Your name:

Your student number:

Final exam for CSC 321
April 11, 2013, 7:00pm – 9:00pm
No aids are allowed.

This exam has two sections, each of which is worth a total of 10 points. Answer all 10 questions in section A, and 5 of the 10 questions in section B.

Section A. Answer all 10 of these questions. Each is worth one mark. These are short questions. When we say "briefly explain", don't start writing a whole page of text. The main idea, in one or two sentences, is enough.

A1. 1 mark.

a) (0.5 marks) Is a Sigmoid Belief Network for supervised learning or for unsupervised learning? Explain briefly.

b) (0.5 marks) Is a mixture of experts for supervised learning or for unsupervised learning? Explain briefly.

A2. 1 mark.

a) (0.5 marks) Is a mixture of Gaussians for supervised learning or for unsupervised learning? Explain briefly.

b) (0.5 marks) Is an autoencoder for supervised learning or for unsupervised learning? Explain briefly.

A3. 1 mark. Briefly explain the trigram method of language modeling.

A4. 1 mark. What is the procedure of 5-fold cross-validation, and what is its advantage over the traditional approach of simply splitting one's available data into a training set and a validation set?

A5. 1 mark. We've seen that averaging the outputs from multiple models typically gives better results than using just one model. Let's say that we're going to average the outputs from 10 models. Of course, we want 10 good models, i.e. models that also perform well individually. What additional property of a collection of 10 models makes that collection a good candidate for output averaging?

A6. 1 mark. What does it mean that a Markov Chain has been run for so long that it has reached **thermal equilibrium**?

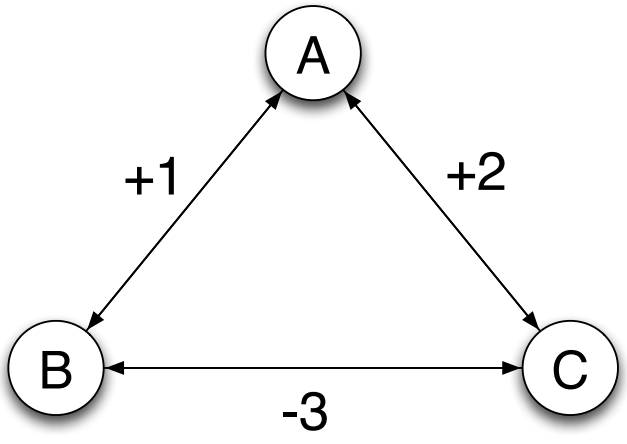
A7. 1 mark. When we're changing the states of the units in a Hopfield network, in search of a low energy configuration, we change the states of the units one at a time. What could go wrong if instead we change the state of multiple units at the same time? Illustrate by drawing a concrete Hopfield network (clearly indicate what the connection weights are) and explaining what goes wrong if we change the state of multiple units at the same time, in your network.

A8. 1 mark. How is training a 1-hidden-layer autoencoder very similar to training a Restricted Boltzmann Machine with CD-1 (Contrastive Divergence 1)? Don't answer with mathematical formulas; just answer in simple English.

A9. 1 mark. In assignment 1, we trained a language model that produced a feature vector for each word in its dictionary. Afterwards, we took all those feature vectors, and mapped them to a two-dimensional space using t-SNE, so that it could be displayed on paper for us to see patterns of similarity in the learned feature vectors. We don't need t-SNE for that: an autoencoder can do it, too. Explain how an autoencoder can be used for that task that t-SNE performed for us in assignment 1.

A10. 1 mark. For the Hopfield network below, write down the energy of all 8 configurations in the following table:

State of A	State of B	State of C	Energy
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	



Section B. Answer 5 of these 10 questions. Each of these is worth 2 marks. If you answer more than 5 of these 10 questions, your worst 5 answers will be used, so just don't do that. If you wrote something for a question and you later decide not to answer that question after all, cross out what you wrote and clearly write "don't mark this". Again, explaining the main idea briefly is better than writing something lengthy.

B1. 2 marks. When you're training a neural network with some form of stochastic gradient descent, there's always the learning rate to choose.

a) (0.5 marks) What problem will result from using a learning rate that's too large, and how can one detect that problem?

b) (0.5 marks) What's the problem if the learning rate is too small, and how can one detect that problem?

c) (1 mark) We can charge the computer with adapting the learning rate during training, if we don't want to do it ourselves. If we do that, every parameter that's being learned can have its own learning rate. This approach is called “adaptive learning rates”. Explain a simple but reasonable strategy for automatically adapting those learning rates during training. You don't have to explain in full pseudo-code; just explain the main idea.

B2. 2 marks.

a) (1 mark) If we have a recurrent neural network (RNN), we can view it as a different type of network by "unrolling it through time". Briefly explain what that entails.

b) (1 mark) Briefly explain how “unrolling through time” is related to “weight sharing” in convolutional networks.

B3. 2 marks.

a) (1 mark) For training a Boltzmann Machine, what is the objective function? Write it down as a mathematical formula, and explain the meaning of the symbols that you use.

b) (1 mark) And what is the gradient of that objective function, for the weight on the connection between unit **i** and unit **j**? Write it down as a mathematical formula, and explain the meaning of the symbols that you use.

B4. 2 marks. A Hopfield network can be made stochastic by introducing a temperature and stochastic state changes for the units. This is most commonly done with temperature T=1, but other temperatures can also be used.

a) (1 mark). Write the mathematical formula for the probability of turning on unit **i**. Your formula will involve the temperature, T. Clearly define any nontrivial symbols that you use.

$P(s_i=1)$ =

b) (0.5 marks) What is the effect of using T=infinity?

c) (0.5 marks) What is the effect of using T=0?

B5. 2 marks. In a deep neural network, or a recurrent neural network, we can get vanishing or exploding gradients because the backward pass of backpropagation is linear, even for a network where all hidden units are logistic.

a) (1 mark) Explain in what sense the backward pass is linear.

b) (1 mark) Why does an Echo State Network not suffer from this problem?

B6. 2 marks. If the hidden units of a network are independent of each other, then it's easy to get a sample from the correct distribution, which is a very important advantage.

a) (0.5 marks) For a Sigmoid Belief Network where the only connections are from hidden units to visible units (i.e. no hidden-to-hidden or visible-to-visible connections), when we condition on the state of the visible units, are the hidden units conditionally independent of each other? Explain very briefly.

b) (0.5 marks) For a Sigmoid Belief Network where the only connections are from hidden units to visible units (i.e. no hidden-to-hidden or visible-to-visible connections), when we don't condition on anything, are the hidden units independent of each other? Explain very briefly.

c) (0.5 marks) For a Restricted Boltzmann Machine, when we don't condition on anything, are the hidden units independent of each other? Explain very briefly.

d) (0.5 marks) For a Restricted Boltzmann Machine, when we condition on the state of the visible units, are the hidden units conditionally independent of each other? Explain very briefly.

B7. 2 marks. In Bayesian learning, we consider not just one, but many different weight vectors. Each of those is assigned a probability by which it is weighted in producing the final output.

a) (1 mark) Write down Bayes' rule as it applies to supervised neural network learning. Clearly define the symbols that you are using.

b) (0.5 marks) Clearly indicate which part of the formula is the "prior distribution", which is the "likelihood term", and which is the "posterior distribution".

c) (0.5 marks) In this context, how is Maximum A Posteriori (MAP) learning different from Maximum Likelihood (ML) learning?

B8. 2 marks. We've seen a variety of generative models. Some were causal generative models, and others were energy-based generative models.

a) (1 mark) Explain the difference between those two types of generative models.

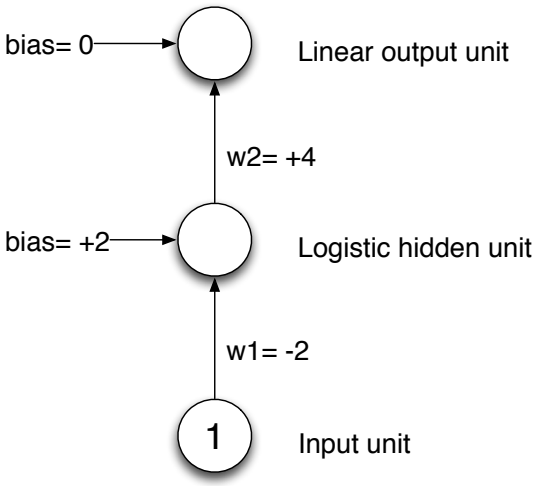
b) (0.5 marks) Give an example of a causal generative model that we studied in class.

c) (0.5 marks) Give an example of an energy-based generative model that we studied in class.

B9. 2 marks. Here you see a very small neural network: it has one input unit, one hidden unit (logistic), and one output unit (linear). Let's consider one training case. For that training case, the input value is 1 (as shown in the diagram), and the target output value is 1. We're using the standard squared error loss function:

$E = (t-y)^2 / 2$

The numbers in this question have been constructed in such a way that you don't need a calculator.



a) (0.5 marks) What is the output of the hidden unit and the output unit, for this training case?

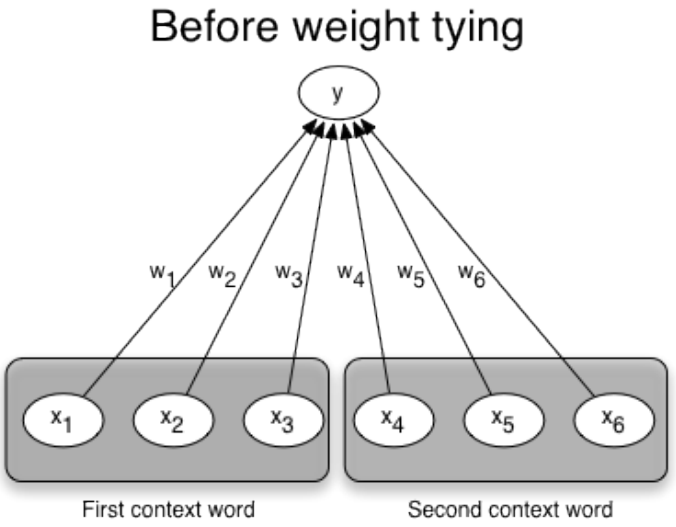
b) (0.5 marks) What is the loss, for this training case?

c) (0.5 marks) What is the derivative of the loss w.r.t. w2, for this training case?

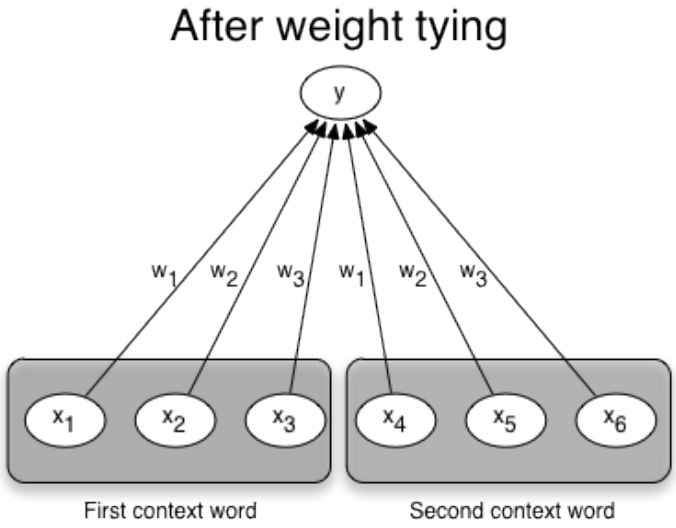
d) (0.5 marks) What is the derivative of the loss w.r.t. w1, for this training case?

B10. 2 marks. *(the text of this question is long, but it's not complicated)*

Suppose that we have a vocabulary of 3 words, "a", "b", and "c", and we want to predict the next word in a sentence given the previous two words. For this network, we don't want to use feature vectors for words: we simply use the local encoding, i.e. a 3-component vector with one entry being 1 and the other two entries being 0.



In the language models that we have seen so far, each of the context words has its own dedicated section of the network, so we would encode this problem with two 3-dimensional inputs. That makes for a total of 6 dimensions. For example, if the two preceding words (the "context" words) are "c" and "b", then the input would be (0, 0, 1, 0, 1, 0). Clearly, the more context words we want to include, the more input units our network must have. More inputs means more parameters, and thus increases the risk of overfitting. Here is a proposal to reduce the number of parameters in the model:



Consider a single neuron that is connected to this input, and call the weights that connect the input to this neuron w1, w2, w3, w4, w5, and w6. w1 connects the neuron to the first input unit, w2 connects it to the second input unit, etc. Notice how for every neuron, we need as many weights as there are input dimensions (6 in our case), which will be the number of words times the length of the context. A way to reduce the number of parameters is to tie certain weights together, so that they share a parameter. One possibility is to tie the weights coming from input units that correspond to the same word but at different context positions. In our example that would mean that w1=w4, w2=w5, and w3=w6 (see the "after" diagram).

Explain the main weakness that that change creates.