Research Question: How well the basic Recurrent Neural Network architecture used for MSD-10 classification task will perform on the MSD-25 classification task?

We would like to explore whether this basic RNN architecture is able to give us an improvement on the MSD-25 classification task in comparison to the MLP achitecture of coursework3.

Extending RNN class functionality

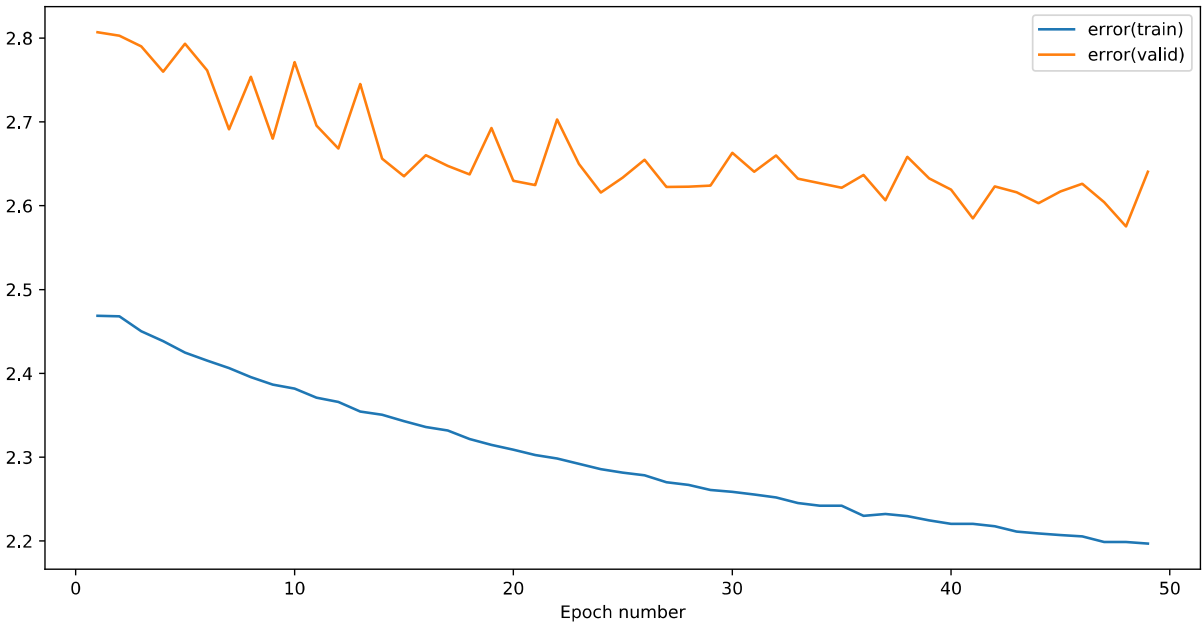
In order to test on MSD-25 class we need to update our `ManualRNN` class. The constructor receives an extra parameter which by default is 10 and can also be set to the value 25.

We also had to generalize the data providers to work with an arbitrary number of classes. For this reason a new module was created found under the path `rnn.data_providers`.
This module contains the class `MSD120RnnDataProvider` which is contains all of the functionality of the `MSD10Genre_120_rnn_DataProvider` that was previously used but by now being agnostic to the number of classes being used. The `MSD120RnnDataProvider` class is now the default class being used for the role of the data provider in `ManualRNN` class.
Note that to complete the functionality we had to re-create the base class of `MSD120RnnDataProvider` class to also be agnostic to the number of classes. So inside `rnn.data_providers` module the `MSDDataProvider` class can be found. This works as long as the number of classes are encoded as the corresponding integer at the prefix of the file which contains the actual data. In other words the filename should begin like *msd-<XX>-...* where *<XX>* should be substituted with the number of the classes.

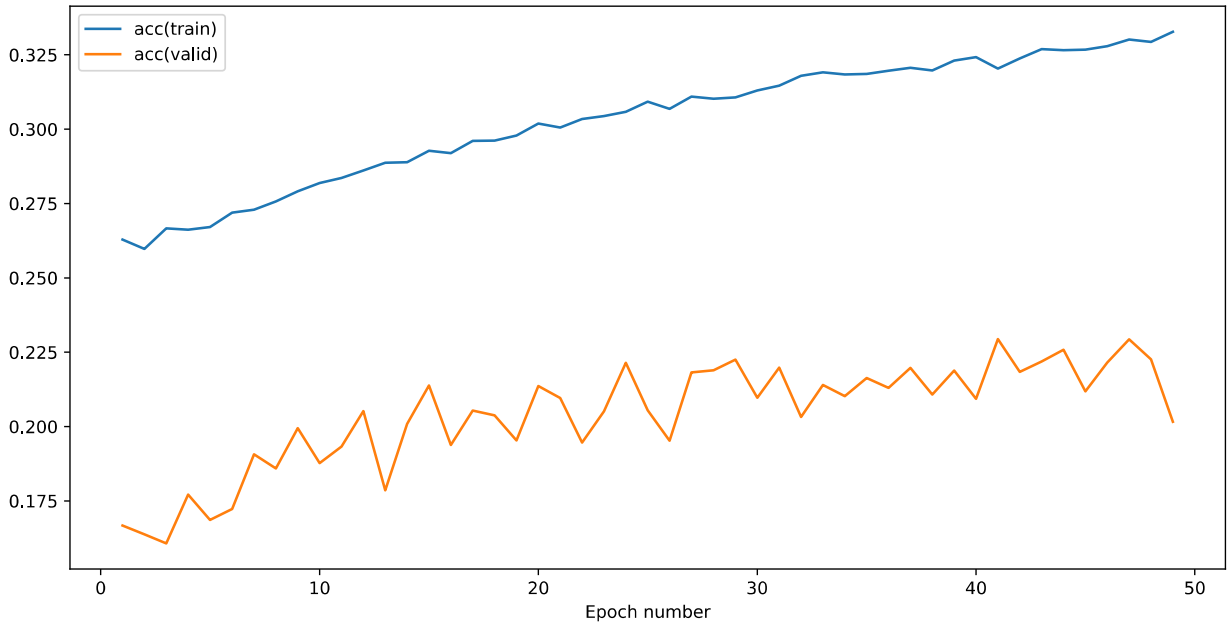
The graph and corresponding architecture is exactly the same as the one used for MSD-10 case.

We are executing the experiment for 50 epochs using the same parameters that worked best for the MSD-10 case:
State Size: 341
Number of Steps: 4

Results



Plot 59: Training & Validation Error – Recurrent Neural Network – State Size: 341 – Num Steps: 4 – MSD 25 classification task – Learning Rate: 1e-4



Plot 60: Training & Validation Accuracy – Recurrent Neural Network – State Size: 341 – Num Steps: 4 – MSD 25 classification task – Learning Rate: 1e-4

Conclusions

We notice that there is a lot of oscillation on the validation accuracy and error which means that the training of our basic RNN model has a hard time to find an equilibrium.

The performance of the validation accuracy is at 22.94% which is considered much lower than the 25% we achieved in coursework3.

We could perhaps achieve a better validation accuracy by performing bayesian optimization on the hyperparameters for the MSD-25 classification task.
However we put in higher priority to address the issue of the high variance of the validation error/accuracy which is apparent in both MSD-10 and MSD-25 classification tasks and which we hypothesize originates from the high variance of the weights of the neural network and as a result it makes training more difficult.