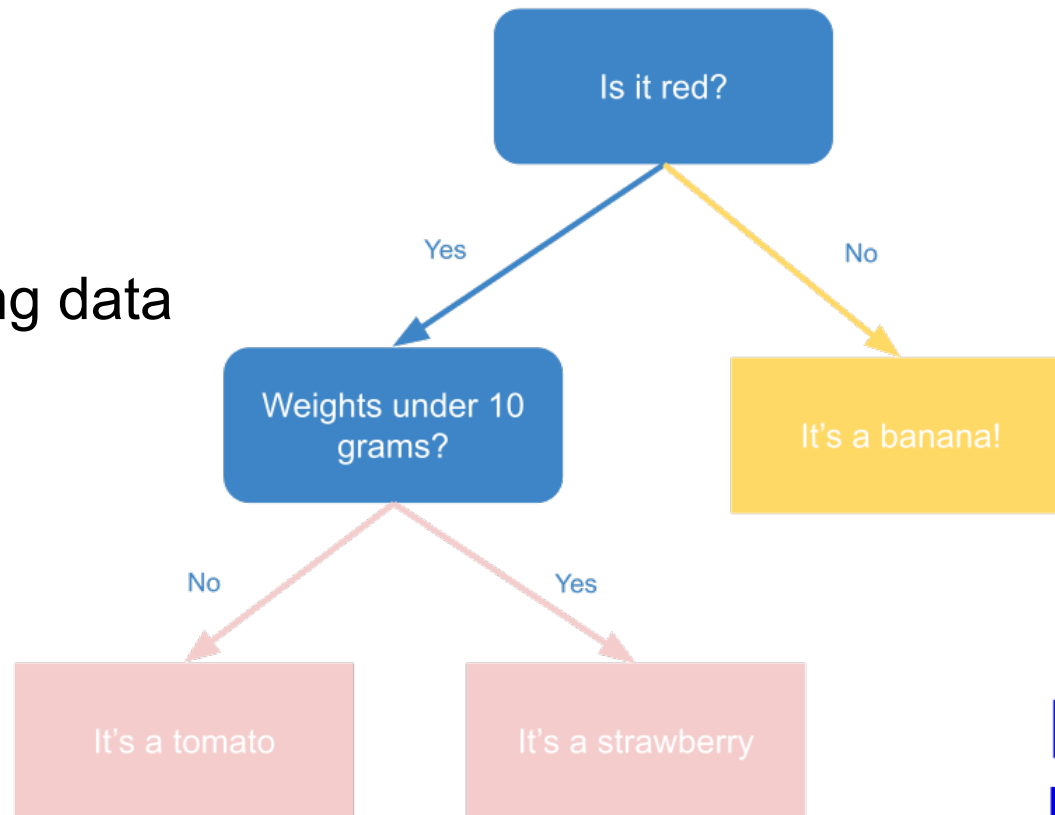# Deep Forest

By Jakub Šimuni

Deep Forest

Motivation:

- Create a deep network that isn't a neural network
- Reduce the number of hyperparameters required
- Use an ensemble of other simpler classifiers

FI
MUNI

Decision tree

- Simplest estimator in the ensemble
- Created from training data

Bagged trees

- **B**ootstrapped **Agg**regat**ed trees**
- Generated from random picks in the training data
- Trained with duplicate entries and excluded entries (reduces overfitting)

Weights under 20 grams?

No → It's a tomato

Yes → It's a strawberry

Is it red?

Yes → It's a tomato

No → It's a banana!

FI MUNI

Random Forest

- An ensemble of bagged trees
- The result is decided from the results of all trees (either averaging or voting)
- Leaves are classes
- Surprisingly accurate for something so simple

F I

M U N I
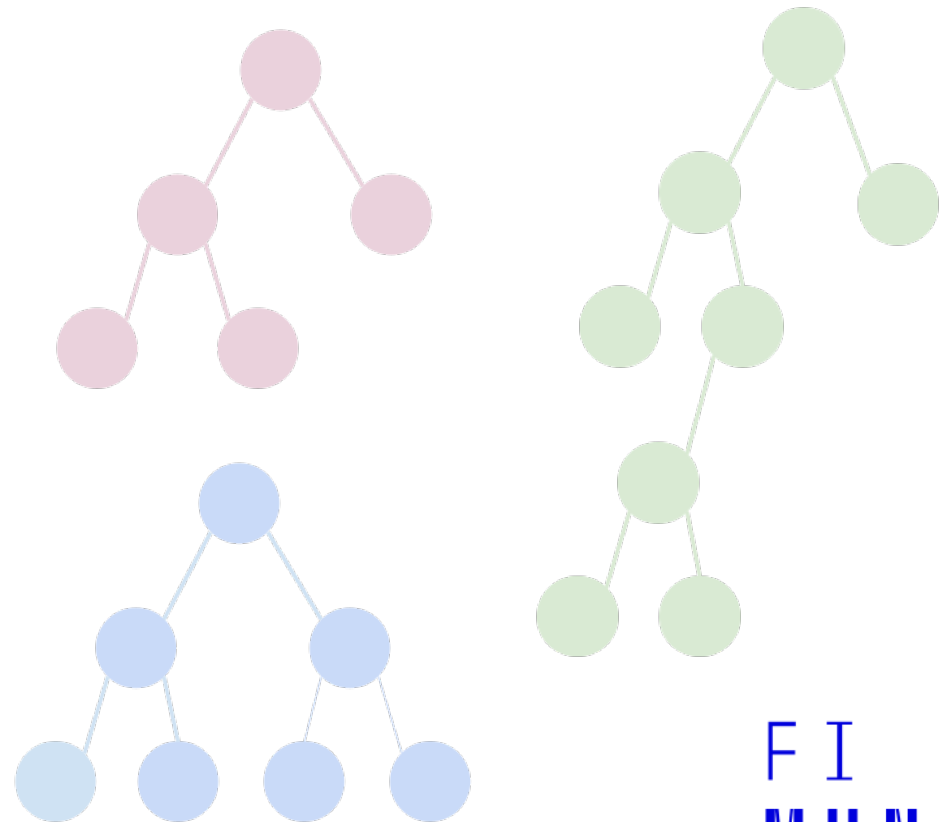
gcForest (Deep Forest implementation)

- (Multi-**g**rained) **c**ascade **Forest**
- An ensemble of random forests
- Create layer of random forests
- Take class distribution of the previous layer and add it into the original input
- Repeat till there is no significant gain

Source:
https://www.ijcai.org/proceedings/2017/0497.pdf

FI
MUNI
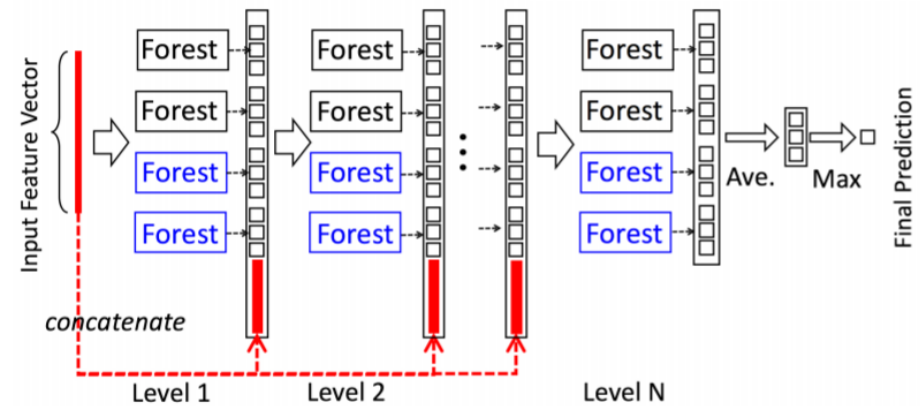
gcForest (Deep Forest implementation)

- Multi grain version saves space
- Input vectors are split and used for training in rounds (e.g., 1st part will train the 1st layer, 2nd part will train the 2nd layer, and so on)
- Implementation of multi-grain is dependent on used classifiers (Random forest is won't be affected by the split, so they are ideal, some other structures might be)

F I
M U N I

Experiences with gcForest

Public gcForest implementation:
https://arxiv.org/abs/1702.08835v2

FI
MUNI

Sources for further research

Original work: https://www.ijcai.org/proceedings/2017/0497.pdf
https://www.cv-foundation.org/openaccess/content_iccv_2015/
papers/
Kontschieder_Deep_Neural_Decision_ICCV_2015_paper.pdf
https://www.statistik.uni-muenchen.de/institut/institutskolloquium/
pdf_daten/ws1718/munich_2017.pdf

FI
MUNI