

Relatório Final - Reconhecimento de Dígitos Numéricos com Keras e MNIST

Este relatório descreve o desenvolvimento de um sistema de reconhecimento de dígitos manuscritos (0 a 9) usando redes neurais com Keras e o dataset MNIST. O foco é apresentar o funcionamento dos códigos de treinamento e predição, e as instruções para sua execução.

1. Objetivo

O projeto visa criar e treinar um modelo de rede neural para reconhecer dígitos numéricos de 0 a 9 a partir de imagens, utilizando o dataset MNIST.

2. Funcionamento Detalhado do Código

O projeto é composto por dois scripts Python 3 na pasta `src/`: `treino_mnist.py` e `prever_mnist.py`, ambos baseados em `tensorflow.keras`.

2.1. `treino_mnist.py` - Treinamento do Modelo

Este script prepara os dados e treina o modelo de Deep Learning:

- **Configuração e Dados:** Parâmetros como épocas, tamanho do lote e caminho do modelo (`models/modelo_mnist_custom.h5`) são definidos. O dataset MNIST é carregado, e as imagens (28x28 pixels) são **normalizadas** para o intervalo `[0, 1]`. Os rótulos são convertidos para **one-hot encoding**.
- **Visualização de Dados:** Cinco **imagens de treino aleatórias** do MNIST são exibidas para demonstrar a diversidade dos dados de entrada.
- **Arquitetura da Rede Neural:** O modelo é uma rede sequencial com camadas densas (`Dense`).
 - **Flatten:** Converte a imagem 2D em um vetor 1D.
 - Duas camadas `Dense` (256 e 128 neurônios, ativação **ReLU**) processam as características.
 - Duas camadas de **Dropout (20%)** são incluídas após as camadas densas para prevenir **overfitting**, desativando aleatoriamente neurônios durante o treino.
 - A camada de saída tem 10 neurônios (um para cada dígito) com ativação **softmax**, que produz probabilidades para cada classe.
- **Compilação e Treinamento Inteligente:**

- O modelo é compilado com otimizador `adam` e função de perda `categorical_crossentropy`.
- **EarlyStopping**: Interrompe o treino se a acurácia de validação não melhorar por 5 épocas, economizando tempo e evitando overfitting.
- **ModelCheckpoint**: Salva automaticamente a melhor versão do modelo (com a maior acurácia de validação) no formato `.h5` na pasta `models/`.
- **Avaliação e Gráficos**: Após o treino, o modelo é avaliado no conjunto de teste. Gráficos de acurácia e perda (treino e validação) são exibidos para análise do desempenho.

2.2. `prever_mnist.py` - Reconhecimento de Dígitos

Este script usa o modelo treinado para fazer previsões:

- **Carregamento do Modelo**: O `modelo_mnist_custom.h5` é carregado da pasta `models/`. Tratamento de erros verifica sua existência.
- **Seleção de Imagem**: O script seleciona uma **imagem de teste aleatória** do MNIST. Uma função auxiliar pré-processa imagens externas (redimensiona para 28x28, converte para tons de cinza e normaliza).
- **Predição e Resultados**: A imagem é passada para o modelo, que retorna um vetor de probabilidades. O dígito previsto é o de maior probabilidade (`np.argmax()`), e a **confiança da predição** é calculada. A imagem é exibida com a predição visual, e os resultados detalhados (rótulo real, previsão, confiança e probabilidades por dígito) são impressos no terminal.

3. Instruções para Executar o Treinamento

1. **Clone o Repositório**: `git clone https://github.com/HenriqueDC2003/DeepLearning_Digitos_Numericos.git` e `cd DeepLearning_Digitos_Numericos`.
 2. **Crie e Ative o Ambiente Virtual (`.venv`)**:
 - `python -m venv .venv`
 - Windows: `.\.venv\Scripts\activate`
 - macOS/Linux: `source ./venv/bin/activate`
 3. **Instale as Dependências**: `pip install -r requirements.txt` (com o `.venv` ativado).
 4. **Execute o Script**: Vá para `cd src` e execute `python treino_mnist.py`. O modelo `modelo_mnist_custom.h5` será salvo em `models/`.
-

4. Instruções para Realizar o Reconhecimento

1. **Pré-requisitos:** Modelo `models/modelo_mnist_custom.h5` deve existir.
2. **Ative o Ambiente Virtual:** Se não estiver ativo, ative-o (ver Seção 3).
3. **Execute o Script:** Vá para `cd src` e execute `python prever_mnist.py`. Uma imagem aleatória do MNIST será exibida com sua predição.