota. Existem exemplos de sensores que conseguem captar respostas do ambiente sem a necessidade de refletores instalados;
 - GPS: estes AGVs possuem receptores GPS que captam o sinal de satélites e assim se identificam no plano, este sendo um método custoso por exigir uma série de sinais externos, geralmente possui sua lógica empregada com radares locais;
 - Guiada por visão: através de uma câmera, o veículo consegue interpretar os objetos a sua frente como uma projeção 2D e se localizar no espaço.

Estes veículos que não necessitam de guias externos também são chamados de AMR, que diferentemente dos AGVs convencionais, possuem inteligência própria

e geralmente se encontram atrelados a contextos de Indústria 4.0 e Internet das Coisas. Novamente, a melhor solução para cada problema depende de sua aplicação, a navegação por caminho física é mais fácil de ser implementada e já pode ser encontrada na indústria a muito tempo enquanto a por caminho virtual exige uma complexidade de mapeamento prévio mas a torna mais versátil.

2.2 CARROS AUTÔNOMOS

Veículos de condução autônoma externa são caracterizados pela capacidade de se movimentar em ambientes não controlados, no contexto urbano, sem a necessidade e interferência de um condutor de maneira direta ([LDSV - USP, 2016](#)). A tecnologia embarcada nesses veículos possui mais de trinta anos, porém, somente no começo do século XXI começou a ser difundida entre o grande público e se tornar mais popular.

2.2.1 História

Sabe-se que estudos envolvendo um veículo terrestre capaz de se locomover sem a necessidade de um motorista é datado de 1920, onde engenheiros e pesquisadores de Ohio, nos Estados Unidos, desenvolveram um veículo de pequeno porte, incapaz de transportar passageiros, que possuía capacidade de receber instruções via rádio e realizar comandos de atuação nos sistemas de direção do veículo, eliminando a necessidade de um condutor. A limitação desse sistema estava no *hardware*, incapaz de tomar decisões e necessitando de instruções externas para guiá-lo. Apesar de simples, esse foi o início da busca de um veículo capaz de se autoconduzir.

Em 1950, sensores capazes de detectar velocidade e localização dos veículos foram utilizados em estradas de teste para captar e enviar informações via rádio para os carros equipados com receptores, sendo o precursor o modelo GM Firebird II em 1956. Porém, tal ideia foi descontinuada pela complexidade de instalação dos sensores nas estradas, assim como o gerenciamento de um tráfego constante e composto por veículos não equipados com a tecnologia. Mesmo com o insucesso da solução, já havia avanço quando comparado com o modelo via rádio, com o sistema possuindo um *software* capaz de tomar decisões como acelerar e frear sem interferência externa.

Com o avanço da computação e do processamento de dados, foi possível desenvolver um veículo equipado com diferentes tipos de sensores, processadores e câmeras capazes de identificar, por exemplo, um carro à frente e evitar colisões descartando informações externas, sendo a decisão tomada pelo próprio sistema presente no veículo. Com isso, o primeiro carro realmente autônomo da história foi o NavLab 1, lançado em 1986 ([ESTADÃO, 2020](#)), demonstrado na Figura 3.

Figura 3 – NavLab 1.



³Disponível em: <https://www.rediscoverthe80s.com>. Acesso em: 3 Mai. 2022.

2.2.2 Veículos autônomos atuais

Mesmo com os conceitos de funcionamento de um veículo autônomo sendo definido em 1980 e o poder computacional dos dispositivos avançarem ano após ano, tal tecnologia permaneceu distante do consumidor final por um longo período de tempo dada sua complexidade de navegação em situações urbanas do cotidiano.

Segundo [Anthony Townsend \(2020\)](#), somente em 2009 empresas, como o Google, iniciaram pesquisas com o intuito de atingir um veículo autônomo urbano. Empresas líderes de mercado como Ford, Mercedes-Benz, Volkswagen e BMW, por exemplo, começaram a desenvolver veículos capazes de dirigir sozinhos. No decorrer do desenvolvimento, tecnologias criadas para os futuros autônomos foram sendo implementadas aos poucos em veículos comerciais, aumentando o nível dos mesmos na escala de automação veicular.

Os veículos começaram no nível 0, possuindo tecnologias como frenagem automática de emergência, detecção de ponto cego e aviso de evasão de faixa, por exemplo, aumentando a complexidade conforme o desenvolvimento aumentava, até chegarem ao nível 1, com correção automática em caso de evasão de faixa e controle de cruzeiro adaptativo, por exemplo.

Porém, somente com o avanço dos carros elétricos, em específico pela marca Tesla, veículos de nível 2 e 3 chegaram ao consumidor final e começaram a se popularizar.

³Imagem retirada de: <https://www.rediscoverthe80s.com/2016/11/navlab-the-selfdriving-car-of-the-80s.html>

de:

<https://www.rediscoverthe80s.com/2016/11/>

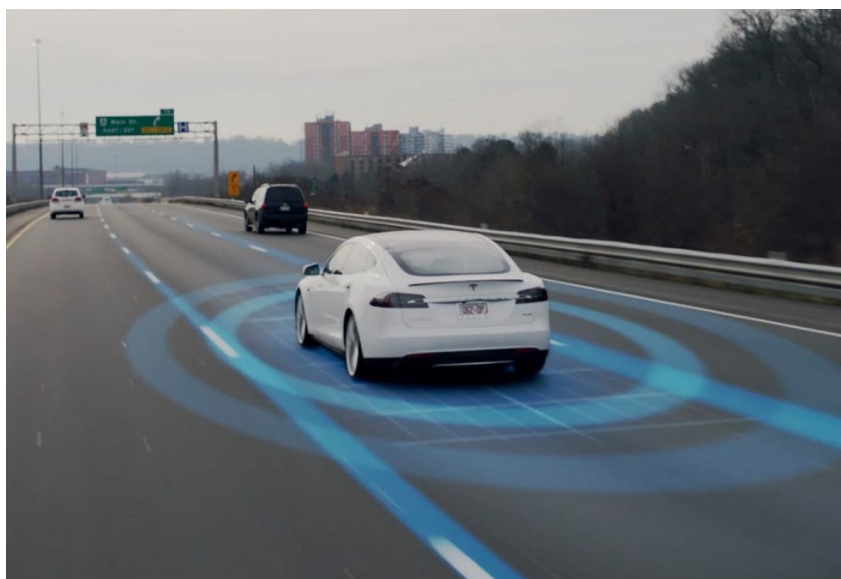
2.2.2.1 Tesla

A Tesla foi criada em 2003 com o intuito de desenvolver bons carros elétricos, capazes de competir diretamente com os a combustão, e, dessa forma, os popularizar. Os carros se tornaram populares entre os entusiastas, consolidando a empresa, mas ainda sim eram considerados de nicho, possuindo um alto valor de produção e não conseguindo atrair a curiosidade e o interesse do grande público.

Em 2015 a Tesla lança seu sistema nível 2 de automação veicular, chamado de Tesla *Autopilot*, para todos os proprietários do modelo *Model S*, demonstrado na Figura 4. O *Autopilot* é capaz de levar o veículo do ponto A ao ponto B, com uma rota traçada anteriormente pelo GPS, de maneira em que o condutor não precise conduzir o veículo durante o percurso.

O Autopilot surgiu como um grande avanço na trajetória ao nível de 5 de automação veicular, sendo usado também como peça publicitária para chamar a atenção do público os veículos elétricos, conseguindo os popularizar. Além disso, chamou também a atenção da indústria automotiva de modo geral, acelerando as montadoras tradicionais de veículos a combustão, dada a cobrança e interesse do consumidor final, em seu desenvolvimento de uma tecnologia de automação similar a da Tesla capaz de atingir o nível 2 ou nível 3.

Figura 4 – Tesla Model S.



⁴Disponível em: <https://www.tesla.com>. Acesso em: 3 Mai. 2022.

Para seu funcionamento, o Autopilot utiliza um *hardware* similar ao NavLab 1, com câmeras 360, sensores ultra-sônicos, sensores infravermelho e uma vasta gama

⁴Imagem retirada de: <https://www.tesla.com/videos/enhance-your-commute-autopilot>

de processadores com propósitos variados. Controlando todo o complexo equipamento presente fisicamente, existe o *software* utilizando o método de visão computacional.

2.2.2.2 Visão Computacional

Visão computacional é um área da inteligência artificial que permite computadores e sistemas digitais abstrair informações de imagens, vídeos, sensores diversos e qualquer outra entrada digital para tomar ações e decisões baseadas nos dados adquiridos nas entradas (IBM, 2019).

Para tal é necessário uma alta quantidade de dados, onde, computacionalmente, seja possível treinar o modelo desenvolvido e interpretar de maneira correta as entradas. Para treinar um modelo capaz de identificar um pneu, por exemplo, é necessário munir as entradas com diversas imagens de pneus em diferentes situações para que seja possível observar um padrão, e, dessa forma, o modelo caracterizá-lo como um pneu.

Duas tecnologias essenciais para o desenvolvimento de um modelo que utiliza visão computacional são: uma divisão do *Machine Learning* chamado *Deep Learning* em conjunto com uma Rede Neural Convolucional.

O *deep learning* utiliza modelos de algoritmos capazes de gerar os padrões, capacitando o modelo a distinguir uma coisa da outra. Para isso utiliza as imagens de treinamento e as difere uma das outras, observando as diferenças gerando categorias para cada imagem e aprendendo com essas mesmas categorias.

Já a Rede Neural Convolucional, auxilia o *deep learning* nessa tarefa de auto aprendizado, quebrando as imagens em pixels e as categorizando por distância, discernindo bordas e formas, gerando, com esses pixels analisados, tags e rótulos. Essas demarcações são utilizadas para gerar convoluções, operação matemática capaz de gerar um terceira função a partir de outras duas funções, predizendo o que está sendo observado. A rede neural roda então uma série de convoluções verificando a acurácia das mesmas até que chegue ao valor definido pelo modelo.

A visão computacional é utilizada por diversas montadoras e se prova até o momento o método mais eficaz para veículos autônomos externos, porém, possui um alto grau de complexidade e esbarra em diversas questões éticas e legais no que se diz respeito à legislação de trânsito pela questão da segurança de um veículo autônomo trafegando entres veículos não autônomos.

3 INTELIGÊNCIA ARTIFICIAL

O campo de inteligência artificial data desde o primeiros computadores criados, porém antes mesmo disso, quando o que se possuía eram apenas máquinas calculadoras e máquinas de programa armazenado, a ideia de se possuir um sistema capaz de executar suas funções de forma automatizada já vivia no imaginário dos programadores e cientistas da época. O termo IA, segundo [Khemani \(2013\)](#), só veio a ser cunhado de maneira oficial por John McCarthy, organizador do workshop de Dartmouth, seu projeto em conjunto com outros cientistas e matemáticos da área da computação que resultou em premissas que viriam a ser utilizadas utilizadas pelos anos a seguir, e também, a apresentação daquele que é considerado o primeiro programa de computador funcional como inteligência artificial, criado por Allen Newell, Herbert Simon e J. C. Shaw, o Logic Theorist.

3.1 ESTRUTURANDO UM PROBLEMA

Para se imaginar uma aplicação a inteligências artificiais, primeiro se faz necessário postular qual é o problema a ser resolvido e quais os passos a serem tomados que possibilitam o alcance de resultados satisfatórios. Em um exemplo prático, é possível imaginar um viajante, que sem qualquer instrução prévia deseja alcançar a cidade de São Paulo partindo de Sorocaba em no máximo duas horas de viagem, o objetivo neste caso é claro e se dá em alcançar a cidade de destino dentro do período de tempo determinado, qualquer outro resultado é visto como insatisfatório.

Analisando como fazer com que este objetivo seja atingido, é possível que as instruções sejam dadas através de movimentos que o carro deve realizar, como seguir em frente ou virar à esquerda, mas pelo tamanho do problema proposto, a quantidade de instruções a serem dadas alcançariam um nível de complexidade muito alto além de serem muito amplas quanto à quantidade. Estas instruções podem vir a ser simplificadas realizando uma diminuição em seu escopo, sendo os possíveis passos deste viajante chegar a cidades vizinhas próximas.

Seus passos então seriam alcançar a cidade de Votorantim, Alumínio ou Araçoiaba da Serra, cada um adicionando um tempo referente a seu trajeto, por exemplo, de início ele não saberá qual dessas cidades possui a melhor rota ou a sequer possibilidade de levá-lo a São Paulo, como nenhuma destas rotas o faz de fato chegar ao objetivo final, cabe à aleatoriedade definir se a escolha sendo feita de fato é a melhor aplicável ou não, adicionando uma camada de incerteza. Tratando-se de modelos computacionais, existem métodos a serem discutidos a seguir que pesarão na tomada de decisão.

Dentre os fatores deste problema estruturado, o viajante assume um ambiente observável, pois este sempre sabe onde está inclusive caso tenha alcançado seu objetivo no tempo necessário, o ambiente também pode ser considerado discreto, pois a quantidade de cidades que delimitam Sorocaba e qualquer outra dentro do trajeto é finita. E o resultado é sempre determinístico, com uma resposta de destino alcançado a tempo ou não.

Fazendo um breve paralelo com um algoritmo que viesse a executar este problema, a entrada de dados seria dada pela posição inicial do viajante, e dadas as possíveis cidades que este pode seguir, um retorno seria dado de uma nova lista de cidades com o tempo percorrido até então, a cada novo passo uma verificação se o objetivo foi alcançado seria feita, sendo esta a condição de parada em caso de sucesso ou falha por tempo limite atingido.

******IMAGEM REPRESENTATIVA******

3.2 UTILIZANDO INTELIGÊNCIAS ARTIFICIAIS

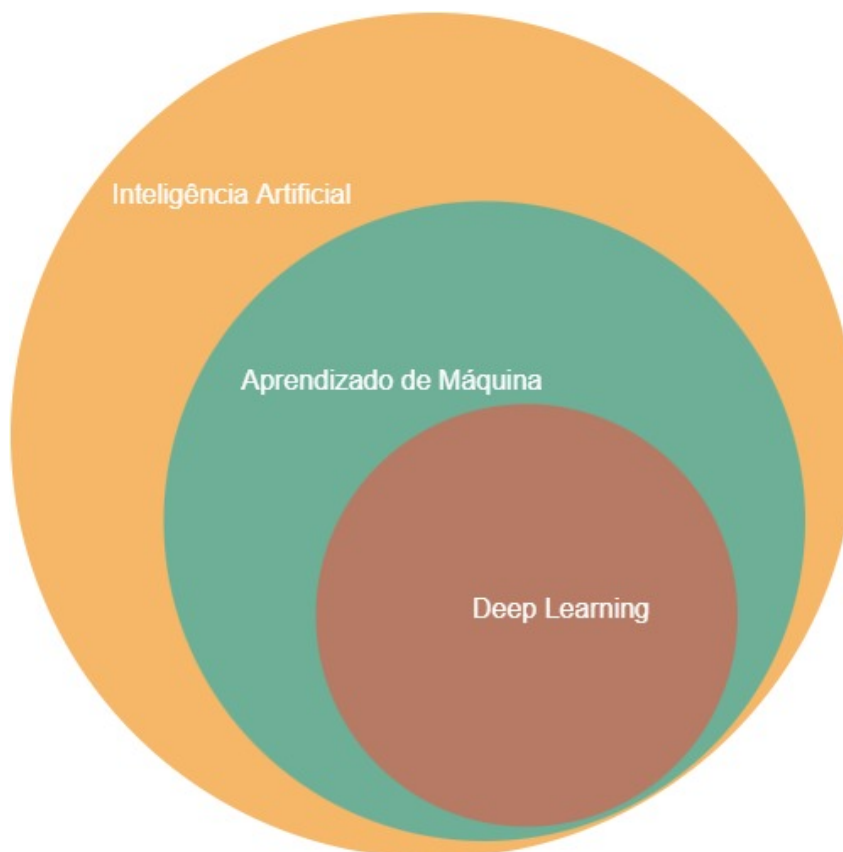
Sistemas Inteligentes são todos aqueles definidos como sistemas formais ou informais utilizados para obter dados e interpretá-los aplicando tecnologias de Inteligência Artificial (IA) e *Business Intelligence* (BI), fornecendo resultados como base para ações (SHARDA; DELEN; TURBAN, 2019). Estes sistemas podem ser considerados especialistas por definição, pois sua base de dados provém de dados abastecidos por um especialista de determinada área, com seu conhecimento armazenado, o sistema consegue emular a tomada de decisão de um profissional inferindo um resultado arbitrário. Ao mesmo tempo que essa é uma solução interessante para diversas aplicações do mercado, sistemas especialistas são pouco versáteis por geralmente não terem a capacidade de gerar resultados fora do universo que já conhecem, os limitando para simples Inteligências Artificiais.

De acordo com o diagrama da Figura 5, dentro dos escopos dos sistemas de Inteligência Artificial, entre aqueles que resolvem tarefas complexas independentemente ou com pouca intervenção humana (RUSSELL, 2010) se encontram as aplicações de Aprendizado de Máquina (ML), que são capazes de melhorar seu desempenho baseando-se numa alteração de seu modelo analítico e classificando os resultados, fazendo com que futuras iterações da execução do sistema sejam influenciadas, criando assim uma “inteligência própria”. O *Deep Learning* é uma metodologia mais recente da qual utiliza de padrões de ML e os associa com padrões biológicos humanos, como a ativação de neurônios simulada por uma rede neural computacional, explorada adiante, esta metodologia utiliza de funções mais complexas como convoluções e múltiplas ativações de um neurônio (JANIESCH; ZSCHECH; HEINRICH, 2021) e geralmente

possui uma vantagem em relação à seu supertipo na resolução de problemas de reconhecimento de placas ou de voz, por conseguirem encontrar padrões nos dados de entrada dentro da própria tarefa sendo executada.

*****DEEP LEARNING -> APRENDIZADO PROFUNDO*****

Figura 5 – Diagrama de Euler da relação entre AI, ML e DL.



Fonte: Autoria própria.

3.3 APRENDIZADO DE MÁQUINA

“Aprendizado” pode ser definido como processo por meio do qual uma nova informação é incorporada à estrutura cognitiva do indivíduo, por se relacionar a um aspecto relevante dessa estrutura. Esse novo conteúdo poderá modificar aquele já existente, dando-lhe outros significados (MICHAELIS, 2022). Tais características supracitadas quando implementadas no universo computacional recebem o nome de Aprendizado de Máquina.

O termo Aprendizado de Máquina foi criado por Arthur Samuel, da IBM, em 1962 após o desenvolvimento de um algoritmo capaz de vencer um humano durante uma partida de damas. Comparado ao que pode ser feito hoje, esse feito quase parece trivial, mas é considerado um marco importante no campo da inteligência artificial (IBM,

2022a). Aprendizado de Máquina consiste em utilizar dados de diferentes fontes e tipos, e, através de algoritmos baseados em métodos e modelos matemáticos, executar diferentes tarefas de maneira similar a um ser humano, fazendo classificações ou predições das situações em que se encontra inserido. Os dados necessários para o treinamento de um modelo de Aprendizado de Máquina podem ser organizados e classificados previamente através de processos de refinamentos de dados ou dados não rotulados, apresentados ao modelo de maneira dispersa e não especificada. A escolha de como tais dados serão utilizados varia de acordo com o algoritmo e método de treino escolhidos.

Os tipos de treinamento de um modelo de Aprendizado de Máquina podem ser divididos em três grandes grupos de técnicas classificatórias primárias, sendo elas: supervisionado, não supervisionado e semi supervisionado.

Supervisionado: De acordo com [Learned-Miller \(2014\)](#), aprendizado supervisionado é uma simples formalização da ideia de aprender através de exemplos. Nessa técnica, os dados fornecidos ao modelo devem passar por um tratamento prévio para normalização e organização. Junto a isso, os dados são separados em treino e teste. Os dados de treino são utilizados para ensinar o modelo, já os de teste são utilizados para verificar a eficácia do aprendizado, comparando o resultado dado pelo modelo com o resultado já conhecido para aquela combinação de valores. Por exemplo, para prever se vai ou não chover com base no clima, é necessário que o algoritmo tenha uma larga base de dados sobre a situação meteorológica de dias que choveram e de dias que não choveram, separando a base em treino e teste e então iniciar o processo de aprendizado, verificando a eficácia durante o processo para, então, ao atingir uma boa capacidade preditiva seja possível utilizar o modelo com dados abstraídos de situações reais.

Não supervisionado: Os algoritmos que utilizam o aprendizado não supervisionado recebem os dados de maneira não necessariamente tratados e não rotulados, sem possuir um resultado específico para uma dada combinação de dados. Esse tipo de aprendizado propicia o descobrimento de padrões, sendo possível os categorizar sem a necessidade de análise e intervenção humana. O objetivo do aprendizado não supervisionado é melhorar o desempenho de tarefas supervisionadas quando não se possui uma grande quantidade de dados ([SUTSKEVER et al., 2015](#)). Essa habilidade de descobrir semelhanças e diferenças entre utilizando diferentes tipos de dados permite que o aprendizado não supervisionado possa ser utilizado em reconhecimento de imagem e segmentação de consumidores, onde é possível caracterizar tendências de compra com base na maneira com que um grupo de usuários utiliza as redes sociais, por exemplo.

Semi supervisionado: Nesse modelo, o algoritmo utiliza um meio termo entre

supervisionado e não supervisionado. Durante o treinamento, ele usa um conjunto de dados rotulados menores para orientar a classificação e a extração de recursos de um conjunto de dados maior e não rotulado (IBM, 2022b). Com isso é possível auxiliar o descobrimento de diferentes padrões e os classificar baseado no aprendizado previamente realizado com dados tratados.

3.4 APRENDIZADO POR REFORÇO

O Aprendizado por Reforço se trata se uma subcategoria de aprendizado de máquina que, de forma sumarizada, mapeia situações a ações de forma a maximizar uma recompensa numérica. O aprendiz não é auxiliado em que ações tomar, mas deve descobrir quais ações rendem mais recompensas ao experimentá-las (KAELBLING; LITTMAN; MOORE, 1996). Esse método exploratório faz contraste com métodos de aprendizado *supervisionados*, em que o algoritmo recebe a informação “correta” e tenta se aproximar ao máximo dessa solução conhecida; e também com os métodos *não supervisionados*, que utilizam métodos de associação e *clustering* para tomar suas decisões. A Figura 6 ilustra as relações e principais diferenças entre esses métodos.

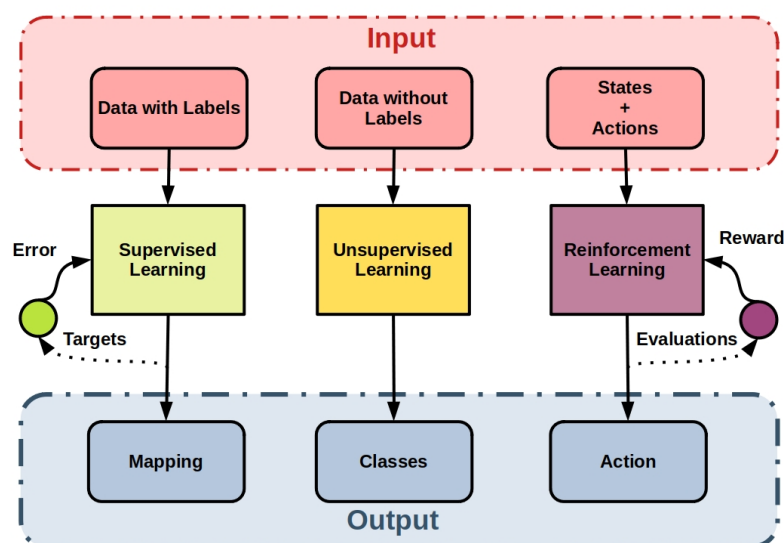
*****Talvez fosse interessante explicar um pouco mais a imagem, citando os termos existentes nela (termos mantidos em inglês). Algo assim: na parte central da imagem temos os sistemas: supervisionado, não supervisionado e por reforço. Acima e abaixo, respectivamente, pode se observar as entradas e saídas.....etc*****

Esse tipo de aprendizado facilita ou até possibilita experimentos em que outros métodos teriam muita dificuldade ou aumentariam a complexidade do problema significativamente. Sutton e Barto (2018) indica que esse campo de estudo tem atraído cada vez mais interesse nas comunidades de IA e ML, pelo seu modo de programar agentes por recompensas e punições através de tentativa e erro e sem precisar indicar *como* o problema deve ser resolvido.

Uma das principais aplicações de aprendizado por reforço se faz em problemas no espaço físico, em que se recebe uma quantidade grande de dados e que a relação entre tais dados e a ação ou resultado se fazem difíceis por meios matemáticos. Enquanto esses algoritmos performam muito bem na resolução dos problemas, o processo de treinamento no mundo real pode ser extremamente custoso e demorado. Nestes casos, simulações (contanto que sejam próximas o suficiente da realidade) podem trazer uma atraente alternativa para simplificar o treinamento de um modelo complexo (RAO et al., 2020).

¹ Imagem retirada de: <https://starship-knowledge.com/supervised-vs-unsupervised-vs-reinforcement>

Figura 6 – Tabela com visualizações de diferentes métodos de aprendizado de máquina.



¹Disponível em: <https://starship-knowledge.com>. Acesso em: 4 Set. 2022.

Tendo uma simulação satisfatória para o treinamento, o problema deixa de ser o tempo e custo de um processo físico de treinamento e passa a ser o poder computacional para calcular tanto a simulação quanto o algoritmo de aprendizado simultaneamente. Neste caso, a aplicação de um método heurístico nas iterações pode beneficiar o sistema na otimização da criação de novos cenários. Além disso, métodos heurísticos evolucionários podem ser mais facilmente paralelizáveis, tendo que as ações intermediárias podem ser descartadas e ter somente o resultado final (usualmente aptidão ou *fitness*) sendo considerado para o aprendizado.

4 TECNOLOGIAS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

4.1 ALGORÍTMO GENÉTICO

O processo evolutivo no campo da biologia é definido por [Ridley \(2009\)](#) como uma mudança na forma e nos comportamentos de uma determinada espécie ao longo das gerações por um processo de seleção natural, “onde o indivíduo mais bem adaptado sobrevive e deixa mais descendentes do que o menos adaptado, o que conduz ao declínio de sua variedade ou espécie, eventualmente conduzindo-a à extinção” ([CARMO; BIZZO; MARTINS, 2009](#)).

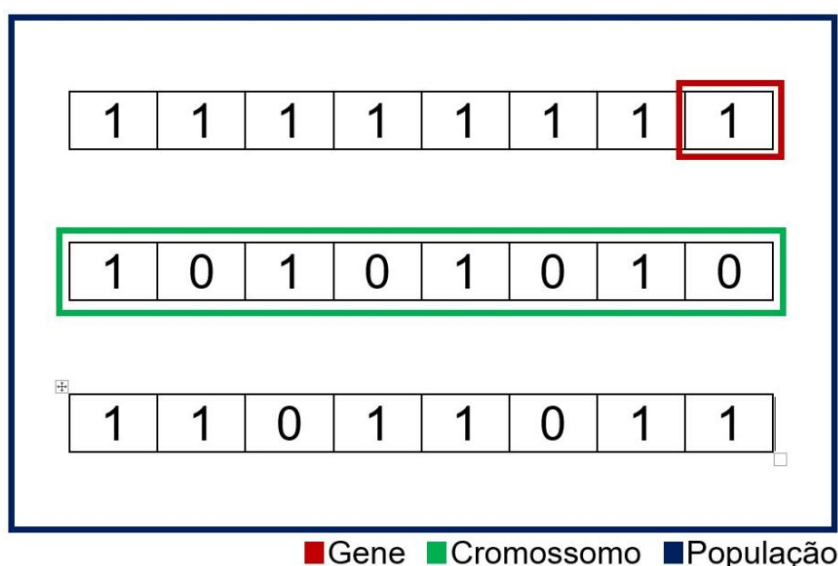
A ideia de um AG é trazer todos esses conceitos embutidos na biologia para modelos computacionais capazes de solucionar problemas complexos de otimização por métodos meta-heurísticos. Tais algoritmos não possuem uma regra de produção para seu desenvolvimento e implementação, sendo apenas necessário abstrair a teoria biológica para o universo da computação. Porém, de acordo com [Mitchell \(1998\)](#), algumas características, por convenção, sempre estão presentes, como, por exemplo: populações de indivíduos, seleção dos mais aptos, cálculo de função fitness, crossovers para a reprodução e mutações aleatórias.

4.1.1 Indivíduo e população

Com base nas informações supracitadas é possível destrinchar cada característica para melhor entendimento, iniciando pelo elemento base do AG, o indivíduo.

Cada indivíduo de uma população é composto por uma sequência de cromossomos. Cada cromossomo é composto por um conjunto de valores binários. Cada valor binário desse conjunto é chamado de gene. Então, com base nisso, é possível concluir que cada indivíduo possui sua forma e comportamento ditados pelo seu respectivo cromossomo e uma população é composta por um conjunto de n diferentes cromossomos, como pode ser visto na Figura 7.

Figura 7 – Gene, Cromossomo e População.



Fonte: Autoria Própria.

4.1.2 Reprodução

A reprodução é a parte considerada mais importante de um AG. É nela que acontece a seleção e o cruzamento de cromossomos da população com o intuito de gerar a evolução dos indivíduos e, dessa forma, atingir o melhor resultado para a situação.

Esse processo de reprodução é feito em etapas e ocorre sempre que uma geração completa um ciclo de execução pré estabelecido e há a necessidade de se criar uma nova geração. Esse processo de reprodução somente chega ao fim quando o resultado gerado pelo AG atinge o valor esperado, ou por uma condição de parada definida anteriormente.

Referente às etapas que constituem a reprodução, estão inclusos os operadores de seleção, *crossover* e mutação.

4.1.2.1 Seleção

A etapa de seleção leva em consideração uma função chamada *fitness*, que é única para cada indivíduo de cada geração após a execução de um ciclo. O cálculo desta função varia e é definida de acordo com o problema que o AG está inserido, utilizando dados essenciais para que seja possível classificar que um cromossomo obteve um bom desempenho em uma determinada situação.

Com a *fitness* de cada indivíduo calculado, “o operador seleciona os cromossomos da população para a reprodução, quanto maior o valor da função fitness, maior a chance desse mesmo cromossomo ser selecionado mais vezes para reprodução” (MITCHELL, 1998). Além dos indivíduos com melhor fitness, são selecionados também, de maneira aleatória, os demais cromossomos, para que a chance de chegar em máximo local seja reduzida. A quantidade de indivíduos selecionados, assim como a proporção de reprodução, deve ser minimamente suficiente para que a geração seguinte possua o mesmo tamanho da anterior.

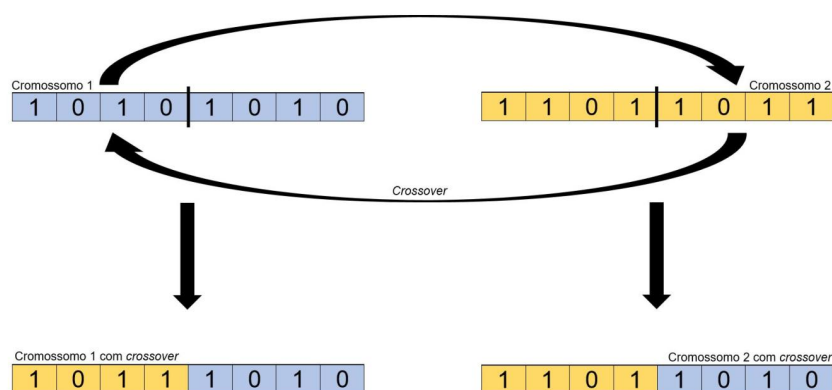
4.1.2.2 Crossover

O crossover é o responsável por gerar a evolução propriamente dita. É durante essa etapa que os indivíduos recebem características novas, porém, não necessariamente melhores. Como dito anteriormente, o AG é um algoritmo meta-heurístico, ou seja, busca otimizar problemas que não possuem necessariamente apenas um resultado bom, mas sim diversos resultados bons. Essa classificação se dá pelo “fato dos cromossomos serem tratados apenas como sequências de bits sem que sejam feitas inferências a respeito dos seus significados” (PAULINO, 2018). Por isso, cromossomos anteriormente bons podem perder suas qualidades e obterem resultados piores na geração seguinte, porém, a vantagem é que o contrário também é verdadeiro. Cromossomos ruins podem ganhar qualidades e obterem resultados melhores.

Essas trocar de características e cruzamentos entre cromossomos que é chamado de crossover. A maneira com a qual o *crossover* é desenvolvido varia de acordo com a situação em que o AG está inserido. Uma das maneiras mais simples de realizar o crossover é pegar dois cromossomos, ou seja, duas cadeias binárias e realizar a inversão da primeira metade do primeiro cromossomo com a segunda metade do segundo cromossomo, como demonstrado na ?? a seguir.

Dessa maneira ambos os cromossomos se beneficiam, ou não, das caracte-

Figura 8 – Crossover entre dois cromossomos.



Fonte: Autoria Própria.

rísticas do outro sem perder todas as suas próprias características. Esse processo cria uma nova população porém, as características da geração anterior, por mais que distribuídas entre os cromossomos, ainda estão presentes, só organizadas de maneira diferente, não tendo uma evolução propriamente dita, para isso que a próxima etapa, mutação, se faz necessária.

4.1.3 Mutação

Após o *crossover*, a nova população possui cromossomos com genes organizados de maneira diferente. A mutação entra para gerar uma alternância nos genes dos cromossomos selecionados aleatoriamente com o intuito de gerar novas características para a população. Porém, apesar de útil, a mutação deve ser utilizada com cautela, tanto na questão de cromossomos selecionados para mutação, quanto na quantidade de genes selecionados por cromossomo, para que não haja alterações descontroladas e nada produtivas para a resolução do problema.

4.2 REDES NEURAIS

Uma Rede Neural é uma abstração computacional baseada no funcionamento do sistema nervoso dos animais. Esse sistema é composto por diversos neurônios e conexões entre esses mesmos neurônios para que sejam enviados sinais com comandos para as demais células do organismo vivo. A capacidade de tomada de decisão advém da organização dos neurônios e suas conexões formando uma grande e complexa rede onde a menor unidade que pode ser encontrada é o neurônio (ROJAS, 2013).

Um neurônio funciona recebendo informações fornecidas por demais neurônios,

gerando uma saída própria baseada nessas mesmas informações. No universo computacional, o modelo com o objetivo de simular tais características de um neurônio é chamado de *perceptron*.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

4.3 NEUROEVOLUÇÃO DE TOPOLOGIAS AUMENTANTES

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus

scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

5 DESENVOLVIMENTO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

5.1 METODOLOGIA

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

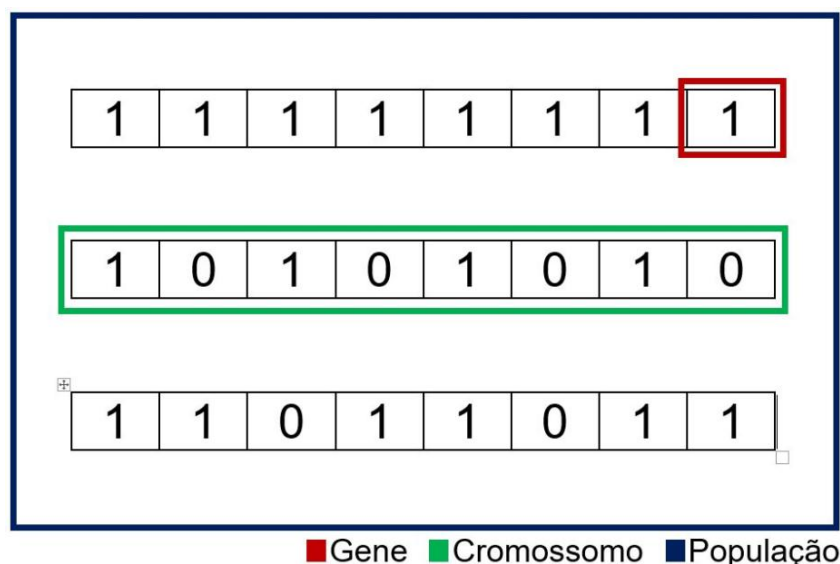
5.2 SIMULAÇÃO

O objetivo proposto implica no treinamento de carros autônomos que pudessem vir a ser dispostos em ambientes práticos de modo a saber lidar com seus arredores e alcançar resultados práticos. Seja o ambiente real uma réplica do circuito presente na aplicação ou um ambiente novo, a validação dos testes vem a acontecer em sua totalidade através de um *software* desenvolvido.

Este espaço virtual visa simular um carro dentro de um circuito fechado cercado por paredes, o qual, pondo em prática sua autonomia, está equipado de sensores que medem sua distância ao bloqueio mais próximo à sua frente, e num intervalo de 45° para ambos os lados até se tornarem perpendiculares, somando-se assim cinco

distâncias a serem medidas. Na Figura 9 pode-se observar a representação gráfica desse espaço.

Figura 9 – Interface gráfica utilizada para a simulação, com o modelo de carro autônomo, trajeto de treino a ser percorrido e botões de controle de *layout*.



Fonte: Autoria Própria.

O ambiente de simulação utilizado como base para os experimentos foi desenvolvido em python com a biblioteca *pygame*, sua interface oferece métodos baseando-se em orientação a objeto para a elaboração de jogos e aplicações multimídia, através desta se fez o design da janela de execução com seus elementos de tela (botões e limites do circuito). Em conjunto, se faz uso da biblioteca de física 2d *pymunk*, para controlar a movimentação e colisão do carro.

A configuração do carro, no que diz respeito à sua interação com ambiente, engloba:

1. A posição inicial do carro e a distância por ele percorrida, utilizadas posteriormente como indicadores de desempenho para o algoritmo de neuroevolução;
2. A forma do carro, a princípio definida como um círculo para facilitar sua movimentação pelo espaço, evitando colisões involuntárias;
3. Sensoriamento do carro, identificando sua distância às paredes, posteriormente sendo utilizado para tomadas de decisão.

6 RESULTADOS

O problema proposto, o qual os testes serão realizados, se dá no próprio ambiente de simulação de carros autônomos criado previamente. Neste, serão aplicados dois diferentes algoritmos de inteligência artificial:

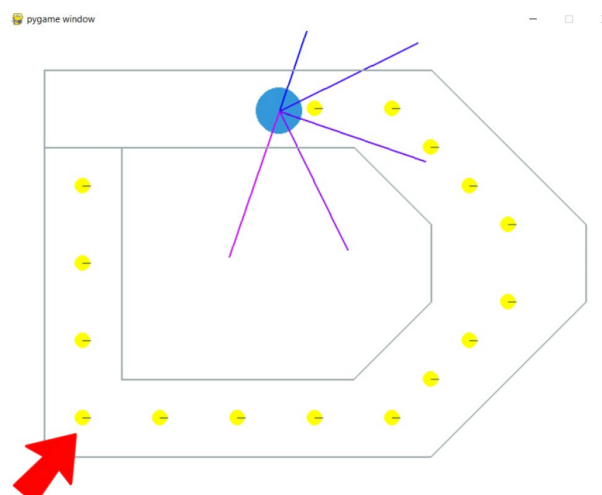
1. NEAT, proveniente do *neat-python* pós parametrização;
2. Neuroevolução, através da aplicação de um algoritmo genético sobre uma rede neural simples (sem topologias aumentantes);

Tais algoritmos trabalham de diferentes formas a encontrar valores de aptidão para suas simulações. De modo a generalizar a classificação de resultados, foi proposto a construção de um valor de pontuação atrelado à distância percorrida pelo carro em sua execução.

Esta métrica pode vir a ser comparada entre métodos a fim de se encontrar quanto tempo é necessário para se alcançar determinada fitness e quanto é alcançado após determinado tempo de execução decorrer.

Os resultados foram obtidos de duas diferentes formas: primeiramente encontrando em que geração do algoritmo este consegue alcançar um valor de *fitness* mínimo necessário para que o carro execute uma volta completa pelo trajeto, definindo assim quão rapidamente ele consegue alcançar uma resposta. A Figura 10 ilustra dentro do ambiente de simulação o alvo a ser alcançado.

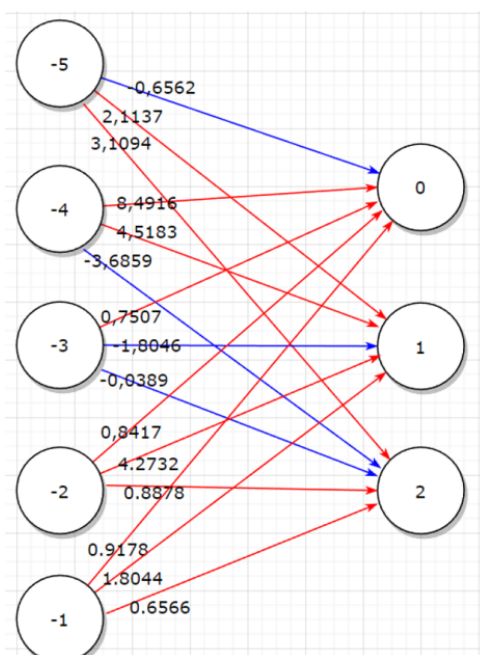
Figura 10 – Canto inferior esquerdo definido como meta a ser alcançada para obtenção de resultado.



Fonte: Autoria Própria.

A Rede Neural inicial é composta por 5 sensores que verificam e alimentam a rede a cada movimento do carro e são elencados com os valores que vão de -5 até -1. Essas 5 entradas estão todas conectadas a cada um das 3 saídas que indicam a direção que o carro deve tomar, as saídas vão de 0 até 2, sendo 0 a indicação de movimento a esquerda, 1 a direita e 2 para frente. Cada conexão entre cada sensor de entrada e cada nó de saída possui um peso específico que serve para a geração do valor de saída. A estrutura indicada acima pode ser encontrada na Figura 11 abaixo.

Figura 11 – Organização da rede, com seus nós e conexões.



Fonte: Autoria Própria.

O processo de aprendizado de uma determinada rede veio a ser executado em um ambiente de treinos, o qual, tratando-se do algoritmo NEAT, a topologia parte de seu estado mais simples até possuir um determinado formato da qual é capaz de finalizar o percurso, sendo tratada como a resposta ideal proposta. O algoritmo de Neuroevolução também partem de seu estado mais simples até as respostas desejadas neste ambiente.

6.1 NEAT

A simulação do treino para o algoritmo NEAT incorpora o aumento das topologias, este pode vir a ser dado pela possibilidade de adições e subtrações de nós nas camadas ocultas da rede e/ou remoção ou adição de conexões entre nós, fatores observados nas redes dos resultados alcançados.

Foram realizadas três execuções com o objetivo de se alcançar a fitness mínima para que uma volta completa seja realizada em torno da pista, mensurado como cerca de 16000 pontos de aptidão com 300 membros na população. Os resultados com as respectivas gerações em que o cálculo foi alcançado se dão na Tabela 1, juntamente da topologia resultante em cada iteração.

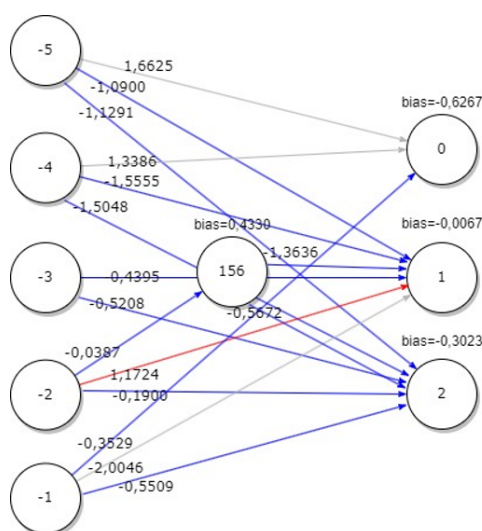
Tabela 1 – Resultados obtidos pelo NEAT até a meta de Fitness ser alcançada.

| Iterações | Topologia | Geração | Fitness | Fitness média |
|-----------|-----------|---------|---------|---------------|
| 1 | (3,13) | 8 | 16379 | 3664 ± 2106 |
| 2 | (3,14) | 3 | 16337 | 3285 ± 2165 |
| 3 | (4,13) | 8 | 16336 | 3684 ± 2126 |

Fonte: Autoria própria.

Como observável na Figura 12, a terceira iteração obteve uma topologia a qual indica a existência de quatro nós e treze conexões totais, este nó intermediário, inserido após um processo de mutação, serve como uma conexão na camada oculta entre a entrada -2 e as saídas 1 e 2. Além desta alteração, a topologia também sofreu alterações com a remoção da conexão entre a entrada -3 e saída 0, julgada pelo algoritmo como uma melhoria para o desempenho durante suas mutações e cruzamentos.

Figura 12 – Rede neural da Iteração 3 de treinos do algoritmo NEAT, com uma conexão excluída e um nó adicional na camada oculta.



Fonte: Autoria Própria.

A segunda bateria de treinos englobava a execução do algoritmo até sua 10ª geração, e a comparação dos resultados obtidos até então. Seus resultados são observáveis na Tabela 2.

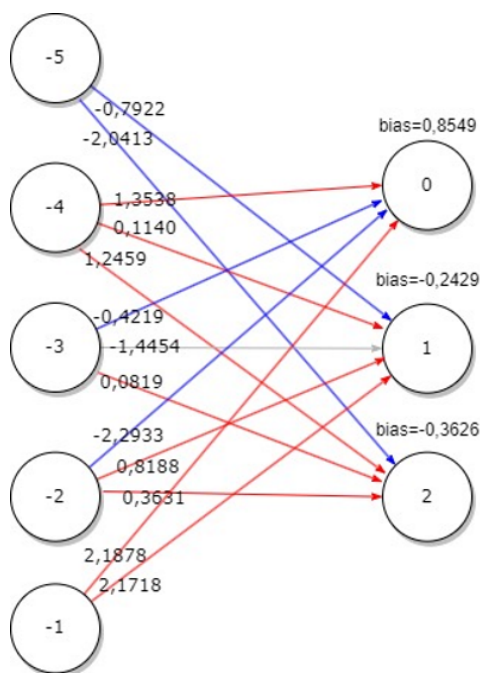
Tabela 2 – Resultados obtidos pelo NEAT após a execução de 10 gerações.

| Iterações | Topologia | Geração | <i>Fitness</i> | <i>Fitness</i> média |
|-----------|-----------|---------|----------------|----------------------|
| 1 | (4,9) | 10 | 16383 | 3191 ± 2524 |
| 2 | (3,12) | 10 | 18528 | 3847 ± 1997 |
| 3 | (4,15) | 10 | 14199 | 3569 ± 1903 |

Fonte: Autoria própria.

Observando a topologia da iteração de número 2, é possível verificar que esta que obteve o melhor resultado com o mesmo tempo de execução das demais, identificou como uma melhora no resultado a quebra da conexão entre a entrada -5 e a saída 0 e a entrada -1 e saída 2, como observável na Figura 13.

Figura 13 – Rede neural da Iteração 2 de treinos do algoritmo NEAT, com duas conexões excluídas e uma desativada.



Fonte: Autoria Própria.

6.2 NEUROEVOLUÇÃO

Para a simulação de um algoritmo básico de Neuroevolução sem a etapa de aumento de topologias, se fez uso da mesma estrutura utilizada anteriormente pelo NEAT, porém sem a possibilidade de haver adições ou subtrações de nós (nas camadas ocultas) ou conexões (entre entradas, nós e saídas).

Como apresentado na Tabela 3, o algoritmo foi executado três vezes com diferentes valores para *fitness* alcançada e em que geração esta se deu, considerando

uma população de 300 integrantes.

A coluna de formato da rede não se faz necessária considerando que este, por se tratar de um algoritmo simples de neuroevolução, possui suas tomadas de decisões realizadas apenas na pesagem de suas arestas, sem alterações de topologia durante a execução.

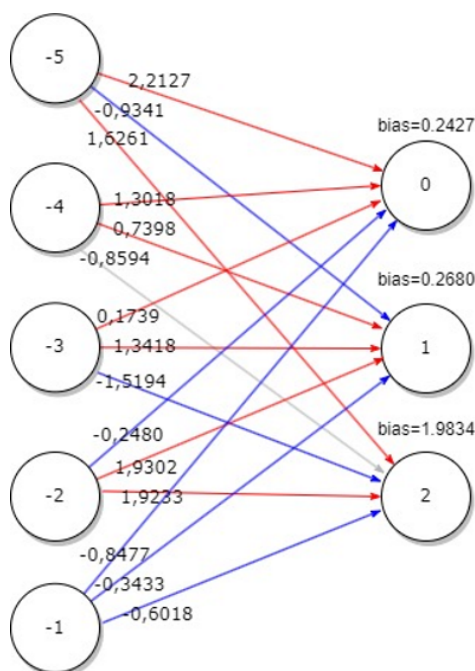
Tabela 3 – Resultados obtidos em suas respectivas gerações após execução do algoritmo de Neuroevolução.

| Iterações | Geração | Fitness | Fitness média |
|------------------|----------------|----------------|----------------------|
| 1 | 3 | 16345 | 3354 ± 2138 |
| 2 | 12 | 16373 | 3839 ± 1930 |
| 3 | 12 | 16382 | 3788 ± 2298 |

Fonte: Autoria própria.

A rede neural referente à iteração que alcançou o resultado mais rapidamente pode ser expressa na Figura 14, com suas devidas conexões e pesos.

Figura 14 – Rede neural da Iteração 1 de treinos do algoritmo de Neuroevolução.



Fonte: Autoria Própria.

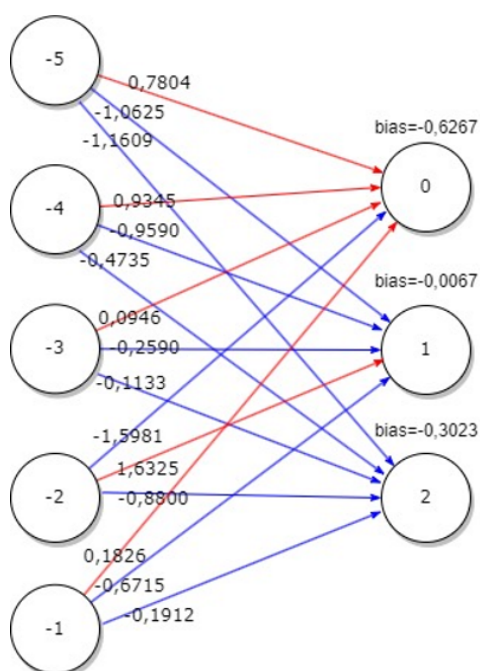
A Tabela 4 ilustra os resultados obtidos para o algoritmo de Neuroevolução considerando três execuções até a décima geração, com a fitness média e máxima alcançadas dentro da população de 300 integrantes para cada iteração. Do mesmo modo, a rede neural com o resultado da iteração de melhor resultado pode ser expressa na Figura 15, com suas devidas conexões e pesos.

Tabela 4 – Resultados obtidos até a décima geração após execução do algoritmo de Neuroevolução.

| Iterações | Geração | <i>Fitness</i> | <i>Fitness média</i> |
|-----------|---------|----------------|----------------------|
| 1 | 10 | 16356 | 3754 ± 2159 |
| 2 | 10 | 19596 | 3923 ± 2101 |
| 3 | 10 | 16335 | 3767 ± 1966 |

Fonte: Autoria própria.

Figura 15 – Rede neural da Iteração 2 após 10 gerações de treinos do algoritmo de Neuroevolução.



Fonte: Autoria Própria.

7 CONSIDERAÇÕES FINAIS

Sed consequat tellus et tortor. Ut tempor laoreet quam. Nullam id wisi a libero tristique semper. Nullam nisl massa, rutrum ut, egestas semper, mollis id, leo. Nulla ac massa eu risus blandit mattis. Mauris ut nunc. In hac habitasse platea dictumst. Aliquam eget tortor. Quisque dapibus pede in erat. Nunc enim. In dui nulla, commodo at, consectetur nec, malesuada nec, elit. Aliquam ornare tellus eu urna. Sed nec metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Phasellus id magna. Duis malesuada interdum arcu. Integer metus. Morbi pulvinar pellentesque mi. Suspendisse sed est eu magna molestie egestas. Quisque mi lorem, pulvinar eget, egestas quis, luctus at, ante. Proin auctor vehicula purus. Fusce ac nisl aliquam ante hendrerit pellentesque. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi wisi. Etiam arcu mauris, facilisis sed, eleifend non, nonummy ut, pede. Cras ut lacus tempor metus mollis placerat. Vivamus eu tortor vel metus interdum malesuada.

Sed eleifend, eros sit amet faucibus elementum, urna sapien consectetur mauris, quis egestas leo justo non risus. Morbi non felis ac libero vulputate fringilla. Mauris libero eros, lacinia non, sodales quis, dapibus porttitor, pede. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Morbi dapibus mauris condimentum nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Etiam sit amet erat. Nulla varius. Etiam tincidunt dui vitae turpis. Donec leo. Morbi vulputate convallis est. Integer aliquet. Pellentesque aliquet sodales urna.

REFERÊNCIAS

- Anthony Townsend. *The 100-Year History of Self-Driving Cars*. 2020. Disponível em: <https://onezero.medium.com/the-100-year-history-of-self-driving-vehicles-10b8546a3318>. Acesso em: 2022-05-03.
- ARRUDA, T. A. *Arquitetura de Hardware e Software para Supervisão e Controle de um Carro Autônomo*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, 2015.
- CARMO, V. A. do; BIZZO, N.; MARTINS, L. A.-C. P. Alfred russel wallace e o princípio de seleção natural. *Filosofia e História da Biologia*, Associação Brasileira de Filosofia e História da Biologia-ABFHiB, v. 4, n. 1, p. 209–233, 2009.
- DAS, S. K.; PASAN, M. K. Design and methodology of automated guided vehicle. *IOSR journal of mechanical and civil engineering*, 04 2016.
- ESTADÃO. *Primeiro projeto de carro autônomo data de 1920*. 2020. Disponível em: <https://summitmobilidade.estadao.com.br/carros-autonomos/primeiro-projeto-de-carro-autonomo-data-de-1920/>. Acesso em: 2022-05-03.
- IBM. *What is Computer Vision?* 2019. Disponível em: <https://www.ibm.com/topics/computer-vision>. Acesso em: 2022-05-03.
- IBM. *Machine Learning*. 2022. Disponível em: <https://www.ibm.com/br-pt/cloud/learn/machine-learning>. Acesso em: 2022-09-04.
- IBM. *Unsupervised Learning*. 2022. Disponível em: <https://www.ibm.com/cloud/learn/unsupervised-learning>. Acesso em: 2022-09-04.
- JANIESCH, C.; ZSCHECH, P.; HEINRICH, K. Machine learning and deep learning. *Electronic Markets*, Springer, v. 31, n. 3, p. 685–695, 2021.
- KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, v. 4, 1996.
- KHEMANI, D. *A first course in artificial intelligence*. [S.l.]: McGraw Hill Education (India), 2013.
- LDSV - USP. *Veículos Autônomos*. 2016. Disponível em: http://www.usp.br/ldsv/?page_id=1856. Acesso em: 2022-05-03.
- LEARNED-MILLER, E. G. Introduction to supervised learning. *I: Department of Computer Science, University of Massachusetts*, p. 3, 2014.
- MICHAELIS. *Aprendizado*. 2022. Disponível em: <https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/aprendizagem>. Acesso em: 2022-09-04.
- MITCHELL, M. *An introduction to genetic algorithms*. [S.l.]: MIT press, 1998.
- PAULINO, A. L. de A. *Neuroevolução de topologias aumentantes com melhorias culturais*. Dissertação (Mestrado) — Universidade Federal do Ceará, Fortaleza, 2018.

- RAO, K. et al. RI-cyclegan: Reinforcement learning aware simulation-to-real. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2020.
- RIDLEY, M. *Evolução*. [S.l.]: Artmed Editora, 2009.
- ROJAS, R. *Neural networks: a systematic introduction*. [S.l.]: Springer Science & Business Media, 2013.
- RUSSELL, S. J. *Artificial intelligence a modern approach*. [S.l.]: Pearson Education, Inc., 2010.
- SAE. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. 2018. Disponível em: https://www.sae.org/standards/content/j3016_201806/. Acesso em: 2022-05-03.
- SAPUTRA, R. P.; RIJANTO, E. Automatic guided vehicles system and its coordination control for containers terminal logistics application. *CoRR*, abs/2104.08331, 2021. Disponível em: <https://arxiv.org/abs/2104.08331>.
- SHARDA, R.; DELEN, D.; TURBAN, E. *Business Intelligence e Análise de Dados para Gestão do Negócio-4*. [S.l.]: Bookman Editora, 2019.
- SUTSKEVER, I. et al. Towards principled unsupervised learning. *arXiv preprint arXiv:1511.06440*, 2015.
- SUTTON, R. S.; BARTO, A. G. *Reinforcement learning: An introduction*. [S.l.]: MIT press, 2018.