Analysis of CNN working with logical decision functions in the task of computer tomography images recognition

Kozinets Roman , Berikov Vladimir

NOVOSIBIRSK STATE UNIVERSITY
*The real science

May 4, 2021

Overview

Introduction

Learnable bag of visual words, patterns

Decision Tree

NeuralPatternTree

Experiments

Introduction

CNNs have achieved superior performance, especialy in computer vision, but their complexity led to increasing demand for interpretability and explainable.

Our study aims to solve explainability of CNN and create classification network with explicit reasoning.

General idea

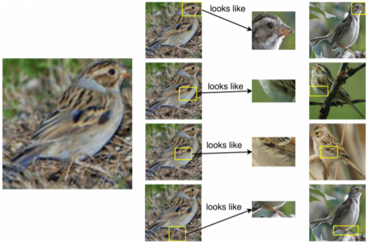
- Competitive performance to classic CNNs
- High interpretability
- End-to-end training and inference
- Image level labels

Combine visual based reasoning with logical-based decision making process to simulate human visual recognition and propagate it in neural network architecture. Human-like visual recognition : This object of type A because it has some part specific for A-type.
Components:

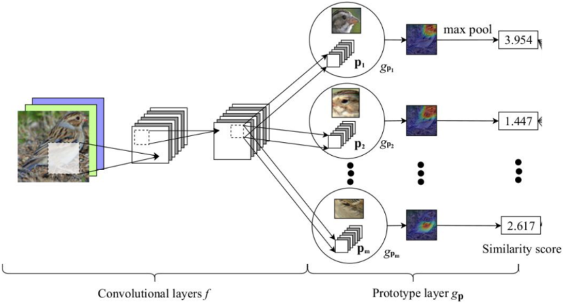
- CNN feature extraction
- learnable bag of visual words (patterns/prototypes)
- Differentiable decision tree as classifier

Patterns

idea was taken from "This Looks Like That: Deep Learning for Interpretable Image Recognition". Patterns represent specific for some classes object parts. To classify image we focus on parts of image and compare with learned patterns, image class is equal to class of images which have the closest patterns.

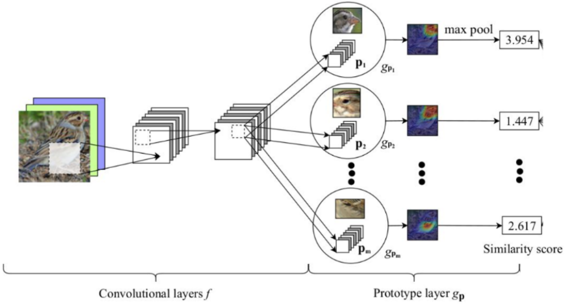


Pattern layer architecture



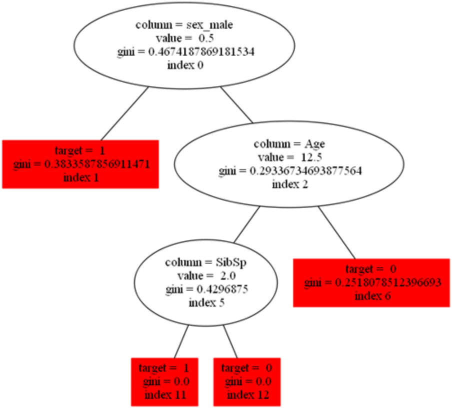
Given on input image x, Convnet produce feature map $z = f(x)$ with size $H \times W \times D$ where W, H the width and height of map and D is depth or number of features. The network learn n patterns $P = \{p_j\}$, each pattern is a vector with shape $1 \times 1 \times D$. Each pattern represent latent representation of some special part of classified object on image.

Pattern layer architecture



Let x be a input image, $z = f(x)$ last convolutional output and P - pattern layer. The g_j pattern unit compute L2 squared distance between all patches of feature map z and p_j pattern. activation feature of pattern layer convert distance to similarity score and pool max similarity score by global max pool of distance map.

Decision Tree



NOVOSIBIRSK STATE UNIVERSITY
*The real science

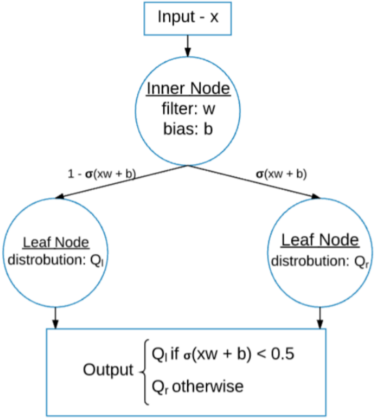
Soft Decision Tree

- Inner node

$$p_i(x) = \sigma(xw_i + b_i)/\tau)$$

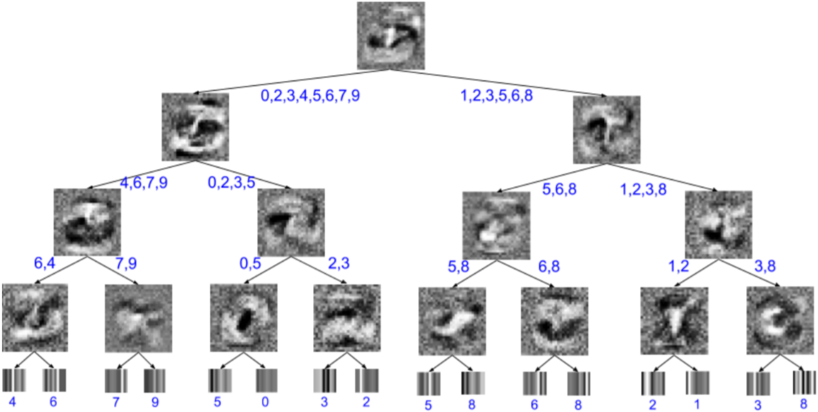
- Leaf node

$$Q_k^l = \frac{(exp(c_k^l))}{\sum exp(exp(c_k^l))}$$



$$L(x) = -log(\sum_{l \in LeafNodes} P^l(x) \sum T_k log(Q_k^l))$$

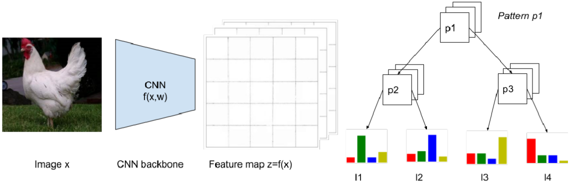
Soft Decision Tree



NeuralPatternTree I

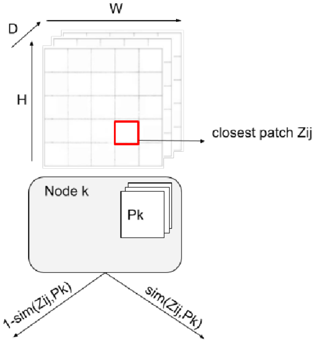
Structure:

- CNN backbone
- Pattern layer
- Soft decision tree classifier



NeuralPatternTree I

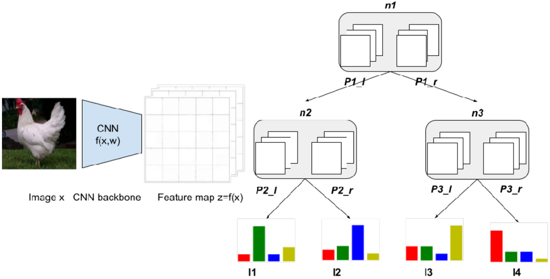
$z = f(x)$ feature map, Pk pattern vector of node N $sim(a, b) = exp(-||a - b||)$ similarity between a, b. $N_{k,right} = (exp(-||Z - P_k||)$, where Z is whole feature map of last conv layer and $N_{k,left} is 1 - N_{k,right}$. Thus node k output similarity closest path to pattern Pk on the right branch and difference on the left branch.



NeuralPatternTree II

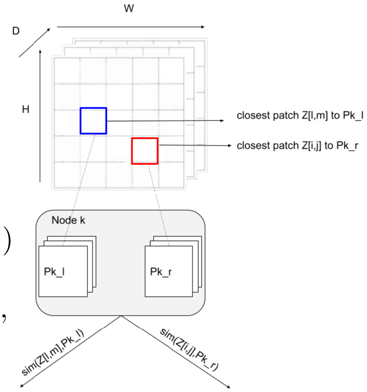
Structure:

- CNN backbone
- Double Pattern layer
- Soft decision tree classifier



NeuralPatternTree II

X input Image, $z = f(x)$ feature map Node N_k take z as input.



$$dist_{l,best} = \min_{i,j}(dist(Z(i,j), P_{k,left}))$$

$$dist_{r,best} = \min_{i,j}(dist((Z(i,j), P_{k,right}))$$

$$Similarity(N_k) = softmax(dist_l, dist_r)$$

$$Node(z)_{k,left} = \frac{exp(-dist_{left})}{\sum_{left,right} exp(-dist_i)},$$

$$Node(z)_{k,right} = \frac{exp(-dist_{right})}{\sum_{left,right} exp(-dist_i)}.$$

NeuralPatternTree Training

- Initialization
  CNN backbone f,g pattern layer,h soft decision tree.
- Stage 1 Training

$$\min_{P,w_f,w_h} \frac{1}{n}\sum_{i=1}^{n} CrsEntr(h \circ g \circ f(x_i), y_i))$$

- Stage 2 Projection

$$p_i \leftarrow argmin_{z \in Z_i} ||z - p_j||_2,$$

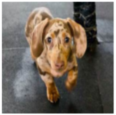
$$where Z_i = \{z : z \in patches(f(x_j)) \forall i.s.t, y_i = k\}.$$

NOVOSIBIRSK STATE UNIVERSITY
*The real science

Animal Dataset

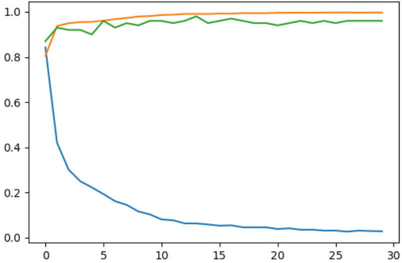
- Classes
  - cow
  - horse
  - dog
  - cat
  - chicken
- Train size 13867, Test size 250
- Augmentations
  - brightness, contrast and saturation
  - random rotation, flip



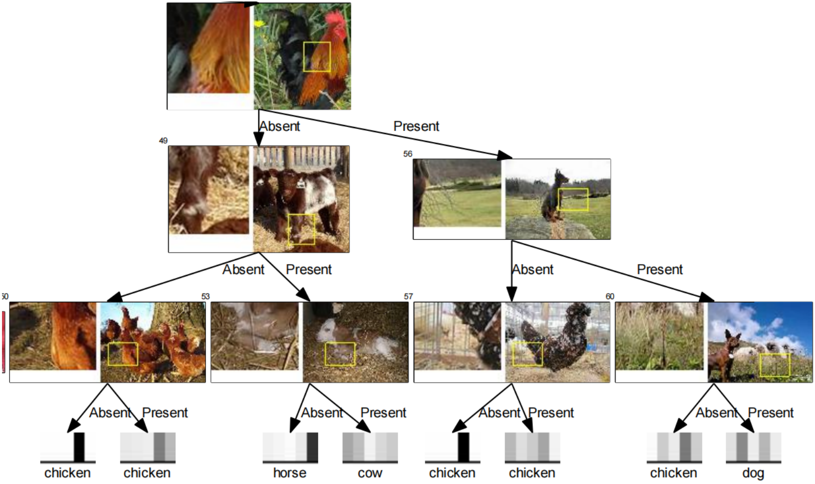




Results

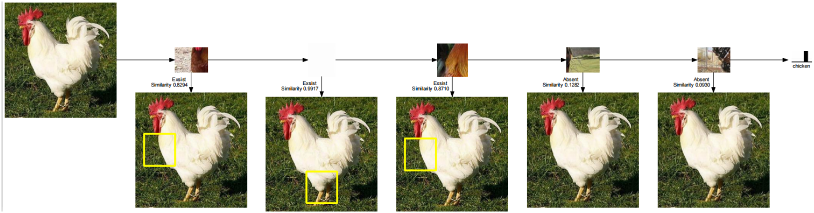
Model	Accuracy (test/train)
NeuralPatternTree I	97.6 / 99.0
NeuralPatternTree II	99.0 / 99.8
ResNet50	99.1 / 99.92



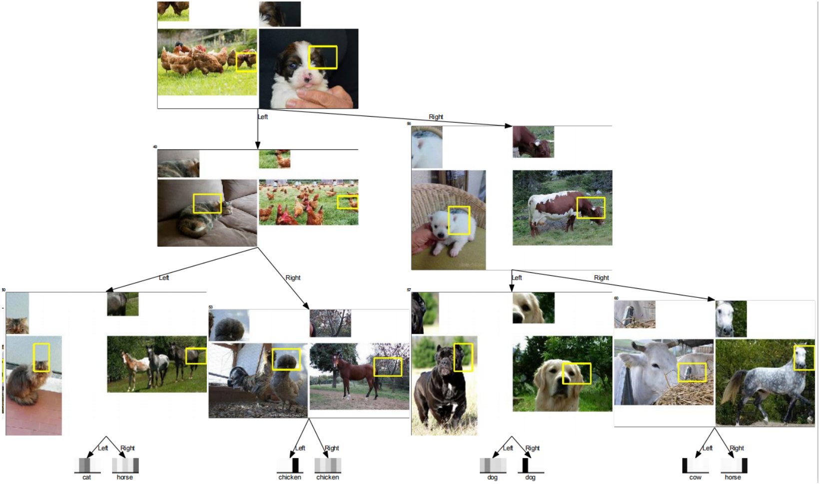
Global Prediction I



Local prediction I



Global prediction II

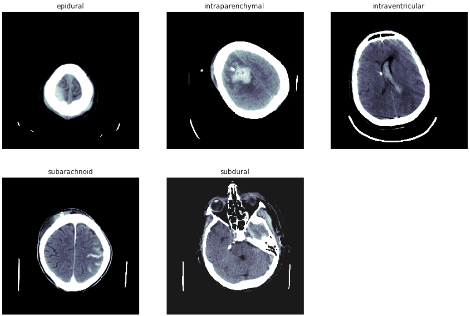


Local prediction II

Error

Intracranial Hemorrhage classification

- Classes
  - epidural
  - intraparenchymal
  - intraventricular
  - subarachnoid
  - subdural
- Train size 41071, Test size 500, image size (224,224) Preprocessing: CT image windowing to 3-channel image
  - brain (40,80)
  - soft (80,200)
  - subdural (40,380)
- Augmentations: Flip,Rotation,Crop



Results

Model	Accuracy (test/train)
NeuralPatternTree I	70.5 / 87.3
NeuralPatternTree II	83.5 / 97.1
ResNet50	76.8 / 92.1

Visualizations

Coming soon

Conclusion

Proposed model comparable to classic CNN. Neural pattern tree require less patterns then ProtoNet. Can be training in end-to-end procedure. Model has explainable features and clear classification process.

Thank for attention