Final Exam

Continue Course

Back to Week 15

✓ **18/18** points earned (100%)

Quiz passed!

✓ 1 / 1 points

1\.
One regularization technique is to start with lots of connections in a neural network, and then remove those that are least useful to the task at hand (removing connections is the same as setting their weight to zero). Which of the following regularization techniques is best at removing connections that are least useful to the task that the network is trying to accomplish?

✓ 1 / 1 points

2\.
Why don't we usually train Restricted Boltzmann Machines by taking steps in the exact direction of the gradient of the objective function, like we do for other systems?

✓ 1 / 1 points

3\.

When we want to train a Restricted Boltzmann Machine, we could try the following strategy. Each time we want to do a weight update based on some training cases, we turn each of those training cases into a full configuration by adding a sampled state of the hidden units (sampled from their distribution conditional on the state of the visible units as specified in the training case); and then we do our weight update in the direction that would most increase the goodness (i.e. decrease the energy) of those full configurations. This way, we expect to end up with a model where configurations that match the training data have high goodness (i.e. low energy).

However, that's not what we do in practice. Why not?

✓ 1 / 1 points

4.

CD-1 and CD-10 both have their strong sides and their weak sides. Which is the main advantage of CD-10 over CD-1?

✓ 1 / 1 points

5.

CD-1 and CD-10 both have their strong sides and their weak sides. Which are significant advantages of CD-1 over CD-10? Check all that apply.

✓ 1 / 1 points

6.

With a lot of training data, is the perceptron learning procedure more likely or less likely to converge than with just a little training data?

Clarification: We're not assuming that the data is always linearly separable.

✓ 1 / 1 points

7.

You just trained a neural network for a classification task, using some weight decay for regularization. After training it for 20 minutes, you find that on the validation data it performs much worse than on the training data: on the validation data, it classifies 90% of the data cases correctly, while on the training data it classifies 99% of the data cases correctly. Also, you made a plot of the performance on the training data and the performance on the validation data, and that plot shows that at the end of those 20 minutes, the performance on the training data is improving while the performance on the validation data is getting worse.

What would be a reasonble strategy to try next? Check all that apply.

✓ 1 / 1 points

8.

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. For which systems, and under which conditions, are the hidden units independent of each other? Check all that apply.

✓ 1 / 1 points

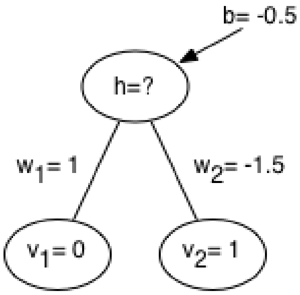
9.

What is the purpose of momentum?

✓ 1 / 1 points

10.

Consider a Restricted Boltzmann Machine with 2 visible units v_1, v_2 and 1 hidden unit h. The visible units are connected to the hidden unit by weights w_1, w_2 and the hidden unit has a bias b. An illustration of this model is given below.



The energy of this model is given by: $E(v_1, v_2, h) = -w_1 v_1 h - w_2 v_2 h - bh$. Recall that the joint probability $P(v_1, v_2, h)$ is proportional to $\exp\left(-E(v_1, v_2, h)\right)$.

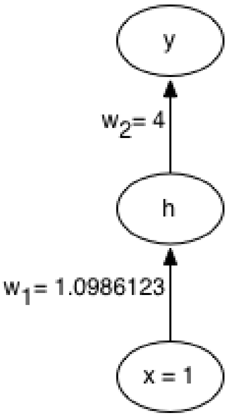
Suppose that $w_1 = 1, w_2 = -1.5, b = -0.5$. What is the conditional probability $P(h = 1|v_1 = 0, v_2 = 1)$? Write down your answer with at least 3 digits

after the decimal point.

✓ 1 / 1 points

11.

Consider the following feed-forward neural network with **one *logistic* hidden neuron** and **one *linear* output neuron**.



The input is given by $x = 1$, the target is given by $t = 5$, the input-to-hidden weight is given by $w_1 = 1.0986123$, and the hidden-to-output

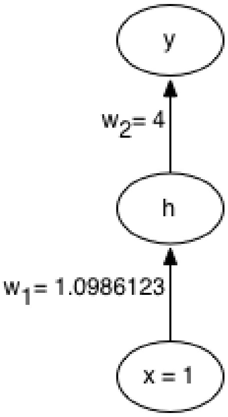
weight is given by $w_2 = 4$ (there are no bias parameters). What is the cost incurred by the network when we are using the **squared error cost**?

Remember that the squared error cost is defined by $\text{Error} = \frac{1}{2}(y - t)^2$. Write down your answer with at least 3 digits after the decimal point.

✓ 1 / 1 points

12.

Consider the following feed-forward neural network with **one *logistic* hidden neuron** and **one *linear* output neuron**.



The input is given by $x = 1$, the target is given by $t = 5$, the input-to-hidden weight is given by $w_1 = 1.0986123$, and the hidden-to-output

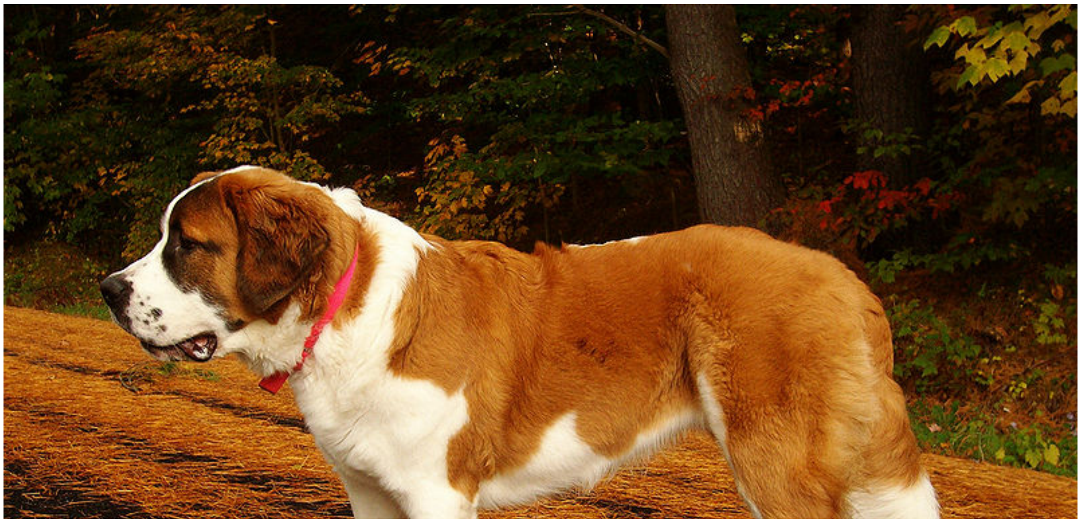
weight is given by $w_2 = 4$ (there are no bias parameters). If we are using the **squared error cost** then what is $\frac{\partial \text{Error}}{\partial w_1}$, the derivative of the error with respect to w_1? Remember that the squared error cost is defined by

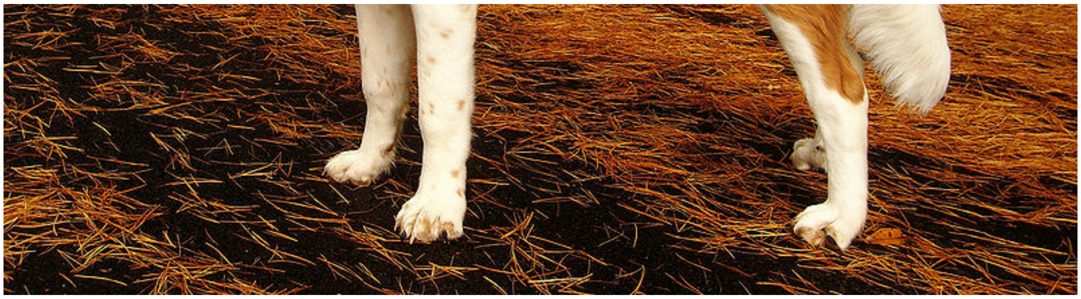
$\text{Error} = \frac{1}{2}(y - t)^2$. Write down your answer with at least 3 digits after the decimal point.

1 / 1
points

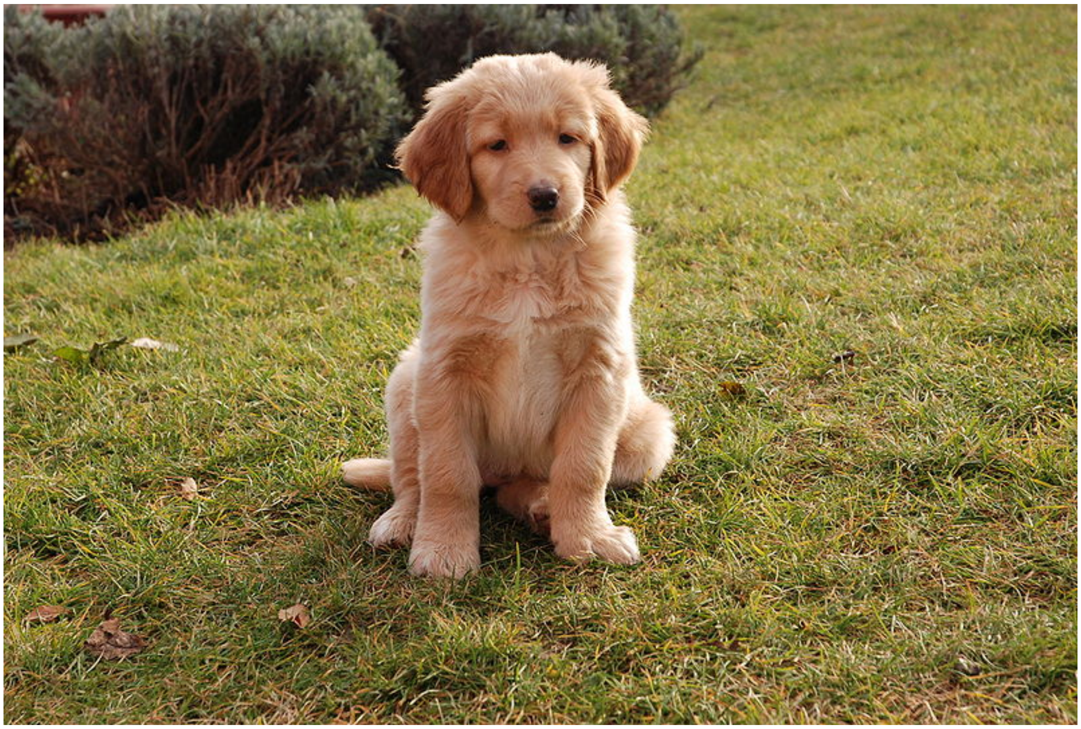
13.

Suppose that we have trained a **semantic hashing** network on a large collection of images. We then present to the network four images: two dogs, a cat, and a car (shown below).

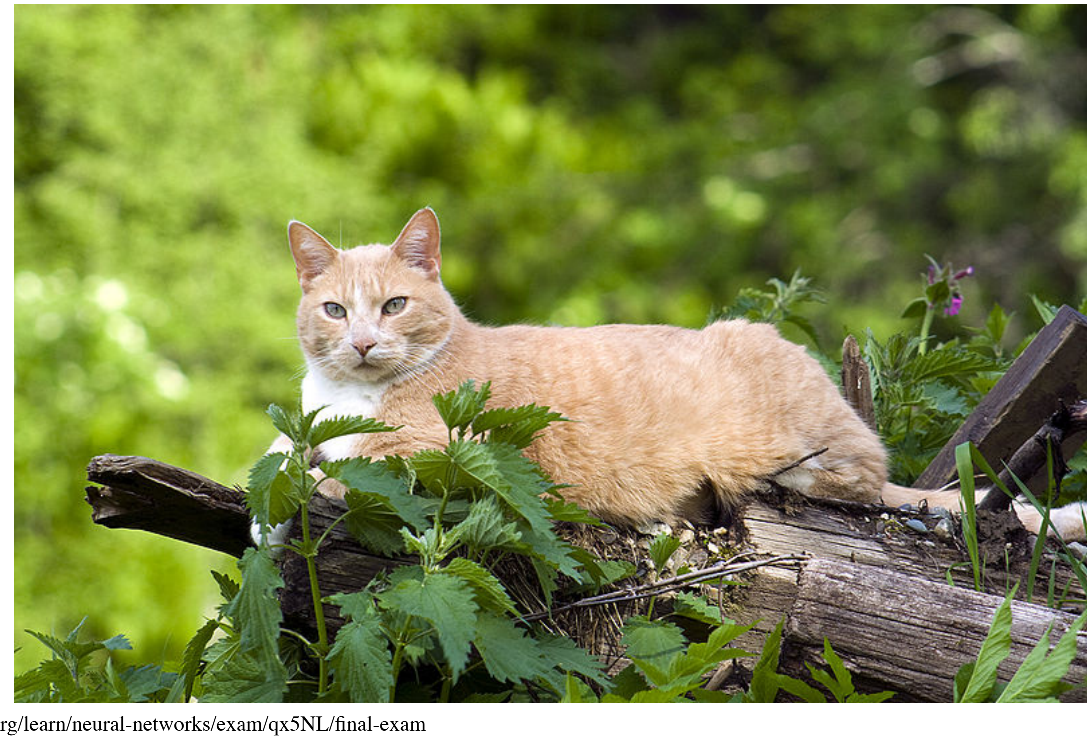


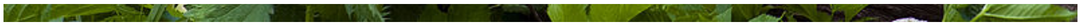


Dog 1



Dog 2





Cat

Car

The network produces four binary vectors:

$$\begin{array}{l} \text{(a)}\ [0,1,1,1,0,0,1] \\ \text{(b)}\ [1,0,0,0,1,0,1] \\ \text{(c)}\ [1,0,0,0,1,1,1] \\ \text{(d)}\ [1,0,0,1,1,0,0] \end{array}$$

One may wonder which of these codes was produced from which of the images. Below, we've written four possible scenarios, and it's your job to select

the most plausible one.

Remember what the purpose of a semantic hashing network is, and use your intuition to solve this question. If you want to quantitatively compare

binary vectors, use the number of different elements, i.e., the *Manhattan distance*. That is, if two binary vectors are [1,0,1] and [0,1,1] then

their Manhattan distance is 2.

✓ 1 / 1 points