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

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

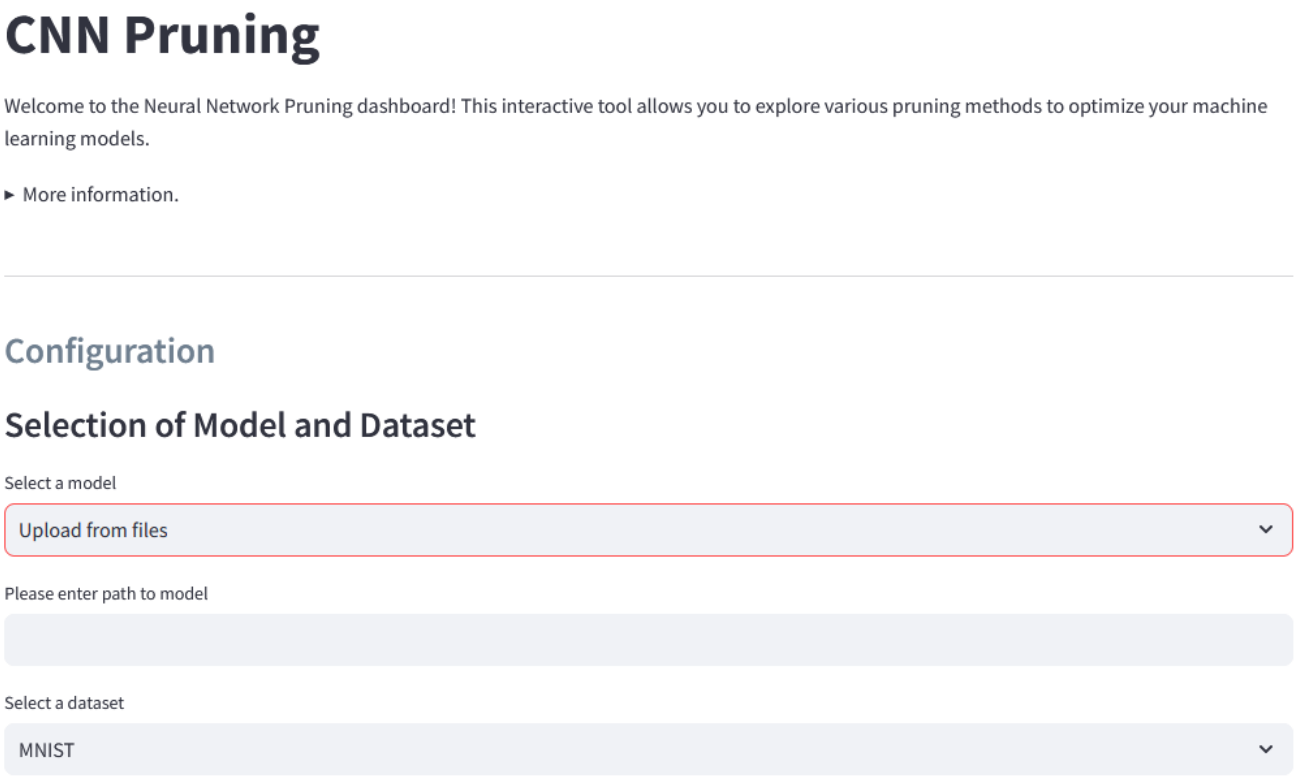
1. Anwendung wie folgt starten.

```
# 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
```

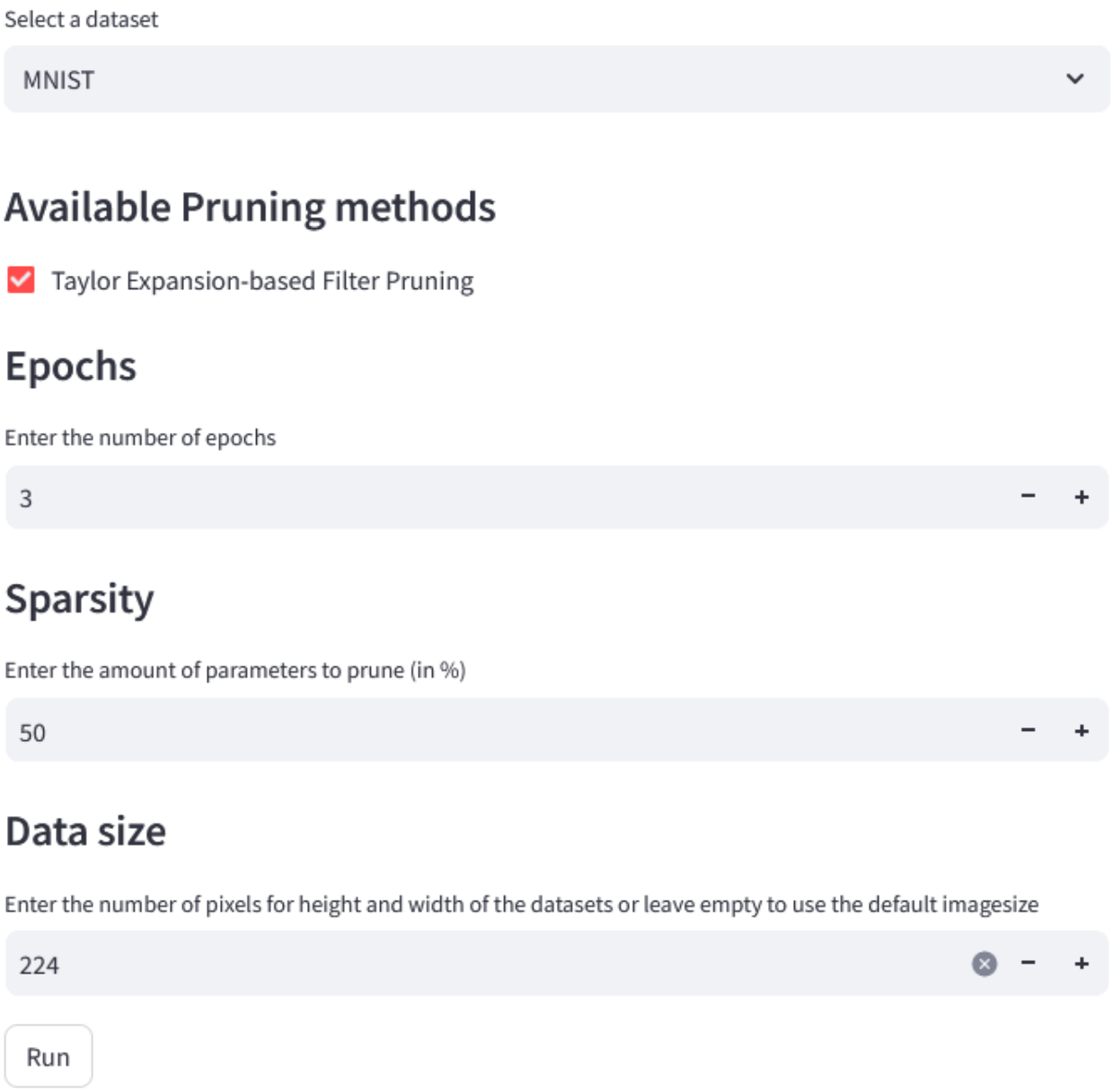
1. http://localhost:8080 öffnen, sollte es sich nicht automatisch öffnen.
   Folgendes Interface sollte sichbar werden:



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.



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
- [x] Anleitung für die Benutzung fertigstellen
- [ ] Implementierung und Integration der APoZ-Pruning-Methode
- [ ] Herausfinden warum die PyTorch-Pruning-Methoden nicht funktionieren

Ideen für die Zukuft

- [ ] 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