classification_example

November 13, 2025

[1]:
```
%reload_ext autoreload
%autoreload 2
```

[2]:
```
from kret_studies import *
from kret_studies.notebook import *
from kret_studies.complex import *

logger = get_notebook_logger()
```

```
Loaded environment variables from /Users/Akseldkw/coding/kretsinger/.env.
```

```
INFO:datasets:JAX version 0.7.2 available.
```

[3]:
```
from kret_studies.kret_torch.nn_mse_ce import ClassificationNN
```

[4]:
```
class ObesityPredNN(ClassificationNN):
    def set_model(self, input_dim: int):
        self.model = nn.Sequential(
            nn.Linear(input_dim, 64),
            nn.ReLU(),
            nn.Linear(64, 32),
            nn.ReLU(),
            nn.Linear(32, 1),
        )
```

[5]:
```
run = start_wandb_run(ObesityPredNN.name(), project="obesity-risk-prediction")
```

```
wandb: Currently logged in as: akseldkw
(akseldkw07) to https://api.wandb.ai. Use `wandb login
--relogin` to force relogin
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

[6]:
```
KAGGLE_DIR = DATA_DIR / "kaggle"
OBESITY_DIR = KAGGLE_DIR / "playground-series-s4e2"
```

[7]:
```
df = pd.read_csv(OBESITY_DIR / "train.csv")
df_test = pd.read_csv(OBESITY_DIR / "test.csv")
```

[8]:
```
df.drop(columns=["id"], inplace=True)
df_test.drop(columns=["id"], inplace=True)

uks_pd.data_cleanup(df)
uks_pd.data_cleanup(df_test)
```

[9]:
```
df.sample(4)
```

[9]:

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS	NObeyesdad
20698	Male	21.012542	1.773664	97.813023	True	True	2.00000	1.890213	Sometimes	False	2.976672	False	0.000000	1.668318	no	Public_Transportation	Obesity_Type_I
17109	Male	22.771612	1.650000	80.000000	True	True	2.00000	3.000000	Sometimes	False	2.000000	False	2.724300	1.127675	no	Public_Transportation	Overweight_Level_II
10497	Female	20.000000	1.500000	65.000000	True	True	3.00000	2.000000	Sometimes	False	1.000000	True	0.000000	1.000000	Sometimes	Public_Transportation	Overweight_Level_I
17422	Female	37.205173	1.663509	80.000000	True	True	2.79166	3.000000	Sometimes	False	1.651548	False	0.245354	0.000000	no	Automobile	Obesity_Type_II

[10]:
```
df_test.sample(3)
```

[10]:

	Gender	Age	Height	Weight	family_history_with_overweight	FAVC	FCVC	NCP	CAEC	SMOKE	CH2O	SCC	FAF	TUE	CALC	MTRANS
5603	Female	20.601222	1.700000	77.426465	True	True	2.959658	3.0	Sometimes	False	1.854161	False	0.000000	0.680746	no	Automobile
2877	Male	18.164768	1.743790	96.738014	True	True	2.000000	3.0	Sometimes	False	2.474132	False	2.819661	0.000000	no	Public_Transportation
5636	Male	21.501721	1.803132	105.031908	True	True	2.941627	3.0	Sometimes	False	2.042073	False	0.987591			

```
5636  0.000000  Sometimes  Public_Transportation
```

[11]:
```
X, y = uks_pd.split_x_y(df, "NObeyesdad")
```

[12]:
```
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2,␣
↪random_state=42)
```

[13]:
```
X_train.shape, X_val.shape
```

[13]:
```
((16606, 16), (4152, 16))
```

[14]:
```
model = ObesityPredNN(patience=20)
```

[15]:
```
model.set_model(input_dim=X.shape[1])
model.post_init()
```

```
[2025-11-13 21:20:54,447 | INFO | ObesityPredNN_v000 ] Loaded model weights and
state from /Users/Akseldkw/coding/kretsinger/data/pytorch/ObesityPredNN_v000.
[2025-11-13 21:20:54,447 | INFO | ObesityPredNN_v000 ] Full State:
{'hparams': {'batchsize': 128,
             'gamma': 0.1,
             'improvement_tol': 0.0001,
             'lr': 0.001,
             'patience': 20,
             'stepsize': 7},
 'state': {'best_eval_accuracy': '12.62%',
           'best_eval_f1': '2.83%',
           'best_eval_loss': inf,
           'best_eval_r2': '-inf%',
           'epochs_trained': 0}}
```

[16]:
```
model.train_model((X_train, y_train), (X_val, y_val), epochs=5_000,␣
↪batch_size=320)
```

```
  0%|          | 19/5000 [00:06<28:53,  2.87it/s]
```

```
[2025-11-13 21:21:01,586 | WARNING | ObesityPredNN_v000 ] Early stopping
activated: no improvement for 20 consecutive epochs.
```

```
  0%|          | 19/5000 [00:07<30:53,  2.69it/s]
```

[17]:
```
y_true = np.random.randint(0, 5, size=100)
y_pred = np.random.randint(0, 5, size=100)
```

[18]:
```
f1_score(y_true, y_pred, average="weighted")
```

[18]:
```
0.16563492063492063
```

[19]:
```
y_true_copy = y_true.copy()
f1_score(y_true_copy, y_true, average="weighted")
```

[19]: 1.0

[20]:
```
model.eval()
xb, yb = next(iter(model._to_dataloader((X_train, y_train), batch_size=128)))
xb, yb = xb.to(model.device), yb.to(model.device)

model.zero_grad()
out = model(xb)
loss = model.get_loss(out, yb)
loss.backward()

max_grad = max(
    (p.grad.abs().max().item() for p in model.parameters() if p.grad is not␣
 ↪None),
    default=0.0,
)
print("Loss:", loss.item(), "Max grad:", max_grad)
```

```
Loss: 0.0 Max grad: 0.0
```

[ ]: