



GPUを用いた機械学習

```
Python 3 (ipykernel) | Idle
Mode: Command | Ln 1, Col 1 | road_following_server.ipynb

Name | Last Modified
dataset_road_f... | 10 days ago
road_following_... | 3 days ago
model_rf_2024... | 10 days ago
dataset_road_f... | 10 days ago
model_rf_2024... | a month ago
model_rf_2024... | a month ago
dataset_road_f... | a month ago
model_rf_2024... | a month ago
dataset_road_f... | a month ago

model = model.to(device)

モデルの訓練
• 画像を一度に8枚ずつ使って訓練する。(このような訓練法をミニバッチ学習という)
• モデルのパラメータ (重み) の更新にはAdamを用いる。

[6]: # ハイパーパラメータなど
# バッチサイズ:8
train_loader = torch.utils.data.DataLoader(train_dataset, batch_size=8, shuffle=True, num_workers=0)
test_loader = torch.utils.data.DataLoader(test_dataset, batch_size=8, shuffle=True, num_workers=0)
optimizer = torch.optim.Adam(model.parameters())
# エポック数
epoch_num = 100

# モデルを保存するファイル
dt = datetime.datetime.now().astimezone(ZoneInfo('Asia/Tokyo'))
model_path = f"model_rf_{dt.strftime('%Y%m%d%H%M%S')}.pth"

# 最高精度の損失を初期化
best_loss = float('inf')

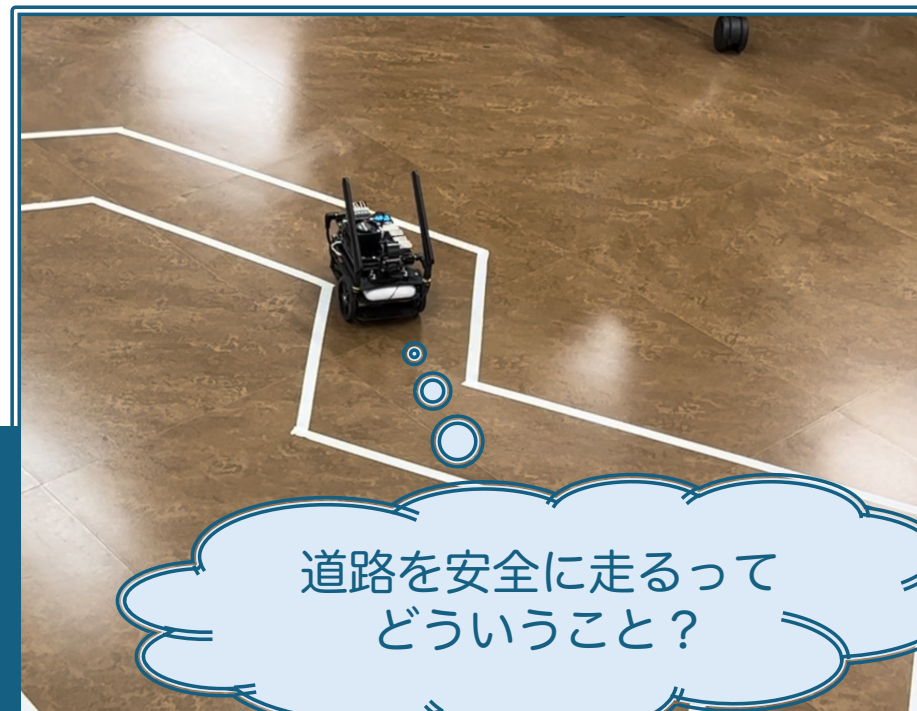
# 開始時間 (計算時間計測用)
time_start = time.time()

# 反復訓練
for epoch in range(epoch_num):
    # 訓練用データでモデルを訓練する
    # 回帰精度のため、モデルの出力と実際の値との誤差は平均二乗誤差 (mean squared error, MSE) で測る
    # 最も精度の高い 訓練用データセットに対する結果の平均値を計算し、更新する
```

プログラミング未経験でもOK

7月29・30・31日 (3日間)
担当者 : 上岡 修平

自分で考えて走る
ロボットに仕上げる



ロボットのAIに 自動運転に必要な2つのタスク

- ① 道路に沿って走る / ② 衝突を察知して回避する

を学習させる実験を通して 機械学習の原理 を学びます。