Universidade la salle

Natanael Ferreira

Canoas, 2025

Trabalho IA – Treinando uma CNN para reconhecer letras de A-Z em estilo Handwritted

Whitepaper - Machine Learning with Kaggle Database A-Z

**Recursos de entrada e recursos gerados:**

* Para treinamento: Base de dados de Imagens já convertidos em Vetorial (Pode ser encontrado no seguinte link: [*https://www.kaggle.com/datasets/sachinpatel21/az-handwritten-alphabets-in-csv-format*](https://www.kaggle.com/datasets/sachinpatel21/az-handwritten-alphabets-in-csv-format))
* Conjunto de imagens 28x28px em escala de cinza (pode ser encontrado no seguinte link: <https://medium.com/analytics-vidhya/optical-character-recognition-using-tensorflow-533061285dd3>)
* Código para converter a imagem completa em pequenas imagens 28x28 com apenas um caractere handwritted (“separateImages.py”)
* Código para testar o modelo (“TestarOModelo.py”)
* Código que implementa e roda o modelo de Deep Learning (“\_\_main\_\_.py”)
* Modelo gerado pelo código (“modelo\_letras.keras”)
* Referência de performance do código, acurácia por época (“./results”)

**Descrição completa do código (\_\_main\_\_.py):**

1-16:Cabeçalho e importação de bibliotecas

22: Carrega a base de dados

25:Define a coluna que representará as letras. Ex: 0=A, 1=B...

26:DATASET.drop exclui somente a coluna 0 e mantém os demais para vetores de letra

29:Normalização, para evitar convergência demorada ou que não aprender, também evita explosão de valores que crescem demais, como IA trabalho com Pesos normalizar é ideal

**Texto

O conteúdo gerado por IA pode estar incorreto.**

33: Precisa ser convertido novamente neste formato pois Redes Convolucionais – CNNS não funcionam com vetores planos, usam estrutura espacial com altura, largura e profundidade (canais), 1 canal pois é grayscale. CNNs aplicam kernels sobre a imagem.

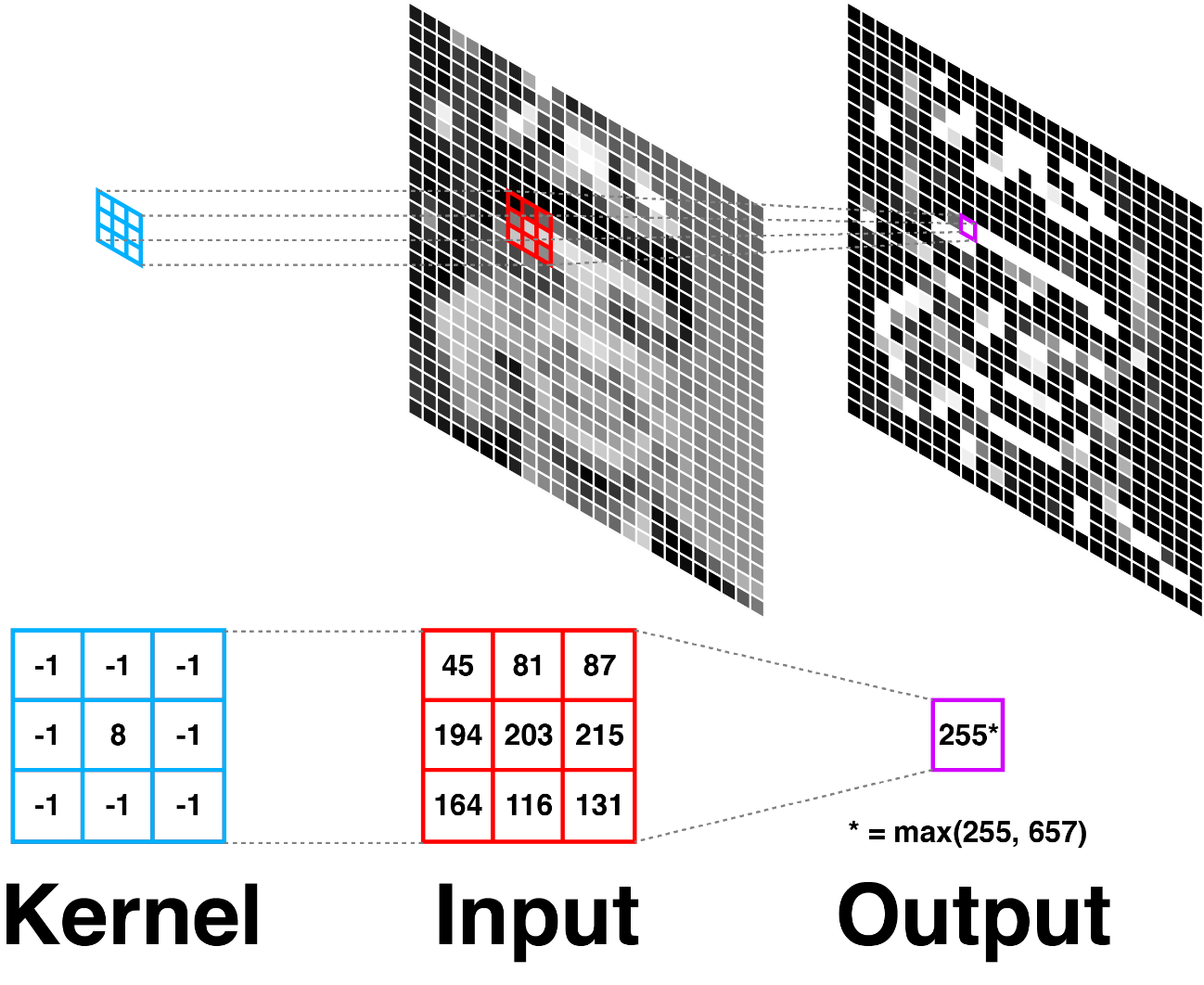


Figura 1 - <https://gregorygundersen.com/image/cnns/41847277995.png>

35: Transforma os rótulos (y) — que originalmente são **números inteiros representando letras** — em **vetores one-hot**.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Figura 2 -Gerado por IA - Chat GPT

Pela função de saída da rede ser Softmax ela espera produzir um vetor com 26 probabilidades: [0.01, 0.00, 0.05, ..., 0.91, ..., 0.00]

Tela de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

Figura 3 - Gerado por IA - ChatGPT

37:Separa a porcentagem de dados de treino(80%) e de teste(20%), para que a rede treine com dados não vistos ainda.

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

51:Cria o modelo que será usado para treinar

O que cada atributo da configuração faz:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Etapa** | **Função** | **Entrada** | **Saída** | **Descrição** |
| **Conv2D(32, 3x3)** | Detector de bordas e texturas | (28, 28, 1) | (26, 26, 32) | Extrai padrões locais |
| **MaxPooling2D(2x2)** | Compressão e destaque de padrões fortes | (26, 26, 32) | (13, 13, 32) | Reduz complexidade e mantém o essencial |
| **Conv2D(64, 3x3)** | Captura de padrões mais abstratos | (13, 13, 32) | (11, 11, 64) | Reconhece formas mais complexas |
| **MaxPooling2D(2x2)** | Nova compressão | (11, 11, 64) | (5, 5, 64) | Garante invariância espacial |
| **Flatten()** | Prepara para camadas densas | (5, 5, 64) = 1600 | vetor(1600) | Transforma imagem em vetor |
| **Dense(128)** | Classificador parcial | (1600) | (128) | Combina padrões extraídos |
| **Dropout(0.5)** | Regularização | (128) | (128) | Evita overfitting desativando alguns perceptrons |
| **Dense(26, softmax)** | Classificador final | (128) | (26) | Gera probabilidades para cada letra (A–Z) |

Texto

O conteúdo gerado por IA pode estar incorreto.

65: ADAM - Adaptive Moment Estimation, Imagine que o modelo previu errado a letra "A" como "C", o modelo ajusta todos os pesos conectados ao padrão da letra "A" para tentar diminuir esse erro na próxima época, de forma adaptativa e inteligente

66: A função de perda (loss) mede o quão errado o modelo está nas previsões. O treinamento tenta minimizar esse valor

67: Mede a porcentagem de acertos (predição com maior probabilidade bate com o índice da label real)

Texto

O conteúdo gerado por IA pode estar incorreto.

70-75:Inicia o treinamento da CNN, define a quantidade de épocas, define os dados em lotes de 128 amostras/imagens para treinar por etapa e define os dados que serão usados para medir a performance de cada época.

Texto

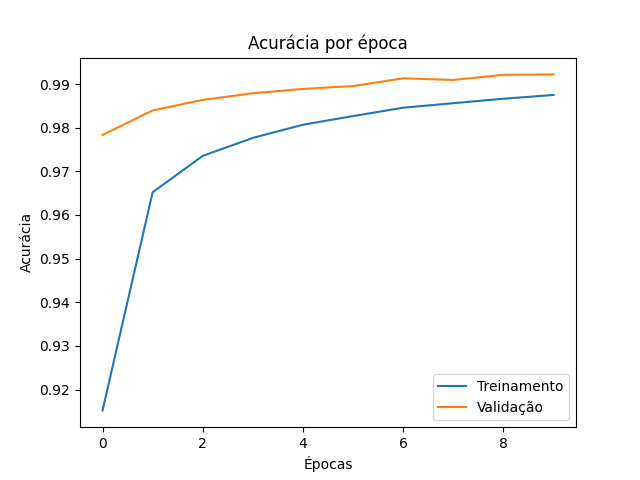
O conteúdo gerado por IA pode estar incorreto.

78:Salva o modelo em formato .keras



82-88:Gera um gráfico apenas informativo para mostrar a acurácia do treinamento.

**Resultado:**



Texto

O conteúdo gerado por IA pode estar incorreto.

91-92: Executa uma predição completa sobre todos os dados **X\_test,** depois compara as predições com os rótulos verdadeiros **y\_test**.



**Descrição completa do código(TestarOModelo.py):**

Overview Geral:

O código busca uma imagem armazenada na pasta “./ImageToTest” de tamanho 28x28 pixels com o nome “imageTest.png” converte em formato vetorial seguindo o padrão da base dados e a partir do modelo gerado, faz a predição, converte o resultado para letra e gera uma saída da letra predita.

**Como treinar o modelo:**

1. **Baixe a base de dados nesse link:** *https://www.kaggle.com/datasets/sachinpatel21/az-handwritten-alphabets-in-csv-format*
2. Rode o código

**Como testar o modelo:**

1. Coloque uma imagem com 28x28 pixels, em formato png dentro da pasta “./ImageToTest/ imageTest.png” e rode o código.
2. Há uma série de imagens para treinamento na pasta “./dataset”