8. Optimising machine learners to learn on their own

to do

- hinton – backpropagation of error – the 1986 nature paper uses examples of people
- 'generalization error is what we care about' ("Lecture 9 | Machine Learning (Stanford)" 2008)
- deep neural nets (Hassabis et al. 2013) should appear here
- the Higgs Boson challenge
- MF in d&p writes about how the individual internalises the disciplinary mechanism – we are all machine learners now?
- the masculinity of machine learning – how to deal with that? some prominent women, but massively masculinist – takes me back to 1996 publication - also use SI of angelaki on geophilosophy of masculinity. See zotero masculinity folder
- the people change alongside the data; their sense of the power of data has a cost for them too
- put in Perlich stuff about data leakage – really important to focus on competition as a way of showing how people do things
- we finally reach people – why so late? And so what?
- use Lazzarato here – semiotics, etc
- MF from archaeology of knowledge on the subject in the discursive formation

Introduction

> If a proposition, a sentence, a group of signs can be called 'statement' , it is not therefore because, one day, someone happened to speak them or put them into some concrete form of writing; it is because the position of the subject can be assigned (Foucault 1972, 95)

Here are the words of Hilary Mason, Chief Scientist at bitly.com (a service that shortens URLs), at a London conference in 2012 called 'Bacon: Things Developers Love':

> You have all of these things that are different – engineering, infrastructure, mathematics and computer science, curiosity and an understanding of human behaviour – that is something that usually falls under the social sciences. At the intersection of all these things are wonderful people. But we're starting to develop something new, and that is - not that all of these things have not been done for a

> very long time - but we are only just now building systems where people, individual people, have all of these tools in one package, in one mind. And there are lots of reasons this is happening now. But its a pretty exciting time to be in any of these things. (Hilary Mason, Chief Scientist, bitly.com) (Bacon 2012)

In front of an audience of several hundred software developers, she describes shifts in the work of programming associated with the growth of large amounts of data associated with 'human behaviour.' At the centre of this shift stand 'wonderful people' who combine practices and knowledges of communication infrastructure, technology, statistics, and 'human behaviour' through curiosity and technical skills. Mason was, in effect, telling her audience of software developers who they should become and at the same time pointing to the some of the expansive changes occurring. The title of her talk was 'machine learning for hackers', and her audience were those hackers or programmers. A change in programming practice - 'machine learning' - is the key to programmers becoming the wonderful people, agents of their own time, capable of doing what is only now just possible because it is all together in 'one package, one mind.' Here, I would tentatively suggest, Mason adumbrates the outline of an agent of anticipation, someone at the intersection of network infrastructure, mathematics and human behaviour. Mason, one of *Fortune* magazines 'Top 40 under 40' business leaders to watch (**???**) also featured in *Glamour*, a teenage fashion magazine (**???**), personifies such a wonderful person.

'It is the privileged machine in this context that creates its marginalized human others' writes Lucy Suchman in her account of the encounters that 'effect "persons" and "machines" as distinct entities (Suchman 2006, 269). Suchman recommends 'recognition of the particularities of bodies and artifacts, of the cultural-historical practices through which human-machine differences are (re-)iteratively drawn, and of the possibilities for and politics of redistribution across the human machine boundary' (285). Could the 'wonderful people' that Mason describes also be marginalized human others? Does the re-distribution of engineering, mathematics, curiosity, infrastructure, or 'something that usually falls under the social sciences' (but perhaps no longer does so) both energise subjects ('its a pretty exciting time to be in any of these things') and assigns them a marginal albeit still pivotal position? This mixture of wonder, curiosity and something much more driven to harden distinctions and reduce the politics and potentials of re-distribution to much more heavily normalised form is the topic of this chapter. The key machine-diagram I draw on here are neural networks, or neural nets, in their various forms ranging from the perceptron, through the multilayer perceptron (MLP), the convolutional neural nets (CNN), recurrent neural nets (RNN) and deep belief networks or deep neural networks of many recent deep learning projects (particularly in machine learning competitions, as discussed below (Dahl 2013)).

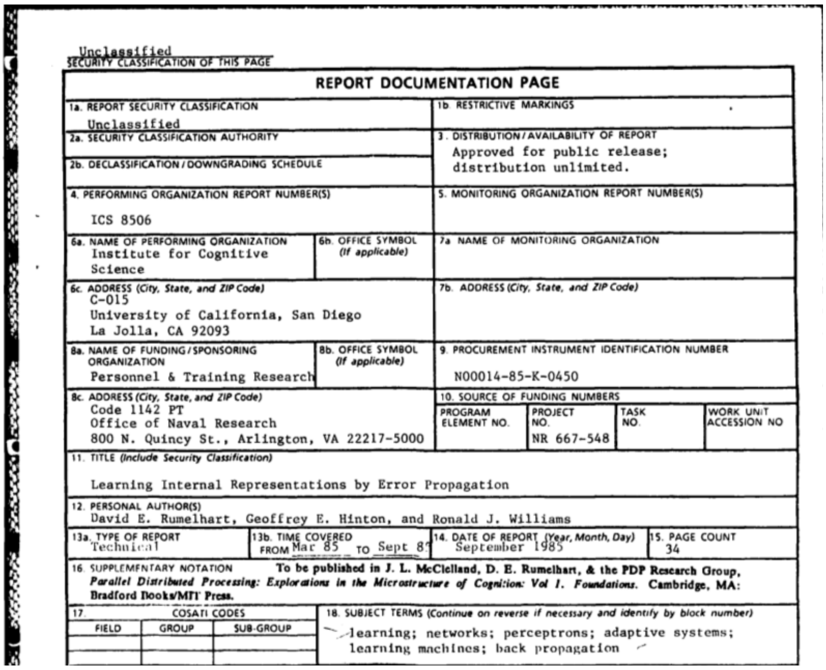
The uneven presence of neural nets in machine learning

In almost every machine learning class, neural nets make some appearance. They have an ambivalent status. In some ways, they renew long-standing hopes in biology, and particularly, brains and cognition as models of computational power, or, sometimes more pragmatically. In other ways, they gain traction as a ways of dealing with changing computational infrastructures, and the difficulties of capitalising on infrastructure that is powerful yet difficult to manage. Neural nets are often described from this deeply split perspective. On the one hand, they stem from a biological inspiration (dating at least back to the work on McCulloch and Pitts in the 1940s [REF TBA]). On the other hand, they respond to changes in information infrastructures and digital devices. For instance, David Ackley, Geoffrey Hinton (an important figure in the inception of neural nets during the 1980s and in the revival of neural net in the form of deep learning in the last decade), and Terrence Sejnowski wrote in the early 1980s:

> Evidence about the architecture of the brain and the potential of the new VLSI technology have led to a resurgence of interest in "connectionist" sys-terns … that store their long-term knowledge as the strengths of the connections between simple neuron-like processing elements. These networks are clearly suited to tasks like vision that can be performed efficiently in parallel networks which have physical connections in just the places where processes need to communicate. … The more difficult problem is to discover parallel organizations that do not require so much problem-dependent information to be built into the architecture of the network. Ideally, such a system would adapt a given structure of processors and communication paths to whatever problem it was faced with (Ackley, Hinton, and Sejnowski 1985, 147–48).

This convergence between brain and 'new VSLI [Very Large Scale Integrated] technology' – semiconductor chips – sought to implement the plasticity of neuronal networks in the parallel distributed processing enabled by very densely packed semiconductor circuits. The problem here was how to organize these connections without having to hardwire domain specificity into 'the architecture of the network.' How could the architectures adapt to the problem in hand?

We saw in **??** that the psychologist Frank Rosenblatt's perceptron (Rosenblatt 1958) first implemented the cybernetic vision of neurones as models of computation (Edwards 1996) . While the perceptron did not weather the criticism of artificial intelligence experts such as Marvin Minsky (Minsky famously showed that a perceptron cannot learn the logical exclusive OR or `XOR` function; (Minsky and Papert 1969)), cognitive psychologists such as David Rumelhart, Geoffrey Hinton and Ronald Williams returned to work with perceptrons, seeking to generalize their operations. In the mid-1980s, they developed the back propagation



Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified
1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY
3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE
4. PERFORMING ORGANIZATION REPORT NUMBER(S)
ICS 8506
5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION
Institute for Cognitive
Science
6b. OFFICE SYMBOL (If applicable)
7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code)
C-015
University of California, San Diego
La Jolla, CA 92093
7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION
Personnel & Training Research
8b. OFFICE SYMBOL (If applicable)
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
N00014-85-K-0450
8c. ADDRESS (City, State, and ZIP Code)
Code 1142 PT
Office of Naval Research
800 N. Quincy St., Arlington, VA 22217-5000
10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO
	NR 667-548		

11. TITLE (Include Security Classification)
Learning Internal Representations by Error Propagation
12. PERSONAL AUTHOR(S)
David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams
13a. TYPE OF REPORT
Technical
13b. TIME COVERED
FROM Mar 85 TO Sept 85
14. DATE OF REPORT (Year, Month, Day)
September 1985
15. PAGE COUNT
34
16. SUPPLEMENTARY NOTATION **To be published in J. L. McClelland, D. E. Rumelhart, & the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Vol 1. Foundations*. Cambridge, MA: Bradford Books/MIT Press.**
17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
learning; networks; perceptrons; adaptive systems;
learning machines; back propagation

Figure 1: An early publication of the back propagation algorithm: Rumelhart, Hinton and William's 1985 paper [@Rumelhart_1985]

algorithm (Rumelhart, Hinton, and Williams 1985; Hinton 1989), a way of adjusting the connections between nodes (neurones) in the network in response to features in the data (see Figure 1). The back propagation algorithm did begin to address the problem of discovering parallel network organizations without reliance on problem-specific architectures. Effectively, an architecture of generalization was implemented. While cognition, and the idea that machines would be cognitive (rather than say, mechanical, calculative, or even algorithmic) constantly organised research work in artificial intelligence for several decades, the development of the back propagation algorithm as a way for a set of connected computational nodes to learn also had strong infrastructural resonances. These resonances continue to echo today. Like the advent of VSLI in the early 1980s, the vast concentrations of computers in contemporary data centres (hundreds of thousands of cores as we saw in the case of Google Compute in the previous chapter **??**) pose the problem of organizing infrastructure so that processes can communicate with each other.

The back propagation algorithm will return below, but for the moment, this overlap between cognition and infrastructures, between people and machines, itself suggests another way of thinking about how 'long-term knowledge' takes shape today. At the same time as infrastructural reorganization takes place around learning, and around the production of statements by machine learners, both human and non-human machine learners are assigned new positions. These positions show a good deal of dispersion, hierarchy and distribution. The subject position in machine learning is a highly relational one, connected to transforms in infrastructure, variations in referentiality (such as we have seen in the construction of the common vector space), and competing forms of authority (as we have seen in the accumulations of different techniques).

Some machine learners attribute a more synaptic function to neural nets. The neural nets also synchronically spread into many difference disciplines: cognitive science, computer science, linguistics, adaptive control engineering, psychology, finance, operations research, etc, and particularly statistics and computer science. This synaptic function did not just extend machine learning. As Ethem Alpaydin writes:

> Perhaps the most important contribution of research on neural networks is this synergy that bridged various disciplines, especially statistics and engineering. It is thanks to this that the field of machine learning is now well established (Alpaydin 2010, 274).

Despite this synaptic function, neural networks have had a somewhat marginal position in relation to existing forms of computation. In early 2002, for instance, while carrying out an ethnographic study of 'extreme programming,' a software development methodology popular at that time (Mackenzie and Monk 2004), I spent several months visiting a company in Manchester developing software for call centres. The purpose of the software was to manage 'knowledge' in call centres such that any query from a caller would be readily answered by call centre staff who would query a knowledge management system. This system was advanced because it contained an artificial neural network that learned to match queries and responses over time. The taciturn neural network expert, Vlad, sat in a different part of the room from the developers working on the databases and the web interfaces of the knowledge management system. His work with the neural network was at the core of the knowledge management system yet outside the orbit of the software development team, who generally regarded Vlad and the neural net as an esoteric and temperamental scientific component. As we have already seen with `kittydar` , today the situation of neural networks has changed. It is no longer exotic or specialized, but in some ways spectacular.

The shifting fortunes of neural nets are discussed in different terms by machine learners themselves, but in recent years they share an awareness of some kind of transformation:

> Neural networks went out of fashion for a while in the 90s - 2005 because they are hard to train and other techniques like SVMs beat them on some problems. Now people have figured out better methods for training deep neural networks, requiring far fewer problem-specific tweaks. You can use the same pretraining whether you want

> a neural network to identify whose handwriting it is or if you want to decipher the handwriting, and the same pretraining methods work on very different problems. Neural networks are back in fashion and have been outperforming other methods, and not just in contests (**???**).

Neural nets also receive uneven attention in the machine learning literature. In Andrew Ng's Stanford CS229 lectures from 2007, they receive somewhat short shrift: around 30 minutes of discussion in Lecture 6, in between Naive Bayes classifiers and several weeks of lectures on support vector machines ("Lecture 6 | Machine Learning (Stanford)" 2008). As he introduces a video of an autonomous vehicle steered by a neural net after a 20 minute training session with a human driver, Ng comments that 'neural nets were the best for many years.' The lectures quickly moves on to the successor, support vector machines. In *Elements of Statistical Learning*, a whole chapter appears on the topic, but prefaced by a discussion of the antecedent statistical method of 'projection pursuit regression.' The inception of 'projection pursuit' is dated to 1974, and thus precedes the 1980s work on neural nets that was to receive so much attention. In *An Introduction to Statistical Learning with Applications in R*, a book whose authors include Hastie and Tibshirani, neural nets are not discussed and indeed not mentioned (James et al. 2013). Textbooks written by computer scientists such as Ethem Alpaydin's *Introduction to Machine Learning* do usually include at least a chapter on them, sometimes under different titles such as 'multi-layer perceptrons' (Alpaydin 2010). Willi Richert and Luis Pedro Coelho's *Building Machine Learning Systems with Python* likewise does not mention them (Richert and Coelho 2013). Cathy O'Neil and Rachel Schutt's *Doing Data Science* mentions them but does not discuss them (Schutt and O'Neil 2013), whereas both Brett Lantz's *Machine Learning with R* (Lantz 2013) and Matthew Kirk's *Thoughtful Machine Learning* (Kirk 2014) devote chapters to them. In the broader cannon of machine learning texts, the computer scientist Christopher Bishop's heavily cited books on pattern recognition dwell extensively on neural nets (Bishop and others 1995; Bishop and Nasrabadi 2006). Amongst statisticians, Brian Ripley's *Pattern Recognition and Neural Networks* (Ripley 1996), also highly cited, placed a great deal of emphasis on them. But these specific documents against a pointillistic background of hundreds of thousands of scientific publications mentioning or making use of neural nets since the late 1980s in the usual litany of fields – atmospheric sciences, biosensors, botany, power systems, water resource management, internal medicine, etc. This swollen publication tide attests to some kind of formation or configuration of knowledge invested in these particular techniques, perhaps more so than other I have discussed so far (logistic regression, support vector machine, decision trees, random forests, linear discriminant analysis, etc.).

The somewhat vacillating presence of neural nets in the machine learning literature itself finds parallels in the fortunes of individual machine learners. Yann LeCun's work on optical character recognition during 1980-1990s is said to have discovered the back propagation algorithm at the same time as Rumelhart, Hinton and Williams (???). His implementations in `LeNet` led many academic machine learning competitions during the 1990s. In 2007, Andrew Ng could casually observe that neural nets *were* the best, but in 2014, LeCun find himself working on machine learning at Facebook. Similarly, the cognitive psychologist Geoffrey Hinton's involvement in the early 1980s work on connectionist learning procedures in neural nets and subsequently on 'deep learning nets' (Hinton and Salakhutdinov 2006) delivers him to Google in 2013. These trajectories between academic research and industry are not unusual. Many of the techniques in machine learning have been incorporated into companies later acquired by other larger companies. Even if there is no spin-off company to be acquired, machine learners themselves have been assigned key positions in many industry settings. Corinna Cortes, co-inventor with Vladimir Vapnik of the support vector machine, heads research at Google New York. Ng himself in 2014 began work as chief scientist for the Chinese search engine, Baidu leading a team of AI researchers specializing in 'deep learning,' the contemporary incarnation of neural nets (Hof 2014). In 2011, Ng led a neural net-based project at Google that had, among other things, detected cats in millions of hours of Youtube videos.[1] In recent years, (2012-2015), work on neural nets has again intensified, most prominently in association with social media platforms, but also in the increasingly common speech and face recognition systems found in everyday services and devices. Many of these neural nets are like `kittydar`, but implemented on a much larger and more distributed scale (for instance, in classifying videos on Youtube).

What accounts for the somewhat uneven fortunes of the neural net amongst machine learners? The unevenness of their performance, from limited curiosity in the late 1960s to best performer in the machine learning image classification competitions of the 1990s, from second best competitor in late 1990s to the spectacular promise of deep belief networks in 2012, suggests that some powerful dynamics or becomings are in play around them. These dynamics are not easily understood in terms of celebrity machine learners (human and non-human) suddenly rising to prominent or privileged positions in the research departments of social media platforms.[2] Nor does it make sense to attribute the rising fortunes of the neural net to the algorithms themselves, as if some decisive advance occurred in algorithms. The algorithms used in neural net have not, as we will see, been radically transformed in their core operations since the 1980s, and even then, the algorithms themselves (principally gradient descent) were not

[1] Unlike the cats detected by `kittydar,` the software discussed in the introduction to this book, the Google experiment did not use supervised learning. The deep learning approach was unsupervised (Markoff 2012). That is, the neural nets were not given images in which cats were labelled to train on.

[2] In any case, social media and search engines cannot be understood apart from the machine learning techniques that have been thoroughly woven through them since their inception. Hence *Elements of Statistical Learning* devotes several pages Google's famous *PageRank* algorithm, describing it as an unsupervised learner (Hastie, Tibshirani, and Friedman 2009, 576–78).

new. There have been important changes in scale (similar to those described in the previous chapter in the case of the `RF-ACE` algorithm and Google Compute), but as is often the case in machine learning, their re-invention occurs through proliferation, changes in scale, re-distributions of knowledge and infrastructure and specific optimisations. While machine learners in their machine form can be assigned a privileged position in the transformations of knowledge and action today, human machine learners are not exactly marginalized, at least in high profile cases such as Ng, LeCun, Hinton and others. Rather, the scale of machine learning seems to be changing both for the algorithms and for the human computer scientists, programmers and engineers.

A privileged machine and its human others

What accounts for this acceleration and slowing-down, the intensified interactions and abandonments of neural nets over the last three decades? Despite their apparent differences in origin, neural net share much with other machine learners. The language of brain, neurones and cognition associated with neural net covers over their much more familiar vector-space, function-finding and optimisations they draw on in practice. 'The central idea,' write Hastie and co-authors, 'is to extract linear combinations of the inputs as derived features, and then model the target as a nonlinear function of these features. The result is a powerful learning method, with widespread applications in many fields' (Hastie, Tibshirani, and Friedman 2009, 389). Much of the operation of the neural net is already familiar to us. The neural networks traverse data in a vector space. In the code vignette shown below, the data is a spreadsheet of information about passengers of the Titanic. The `titanic` dataset, like `iris` or `boston` is often used in contemporary machine learning pedagogy. It is for instance, the main training dataset used by (kaggle.com)[http://kaggle.com], an online machine learning competition site I will discuss below. The first few lines of the `R` code load the dataset and transform it into vector space. For instance, variables such as `sex` that take values such as `male` and `female` become vectors of `1` and `0` in a new variable `sexmale`.

```
{r neural_net1, echo=TRUE, message=FALSE, warning=FALSE,
cache=TRUE, fig.cap=''} library(neuralnet) titanic = read.csv('data/titanic3.csv')
titanic_transformed = as.data.frame(model.matrix(~survived +
age+ pclass + fare+ sibsp + sex + parch + embarked, titanic))
head(titanic_transformed) train_index = sample.int(nrow(titanic)/2)
titanic_train = titanic_transformed[train_index,] titanic_net =
neuralnet(survived ~ age +pclass + fare + sexmale + sibsp +
parch + embarkedC + embarkedQ + embarkedS, data=titanic_train,
err.fct='ce', linear.output=FALSE, hidden=5) titanic_test =
titanic_transformed[-train_index,] test_error = round(sum( 0.5 <
compute(titanic_net, titanic_test[,-c(1,2)])$net.result)/sum(titanic_test$survived),
2)
```

The line of the code that constructs a neural net using the `neuralnet` library (**???**), and the description of the classifier here is a familiar one. Despite its biological figuration, the neural net models whether someone `survived` the wreck of the Titanic in terms of their age, class of fare (`pclass`), sex, number of siblings/spouse (`sibsp`), number of parents/children (`parch`) and port of departure:

```
survived ~ age +pclass + fare + sexmale + sibsp + parch + embarkedC
+ embarkedQ + embarkedS
```

As is normally the case in `R` model formula, the response or target variable `survived` is expressed as a combination of other variables. In this case, the plus sign `+` indicates that the combination is linear or additive. As Hastie puts, 'the central idea is to extract linear combinations of the inputs' or predictor variables. If this model formula looks so similar to other machine learning techniques we have been discussing, what do neural networks add? Why did and do so many people turn to them?

In the code vignette above, the expression `hidden = 5` points to the distinctive features of these techniques, features that do not appear in the model formula. These 'hidden units' are key to neural networks. As Rumelhart, Winton and Williams announce the algorithm in a latter to *Nature* in 1986:

> We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure (**???**).

Again, the common reference to biology, but the description of the 'new learning procedure' starts to sound more like machine learning. There is talk of minimizing a measure of difference between actual and desire output vectors, as well as mention of 'features' and 'weights' (usually a synonym for model parameters). The novelty consists in the 'hidden' units whose interactions 'represent important features.' In other words, the flat additive combination of features expressed in the model formula above does not convey the interactions of these units. Yet these units can only viably interact in the neural nets because the back-propagation algorithm offers a way to create 'useful, new features' from the data.

`{r plot_neural_net, echo=FALSE, cache=TRUE, fig.cap=''} source('plot_nn.r')`

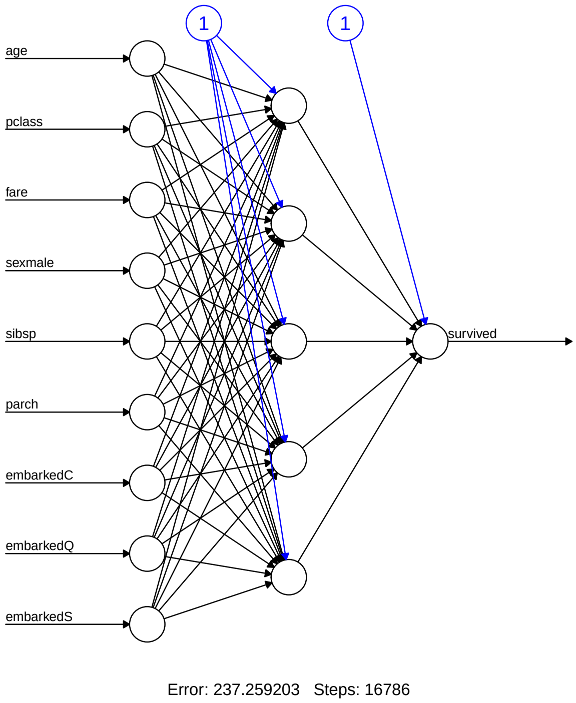


Figure 2: Neural network topology for ‘titanic‘ data

```
pdf(file = 'figure/titanic_net.pdf') print(plot.nn(titanic_net,
fontsize=8, show.weights=FALSE)) dev.off()
```

The big picture is that given data, a real-world classification problem, and constraints, you need to determine:

```
>1. Which classifier to use
>2. Which optimization method to employ
>3. Which loss function to minimize
>4. Which features to take from the data
>5. Which evaluation metric to use
```

```
[@Schutt_2013, 116]
```

People at the intersection or in a hierarchy

How would we describe the figure of human machine learner in this setting? In *The Archaeology of Knowledge*, Michel Foucault refers to the 'position of the subject' as an anchor point for the power-laden, epistemically conditioning enunciative functions called 'statements.' When a subject position can be assigned, propositions, diagrams, numbers, models, calculations and data structures can come together in statements, as 'the specific forms of an accumulation' (Foucault 1972, 125). But this anchor point is not a unifying point grounded in interiority, in intentionality or even in single speaking position or voice (that of the machine learning expert, for instance). On the contrary, 'various enunciative modalities manifest his [sic] dispersion' (Foucault 1972, 54). In the mist of this dispersion (a dispersion that is the main focus of this chapter), the position of subject is linked to operations that determine statements that become a kind of law for the subject. As Foucault puts it, in a formulation that effectively anticipates accounts of performativity that gain currency elsewhere several decades later,

> in each case the position of the subject is linked to the existence of an operation that is both determined and present; in each case, the subject of the statement is also the subject of the operation (he who establishes the definition of a straight line is also he who states it; he who posits the existence of a finite series is also, and at the same time, he who states it) ; and in each case, the subject links, by means of this operation and the statement in which it is embodied , his future statements and operations (as an enunciating subject, he accepts this statement as his own law) (Foucault 1972, 94–95).

It is indeed fitting that Foucault's examples here include subjects who say things like 'I call straight any series of points that …,' since these are just the kinds of sentences that operate in machine learning. The *operation*, however, is crucial, since the operation opens onto many different practices and techniques (function finding, optimisation, transformation of data into the common vector space, mobilisation of probability distributions as a kind of rule of existence for learnable situations, etc.) that accompany, ornament, armour and diagram the statement.

– cf Ng, Hassabis, Hinton, Le Cun, etc. with others

Competition and the work of optimisation

Programming as backpropagation

> It is as if the error propagates from the output y back to the inputs and hence the name *backpropagation* was coined (Alpaydin 2010, 250).

$$z_h = \frac{1}{1 + exp[-(\sum_{j=1}^{d} w_{hj} x_j + w_{h0})]}, h = 1..., H \quad (1)$$

The privilege of machine learning

Conclusion

References

Ackley, David H., Geoffrey E. Hinton, and Terrence J. Sejnowski. 1985. "A learning algorithm for Boltzmann machines." *Cognitive science* 9 (1): 147–69. http://www.sciencedirect.com/science/article/pii/S0364021385800124.

Alpaydin, E. 2010. *Introduction to machine learning*. Cambridge, Massachusetts; London: The MIT Press.

Bacon. 2012. "Hilary Mason - Machine Learning for Hackers." http://vimeo.com/43547079.

Bishop, Christopher M., and Nasser M. Nasrabadi. 2006. *Pattern recognition and machine learning*. Vol. 1. springer New York. http://www.library.wisc.edu/selectedtocs/bg0137.pdf.

Bishop, Christopher M., and others. 1995. "Neural Networks for Pattern Recognition." http://www.engineering.upm.ro/master-ie/sacpi/mat_did/info068/docum/Neural/%20Networks/%20for/%20Pattern/%20Recognition.pdf.

Dahl, George. 2013. "Deep Learning How I Did It: Merck 1st place interview. no free hunch." http://blog.kaggle.com/2012/11/01/deep-learning-how-i-did-it-merck-1st-place-interview/

Edwards, Paul N. 1996. *The Closed World : Computers and the Politics of Discourse in Cold War*. Inside Technology. Cambridge, Mass. ; London: MIT Press.

Foucault, Michel. 1972. "The Archaeology of Knowledge and the Discourse on Language (Trans: A. Sheridan)." *Pantheon, New York*.

Hassabis, Demis, R. Nathan Spreng, Andrei A. Rusu, Clifford A. Robbins, Raymond A. Mar, and Daniel L. Schacter. 2013. "Imagine All the People: How the Brain Creates and Uses Personality Models to Predict Behavior." *Cerebral Cortex*, bht042. http://cercor.oxfordjournals.org/content/early/2013/03/04/cercor.bht042.short.

Hastie, Trevor, Robert Tibshirani, and Jerome H. Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer.

Hinton, Geoffrey E. 1989. "Connectionist Learning Procedures." *Artificial Intelligence* 40 (1): 185–234. http://www.sciencedirect.com.ezproxy.lancs.ac.uk/science/article/pii/0004370289900490.

Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. 2006. "Reducing the dimensionality of data with neural networks." *Science* 313 (5786): 504–7. http://www.sciencemag.org/content/313/5786/504.short.

Hof, Robert D. 2014. "Chinese Search Giant Baidu Thinks AI Pioneer Andrew Ng Can Help It Challenge Google and Become a Global Power. MIT Technology Review." http://www.technologyreview.com/featuredstory/530016/a-chinese-internet-giant-starts-to-dream/.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Springer. http://link.springer.com/content/pdf/10.1007/978-1-4614-7138-7.pdf.

Kirk, Matthew. 2014. *Thoughtful Machine Learning: A Test-Driven Approach*. 1 edition. Sebastopol, Calif.: O'Reilly Media.

Lantz, Brett. 2013. *Machine Learning with R*. Birmingham: Packt Publishing.

"Lecture 6 | Machine Learning (Stanford)." 2008. http://www.youtube.com/watch?v=qyyJKd-zXRE/&feature=youtube_gdata_player.

"Lecture 9 | Machine Learning (Stanford)." 2008. https://www.youtube.com/watch?v=tojaGtMPo5U/&feature=youtube_gdata_player.

Mackenzie, Adrian, and Simon Monk. 2004. "From Cards to Code: How Extreme Programming Re-Embodies Programming as a Collective Practice." *Computer Supported Cooperative Work (CSCW)* 13 (1): 91–117.

Markoff, John. 2012. "In a Big Network of Computers, Evidence of Machine Learning." *The New York Times*, June. http://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html.

Minsky, Marvin, and Seymour Papert. 1969. "Perceptron: an introduction to computational geometry." *The MIT Press, Cambridge, expanded edition* 19: 88.

Richert, Willi, and Luis Pedro Coelho. 2013. *Building Machine Learning Systems with Python*. Birmingham: Packt Publishing.

Ripley, Brian. 1996. *Pattern Recognition and Neural Networks. 1996*. Cambridge ; New York: Cambridge University Press.

Rosenblatt, F. 1958. "The perceptron: A probabilistic model for information storage and organization in the brain." *Psychological Review* 65 (6): 386–408. doi:10.1037/h0042519.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. 1985. *Learning Internal Representations by Error Propagation*. DTIC Document. http://oai.dtic.mil/oai/oai?verb=getRecord/&metadataPrefix=html/&identifier=ADA164453.

Schutt, Rachel, and Cathy O'Neil. 2013. *Doing Data Science.* Sebastopol, Calif.: O'Reilly & Associates Inc.

Suchman, Lucy. 2006. *Human and Machine Reconfigurations: Plans and Situated Actions.* 2nd ed. Cambridge University Press.