Motivation

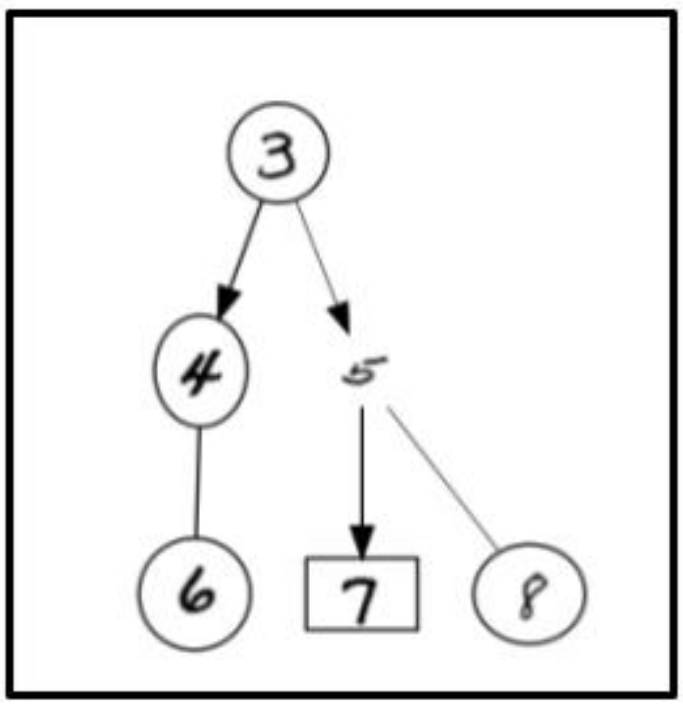
Autograding software for questions with binary tree drawing.

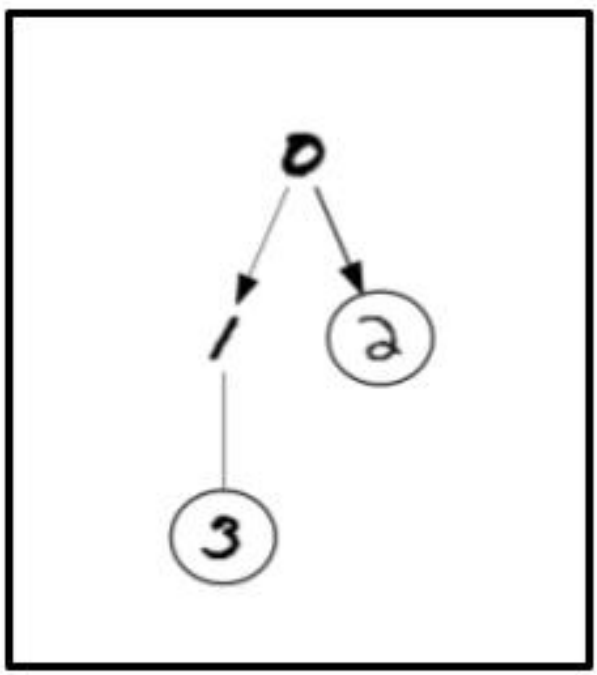
Problem

Given an image of a binary tree, produce a unique encoding of it.

Approach

Formulate a supervised learning problem and train a transformer.





Extracting Binary Trees

Chami Lamelas,

Models

PyTorch ViT (+/- pretrained weights),
Dive to Deep Learning ViT (D2L)

Preprocessing

Data augmentation, downsizing, and normalization.

Training

1. Train transformer to predict digits present in images.
2. Save model, then have it train to predict edge sets for images.

from Tree Drawings

Ryan Polhemus

Loss Formulation

Binary cross entropy loss applied to sigmoid of transformer outputs.

Edge Set Prediction

1. Use saved digit model to predict digits in image using thresholding after sigmoid.
2. Build complete graph of digits.
3. Weight graph using edge set transformer's outputs.
4. Compute edge set corresponding to minimum spanning tree of graph.

Discussion

- Extra Variety models learn to get high performance very quickly.
- D2L outperforms PyTorch ViT when it is not pretrained.
- Extra Variety models more likely memorizing than learning to generalize.
- Found via additional experiments that removing various components of training and prediction processes did not greatly decrease performance.
- Learning Extra Variety+ is much more difficult.

Vision Transformer (ViT)

- Learns distances in an image.
- Aggregates information from across the image.

$\Rightarrow$ Good for learning place of nodes in the overall tree.

Edge Set

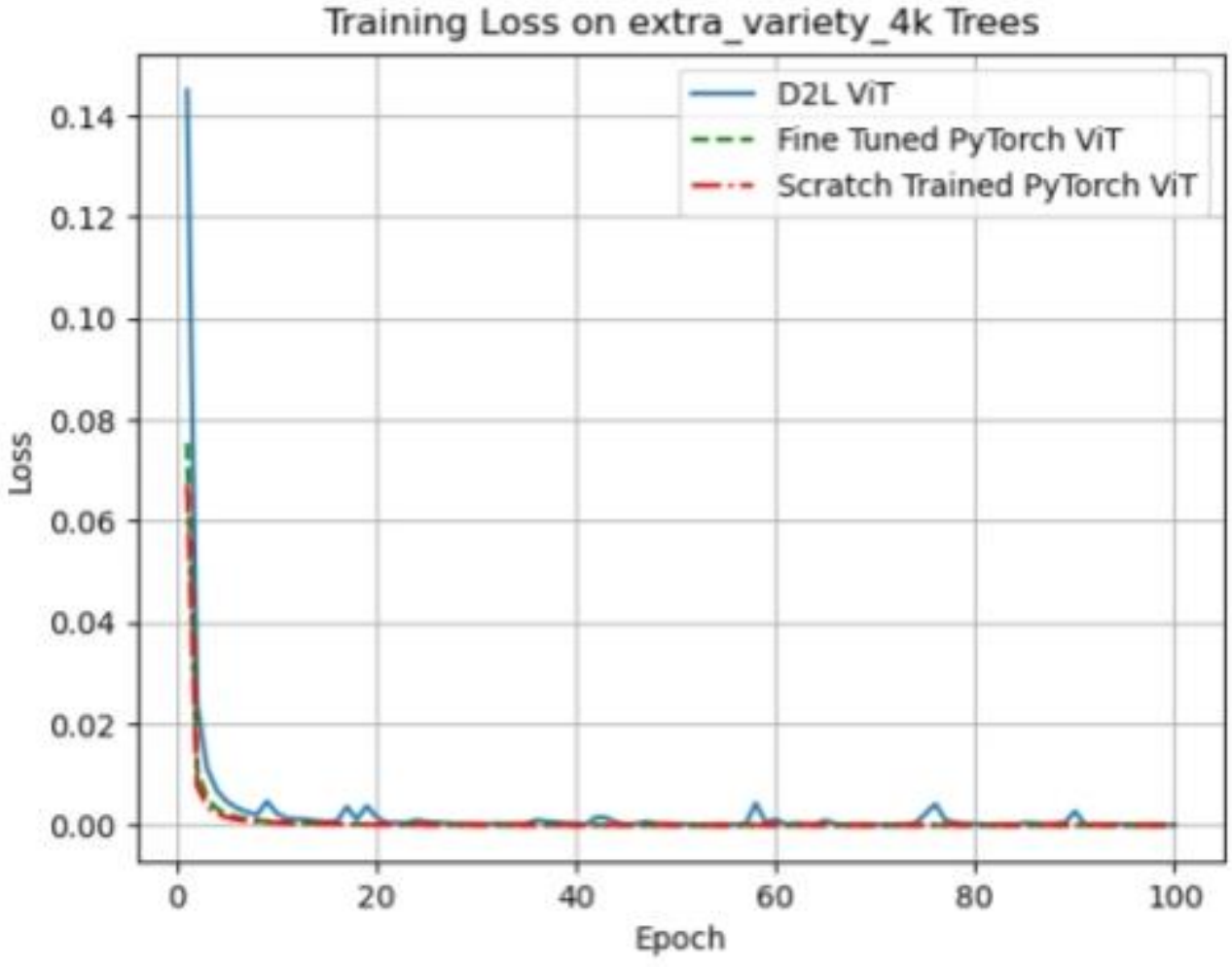
- Can be used to uniquely encode a subset of trees as binary vectors.

$\Rightarrow$ We can formulate learning task as a multilabel binary classification problem.

Extra Variety

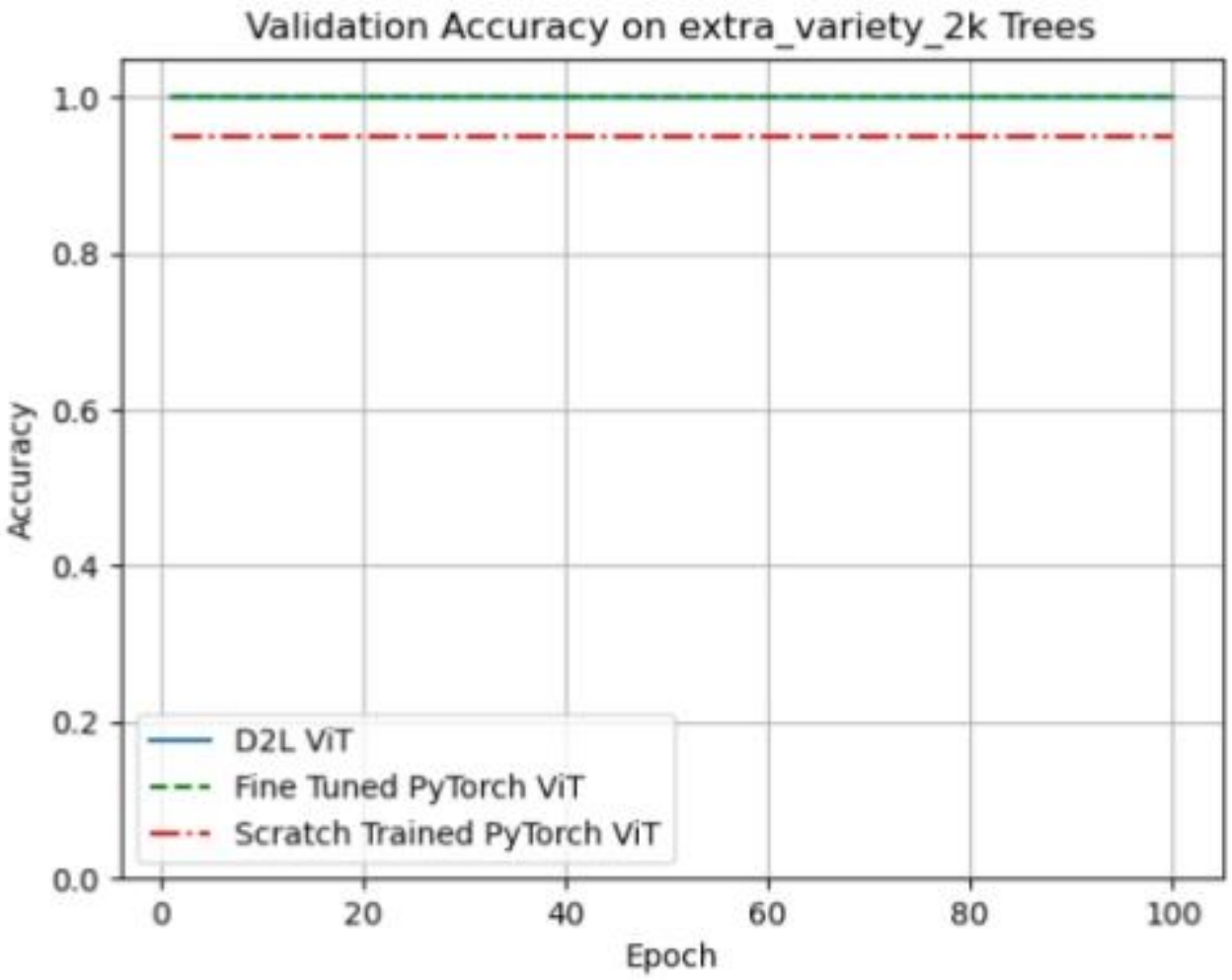
The Extra Variety dataset has additional limitations:

- Root node value is 0.
- Node values increase incrementally.
- Tree must be complete.



Experiments

Model	Test Accuracy
D2L	0.998
Fine-tuned PyTorch	1.000
Scratch PyTorch	0.933



- Training loss curves for digits and trees were more gradual (may be able to learn more).
- Extra Variety+ digit accuracy is closely tied to edge set accuracy.
- Results indicate good generalization with similar performance on train, validation, and test.

Lessons Learned

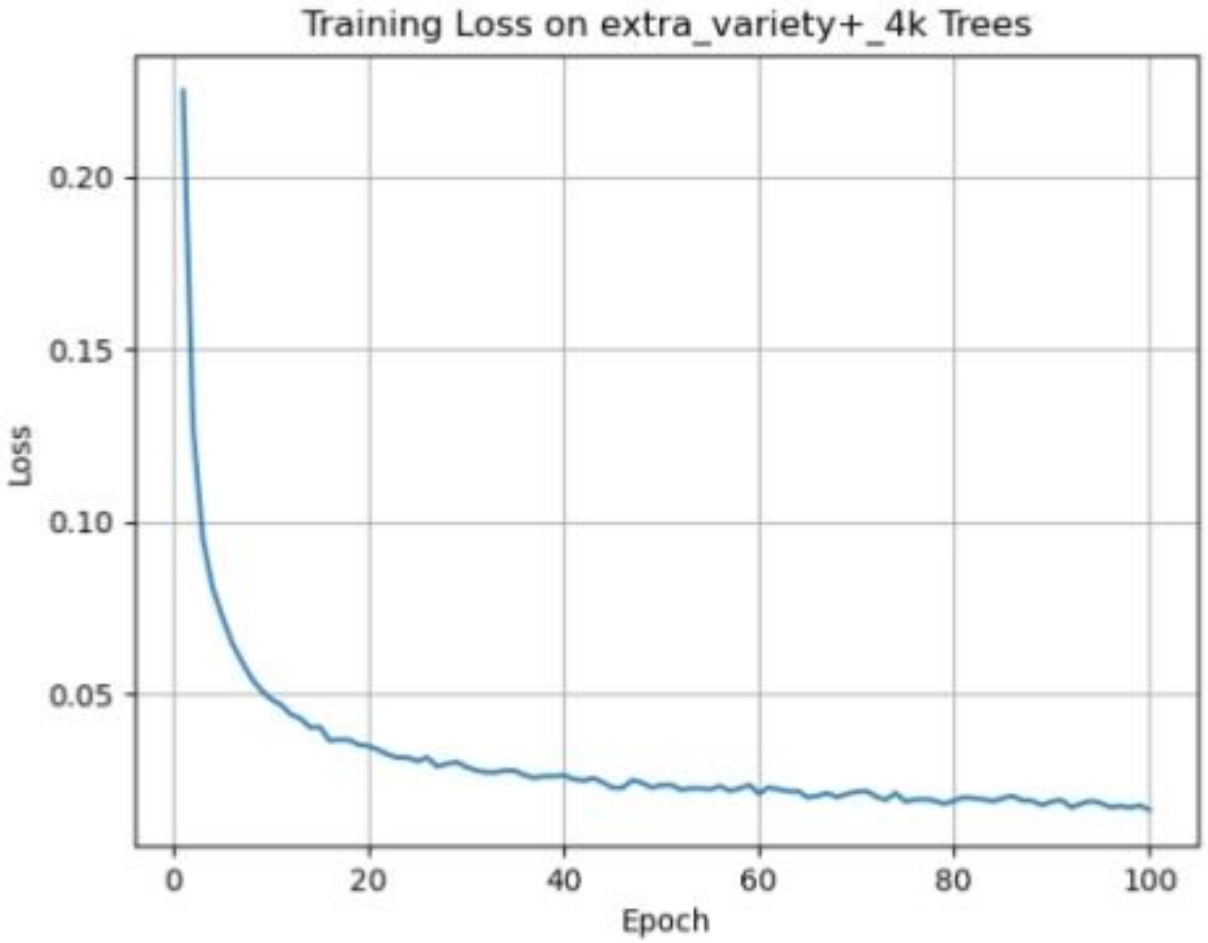
Training transformers is difficult.

- Resource intensive.
- May require downsized images.
- Prone to overfitting on small datasets.

Data Generation

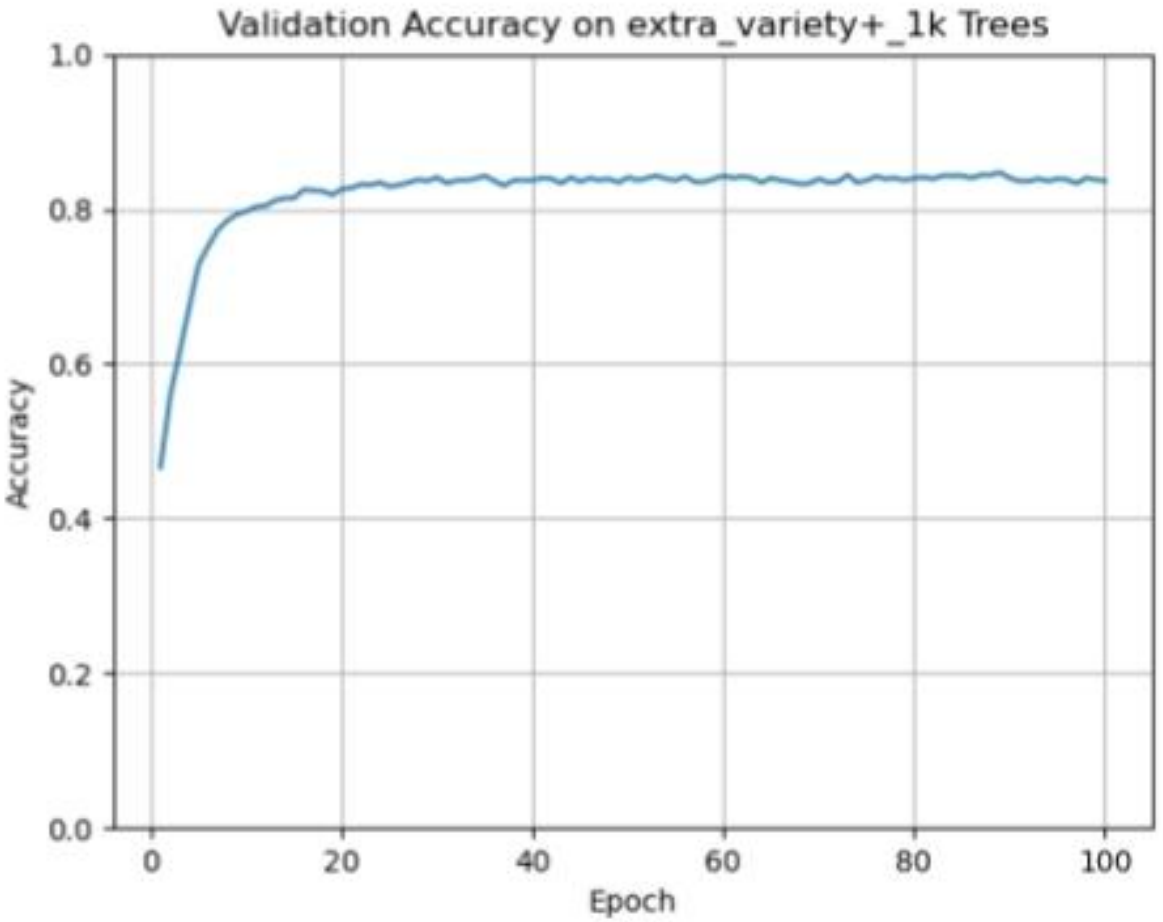
- Generate a valid tree, then convert to Graphviz, insert MNIST numbers, and render as PNG.
- Tree limitations:
  - Maximum of 10 nodes.
  - Node values increase top to bottom and left to right.
  - Values must be unique digits.
  - If a node has a right child, it must have a left child.
- Trees were generated randomly, with additional random elements.
- Values were weighted to encourage a variety of tree sizes and nodes.

Extra Variety+



- Extra Variety models performed terribly on Extra Variety+.
- Used pretrained PyTorch ViT
  + Data augmentation to increase training set.
  + Dropout for regularization.

Experiments



Split	Accuracy
Train	0.8725
Validation	0.8470
Test	0.8485

Conclusions

- Extra Variety dataset is very easy for baseline transformers to achieve high performance on.
- Approach for more complex Extra Variety+ dataset shows decent performance.
- Accuracy could most likely be improved with better tuning and hardware.
- Provides proof of concept for software to generate encodings of trees from images.