M9-L1-P1

November 7, 2023

1 M9-L1 Problem 1

Here, you will implement three loss functions from scratch in numpy: MAE, MSE, and MAPE.

[ ]:
```
import numpy as np

y_gt1 = np.array([1,2,3,4,5,6,7,8,9,10])
y_pred1 = np.array([1,1.3,3.1,4.6,5.9,5.9,6.4,9.2,8.1,10.5])

y_gt2 = np.array([-3.23594765, -3.74125693, -2.3040903 ,  0.        ,  0.
 ↪30190142, -1.68434859,  1.10160357,  0.8587438 ,  1.76546802,  3.13787123, 3.
 ↪72990216,  5.89871795,  6.06406803,  6.28329118,  7.46406525, 8.21246221, 10.
 ↪23145281,  9.39080133, 10.76761316, 10.45903557, 9.61872736, 13.68392163, 14.
 ↪75332509, 14.00530973, 17.87581523, 15.01028079, 17.36899084, 17.99463433,␣
 ↪20.57318325, 21.36834867, 20.91252318, 21.99432414, 21.58696173, 21.
 ↪35253687, 23.84400704, 25.20685402, 27.13938159, 27.97005662, 27.23893581,␣
 ↪28.18254573, 28.29488138, 28.78200226, 29.35433587, 33.86996731, 32.2681256␣
 ↪, 33.19828933, 33.24215413, 36.13102571, 34.59822336, 36.85796679, 37.
 ↪03382637, 39.17478129, 39.13565951, 39.32441832, 41.33545414, 42.65055409,␣
 ↪43.1473253 , 44.24186584, 44.1636577 , 45.29382449, 45.84269107, 47.
 ↪01418421, 47.41917695, 47.36462649, 50.12692109, 50.40629987, 50.03646832,␣
 ↪52.98803478, 52.47654002, 54.29436964, 55.83010066, 56.08857887, 57.9575825␣
 ↪, 56.44194186, 58.93769518, 58.7091293 , 59.3817281 , 60.53226145, 61.
 ↪65814444, 62.88444817, 62.52171885, 65.44628103, 65.86970284, 64.72638258,␣
 ↪68.60946432, 69.87568716, 70.01716341, 69.51704486, 69.48480293, 72.
 ↪46859314, 71.86955033, 74.3537582 , 74.19817397, 75.82512388, 76.0634371 ,␣
 ↪77.27222973, 77.43474244, 80.06869878, 79.26832623, 80.40198936])
```

```
y_pred2 = np.array([-3.17886560e+00, -3.72628642e+00, -2.28154027e+00, -2.
↪42424242e-06, 2.96261368e-01, -1.70080838e+00,  1.09113641e+00,  8.
↪60043722e-01, 1.76729042e+00,  3.12498677e+00,  3.72452933e+00,  5.
↪81293300e+00, 6.01791742e+00,  6.27564586e+00,  7.43093457e+00,  8.
↪18505900e+00, 1.00785853e+01,  9.41006754e+00,  1.07339029e+01,  1.
↪05483666e+01, 9.86429504e+00,  1.35944803e+01,  1.46257911e+01,  1.
↪41092530e+01, 1.74700758e+01,  1.52285866e+01,  1.73610430e+01,  1.
↪80283176e+01, 2.02578402e+01,  2.10543695e+01,  2.08801196e+01,  2.
↪19111495e+01, 2.17786086e+01,  2.17754891e+01,  2.39269636e+01,  2.
↪51674432e+01, 2.68054871e+01,  2.76337491e+01,  2.73444399e+01,  2.
↪82677426e+01, 2.85915692e+01,  2.91907133e+01,  2.98552019e+01,  3.
↪32092384e+01, 3.24325813e+01,  3.33437229e+01,  3.36586115e+01,  3.
↪58501097e+01, 3.51566050e+01,  3.69363787e+01,  3.73654528e+01,  3.
↪90232127e+01, 3.93355670e+01,  3.97886962e+01,  4.13471034e+01,  4.
↪24678677e+01, 4.31186248e+01,  4.41080463e+01,  4.44437982e+01,  4.
↪54581242e+01, 4.61509657e+01,  4.71832256e+01,  4.78047650e+01,  4.
↪81822755e+01, 5.00379827e+01,  5.06088232e+01,  5.08521636e+01,  5.
↪27428151e+01, 5.29526597e+01,  5.42661662e+01,  5.54230479e+01,  5.
↪60162341e+01, 5.72972123e+01,  5.71389028e+01,  5.87005639e+01,  5.
↪91111760e+01, 5.98988234e+01,  6.08826528e+01,  6.18502423e+01,  6.
↪28491288e+01, 6.32501917e+01,  6.48567227e+01,  6.55629719e+01,  6.
↪57207391e+01, 6.75883810e+01,  6.85509197e+01,  6.91918142e+01,  6.
↪96421235e+01, 7.02288144e+01,  7.17044458e+01,  7.21593122e+01,  7.
↪34448231e+01, 7.40436375e+01,  7.50845851e+01,  7.57923722e+01,  7.
↪67262442e+01, 7.74266118e+01,  7.86387737e+01,  7.91677250e+01,  8.
↪00787815e+01])
```

1.1 Mean Absolute Error

Complete the definition for `MAE(y_gt, y_pred)` below.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|$$

`MAE(y_gt1, y_pred1)` should return 0.560.

[ ]:
```
def MAE(y_gt, y_pred):
    return np.sum(abs(y_gt - y_pred)) / len(y_gt)

print(f"MAE(y_gt1, y_pred1) = {MAE(y_gt1, y_pred1):.3f}")
print(f"MAE(y_gt2, y_pred2) = {MAE(y_gt2, y_pred2):.3f}")
```

```
MAE(y_gt1, y_pred1) = 0.560
MAE(y_gt2, y_pred2) = 0.290
```

1.2 Mean Squared Error

Complete the definition for `MSE(y_gt, y_pred)` below.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(e_i)^2 = \frac{1}{n}e^T e$$

`MSE(y_gt1, y_pred1)` should return 0.454.

[ ]:
```
def MSE(y_gt, y_pred):
    return np.sum(np.power((y_gt - y_pred),2)) / len(y_gt)

print(f"MSE(y_gt1, y_pred1) = {MSE(y_gt1, y_pred1):.3f}")
print(f"MSE(y_gt2, y_pred2) = {MSE(y_gt2, y_pred2):.3f}")
```

```
MSE(y_gt1, y_pred1) = 0.454
MSE(y_gt2, y_pred2) = 0.174
```

1.3 Mean Absolute Percentage Error

Complete the definition for `MAPE(y_gt, y_pred, epsilon)` below.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{|y_i| + \varepsilon} = \frac{1}{n}\sum_{i=1}^{n}\frac{|e_i|}{|y_i| + \varepsilon}$$

`MAPE(y_gt1, y_pred1, 1e-6)` should return 0.112.

[ ]:
```
def MAPE(y_gt, y_pred, epsilon=1e-6):
    return np.sum(abs(y_gt - y_pred)/(abs(y_gt) + epsilon)) / len(y_gt)

print(f"MAPE(y_gt1, y_pred1) = {MAPE(y_gt1, y_pred1):.3f}")
print(f"MAPE(y_gt2, y_pred2) = {MAPE(y_gt2, y_pred2):.3f}")
```

```
MAPE(y_gt1, y_pred1) = 0.112
MAPE(y_gt2, y_pred2) = 0.032
```