

---

# PROJETO 4

---

Anna Clara Ferraz  
cc23306@g.unicamp.br

Vinícius Dos Santos Andrade  
cc22333@g.unicamp.br

## 1 Introdução

Uma **rede neural artificial** é um método de IA (Inteligência Artificial) que busca simular os processos que o cérebro realiza para aprender e tomar decisões com base em padrões. Dentre os diversos tipos de redes neurais artificiais, destaca-se a **Rede Neural Convolucional (ConvNet/CNN)**, que foi aplicada com sucesso pela primeira vez em 1998 por *Yann LeCun*. Essa rede é capaz de identificar informações significativas em imagens, vídeos e outras entradas visuais, diferenciando-as a partir dessas características. A **CNN** foi inspirada nos estudos de *Hubel e Wiesel*, realizados em 1962, que demonstraram que certos neurônios são ativados conjuntamente quando expostos a linhas ou curvas. Assim, a **CNN** reproduz o reconhecimento visual, filtrando linhas e curvas em cada camada para, ao final, reconstruir a imagem analisada. Atualmente, as aplicações da **ConvNet** podem ser observadas nas áreas de marketing, saúde, varejo e automotivo.

A estrutura de uma **Rede Neural Convolucional** é constituída por diversas camadas que podem ser generalizadas como: **Camadas Convolucionais**, **Camadas de Pooling** e **Camadas Fully Connected**. Seu funcionamento consiste em:

### 1.1 Entrada

A rede recebe como entrada uma imagem que é vista como uma matriz de 3 dimensões, sendo a altura e largura determinantes do tamanho da imagem, e a profundidade, que indica os canais de cores **RGB** (abreviatura do sistema de cores aditivas formado pelos tons de Vermelho (Red), Verde (Green) e Azul (Blue)).

## 1.2 Camadas Convolucionais

A imagem passa pelas **camadas convolucionais** que são consideradas o núcleo dessa rede, o que justifica o nome dado a ela. Essas camadas são responsáveis por aplicar **filtros** (também chamados de **Kernels**) a pequenas regiões da entrada, gerando mapas de características. Detectam padrões locais como bordas, texturas e formas na imagem. Esses **Kernels** são aplicados em uma janela deslizante que percorre toda a imagem e aplica o filtro a cada parte dela. Nessas camadas existe a **Rectified Linear Unit**, uma função de ativação que aplica uma transformação não linear, mantendo os valores positivos e zerando os negativos, ajudando a introduzir não-linearidade no modelo, o que permite o aprendizado de relações mais complexas.

## 1.3 Camadas de Pooling

Junto das **camadas Convolucionais**, a imagem é tratada nas **camadas de Pooling**, que reduzem a dimensionalidade dos mapas de características, resumindo a informação e tornando o processo mais eficiente. Existem dois tipos principais de **pooling**: **Max Pooling**, que seleciona o valor máximo em uma região específica do mapa de características; e **Average Pooling**, que calcula a média dos valores em uma região específica.

## 1.4 Camadas Fully Connected

As características obtidas são passadas para as **camadas Fully Connected**, onde o objetivo dessas camadas é combinar as características extraídas pelas **camadas convolucionais** e realizar a classificação final. Também chamadas de **camadas totalmente conectadas**, as **camadas Fully Connected** são mais parecidas com as camadas em redes neurais tradicionais, onde cada neurônio está conectado a todos os neurônios da camada anterior.

## 1.5 Saída

A saída é uma probabilidade associada a cada classe. Como exemplo, pode-se pensar que a entrada foi a foto de um gato e a saída pode indicar uma alta probabilidade de que a imagem seja de um gato.

## 2 Problema proposto

O problema que será apresentado nesse relatório foi proposto pelo professor da matéria de Tópicos em Inteligência Artificial, o qual consiste em completar a implementação das funções `load_data` e `get_model` a fim de treinar e validar uma **Rede Neural Convolucional** com o intuito de fazê-la reconhecer e distinguir sinais de trânsito com base em imagens desses sinais – um desafio inspirado diretamente pela aplicação da **visão computacional** no avanço dos **carros autônomos**.

## 3 Implementação

O desenvolvimento da **Rede Neural Convolucional** foi feito utilizando a linguagem de programação **Python**, junto das bibliotecas **TensorFlow** (usado para a construção, treinamento e avaliação da rede neural), **NumPy** (para o processamento eficiente de dados numéricos), **Pillow** (usada para manipulação e pré-processamento de imagens) e **scikit-learn** (usada para dividir os dados em conjuntos que serão utilizados para o treinamento da rede neural ou para o teste dela). Os dados utilizados para o treinamento e testes foram sinais de trânsito do conjunto de dados **GTSRB** (*German Traffic Sign Recognition Benchmark*), esses dados serão chamados também de "classes" ao longo do relatório.

## 4 Resultados

Os resultados obtidos tanto na fase de treinamento quanto na de validação foram extremamente positivos, como possível constatar nos gráficos abaixo:

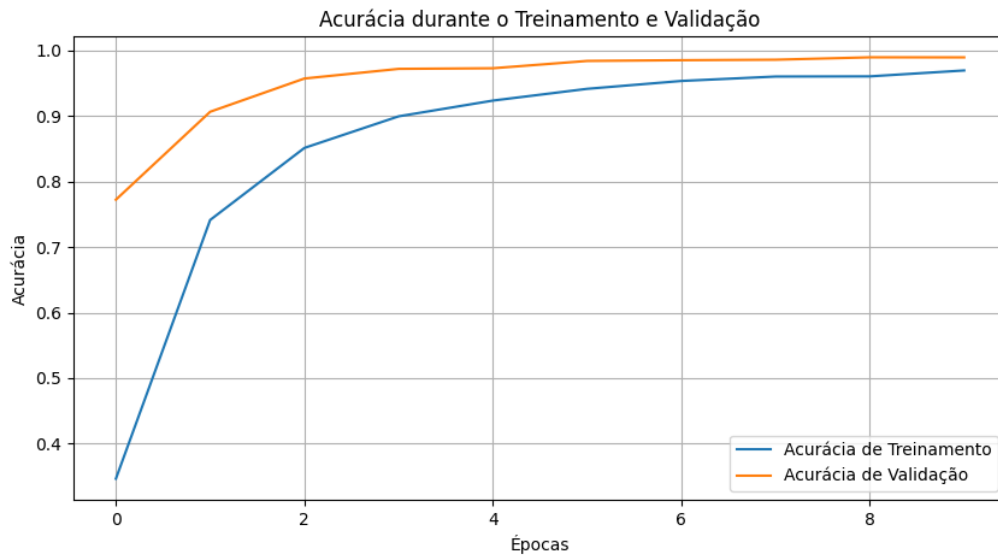


Figura 1: Gráfico ilustrativo da Acurácia durante o Treinamento e Validação

A figura 1 demonstra a acurácia da resposta que a rede neural retornou e a resposta correta, onde uma época sinaliza um ciclo completo de apresentação de todos os dados de treinamento para a rede neural e a acurácia a proximidade entre o valor obtido experimentalmente e o valor verdadeiro, sendo 1.0 equivalente a uma resposta 100% correta. Pode-se observar que o treinamento foi muito bem sucedido pois a partir da 3ª época o nível de proximidade entre a resposta da CNN e da resposta correta está grande e começa a ser um nível de acurácia constante. E a precisão nos testes de validação é ainda melhor, pois já na 2ª época a ConvNet já mostra mais de 90% de precisão em suas respostas.

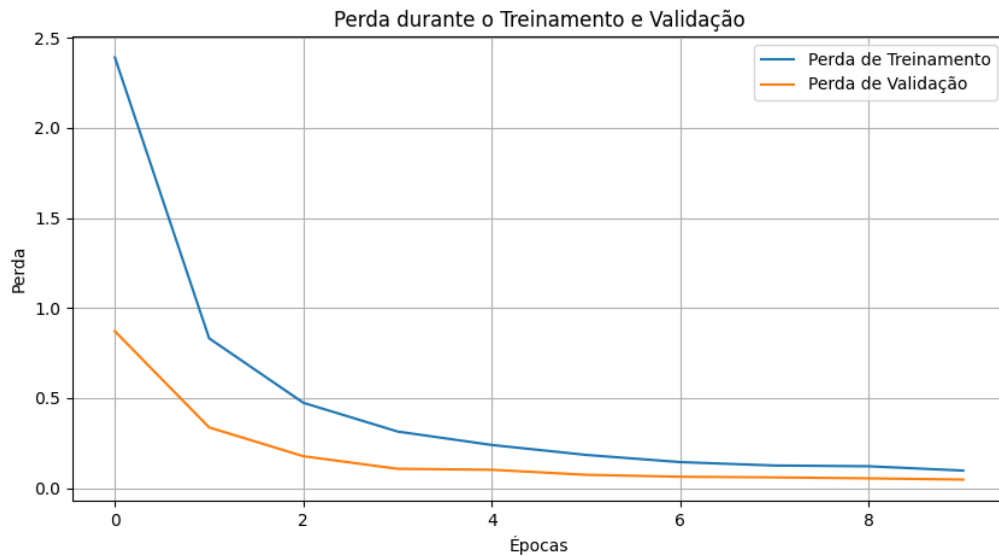


Figura 2: Gráfico ilustrativo da perda durante o treinamento e validação

A figura 2 demonstra a perda, ou seja, a diferença entre as previsões do modelo e os valores reais. Observa-se que nas primeiras épocas a rede apresenta uma grande perda mas que começa a apresentar uma diminuição contínua tanto no treinamento quanto na validação, o que mostra que a rede neural está aprendendo de forma eficaz.

Também é possível visualizar os resultados dos testes de validação da CNN através de uma tabela que possui métricas para cada classe (placa de trânsito) individualmente:

	precision	recall	f1-score	support
<b>Limite de Velocidade (20km/h)</b>	0.99	0.97	0.98	150
<b>Limite de Velocidade (30km/h)</b>	0.99	0.99	0.99	1500
<b>Limite de Velocidade (50km/h)</b>	1.00	0.99	0.99	1500
<b>Limite de Velocidade (60km/h)</b>	0.97	1.00	0.98	960
<b>Limite de Velocidade (70km/h)</b>	0.99	1.00	0.99	1320
<b>Limite de Velocidade (80km/h)</b>	0.99	0.97	0.98	1260
<b>Fim do Limite de Velocidade (80km/h)</b>	1.00	1.00	1.00	300
<b>Limite de Velocidade (100km/h)</b>	1.00	0.98	0.99	960
<b>Limite de Velocidade (120km/h)</b>	0.98	0.99	0.99	960
<b>Proibido Ultrapassar</b>	0.99	0.99	0.99	990
<b>Proibido Ultrapassar Veículos acima de 3.5t</b>	1.00	0.99	1.00	1350
<b>Interseção com Prioridade</b>	1.00	1.00	1.00	900
<b>Estrada Principal</b>	1.00	1.00	1.00	1410
<b>Dê a Preferência</b>	1.00	1.00	1.00	1440

	<b>precision</b>	<b>recall</b>	<b>f1-score</b>	<b>support</b>
<b>Pare</b>	1.00	1.00	1.00	540
<b>Trânsito Proibido</b>	1.00	0.99	0.99	420
<b>Caminhões Proibidos</b>	1.00	1.00	1.00	300
<b>Entrada Proibida</b>	1.00	1.00	1.00	750
<b>Perigo</b>	1.00	1.00	1.00	810
<b>Curva Perigosa à Esquerda</b>	0.98	0.99	0.99	150
<b>Curva Perigosa à Direita</b>	1.00	0.97	0.99	240
<b>Dupla Curva</b>	1.00	0.99	1.00	240
<b>Desnível na Pista</b>	0.99	1.00	0.99	270
<b>Pista Escorregadia</b>	1.00	0.99	1.00	360
<b>Estreitamento de Pista</b>	0.99	1.00	1.00	180
<b>Obras</b>	0.99	1.00	1.00	1020
<b>Semáforo à Frente</b>	1.00	0.99	1.00	420
<b>Pedestres</b>	1.00	0.98	0.99	180
<b>Crianças</b>	0.99	1.00	1.00	360
<b>Ciclistas</b>	1.00	0.98	0.99	180
<b>Perigo de Neve/Gelo</b>	0.99	1.00	1.00	300
<b>Animais Selvagens</b>	1.00	1.00	1.00	540
<b>Fim de Todas as Restrições</b>	0.99	1.00	0.99	180
<b>Vire à Direita</b>	1.00	1.00	1.00	480
<b>Vire à Esquerda</b>	0.99	1.00	1.00	300
<b>Siga em Frente</b>	0.99	1.00	1.00	810
<b>Em Frente ou à Direita</b>	1.00	1.00	1.00	270
<b>Em Frente ou à Esquerda</b>	1.00	0.99	0.99	150
<b>Mantenha-se à Direita</b>	1.00	1.00	1.00	1380
<b>Mantenha-se à Esquerda</b>	1.00	1.00	1.00	210
<b>Rotatória Obrigatória</b>	0.99	1.00	0.99	240
<b>Fim da Proibição de Ultrapassar</b>	1.00	0.98	0.99	180
<b>Fim da Proibição de Ultrapassar Caminhões</b>	0.99	1.00	0.99	180
<b>accuracy</b>	–	–	<b>0.99</b>	<b>26640</b>
<b>macro avg</b>	<b>0.99</b>	<b>0.99</b>	<b>0.99</b>	<b>26640</b>
<b>weighted avg</b>	<b>0.99</b>	<b>0.99</b>	<b>0.99</b>	<b>26640</b>

A métrica **precision** indica a proporção de verdadeiros positivos em relação ao total de positivos previstos, **recall** é a proporção de verdadeiros positivos em relação ao total de positivos reais, o **f1-score** indica a média harmônica entre precisão e recall, fornecendo uma

[illegible]

Na matriz de confusão ambos os eixos representam classes, onde no eixo Y (vertical) tem-se as classes verdadeiras e no eixo X (horizontal) as classes previstas pela CNN. Cada célula (i, j) representa o número de vezes que a classe verdadeira  $i$  foi prevista como classe  $j$ , o que faz com que a diagonal indique os acertos enquanto as células fora dessa diagonal indicam erros

de classificação. Como observado na Matriz de Confusão apresentada, os valores na diagonal principal estão altos, o que significa que o modelo está classificando corretamente os dados com uma alta frequência.



## 5 Conclusão

A implementação da Rede Neural Convolucional para a classificação de sinais de trânsito apresentou resultados altamente satisfatórios, atingindo e superando os objetivos propostos. Os dados demonstram que, a partir da terceira época, a acurácia de treinamento estabilizou-se em níveis elevados, indicando um aprendizado consistente dos padrões presentes nos sinais. Além disso, a precisão obtida nos testes de validação ultrapassou 90% já na segunda época, evidenciando a eficácia e a capacidade de generalização do modelo desenvolvido.

Esses resultados reforçam o potencial das redes neurais convolucionais em aplicações críticas como a identificação de sinais de trânsito, essencial para o avanço de veículos autônomos. O alto desempenho alcançado em classes fundamentais, como "Pare" e "Dê a Preferência", destaca a robustez do modelo e sua relevância para a segurança viária. Os insights derivados das análises confirmam que a combinação adequada de técnicas de pré-processamento, arquitetura de rede e seleção de hiperparâmetros é crucial para o sucesso em tarefas de visão computacional. O projeto abre caminho para futuras pesquisas visando aprimorar ainda mais a precisão e a eficiência desses sistemas.