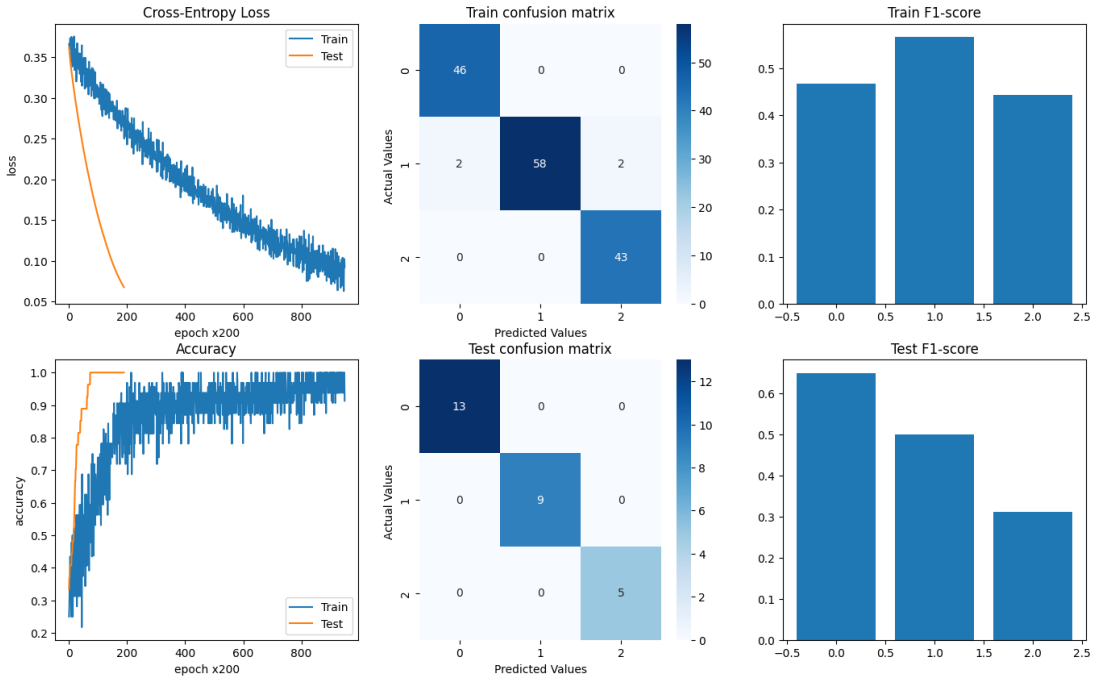
Wine classification

1) Implement F1-score and confusion matrix

```python
def get_f1_score(matrix: np.ndarray) -> dict:
    score = {}

    for item_idx in range(matrix.shape[0]):
        tp = matrix[item_idx, item_idx]
        tn = matrix[item_idx].sum() - tp
        fn = matrix[:, item_idx].sum() - tp
        fp = matrix.sum() - tp - tn - fn

        score[item_idx] = (2 * tp) / (2 * tp + fp + fn)

    return score
```

```python
def get_confusion_matrix(expected, predicted) -> np.ndarray:
    matrix = np.zeros(shape=(3, 3), dtype=np.int)

    for expected_item, predicted_item in zip(expected, predicted):
        matrix[expected_item][predicted_item] += 1

    return matrix
```



Conv2d (PyTorch)

1) Implement Conv2D task, but instead of using MNIST please change dataset and model to use LFW dataset: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_lfw_people.html#sklearn.datasets.fetch_lfw_people

Tried to adjust the model, but never got good results for accuracy

```
self.encoder = torch.nn.Sequential(
    Conv2d(in_channels=1, out_channels=3, kernel_size=9, stride=2, padding=1),
    ReLU(),
    Conv2d(in_channels=3, out_channels=6, kernel_size=7, stride=2, padding=1),
    ReLU(),
    Conv2d(in_channels=6, out_channels=12, kernel_size=5, stride=2, padding=1),
    ReLU(),
    Conv2d(in_channels=12, out_channels=24, kernel_size=3, stride=2, padding=1),
    ReLU(),
    Conv2d(in_channels=24, out_channels=48, kernel_size=3, stride=2, padding=1)
)

o_1 = get_out_size(INPUT_SIZE, kernel_size=9, stride=2, padding=1)
o_2 = get_out_size(o_1, kernel_size=7, stride=2, padding=1)
o_3 = get_out_size(o_2, kernel_size=5, stride=2, padding=1)
o_4 = get_out_size(o_3, kernel_size=3, stride=2, padding=1)
o_5 = get_out_size(o_4, kernel_size=3, stride=2, padding=1)

self.fc = Linear(
    in_features=48*o_5*o_5,
    out_features=feature_count
)
```