

Экспериментальная часть

МІРТ

1 ноября 2022

Обучение модели TFF

1. Датасет HCP – весит 6 ТБ, не удалось скачать субсет
2. Датасет UCLA – скачивается, но не предобработан
3. Датасет COBRE (и другие) – то же не предобработаны
4. На вход модели подаётся тензор [batch, inChannels=2, W, H, D, T=1]
5. Функция предобработки (по снимку получение voxel norm снимка и global norm снимка) находится в файле preprocessing

Далее рассмотрим псевдокод алгоритма обучения

Обзор алгоритма обучения (TFF)

```
def main():
    model_phase1 = run_phase(args, None, 'autoencoder_reconstruction')
    model_phase2 = run_phase(args, model_phase1, 'tranformer_reconstruction')
    model_phase3 = run_phase(args, model_phase2, f'fine_tune_{fine_tune_task}')
    test(args, model_weights_path_phase3)

def run_phase(args, model_path, phase_name):
    S = ['train', 'val']
    trainer = Trainer(sets=S, **args)
    trainer.training()
    return model

class Trainer():
    def training(self):
        for epoch in range(self.nEpochs):
            self.train_epoch(epoch)

    def train_epoch(self, epoch):
        self.train()
        for input_dict in self.train_loader:
            self.optimizer.zero_grad()
            loss_dict, loss = self.forward_pass(input_dict)
            loss.backward()
            self.optimizer.step()
```

Обзор алгоритма обучения (TFF)

```
def forward_pass(self, input_dict):
    output_dict = self.model(input_dict['fmri_sequence'])
    loss_dict, loss = self.aggregate_losses(input_dict, output_dict)
    return loss_dict, loss

def aggregate_losses(self, input_dict, output_dict):
    final_loss_dict = {}
    final_loss_value = 0
    for loss_name, current_loss_dict in self.writer.losses.items():
        if current_loss_dict['is_active']:
            loss_func = getattr(self, 'compute_' + loss_name)
            current_loss_value = loss_func(input_dict, output_dict)
            ...
    final_loss_dict['total'] = final_loss_value.item()
    return final_loss_dict, final_loss_value
```

Обзор алгоритма обучения (подсчёт лоссов)

```
def compute_reconstruction(self, input_dict, output_dict):
    fmri_sequence = input_dict['fmri_sequence'][:,0].unsqueeze(1)
    reconstruction_loss = self.reconstruction_loss_func(
        output_dict['reconstructed_fmri_sequence'],
        fmri_sequence
    )
    return reconstruction_loss

def compute_intensity(self, input_dict, output_dict):
    per_voxel = input_dict['fmri_sequence'][:,1,:,:,:,:]
    voxels = get_intense_voxels(per_voxel, output_dict['reconstructed_fmri_sequence'].shape)
    output_intense = output_dict['reconstructed_fmri_sequence'][voxels]
    truth_intense = input_dict['fmri_sequence'][:,0][voxels.squeeze(1)]
    intensity_loss = self.intensity_loss_func(
        output_intense.squeeze(),
        truth_intense
    )
    return intensity_loss

def compute_perceptual(self, input_dict, output_dict):
    fmri_sequence = input_dict['fmri_sequence'][:,0].unsqueeze(1)
    perceptual_loss = self.perceptual_loss_func(
        output_dict['reconstructed_fmri_sequence'],
        fmri_sequence
    )
    return perceptual_loss
```

Датасет

1. Данные

- 389.728 секунд видео (24 кадра в секунду, размер кадра 480x640)
- 641 измерение fMRI для пользователя (частота 0.608 измерений в секунду)

2. Датасет с fMRI

- элемент датасета – полная запись fMRI для пользователя
- пользователи поделены на трейн и тест
- каждый элемент является тензором размером (40, 64, 64, 641)
- один элемент загружается примерно 15 секунд
- весь датасет в память не помещается (загрузить заранее нельзя)

3. Датасет с fMRI и видео

- элемент датасета – k последовательных измерений fMRI, n соответствующих фреймов видео
- пользователи поделены на трейн и тест
- последний интервал из k измерений fMRI отнесен к тесту, в трейн входят все непересекающиеся с ним интервалы

Планы на следующую неделю

1. Обработать датасеты UCLA и другие ($4D \rightarrow [\text{global norm}, \text{voxel norm}]$)
2. pipeline:
 - a) $4D \rightarrow [2, W, H, D, T]$
 - b) Обучение модели на произвольном обработанном датасете, адаптируя код из TFF
3. Выяснить, какие должны быть W, H, D ?