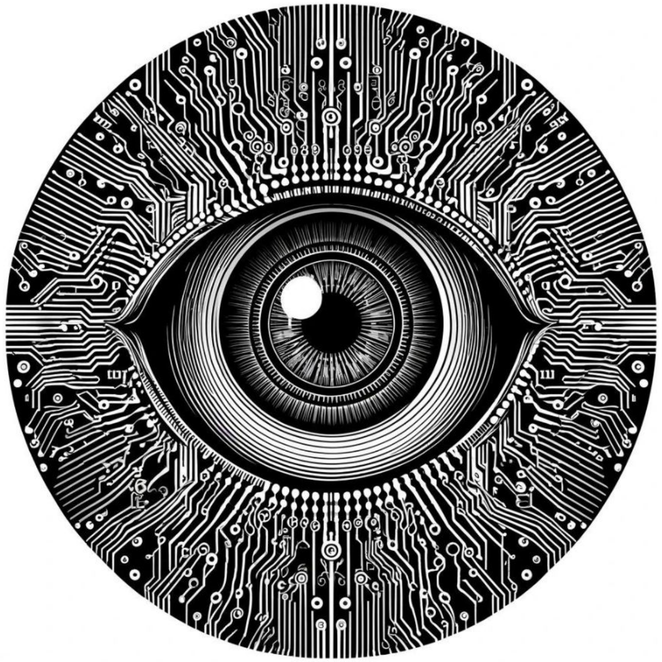
Performance Metrics for Classification and Experiment Tracking



Antonio Rueda-Toicen

SPONSORED BY THE

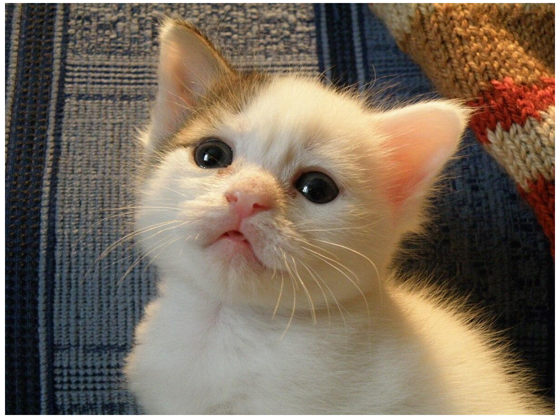
Federal Ministry of Education and Research

KI Service Zentrum by Hasso-Plattner-Institut HPI

Learning goals

- Understand the use of performance metrics for classification
- Identify elements in a confusion matrix
- Explore the effect of threshold moving on classification metrics
- Compute accuracy, precision, recall and f1 score for classifiers
- Checkpointing models with the best performance
- Logging experiments into `wandb`

Confusion matrices



$$y = P(\text{cat}) = 1.0$$

$$\hat{y} = P(\text{cat}) = 0.95$$

$$\text{True Positive (TP)} \Rightarrow y = 1, \hat{y} = 1$$
$$\text{True Negative (TN)} \Rightarrow y = 0, \hat{y} = 0$$
$$\text{False Positive (FP)} \Rightarrow y = 0, \hat{y} = 1$$
$$\text{False Negative (FN)} \Rightarrow y = 1, \hat{y} = 0$$

'positive' = 1, `negative' = 0 (not interchangeable)

$$\begin{bmatrix} \text{TP: } 50 & \text{FN: } 10 \\ \text{FP: } 5 & \text{TN: } 100 \end{bmatrix}$$

A classifier outputs probabilities, so how do we define that y_hat = 1?

$$\text{Threshold for positive: } P > 0.5$$

Moving the threshold for ŷ = 1

Threshold for positive: $P > 0.5$

Threshold for positive: $P > 0.9$

Interpretability with performance metrics

assume that a classifier predicts the correct class with confidence 0.8 seven times and with confidence 0.1 three times

Cross Entropy Loss

$$-\frac{1}{10}\left[7\ln(0.8) + 3\ln(0.1)\right]$$

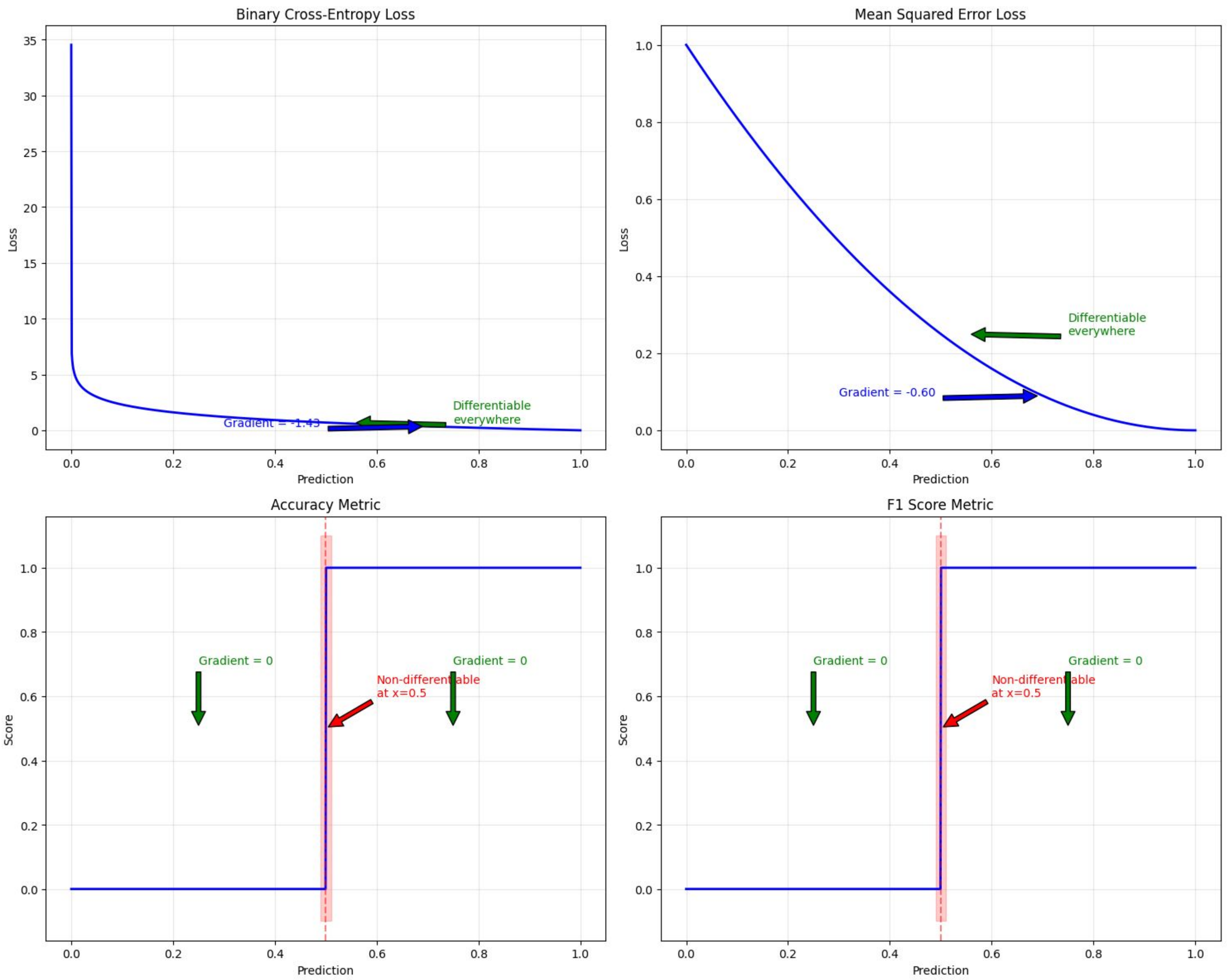
Threshold for positive: $P > 0.5$

$$\text{accuracy} = \frac{\#\text{correct predictions}}{\#\text{predictions}}$$

Numerically,

- $\ln(0.8) \approx -0.2231$, so $7\ln(0.8) \approx -1.5617$.
- $\ln(0.1) \approx -2.3026$, so $3\ln(0.1) \approx -6.9078$.
- Total is -8.4695; divided by 10, it's about 0.847.

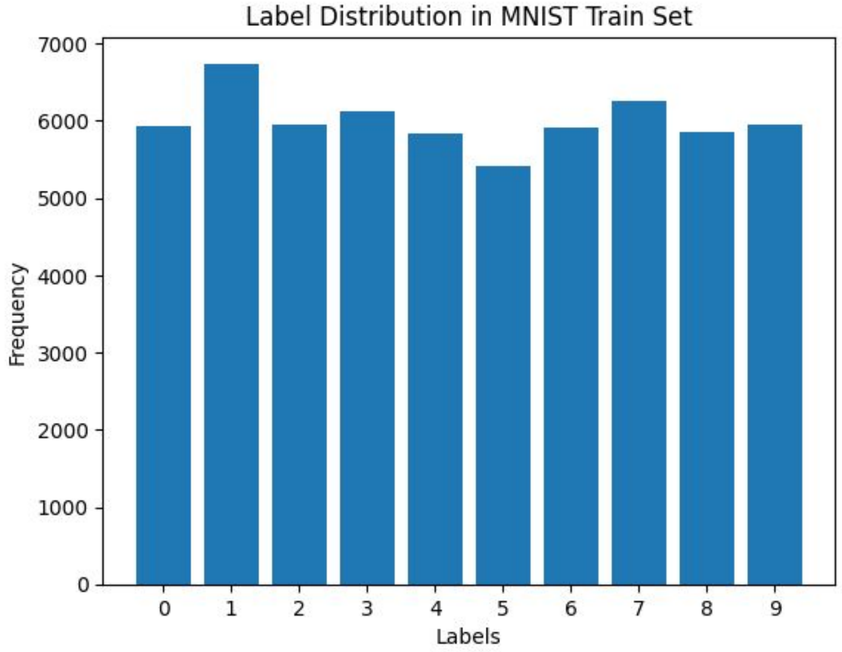
Performance metrics vs loss functions




```
Gradient Examples:
BCE gradient at pred=0.7:
-1.429
MSE gradient at pred=0.7:
 -0.600
Accuracy 'gradient' at pred=0.7:
 0.000
F1 'gradient' at pred=0.7:
 0.000
```

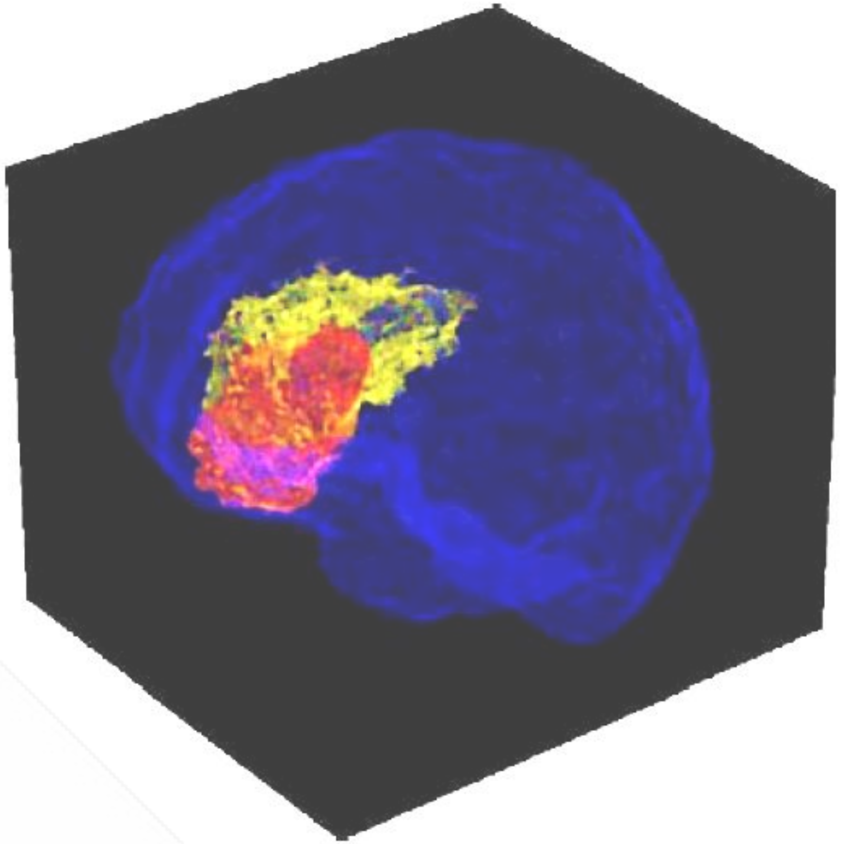
Multiclass accuracy is only good for balanced datasets

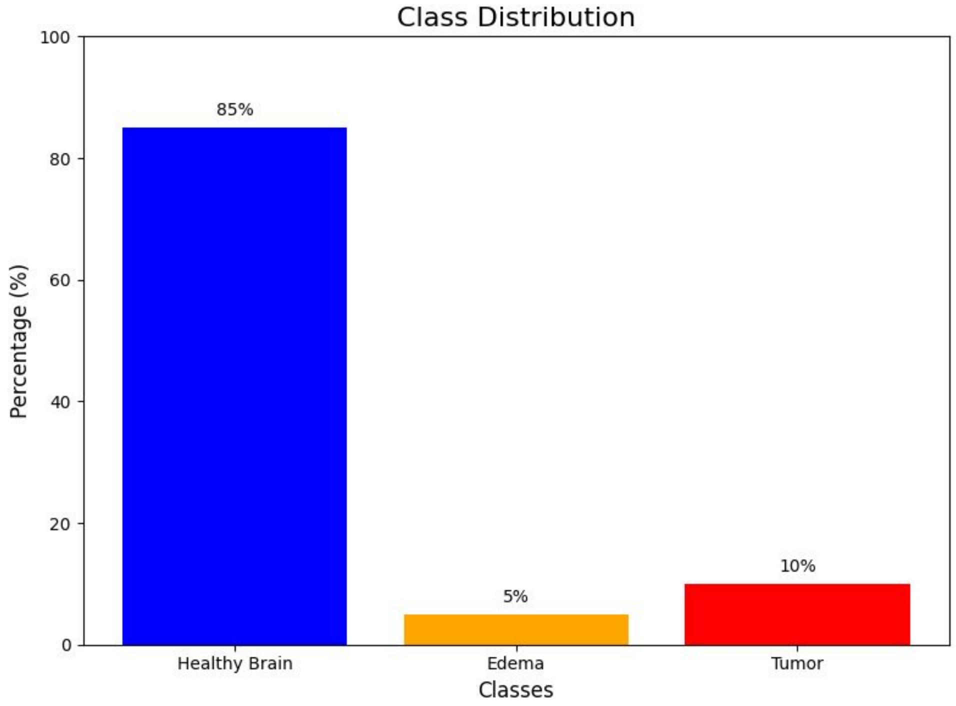
```python
def get_batch_accuracy(output, y):
    # Here we give as prediction
    # the label where we have the top confidence
    # No threshold, just top confidence
    pred = probs.argmax(dim=1, keepdim=True)
    correct = pred.eq(y.view_as(pred)).sum().item()
    return correct / probs.shape[0]
```



$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} = \frac{TP + TN}{TP + FP + FN + TN}$$

Accuracy is misleading in imbalanced datasets





'no tumor' in the image, majority class prediction

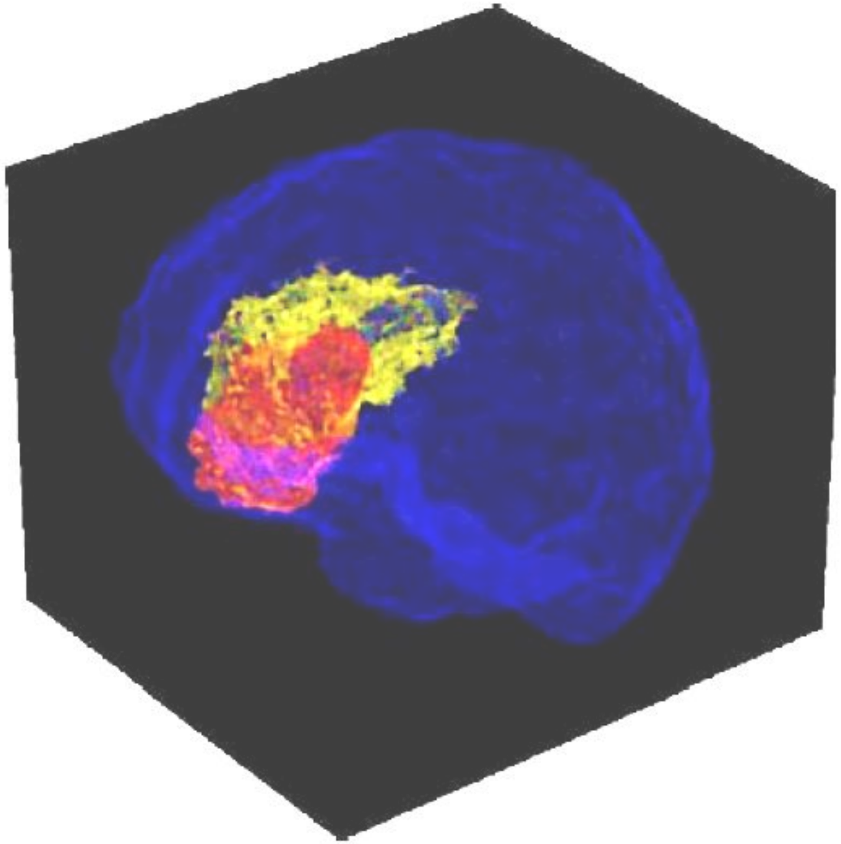
$$\text{Correct Predictions} = 900$$

$$\text{Total Predictions} = 1000$$

$$\text{Accuracy} = \frac{900}{1000} = 0.9\,(90\%)$$

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} = \frac{TP + TN}{TP + FP + FN + TN}$$

Tradeoffs in precision and recall



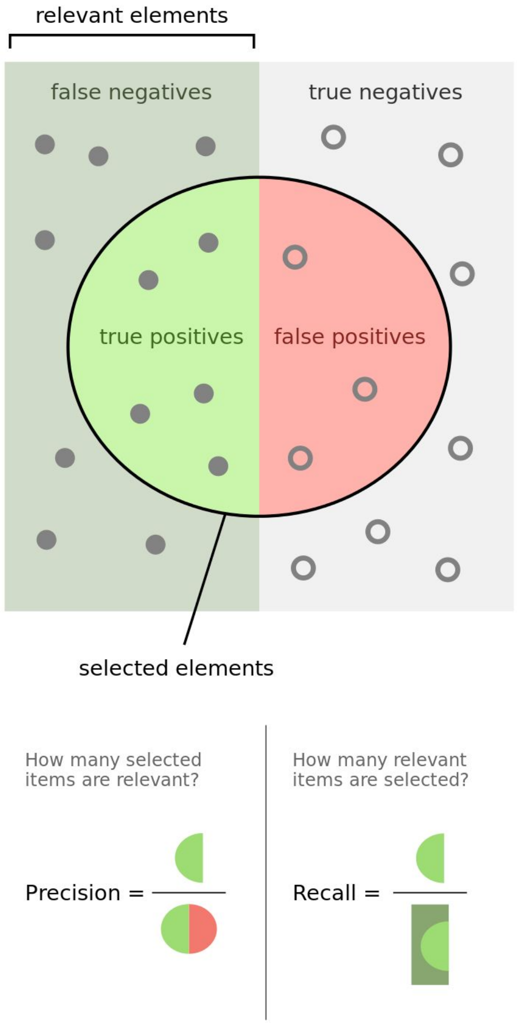
Labeling just one voxel as tumor, with the prediction being correct = perfect precision

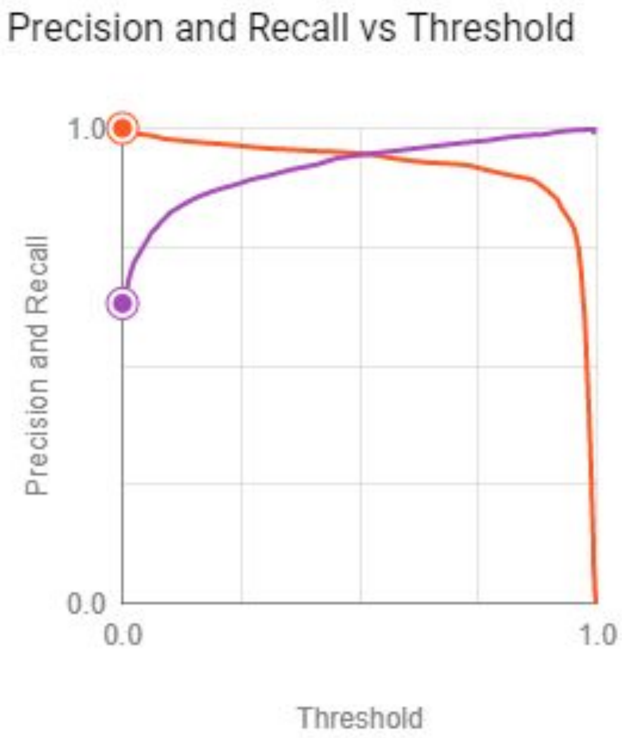
$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Labeling the whole brain as tumor = perfect recall

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

Thresholds for y_hat = 1, precision, recall, and f1 metrics





$$F_1 = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

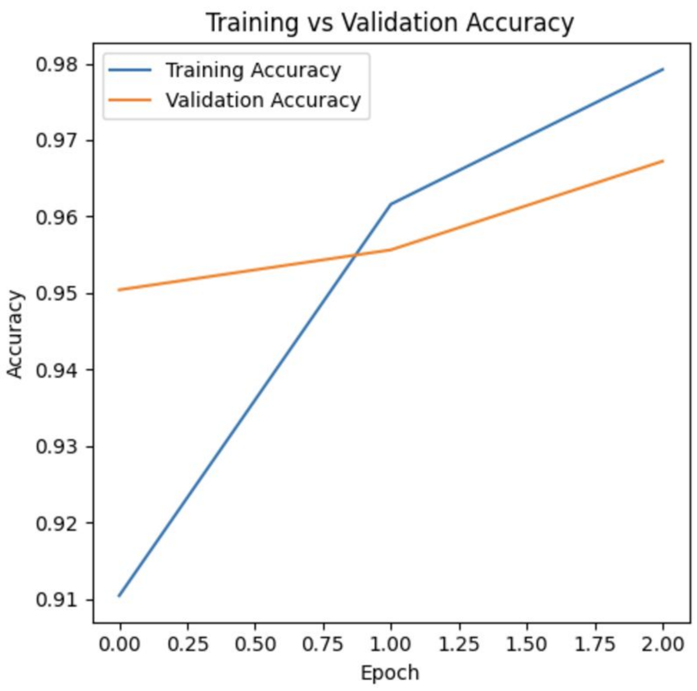
Checkpointing models with the best performance

```
for epoch in range(epochs):

    train_loss, train_acc = train()
    valid_loss, valid_acc = validate()

    # Save model if validation
    #  accuracy improves
    if valid_acc > best_valid_accuracy:
        best_valid_accuracy = valid_acc

        # We save the model as a dictionary
        torch.save(model.state_dict(),
                   'best_model.pth')
```



Logging experiments and models on `wandb`

Image from wandb.ai

Summary

Confusion matrices are the basis of performance metrics for classification

- Threshold moving for the positive class affects them and makes the model more precise or sensitive
- The balance of a dataset impacts the metric used: accuracy for balanced datasets, precision and recall for imbalanced ones

Performance metrics for classification are non-differentiable

- We use them for interpretability, not to directly update the weights

After aligning on a performance metric, we use it to checkpoint our models

- Experiment tracking allows us to benchmark our training runs

SPONSORED BY THE
Federal Ministry of Education and Research