DEEP NEURAL DECISION FORESTS (NOTES)

Kontschieder, Fiterau, Criminisi and Bulo

November 20, 2015

Microsoft Research, CMU and Fondazione Bruno Kessler

Memory

Random forests

- Amazing (high dimensional data, non linear, multi-class, easily distributed)
- But need decent features

Modern ML/Deep learning

- Unify learning feature representations with their classifiers
- no need for data scientists

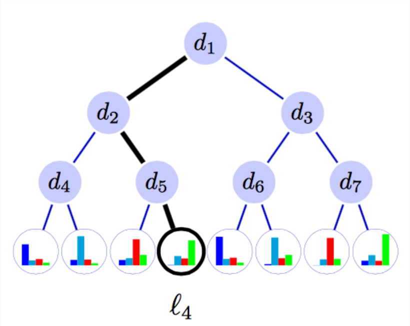
Summary

Add feature representations to Random Forests

- stochastic, differentiable/back-prop'able decision tree
- minimizes arbitrary loss function
- appended to CNN (less interesting)

Model

- X, Y: input and output space
- L: terminal/prediction node
- N: decision/splitting node



Prediction

Each decision node, $n \in N$, splits stochastically. Implemented as Bernoulli with mean

$$d_n : X \rightarrow [0, 1] \tag{1}$$

Each terminal node/leaf holds a probability distribution, π_l over the classes, Y.

Therefore, given an input, x, and a tree, T, the probability of y is:

$$P_T(y|x) = \sum_{l \in L} \pi_{l,y} \mu_l(x) \tag{2}$$

where $\mu_l(x)$ is the probability of ending up in l.

Binary tree allows simple expression for $\mu_l(x)$. Let $l \swarrow n$ and $n \searrow l$ be 1 if l is left and right respectively of n, otherwise 0.

$$\mu_l(x) = \prod_{n \in N} d_n(x)^{l \swarrow n} \left(1 - d_n(x)\right)^{n \searrow l} \tag{3}$$

(Note: adding randomness is a nice trick to preserve differentiability)

Implementation

- decision nodes are arbitrary neural networks, with sigmoid activation
- leaf/terminal nodes are normalized parameters
- forest is an ensemble of trees, with the prediction being the average across trees

Learning

- SGD for internal nodes (gradients in paper)
- given choice of parameters for internal nodes, choice of parameters for leafs can be framed as solution to fixed point problem, with convergence guarantees