

When to use decision trees?

Both decision trees, including tree ensembles as well as neural networks are very powerful, very effective learning algorithms. When should you choose one or the other? Let's look at some of the pros and cons of each. Decision trees and tree ensembles will often work well on tabular data, also called structured data. What that means is if your dataset looks like a giant spreadsheet then decision trees would be worth considering.

For example, in the housing price prediction application we had a dataset with features corresponding to the size of the house, the number of bedrooms, the number of floors, and the age at home. That type of data stored in a spreadsheet with either categorical or continuous valued features and both for classification or for regression task where you're trying to predict a discrete category or predict a number.

All of these problems are ones that decision trees can do well on. In contrast, I will not recommend using decision trees and tree ensembles on unstructured data. That's data such as images, video, audio, and texts that you're less likely to store in a spreadsheet format. Neural networks as we'll see in a second will tend to work better for unstructured data task.

One huge advantage of decision trees and tree ensembles is that they can be very fast to train. You might remember this diagram from the previous week in which we talked about the iterative loop of machine learning development. If your model takes many hours to train then that limits how quickly you can go through this loop and improve the performance of your algorithm.

Decision Trees vs Neural Networks

Decision Trees and Tree ensembles

- Works well on tabular (structured) data
- Not recommended for unstructured data (images, audio, text)
- Fast



But because decision trees, including tree ensembles tend to be pretty fast to train, that allows you to go to this loop more quickly and maybe more efficiently improve the performance of your learning algorithm. Finally, small decision trees maybe human interpretable. If you are training just a single decision tree and that decision tree has only say a few dozen nodes you may be able to print out a decision tree to understand exactly how it's making decisions.

I think that the interpretability of decision trees is sometimes a bit overstated because when you build an ensemble of 100 trees and if each of those trees has hundreds of nodes, then looking at that ensemble to figure out what it's doing does become difficult and may need some separate visualization techniques. But if you have a small decision tree you can actually look at it and see, oh, it's classifying whether something is a cat by looking at certain features in certain ways.

If you've decided to use a decision tree or tree ensemble, I would probably use XGBoost for most of the applications I will work on. A slight downside of a tree ensemble is that it is a bit more expensive than a single decision tree.

If you had a very constrained computational budget you might use a single decision tree but other than that setting I would almost always use a tree ensemble and use XGBoost in particular. How about neural networks? In contrast to decision trees and tree ensembles, it works well on all types of data, including tabular or structured data as well as unstructured data. As well as mixed data that includes both structured and unstructured components.

Decision Trees vs Neural Networks

Decision Trees and Tree ensembles

- Works well on tabular (structured) data
- Not recommended for unstructured data (images, audio, text)
- Fast
- Small decision trees may be human interpretable

Whereas on tabular structured data, both neural networks and decision trees are often competitive on unstructured data, such as images, video, audio, and text, a neural network will really be the preferred algorithm and not the decision tree or a tree ensemble. On the downside though, neural networks may be slower than a decision tree. A large neural network can just take a long time to train.

Other benefits of neural networks include that it works with transfer learning and this is really important because for many applications we have only a small dataset being able to use transfer learning and carry out pre-training on a much larger dataset that is critical to getting competitive performance. Finally, if you're building a system of multiple machine learning models working together, it might be easier to string together and train multiple neural networks than multiple decision trees.

The reasons for this are quite technical and you don't need to worry about it for the purpose of this course. But it relates to that even when you string together multiple neural networks you can train them all together using gradient descent. Whereas for decision trees you can only train one decision tree at a time. That's it. You've reached the end of the videos for this course on Advanced Learning Algorithms.

Decision Trees vs Neural Networks

Decision Trees and Tree ensembles

- Works well on tabular (structured) data
- Not recommended for unstructured data (images, audio, text)
- Fast
- Small decision trees may be human interpretable

Neural Networks

- Works well on all types of data, including tabular (structured) and unstructured data
- May be slower than a decision tree
- Works with transfer learning
- When building a system of multiple models working together, it might be easier to string together multiple neural networks

Thank you for sticking with me all this way and congratulations on getting to the end of the videos on advanced learning algorithms. You've now learned how to build and use both neural networks and decision trees and also heard about a variety of tips, practical advice on how to get these algorithms to work well for you. But even if all that you've seen on supervised

learning, that's just part of what learning algorithms can do. Supervised learnings need labeled datasets with the labels Y on your training set.

There's another set of very powerful algorithms called unsupervised learning algorithms where you don't even need labels Y for the algorithm to figure out very interesting patterns and to do things with the data that you have. I look forward to seeing you also in the third and final course of this specialization which should be on unsupervised learning. Now, before you finish up this course I hope you also enjoy practicing the ideas of decision trees in their practice quizzes and in their practice labs.