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 <u>6-7%</u>**
- **a connectionist model to build and play with on some sets, write a report: <u>30%</u>**
- **Final Exam in the RDS (<u>50%</u>)**

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.

The course so far..

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

probability properties

- if $P(X|I)>P(Y|I)$, $P(Y|I)>P(Z|I)$, then $P(X|I)>P(Z|I)$
- $P(X|I) = 1 - P(\neg X|I)$
- $P(X,Y|I) = P(X|I)P(Y|X,I)$

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:

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

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.

Priors: P(M)

- The use of priors is often criticised because it might introduce an arbitrary element into the inference mechanism.
- In reality:
  - Often, the more data we have, the less important priors become.
  - Non-informative priors (e.g. uniform) can often be derived.
  - Priors are really always used, even when they aren't mentioned, so it is better to deal with them explicitly.
  - In the Bayesian framework we can at least assess their impact.

Likelihood: P(D|M)

- **This can be dealt with easily if our model can assess naturally the probability of an observation D (e.g. if the outputs of a softmax MLP could be interpreted as probabilities..), or *generates* data with a certain probability (think of Boltzmann machines).**

- **In NNs we used Error functions; how can we translate them into likelihoods? We will need to make some assumption on how the error is distributed.**

Using the Bayesian machinery

- We can try to select the model that minimises the following error:

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

- This is called MAP (maximum a posteriori) estimate, because we are maximising the posterior.

MAP and ML

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

- The $P(D)$ term does not depend on the model, so we can ignore it during maximisation:
- $E = -\log P(D|M) - \log P(M)$

- If $P(M)$ is uniform over all the models (or if we just don't like it), then we can consider just the likelihood $P(D|M)$. In this case we are performing ML (maximum likelihood):
- $E = -\log P(D|M)$

How to do the M part of MAP and ML

- ML and MAP give us an objective, but not a method to achieve it.
- Practically performing the maximisation can be a rough task, and there are a lot of techniques to do it. For instance gradient descent (but also simulated annealing, etc. etc.)..

MAP/ML, NNs and supervised learning

- **How can we apply the MAP/ML idea to an NN with weights w?**
- **Let's assume that we have a set of examples (d_i,t_i) where d_i is the i-th input and t_i the corresponding target.**
- **Then we can rewrite the likelihood as:**

$$P((d_i, t_i)|w) = P(t_i|d_i, w)P(d_i|w) = P(t_i|d_i, w)P(d_i)$$

MAP/ML, NNs and supervised learning

- **The whole error (minus log posterior) can be written as:**

$$-logP(w|D) = -logP(D|w) - logP(w) + logP(D) =$$

$$-log\prod_{i=1}^{N}[P(t_i|d_i, w)P(d_i)] - logP(w) + logP(D) =$$

$$-\sum_{i=1}^{N} logP(t_i|d_i, w) - \sum_{i=1}^{N} logP(d_i) - logP(w) + logP(D)$$

MAP/ML, NNs and supervised learning

- The terms expressing the probability of the inputs don't depend on the weights of the model.
- What we are interested in minimising is:

$$-\sum_{i=1}^{N} logP(t_i|d_i, w) - logP(w)$$

The prior

- If we assume that P(w) has a Gaussian distribution, -log P(w) contributes a term proportional to $-w^2$ to the error. This is precisely ***weight decay***.
- ***Weight sharing*** can also be included into the prior.