深層学習 - Deep Learning 基礎講座 最終課題レポート

東京大学工学系研究科　原子力国際専攻 D1　岡本 駿一、37237394

VQA に取り組んだ。Github の URL は以下である。また、課題のスコアを得たソースコードは、VQA-competition の main_case08.py である。
https://github.com/oka-shun/dl_lecture_competition_pub.git

- **実行環境**

個人の解析環境にて、課題に取り組んだ。OS は Ubuntu20.04, GPU は NIVIDIA RTX A6000 である。

- **工夫点**

1. **質問文の前処理**

質問文の前処理を施した。小文字として統一、冠詞の削除などを行った。処理内容は process_text 関数と同じである。VQADataset クラスの__getitem__関数に以下のように実装した。

```
        image = Image.open(f"{self.image_dir}/{self.df['image'][idx]}")
        image = self.transform(image)
        question = np.zeros(len(self.idx2question) + 1)  # 未知語用の要素を追加
        ### modify
        question_words = process_text(self.df["question"][idx]).split(" ")
        ###
        for word in question_words:
            try:
                question[self.question2idx[word]] = 1  # one-hot表現に変換
            except KeyError:
                question[-1] = 1  # 未知語
```

2. **回答コーパスの拡張**

訓練データに存在しない回答以外も出力できるようコーパスを拡張した。huggingface の VizWiz に利用できる class_mapping をコピーして利用した。VQAData クラスの__init__関数内に以下のように実装した。

```
        ### add answer copus
        answer_copus = pandas.read_csv(r'VizWiz_class_mapping.csv')
        self.answer2idx = dict(zip(answer_copus['answer'], answer_copus['class_id']))
        self.idx2answer = {v: k for k, v in self.answer2idx.items()}
```

3. **画像のデータ拡張（Data augmentation）**

画像に対して、データ拡張を行った。ここでは、ランダムに左右反転と上下反転を行うデータ拡張を適用した（RandomHorizontalFlip, RandomVerticalFlip）。このとき、40%の

確率で左右反転と上下反転が発生するように設定した。さらに、Global Contrast Normalization (GCN)を適用した。GCN は以下の数式で計算できる。

$$x^{norm} = \frac{x - \bar{x}}{\sqrt{\sigma_x}}$$

4. 学習率のスケジュール

学習率のスケジュールを追加した。使用した関数は StepLR である。学習率の初期値は、2.0×10^{-3} であり、50 エポック毎に、0.5 倍ずつ減衰する設定にした。学習率の遷移は図 1 に併せて示す。

5. ResNet の比較

ここでは、ベースラインコードに記載の ResNet18（main_case08.py）と ResNet50（main_case09.py）の比較を行った。訓練中の損失関数および正解率（Acc）の推移を比較する。訓練中の損失関数と正解率の推移は、異なるものの、300 エポック終了時は損失関数と正解率ともに、概ね同じ値であった。テストデータの結果として、ResNet18 は、0.44279 であり、ResNet50 は 0.41129 であった。よって、ResNet18（main_case08.py）で計算したスコアを提出した。

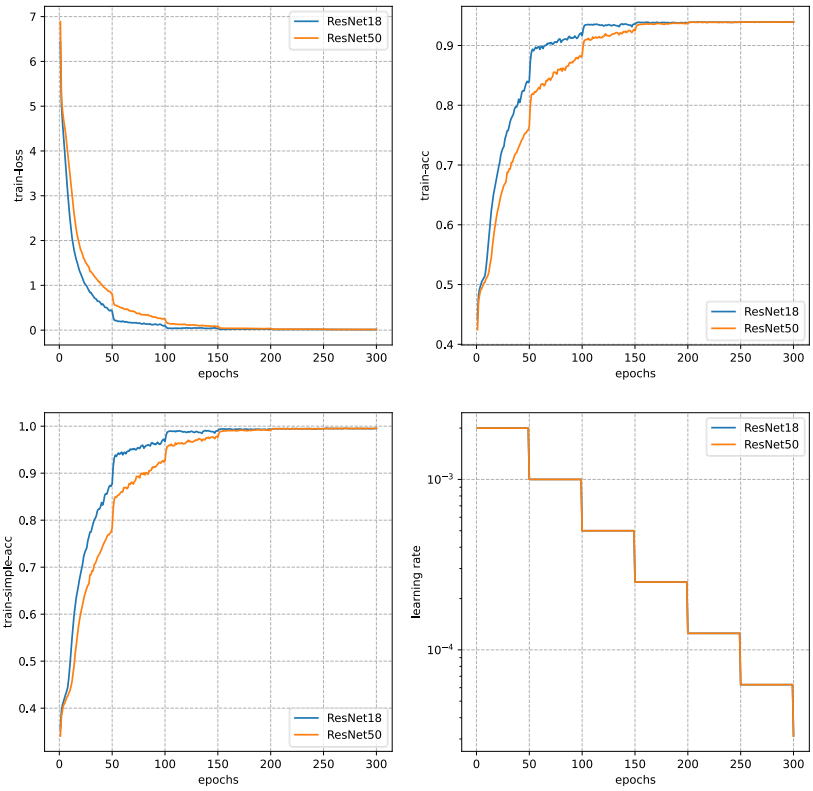

図 1 訓練中の損失関数および正解率の推移