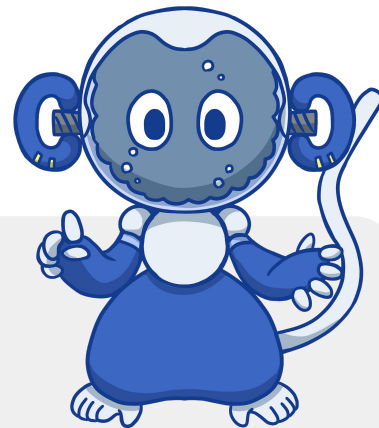


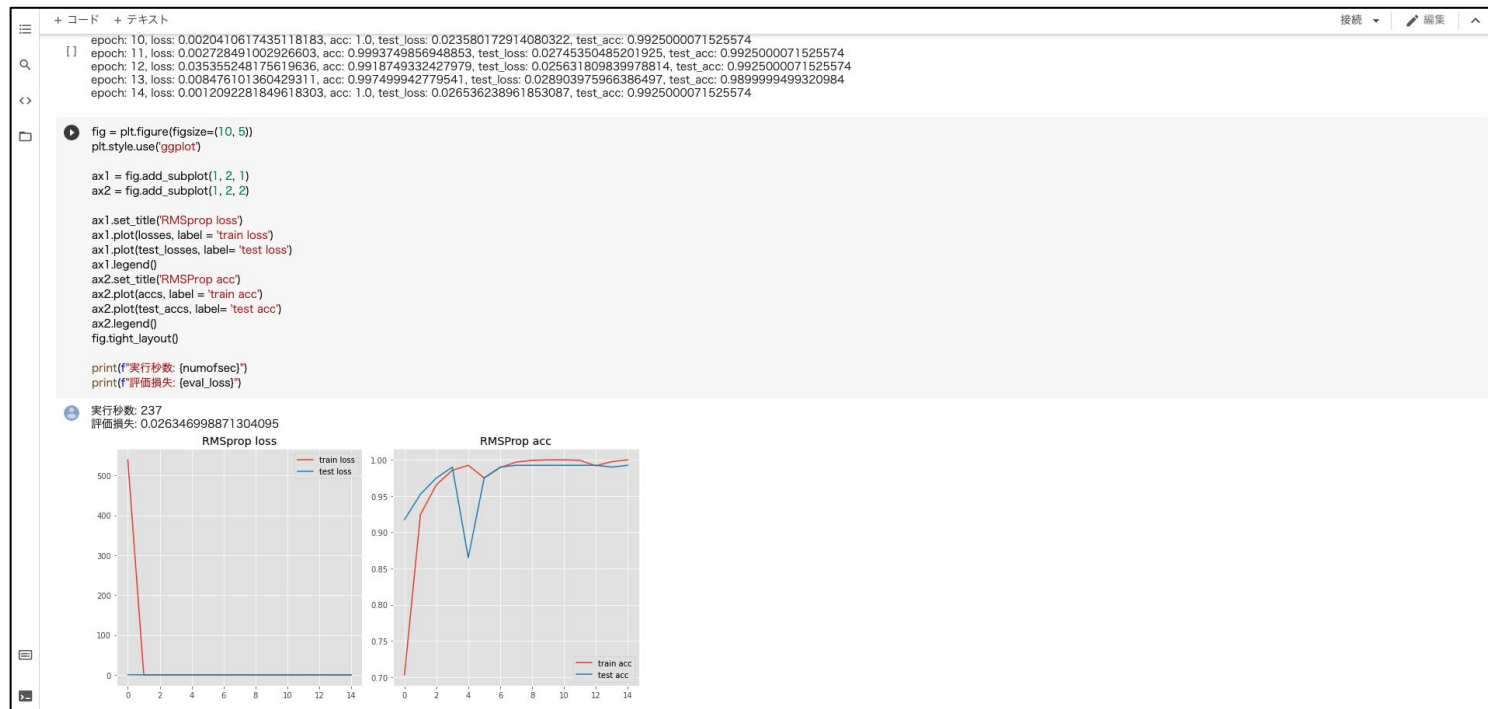
プロダクト開発試験

1. ipynbなどに学習結果・予測デモが出力されている
2. 理解した上で工夫がされている痕跡（コメント）がある
3. 結果についての考察がされている
4. 深層学習(中間層3層以上)を利用すること

インターネット上のコードを一部流用してもOK！
ただ、丸々コピペして提出したようなコードは不合格だよ。
自分の工夫や考察をしっかり入れてね。



1. ipynbなどに学習結果・予測デモが出力されている



2. 理解した上で**工夫**がされている痕跡（コメント）がある



工夫ってなんだろう？

精度をあげるための妥当性が見受けられる工夫ならOKだよ！
講義内で紹介されている手法を使用しても、
妥当性がなければNGだから気をつけてね。

例) 画像処理の場合

○Pythonによる画像データ増し（クロープ使用など）

○GANを使用して画像生成

○ハイパーパラメータ最適化

×根拠や説明なく説明変数を増やしている

×タスクに対して不適切なデータ前処理（MNISTでローテーション）

3. 結果についての考察がされている

- 学習結果を受けて考察する。

例) lossが減らない→なぜlossが減らないかを考察する など

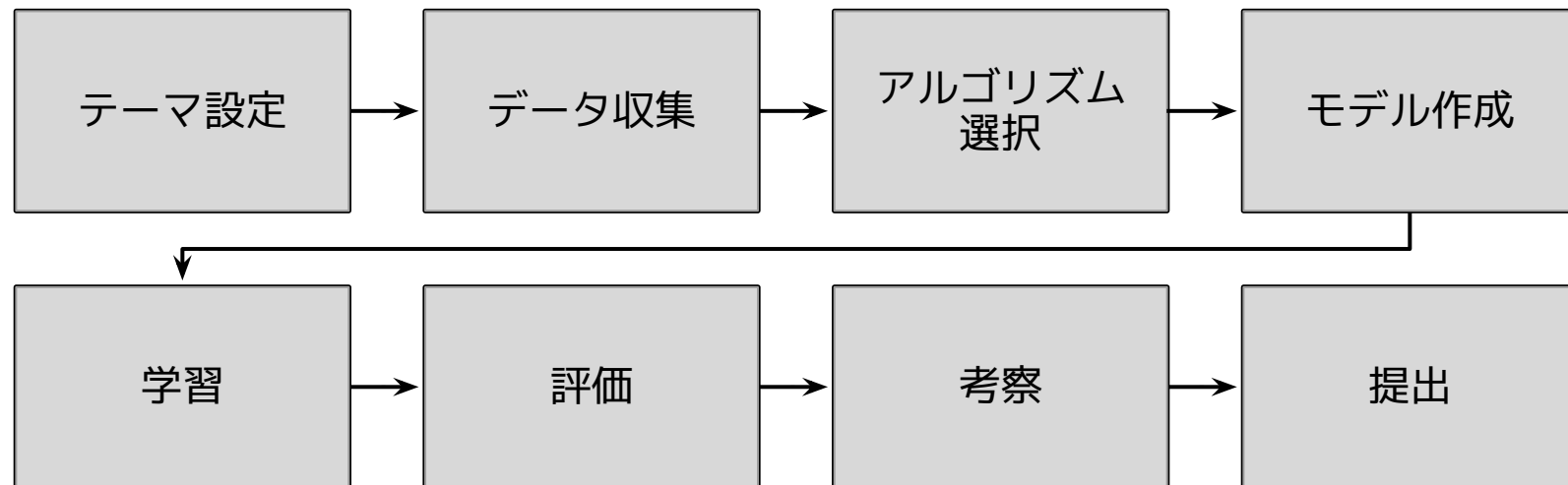
- 今後の精度向上につながる内容を調査し、記載する。

例) より発展させてやってみたいモデル

最新論文などから最適化手法をピックアップ など

4. 深層学習(中間層3層以上)を利用すること

※ランダムフォレストなどの機械学習手法は深層学習(中間層3層以上)に含まれません。



※テーマ設定でお悩みの方は「オンライン学習相談」をおすすめしております。

- 難易度はテーマによって異なります。
 - 難しい例：自作データを複数の最新モデルで学習し考察する
 - 簡単な例：cifar-10を異なるモデルで比較し考察する
- テーマ選びが難しい場合は「【プロダクト開発試験】参考テーマ」から選択してください。

- データの収集が最初の山になります。
- オープンデータ
 - kaggle <https://www.kaggle.com/datasets>
 - AWSデータセット <https://registry.opendata.aws/>、
 - cifar-10、cifar-100などその他
- ご自分でデータを用意することも可能です。その場合はディープラーニングできる形に前処理する必要があります。

- PyTorch・TensorFlow・Keras など、フレームワークの使用を推奨しています。
- ライブラリの紹介
 - PyTorch：利用者が急激に伸びている。記述が簡単で高速。
 - TensorFlow：利用者数が最も多く、文献や記事が豊富。自由度が高く高速。
- Numpyを使って自作ライブラリで実装しても構いません。

- つくりながら学ぶ！
PyTorchによる発展ディープラーニング
著者 小川雄太郎



- 直感 Deep Learning
—Python×Kerasでアイデアを
形にするレシピ
著者 Antonio Gulli, Sujit Pal

