Artificial Neural Networks 5

Roger Barlow
The University of Huddersfield

CODATA School on Data Science

Autumn 2021

Everything Else

- Regression with Neural Networks
- Boosted Decision Trees
- Support Vector Machines
- Unsupervised Learning
- Deep Networks
- Problems

Using Neural Networks for Regression

Example: house price depends on floor area, number of rooms, location, type of street, distance to school, garden....

Take data from recent local house sales and train neural network

Very straightforward: 'truth' is now desired value (suitably scaled), rather than binary 0 or 1.

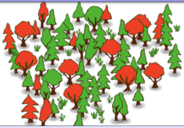
Quicker than messing around with polynomials.

Alternatives (1) Boosted Decision Trees

Basic Tree: Find most sensitive variable and best cut on it, dividing sample in two (branching)
Repeat in each of 2,4,8,16…
branches until all events classified

Bagging (Bootstrap Aggregation): Avoid overtraining by creating many trees, sampling with replacement. Then take majority vote

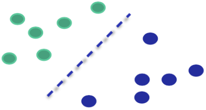
Random Forest: as Bagging, but each tree restricted to random subset of input variables. Avoids trees all going for the obvious and looking the same



Boosting: weight events that early trees find hard to classify, so later trees focus on them

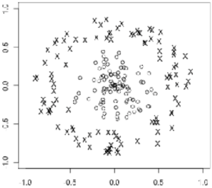
Alternatives (2) Support Vector Machines

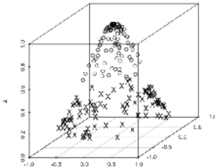
Draw the best line/plane/hyperplane dividing events into 2 classes (widest margin + fewest misclassified)



In choosing this line/plane/whatever only the datapoints near the boundary need be considered - the 'support vector'

As necessary, distort into higher dimensions to make this work





Which is best?

Generalisations

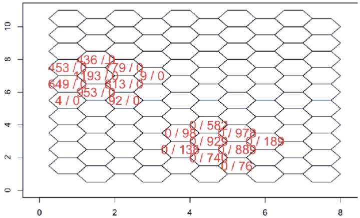
BDTs cope well with categorical data. ANNs less well, SVMs even worse.

SVMs work best when there is a clear divide (even if it's complicated) between signal and background

Word on the street is that deep nets can beat BDTs

The only way to be sure it so try and see

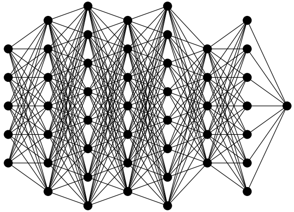
Single 2D layer - square or hexagonal topology
Each has set of weights - same number as input data
There is an R package 'kohonen' you can download



1. Present a data item
2. Find the node whose weights are best match to this event
3. Adjust this node's weights to be more like the data item
4. Adjust nearby nodes' weights to be more like the data item, but by smaller amount
5. Adjust next-to-nearby nodes' weights to be less like the data item
6. Repeat

After training, network will group data into clusters of similar events

Deep Learning
Networks with many layers



Back propagation doesn't work well for more than a few layers
A weight in a late layer affects the result and is rewarded or punished accordingly - OK
A weight in an early layer affects the result through many different paths, some good some bad, and rewards/punishments tend to wash out

Instead: Encourage early layers to pick out *features* - maximise spread of outputs
Then use supervised backpropagation learning in later layers
Use this to recognise useful features and prune useless ones
Needs enormous training samples and superfast computers - which we have

Very powerful and enables many new applications

Data pre-processing

Chances of a successful network are greatly improved by intelligent and knowledgable pre-processing of data.

Don't just throw the numbers at the network and expect miracles.

Shifting, Scaling. Principal Component Analysis.

For images: rotation, stretching, and feature extraction

Problems

Getting it wrong: the tank story

A NN was trained to tell ‘our’ tanks from ‘their’ tanks using photographs: it did unbelievably well. But it turned out that ’our’ tanks were photographed on sunny days and ’their’ tanks on cloudy days. This is almost certainly an urban myth, but it’s a good story. (Recent variant: the NN trained to recognise criminal faces from photographs...)

Sample bias

Facial recognition AI trained on unrepresentative samples (e.g. students in predominantly male white universities) fails to handle real-world data. (One reason it’s illegal in many places)

The prosecutor’s fallacy

Suppose a NN finds a data feature for failing students, or loan defaulters... To treat everyone with that feature as a failing student/loan defaulter ignores the fact that most students pass, most loans are repaid...

Conclusion

There are many types of neural network (and BDTs and SVMs)

They have many uses and there is lots of software available

They should not be left to the experts: they're not hard to use

Activity for you

Find a way to apply one in your research