Δημιουργία Μουσικής με χρήση Deep Learning

Κωνσταντίνος Σκουρογιάννης
kostskouros@outlook.com

Ηρακλής Παλληκάρης
iraklis.pallikaris@gmail.com

Γενικά

Αυτή η εργασία στοχεύει στη χρήση τεχνικών Deep Learning για τη δημιουργία μουσικής. Η βασική μας μέθοδος χρησιμοποιεί μια αρχιτεκτονική με transformer για να εκτελέσει sequence generation χρησιμοποιώντας next token prediction. Χρησιμοποιώντας την ίδια ιδέα του next token generation, εφαρμόσαμε ένα μοντέλο RNN για να προβλέψουμε την επόμενη νότα με μια ακολουθία εισόδου για ένα κομμάτι και ένα κομμάτι πέντε οργάνων. Εφαρμόσαμε επίσης ένα μοντέλο CNN για να δημιουργήσουμε μια πιο καλοσχηματισμένη διάταξη μεταξύ των πέντε οργάνων χρησιμοποιώντας τόσο ένα CNN μελωδίας όσο και ένα CNN αρμονίας υπό όρους. Για να ενσωματώσουμε περισσότερη τυχαιότητα στην παραγόμενη έξοδο, χρησιμοποιήσαμε ένα VAE για να κωδικοποιήσουμε τις ακολουθίες σε ένα latent space και προσθέσαμε θόρυβο για να δημιουργήσουμε παραλλαγή στην παραγόμενη έξοδο. Τέλος, προσπαθήσαμε να χρησιμοποιήσουμε ένα GAN, αλλά δυσκολευτήκαμε στην εκπαίδευση του μοντέλου και αντιμετωπίσαμε κατάρρευση λειτουργίας. Συνολικά, βρήκαμε τη μεγαλύτερη επιτυχία στο μοντέλο VAE μας, με τα αποτελέσματα αυτού του μοντέλου να είναι τα πιο καλά διαμορφωμένα και να παρουσιάζουν σημαντικές διαφορές.

Εισαγωγή

Το Deep Learning έχει μεταμορφώσει ριζικά τα πεδία του computer vision και της επεξεργασίας φυσικής γλώσσας, όχι μόνο σε classification αλλά και generation tasks, επιτρέποντας τη δημιουργία απίστευτα ρεαλιστικών εικόνων καθώς και τεχνητά δημιουργημένων ειδήσεων. Τι γίνεται όμως με τον τομέα του ήχου – ή πιο συγκεκριμένα – της μουσικής; Σε αυτή την εργασία, στοχεύουμε να δημιουργήσουμε νέες αρχιτεκτονικές νευρωνικών δικτύων για τη δημιουργία νέας μουσικής, χρησιμοποιώντας 20.000 δείγματα MIDI διαφορετικών ειδών από το Lakh Piano Dataset, ένα δημοφιλές σύνολο δεδομένων αναφοράς για πρόσφατες εργασίες παραγωγής μουσικής.

Ιστορικό

Η παραγωγή μουσικής με τη χρήση τεχνικών DL είναι ένα θέμα ενδιαφέροντος τις τελευταίες δύο δεκαετίες. Η μουσική αποδεικνύεται μια διαφορετική πρόκληση σε σύγκριση με τις εικόνες, μεταξύ τριών βασικών διαστάσεων: Πρώτον, η μουσική είναι χρονική, με ιεραρχική δομή με εξαρτήσεις διαχρονικά. Δεύτερον, η μουσική αποτελείται από πολλαπλά όργανα που αλληλεξαρτώνται και ξεδιπλώνονται με την πάροδο του χρόνου. Τρίτον, η μουσική ομαδοποιείται σε συγχορδίες, arpeggio και μελωδίες – επομένως κάθε time-step μπορεί να έχει πολλαπλές εξόδους.

Ωστόσο, τα ηχητικά δεδομένα έχουν πολλές ιδιότητες που τα καθιστούν εξοικειωμένα κατά κάποιο τρόπο με αυτό που μελετάται συμβατικά στη βαθιά μάθηση (όραση υπολογιστή και επεξεργασία φυσικής γλώσσας ή NLP). Η διαδοχική φύση της μουσικής μας θυμίζει το NLP, για το οποίο μπορούμε να

χρησιμοποιήσουμε τα επαναλαμβανόμενα νευρωνικά δίκτυα. Υπάρχουν επίσης πολλά «κανάλια» ήχου (όσον αφορά τους τόνους και τα όργανα), που θυμίζουν εικόνες για τις οποίες μπορούν να χρησιμοποιηθούν Συνελικτικά Νευρωνικά Δίκτυα. Επιπλέον, τα βαθιά παραγωγικά μοντέλα είναι συναρπαστικοί νέοι τομείς έρευνας, με τη δυνατότητα δημιουργίας ρεαλιστικών συνθετικών δεδομένων. Μερικά παραδείγματα είναι Variational Autoencoders (VAE) και Generative Adversarial Neworks (GAN), καθώς και μοντέλα γλώσσας στο NLP.

Οι περισσότερες αρχικές τεχνικές παραγωγής μουσικής έχουν χρησιμοποιήσει επαναλαμβανόμενα νευρωνικά δίκτυα (RNN), τα οποία φυσικά ενσωματώνουν εξαρτήσεις διαχρονικά. Ο Skuli (2017) χρησιμοποίησε LSTM για να δημιουργήσει μουσική με ένα όργανο με τον ίδιο τρόπο όπως τα γλωσσικά μοντέλα. Αυτή η ίδια μέθοδος χρησιμοποιήθηκε από τον Nelson (2020), ο οποίος την προσάρμοσε για να δημιουργήσει μουσική lo-fi.

Πρόσφατα, τα Συνελικτικά Νευρωνικά Δίκτυα (CNN) χρησιμοποιήθηκαν για τη δημιουργία μουσικής με μεγάλη επιτυχία, με το DeepMind το 2016 να δείχνει την αποτελεσματικότητα του WaveNet, το οποίο χρησιμοποιεί διευρυμένες συνελίξεις για τη δημιουργία ακατέργαστου ήχου. Ο Yang (2017) δημιούργησε το MidiNet, το οποίο χρησιμοποιεί Deep Convolutional Generative Adversarial Networks (DCGANs) για να δημιουργήσει μουσικές ακολουθίες πολλών οργάνων που μπορούν να εξαρτηθούν τόσο από τη μουσική της προηγούμενης γραμμής, όσο και από τη συγχορδία της τρέχουσας γραμμής. Η ιδέα του GAN προωθήθηκε περαιτέρω από τον Dong το 2017 με το MuseGAN, το οποίο χρησιμοποιεί πολλαπλές γεννήτριες για να επιτύχει συνθετική μουσική πολλών οργάνων που σέβεται τις εξαρτήσεις μεταξύ των οργάνων. Ο Dong χρησιμοποίησε το Wasserstein-GAN with Gradient Penalty (WGAN-GP) για μεγαλύτερη σταθερότητα στο training.

Τέλος, καθώς οι τελευταίες εξελίξεις στο NLP έχουν γίνει με attention networks και transformers, έχουν γίνει παρόμοιες προσπάθειες να εφαρμοστούν μετασχηματιστές στη δημιουργία μουσικής. Ο Shaw (2019) δημιούργησε το MusicAutobot, το οποίο χρησιμοποιεί έναν συνδυασμό BERT, Transformer-XL και Seq2Seq για να δημιουργήσει μια μηχανή πολλαπλών εργασιών που μπορεί τόσο να δημιουργήσει νέα μουσική όσο και να δημιουργήσει αρμονία υπό την προϋπόθεση ότι υπάρχουν άλλα όργανα.

Dataset

Τα δεδομένα μας προήλθαν από το σύνολο δεδομένων Lakh Pianoroll, μια συλλογή από 174.154 multitrack pianorolls που προέρχονται από το σύνολο δεδομένων Lakh MIDI και επιμελήθηκε το Music and AI Lab at the Research Center for IT Innovation, Academia Sinica. Χρησιμοποιήσαμε την έκδοση LPD-5 του συνόλου δεδομένων, η οποία περιλαμβάνει κομμάτια για πιάνο, ντραμς, κιθάρα, μπάσο και έγχορδα, επιτρέποντάς μας να δημιουργήσουμε περίπλοκη και πλούσια μουσική και να δείξουμε την ικανότητα των μοντέλων παραγωγής μας να διασκευάζουν μουσική σε διαφορετικά όργανα. Χρησιμοποιήσαμε το καθαρισμένο υποσύνολο του συνόλου δεδομένων Lakh Pianoroll, το οποίο περιλαμβάνει 21.245 αρχεία MIDI. Κάθε ένα από τα αρχεία είχε αντίστοιχα μεταδεδομένα, επιτρέποντάς μας να προσδιορίσουμε πληροφορίες για κάθε αρχείο, όπως το όνομα του καλλιτέχνη και του τίτλου.

Αρχική μέθοδος: Next-Note Prediction with RNNs

Για να δημιουργήσουμε μια βάση στη δημιουργία μουσικής που μπορούμε να βελτιώσουμε, χρησιμοποιήσαμε επαναλαμβανόμενα νευρωνικά δίκτυα (RNN), μια υπάρχουσα και εύκολα αναπαραγόμενη μέθοδο. Η δημιουργία μουσικής διαμορφώνεται ως πρόβλημα πρόβλεψης επόμενης νότας.

(Αυτή η μέθοδος είναι πολύ παρόμοια με τα μοντέλα γλώσσας που βασίζονται σε υποτροπές που χρησιμοποιούνται στο NLP) Αυτό θα μας επέτρεπε να δημιουργήσουμε όση μουσική θέλαμε περνώντας συνεχώς τη νότα που δημιουργήθηκε πίσω στο μοντέλο.

Όσον αφορά την εφαρμογή, χρησιμοποιήσαμε την Gated Recurrent Unit (GRU) αντί για το vanilla RNN, λόγω της καλύτερης ικανότητάς του να διατηρεί μακροπρόθεσμες εξαρτήσεις (dependencies). Κάθε GRU θα λάμβανε την ενεργοποίηση και την έξοδο του προηγούμενου επιπέδου ως είσοδο και η έξοδος θα ήταν η επόμενη νότα με δεδομένη την προηγούμενη ενεργοποίηση και είσοδο.

Για να δημιουργήσουμε τα δεδομένα που απαιτούνται για την εκπαίδευση του επαναλαμβανόμενου νευρωνικού μας δικτύου, πρώτα αναλύσαμε τις νότες πιάνου του συνόλου δεδομένων μας, αντιπροσωπεύοντας κάθε αρχείο ως μια λίστα με νότες που βρίσκονται στο αρχείο. Στη συνέχεια δημιουργήσαμε τις ακολουθίες εισόδου εκπαίδευσης παίρνοντας υποσύνολα της λίστας αναπαράστασης για κάθε τραγούδι και δημιουργήσαμε τις αντίστοιχες ακολουθίες εξόδου εκπαίδευσης παίρνοντας απλώς την επόμενη νότα κάθε υποσυνόλου. Με αυτήν την είσοδο και έξοδο εκπαίδευσης, το μοντέλο θα εκπαιδευόταν να προβλέψει την επόμενη νότα, η οποία θα μας επέτρεπε στη συνέχεια να περάσουμε σε οποιαδήποτε ακολουθία από νότες και να πάρουμε μια πρόβλεψη για την επόμενη νότα. Κάθε ακολουθία εισόδου περνούσε σε ένα embedded layer που δημιουργούσε embeddings μεγέθους 96. Αυτή η ενσωμάτωση στη συνέχεια πέρασε σε μια περιφραγμένη επαναλαμβανόμενη μονάδα με ένα μόνο στρώμα, το οποίο στη συνέχεια πέρασε σε ένα πλήρως συνδεδεμένο επίπεδο για να εξάγει μια κατανομή πιθανότητας του επόμενου Σημείωση. Θα μπορούσαμε να επιλέξουμε τη νότα με την υψηλότερη πιθανότητα ως την επόμενη προβλεπόμενη νότα, αλλά αυτό θα οδηγούσε σε ντετερμινιστικές ακολουθίες χωρίς παραλλαγή. Ως εκ τούτου, λαμβάνουμε δείγμα της επόμενης σημείωσης από μια πολυωνυμική κατανομή με τις πιθανότητες εξόδου. Αυτό το embedding στη συνέχεια περνάει σε ένα gated recurrent unit με ένα μόνο στρώμα, το οποίο στη συνέχεια περνάει σε ένα fully connected layer για να εξάγει μια κατανομή πιθανότητας της επόμενης νότας. Θα μπορούσαμε να επιλέξουμε τη νότα με την υψηλότερη πιθανότητα ως την επόμενη προβλεπόμενη νότα, αλλά αυτό θα οδηγούσε σε ντετερμινιστικές ακολουθίες χωρίς παραλλαγή. Ως εκ τούτου, λαμβάνουμε δείγμα της επόμενης νότας από μια πολυωνυμική κατανομή με τις πιθανότητες εξόδου.

Αξιολόγηση:

Ενώ το μοντέλο πρόβλεψης επόμενης νότας RNN είναι εύκολο και καθαρό στην εφαρμογή, η μουσική που δημιουργείται ακούγεται κάθε άλλο παρά ιδανική και υπάρχει πολύ περιορισμένη χρησιμότητα. Επειδή κωδικοποιούμε κάθε νότα σε ένα token και προβλέπουμε μια κατανομή πιθανότητας στις κωδικοποιήσεις, μπορούμε πραγματικά να το κάνουμε αυτό μόνο για ένα όργανο, επειδή για πολλά όργανα, ο αριθμός των συνδυασμών των νοτών αυξάνεται εκθετικά. Επίσης, η υπόθεση ότι κάθε νότα έχει το ίδιο μήκος σίγουρα δεν αντικατοπτρίζει τα περισσότερα μουσικά έργα.

Multi-instrument RNN

Ως εκ τούτου, επιδιώξαμε να εξερευνήσουμε άλλες μεθόδους για τη δημιουργία μουσικής για πολλά όργανα ταυτόχρονα και καταλήξαμε στο Multi-Instrument RNN.

Αντί να κωδικοποιήσουμε τη μουσική σε μοναδικές νότες/χορδές όπως κάναμε στην αρχική ιδέα, δουλέψαμε απευθείας με το piano roll 5 x 128 πολλαπλών οργάνων piano roll σε κάθε βήμα, και μετά flattened ώστε να γίνει ένα διάνυσμα 640 διαστάσεων που αντιπροσωπεύει τη μουσική σε κάθε βήμα-βήμα.

Στη συνέχεια, εκπαιδεύσαμε ένα RNN να προβλέπει το διάνυσμα 640 διαστάσεων του επόμενου χρονικού βήματος, δεδομένης της προηγούμενης ακολουθίας μήκους 32 διανυσμάτων 640 διαστάσεων.

Αν και αυτή η μέθοδος θεωρητικά θα είχε νόημα, ήταν δύσκολο να παραχθούν ικανοποιητικά αποτελέσματα λόγω της δυσκολίας παραγωγής ποικιλίας που ήταν συμπληρωματική σε όλα τα όργανα.

- Στη ρύθμιση ενός οργάνου, λάβαμε δείγμα από μια πολυωνυμική κατανομή με πιθανότητα σταθμισμένη με τις βαθμολογίες softmax εξόδου για να δημιουργήσουμε την επόμενη νότα. Ωστόσο, δεδομένου ότι όλα τα όργανα τοποθετούνται μαζί στο διάνυσμα 640 διαστάσεων, η δημιουργία της επόμενης νότας χρησιμοποιώντας βαθμολογίες softmax-ed σε ολόκληρο το διάνυσμα 640d θα μπορούσε να σημαίνει ότι ορισμένα όργανα θα μπορούσαν ενδεχομένως να έχουν πολλές νότες ενώ ορισμένα όργανα δεν έχουν καμία.
- Προσπαθήσαμε να λύσουμε αυτό το πρόβλημα εκτελώντας τη συνάρτηση softmax ξεχωριστά για καθένα από τα 5 διανύσματα 128 διαστάσεων του οργάνου, ώστε να μπορέσουμε να διασφαλίσουμε ότι θα δημιουργήσουμε έναν ορισμένο αριθμό νότων για κάθε όργανο.
- Ωστόσο, αυτό σήμαινε ότι η δειγματοληψία για κάθε όργανο ήταν ανεξάρτητη μεταξύ τους. Αυτό σημαίνει ότι η παραγόμενη ακολουθία πιάνου δεν θα ήταν συμπληρωματική με τις ακολουθίες των άλλων οργάνων. Για παράδειγμα, εάν η συγχορδία C-E-G δειγματίζεται από τη σειρά, το μπάσο δεν έχει τρόπο να το ενσωματώσει και θα μπορούσε να δοκιμάσει τη χορδή D-F-A, η οποία είναι αρμονικά ασύμφωνη και όχι συμπληρωματική.
- Επιπλέον, υπήρχε το πρόβλημα να μην γνωρίζουμε πόσες νότες πρέπει να δειγματίζονται για κάθε όργανο κάθε φορά. Αυτό το πρόβλημα δεν υπήρχε στη ρύθμιση ενός οργάνου επειδή οι μονές νότες και οι συγχορδίες πολλών νότων κωδικοποιούνται όλες ως αναπαραστάσεις ακέραιων αριθμών. Αντιμετωπίσαμε αυτό το ζήτημα δειγματίζοντας έναν καθορισμένο αριθμό νότων για κάθε χρονικό βήμα (π.χ. 2 για πιάνο, 3 για κιθάρα) από το πολυώνυμο. Αλλά αυτό ήταν ανεπιτυχές καθώς η μουσική που δημιουργήθηκε ακουγόταν πολύ τυχαία και μη μουσική.

Δείγματα ήχου της μουσικής που δημιουργήθηκε από αυτά τα δύο τραγούδια βρίσκονται στο φάκελο Music με το όνομα RNN Multitrack.wav.

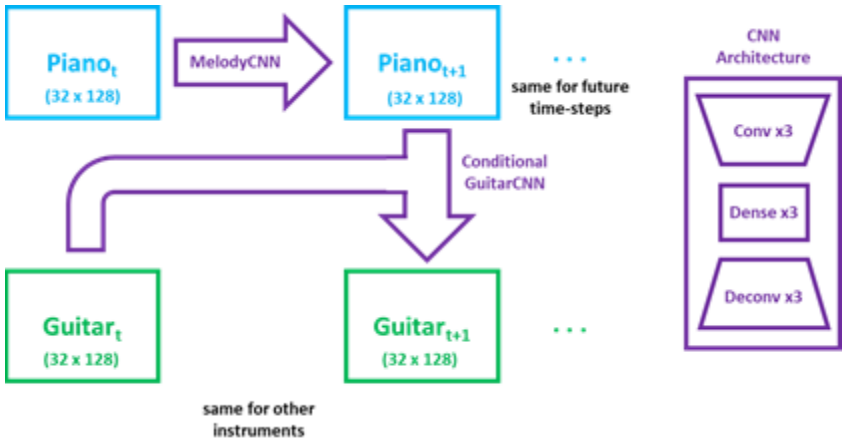
CNN

Από αυτό το σημείο και μετά, αποφασίσαμε να επικεντρωθούμε στα Συνελικτικά Νευρωνικά Δίκτυα (CNN) αντί στα RNN για να δημιουργήσουμε ακολουθίες μουσικής. Το CNN θα δημιουργούσε απευθείας μια ακολουθία μήκους 32 βγάζοντας έναν τρισδιάστατο tensor 5 x 32 x 128. Αυτό θα έλυνε το πρόβλημα του να μην γνωρίζουμε πόσες νότες να δημιουργήσουμε και να χρειάζεται να χρησιμοποιήσουμε πολυωνυμική δειγματοληψία. Οι αρχιτεκτονικές του CNN, όπως το WaveNet, έχουν αποδειχθεί ότι επιτυγχάνουν εξίσου καλή, αν όχι καλύτερη απόδοση με τα RNN στη δημιουργία ακολουθιών. Επιπλέον, εκπαιδεύονται πολύ πιο γρήγορα λόγω βελτιστοποιήσεων απόδοσης με συνελικτικές λειτουργίες.

MelodyCNN και Conditional HarmonyCNNs

Προκειμένου να δημιουργήσουμε πολλά κομμάτια οργάνων συμβατά μεταξύ τους, δοκιμάσαμε ένα μοντέλο παραγωγής δύο μερών που περιλαμβάνει ένα MelodyCNN για τη δημιουργία μελωδίας επόμενου βήματος, καθώς και ένα Conditional-HarmonyCNN για τη δημιουργία οργάνων που δεν είναι πιάνο,

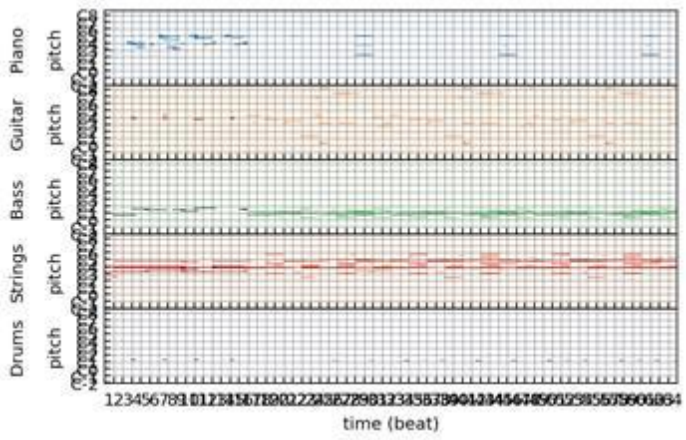
δεδομένου της μελωδίας για το ίδιο χρονικό βήμα καθώς και τη μουσική αυτού του οργάνου για το τελευταίο χρονικό βήμα.



Εικόνα 1: Αρχιτεκτονική του MelodyCNN + Conditional HarmonyCNN που χρησιμοποιείται για τη δημιουργία μουσικής.

Δεδομένου ότι τα μεγέθη εισόδου και εξόδου είναι τα ίδια (32 x 128), η αρχιτεκτονική του MelodyCNN που χρησιμοποιήθηκε ήταν συμμετρική, με 3 συνελικτικά στρώματα, 3 πυκνά στρώματα και 3 αποσυνελικτικά στρώματα. Το Conditional HarmonyCNN χρησιμοποίησε 3 συνελικτικά στρώματα για κάθε μία από τις εισόδους (πιάνο καθώς και την προηγούμενη του οργάνου), στη συνέχεια συνέδεσε τους tensors που προέκυψαν πριν περάσουν από πυκνά και αποσυνελικτικά στρώματα. Ως εκ τούτου, το MelodyCNN μαθαίνει μια χαρτογράφηση μεταξύ των ακολουθιών πιάνου σε διαδοχικά χρονικά βήματα, ενώ τα Conditional HarmonyCNN χαρτογραφούν από τον μουσικό χώρο του πιάνου στα άλλα όργανα.

Χρησιμοποιώντας τα 5 CNN συνολικά (ένα για κάθε όργανο), μπορεί να δημιουργηθεί νέα μουσική επαναληπτικά με μια αρχική ακολουθία πολλών οργάνων. Πρώτον, το MelodyCNN χρησιμοποιείται για την πρόβλεψη της επόμενης ακολουθίας πιάνου και τα Conditional HarmonyCNN για την πρόβλεψη των άλλων οργάνων.



Εικόνα 2: Pianoroll μουσικής που δημιουργήθηκε από το MelodyCNN + Conditional Harmony CNN.

Αυτό το πλαίσιο ήταν επιτυχές στη δημιουργία μουσικών ακολουθιών πολλών οργάνων όπου τα όργανα ακούγονται μουσικά συμπληρωματικά. Ωστόσο, η αλλαγή της αρχικής ακολουθίας από την οποία

παράγεται η μουσική οδήγησε σε πολύ μικρή διαφοροποίηση στη μουσική που παράγεται, όπως φαίνεται στο piano roll παραπάνω: οι τρεις παραγόμενες ακολουθίες είναι σχεδόν ίδιες μεταξύ τους.

Δείγματα ήχου της μουσικής που δημιουργήθηκε από αυτά τα δύο τραγούδια βρίσκονται στο φάκελο Music με τα ονόματα «MelodyCNN+ConditionalCNN», «MelodyCNN All Same», «MelodyCNN All Same2».

Αυτό δείχνει ότι τα CNN πιθανότατα συνέκλιναν στην έξοδο μόνο ενός μικρού υποσυνόλου κοινών ακολουθιών στα δεδομένα εκπαίδευσης που ελαχιστοποιούσαν την απώλεια εκπαίδευσης. Πρέπει να βρεθεί μια άλλη μέθοδος για τη δημιουργία κάποιας ποικιλίας στη μουσική που παράγεται, με την ίδια είσοδο, και για να το πετύχουμε αυτό, στραφούμε στα VAE.

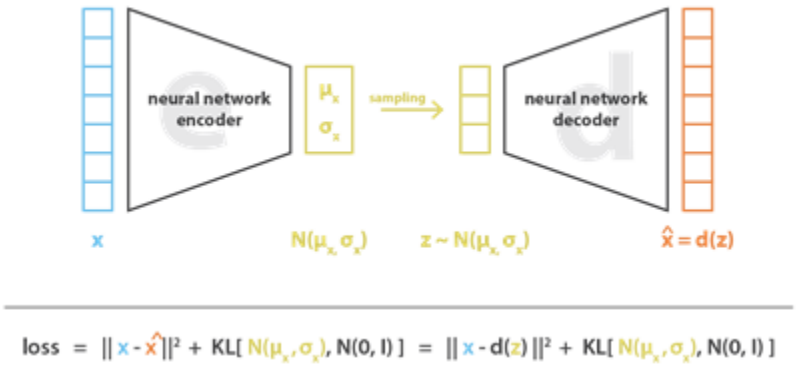
Variational Autoencoders (VAEs)

Πληροφορίες για VAEs

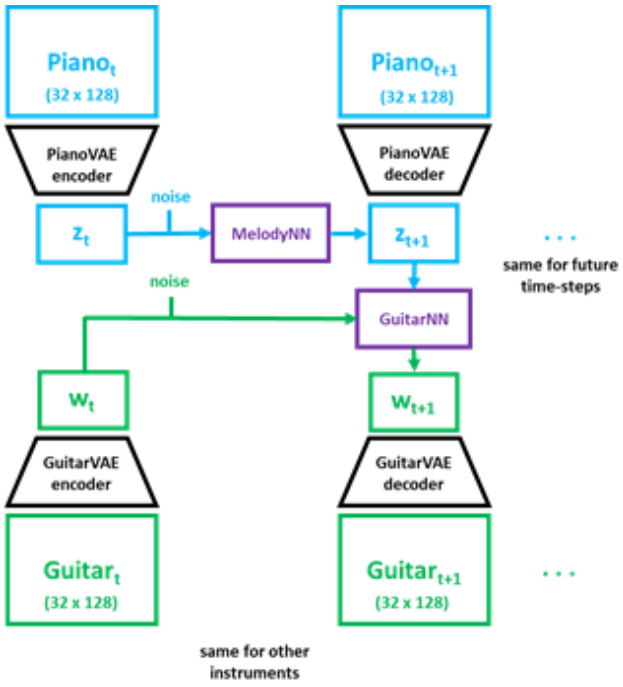
Ένας Variational Autoencoder (VAE) είναι ένας αυτόματος κωδικοποιητής στον οποίο η εκπαίδευση ρυθμίζεται για να διασφαλιστεί ότι το latent space έχει καλές ιδιότητες που επιτρέπουν μια διαδικασία παραγωγής. Δύο τέτοιες ιδιότητες είναι η συνέχεια - τα κοντινά σημεία στον latent χώρο θα πρέπει να δίνουν παρόμοια σημεία μόλις αποκωδικοποιηθούν και η πληρότητα - ένα σημείο που λαμβάνεται δείγμα από τον latent χώρο θα πρέπει να δίνει ουσιαστικό περιεχόμενο μόλις αποκωδικοποιηθεί.

Ένας vanilla autoencoder κωδικοποιεί τις εισόδους σε ένα διάνυσμα στο latent space, αλλά δεν έχει καμία εγγύηση ότι ο latent χώρος ικανοποιεί τη συνέχεια και την πληρότητα που επιτρέπουν τη δημιουργία νέων δεδομένων. Αντίθετα, ένα VAE κωδικοποιεί μια είσοδο ως κατανομή σε latent space. Συγκεκριμένα, υποθέτουμε ότι η latent κατανομή θα ακολουθεί Gaussian, επομένως ο κωδικοποιητής που κωδικοποιεί μια κατανομή είναι ισοδύναμος με τον κωδικοποιητή που εξάγει τις παραμέτρους μέσης και τυπικής απόκλισης της κανονικής κατανομής.

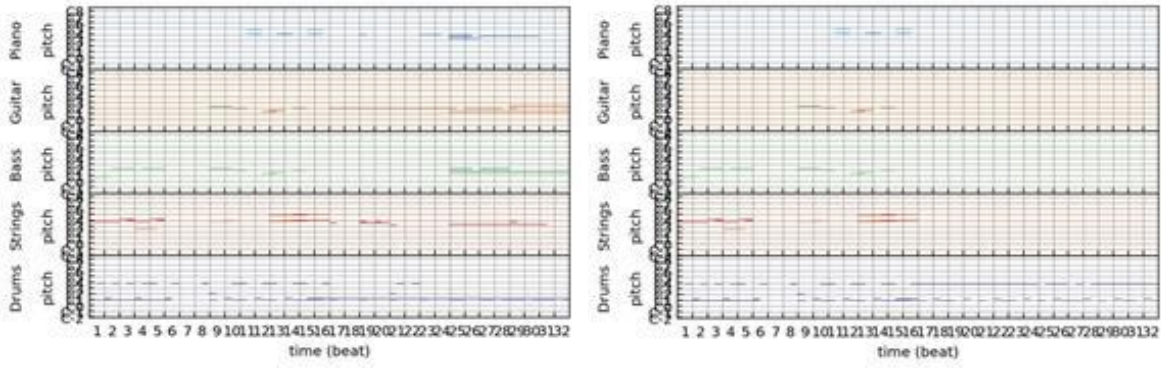
Για την εκπαίδευση του VAE, χρησιμοποιείται μια συνάρτηση απώλειας δύο όρων: ένα σφάλμα ανακατασκευής (διαφορά μεταξύ αποκωδικοποιημένων εξόδων και εισόδων), καθώς και ένας όρος κανονικοποίησης (KL-απόκλιση μεταξύ της latent κατανομής και του τυπικού Gaussian) για την κανονικοποίηση της latent κατανομής στο να είναι όσο το δυνατόν πιο κοντά στο τυπικό φυσιολογικό.



Εικόνα 3: Απεικόνιση του πώς λειτουργεί ένας Αυτόματος Κωδικοποιητής Μεταβλητών (VAE).



Εικόνα 4: Αρχιτεκτονική του VAE-NN που χρησιμοποιούμε για τη δημιουργία μουσικής



Εικόνα 5: Δύο pianorolls που δημιουργούνται από την ίδια αρχική σειρά. Ένα παράδειγμα παραλλαγής που εμφανίζεται στην έξοδο μουσικής φαίνεται παραπάνω. Και τα δύο παραπάνω κομμάτια είχαν την ίδια αρχική σειρά, αλλά τα drumbeats που δημιουργήθηκαν ήταν ελαφρώς διαφορετικά. Επίσης, το πρώτο κομμάτι είχε τμήμα πιάνου προς το τέλος, ενώ το δεύτερο κομμάτι όχι, και τα υπό όρους NN ανταποκρίθηκαν μεταβάλλοντας τα συνοδευτικά μουσικά κομμάτια που δημιουργήθηκαν.

Εφαρμογή

Ως εκ τούτου, εφαρμόζουμε VAE στην δημιουργίας μουσικής. Η προηγούμενη είσοδος πιάνου κωδικοποιείται από το πιάνο VAE σε μια latent κωδικοποίηση πιάνου διάστασης K, zt. Στη συνέχεια, ο τυχαίος θόρυβος προστίθεται στις μέσες παραμέτρους της κωδικοποιημένης latent κατανομής. Η τυπική απόκλιση αυτού του τυχαίου θορύβου είναι μια υπερπαράμετρος που ο χρήστης μπορεί να συντονίσει με βάση το μέγεθος της διακύμανσης που επιθυμεί. Οι latent παράμετροι zt εισάγονται στη συνέχεια στο MelodyNN, ένα Perceptron πολλαπλών επιπέδων που μαθαίνει μια χαρτογράφηση από τη latent κατανομή της προηγούμενης ακολουθίας πιάνου στην latent κατανομή της επόμενης ακολουθίας πιάνου. Στη συνέχεια, η έξοδος zt+1 αποκωδικοποιείται για να δημιουργηθεί η επόμενη έξοδος πιάνου.

Τα VAE ειδικά για όργανα εκπαιδεύονται και στα άλλα τέσσερα όργανα (κιθάρα, μπάσο, έγχορδα, ντραμς).

Στη συνέχεια, παρόμοια με το ConditionalCNN νωρίτερα, χρησιμοποιούμε ένα ConditionalNN, ένα άλλο MLP που λαμβάνει τις παραμέτρους latent πιάνου επόμενης περιόδου που δημιουργήθηκαν zt+1 καθώς και τις latent παραμέτρους κιθάρας προηγούμενης περιόδου wt και μαθαίνει μια αντιστοίχιση για τις latent παραμέτρους στην κιθάρα της επόμενης περιόδου wt+1. Στη συνέχεια, το wt+1 αποκωδικοποιείται από τον αποκωδικοποιητή VAE του συγκεκριμένου οργάνου για να παραχθεί η έξοδος κιθάρας επόμενης περιόδου. Εκπαιδεύονται 4 ConditionalNN, ένα για κάθε όργανο που δεν είναι πιάνο, που επιτρέπουν τη δημιουργία της επόμενης ακολουθίας 5 οργάνων.

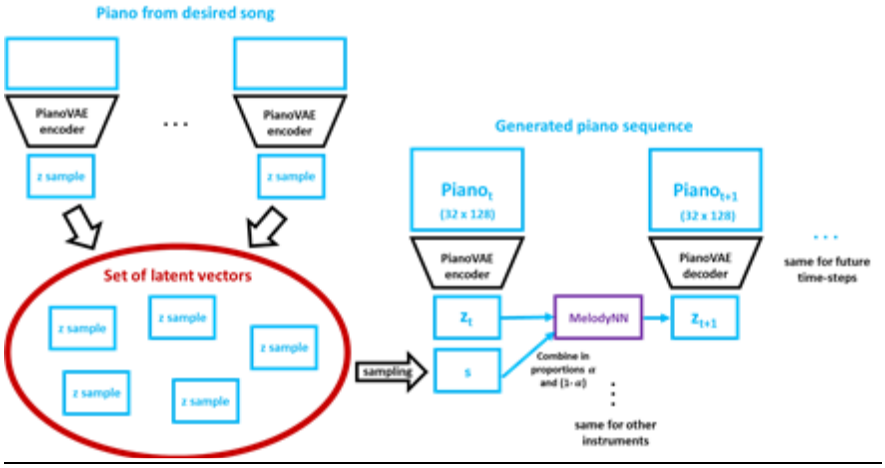
Έτσι, χαρτογραφώντας τις μουσικές εισόδους σε latent κατανομές με VAE, μπορούμε να εισάγουμε παραλλαγή στην παραγόμενη έξοδο μουσικής προσθέτοντας τυχαίο θόρυβο στις παραμέτρους της κωδικοποιημένης latent κατανομής. Λόγω της συνέχειας, αυτό διασφαλίζει ότι μετά την προσθήκη τυχαίου θορύβου, οι αποκωδικοποιημένες είσοδοι είναι παρόμοιες αλλά διαφορετικές από τις αρχικές εισόδους, και λόγω της πληρότητας, διασφαλίζει ότι δίνουν ουσιαστικές μουσικές εξόδους που είναι παρόμοιες με τη διανομή μουσικής εισόδου.

Αποτελέσματα

Εκπαιδεύτηκαν VAE latent διάστασης 8, 16, 32 και 64. Στο τέλος, χρησιμοποιήθηκε ένας latent χώρος 16 διαστάσεων για την εκπαίδευση των υπό όρους NN, αφού τα μουσικά δείγματα είναι σχετικά αραιά στο μουσικό χώρο. Μετά την εκπαίδευση των υπό όρους NN, διαπιστώνουμε ότι η μέθοδος VAE+NN είναι επιτυχής στη δημιουργία εξόδων πολλαπλών οργάνων που ακούγονται συνεπείς, καθώς και ότι έχουν κατάλληλες ποσότητες παραλλαγών για να είναι αισθητικά ευχάριστες. Ο τυχαίος θόρυβος τυπικών αποκλίσεων μεταξύ 0,5 και 1,0 βρέθηκε να δημιουργεί την καλύτερη ποσότητα διακύμανσης.

Δείγματα ήχου της μουσικής που δημιουργείται με χρήση των VAE μπορούν να βρεθούν στον φάκελο Music με ονόματα από «VAE Good 1» έως 4.

Δημιουργία μουσικής σε ορισμένα στυλ



Εικόνα 6: Μέθοδος δημιουργίας μουσικής σε σχέση με κάποιο στυλ

Το πλαίσιο VAE-NN που εξηγήθηκε παραπάνω μας επιτρέπει μια απλή μέθοδο δημιουργίας μουσικής με βάση συγκεκριμένα στυλ, όπως ένας συγκεκριμένος καλλιτέχνης, είδος ή χρονιά. Για παράδειγμα, αν θέλαμε να δημιουργήσουμε μουσική στο στυλ του Thriller του Michael Jackson, θα μπορούσαμε να:

1. Σπάσουμε το τραγούδι σε ακολουθίες 32 βημάτων και κωδικοποιούμε το pianoroll κάθε ακολουθίας στον latent χώρο χρησιμοποιώντας τον κωδικοποιητή VAE κάθε οργάνου. Αποθηκεύουμε τις μοναδικές ακολουθίες σε ένα σετ για κάθε όργανο.
2. Κατά τη δημιουργία μουσικής από μια αρχική ακολουθία, λαμβάνεται δείγμα από ένα latent διάνυσμα ανά όργανο από αυτό το σύνολο. Αυτό το δειγματοληπτιμένο latent διάνυσμα (από το επιθυμητό τραγούδι μας) s στη συνέχεια γίνεται interpolated με το latent διάνυσμα της προηγούμενης ακολουθίας z_t για να δημιουργηθεί ένα νέο latent διάνυσμα $z_t'=\alpha \boldsymbol{s}+(1-\alpha)z_t$, με το α να είναι το latent sample factor, η υπερπαράμετρος δηλαδή που μπορεί να συντονιστεί. (Επιλέξτε υψηλότερες τιμές α στη μουσική που δημιουργείται για να εξαρτηθείτε σημαντικά από το επιθυμητό στυλ)
3. Χρησιμοποιούμε το z_t' αντι του z_t ως την είσοδο στο MelodyNN για να δημιουργήσουμε το νέο latent διάνυσμα και ως εκ τούτου την παραγόμενη ακολουθία πιάνου.

Χρησιμοποιώντας αυτή τη μέθοδο και α=0,5, δημιουργήσαμε νέα μουσική με βάση πολλά τραγούδια, μερικά παραδείγματα είναι το Thriller του Michael Jackson και το I Want It That Way των Backstreet Boys. Είχαμε επιτυχία στη δημιουργία δειγμάτων ήχου που έχουν κάποια ομοιότητα με το αρχικό τραγούδι, αλλά και με κάποια παραλλαγή. (Για άλλη μια φορά, η έκταση της διακύμανσης μπορεί να ρυθμιστεί και με την υπερπαράμετρο noise_sd). Κάποιος μπορεί ακόμη και να δημιουργήσει μουσική με βάση δείγματα που είναι ένα υβρίδιο διαφορετικών καλλιτεχνών ή στυλ, επιτρέποντας έτσι στους λάτρεις της μουσικής να συνθέσουν μουσική συνδυάζοντας τα στυλ διαφορετικών μουσικών σταρ.

Δείγματα ήχου της μουσικής που δημιουργήθηκε από αυτά τα δύο τραγούδια βρίσκονται στο φάκελο Music με τα ονόματα «VAE - Thriller» και «VAE - I Want It That Way».

GANs

Εμπνευσμένοι από την επιτυχία του MidiNet, το οποίο χρησιμοποίησε τα Deep Convolutional Generative Adversarial Networks (DCGAN) για την παραγωγή μουσικής με ρεαλιστικό ήχο, προσπαθήσαμε να χρησιμοποιήσουμε και GAN για τη δημιουργία μουσικής. Είναι γνωστό ότι τα GAN παράγουν εξαιρετικά ρεαλιστικά συνθετικά δείγματα στο πεδίο της όρασης υπολογιστών, καλύτερα από τα VAE. Αυτό συμβαίνει επειδή τα GAN δεν εκτιμούν τη ρητή πυκνότητα πιθανότητας της κατανομής, ενώ τα VAE προσπαθούν να βελτιστοποιήσουν το lower variational bound. Ωστόσο, είναι γνωστό ότι τα GAN είναι πολύ δύσκολο να εκπαιδεύονται με επιτυχία.

Χρησιμοποιήσαμε μια γεννήτρια με 6 αποσυνελικτικά στρώματα, λαμβάνοντας ένα διάνυσμα θορύβου 100 διαστάσεων και δημιουργώντας μια ακολουθία μουσικής πολλαπλών οργάνων 5 x 32 x 128. Ο discriminator έχει την αντίθετη αρχιτεκτονική, λαμβάνοντας μια ακολουθία μουσικής 5 x 32 x 128, περνώντας την μέσα από 6 συνελικτικά επίπεδα και εξάγοντας μια πιθανότητα το δείγμα να είναι πραγματικό.

Και για τη γεννήτρια και για τον discriminator, χρησιμοποιήθηκε η ενεργοποίηση PReLU, καθώς και η κανονικοποίηση batch για τα συνελικτικά στρώματα. Το Adam optimizer χρησιμοποιήθηκε και για τα δύο.

Επιχειρήθηκαν οι ακόλουθες μέθοδοι για τη βελτίωση της σταθερότητας του GAN:

- Εξομάλυνση ετικετών: Αντί να χρησιμοποιούμε συγκεκριμένες ετικέτες 0 ή 1 για τις δημιουργημένες ή πραγματικές εικόνες αντίστοιχα, προσθέτουμε τυχαίο θόρυβο στην ετικέτα (έτσι ώστε οι δημιουργούμενες εικόνες να έχουν ετικέτα μεταξύ 0 και 0,1 και οι πραγματικές εικόνες να έχουν ετικέτα μεταξύ 0,9 και 1).
- Αντιστοίχιση χαρακτηριστικών: Προσθήκη regularizers L2 για την επιβολή των διανομών των πραγματικών και των παραγόμενων δεδομένων να είναι κοντά. Χρησιμοποιήθηκαν δύο regularizers: ο πρώτος στην απόλυτη διαφορά της αναμενόμενης τιμής εισόδων πραγματικής έναντι της παραγόμενης εικόνας, και ο δεύτερος στην απόλυτη διαφορά αναμενόμενης τιμής των εξόδων του πρώτου συνελικτικού επιπέδου για τις εισόδους πραγματικών και παραγόμενων εικόνων.
- Two Time-Scale Update Rule (TTUR): Χρήση υψηλότερου learning rate για τον discriminator που κάνει διάκριση σε σχέση με τη γεννήτρια
- Ρύθμιση του learning rate

Παρά τις πολλές προσπάθειες, η εκπαίδευση του GAN αποδείχθηκε ανεπιτυχής στη δημιουργία μιας ποικιλίας μουσικής με ρεαλιστικό ήχο. Υπήρξαν περιπτώσεις κατάρρευσης λειτουργίας, όπως το δείγμα ήχου που δημιουργήθηκε παρακάτω, το οποίο είναι 100 δείγματα που δημιουργούνται από διαφορετικά διανύσματα θορύβου που συνδέονται μεταξύ τους. Τα παραγόμενα δείγματα είναι ως επί το πλείστων παρόμοια. Δείγματα ήχου της παραγόμενης μουσικής βρίσκονται στο φάκελο Music με το όνομα «GAN Mode Collapse». Άλλες προσπάθειες απέτυχαν να μάθουν κάτι ουσιαστικό.

Συμπεράσματα

Συνολικά, έχουμε εφαρμόσει μια ποικιλία μεθόδων βαθιάς εκμάθησης στο πρόβλημα της παραγωγής μουσικής με διαφορετικά επίπεδα επιτυχίας. Η βασική μας μέθοδος χρησιμοποίησε ένα μοντέλο επαναλαμβανόμενου νευρωνικού δικτύου τόσο για ένα μόνο κομμάτι όσο και για πολλαπλά κομμάτια. Ενώ αυτό το μοντέλο βρήκε μεγαλύτερη επιτυχία όσον αφορά τη μουσικότητα των παραγόμενων νοτών, ήταν πολύ περιορισμένη στη χρησιμότητα καθώς μπορούσε να παράγει νότες μόνο στους ρυθμούς των τετάρτων. Στη συνέχεια προχωρήσαμε σε ένα μοντέλο συνελικτικού νευρωνικού δικτύου, χρησιμοποιώντας ένα vanilla CNN για την παραγωγή του κομματιού πιάνου και ένα υπό όρους CNN που χρησιμοποιούσε το κομμάτι του πιάνου για την παραγωγή των άλλων κομματιών οργάνων. Βρήκαμε ότι τα κομμάτια που παράγονται από τα μοντέλα του CNN ήταν πολύ πιο καλοσχηματισμένα και συνεκτικά, επειδή χρησιμοποιούσαμε μοντέλα υπό όρους.

Η αρχιτεκτονική βασισμένη σε VAE ήταν η πιο επιτυχημένη συνεισφορά για το έργο μας. Κωδικοποιώντας ακολουθίες σε έναν latent χώρο χρησιμοποιώντας ένα VAE, μπορούμε στη συνέχεια να προσθέσουμε θόρυβο στον latent χώρο για να αυξήσουμε τη διακύμανση της παραγόμενης εξόδου με ελεγχόμενο τρόπο, διατηρώντας παράλληλα την ομοιότητα μεταξύ της προηγούμενης ακολουθίας, βελτιώνοντας τελικά τη μοναδικότητα της μουσικής που δημιουργείται.

Ιδιαίτερες ευχαριστίες στον Καθηγητή μας Θεόδωρο Γιαννακόπουλο που μας δίδαξε το μάθημα του στο Deep Learning,

References

Data Sources
Lakh Pianoroll Dataset: https://salu133445.github.io/lakh-pianoroll-dataset/
Genre Labels for songs in the Million Song Dataset: https://www.tagtraum.com/msd_genre_datasets.html

Code Sources
VAE: Code adapted from https://github.com/CIS-522/course-content/tree/main/W08_VAE_GANs
GAN:

- training code adapted from https://nbviewer.org/github/CIS-522/course-content/blob/main/W08_VAE_GANs/students/CIS_522_W8D2_Tutorial_%E2%80%93_Student_Version.ipynb?flush_cache=true
- as well as MidiNet (https://github.com/annahung31/MidiNet-by-pytorch)
- feature matching, label smoothing code adapted from MidiNet

Next-note prediction:

- Training code adapted from https://github.com/Skuldur/Classical-Piano-Composer
- RNN language model evaluation and multinomial code adapted from https://nbviewer.org/github/CIS-522/course-content/blob/main/W09_RNNs/students/CIS_522_W9D2_Tutorial_%E2%80%93_Student_Version.ipynb?flush_cache=true

Pre-processing Lakh Midi Dataset

- Code adapted from https://nbviewer.jupyter.org/github/craffel/midi-dataset/blob/master/Tutorial.ipynb
- Code to convert from MIDI to music21 adapted from https://github.com/Skuldur/Classical-Piano-Composer

Code Sources
https://ai.plainenglish.io/building-a-lo-fi-hip-hop-generator-e24a005d0144

https://towardsdatascience.com/how-to-generate-music-using-a-lstm-neural-network-in-keras-68786834d4c5

https://towardsdatascience.com/creating-a-pop-music-generator-with-the-transformer-5867511b382a

https://towardsdatascience.com/practical-tips-for-training-a-music-model-755c62560ec2

https://towardsdatascience.com/a-multitask-music-model-with-bert-transformer-xl-and-seq2seq-3d80bd2ea08e

https://towardsdatascience.com/how-to-remix-the-chainsmokers-with-a-music-bot-6b920359248c

MuseGAN: https://arxiv.org/abs/1709.06298

MidiNet: https://arxiv.org/abs/1703.10847