Connectionist Computing
COMP 30230/41390

Gianluca Pollastri

office: E0.95, Science East.

email: gianluca.pollastri@ucd.ie

Credits

- **Geoffrey Hinton, University of Toronto.**
  - **borrowed some of his slides for "Neural Networks" and "Computation in Neural Networks" courses.**

- **Ronan Reilly, NUI Maynooth.**
  - **slides from his CS4018.**

- **Paolo Frasconi, University of Florence.**
  - **slides from tutorial on Machine Learning for structured domains.**

Lecture notes on Brightspace

- **Strictly confidential...**
- **Slim PDF version will be uploaded later, typically the same day as the lecture.**
- **If there is demand, I can upload onto Brightspace last year's narrated slides.. (should be very similar to this year's material)**

Books

- No book covers large fractions of this course.

- Parts of chapters 4, 6, (7), 13 of Tom Mitchell's "Machine Learning"
- Parts of chapter V of Mackay's "Information Theory, Inference, and Learning Algorithms", available online at:

http://www.inference.phy.cam.ac.uk/mackay/itprnn/book.html

- Chapter 20 of Russell and Norvig's "Artificial Intelligence: A Modern Approach", also available at:

http://aima.cs.berkeley.edu/newchap20.pdf

- More materials later..

Marking

- **3 landmark papers to read, and submit a 10-line summary on Brightspace about: each worth 6-7%**
- **a connectionist model to build and play with on some sets, write a report: 30%**
- **Final Exam in the RDS (50%)**

Programming assignment

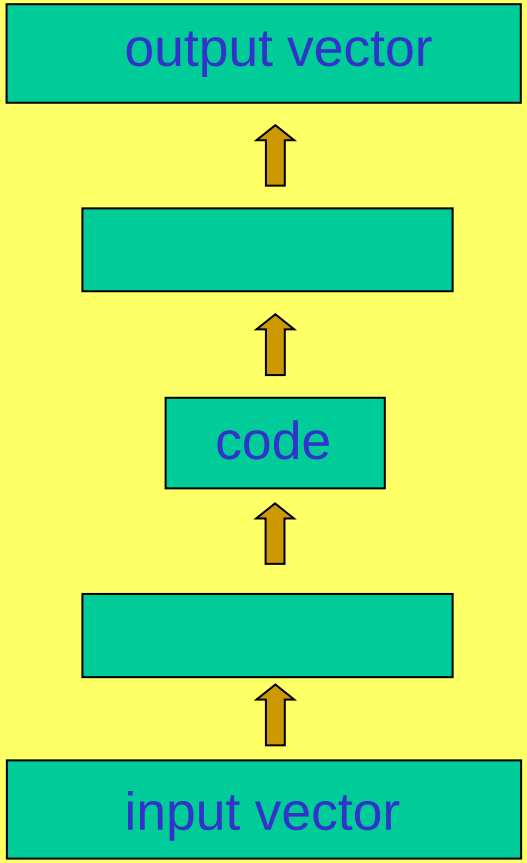
- Implement a Multi-Layer Perceptron, test it.
- The description on Brightspace.
- Submit through Brightspace code and test results by Dec the 5th at 23:59, any time zone of your choice (Baker Island?).
- 30% of the overall mark
- One third of a grade down every day late, that is: if you deserve an A and you're 1 day late you get an A-, 2 days late a B+, etc.

Unsupervised learning

- Without a desired output or reinforcement signal it is much less obvious what the goal is.
- Discover useful structure in large data sets without requiring a supervisory signal
  - Create representations that are better for subsequent supervised or reinforcement learning
  - Build a density model that can be used to:
    - Classify by seeing which model likes the test case data most (model selection)
    - Monitor a complex system by noticing improbable states.
    - Extract interpretable factors (causes or constraints)
- Improve learning speed for high-dimensional inputs

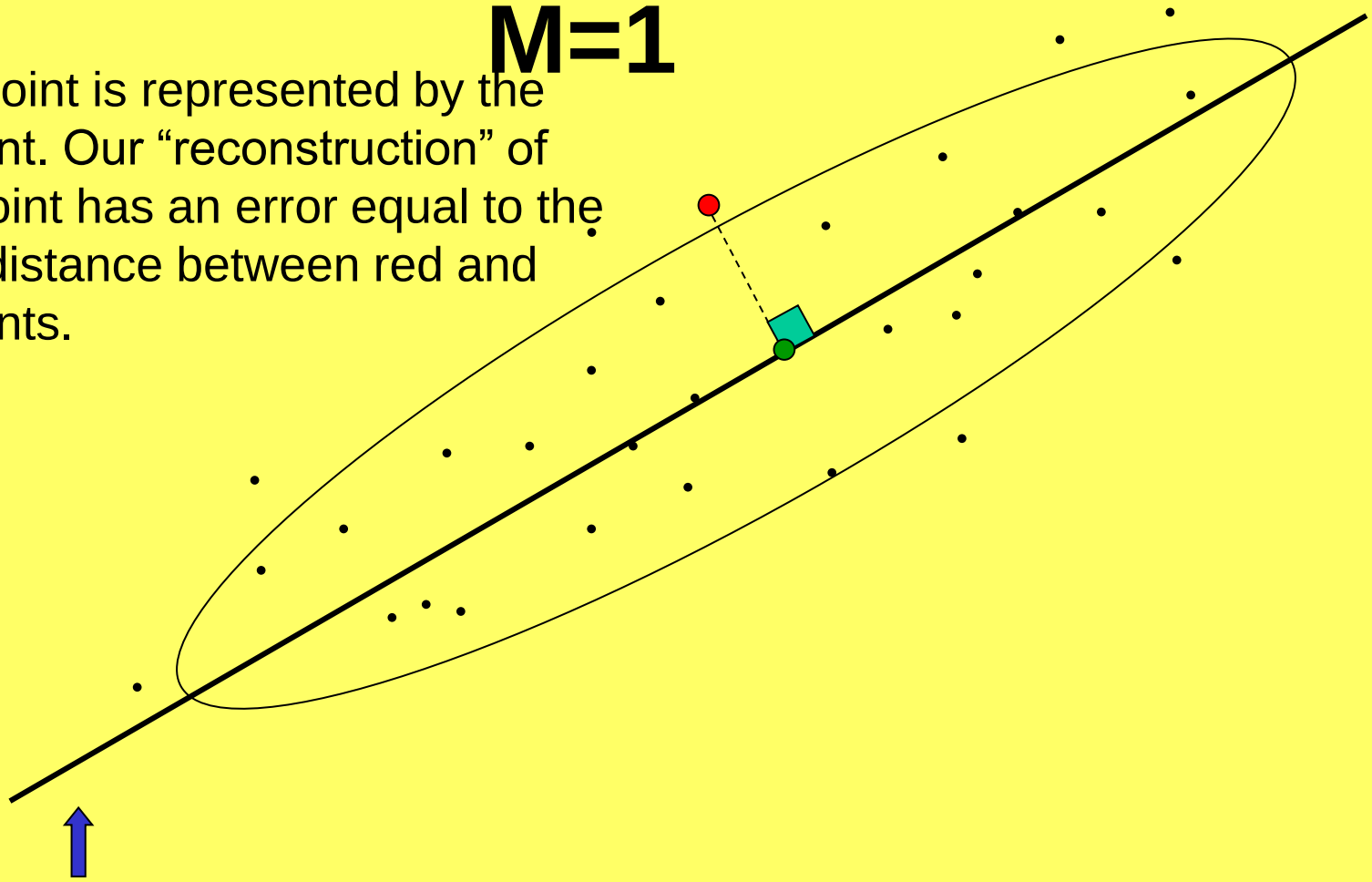
Using backprop for unsupervised learning

- **Try to make the output be the same as the input in a network with a central bottleneck.**
- **The activities of the hidden units in the bottleneck form an efficient code. The bottleneck does not have room for redundant features.**
- **Good for extracting independent features**



A picture of PCA with N=2 and M=1

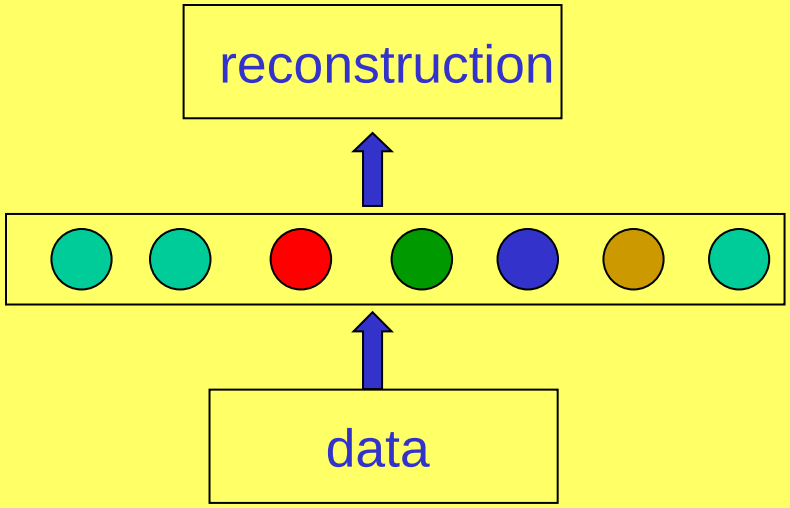
The red point is represented by the green point. Our “reconstruction” of the red point has an error equal to the squared distance between red and green points.

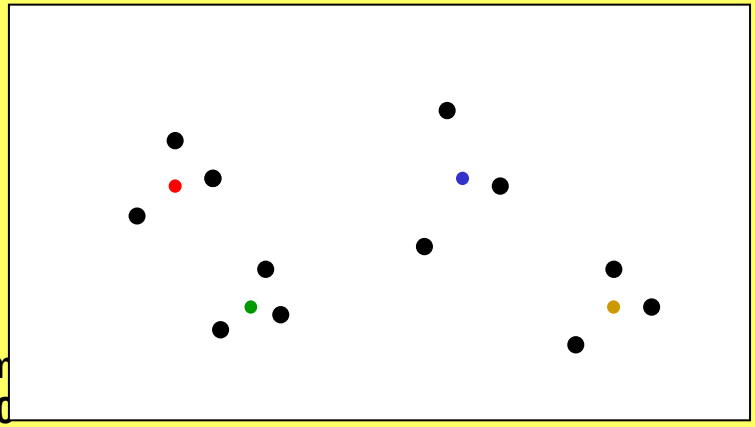


First principal component:
Direction of greatest variance

Self-supervised backprop in the non-linear case

- If we force the hidden unit whose weight vector is closest to the input vector to have an activity of 1 and the rest to have activities of 0, we get clustering.
- The weight vector of each hidden unit (HU->output) represents the centre of a cluster.
- Input vectors are reconstructed as the nearest cluster centre.
- Number of clusters = number of HU.





Linear vs. non-linear self-associating networks

- **Linear is global but not very expressive**
- **Non-linear tends to be more local but can be as expressive as you want.**

- **Precisely what you are doing with non-linear networks will depend on the details (hidden unit type, number of layers, etc.)**

The course so far..

- **History: Connectionism. Simple models of the brain. Rosenblatt's perceptron. Minsky and Papert's critique to connectionism. Associators. Hopfield nets. Boltzmann machine.**
- **Learning: Supervised learning. PAC learning, VC dimension.**
- **MLP: Backpropagation. Expressive power. Complexity. Applications: NetTalk, SS prediction, handwritten digits. Invariances. Softmax and relative entropy. Overfitting. Gradient descent problems/solutions.**
- **Non-supervised learning: Reinforcement learning: Tesauro's paper. Unsupervised learning: PCA and Self-supervised clustering.**

Models

- **So far we made models of phenomena.**
- **Our models, in general, were highly complex hypotheses based on the data, and on some parameters, or weights.**
- **So, it is M=M(W).**
- **Can we make the case that we've been picking the *most plausible* models given the data we observed?**

plausibility

- **Let's call f(X|I) the plausibility of (our degree of belief in) X given that we have observed information I.**
- **We may want to require that f() has a few simple properties.**

plausibility properties

- if f(X|I)>f(Y|I), f(Y|I)>f(Z|I), then f(X|I)>f(Z|I)
- f(X|I) = F[f(¬X|I)]
- f(X,Y|I) = G[f(X|I), f(Y|X,I)]

we know f

- **Given the properties above, it is always possible to map f() into the [0,1] interval.**
- **Once we mapped f(), it turns out that F[] and G[] are defined:**
- **F[x] = 1-x**
- **G[x,y] = xy**
- **Let's call this thing *probability***

probability properties

- **if P(X|I)>P(Y|I), P(Y|I)>P(Z|I), then P(X|I)>P(Z|I)**
- **P(X|I) = 1 - P(¬X|I)**
- **P(X,Y|I) = P(X|I)P(Y|X,I)**

Bayes' rule

- **Since it's obvious that it must be P(X,Y|I) = P(Y,X|I), we can expand the third rule:**

- **P(X|Y,I) = P(Y|X,I)P(X|I)/P(Y|I)**

- **this is known as Bayes' rule**
- **P(X|I) is called *prior* probability**
- **P(X|Y,I) is called *posterior***
- **Bayes' rule tells us how to update our belief in X after we've observed Y.**

Models, data and Bayes

- **Let's now assume that we want to gauge the probability of a model M=M(w) given the data D we have observed.**
- **We can rewrite Bayes rule (I is implied) as:**

- **$P(M|D) = P(D|M)P(M)/P(D)$**

Prior, posterior, likelihood

- **P(M) is our estimate that model M is correct before seeing D.**
- **P(M|D) is our estimate that M is correct after having looked at D.**
- **P(D|M) is generally called *likelihood*.**

Posterior as next prior

- **If we aquire data serially, as D_1, D_2, .. D_n, then we can rewrite Bayes as:**

- **$P(M| D_n, D_{n-1}, .. D_1) = P(M| D_{n-1}, .. D_1)$ $P(D_n|M, D_{n-1}, .. D_1) /P(D_n | D_{n-1}, .. D_1)$**

- **Our posterior at step n-1 becomes prior for step n.**

taking the logs

- **Probabilities can be small. And often we rather deal with sums than multiplications.**
- **Taking the logs of Bayes' rule:**

- **$\log P(M|D) = \log P(D|M) + \log P(M) - \log P(D)$**