Class2Str: End to End Latent Hierarchy Learning

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY HYDERABAD

Soham Saha*, Girish Varma*, C.V. Jawahar

Center for Visual Information Technology, Kohli Center for Intelligent Systems, IIIT Hyderabad

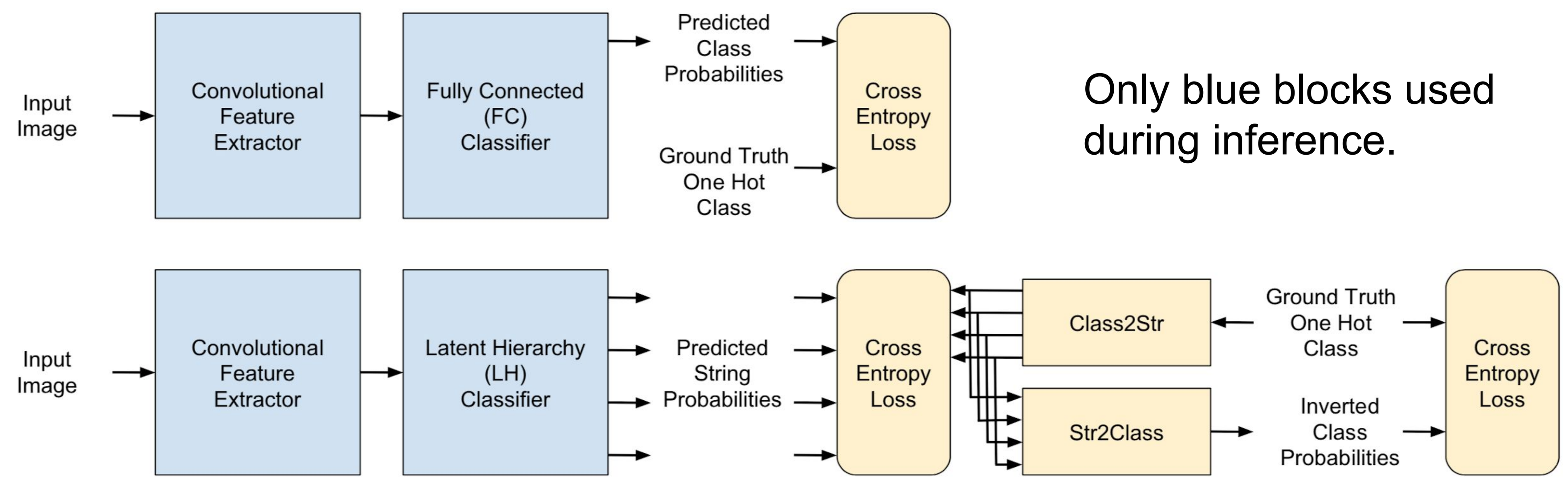
IAPR ICPR 2018 INTERNATIONAL CONFERENCE ON PATTERN RECOGNITION

MOTIVATION

- One-hot representation of classes assumes class independence which might not be the case.
- Classes have visual similarity and often form a hierarchy. Learning this hierarchy increases the ease of classification.
- Learning a latent hierarchy explicitly in the neural network architecture, could also improve efficiency.

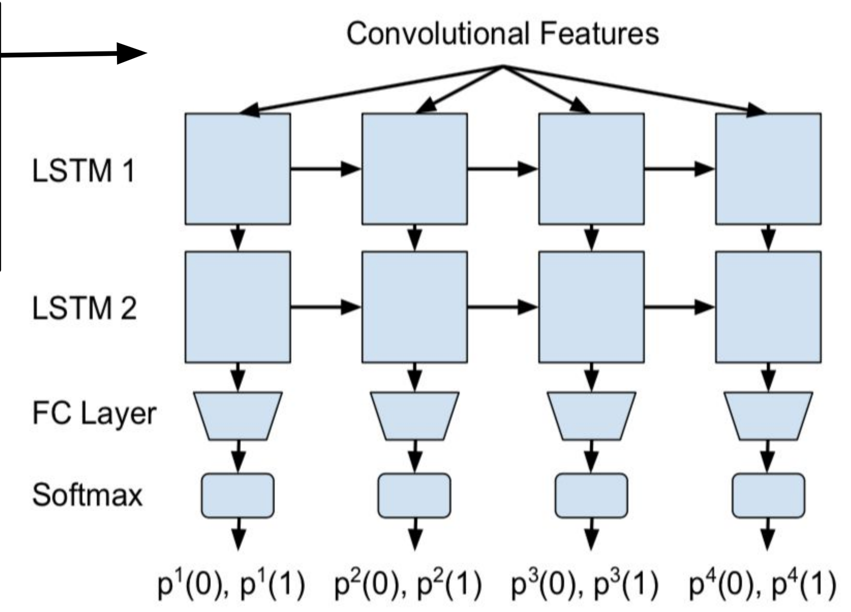
APPROACH

- We model hierarchy as a binary tree and map every leaf node to the prefix string given by the path in the tree from root to leaf.
- Deep Neural networks consists of a Feature Extractor followed by a Classifier Network which is Fully Connected.
- We replace the fully connected classifier with a latent hierarchy classifier which learns a latent hierarchy by means of a **Class2Str** network.
- To ensure that the Class to String mapping learnt is a one-one mapping, we also have a **Str2Class** network which inverts the mapping.
- **Class2Str** and **Str2Class** is replaced by the learnt static map from class to string during inference, increasing the efficiency.



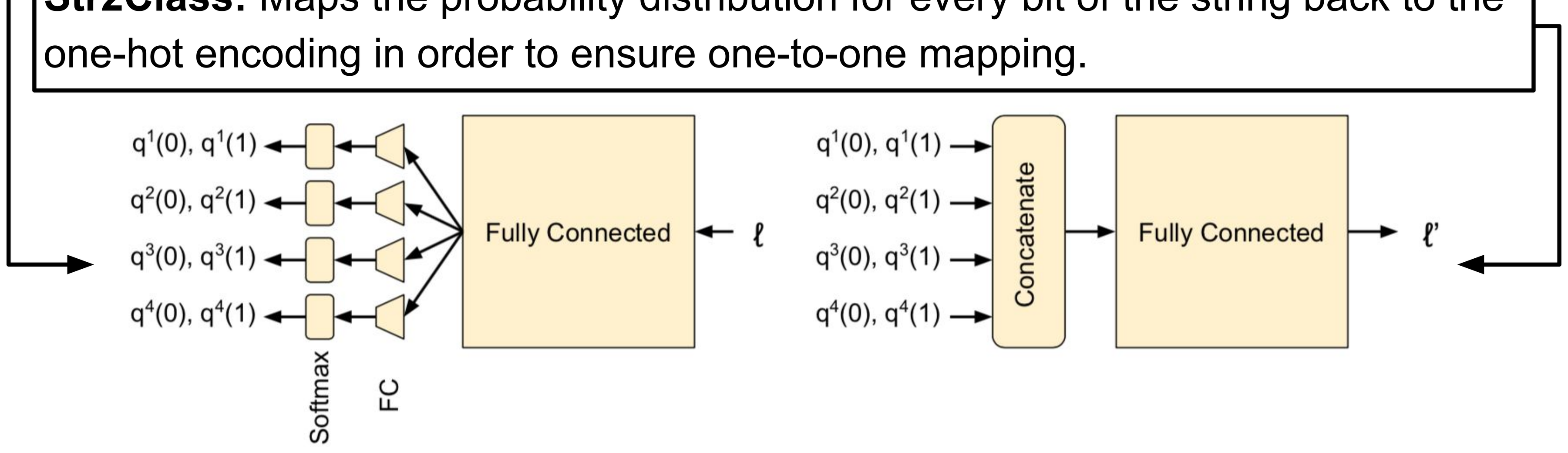
Only blue blocks used during inference.

Latent Hierarchical Classifier: Classifier network which predicts the probability distribution of the strings (labels for our method).



Class2Str: Takes the one-hot encoding of the class labels as input and maps it to a probability distribution for every bit of a string.

Str2Class: Maps the probability distribution for every bit of the string back to the one-hot encoding in order to ensure one-to-one mapping.



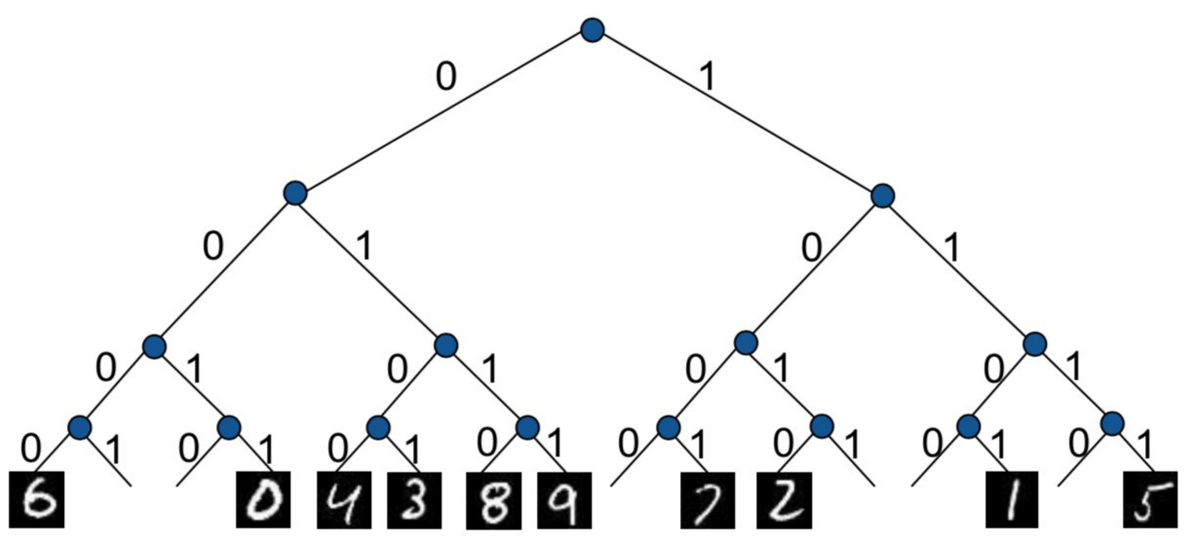
LEARNING A LATENT HIERARCHY

We propose a **structured loss** for training and learning a latent hierarchy. The loss function we use is as under:

$$\alpha H(\ell, \ell') + \beta \sum_{i=1}^{L} \mu^i H(p^i, q^i) - \gamma \sum_{i=1}^{L} (q^i(0)^2 + q^i(1)^2) + \delta L^2(W)$$

- The first component of this loss calculates the cross entropy between the Class2Str and Str2Class outputs.
- The second component is the cross entropy between the predicted and the learnt probabilities of each bit of the string.
- The third component is for introducing bias in the learnt string probabilities and ensuring that they are close to 1 or 0.
- The final component acts as a regularization for the entire weight spapce.

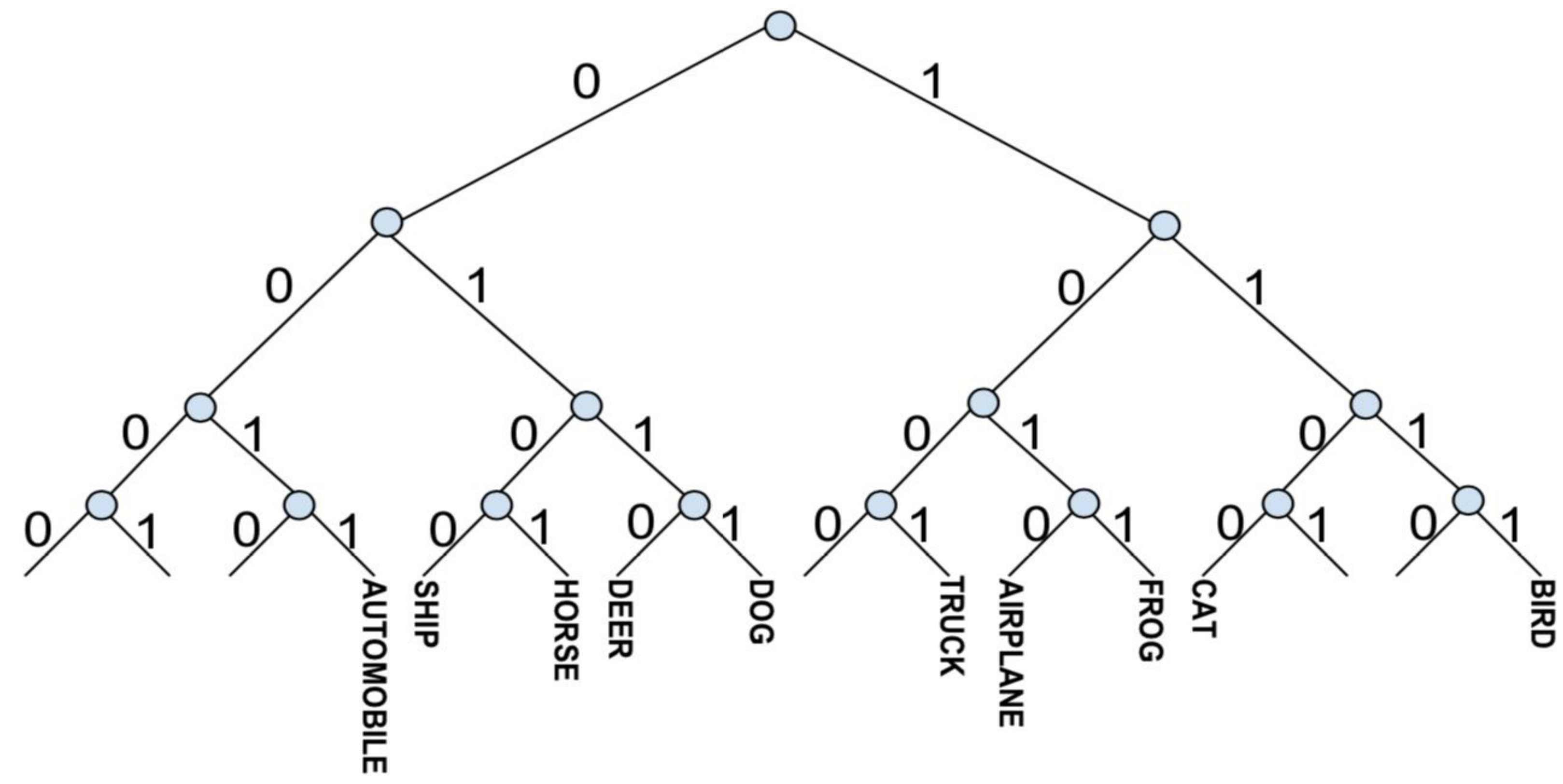
We construct a tree from the learnt string embedding to form a hierarchy. The learnt MNIST hierarchy shows classes like 3,8 and 9 closer to each other.



Training to learn a multi-level hierarchy

$$\sum_{i=1}^{L} \mu^i H(p^i, q^i)$$

This component ensures that a multi-level hierarchy is learn by penalizing misclassification at an earlier node of the tree more than the later nodes. The decay factor $\mu < 1$ ensures this.

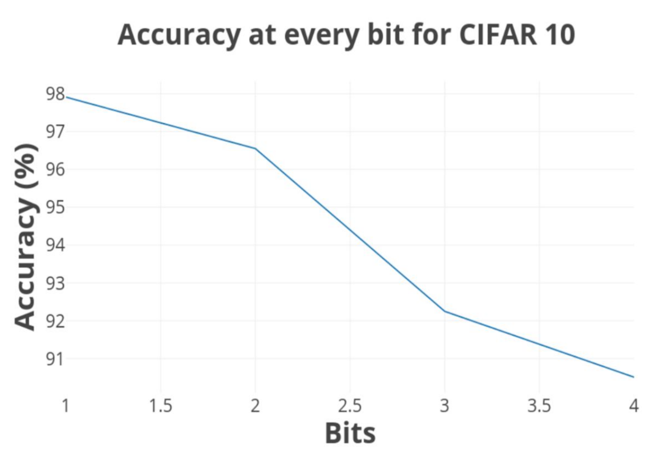


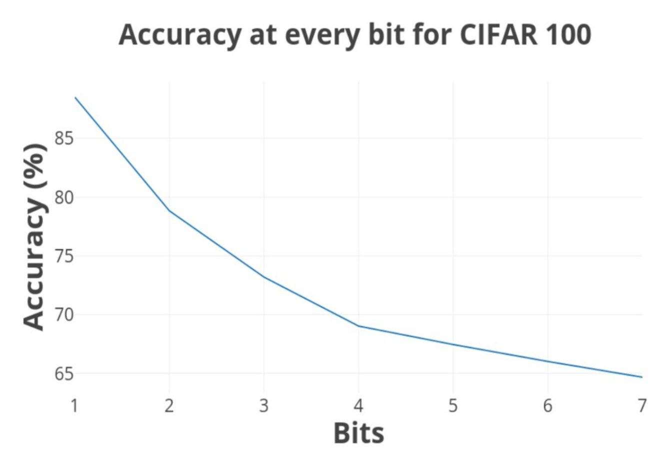
Learnt hierarchy for CIFAR 10 shows that visually similar objects such as dog, deer and horse have a longer common prefix. Hence, this results in them being closer to each other in the leaves of the hierarchy tree.

RESULTS

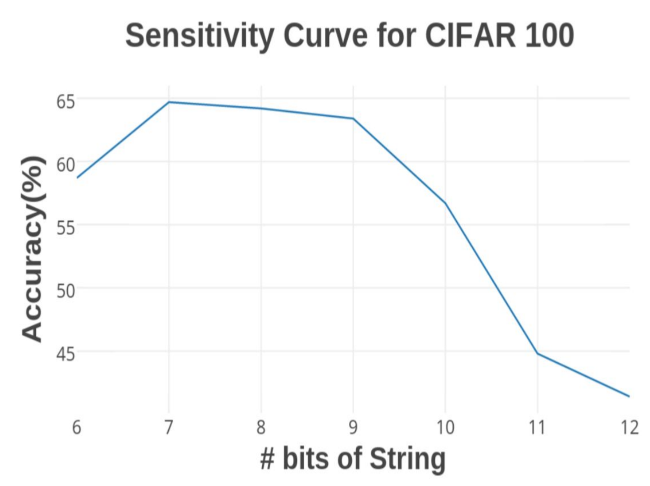
98% parameter reduction for CIFAR 100 and 41% reduction for Imagenet 1K without loss in accuracy.

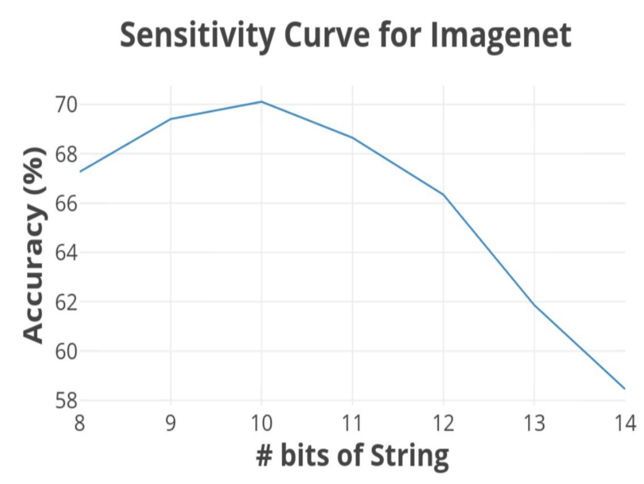
Dataset	% Acc of FC Classifier	% Acc of LH Classifier	% Acc of reduced FC Classifier	#parameters in FC Classifier	#parameters in LH Classifier	Reduction in parameters	Reduction in test time per image
MNIST	99.38	99.36	98.45	1.61 M	31 K	98%	13.9%
CIFAR 10	90.43	90.51	88.40	1.58 M	6 K	99%	15.9%
	% Acc in Maxout: 90.65, Network in Network: 91.2, Deeply Supervised Networks : 91.78						
CIFAR 100	64.65	64.67	57.90	1.58 M	30 K	98%	14.8%
	% Acc in Maxout: 61.43, Network in Network: 64.32, Deeply Supervised Networks : 65.43						
Imagenet 1K	70.51	70.11	67.88	123.63 M	72.42 M	41%	5.5%





Accuracy at every bit for the predicted string. The string length is a hyperparameter which has been varied.

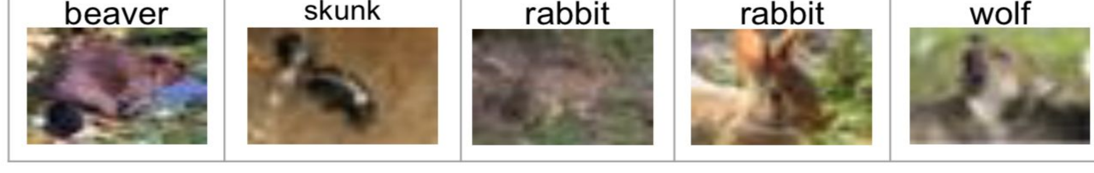




Classification accuracy for using strings of different length for the prediction. For each dataset, we get an optimum length for representation.

Qualitative Results

Learnt strings with the highest length of the longest common prefix (for CIFAR) show that visually similar classes are close to each other in the learnt hierarchical tree.

Image Examples (Classes)	Strings	Length of the Longest Common Prefix
shrew, shrew, porcupine, otter, otter 	0011100 (shrew) , 0011110 (porcupine) , 0011111 (otter)	5
lobster, aquarium_fish, flatfish, crocodile, ray 	0101010 (lobster) , 0101011 (aquarium fish) , 0101100 (flatfish) , 0101110 (crocodile) , 0101111 (ray)	4
maple_tree, lawn_mower, lawn_mower, tiger, worm 	1011011 (maple_tree) , 1011101 (lawn_mower), 1011110 (tiger), 1011111 (worm)	4
pine_tree, pine_tree, palm_tree, palm_tree, table 	0111101 (pine_tree) , 01111110 (palm_tree), 0111111 (table)	5
beaver, skunk, rabbit, rabbit, wolf 	1111000 (beaver), 1111001 (skunk) , 1111010 (rabbit) , 1111011 (wolf)	4

soham.saha@research.iiit.ac.in, girish.varma@iiit.ac.in, jawahar@iiit.ac.in