Malakh Trainer - technická dokumentácia

Generované programom Doxygen 1.10.0

Kapitola 1

Register hierarchie tried

1.1 Hierarchia tried

Tu nájdete zoznam, vyjadrujúci vzťah dedičnosti tried. Je zoradený približne (ale nie úplne) podľa abecedy:

Kapitola 2

Register tried

2.1 Zoznam tried

Nasledujúci zoznam obsahuje predovšetkým identifikáciu tried, ale nachádzajú sa tu i ďalšie netriviálne prvky, ako sú štruktúry (struct), uniony (union) a rozhrania (interface). V zozname sú uvedené ich stručné popisy:

Kapitola 3

Register súborov

3.1 Zoznam súborov

Tu nájdete zoznam všetkých dokumentovaných súborov so stručnými popismi:

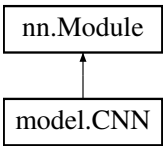
Kapitola 4

Dokumentácia tried

4.1 Dokumentácia triedy model.CNN

Architektúra konvolučnej neurónovej siete.

Diagram dedičnosti pre triedu model.CNN



Verejné metódy

- __init__ (self, conv_layers, conv_norm, conv_activ, fc_layers, fc_activ)

  Konštruktor konvolučnej neurónovej siete.
- forward (self, x)

  Predná propagácia cez neurónovú sieť.

Verejné atribúty

- **conv**
- **fc_input**
- **fc**
- **output**

4.1.1 Detailný popis

Architektúra konvolučnej neurónovej siete.

4.1.2 Dokumentácia konštruktoru a deštruktoru

4.1.2.1 __init__()

```
model.CNN.__init__ (
              self,
              conv_layers,
              conv_norm,
              conv_activ,
              fc_layers,
              fc_activ )
```

Konštruktor konvolučnej neurónovej siete.

Parametre

conv_layers	Počet konvolučných vrstiev.
conv_activ	Aktivačná vrstva konvolučných vrstiev.
fc_layers	Počet plne prepojených vrstiev.
fc_activ	Aktivačná vrstva plne prepojených vrstiev.

4.1.3 Dokumentácia k metódam

4.1.3.1 forward()

```
model.CNN.forward (
              self,
              x )
```

Predná propagácia cez neurónovú sieť.

Parametre

x	Vstup na ktorom vykonávame predikciu.

Návratová hodnota

Predikovaná hodnota.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- model.py

4.2 Dokumentácia triedy datamodule.DataModuleBitboards

Dátový modul pre bitboardovú formu MvsFS datasetu.

Verejné metódy

- __init__ (self, filename, ratio, batch_size)

  Konštruktor dátového module bitboardovej formy MvsFS datasetu.

Verejné atribúty

- **ratio**
- **train_dataset**
- **val_dataset**
- **test_dataset**
- **batch_size**
- **train_loader**
- **val_loader**
- **test_loader**

4.2.1 Detailný popis

Dátový modul pre bitboardovú formu MvsFS datasetu.

4.2.2 Dokumentácia konštruktoru a deštruktoru

4.2.2.1 __init__()

```
datamodule.DataModuleBitboards.__init__ (
              self,
              filename,
              ratio,
              batch_size )
```

Konštruktor dátového module bitboardovej formy MvsFS datasetu.

Parametre

filename	Názov vstupného csv súboru.
ratio	Pomer medzi trénovacími a validačnými dátami (validačné a testovacie dáta majú pomer 0.5).
batch_size	Veĺkosť dávky pri načítavaní dát.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- datamodule.py

4.3 Dokumentácia triedy datamodule.DataModuleImages

Dátový modul pre obrázkovú formu MvsFS datasetu.

Verejné metódy

- __init__ (self, filename, ratio, batch_size)

  Konštruktor dátového module obrázkovej formy MvsFS datasetu.

Verejné atribúty

- **ratio**
- **train_dataset**
- **val_dataset**
- **test_dataset**
- **batch_size**
- **train_loader**
- **val_loader**
- **test_loader**

4.3.1 Detailný popis

Dátový modul pre obrázkovú formu MvsFS datasetu.

4.3.2 Dokumentácia konštruktoru a deštruktoru

4.3.2.1 __init__()

```
datamodule.DataModuleImages.__init__ (
              self,
              filename,
              ratio,
              batch_size )
```

Konštruktor dátového module obrázkovej formy MvsFS datasetu.

Parametre

filename	Názov vstupného csv súboru.
ratio	Pomer medzi trénovacími a validačnými dátami (validačné a testovacie dáta majú pomer 0.5).
batch_size	Veĺkosť dávky pri načítavaní dát.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- datamodule.py

4.4 Dokumentácia triedy trainer.EarlyStopper

Trieda zodpovedná za predčasné zastavenie trénovania.

Verejné metódy

- __init__ (self, patience=1, min_delta=0)
- early_stop (self, val_loss)

  V tejto funkcii sa kontroluje, či aktuálna validačná strata je lepšia ako najlepší možný výsledok.

Verejné atribúty

- **patience**
- **min_delta**
- **counter**
- **min_val_loss**

4.4.1 Detailný popis

Trieda zodpovedná za predčasné zastavenie trénovania.

4.4.2 Dokumentácia konštruktoru a deštruktoru

4.4.2.1 __init__()

```
trainer.EarlyStopper.__init__ (
              self,
              patience = 1,
              min_delta = 0 )
```

```
Konštruktor predčasného ukončovatel'a trénovania.

@param patience: Počet epoch, ktoré je možné vykonat' po sebe bez vylepšenia výsledkov trénovania.
@param min_delta: Minimálny rozdiel medzi najlepším výsledkom a aktuálnym výsledkom potrebný aby sa epocha pov
```

4.4.3 Dokumentácia k metódam

4.4.3.1 early_stop()

```
trainer.EarlyStopper.early_stop (
              self,
              val_loss )
```

V tejto funkcii sa kontroluje, či aktuálna validačná strata je lepšia ako najlepší možný výsledok.

Ak áno, tak sa počítadlo trpezlivosti reštartuje na 0. Ak nie a chyby je vyššia od najlepšieho výsledku o min_delta hodnotu, tak sa inkrementuje počítadlo trpezlivosti. Ak počítadlo trpezlivosti dosiahlo maximálnu hodnotu, tak sa ukončí trénovanie.

Parametre

val_loss	Validačnú strata aktuálnej epochy.

Návratová hodnota

True ak sa trénovanie má predčasne ukončiť, False ak trénovanie má pokračovať.

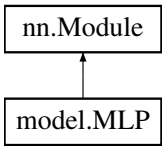
Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- trainer.py

4.5 Dokumentácia triedy model.MLP

Architektúra plne prepojenej neurónovej siete.

Diagram dedičnosti pre triedu model.MLP



Verejné metódy

- __init__ (self, input_features, hidden_features, output_features, layers, activ)
  Konštruktor plne prepojenej neurónovej siete.
- forward (self, x)
  Predná propagácia cez neurónovú sieť.

Verejné atribúty

- **input**
- **hidden**
- **output**

4.5.1 Detailný popis

Architektúra plne prepojenej neurónovej siete.

4.5.2 Dokumentácia konštruktoru a deštruktoru

4.5.2.1 __init__()

```
model.MLP.__init__ (
              self,
              input_features,
              hidden_features,
              output_features,
              layers,
              activ )
```

Konštruktor plne prepojenej neurónovej siete.

Parametre

input_features	Počet neurónov vstupnej vrstvy neurónovej siete.
hidden_features	Počet neurónov skrytej vrstvy neurónovej siete.
output_features	Počet neurónov výstupnej vrstvy neurónovej siete.
layers	Počet skrytých vrstiev.
activ	Aktivačná funkcia neurónovej siete.

4.5.3 Dokumentácia k metódam

4.5.3.1 forward()

```
model.MLP.forward (
              self,
              x )
```

Predná propagácia cez neurónovú sieť.

Parametre

x	Vstup na ktorom vykonávame predikciu.

Návratová hodnota

Predikovaná hodnota.

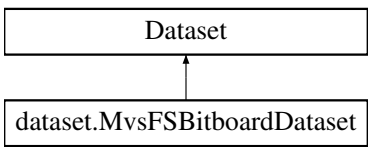
Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- model.py

4.6 Dokumentácia triedy dataset.MvsFSBitboardDataset

Dataset MvsFS v bitboardovej forme pre MLP architektúru.

Diagram dedičnosti pre triedu dataset.MvsFSBitboardDataset



Verejné metódy

- __init__ (self, data)

  Konštruktor bitboardovej formy MvsFS datasetu.
- __len__ (self)

  Interná funkcia na výpočet veľkosti datasetu.
- __getitem__ (self, idx)

  Interná funkcia na výber položky datasetu.

Verejné atribúty

- **data**

4.6.1 Detailný popis

Dataset MvsFS v bitboardovej forme pre MLP architektúru.

4.6.2 Dokumentácia konštruktoru a deštruktoru

4.6.2.1 __init__()

```
dataset.MvsFSBitboardDataset.__init__ (
              self,
              data )
```

Konštruktor bitboardovej formy MvsFS datasetu.

Parametre

data	Dáta MvsFS datasetu.

4.6.3 Dokumentácia k metódam

4.6.3.1 __getitem__()

```
dataset.MvsFSBitboardDataset.__getitem__ (
              self,
              idx )
```

Interná funkcia na výber položky datasetu.

Parametre

idx	Identifikátor položky.

Návratová hodnota

Vstupné vlastnosti a ich label.

4.6.3.2 __len__()

```
dataset.MvsFSBitboardDataset.__len__ (
              self )
```

Interná funkcia na výpočet veľkosti datasetu.

Návratová hodnota

Veľkosť datasetu

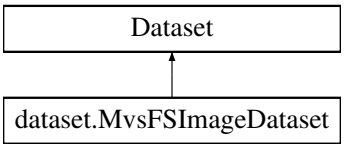
Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- dataset.py

4.7 Dokumentácia triedy dataset.MvsFSImageDataset

Dataset MvsFS v obrázkovej forme pre CNN architektúru-.

Diagram dedičnosti pre triedu dataset.MvsFSImageDataset



Verejné metódy

- __init__ (self, data)
  Konštruktor obrázkovej formy MvsFS datasetu.
- __len__ (self)
  Interná funkcia na výpočet veľkosti datasetu.
- __getitem__ (self, idx)
  Interná funkcia na výber položky datasetu.

Verejné atribúty

- **data**

4.7.1 Detailný popis

Dataset MvsFS v obrázkovej forme pre CNN architektúru-.

4.7.2 Dokumentácia konštruktoru a deštruktoru

4.7.2.1 __init__()

```
dataset.MvsFSImageDataset.__init__ (
            self,
            data )
```

Konštruktor obrázkovej formy MvsFS datasetu.

Parametre

data	Dáta MvsFS datasetu.

4.7.3 Dokumentácia k metódam

4.7.3.1 __getitem__()

```
dataset.MvsFSImageDataset.__getitem__ (
              self,
              idx )
```

Interná funkcia na výber položky datasetu.

Parametre

idx	Identifikátor položky.

Návratová hodnota

Vstupné vlastnosti a ich label.

4.7.3.2 __len__()

```
dataset.MvsFSImageDataset.__len__ (
              self )
```

Interná funkcia na výpočet veľkosti datasetu.

Návratová hodnota

Veľkosť datasetu

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- dataset.py

4.8 Dokumentácia triedy trainer.Trainer

Trieda zodpovedná za trénovanie neurónových sietí.

Verejné metódy

- __init__ (self, datamodule, model, input_shape, criterion, optimizer, patience, min_delta, max_epoch, output_dir)

  Konštruktor konfigurovateľného trénera neurónových sietí.
- fit (self, checkpoint=None)

  Táto funkcia spustí trénovanie neurónovej siete.
- train_epoch (self, epoch)

  Spustenie trénovacej epochy.
- val_epoch (self, epoch)

  Spustenie validačnej epochy.
- test_epoch (self)

  Spustenie testovacej epochy.
- epoch (self, dataloader, training, caption)

  Generická metóda pre všetky epochy.
- load_checkpoint (self, filename)

  Načítanie priebežných výsledkov trénovania z checkpointu.
- save_checkpoint (self, filename)

  Uloženie priebežných výsledkov trénovania do checkpointu.
- save_model (self)

  Uloženie výsledného modelu do jit formátu.
- save_plot (self, filename, caption, metric_name, train_values, val_values)

  Uloženie trénovacích a validačných metrík do grafu po ukončení trénovania.

Verejné atribúty

- **device**
- **datamodule**
- **model**
- **input_shape**
- **criterion**
- **optimizer**
- **early_stopper**
- **max_epoch**
- **output_dir**
- **train_losses**
- **val_losses**
- **cur_epoch**

4.8.1 Detailný popis

Trieda zodpovedná za trénovanie neurónových sietí.

4.8.2 Dokumentácia konštruktoru a deštruktoru

4.8.2.1 __init__()

```
trainer.Trainer.__init__ (
              self,
              datamodule,
              model,
              input_shape,
              criterion,
              optimizer,
              patience,
              min_delta,
              max_epoch,
              output_dir )
```

Konštruktor konfigurovateľného trénera neurónových sietí.

Parametre

datamodule	Dátový modul obsahujúci MvsFS dataset.
model	Architektúra neurónovej siete.
input_shape	Tvar vstupu do neurónovej siete.
criterion	Loss funkcia.
optimizer	Optimalizačný algoritmus trénovania.
patience	Počet epoch, ktoré je možné vykonať po sebe bez vylepšenia výsledkov trénovania.
min_delta	Minimálny rozdiel medzi najlepším výsledkom a aktuálnym výsledkom potrebný aby sa epocha považovala za neúspešnú.
max_epoch	Maximálny počet vykonaných epoch.
output_dir	Priečinok kde sa uložia výsledky trénovania.

4.8.3 Dokumentácia k metódam

4.8.3.1 epoch()

```
trainer.Trainer.epoch (
              self,
              dataloader,
              training,
              caption )
```

Generická metóda pre všetky epochy.

Parametre

dataloader	Trieda pomocou ktorej periodicky čítame dávky nášho datasetu.
training	Ak je hodnota True, tak vykonávame spätnú propagáciu. Ak je hodnota False, tak spätnú propagáciu nevykonávame.
caption	Popis aktuálnej epochy.

4.8.3.2 fit()

```
trainer.Trainer.fit (
              self,
              checkpoint = None )
```

Táto funkcia spustí trénovanie neurónovej siete.

Parametre

checkpoint	Názov súboru s checkpointom trénovania. Ak nechceme použíť checkpoint a chceme začať trénovať od začiatku, tak zadáme null hodnotu.

4.8.3.3 load_checkpoint()

```
trainer.Trainer.load_checkpoint (
              self,
              filename )
```

Načítanie priebežných výsledkov trénovania z checkpointu.

Parametre

filename	Súbor z ktorého načítame checkpoint.

4.8.3.4 save_checkpoint()

```
trainer.Trainer.save_checkpoint (
              self,
              filename )
```

Uloženie priebežných výsledkov trénovania do checkpointu.

Parametre

filename	Súbor do ktorého uložíme checkpoint.

4.8.3.5 save_model()

```
trainer.Trainer.save_model (
              self )
```

Uloženie výsledného modelu do jit formátu.

Tento model používame v šachovom engine Malakh.

4.8.3.6 save_plot()

```
trainer.Trainer.save_plot (
              self,
              filename,
              caption,
              metric_name,
              train_values,
              val_values )
```

Uloženie trénovacích a validačných metrík do grafu po ukončení trénovania.

Pomocou týchto grafov môžeme vyhodnotiť priebeh trénovania a odpozorovať underfitting alebo overfitting.

Parametre

filename	Súbor do ktorého uložíme graf.
caption	Názov uloženého grafu.
metric_name	Názov zobrazenej metriky.
train_values	Hodnoty trénovacích epoch trénovania.
val_values	Hodnoty validačných epoch trénovania.

4.8.3.7 test_epoch()

```
trainer.Trainer.test_epoch (
              self )
```

Spustenie testovacej epochy.

V tejto epoche nevykonávame spätnú propagáciu. Táto epocha je spustená po ukončení trénovanie pre vyhodnotenie efektivity výsledného modelu.

4.8.3.8 train_epoch()

```
trainer.Trainer.train_epoch (
              self,
              epoch )
```

Spustenie trénovacej epochy.

V tejto epoche vykonávame spätnú propagáciu.

Parametre

epoch	Poradie epochy.

4.8.3.9 val_epoch()

```
trainer.Trainer.val_epoch (
```

```
        self,
        epoch )
```

Spustenie validačnej epochy.

V tejto epoche nevykonávame spätnú propagáciu.

Parametre

epoch	Poradie epochy.

Dokumentácia pre túto triedu bola generovaná z nasledujúceho súboru:

- trainer.py

Kapitola 5

Dokumentácia súborov

5.1 Dokumentácia súboru datamodule.py

Dátové moduly majú na starosť rozdelenie vstupných dát na trénovacie, validačné a testovacie dáta.

Triedy

- class datamodule.DataModuleBitboards

  Dátový modul pre bitboardovú formu MvsFS datasetu.
- class datamodule.DataModuleImages

  Dátový modul pre obrázkovú formu MvsFS datasetu.

5.1.1 Detailný popis

Dátové moduly majú na starosť rozdelenie vstupných dát na trénovacie, validačné a testovacie dáta.

Autor

Martin Šváb

Dátum

Máj 2024

5.2 Dokumentácia súboru dataset.py

Definícia vstupného datasetu MvsFS.

Triedy

- class dataset.MvsFSBitboardDataset

  Dataset MvsFS v bitboardovej forme pre MLP architektúru.
- class dataset.MvsFSImageDataset

  Dataset MvsFS v obrázkovej forme pre CNN architektúru-.

5.2.1 Detailný popis

Definícia vstupného datasetu MvsFS.

Autor

Martin Šváb

Dátum

Máj 2024

5.3 Dokumentácia súboru experiment.py

Definícia konfigurovateľných experimentov.

Funkcie

- experiment.mlp_experiment (input, output_dir, hidden_features, layers)
  Experiment na testovanie efektivity MLP architektúry a bitboardovej formy MvsFS datasetu.
- experiment.cnn_experiment (input, output_dir, fc_layers)
  Experiment na testovanie efektivity CNN architektúry a obrázkovej formy MvsFS datasetu.

5.3.1 Detailný popis

Definícia konfigurovateľných experimentov.

Autor

Martin Šváb

Dátum

Máj 2024

5.3.2 Dokumentácia funkcií

5.3.2.1 cnn_experiment()

```
experiment.cnn_experiment (
              input,
              output_dir,
              fc_layers )
```

Experiment na testovanie efektivity CNN architektúry a obrázkovej formy MvsFS datasetu.

Parametre

input	Názov vstupného csv súboru.
output_dir	Priečinok kde sa uložia výsledky experimentu.
fc_layers	Počet plne prepojených vrstiev CNN architektúry.

5.3.2.2 mlp_experiment()

```
experiment.mlp_experiment (
              input,
              output_dir,
              hidden_features,
              layers )
```

Experiment na testovanie efektivity MLP architektúry a bitboardovej formy MvsFS datasetu.

Parametre

input	Názov vstupného csv súboru.
output_dir	Priečinok kde sa uložia výsledky experimentu.
hidden_features	Počet neurónov skrytých vrstiev MLP architektúry.
layers	Počet skrytých vrstiev MLP architektúry.

5.4 Dokumentácia súboru model.py

Definícia architektúr neurónových sietí.

Triedy

- class model.MLP

  Architektúra plne prepojenej neurónovej siete.
- class model.CNN

  Architektúra konvolučnej neurónovej siete.

5.4.1 Detailný popis

Definícia architektúr neurónových sietí.

Autor

Martin Šváb

Dátum

Máj 2024

5.5 Dokumentácia súboru trainer.py

Definícia trénovania neurónovej siete.

Triedy

- class trainer.EarlyStopper

  Trieda zodpovedná za predčasné zastavenie trénovania.
- class trainer.Trainer

  Trieda zodpovedná za trénovanie neurónových sietí.

Funkcie

- trainer.decide_device ()

  Pomocná funkcia na rozhodnutie použítia CPU alebo cuda režimu.

5.5.1 Detailný popis

Definícia trénovania neurónovej siete.

Autor

Martin Šváb

Dátum

Máj 2024

5.5.2 Dokumentácia funkcií

5.5.2.1 decide_device()

```
trainer.decide_device ( )
```

Pomocná funkcia na rozhodnutie použítia CPU alebo cuda režimu.

Návratová hodnota

Ak je cuda dostupné tak hodnota cuda, ak nie je dostupné tak hodnota cpu.

Register