

Model Pruning

Dimensionreduzierung von CNNs für die Bildverarbeitung

Dieses Projekt wurde im Rahmen des Automatisierungstechnischen Projekts der TU-Berlin am Fachgebiet für Industrielle Automatisierungstechnik erstellt. Dabei ging es darum die Dimensionen von KI-Modellen durch Model Pruning zu reduzieren, um die Effizienz und Geschwindigkeit der Modelle zu verbessern, ohne dabei signifikant die Genauigkeit einzubüßen.

Das Ergebnis des Projekts ist ein Interface, mit dem die Taylor-Pruning Methode getestet werden kann. Das Pruning kann bei CNN-Modellen im .pth-Format erfolgen. Ein beispielhaftes ResNet-Modell ist im Ordner **Models** hinterlegt.

Inhaltsverzeichnis

- [Installation](#)
- [Benutzung](#)
- [To-Dos](#)
- [Lizenz](#)

Installation

Vor den nachfolgenden Installationsanleitungen zunächst dieses Repository klonen.

```
# Klonen des Repositorys
git clone https://github.com/rabeaifeanyi/Model-Pruning.git

# In das Projektverzeichnis wechseln
cd Model-Pruning
```

Die App kann auf drei Arten installiert werden: Docker, Conda oder Virtual Enviropnment.

Installation mit Docker

```
# Docker-Image bauen
docker build -t model-pruning .
```

Installation mit Conda-Environment (empfohlen)

Siehe [Conda Cheat Sheet](#)

```
# Environment erstellen
conda env create -f environment.yml
```

```
# Environment aktivieren  
conda activate model-pruning
```

Alternative manuelle Installation

Es wird Python 3.8 oder höher benötigt.

```
# (Optional) Virtuelle Umgebung erstellen und aktivieren  
python -m venv venv  
source venv/bin/activate # Für Unix oder MacOS  
venv\Scripts\activate # Für Windows  
  
# Requirements installieren  
pip install -r requirements.txt
```

Benutzung

Vorbereitung

1. Überprüfen, ob die Installation erfolgreich war.
2. In das Verzeichnis *Model-Pruning* navigieren.

Start der Anwendung und Auswählen des Modells

1. Anwendung wie folgt starten.

```
# MIT DOCKER  
# Starten des Docker Containers, wenn kein eigenes Modell vorhanden ist  
docker run --gpus all -p 8080:8080 model-pruning  
  
# Starten des Containers mit eigenem Modell  
docker run --gpus all -p 8080:8080 -v <Modell-Pfad>:/app/Models/<Name vom  
Modell>.pth -d model-pruning #Pfad und Namen anpassen  
  
# ANSONSTEN  
# App mit Streamlit starten  
streamlit run main.py
```

1. <http://localhost:8080> öffnen, sollte es sich nicht automatisch öffnen.
Folgendes Interface sollte sichtbar werden:

CNN Pruning

Welcome to the Neural Network Pruning dashboard! This interactive tool allows you to explore various pruning methods to optimize your machine learning models.

► More information.

Configuration

Selection of Model and Dataset

Select a model

Upload from files

Please enter path to model

Select a dataset

MNIST

2. Unter *Select a model* muss jetzt **Upload from files** ausgewählt werden und der Pfad zum Modell angegeben werden.

- **Docker**

`/Models/<Name vom Modell>.pth`

Unser Beispiel-Modell heißt `resnet_model_mnist.pth`.

- **Ansonsten**

Lokalen Pfad zum Modell angeben.

Beispiel-Modell: `<Dir-Pfad>/Models/resnet_model_mnist.pth`

3. Pfad zum Datensatz angeben.

Beispiel-Modell: MNIST auswählen.

Konfiguration der Pruning-Parameter

1. Pruning-Methode auswählen.

Siehe To-Dos, bisher ist nur die Taylor-Pruning-Methode fertig implementiert!

2. Anzahl der Epochen für das Training angeben.

3. Sparsity in Prozent festlegen.

4. Gegebenenfalls die Bildgröße für die Datensätze festlegen.

Achtung! Beim Beispiel-Modell Bildgröße 224 angeben.

5. Auf **Run** klicken.

Select a dataset

MNIST



Available Pruning methods

☒ Taylor Expansion-based Filter Pruning

Epochs

Enter the number of epochs

3



Sparsity

Enter the amount of parameters to prune (in %)

50



Data size

Enter the number of pixels for height and width of the datasets or leave empty to use the default imagesize

224



Run

Ergebnisse und Evaluation

1. Wenn alles funktioniert hat, sollte der Pruning Prozess starten. Das kann einige Minuten bis mehrere Stunden dauern.
2. Nach dem Laden sollten die Ergebnisse sichtbar sein.
3. Die geprunten Modelle werden im Verzeichnis *Models_pruned* gespeichert.

To-Dos

Dringend

- ☐ Docker: Download des geprunten Modells ermöglichen
- ☐ Falsche Pfade auffangen

Aktuell

- ☐ Testen der Pipeline mit fremden Modellen

- ☐ Installationsanleitungen testen
- ☐ Beispiel Projekt abrufbar machen
- ☐ Modelle, die im Models-Ordner sind sollen automatisch erkannt werden, anstatt nur als vom User eingegebenen Pfad
- ☒ Anleitung für die Benutzung fertigstellen
- ☐ Implementierung und Integration der APoZ-Pruning-Methode
- ☐ Herausfinden warum die PyTorch-Pruning-Methoden nicht funktionieren

Ideen für die Zukunft

- ☐ Speichern von Zwischenständen und geprunten Modellen, so dass bessere Vergleichbarkeit möglich ist, dafür Streamlit Session States integrieren. Bisher muss dafür der Pfad vom bereits geprunten Modell manuell eingetippt werden.
- ☐ Falls der Rechenvorgang abbricht oder die Seite neu-lädt, sollte nicht alles erneut berechnet werden müssen, mit Cache arbeiten oder Streamlit Session States einarbeiten

Lizenz

Copyright (c) 2023

TU Berlin, Institut für Werkzeugmaschinen und Fabrikbetrieb Fachgebiet Industrielle Automatisierungstechnik

Authors: Leandro Carrión Benenwart, Rabea Eschenhagen, Robert Komorowski, Sedat Süzer, Tom Wolf

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

DISCLAIMER

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.