

Torch Models

mlumina.torch_models

`TorchModelWrapper` (`src/mlumina/torch_models/torchModelWrapper.py`) — тонкая обёртка над произвольным `torch.nn.Module` с единым интерфейсом обучения/инференса, плюс примитивный полный перебор архитектур для табличных данных.

???????? ? ?????????

```
import torch.nn as nn
from mlumina.torch_models import TorchModelWrapper

model = nn.Sequential(nn.Linear(20, 64), nn.ReLU(), nn.Linear(64, 3))
wrapper = TorchModelWrapper(model) # device выбирается автоматически (cuda, если доступна,
иначе cpu)

wrapper.learn(X_train, y_train, epochs=10, batch_size=32, lr=1e-3)

probs = wrapper.predict(X_test, return_probs=True) # softmax по классам
labels = wrapper.predict(X_test, return_probs=False) # argmax

wrapper.save("models/model.pt") # torch.save(model.state_dict(), path)
```

`learn` — обычный mini-batch цикл на `Adam`, по умолчанию `nn.CrossEntropyLoss` (т.е. рассчитан на классификацию с целыми метками классов, не на регрессию — для регрессии нужно передавать свой `criterion`). Прогресс печатается через `print`, а не через логгер `mlumina`.

auto_search — ???????? ?????????????

```
best_wrapper = TorchModelWrapper.auto_search(X, y, epochs=4)
```

Статический метод. Полным перебором проверяет **54 комбинации** (6 наборов слоёв × 3 dropout × 3 функции активации: `ReLU`, `GELU`, `LeakyReLU`) полносвязных сетей вида `Linear` →

`activation → Dropout → ... → Linear(num_classes)`, обучает каждую `epochs` эпох на 80% данных, выбирает лучшую по accuracy на оставшихся 20%. Возвращает новый `TorchModelWrapper` с лучшей моделью.

Ограничения (описаны прямо в докстринге метода):

- только для быстрого прототипирования, не для больших датасетов — каждая из 54 моделей обучается синхронно, без Ray/параллелизма;
- нет кросс-валидации, один фиксированный 80/20 сплит;
- архитектуры — только полносвязные (MLP), число классов выводится как `len(torch.unique(y))`, т.е. предполагается классификация.

Revision #1

Created 2026-09-23 06:40:06 UTC by Маслов Алексей

Updated 2026-09-23 06:40:07 UTC by Маслов Алексей