SongRNN: Music Generation Through a Character Level LSTM

Brandon Szeto
Jacobs School of Engineering
University of California, San Diego
San Diego, CA 92122
bszeto@ucsd.edu

Darren Yu
Jacobs School of Engineering
University of California, San Diego
San Diego, CA 92122
dmyu@ucsd.edu

Nathaniel Thomas
Jacobs School of Engineering
University of California, San Diego
San Diego, CA 92122
nathomas@ucsd.edu

Abstract

With a set of musical samples in ABC notation, we can generate music with a character-level LSTM. We can accomplish this task by feeding our LSTM musical characters in a sequence, and the network would slowly learn to predict the upcoming sequence of notes. Once trained, our model could generate music samples based on a short sequence of prompted notes. The style of the music generated by our final model has a rich, mature, and elaborate feel. By modifying the network depth, layer size, and dropout rate, we could tune our model to generate a range of musical samples. After hyperparameter tuning, we found that our LSTM created more elegant musical samples than a traditional recurrent neural network. The ability of the LSTM to drop information allows it to resemble the original test samples better than a standard RNN.

Preprint. Under review.

Introduction

Testing123

Related Works

U-Net

U-Net Ronneberger et al. [2015] has proven to be highly effective for biomedical image segmentation tasks. Its architecture features a contracting path to capture context and an expansive path for precise localization. The innovative use of skip connections facilitates the flow of fine-grained details across different layers, enhancing the model's ability to accurately delineate object boundaries.

ERFNet

ERFNet Romera et al. [2018] introduces a novel factorized convolutional layer that significantly reduces the number of parameters while maintaining expressive power. This reduction in parameters enables faster inference without compromising performance, making it well-suited for deployment in resource-constrained environments.

ResNet

ResNet He et al. [2015] addresses the challenges of training very deep neural networks by employing residual connections. These connections enable the direct flow of information across layers, mitigating the vanishing gradient problem and facilitating the training of extremely deep networks.

Relevance to our model

We test the performance of the above models on the image segmentation task using the PASCAL VOC-2007 dataset. We look for differences in the models and their relative performance in comparison to the basic fully connected network that we started with.

Methods

Baseline

Our baseline architecture consists of an encoder, decoder, classifier, and activation. We used the Adam gradient descent optimizer

Encoder

We have five convolutional layers that increases the depth of the orginal 3 channels to 32 to 64 to 128 to 256 to 512 each using a size 3 kernel, padding of 1, a stride of 2, and no dilation. Each layer sees a small decrease in the height and width of the layer according to the expression $\frac{W-F+2P}{2}+1$. The outputs of each convolutional layer are subsequently passed through a ReLU activation function and a batch normalization layer.

Decoder

We have five deconvolutional, or upsampling layers that decreases the depth of the final 512 deep layer output from the encoder. This is decreases from 512 to 256 to 128 to 64 to 32 each using a size 3 kernel, padding of 1, a stride of 2, and no dilation. Each layer sees a small decrease in the height and width of the layer according to the expression $S(W-1)+F-2P$. Similarly, the outputs of each deconvolutional layer are subsequently passed through a ReLU activation function and a batch normalization layer.

Classifier and Activation

In our final layer, we have a 1×1 convolutional kernel working as a classifier, and a softmax activation layer. This layer projects a probability distribution stream over the 21 classes.

Improvements Over Baseline

Data Augmentation

To enhance the robustness of our model, we applied data augmentation techniques to our dataset. This involved performing various transformations on the input images, such as mirror flips, rotations, and crops. During the process, we must ensure that the same transformations are applied to the corresponding labels to maintain data integrity throughout the augmentation process. We found that data augmentation improved on the average Jaccard index on our models by around 0.03-0.05.

Cosine Annealing

In order to optimize the learning rate dynamically throughout the training process, we implemented the cosine annealing learning rate scheduler. This technique adjusts the learning rate in a cosine-shaped manner, effectively annealing it towards zero as training progresses. By aligning the learning rate adjustments with the number of epochs, we aim to improve the convergence and generalization capabilities of our model. Training our models with cosine annealing had a minimal improvement in both our accuracy and Jaccord index.

Class Imbalance

To mitigate the challenges posed by class imbalance, particularly addressing rare classes, we employed strategies to alleviate this issue. One approach is to apply a weighted loss criterion, which assigns higher weights to the infrequent classes during the optimization process. By doing so, we incentivize the network to pay more attention to these underrepresented classes, thus improving its ability to accurately classify them. Otherwise, the model could simply learn to label the entire image the background and still achieve decent pixel accuracy. The introduction of weight decay had a considerable improvement in our Jaccard index and was used in all our models moving on.

Experimentation

In this work, we introduce DarrenNet, a novel architecture inspired by the erfnet architecture but distinguished by its enhanced efficiency and superior performance. Leveraging a modified erfnet structure, DarrenNet incorporates additional convolution layers and employs higher dropout, resulting in a more intricate and efficient network. The augmented convolutional layers contribute to a deeper understanding of spatial features, while increased dropout aids in regularization, preventing overfitting. The fusion of these modifications yields a model that not only operates more efficiently but also achieves superior performance in various tasks. We are also able to use pretrained erfnet encoders, which improve our performance further.

Below are the descriptions (in table format) and regularization techniques used for each architecture.

UNet Techniques

The following techiques were used on UNet: Kaiming weight initialization, frequency weight penalty, cosine annealing, and vertical/horizontal augmentation.

Results

Baseline FCN

Our baseline FCN resulted in a 0.06 IOU and a 75% pixel accuracy on the test set.

FCN with augmentation

Our FCN with augmentation resulted in a 0.06 IOU and a 72% pixel accuracy on the test set. We noticed that the loss convergence was slower.

FCN with Cosine Annealing LR

Our FCN with Cosine Annealing LR resulted in a 0.05 IOU and a 74% pixel accuracy on the test set.

FCN with Custom Class Weights

Our FCN with custom class weights resulted in a slightly worse 0.04 IOU and a 67% pixel accuracy on the test set.

DarrenNet

In Figure **??**, we observe the outcomes of DarrenNet. The model has a respectable IOU (0.05) and accuracy (75%), while maintaining very fast training speeds.

In Figure **??**, we observe the outcomes of DarrenNet with transfer learning using resnet34's encoder. The model leverages pre-trained weights, demonstrating superior performance on the test set. The transfer learning strategy significantly boosts both IOU (0.10) and accuracy (78%), indicating that the network effectively transfers knowledge from the source domain to the target domain.

Figure **??** represents the results of DarrenNet with both transfer learning and data augmentation. This combination appears to be highly effective, as the model achieves impressive segmentation results on the test set. The incorporation of augmented data during training, along with knowledge transfer, contributes to enhanced generalization capabilities.

Figure **??** displays the results of the DarrenNet with Augment Affine and transfer learning. The model exhibits strong performance on the test set, achieving a high Intersection over Union (IOU) at 0.15 and 82% accuracy. The augmentation techniques, particularly affine transformations, seem to enhance the model's ability to generalize well to unseen data, resulting in improved segmentation accuracy.

The UNet architecture's results are depicted in Figure **??**. The model demonstrates below average segmentation performance, achieving poor IOU (0.05) and accuracy (0.55) on the test set.

Discussion

Baseline

The baseline model, though straightforward to implement, exhibited limitations in segmentation performance with an IOU of 0.06 and a pixel accuracy of 75%. The simplicity of the architecture hindered its ability to capture intricate features in the data.

Improved Baseline (FCN with Augmentation)

In an effort to address overfitting, the baseline model was enhanced with data augmentation. Despite these improvements, the model faced challenges with slower loss convergence and only achieved a marginal increase in performance, resulting in an IOU of 0.06 and a decreased pixel accuracy of 72%.

DarrenNet:

DarrenNet, incorporating advanced architectural features and augmentation techniques such as Affine transformations, displayed notable improvements with an IOU of 0.15 and a pixel accuracy of 82%. The architecture and augmentation strategy played pivotal roles in enhancing segmentation results.

Transfer Darrennet:

The application of transfer learning with DarrenNet, utilizing pre-trained weights from resnet34, demonstrated significant performance gains with an IOU of 0.10 and a pixel accuracy of 78%. Successful knowledge transfer from the source domain contributed to the improved segmentation performance.

UNet:

The UNet architecture, characterized by its ability to capture detailed spatial information, showed below-average baseline performance with an IOU of 0.05 and a pixel accuracy of 55%. However, when integrated with transfer learning, UNet outperformed the baseline, achieving an IOU of 0.08 and a pixel accuracy of 79%.

Performance Differences (Between Implementations):

Comparative analysis revealed that DarrenNet consistently outperformed the baseline, showcasing the significance of architectural enhancements. Transfer learning, as evidenced in both Transfer DarrenNet and UNet with transfer, played a critical role in improving segmentation results, emphasizing the importance of leveraging pre-trained weights for feature extraction.

Insights from Loss Curves, Tables, Visualizations:

Examination of loss convergence in the improved baseline highlighted challenges in augmentation effectiveness. Visualizations, loss curves, and performance metrics provided insights into the impact of architectural choices and transfer learning strategies, offering valuable information for further model refinement.

Contributions

Brandon Szeto: Data evaluation, loss graphs, diagrams, write up (methods, results, and discussion).

Darren Yu: RNN hyperparameter tuning, feature evaluation, and write up (abstract, introduction, and related work),

Nathaniel Thomas: SongRNN, model architecture, model training, and music generation.

References

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

Eduardo Romera, José M. Álvarez, Luis M. Bergasa, and Roberto Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation, 2018.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.