



Improving Business Decision Making with Bayesian Artificial Intelligence

Dr. Michael Green

2017-10-03



GOTO Copenhagen 2017
Conference Oct. 1-3, 2017

Improving Business Decision Making with Bayesian Artificial Intelligence

Michael Green

 Follow us @gotocph



Agenda

- Overview of AI and Machine learning
- Why do we need more?
- Our Bayesian Brains
- Probabilistic programming
- Tying it all together



Overview of AI and Machine learning

“ *AI is the behaviour
shown by an agent in an
environment that seems
to optimize the concept
of future freedom* ”



What is Artificial Intelligence?

Artificial Narrow Intelligence

- Classifying disease
- Self driving cars
- Playing Go

Artificial General Intelligence

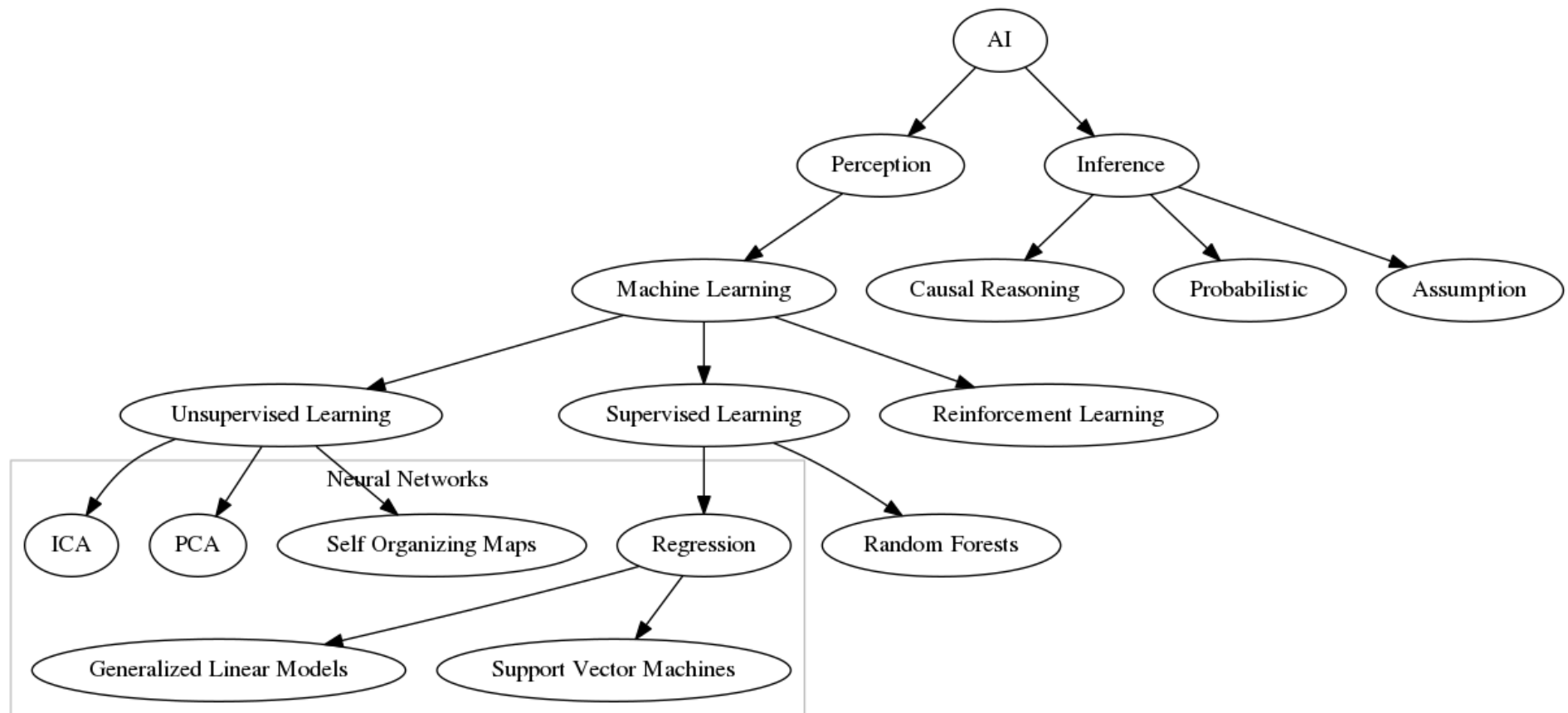
- Using the knowledge of driving a car and applying it to another domain specific task
- In general transcending domains

Artificial Super Intelligence

- Scaling intelligence and moving beyond human capabilities in all fields
- Far away?



The AI algorithmic landscape



Why do we need more?

Machine learning can only take us so far

Why is that?

- **Data:** Data is not available in cardinality needed for many real world interesting applications
- **Structure:** Problem structure is hard to detect without domain knowledge
- **Identifiability:** For any given data set there are many possible models that fit really well to it with fundamentally different interpretations
- **Priors:** The ability to add prior knowledge about a problem is crucial as it is the only way to do science
- **Uncertainty:** Machine learning application based on maximum likelihood cannot express uncertainty about it's model



The Bayesian brain

Domain space

$$p(x, y, \theta)$$

Machine learning

$$p(y|\theta, x)$$

Inference

$$p(\theta|y, x) = \frac{p(y|\theta, x) p(\theta|x)}{\int p(y, \theta|x) d\theta}$$

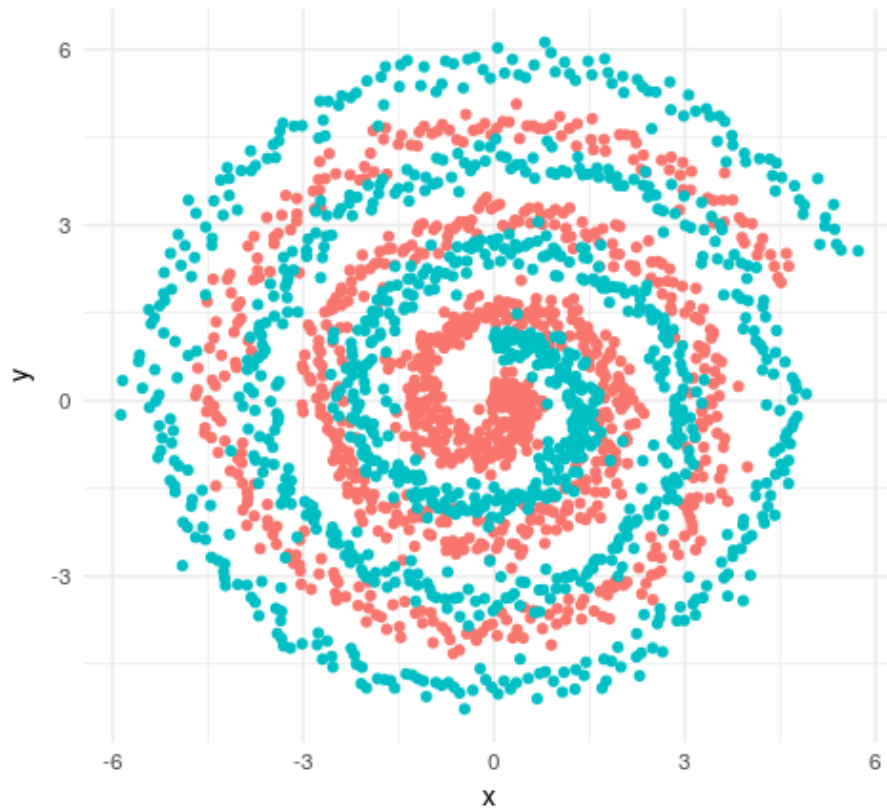


“*You cannot do science
without assumption!*”



A Neural Networks example

Spiral data



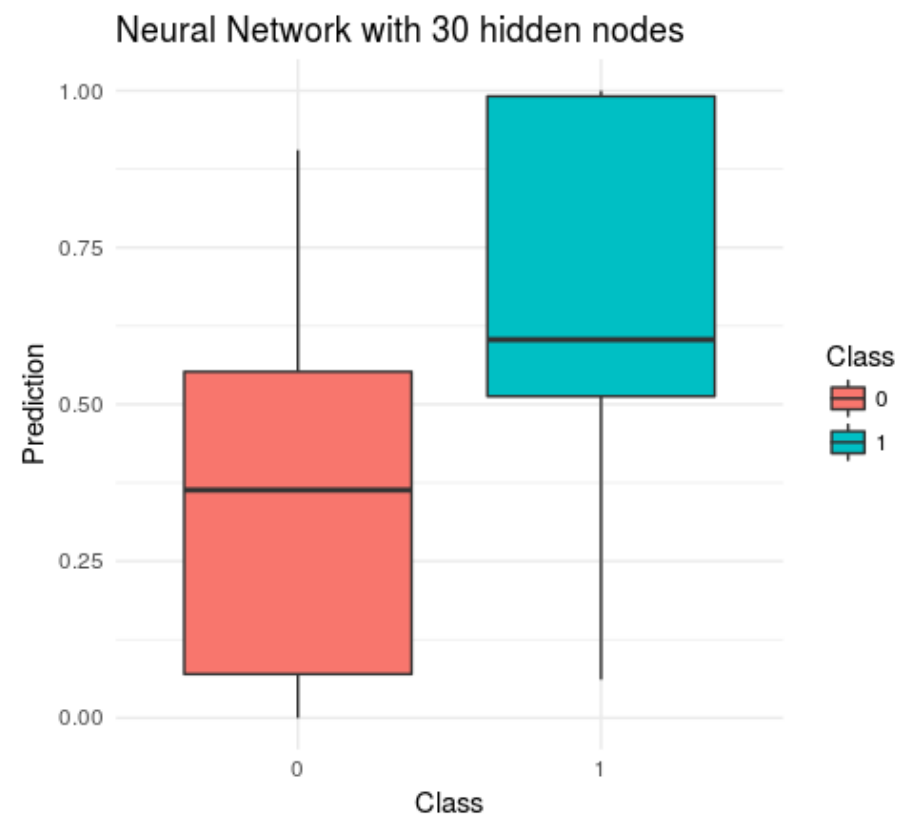
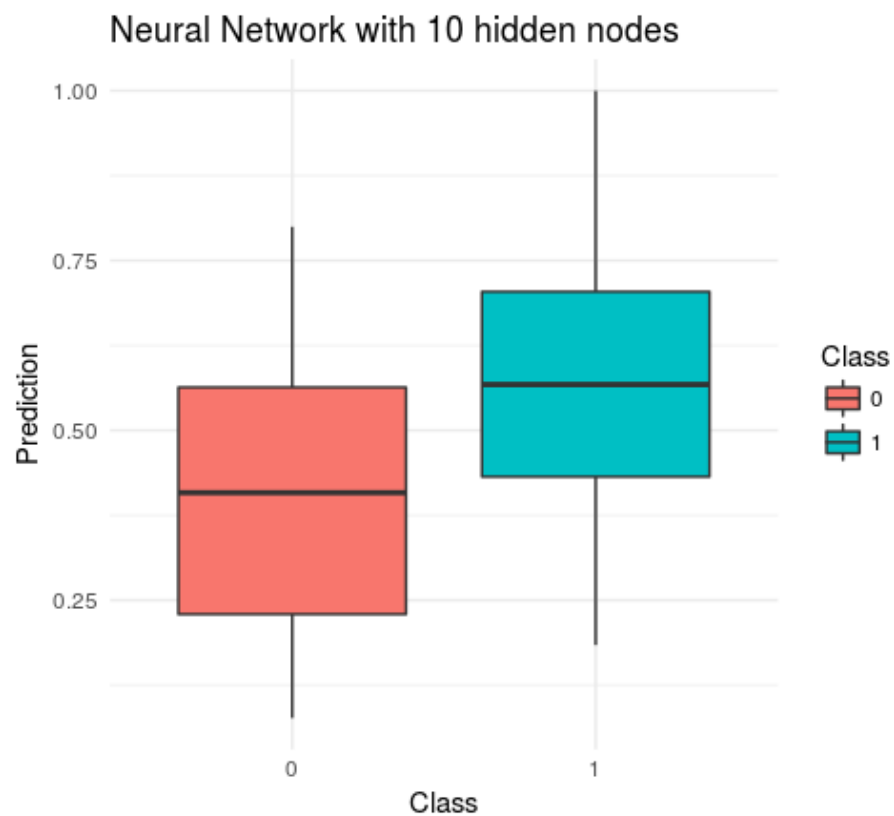
Overview

This spiral data feature two classes and the task is to correctly classify future data points

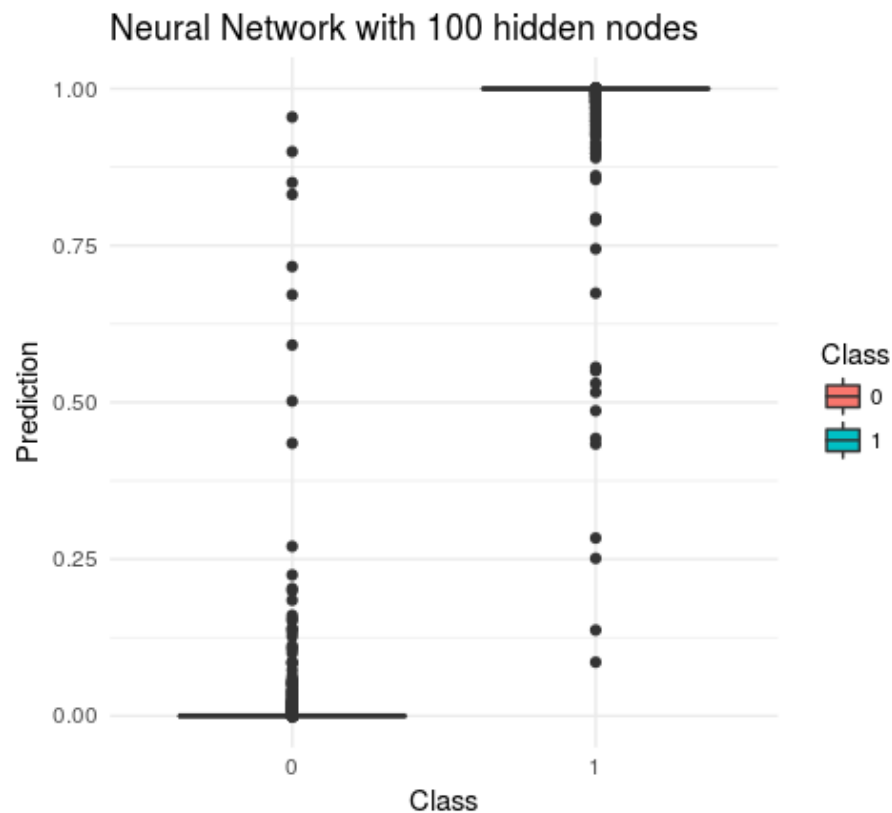
Features of this data



Running a Neural Network



Running a Neural Network



Accuracy

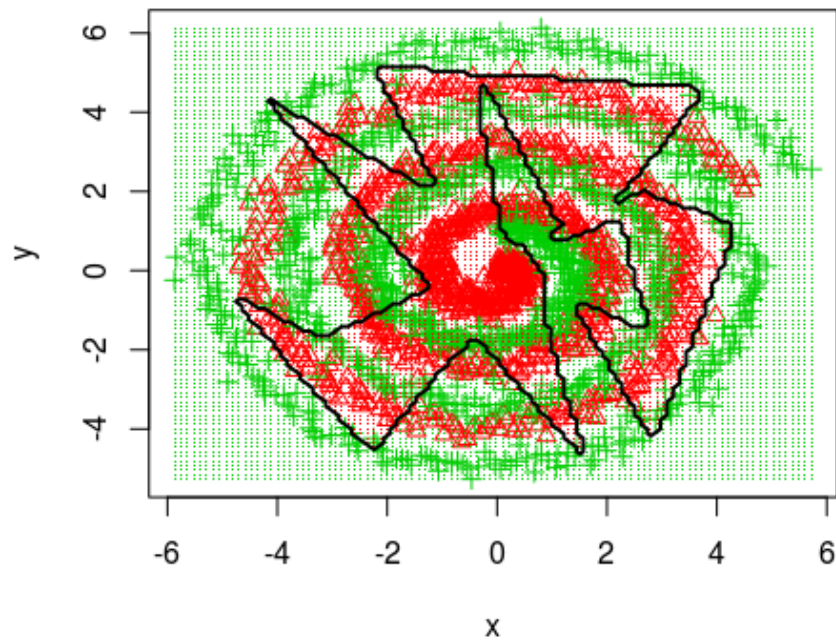
Hidden nodes	Accuracy	AUC
10	65%	74%
30	71%	82%
100	99%	100%

Only at 100 latent variables in the hidden layer do we reach the accuracy we want

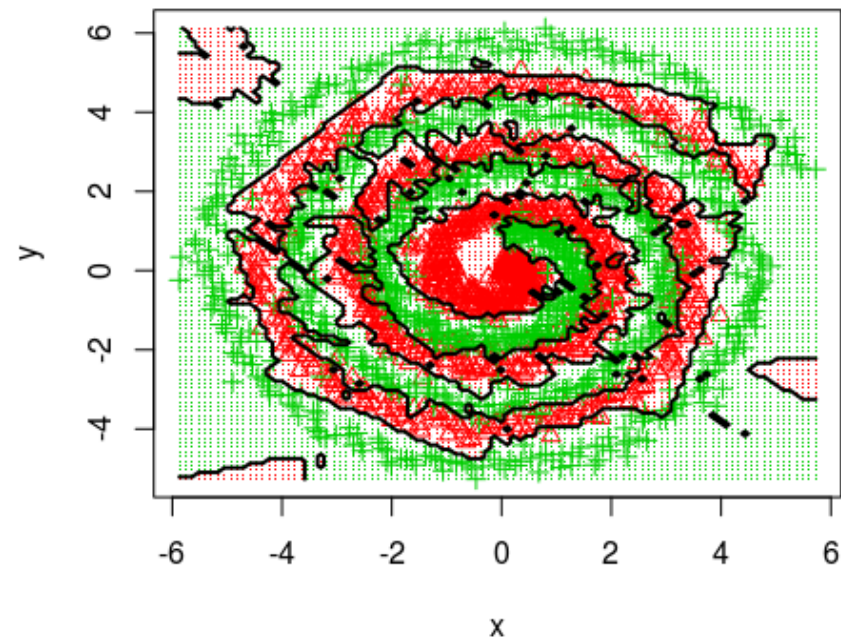


Decision boundaries

10 hidden nodes

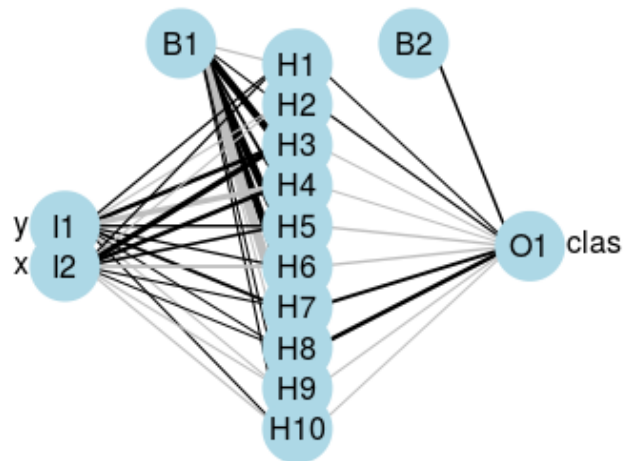


100 hidden nodes

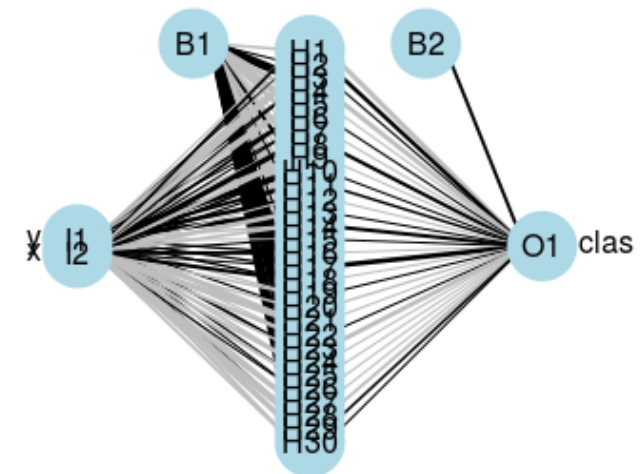


Network architectures

10 Hidden nodes

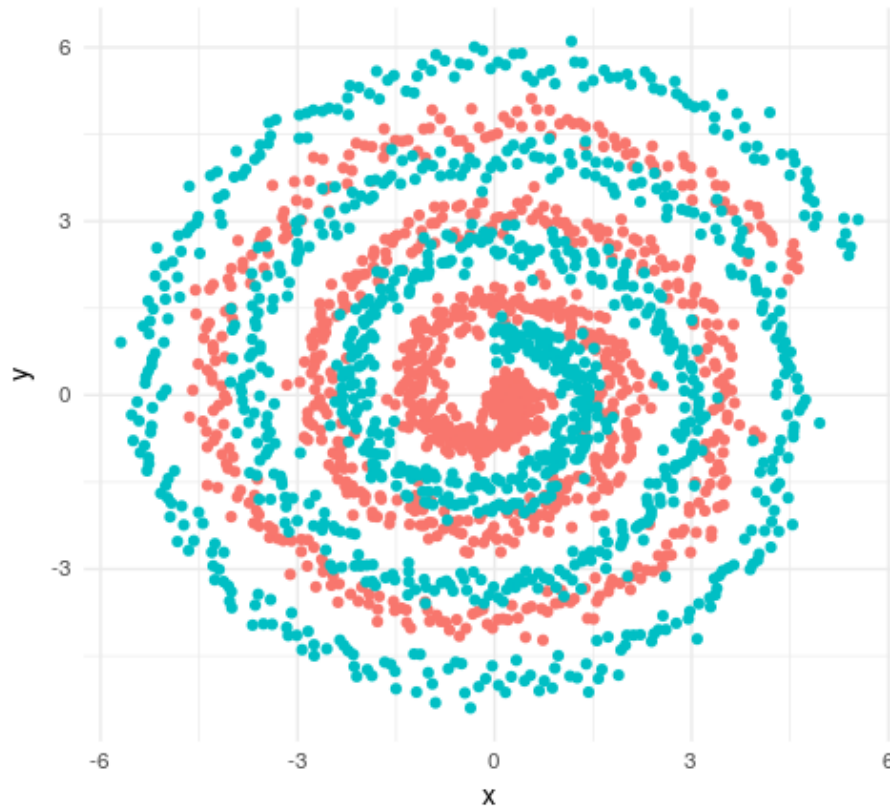


30 Hidden nodes

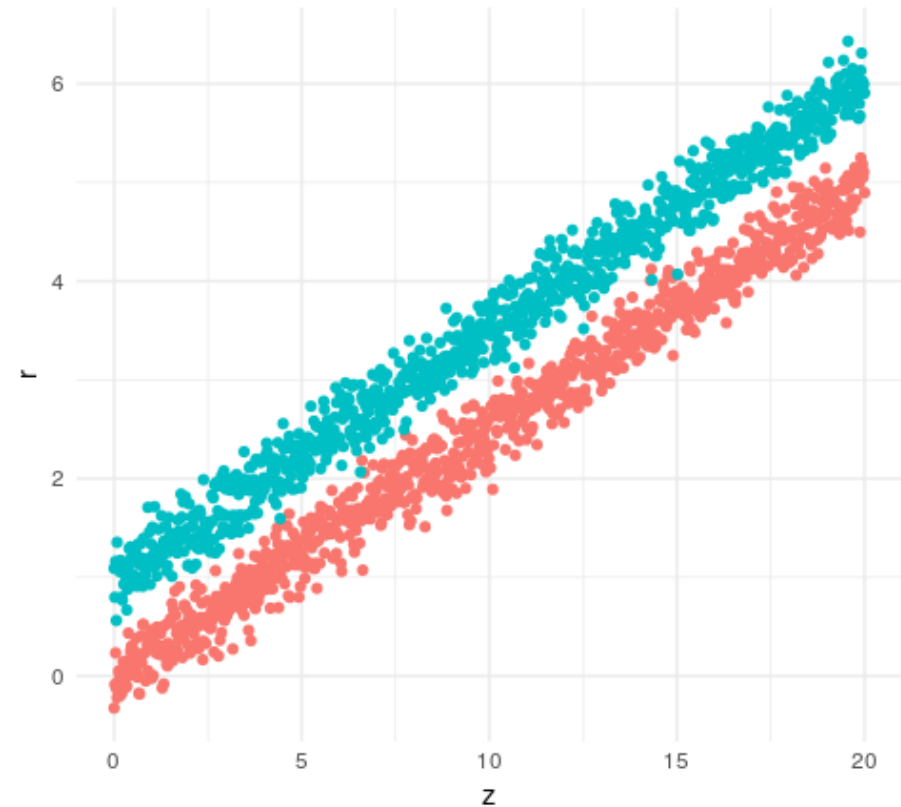


Proper modeling of the problem

Cartesian coordinates



Polar coordinates



A probabilistic programming take

“*Probabilistic programming is an attempt to unify general purpose programming with probabilistic modeling*”



Learning the data

$$x \sim \mathcal{N}(\mu_x, \sigma_x)$$

$$y \sim \mathcal{N}(\mu_y, \sigma_y)$$

$$\mu_x = (r + \delta) \cos\left(\frac{t}{2\pi}\right)$$

$$\mu_y = (r + \delta) \sin\left(\frac{t}{2\pi}\right)$$

$$\delta \sim \mathcal{N}(0.5, 0.1)$$

- Instead of throwing a lot of nonlinear generic functions at this beast we could do something different

- From just looking at the data we can see that the generating functions must look like
- Which fortunately can be



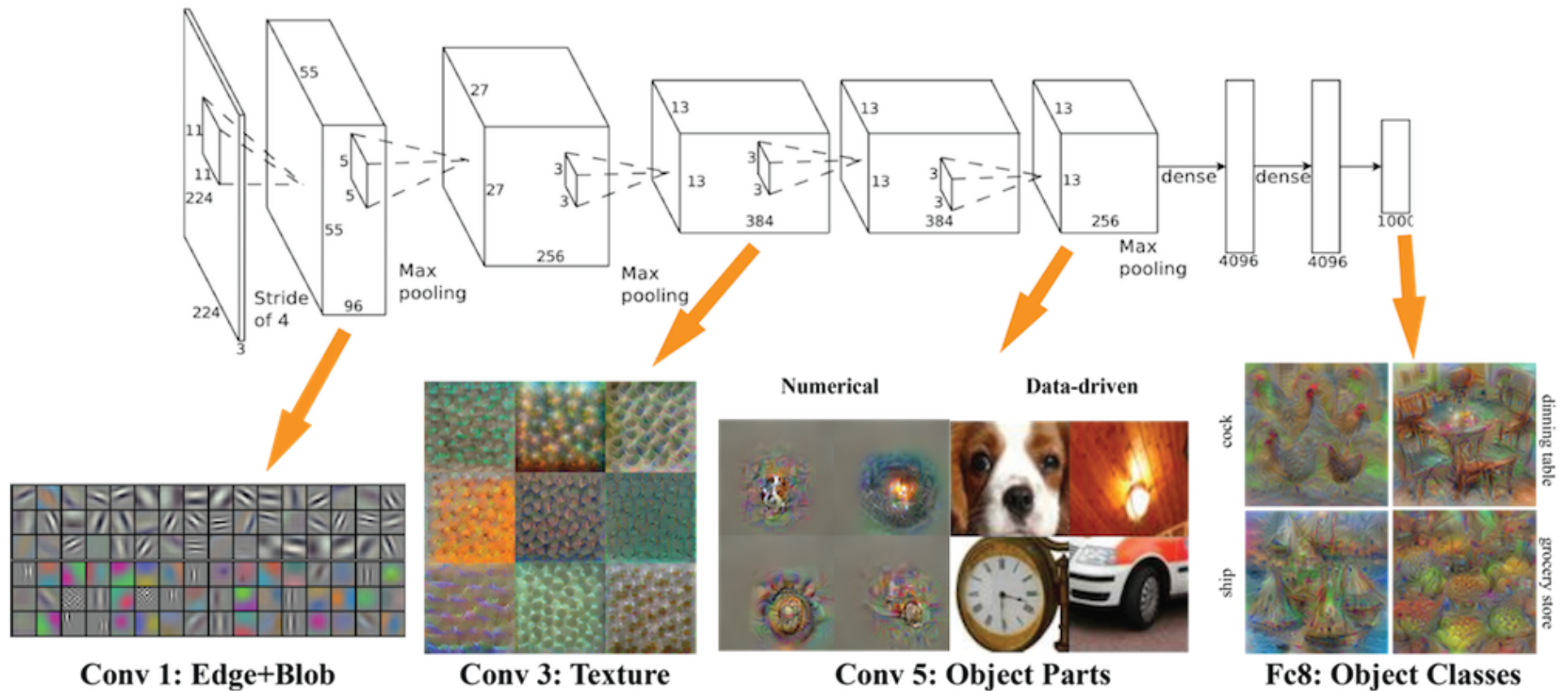
What we gain from this

- We get to put our knowledge into the model solving for mathematical **structure**
- A generative model can be realized
- Direct measures of **uncertainty** comes out of the model
- No crazy statistical only results due to **identifiability** problems

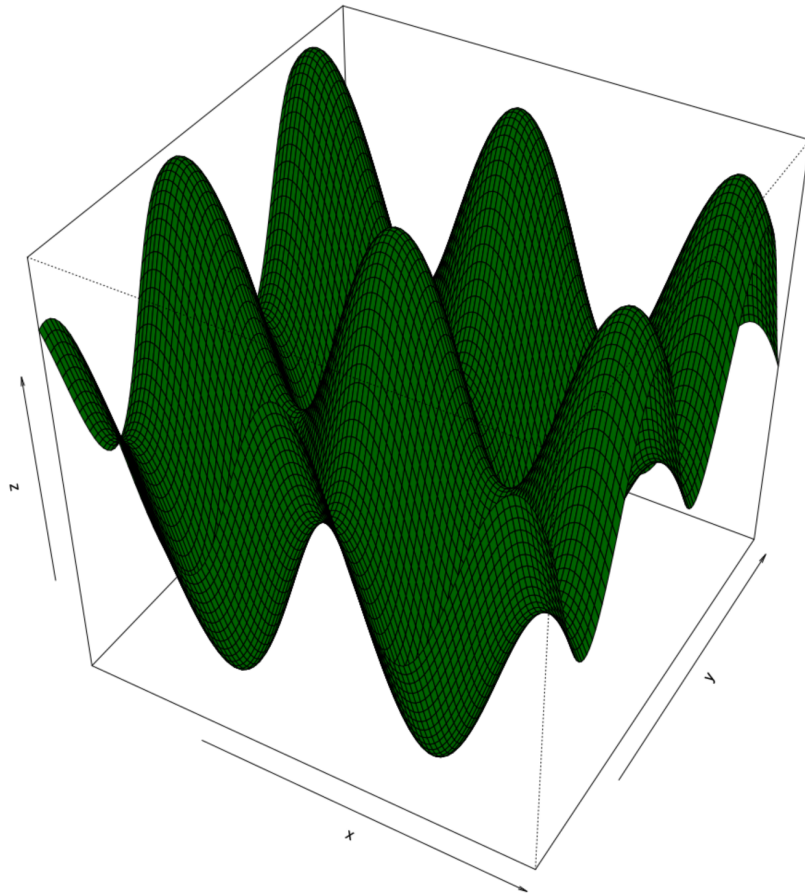


Deep Learning

Deep learning is just a stacked neural network



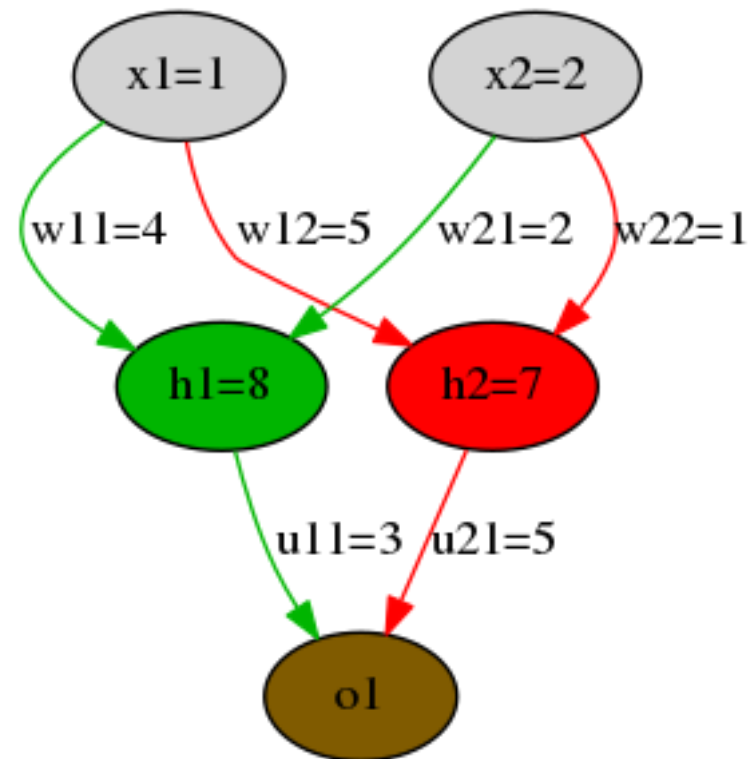
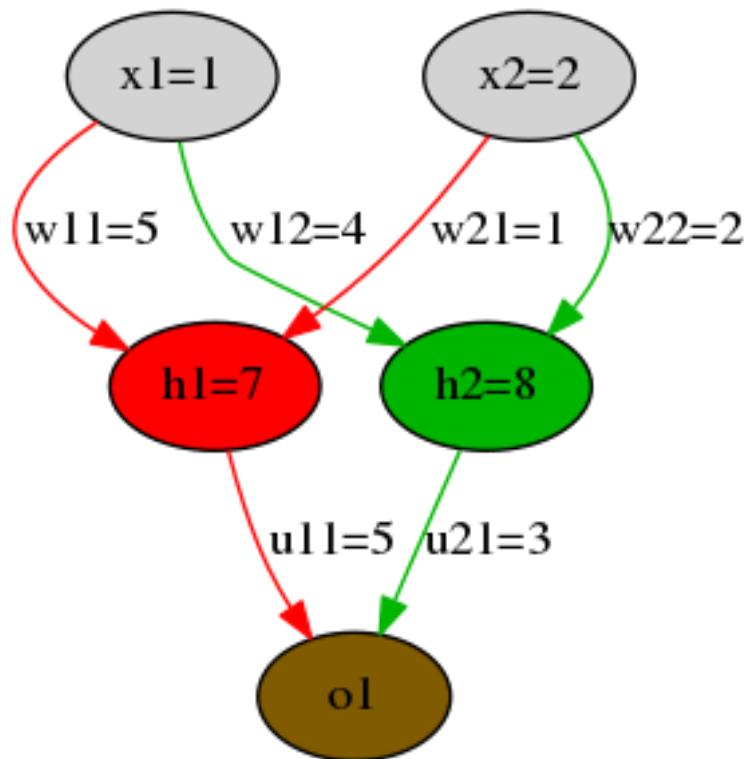
Degeneracy in Neural Networks



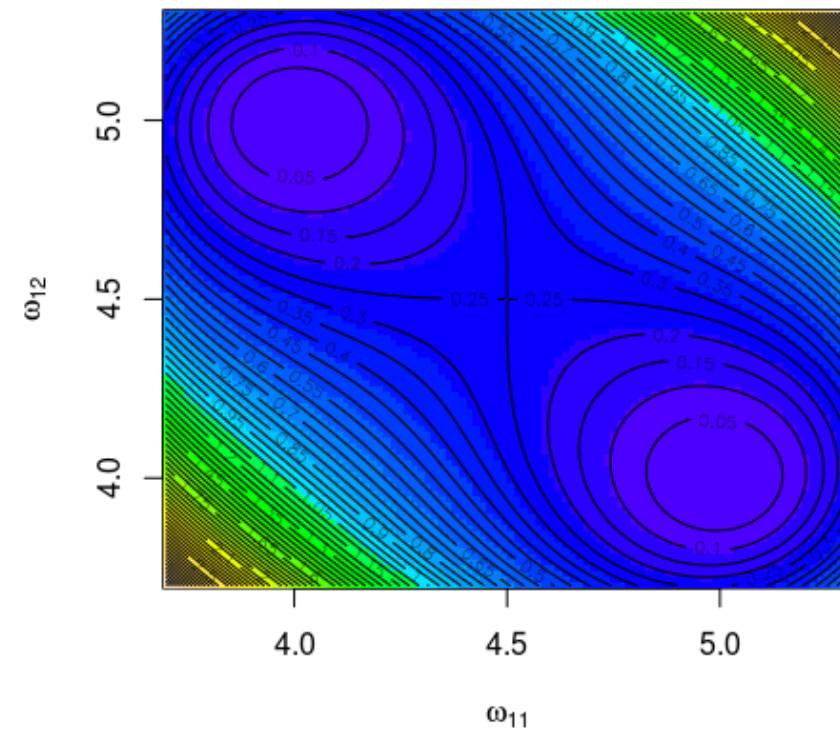
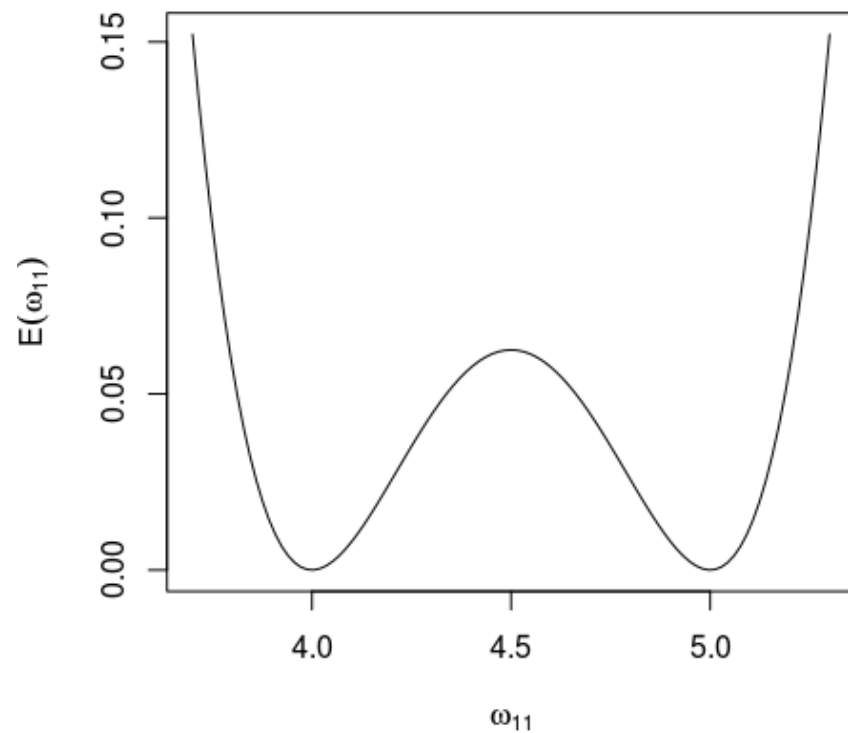
- A neural network is looking for the deepest valleys in this landscape
- As you can see there are many available



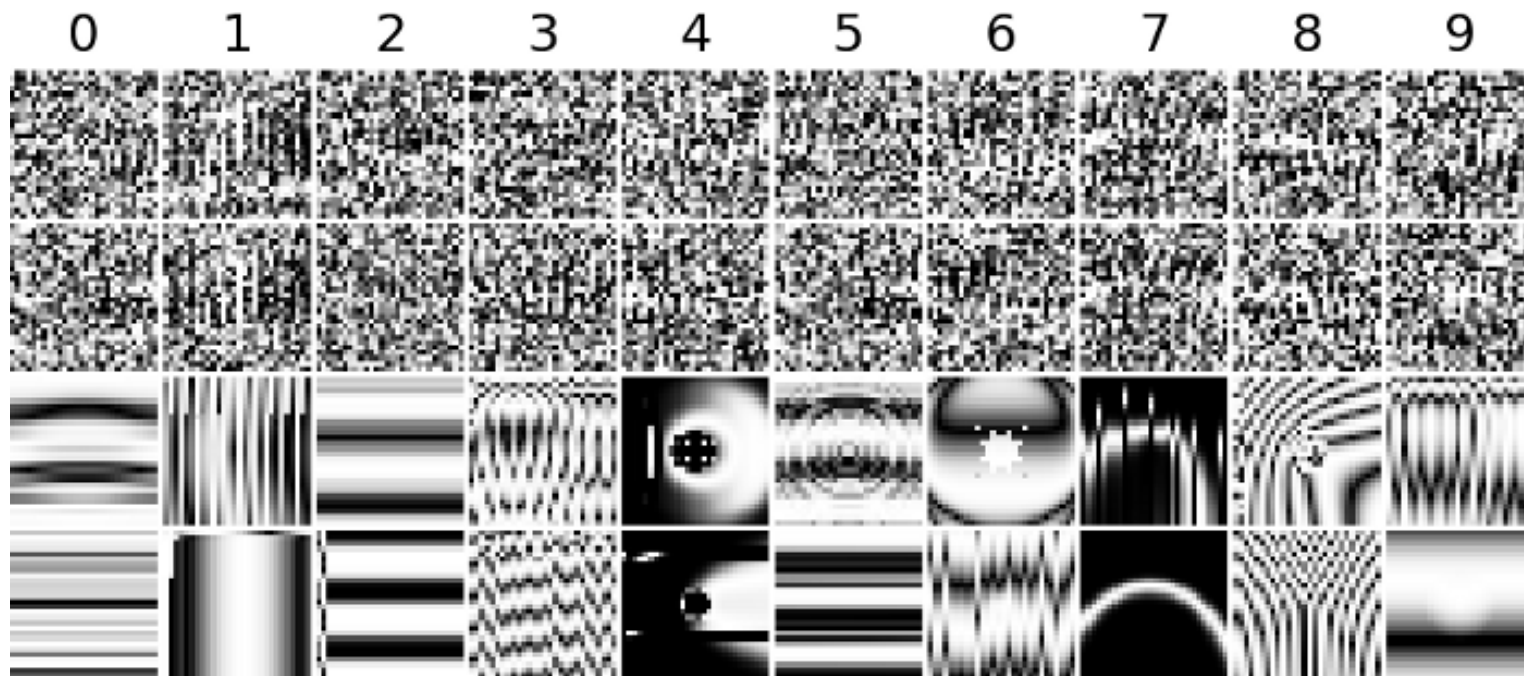
Degeneracy is in the structure



Energy landscape in the ω_{11}, ω_{12} parameters



So what's my point?

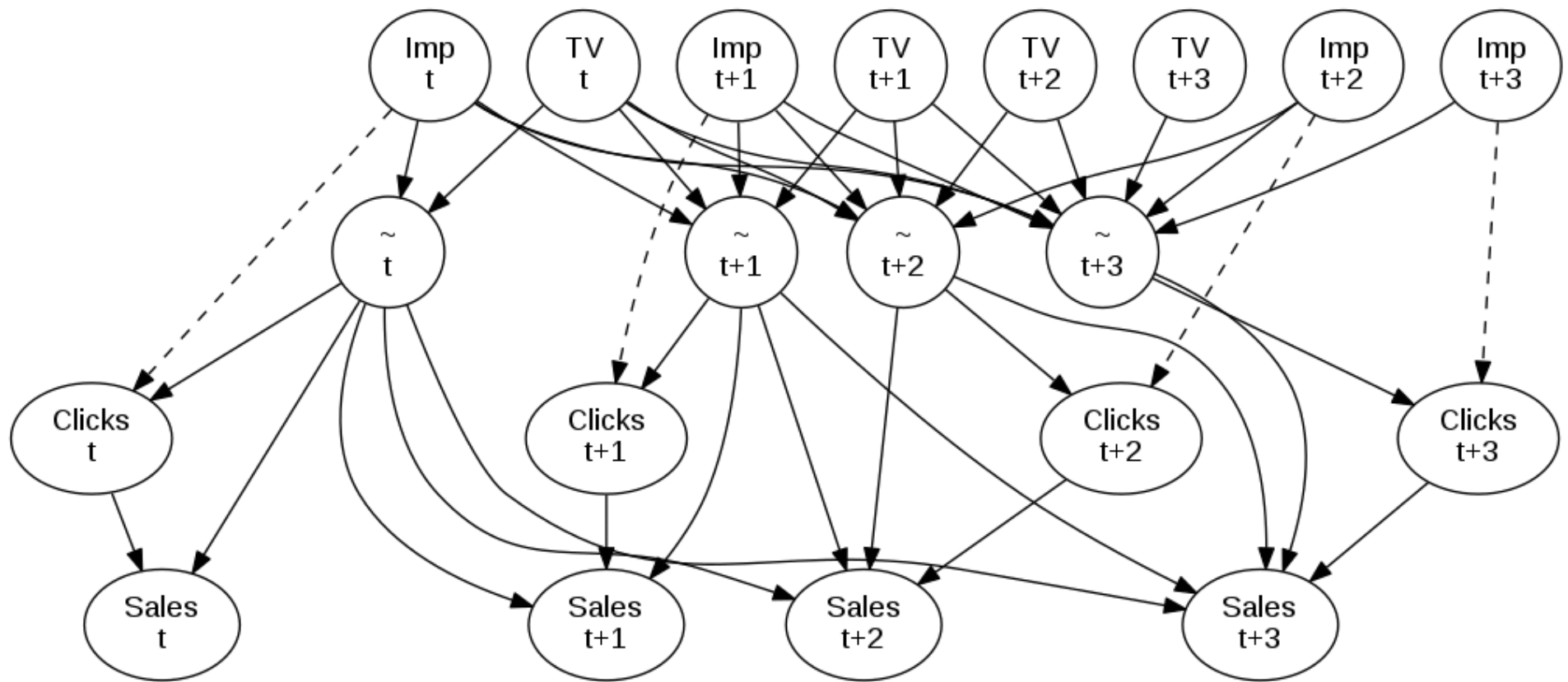


The point is that these spurious patterns will be realized in most if not all neural networks and their representation of the reality they're trying to predict will be inherently wrong. Read the paper by Nguyen A, Yosinski J, Clune J

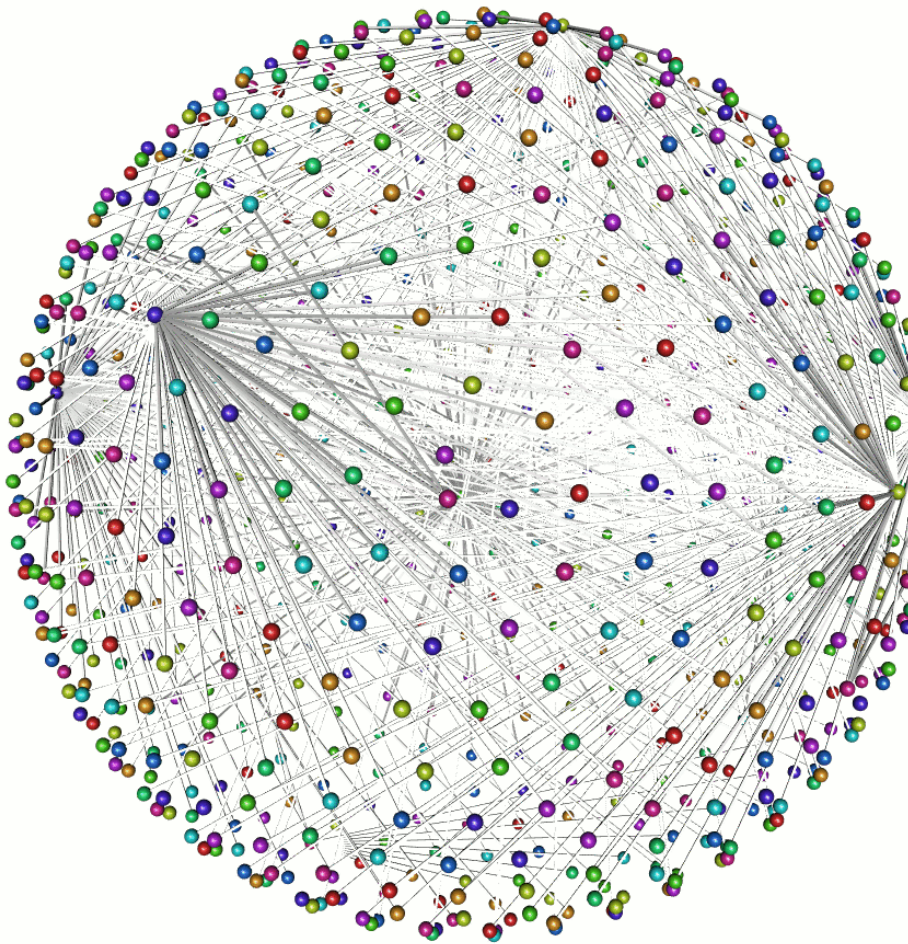


An example regarding time

Events are not temporally independent



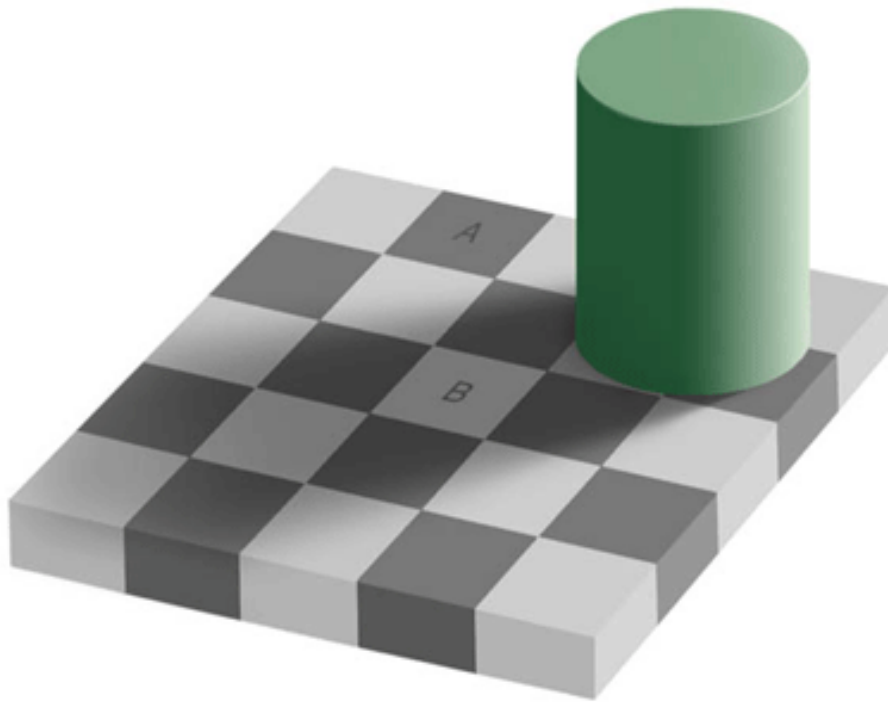
A real world example from Blackwood



- Every node in the network represents a latent or observed variable and the edges between

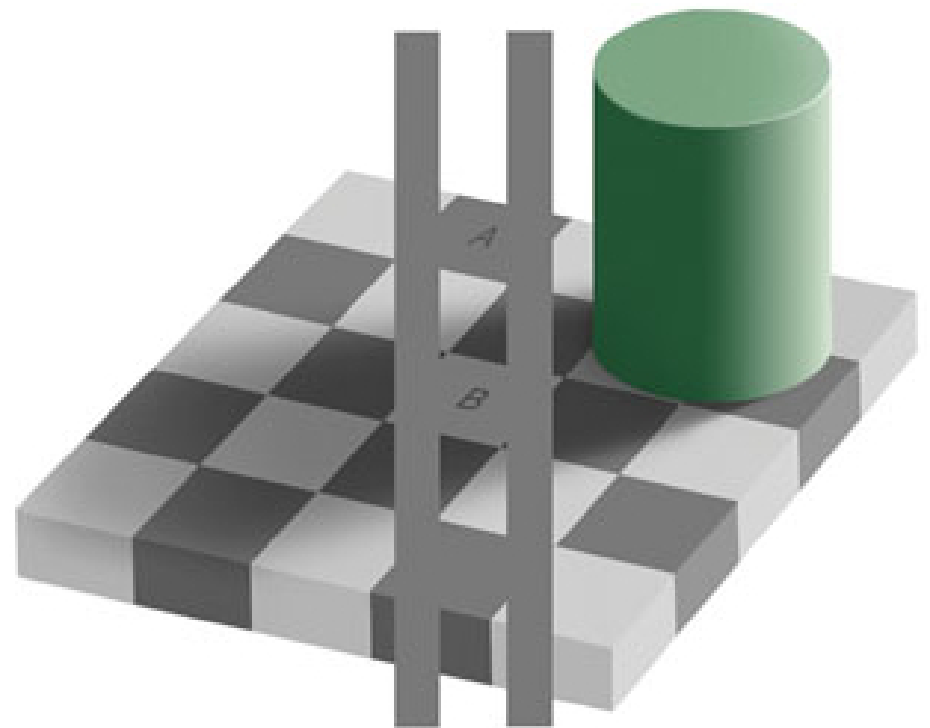
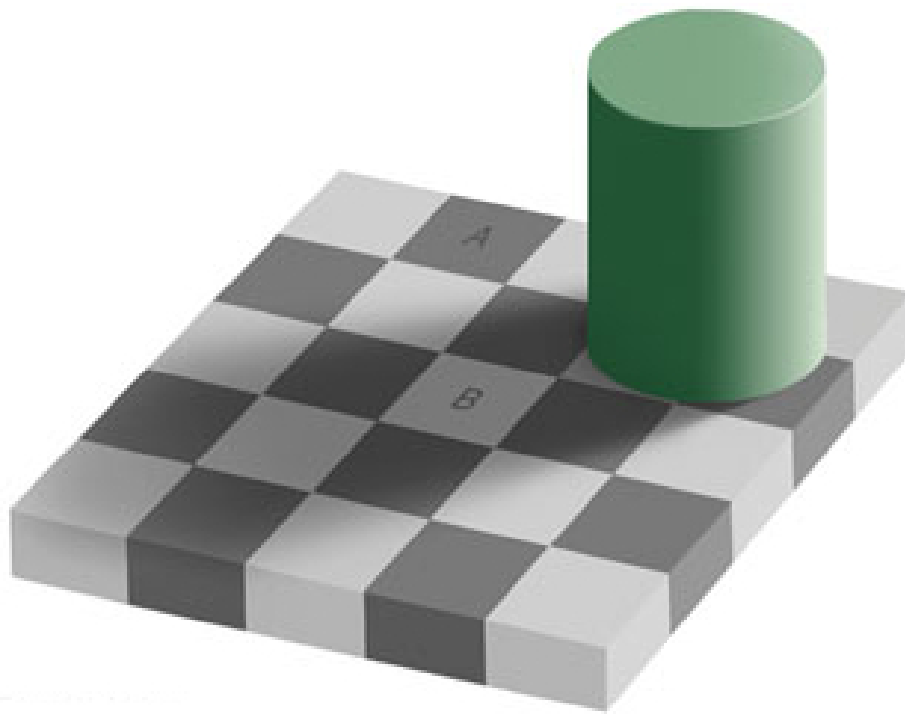
Our Bayesian brains

About cognitive strength



- Our brain is so successful because it has a strong anticipation about what will come
- Look at the tiles to the left and judge the color of the A and B tile
- To a human this task is easy because

The problem is only that you are wrong



Probabilistic programming

What is it?

Probabilistic programming creates systems that help make decisions in the face of uncertainty. Probabilistic reasoning combines knowledge of a situation with the laws of probability. Until recently, probabilistic reasoning systems have been limited in scope, and have not successfully addressed real world situations.

- It allows us to specify the models as we see fit
- Curse of dimensionality is gone
- We get uncertainty measures for all parameters
- We can stay true to the scientific principle
- We do not need to be experts in MCMC to use it!



Enter Stan a probabilistic programming language

Users specify log density functions in Stan's probabilistic programming language and get:

- full Bayesian statistical inference with MCMC sampling (NUTS, HMC)
- approximate Bayesian inference with variational inference (ADVI)
- penalized maximum likelihood estimation with optimization (L-BFGS)

Stan's math library provides differentiable probability functions & linear algebra (C++ autodiff). Additional R packages provide expression-based linear modeling, posterior visualization, and leave-one-out cross-validation.



A note about uncertainty

Task

- Suppose I gave you a task of investing 1 million USD in either Radio or TV advertising
- The average ROI for Radio and TV is 0.5
- How would you invest?

Further information

- Now I will tell you that the ROI's are actually distributions
- Radio and TV both have a minimum value of 0

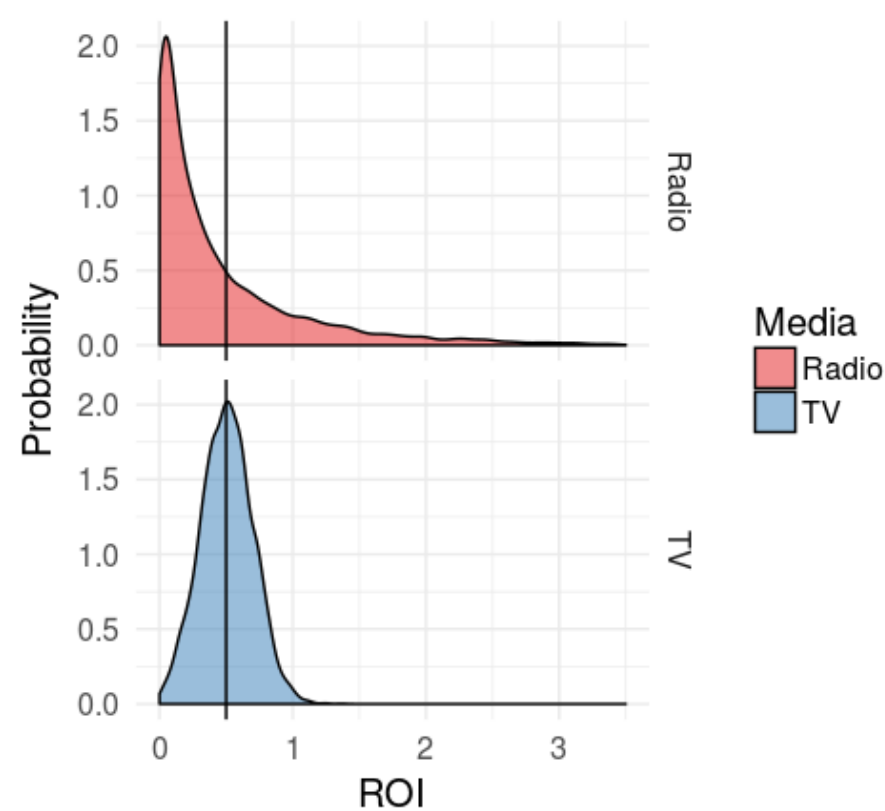
- Radio and TV have a maximum of 9.3 and 1.4 respectively
- Where do you invest?

Solution

- How to think about this?
- You need to ask the following question
- What is $p(ROI > 0.3)$?



A note about uncertainty - Continued



	Radio	TV
Mean	0.5	0.5
Min	0.0	-0.3
Max	9.3	1.4
Median	0.2	0.5
Mass	0.4	0.9
Sharpe	0.7	2.5



“*You cannot make
optimal decisions
without quantifying what
you don't know*”



Tying it all together

Deploying a Bayesian model using R

- There's a Docker image freely available with an up to date R version installed and the most common packages
- <https://hub.docker.com/r/drmike/r-bayesian/>

Features

- R: Well you know
- RStan: Run the Bayesian model
- OpenCPU: Immediately turn your R packages into REST API's



How to use it

First you need to get it

- `sudo docker pull dr mike/r-bayesian`
- `sudo docker run -it dr mike/r-bayesian bash`

You can also test the imbedded stupid application

- `docker run -d -p 80:80 -p 443:443 -p 8004:8004 dr mike/r-bayesian`
- `curl http://localhost:8004/ocpu/library/stupidweather/R/predictweather/json -H "Content-Type: application/json" -d '{"n":6}'`



Conclusion

Take home messages

- The time is ripe for marrying machine learning and inference machines
- Don't get stuck in patterns using existing model structures
- Stay true to the scientific principle
- Always state your mind!
- Be free, be creative and most of all have fun!



Session Information

For those who care

```
## setting value
## version R version 3.4.2 (2017-09-28)
## system x86_64, linux-gnu
## ui X11
## language en_US:en
## collate en_US.UTF-8
## tz Europe/Copenhagen
## date 2017-10-03
##
## package * version date source
## assertthat 0.2.0 2017-04-11 CRAN (R 3.3.3)
## backports 1.1.0 2017-05-22 CRAN (R 3.4.0)
## base * 3.4.2 2017-09-29 local
## bindr 0.1 2016-11-13 cran (@0.1)
## bindrcpp * 0.2 2017-06-17 cran (@0.2)
## bitops 1.0-6 2013-08-17 CRAN (R 3.3.0)
## caTools 1.17.1 2014-09-10 CRAN (R 3.4.0)
## colorspace 1.3-2 2016-12-14 CRAN (R 3.4.0)
## compiler 3.4.2 2017-09-29 local
## datasets * 3.4.2 2017-09-29 local
## devtools 1.13.3 2017-08-02 CRAN (R 3.4.1)
## digest 0.6.12 2017-01-27 CRAN (R 3.4.0)
## dplyr * 0.7.2 2017-07-20 cran (@0.7.2)
## evaluate 0.10.1 2017-06-24 cran (@0.10.1)
##  gdata 2.18.0 2017-06-06 cran (@2.18.0)
## ggplot2 * 2.2.1 2016-12-30 CRAN (R 3.3.2)
```

goto;
copenhagen

GOTO Copenhagen 2017
Conference Oct. 1-3, 2017



**Click 'Rate Session'
to rate session
and ask questions.**

 Follow us @gotocph



goto;
copenhagen

GOTO Copenhagen 2017
Conference Oct. 1-3, 2017



Please

**Remember to
rate this session**

Thank you!

 Follow us @gotocph





GOTO Copenhagen 2017
Conference Oct. 1-3, 2017

Did you **remember**
to rate the previous
session ?

 Follow us @gotocph



goto;
copenhagen

 Follow us @gotocph

