

Decision Trees, Random Forests and Neural Networks



Decision Tree

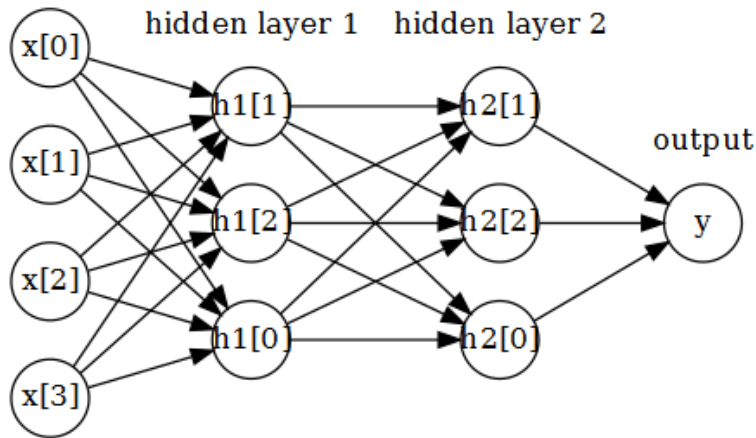
- ▶ Learn a hierarchy of if/else questions
- ▶ 3 main parameters:
 1. How deep, i.e. how many iterations?
 2. How many samples at least at a leaf?
 3. How many samples at most at a leaf?
- ▶ Typically you adjust one parameter only
- + Easy to understand and no scaling necessary
- Tend to overfit

Random Forests

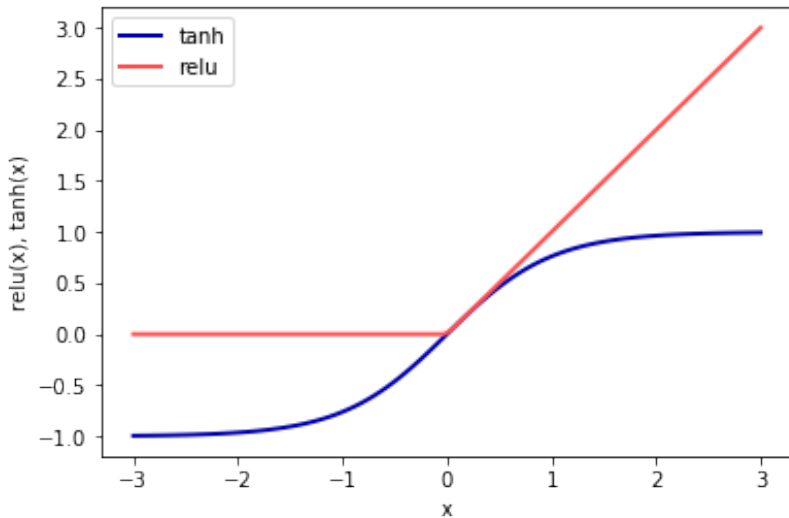
- ▶ Ensembles of Decision Trees, that are slightly different from each other
- ▶ To reduce overfitting
- ▶ Two parameters (on top of each tree's parameter):
 1. How many trees?
 2. How many features at most should each tree look at?
- + All the benefits of trees, yet less overfitting
 - No inspection possible, heavy CPU usage (but easy to parallelize), and not replicable

What is a Neural Network?

inputs



Activation function



from: Andreas Müller and Sarah Guido (2016): Introduction to Machine Learning with Python, O'Reilly

Neural Networks

- ▶ Expects standardized data
- ▶ Many parameters:
 1. How many layers?
 2. How many units (nodes) (per layer)?
 3. Which activation function?
 4. Regularization strength?
 5. Underlying algorithm? (and their respective parameters)
 6. ...
- + Can be infinitely complex, often beat other algorithms
- Much slower than other algorithms