read_lstm_results_tensorboard

January 13, 2023

[ ]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from tensorboard.backend.event_processing.event_accumulator import␣
 ↪EventAccumulator
import matplotlib.pyplot as plt
import os
```

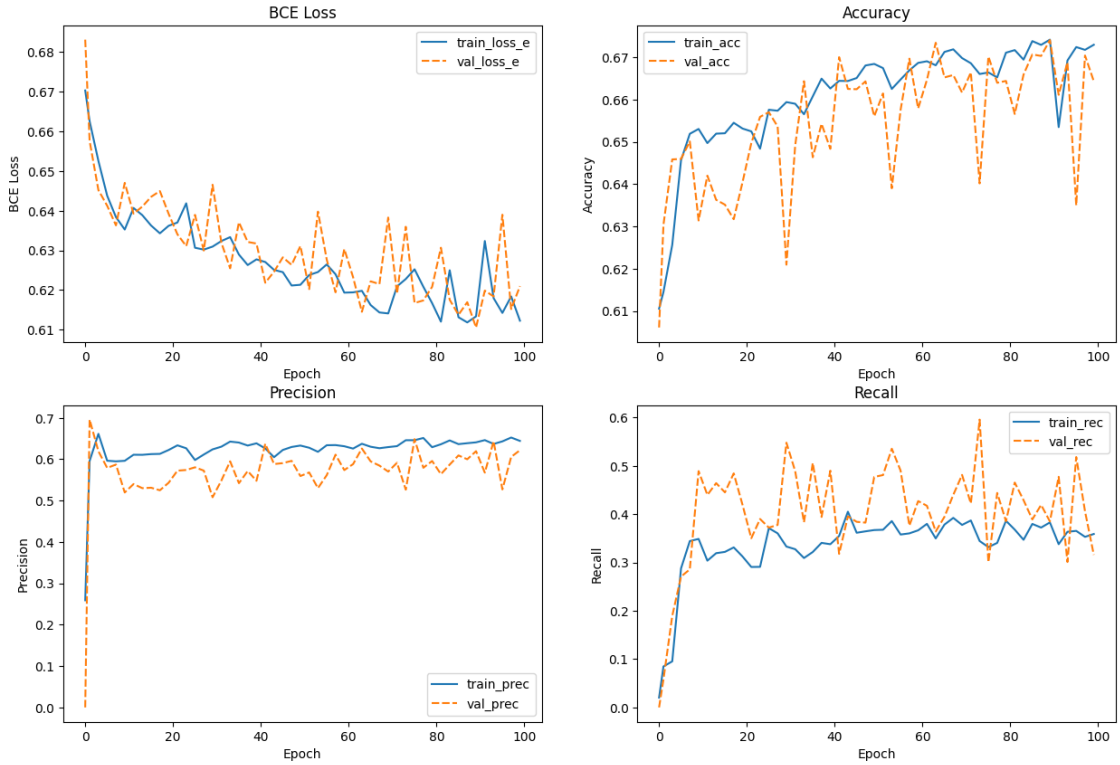
[ ]:
```
def _get_metrics_from_tensorboard(event_acc, scalar):
    train_metrics, val_metrics = event_acc.Scalars(f'{scalar}/train'),␣
 ↪event_acc.Scalars(f'{scalar}/valid')
    train_df, val_df = pd.DataFrame(train_metrics), pd.DataFrame(val_metrics)

    train_df.drop(columns=['wall_time'], inplace=True)
    val_df.drop(columns=['wall_time'], inplace=True)

    train_df.rename(columns={'value': f'train_{scalar}'}, inplace=True)
    val_df.rename(columns={'value': f'val_{scalar}'}, inplace=True)
    results = pd.merge(train_df, val_df, on='step', how='outer')
    results = results.dropna()
    results = results.drop(columns=['step'])
    return results


def tensorboard_results(log_dir, experiment_name):
    events = EventAccumulator(log_dir)
    events.Reload()

    print(events)
    loss, acc, prec, rec = (
        _get_metrics_from_tensorboard(events, 'loss_e'),
        _get_metrics_from_tensorboard(events, 'acc'),
        _get_metrics_from_tensorboard(events, 'prec'),
        _get_metrics_from_tensorboard(events, 'rec')
    )

    result_keys = {
```

```
        'BCE Loss': loss,
        'Accuracy': acc,
        'Precision': prec,
        'Recall': rec
    }

    fig, axes = plt.subplots(2, 2, figsize=(15, 10))
    axes = axes.flatten()
    fig.suptitle(f'Loss, Accuracy, Precision, Recall for {experiment_name}',␣
 ↪fontsize=16)



    results_matrix = pd.concat([loss, acc, prec, rec], axis=1)
    for i, (key, value) in enumerate(result_keys.items()):
        fig_ = sns.lineplot(value, ax=axes[i])
        fig_.set(xlabel='Epoch', ylabel=key, title=key)
    return results_matrix

def top_metrics(results):
    metric_container = []
    for experiment_name, df in results.items():
        metric_container.append(_best_metrics(df, experiment_name))

    return pd.DataFrame(metric_container)

def _best_metrics(df, name):
    return {
        'Experiment': name,
        'BCE Loss': df['val_loss_e'].min().round(4),
        'Accuracy': df['val_acc'].max().round(4),
        'Precision': df['val_prec'].max().round(4),
        'Recall': df['val_rec'].max().round(4),
    }
```

[ ]:
```
log_dir_seq_10 = os.path.join(
    's3://dissertation-data-dmiller/lstm_experiments/results',
    'sequence_length_10',
    '10m',
    '2023_01_12_18_29',
    'version_0'
)
```

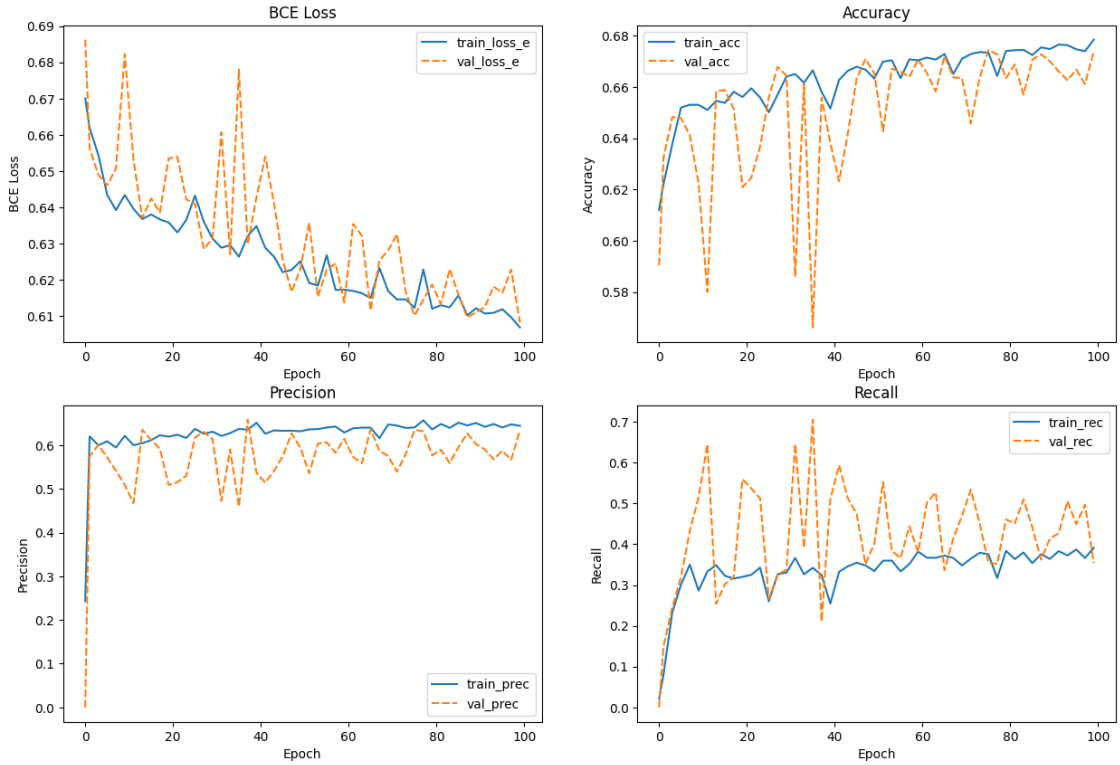
[ ]:
```
seq_10_results = tensorboard_results(log_dir_seq_10, 'Sequence Length 10')
```

```
<tensorboard.backend.event_processing.event_accumulator.EventAccumulator object
at 0x16e0cfee0>
```



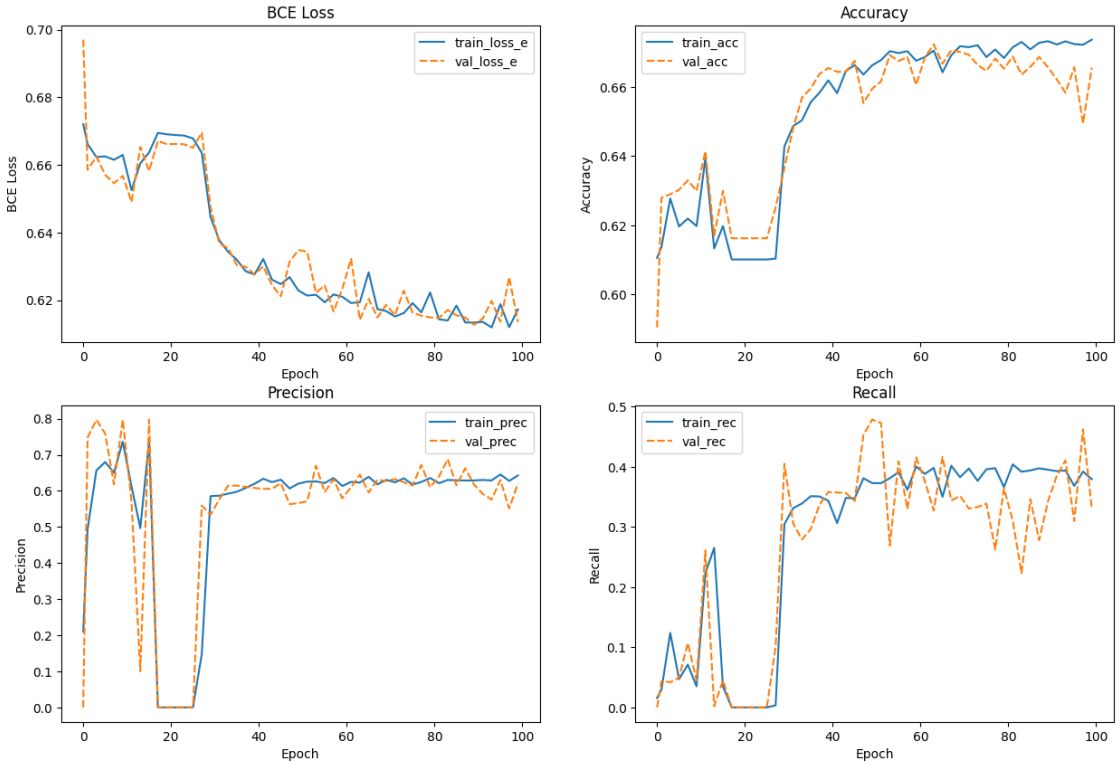
[ ]:
```
log_dir_seq_20 = os.path.join(
    's3://dissertation-data-dmiller/lstm_experiments/results',
    'sequence_length_20',
    '10m',
    '2023_01_13_11_27',
    'version_0'
)

seq_20_results = tensorboard_results(log_dir_seq_20, 'Sequence Length 20')
```

```
<tensorboard.backend.event_processing.event_accumulator.EventAccumulator object
at 0x2842237f0>
```



[ ]:
```
sequence_length_30 = os.path.join(
    's3://dissertation-data-dmiller/lstm_experiments/results',
    'sequence_length_30',
    '10m',
    '2023_01_12_18_30/',
    'version_0'
)

seq_30_results = tensorboard_results(sequence_length_30, 'Sequence Length 30')
```

```
<tensorboard.backend.event_processing.event_accumulator.EventAccumulator object
at 0x17fca24f0>
```



[ ]:
```
seq_30_results.to_csv('10m/seq_30_results.csv')
seq_20_results.to_csv('10m/seq_20_results.csv')
seq_10_results.to_csv('10m/seq_10_results.csv')
```

[ ]:
```
# min valid loss, min valid acc, max valid prec, max valid rec

performance_summary = top_metrics({
    'sequence_10': seq_10_results,
    'sequence_20': seq_20_results,
    'sequence_30': seq_30_results
})
```

[ ]:
```
performance_summary
```

[ ]:

	Experiment	BCE Loss	Accuracy	Precision	Recall
0	sequence_10	0.6106	0.6741	0.6959	0.5957
1	sequence_20	0.6081	0.6743	0.6585	0.7055
2	sequence_30	0.6127	0.6723	0.7975	0.4784