Research Question: Could Curriculum Learning help the MLP Deep Neural Network to achieve better classification accuracy?

Curriculum Learning is the notion of training a deep neural network gradually going from easier examples to more difficult ones.

In our MSD classification problem the notion of easier versus more difficult is hypothesized to be represented by the cross entropy.
Songs described as easier will have a small cross entropy while songs described as more difficult will have a higher cross entropy.

In a nutshell curriculum learning can be described as follows

- First we train our best version of the neural net
- At the last epoch we get all of the cross entropies for all the instances
- Sort the instances based on the cross entropies.
- Start training the instances from easier to harder

Collect Cross Entropies

We already use cross entropy as an error metric to be minimized so no need to add anything else to the graph.

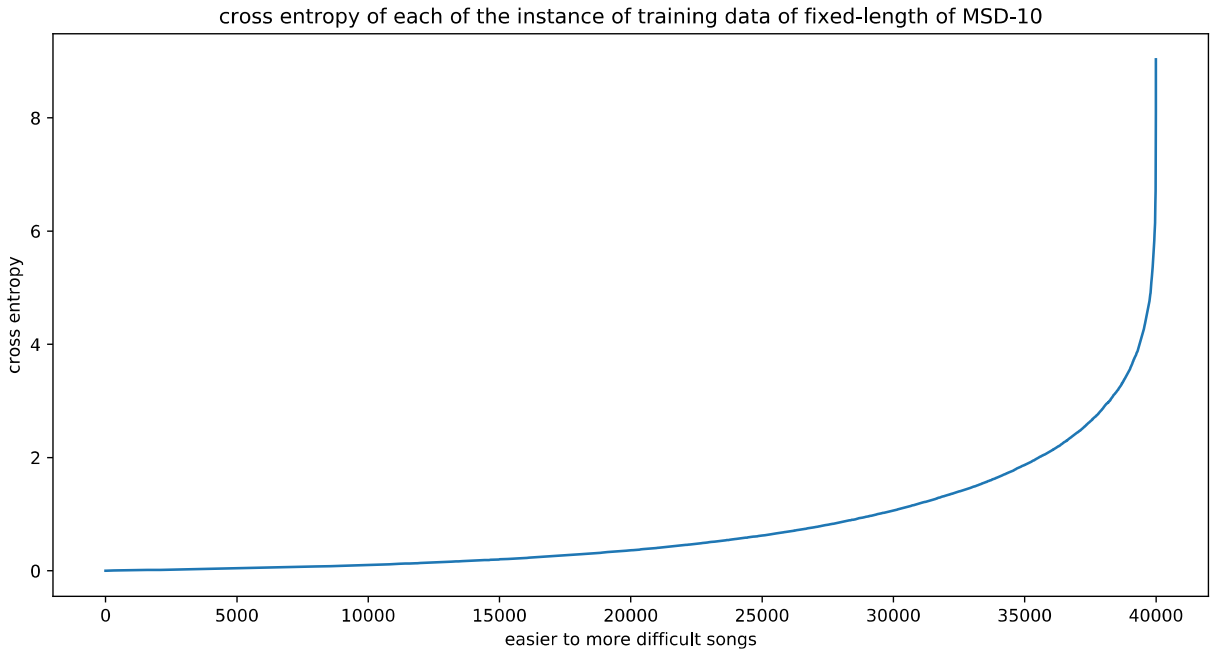
We get our best trained model from the baseline architecture we described above and we run it by collecting all the cross entropies in an array. The indices of this array correspond to the song indices.

Here is the table with the best and worst cross entropies of the training data.

Best Cross Entropies		**Worst Cross Entropies**	
Song ID	**Cross Entropy**	**Song ID**	**Cross Entropy**
36530	0.00018655	4277	6.81745005
22504	0.00040487	20715	6.82365227
23923	0.00047792	10365	7.14912271
14754	0.00050687	21103	7.1554656
36193	0.00051187	35860	7.25000525
5299	0.00063399	19281	7.38539267
8800	0.00075622	33962	7.66791773
7080	0.00076277	22192	8.00147915
23828	0.00079064	17289	8.09997749
21521	0.00084424	22241	9.0351572

Mean value of all cross entropies: `0.7685535595076799`

Variance of all cross entropies: `0.96230627796890367`



Plot 5: Cross Entropies for all of the Song Instances for the fixed-length version of the MSD-10 classification task. They are outputs of the optimal version of the fully trained MLP architecture

Curriculum Learning Data Provider

The `MSD10Genre_Ordered` class takes as input the key_order parameter which is expected to be an array of equal length with the training data containing the specified order of the integers that play the role of the keys on the training dataset.
This is achieved by overriding the `next` function.

The order of the keys can be reversed with the parameter `reverse_order`.
This class does not let `shuffle_order` and `rng` parameters to be set and it always set `shuffle_order` to False at the parent class.

`MSD10Genre_CurriculumLearning` class

This class extends the `MSD10Genre_Ordered` class.

Its constructor accepts the parameter `cross_entropies` which will be the list with cross entropies corresponding to each song unsorted.
Inside the constructor body the cross_entropies get sorted from smaller to larger values and the corresponding keys are passed to the parent constructor.

Parameters:

- `curriculum_step`: We have the flexibility to set how many batches are to be considered in the same group, whether easier or harder. In other words if we set curriculum step to 10 then we consider the first ten batches, or equivalently the 50x10=500 first instances, to belong in the same group of the most easy batches. And then we take the next ten batches and then the next ten and so on.

- `repetitions`: As we train from the curriculum level to the next we might want to persist in the current curriculum level for more than one repetitions. In other words if we are at the first curriculum level and the curriculum step is set to 10 and the repetitions parameter is set to 2 then we are going to train the corresponding 10 batches as many times as the repetitions parameter, in our example twice.
- `repeat_school_class`: This is a boolean parameter which controls whether we are going to keep the instances of the previous curriculum level when transitioning from one curriculum level to the next. In other words if we have 500 instances at the first curriculum level then this boolean control if we are going to train the both the 500 of previous curriculum level and 500 of next curriculum level, entire 1000 instances, on the next curriculum level or only the 500 instances that correspond to the next curriculum level.
- `shuffle_cur_curriculum`: This boolean parameter controls whether the batches that belong in the same curriculum level will be fetched during training sequentially, in the order depended on the cross entropy, or whether they will be shuffled.
- `reverse_order`: This is a boolean parameter to control whether the curriculum learning will happen from easy to hard as is the normal order or from hard to easy, the reverse order.
- `enable_auto_level_incr`: This is a boolean parameter to control whether the curriculum level will increase as we go along the epochs in training, depending on the repetitions parameter, or whether this automatic increment is disabled. By setting it to false the `repetitions` parameter is neglected and we allow the user of this class to call the `updateCurriculumLevel` method to increase the curriculum level manually.

Curriculum Learning Experiment 1

Curriculum Step: 200

Repetitions: 20

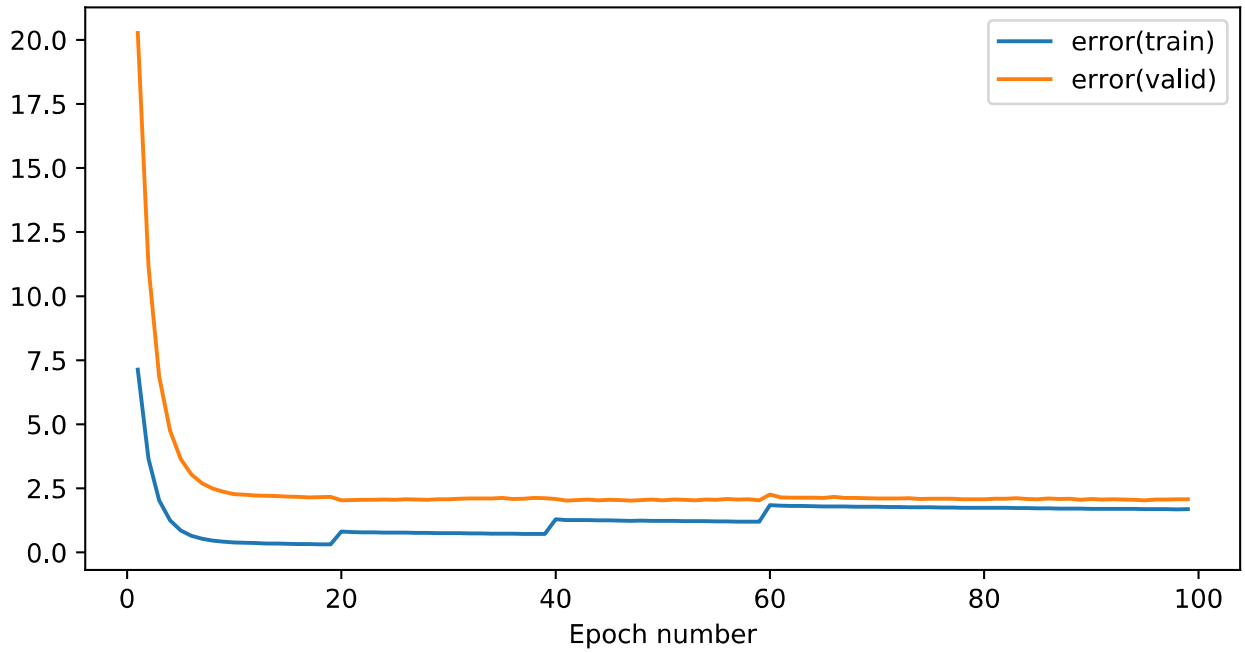
Number of Epochs: 100

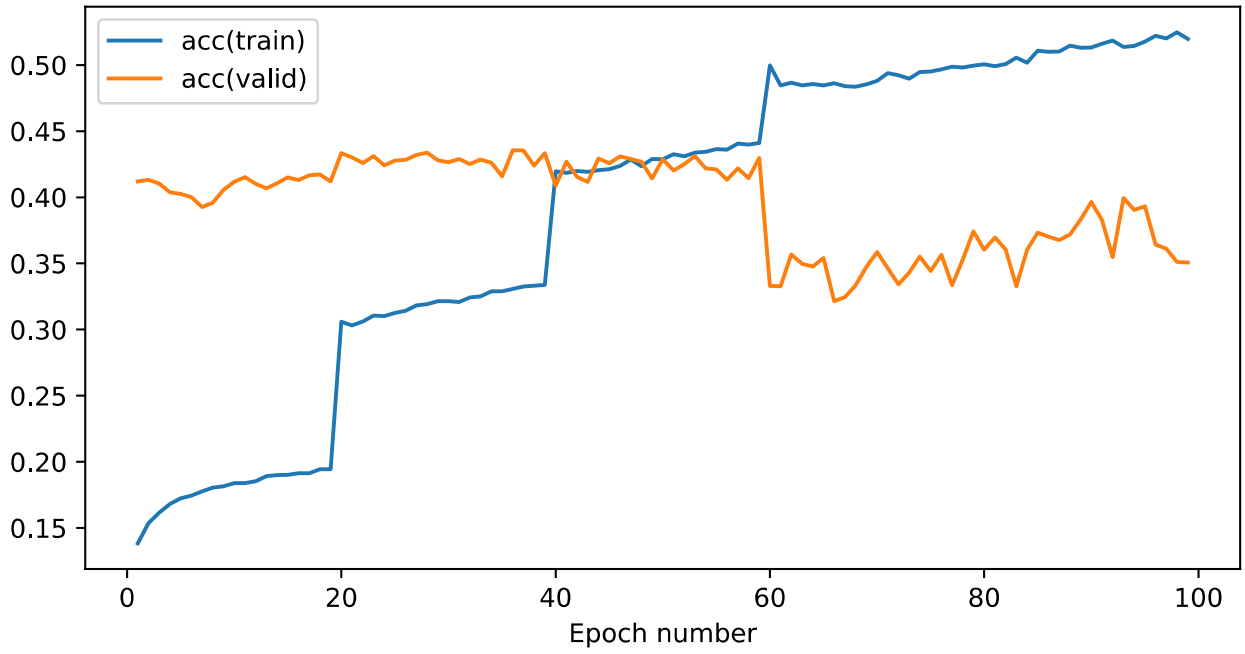
L2 regularization: 1e-1

Initially we are setting a high curriculum step, which breaks the instances in four parts. Also the repetitions of training for each part is set to 20.

We are allowing the network to run for 20 more epochs after the last curriculum level

Results





Conclusions

We notice that having only the 200 easier batches at the beginning is enough to bring the validation accuracy to ~43% which is not too bad in comparison to previous experiments. As the curriculum level increases we note these jumps at the training accuracy and error. Training accuracy is getting better with more instances but we note that the training error is getting worse.

The validation error seems steady but this is not the same with the accuracy. We note that especially when the fourth set of the most difficult songs comes along the accuracy has a sudden drop and it does not seem to be able to increase at the next 40 epochs. There are lots of oscillations which means that the neural network has shifted from a not so good place to a worse one.

The higher L2 regularization that was used in comparison with the baseline provided a steady validation error but could have also be blamed for underfitting the model.

Curriculum Learning Experiment 2

Curriculum Step: 200

Repetitions: 3

Number of Epochs: 32

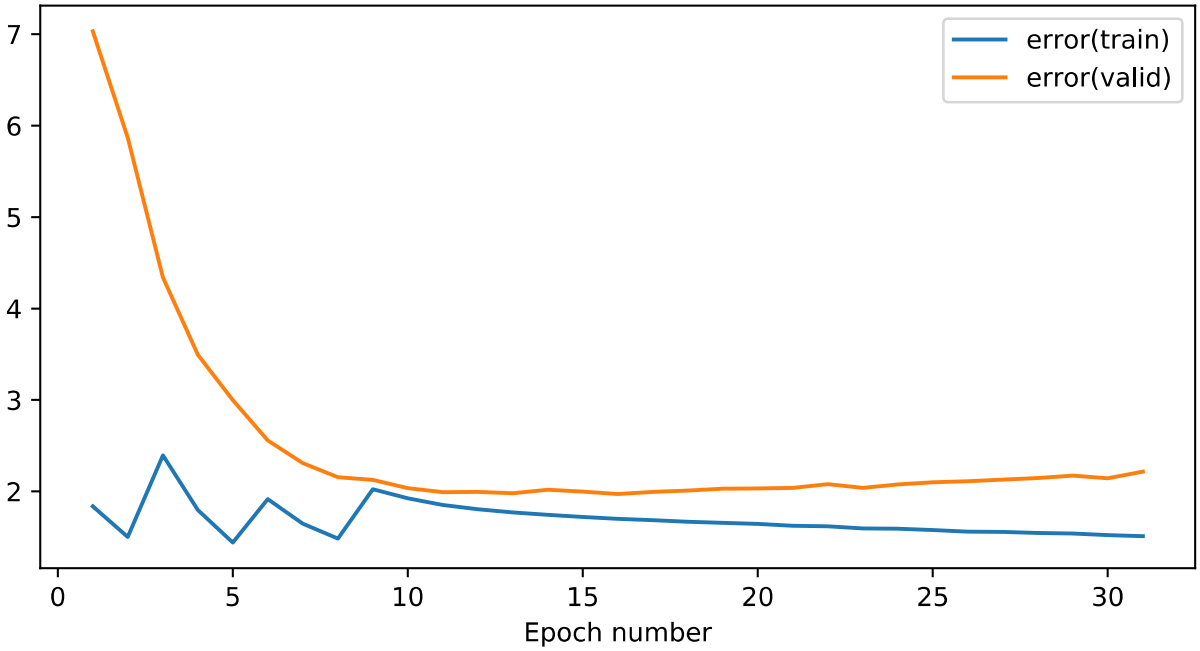
Again we are allowing the training to run for 20 more epochs after the end of the training of the last curriculum level.

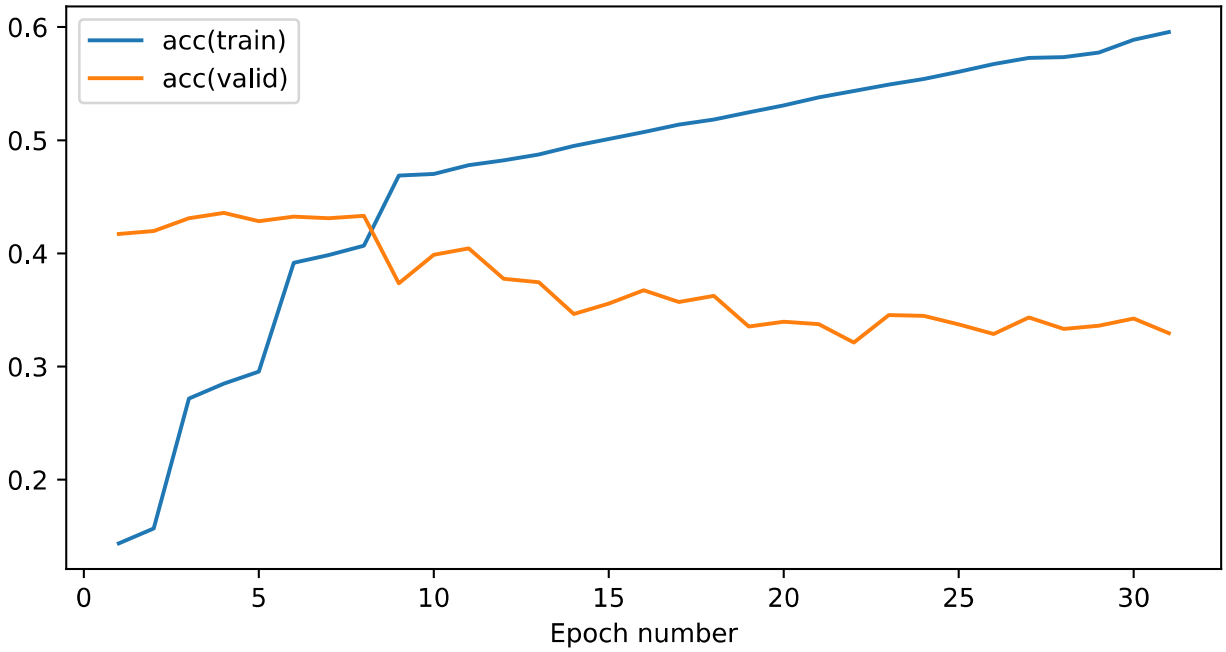
Note that the total number of epochs is proportional to the `repetitions` parameter.

Note that **L2 regularization** is back to the value of **1e-2** as is in the baseline.

In the second experiment we would like to decrease the number of repetitions to avoid giving the chance to the neural network to overfit to the features of the easier example, hypothesizing that this might be the case of the unwanted final results of the previous experiment.

Results





Conclusions

Unfortunately a similar story of the previous experiment is unravelled in the above plots as well but in a smaller scale this time because we have a smaller number of repetitions.

It seems that this pretraining of the neural network with the easier examples which is what it is in fact achieved through curriculum learning does not work as we wanted it. When the neural network is trained with the most difficult examples it is already pretrained in a way that causes the neural network to find an local minimum that is not desirable in terms of regularization. The final effect is that the validation accuracy is worse than our baseline.

The smaller L2 regularization factor, in comparison with the previous experiment, did not provide sufficient regularization and the model suffers from overfitting effects from epoch 15 onwards.

Curriculum Learning Experiment 3

Curriculum Step: 200

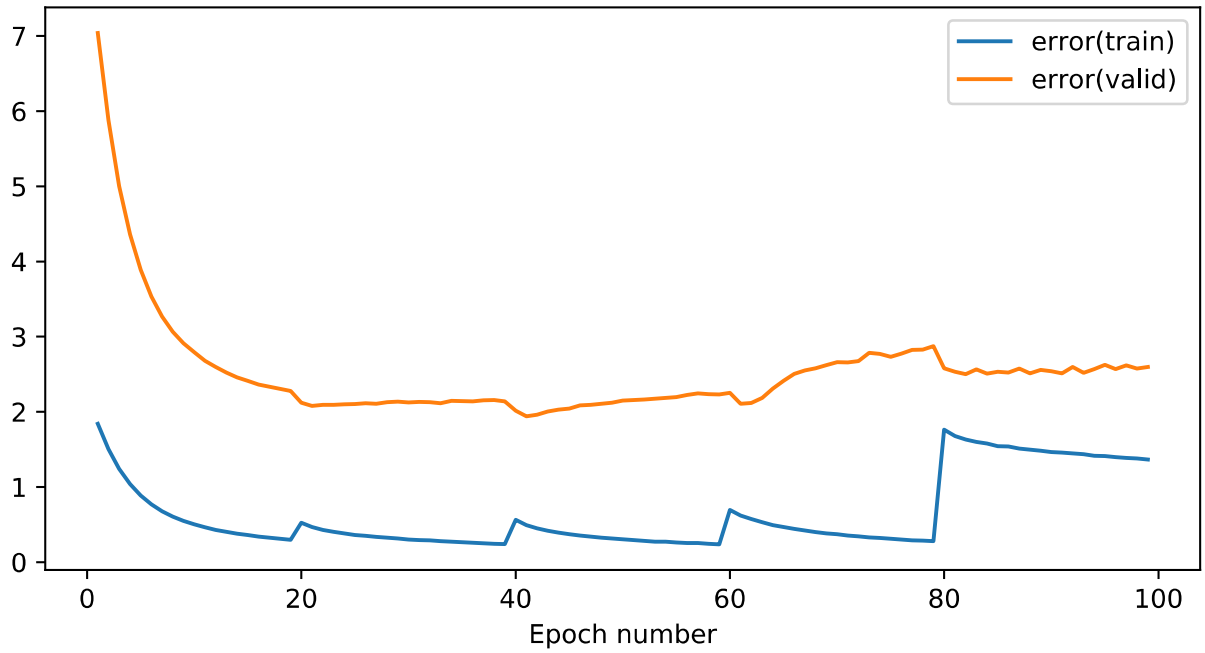
Repetitions: 20

Number of Epochs: 100

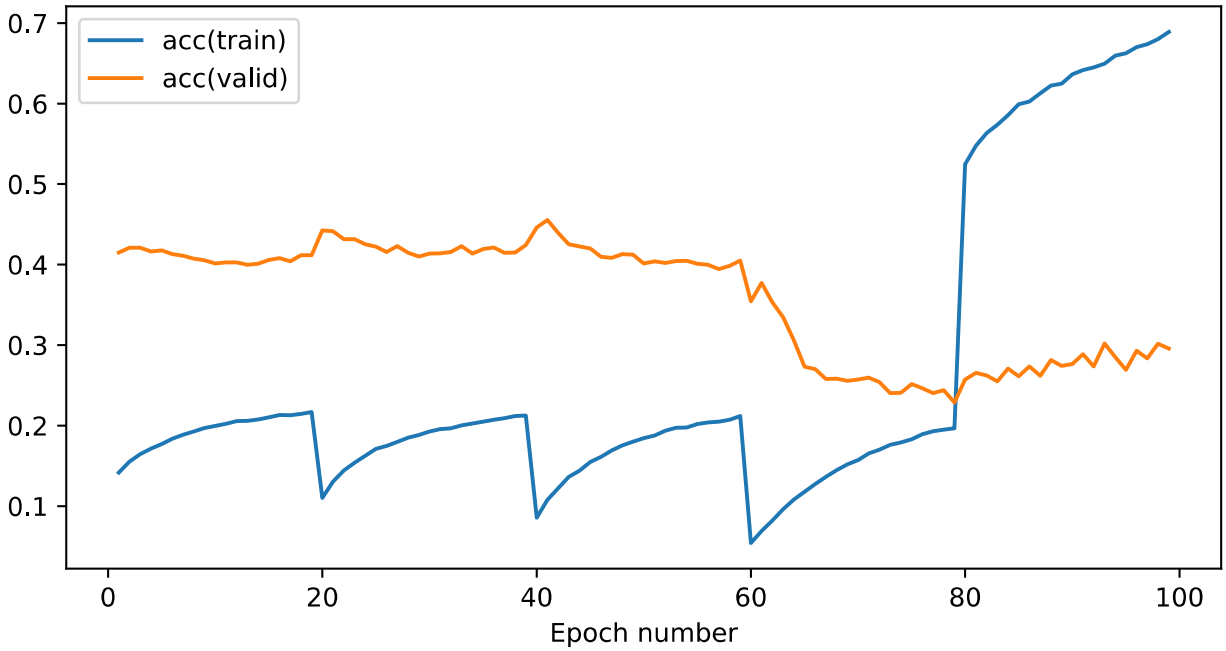
Repeating School Class: False

Note that this is the same as experiment 1 but with two differences. First of all we are letting the L2 regularization be small, because even if there are some overfitting effects we might supress the model with a higher L2 regularization factor and we do not want to do that right now because we are not trying to tweak, rather we are trying to debug.
And secondly we are setting the repeating of school class to false. We are testing to see if we might get a better performance than previous curriculum learning experiments by training only the batches of the current curriculum level at a time and then running 20 epochs by including all the batches.

Results



Plot 6: Training & Validation Error – Curriculum Learning – Curriculum Step: 200 – Repetitions: 20 – Number of Epochs: 100 – Repeating School Class: False



Plot 6: Training & Validation Accuracy – Curriculum Learning – Curriculum Step: 200 – Repetitions: 20 – Number of Epochs: 100 – Repeating School Class: False

Conclusions

Here in this experiment we see that when transitioning from first or second curriculum level to the next, the training error comes with a small rise but not a big one, meaning that the neural network is adequately pretrained to handle the next more difficult set of songs.

We notice that after the 80 epochs where the curriculum training is finished and the neural network is asked to optimize in regards of all the songs in the training dataset, the training error has a very big rise which means that the

pretraining did not work well to train an optimal model in regards to training. We could consider that as not in such thing because we are mostly interested on unseen data and the validation error comes with a small drop and the validation accuracy is not affected a lot which in fact it has a small rising trend.

Curriculum Learning Experiment 4

Curriculum Step: 200

Repetitions: 20

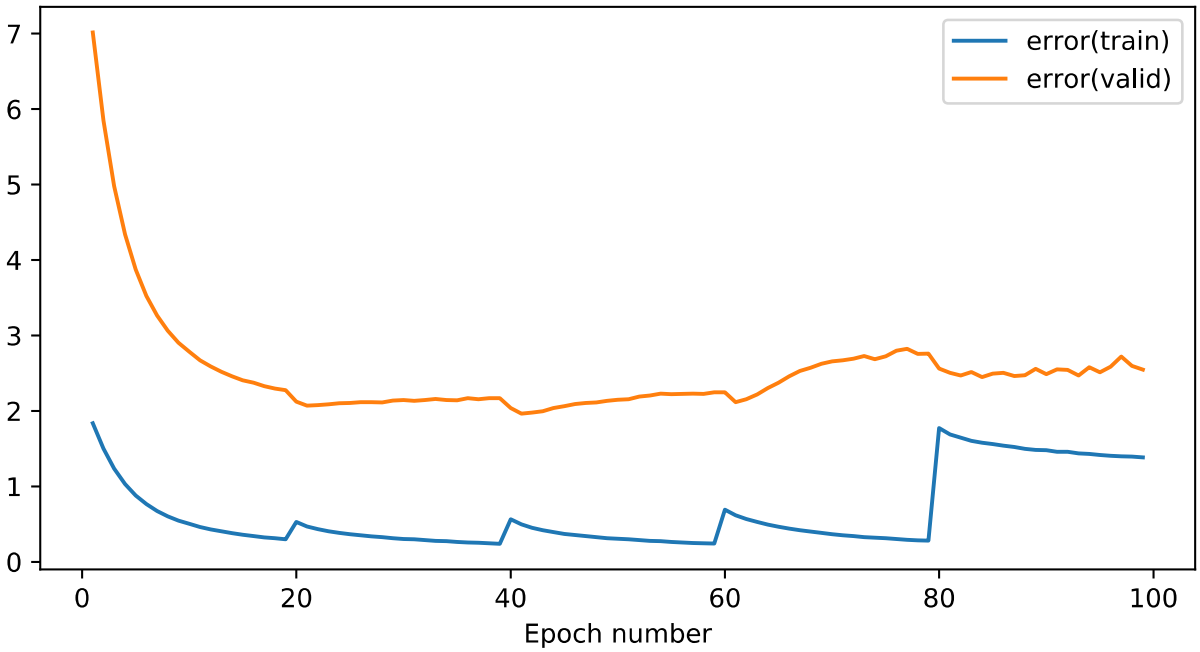
Number of Epochs: 100

Repeating School Class: False

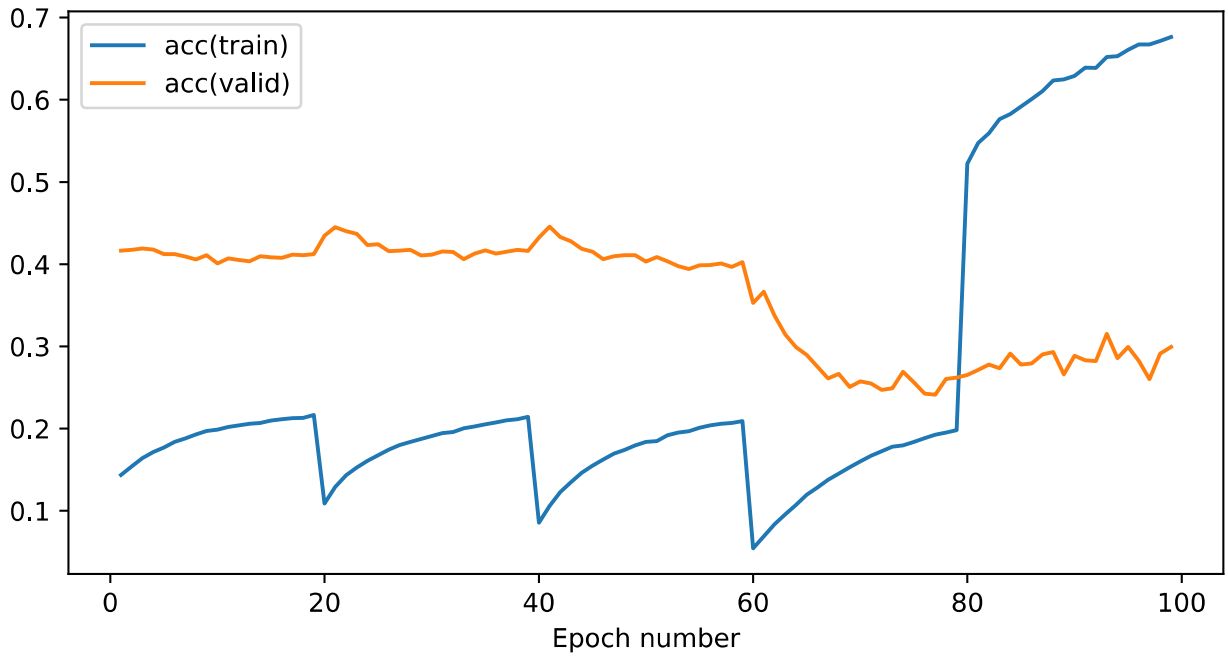
Shuffling Instances of Current Curriculum Level: True

Here we are repeating the above experiment but we are not iterating over the batches, of the same curriculum level, sequentially but rather we are shuffling them on every epoch.

Results



Plot 7: Training & Validation Error – Curriculum Learning – Curriculum Step: 200 – Repetitions: 20 – Number of Epochs: 100 – Repeating School Class: False – Shuffling Instances of Current Curriculum Level: True



Plot 8: Training & Validation Accuracy – Curriculum Learning – Curriculum Step: 200 – Repetitions: 20 – Number of Epochs: 100 – Repeating School Class: False – Shuffling Instances of Current Curriculum Level: True

Conclusions

We see that shuffling the instances plays a vital role on generalizing because with only shuffling we are able to see a slight increase of the validation accuracy the last 20 epochs where all the instances are present in the training process.

However the overall conclusion from the last two experiments where the repeat school class parameter is set to false and not all instances are present it looks to perform even worse than before. The lack of instances finally result in a worse case for the trained model. We will avoid setting this parameter to False in next experiments.

Curriculum Learning Experiment 5

Curriculum Step: 200

Repetitions: 5

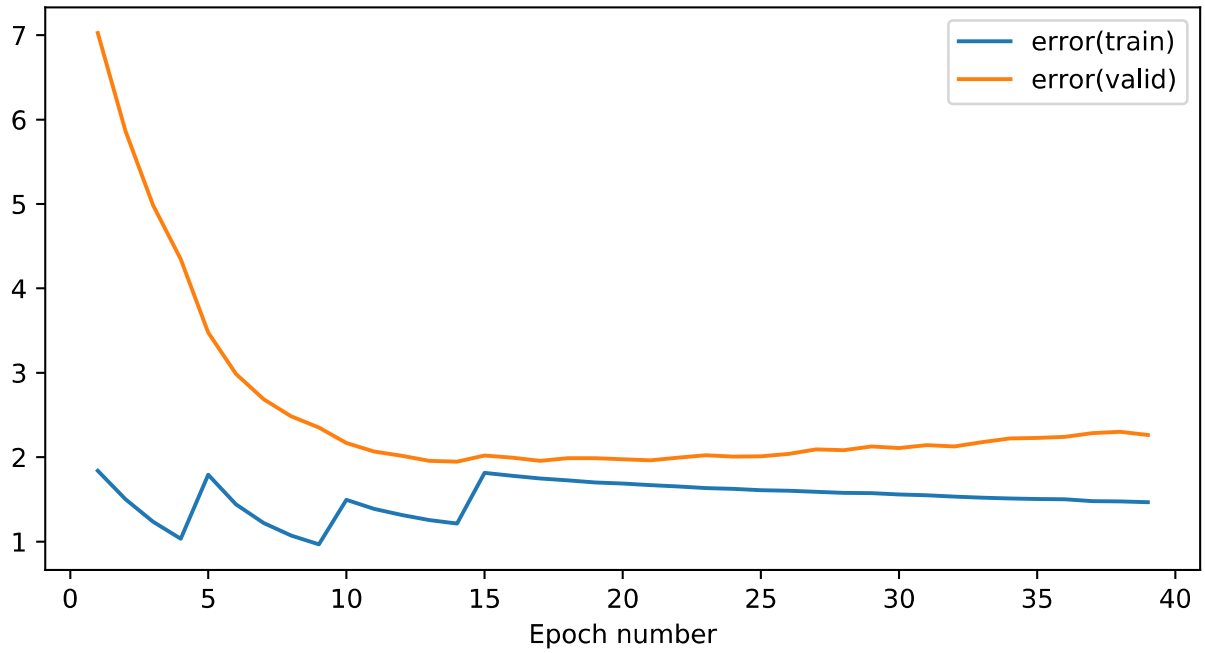
Number of Epochs: 40

Shuffling Instances of Current Curriculum Level: True

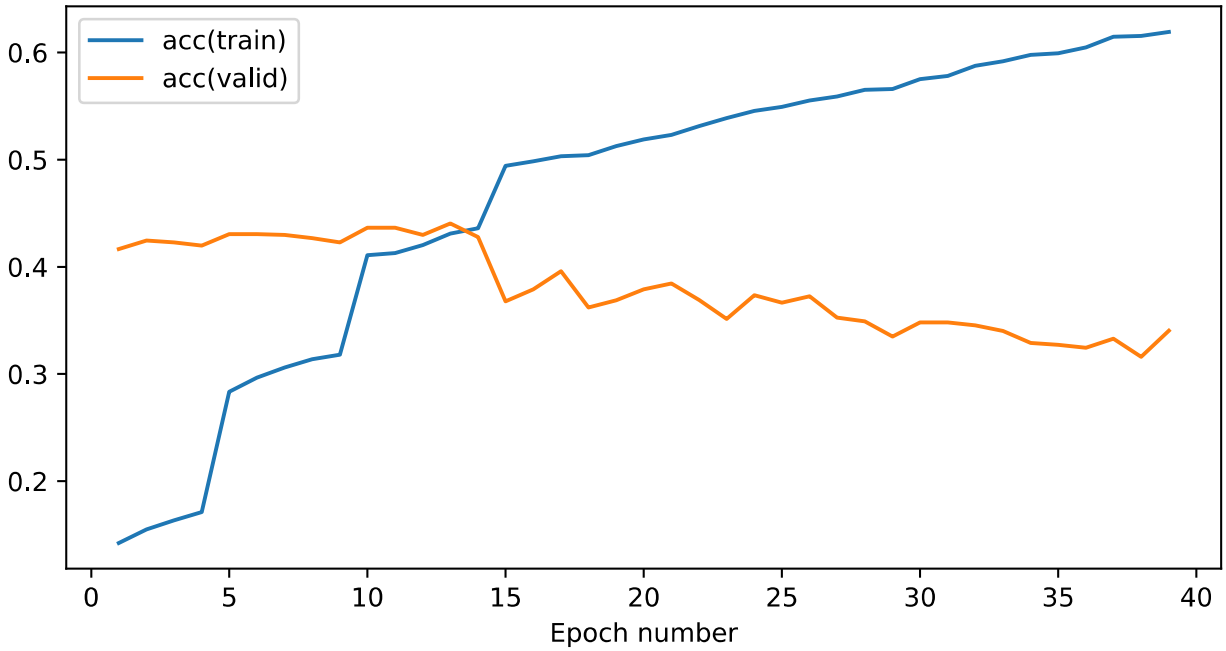
Now we are turning the **Repeating School Class** parameter back to its default which is **True** but we keep shuffling enabled.

Note that we try to also reduce the number of repetitions as we hypothesize that too many repetitions might lead to overfitting to the instances that correspond to a particular curriculum level.

Results



Plot 9: Training & Validation Error – Curriculum Learning – Curriculum Step: 200 – Repetitions: 5 – Number of Epochs: 40 - Shuffling Instances of Current Curriculum Level: True – Repeating School Class: True



Plot 10: Training & Validation Accuracy – Curriculum Learning – Curriculum Step: 200 – Repetitions: 5 – Number of Epochs: 40 - Shuffling Instances of Current Curriculum Level: True – Repeating School Class: True

Conclusions

Trying the experiment within these fewer epochs does not yield as bad results as before as the validation accuracy remains above 30% for the most part. However the same behavior we have seen in previous experiments is repeated here with the drop of validation accuracy after introducing the songs that correspond to the most difficult curriculum level.

Curriculum Learning Experiment 6

Curriculum Step: 50

Repetitions: 10

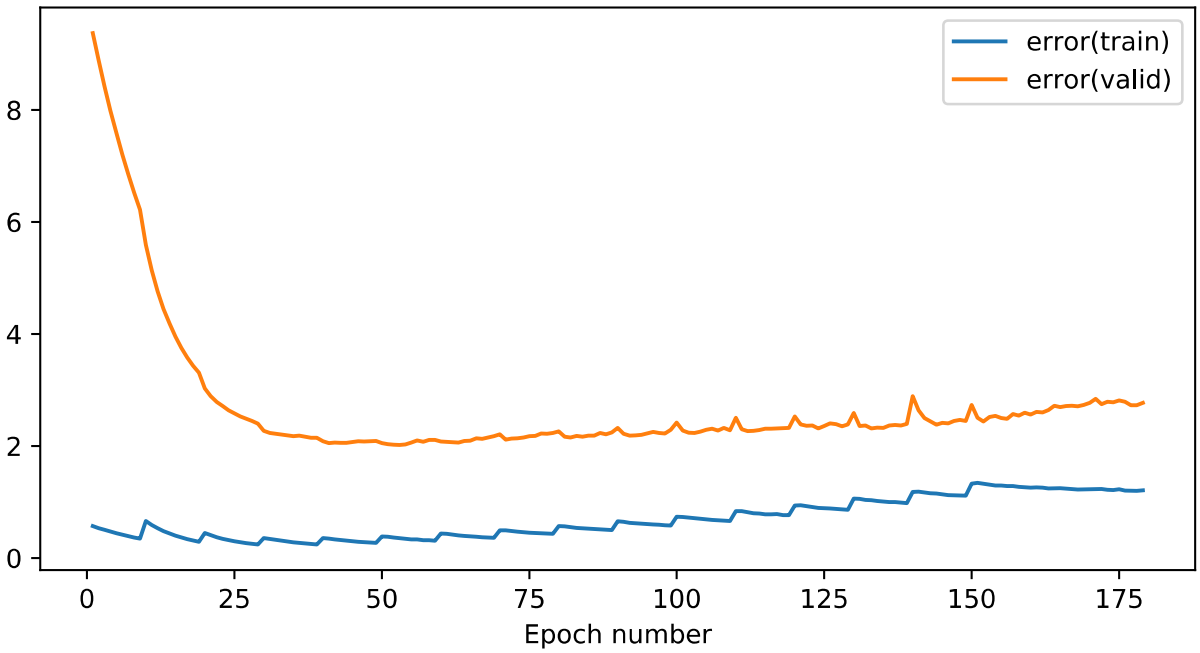
Number of Epochs: 180

Shuffling Instances of Current Curriculum Level: True

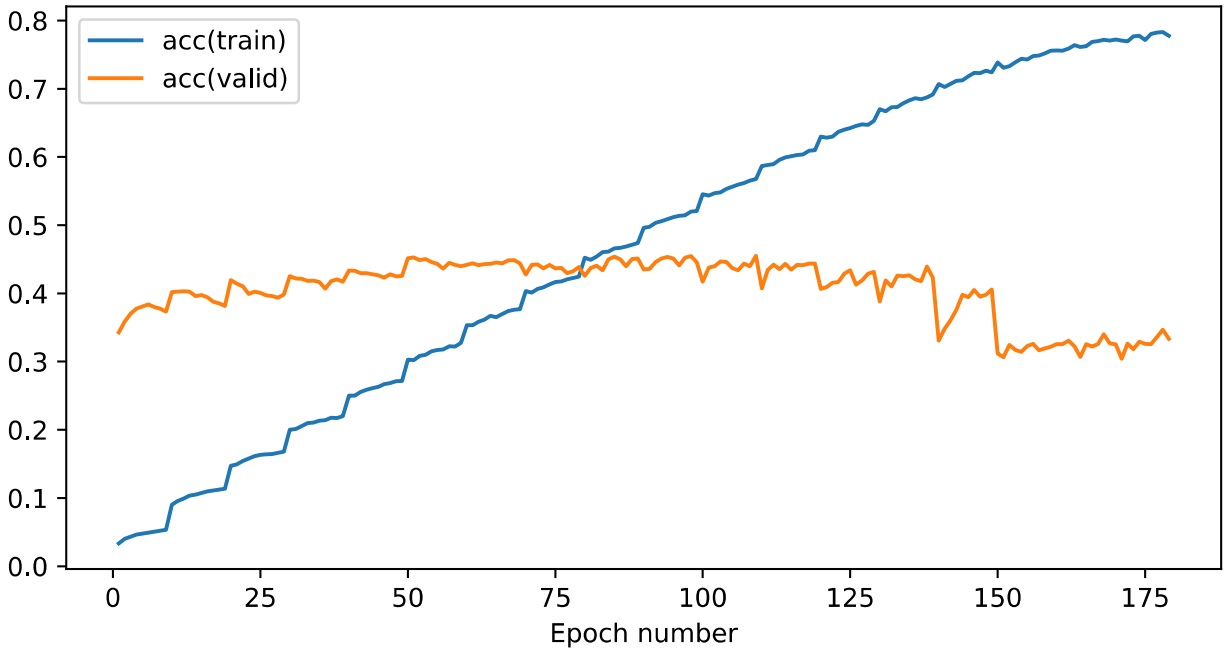
One thing we might give credit for the poor performance is that we might have set a very large curriculum step and we need to train the neural network more slowly by providing more curriculum levels. We do this by setting the curriculum step 4 times smaller than before which effectively creates a total of 16 curriculum levels.

We also increase the number of repetitions because we would like to see if any kind of negative effects we have seen before appear here as well. We feel that 5 epochs per curriculum level is not going to provide the necessary resolution of what is the current behavior.

Results



Plot 11: Training & Validation Error – Curriculum Learning – Curriculum Step: 50 – Repetitions: 10 – Number of Epochs: 180 – Shuffling Instances of Current Curriculum Level: True



Plot 12: Training & Validation Accuracy – Curriculum Learning – Curriculum Step: 50 – Repetitions: 10 – Number of Epochs: 180 – Shuffling Instances of Current Curriculum Level: True

Conclusions

The validation accuracy above peaks at 46% which is the first one from the experiments so far to reach at a higher level closer to the one we had achieved without curriculum learning.

However while achieving high accuracy with having considered only less than the total amount of songs we see that as new songs come along the validation accuracy slightly drops and when ever harder to classify songs come along the validation accuracy drops even further coming to similar results as the ones we saw in the experiments above.

Curriculum Learning Experiment 7

Curriculum Step: 50

Repetitions: 10

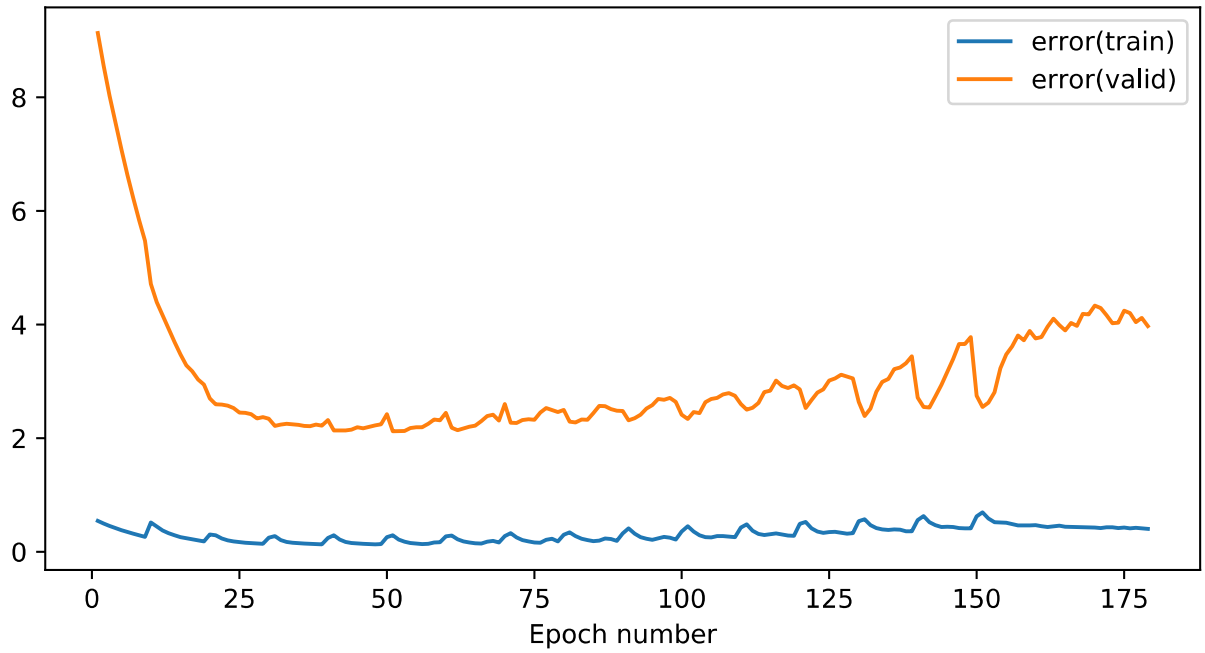
Number of Epochs: 180

Shuffling Instances of Current Curriculum Level: True

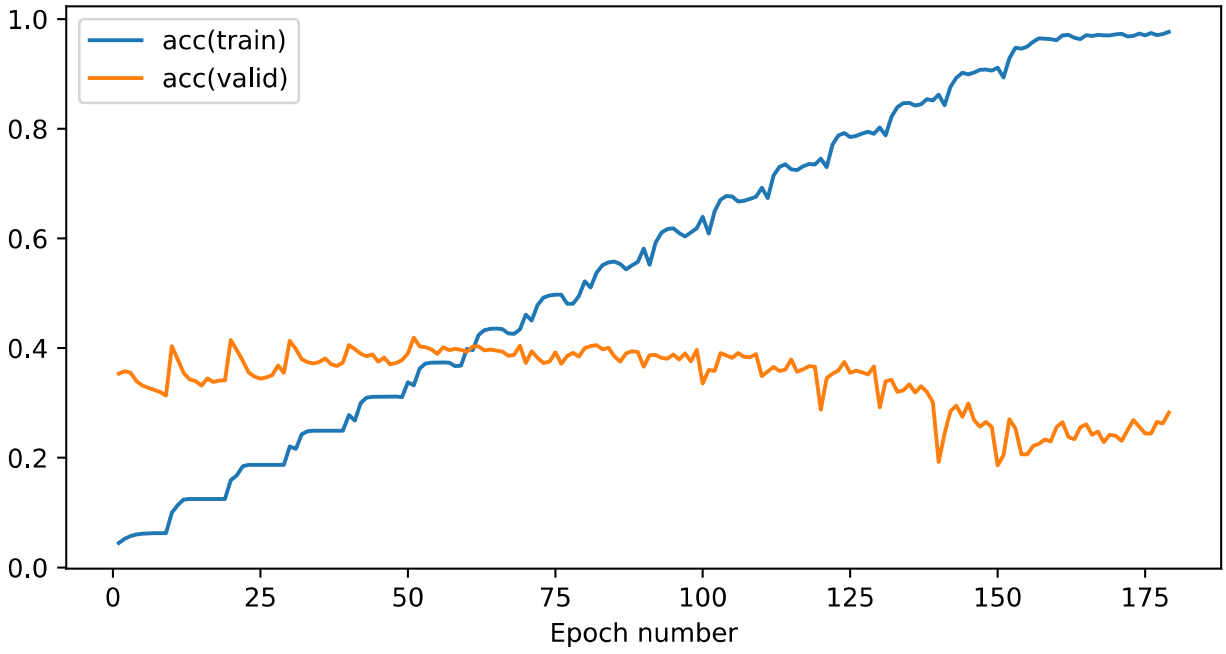
Dropout Keep Probabilities: Both 100%

We see here that both dropout probabilities are set to 100% which means that we have effectively disabled dropout. The role of this experiment is to see how the network behaves without dropout feature because dropout introduces some noise in the layers which might make it too hard to converge to anything useful within the few epochs provided for each curriculum level.

Results



Plot 13: Training & Validation Error – Curriculum Learning – Curriculum Step: 50 – Repetitions: 10 – Number of Epochs: 180 – Shuffling Instances of Current Curriculum Level: True – Dropout Keep Probabilities: Both 100%



Plot 14: Training & Validation Accuracy – Curriculum Learning – Curriculum Step: 50 – Repetitions: 10 – Number of Epochs: 180 – Shuffling Instances of Current Curriculum Level: True – Dropout Keep Probabilities: Both 100%

Conclusions

We can safely conclude comparing this and the previous experiment that dropout is necessary in curriculum learning. With lack of dropout overfitting effects are quite vivid even by leaving l2 regularization enabled. The final validation accuracy is much worse that we would expect.

Curriculum Learning Experiment 8

Curriculum Step: 50

Repetitions: 5

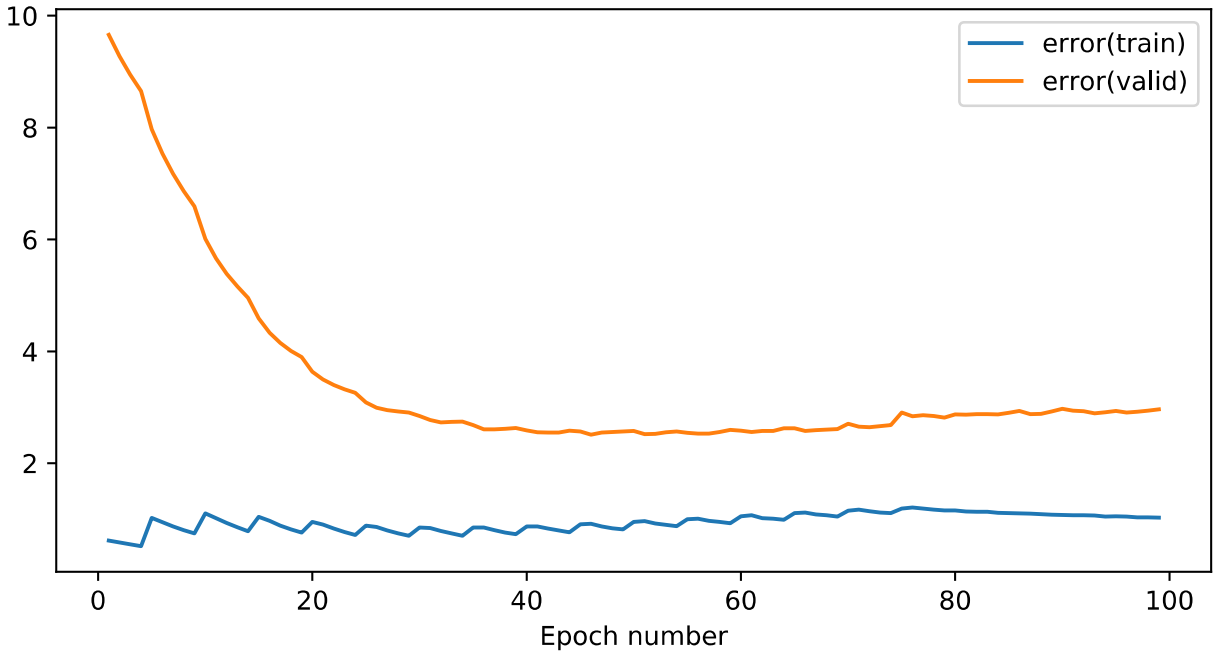
Number of Epochs: 100

Shuffling Instances of Current Curriculum Level: True

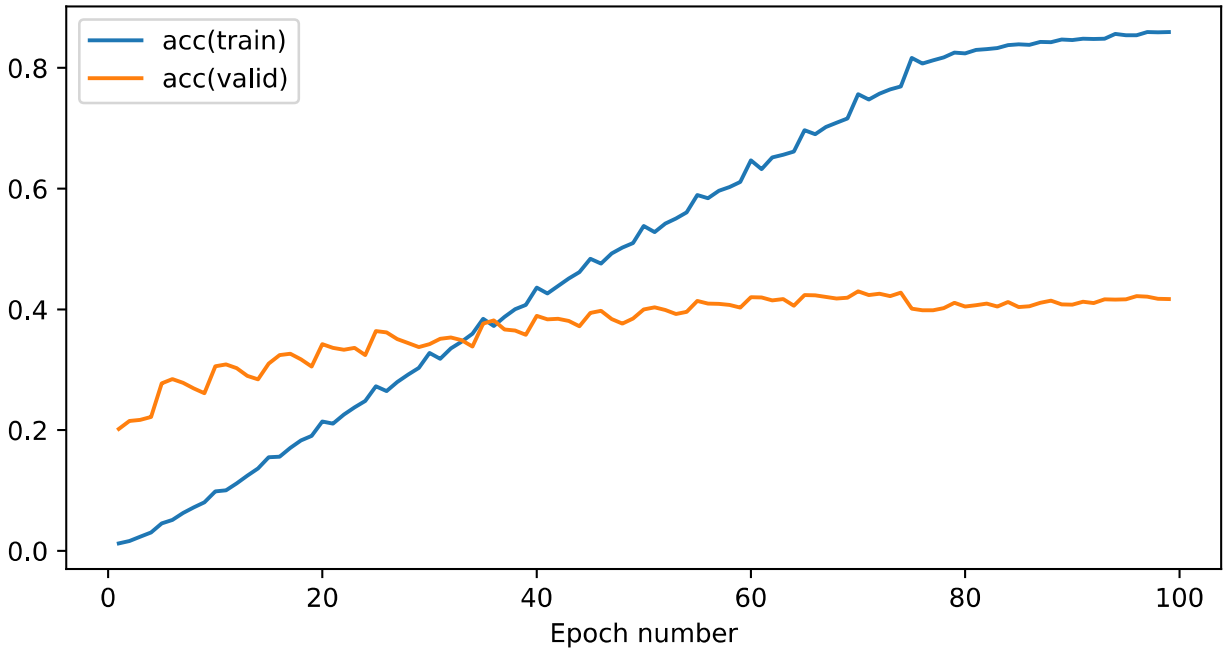
Dropout Keep Probabilities: Both 90%

Since Dropout is vital we are introducing again dropout regularization with relatively high keep probabilities of 90% both for the input and the hidden layers. We are also lowering the number of repetitions to not let the model train too much on a particular curriculum level but keep learning as new curriculum levels come along.

Results



Plot 15: Training & Validation Error – Curriculum Learning – Curriculum Step: 50 – Repetitions: 5 – Number of Epochs: 100 – Shuffling Instances of Current Curriculum Level: True – Dropout Keep Probabilities: Both 90%



Plot 16: Training & Validation Accuracy – Curriculum Learning – Curriculum Step: 50 – Repetitions: 5 – Number of Epochs: 100 – Shuffling Instances of Current Curriculum Level: True – Dropout Keep Probabilities: Both 90%

Conclusions

Note that in comparison to previous experiments our chosen hyperparameters are not too bad in creating a more stable system. The validation error has a small rising trend which suggests overfitting but only on a small scale. However in terms of generalization the model, even if perfoming steadily, it results in a low validation accuracy. The upside is that there are not so big drops on the validation accuracy as the new more difficult songs come along.

Self Paced

Experiments below neglect the repetitions parameter and they follow a self-paced approach. With this approach we do not set a specific number of epochs for the experiment to run but we stop the run at a curriculum level based on conditions. Here the condition is a simple one where the current validation error must be lower than the previous validation error. In other words any kind of sign that we are overfitting by having the validation error increase is considered the end of the current curriculum level and we go to the next one.

Curriculum Learning Experiment 9

Curriculum Step: 50

Number of Epochs: 94

Shuffling Instances of Current Curriculum Level: True

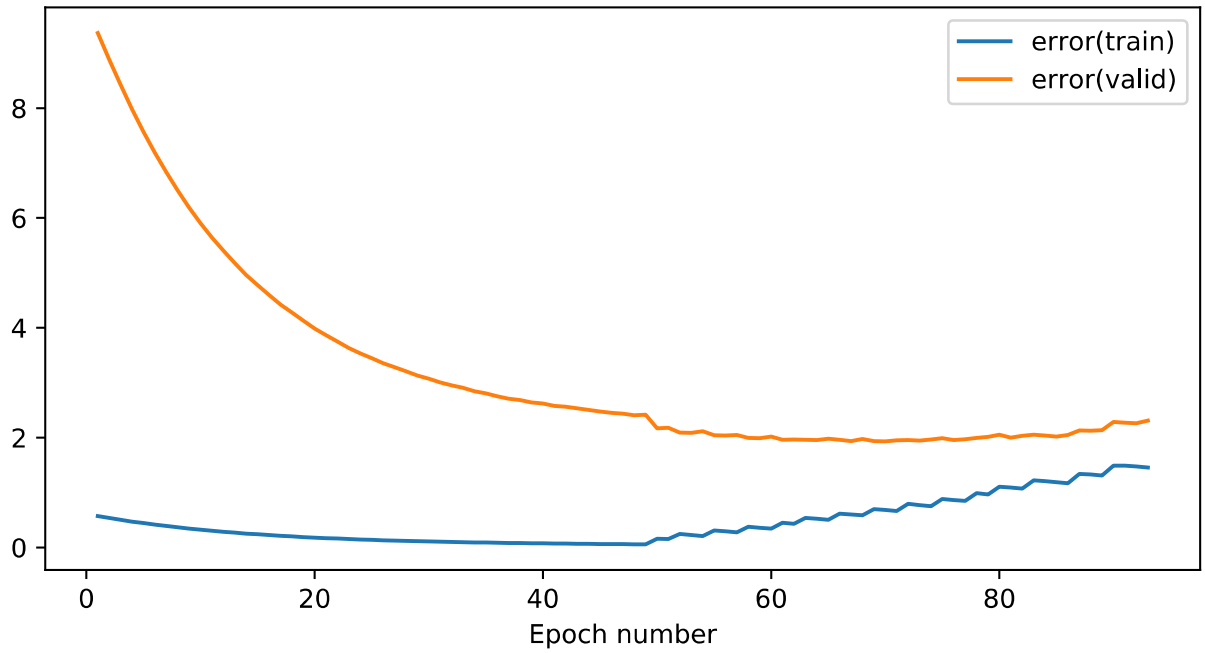
Dropout Keep Probabilities: Both 80%

Self Paced: True

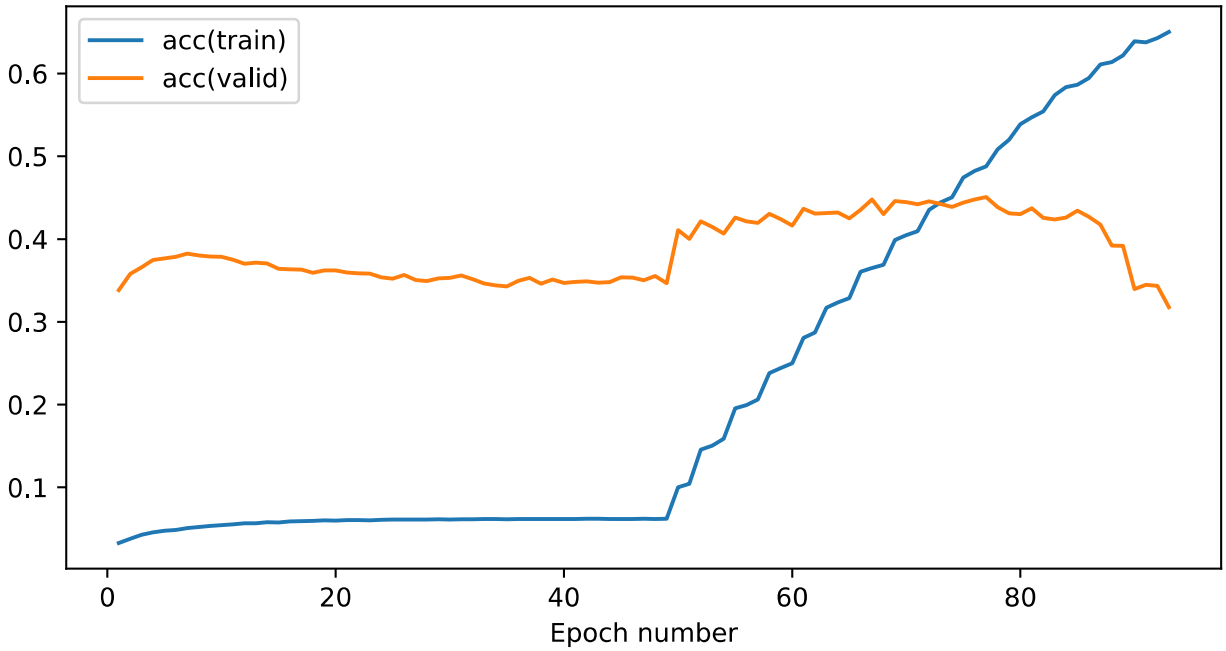
Dropout keep probability is being lowered closer to the values that were set for our baseline.

We want to see the behavior of self-paced curriculum learning.

Results



Plot 17: Training & Validation Error - Curriculum Learning - Curriculum Step: 50 – Number of Epochs: 94 – Shuffling Instances of Current Curriculum Level: True – Dropout Keep Probabilities: Both 80% – Self Paced: True



Plot 18: Training & Validation Accuracy - Curriculum Learning - Curriculum Step: 50 – Number of Epochs: 94 – Shuffling Instances of Current Curriculum Level: True – Dropout Keep Probabilities: Both 80% – Self Paced: True

Conclusions

Note how the system trains on the first curriculum level for several epochs before it starts increasing the curriculum level. Within this period, the first ~50 epochs, even though validation error has dropped significantly there is also a drop in the validation accuracy which means that the system is not particularly getting better.

We can assume that different magnitudes of regularization are required for different levels of the curriculum level. So even being self-paced we are considering that the problem is getting a lot harder because we need either a

smaller dropout keep probability to introduce more artificial noise in the inputs and this dropout keep probbility must be just right, not above or below, or altenatively we need to dynamically set the dropout probabilities and/or the L2 regularization factor.

Summarized Conclusions for Curriculum Learning for Experiments 1-9

So far we have not managed to achieve any metric better with curriculum learning in comparison with previous experiments. So it seems that at the end of the training the neural network is encountered with the most difficult songs, songs it has trouble to classify. These songs cause the largest changes in the variables of the neural network and this may be the cause of a gradient explosion, even if this is short term.

Another way to look at what curriculum learning is trying to achieve is to see it as pretraining. What actually is happening is that we have a good enough trained network with the first group of easy instances and then the network is pretrained in a way that theoritically would be better prepared for the most difficult instances that come along. The key here is the phrase “good enough trained network”. Because it seems that our originally globally optimized hyperparameters as the L2 regularization factor and the Dropout keep probabilities were chosen for another network. Also we do not have an easy way of dynamically changing the hyperparameters as we go along. To be honest we tried this approach in coursework 3 with our simple dynamic dropout algorithm but this did not result in a more stable system.

So we hypothesize that curriculum learning could in fact bring better results but only if in every curriculum level we could achieve a generalized model that would classify unseen data with the same accuracy or better than the previous curriculum level.

(Reverse) Curriculum Learning Experiment 10

Reverse Order: True

Curriculum Step: 50

Number of Epochs: 109

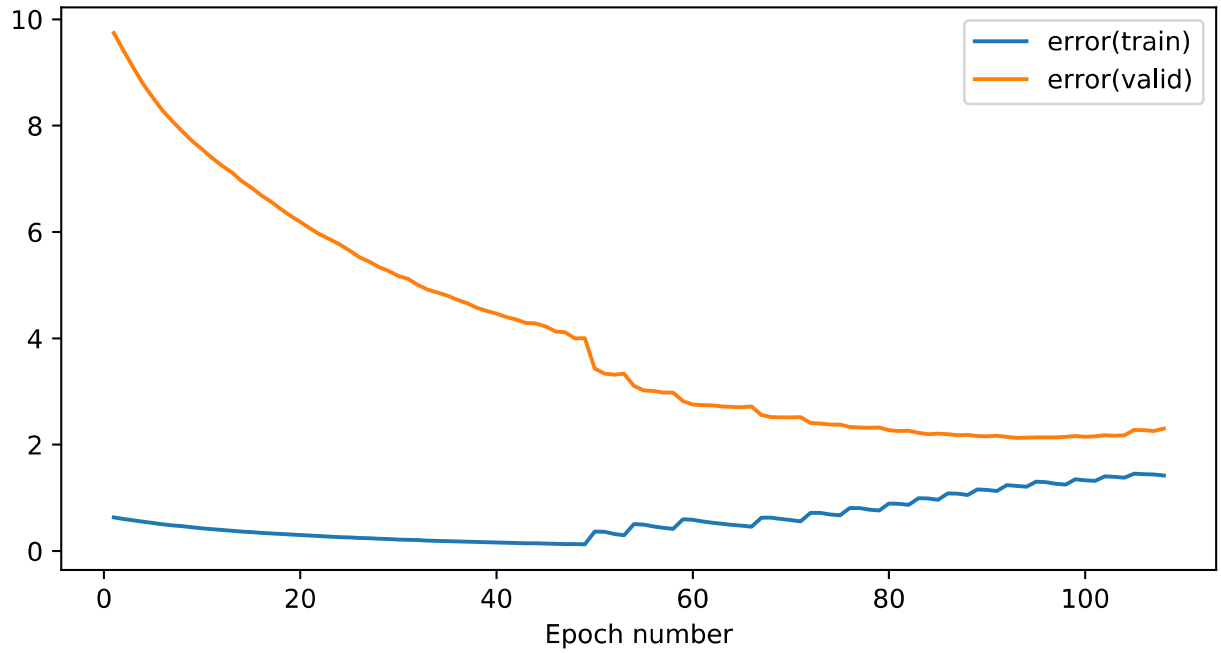
Shuffling Instances of Current Curriculum Level: True

Dropout Keep Probabilities: Both 80%

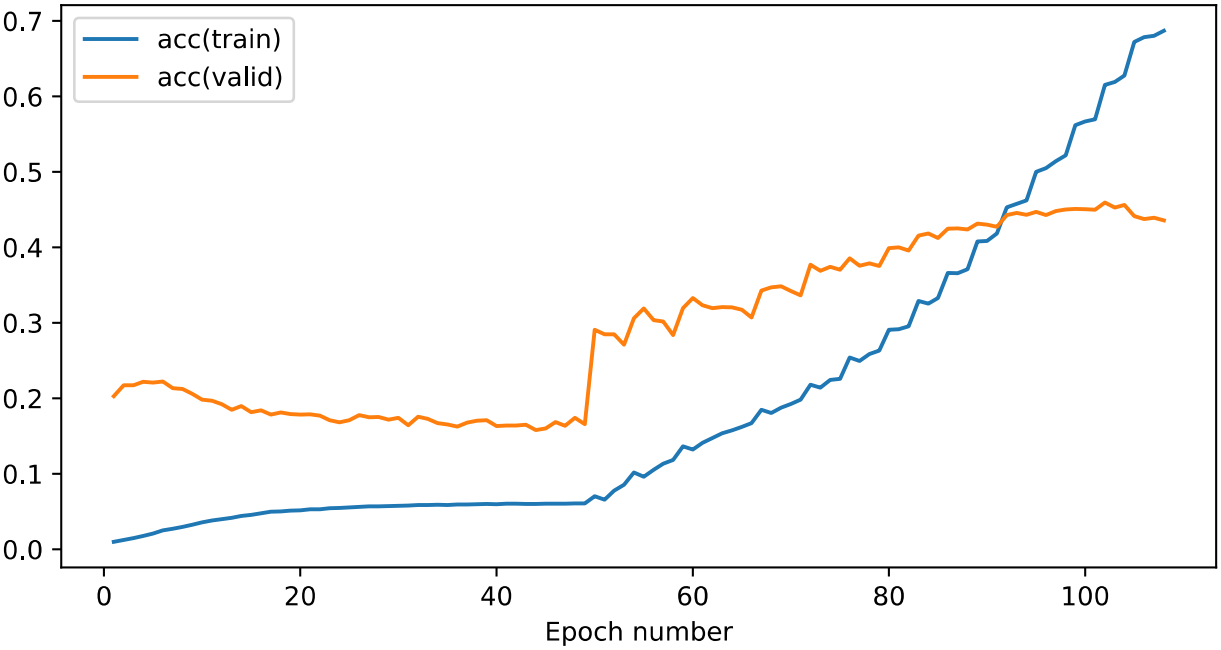
Self Paced: True

Here we are looking curriculum learning from another perspective. We are **reversing the entire procedure** by starting learning from the most difficult instances and moving towards the easier instances. In a sense this could be seen to have similarities with **annealing dropout** because we are starting from instances which are very noisy and we are gradually moving towards instances which are less noisy and easier for our classifier.

Results



Plot 19: Training & Validation Error – Curriculum Learning – Reverse Order: True – Curriculum Step: 50 – Number of Epochs: 109 – Shuffling Instances of Current Curriculum Level: True – Dropout Keep Probabilities: Both 80% – Self Paced: True



Plot 20: Training & Validation Accuracy – Curriculum Learning – Reverse Order: True – Curriculum Step: 50 – Number of Epochs: 109 – Shuffling Instances of Current Curriculum Level: True – Dropout Keep Probabilities: Both 80% – Self Paced: True

Conclusions

From a first look just by reversing the order of the curriculum we have better results in the experiments because we have an almost constant rising trent at the validation accuracy and an almost constant descreasing trend at the validation error. In addition we have achieved a peak at the validation accuracy of 46% which is relatively a good

validation accuracy even though not better than what we have already achieved in previous experiments to make this process of reversed curriculum learning worthwhile.

(Reverse) Curriculum Learning Experiment 11

Reverse Order: True

Curriculum Step: 5

Number of Epochs: 701

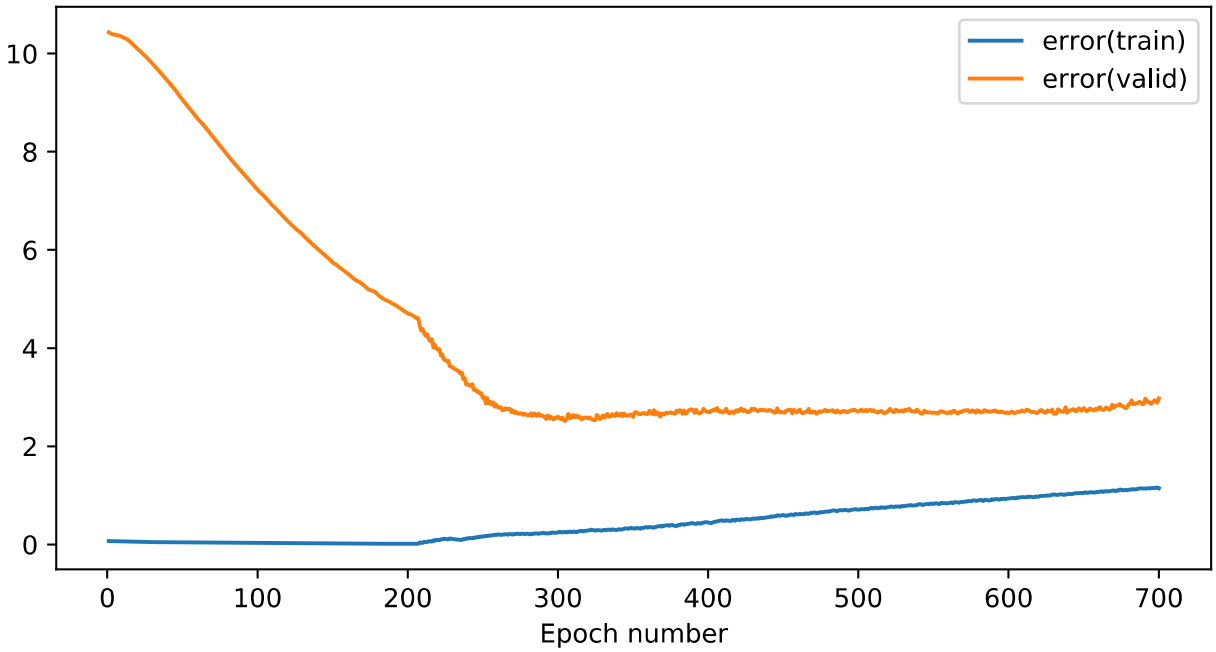
Shuffling Instances of Current Curriculum Level: True

Dropout Keep Probabilities: 70% for input dropout keep probability, 83.7125% for dropout keep probability of hidden layers

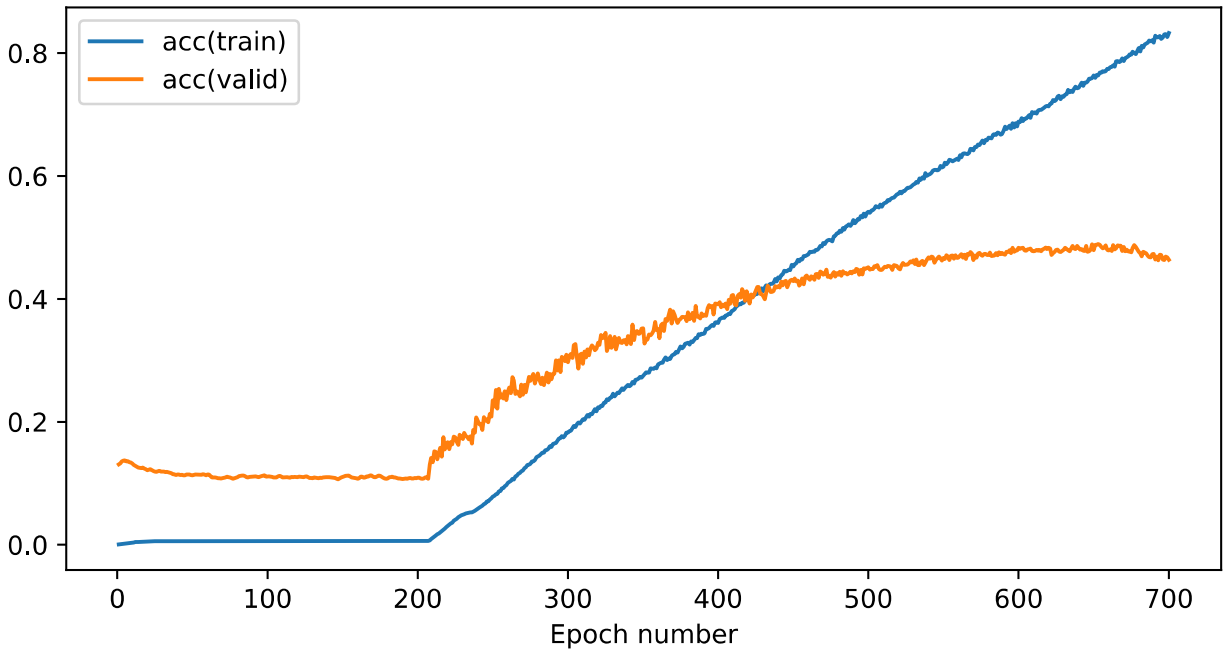
Self Paced: True

Because of the promising results of the previous experiment here we are considering a very similar experiment where the dropout keep probabilities have been set as were the original ones on our baseline classifier. The main difference though is that we have set a much smaller curriculum step of only five(5). This means that the process is going to be much slower and the entire process lasted for 701 epochs.

Results



Plot 21: Training & Validation Error – Curriculum Learning – Reverse Order: True – Curriculum Step: 5 – Number of Epochs: 701 – Shuffling Instances of Current Curriculum Level: True – Dropout Keep Probabilities: 70% for input dropout keep probability, 83.7125% for dropout keep probability of hidden layers – Self Paced: True



Plot 22: Training & Validation Accuracy – Curriculum Learning – Reverse Order: True – Curriculum Step: 5 – Number of Epochs: 701 – Shuffling Instances of Current Curriculum Level: True – Dropout Keep Probabilities: 70% for input dropout keep probability, 83.7125% for dropout keep probability of hidden layers – Self Paced: True

Conclusions

Following a more thorough approach where the curriculum step was quite smaller the results are better than before since for several epochs the validation accuracy peaked at 49% which is equivalent to what we had achieved with our baseline classifier. The drawback is that this process was much slower and it did not achieve better results to make it more attractive in any way.

Unfortunately still we lack either the time resources or the computational resources to be able and properly regularize our neural network, in a brute force way, at every curriculum step and thus provide the best possible result.

On the other hand it is difficult to say if we have reached the global optimum for our MLP classifier and blame that of reaching our upper bound of classification performance with MLP architecture.

Since songs are series of time depended segments then it makes sense to try an architecture which takes advantage of time dependencies such as the Recurrent Neural Network.