

Summer job at the National Physical Laboratory, my Part II essay on perceptrons and meeting David Marr

Mike Gordon

Contents

The National Physical Laboratory (NPL)	1
Part II essay on perceptrons	5
Meeting David Marr	6

The National Physical Laboratory (NPL)

In the summer vacation following my second undergraduate year I got a job at the NPL. I was assigned to [Julian Ullman](#), in whose office I was put. Apparently he is now [in his 80s and is still coding and writing academic papers](#), though I haven't seen him since 1969.

I chatted a lot with Ullman about pattern recognition theory in general. I was also given a concrete project to use NPL software to write a program to recognise hand written characters. I don't have the report I wrote and my memory is fuzzy, so some of what follows is rather any vague and may not be entirely accurate. The character recognition software I used was a domain-specific language implemented by Don Bell, a scientist at NPL. I remember the language being called "DBTL" – Don Bell's Tree Language – though maybe that was just my private name. The language is described in Bell's paper "[Display Assisted Design of Sequential Decision Logic for Pattern Recognition](#)".

A program in DBTL would read in a file containing digitised characters and perform a sequence of tests on each character to try to identify it. A file of the results for each character identification attempt would be printed. DBTL was

an if-then-else language for writing decision trees. The tests at the nodes of the trees checked for features due to Parks like those shown in the extract from page 174 of Ullman's book in Figure 1 in Figure 1 below.

Parks³⁴³ has described a system which uses features such as \perp , Υ , γ , \vdash , \angle , \vee , \sphericalangle , and limb endings in all of the eight orientations $0^\circ, 45^\circ, 90^\circ, \dots, 315^\circ$, except, for example, that only 0° and 45° orientations of the \perp feature are used, since its 0° and 90° orientations are indistinguishable.

Figure 1:

The following diagram (see Figure 2) is from Bell's paper and shows features being extracted for the numeral "3"

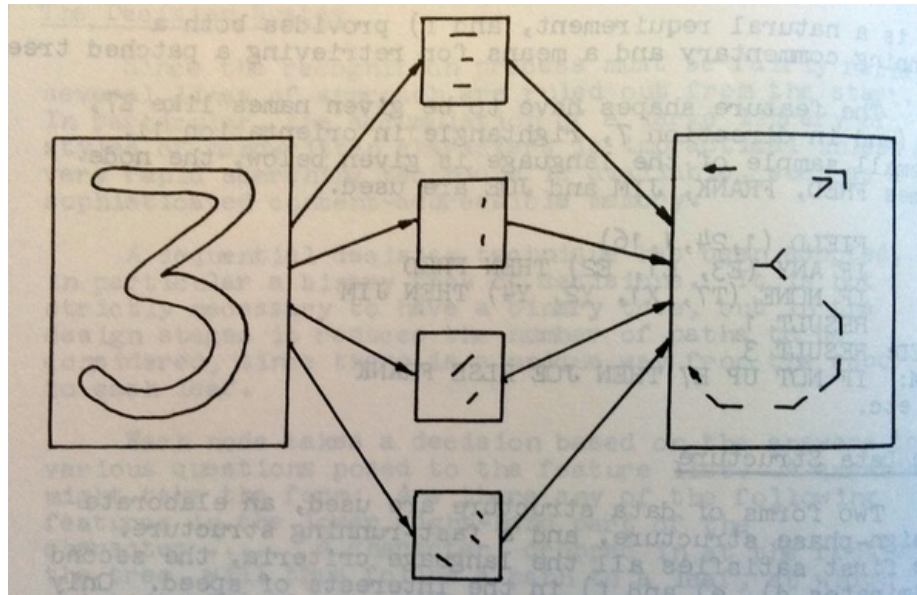


Figure 2:

My job was to look at lots of instances of features corresponding to single characters and then to write a decision tree in DBTL to recognise the character. The two extracts from Bell's paper below (in Figures 3 and 4) give an impression of what DBTL code is like.

DBTL was implemented in the "high-level assembly language PL-516" that ran on a Honeywell DDP-516 mini-computer (picture in Figure 5 below).

Each node takes a decision based on the answers to various questions posed to the feature list. A question might take the form: Are there any of the following features in the lower right-hand part of the character A character, dropped in at the root of the tree, will thus follow a path to a leaf, at which point it will be recognized or rejected.

Figure 3:

The feature shapes have to be given names like E7, R1 (end in direction 7, rightangle in orientation 1). A small sample of the language is given below, the node names FRED, FRANK, JIM and JOE are used.

```
FIELD (1,24,1,16)
IF ANY (E3, E1, E2) THEN FRED
IF NONE (T7, X1, Y2, Y4) THEN JIM
RESULT 1
FRED: RESULT 3
JIM: IF NOT UP E7 THEN JOE ELSE FRANK
etc.
```

Figure 4:

I was taught how to manually key in a loader using switches on the front of the machine and then load my code via a paper tape. This was the first time I'd used a computer. At school we'd had some classes in programming from a visiting teacher (in Algol60 I think, but it may have been FORTRAN). We composed handwritten code which was then transferred to punched cards and run on a machine in some company. We got the resulting printouts back at the next lesson. I also used the interactive [FOCAL](#) system as part of practical classes associated (if I remember correctly) with an extremely boring university [course in numerical analysis](#) taught by [Maurice Wilkes](#).

My character recognition experiments were not that successful: perhaps 80% of the characters were recognised correctly. Character recognition is now a solved problem using neural nets (see [Chapter 1 of neuralnetworksanddeeplearning.com](#)). History has shown that the NPL method wasn't on the right track, though perhaps one can see aspects of the NPL ideas lurking in deep neural networks: for example, consider the diagram below in [Figure 6](#) below (taken from [Chapter 1](#) of the online textbook mentioned above):

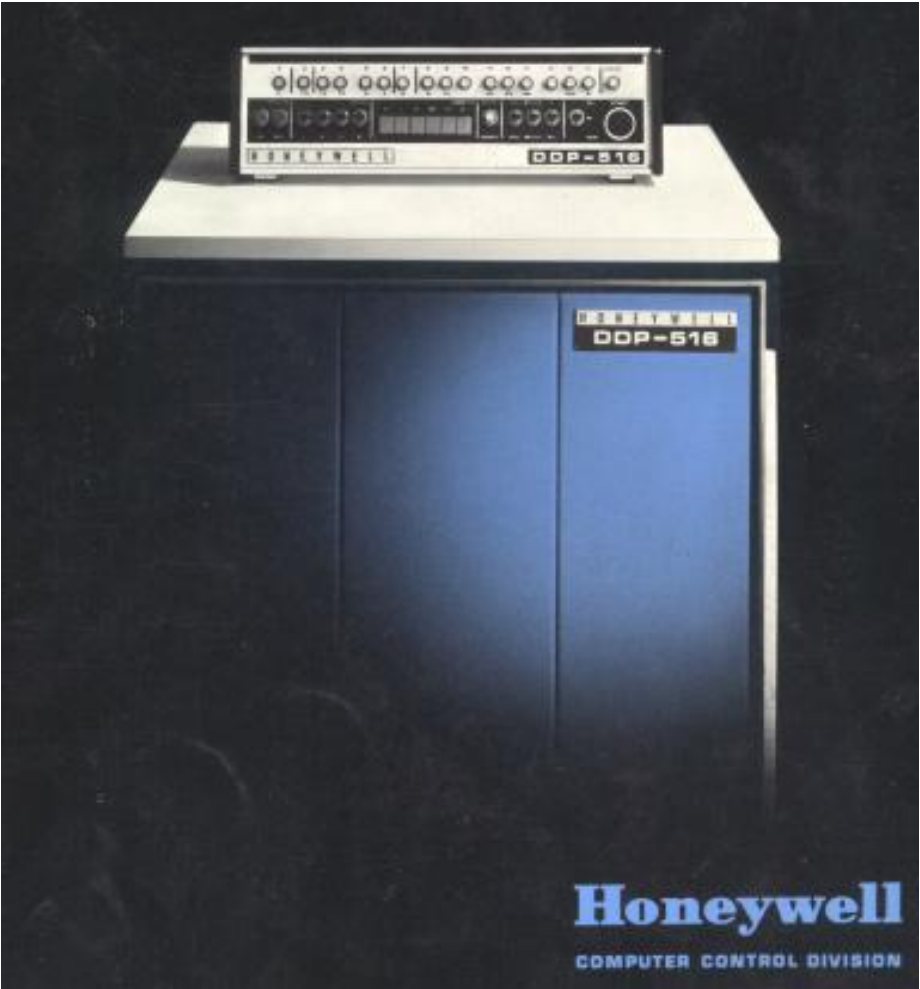


Figure 5:

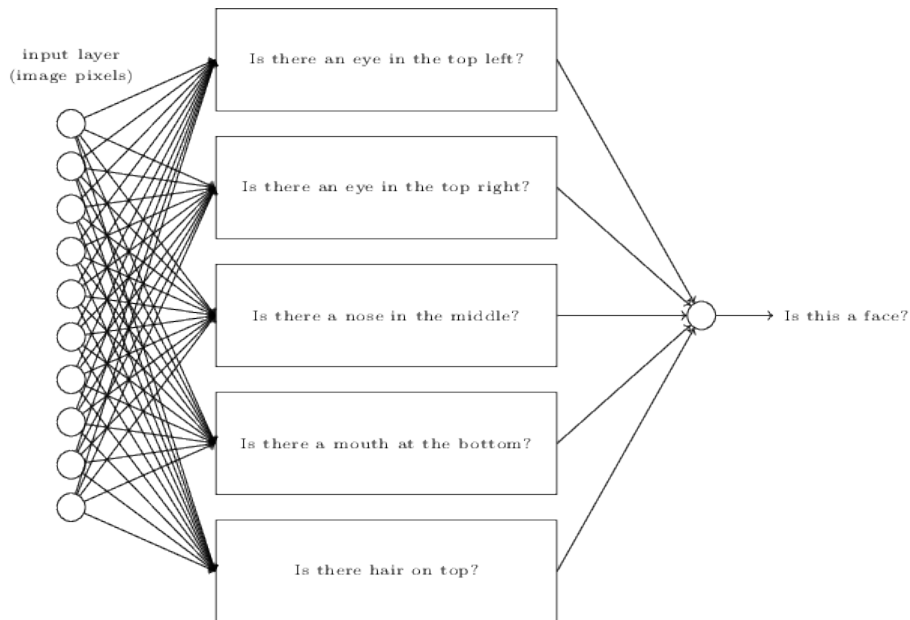


Figure 6:

Writing character recognition programs at the NPL was my first experience of using a computer for symbolic processing and it was the first time I enjoyed programming. My time at NPL got me interested in artificial intelligence, with the result that after I graduated I enrolled in the postgraduate Diploma in Machine Intelligence and Perception at Edinburgh and then stayed on in Edinburgh for my PhD (though I ended up working on programming language semantics, not AI).

Part II essay on perceptrons

During my third year at Cambridge there was an experimental scheme allowing Part II Mathematical Tripos students to submit an essay. Inspired by my time at NPL, I decided to submit an essay on [Perceptrons](#). I can't remember how I stumbled across perceptrons. My student room was in Harvey Court near to the University Library and I used to like wandering over there to peruse random journals (I still like to do this). I may well have focused on journals with pattern recognition papers because of my interest arising from my NPL project. This perusing may have lead me to perceptrons.

Incidentally, my habit of random perusing in the University Library also lead me to develop an interest in Chomsky's then new ideas on generative grammar.

After my PhD at Edinburgh I returned to Cambridge to study linguistics – but that’s a story for another article.

In [my Part II essay](#) I discussed some mathematical results about what perceptions could learn to recognise. I covered both positive and negative theorems: i.e. classes of inputs perceptions could learn to recognise and classes it couldn’t. The positive results came from the 1964 paper “Theoretical foundations of the potential function method in pattern recognition learning” by Aizerman, Mark A., Emmanuel M. Braverman, and Lev I. Rozonoér in *Automation and Remote Control* 25: 821-837. I don’t remember what lead me to this obscure journal of translations of Russian papers. Probably it was my random perusing in the University Library. The source of the negative results – theorems about classes of inputs perceptrons couldn’t learn to recognise – was the book [Perceptrons](#) by Minsky and Papert, which had then just been published.

Meeting David Marr

My essay [was evaluated](#) by [David Marr](#), who was then a Fellow at Trinity College and not yet famous. He invited me to a soiree-style party in his rooms, but I was totally intimidated and escaped as soon as I could. If I remember correctly, the Part II essays were put into three categories. I forget what the categories were, but mine was put into the category “highly commended”, which pleased me a lot, although all the other essays I knew about got this score too. My examination marks up to then had not been very good. I ended up with a better Part II result than I expected and I wondered whether my essay result influenced this, though officially it should not have.

Perceptrons were an early form of machine learning. Had I pursued my interest in this area perhaps my career would have taken a very different direction! The part of the Edinburgh School of Artificial Intelligence where I ended up for my graduate studies was the Department of Machine Intelligence and Perception. Another part of the school was the Theoretical Psychology Unit run by Christopher Longuet-Higgins. The neural networks superstar [Geoffrey Hinton](#) was a PhD student in this unit at about the same time, but I don’t recall ever meeting him then or since. Unfortunately, [personality clashes and bad academic politics](#) resulted in there being not much scientific collaboration between the different Edinburgh AI groups. I wonder how my life might have unfolded if I’d become involved with Marr’s coterie at Cambridge, got to know Hinton at Edinburgh and pursued research inspired by these people.