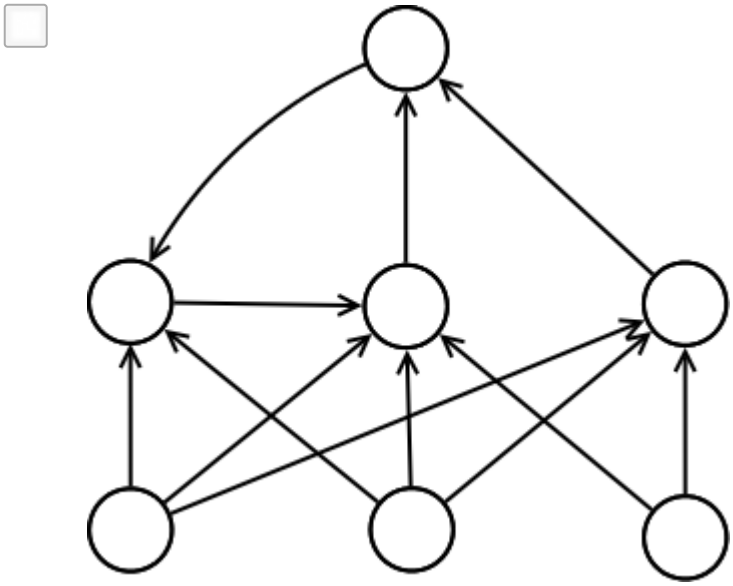
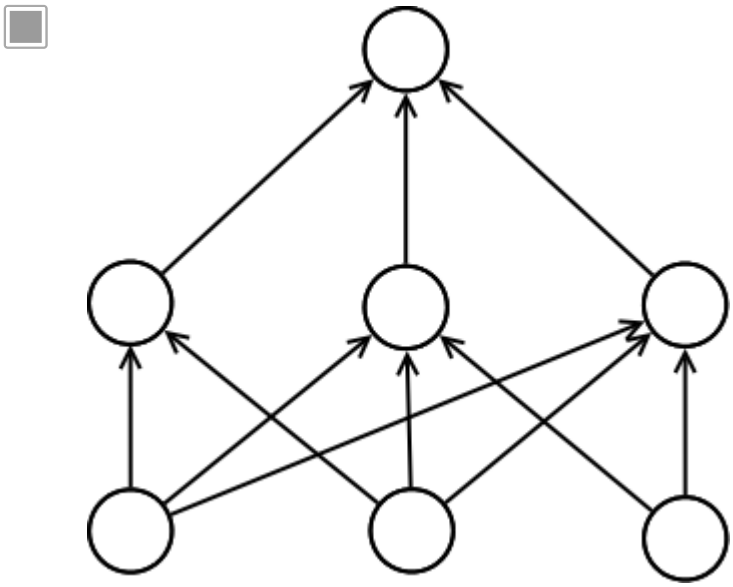
Lecture 3 Quiz

Quiz, 6 questions

1 point

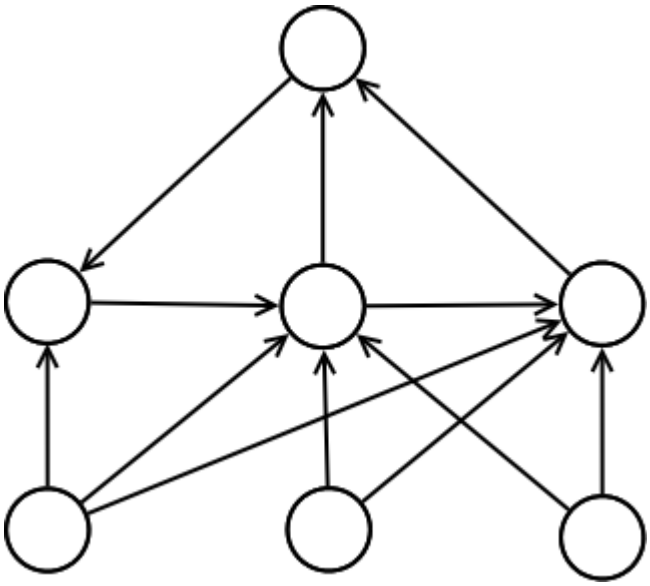
1.
Which of the following neural networks are examples of a feed-forward neural network?

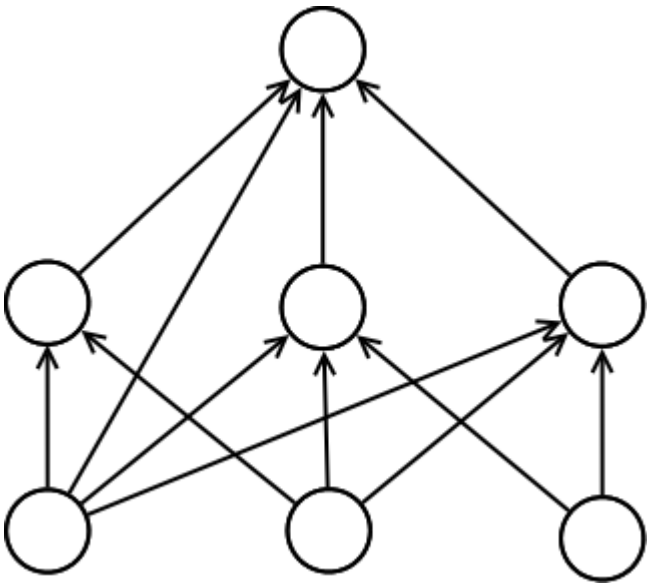




Lecture 3 Quiz

Quiz, 6 questions





1 point

2.

Consider a neural network with only one training case with input $\mathbf{x} = (x_1, x_2, \ldots, x_n)^\top$ and correct output t. There is only one output neuron, which is logistic, i.e. $y = \sigma(\mathbf{w}^\top \mathbf{x})$ (notice that there are no biases). The loss function is squared error. The network has no hidden units, so the inputs are directly connected to the output neuron with weights $\mathbf{w} = (w_1, w_2, \ldots, w_n)^\top$. We're in the process of training the neural network with the backpropagation algorithm. What will the algorithm add to w_i for the next iteration if we use a step size (also known as a learning rate) of ϵ?

- $\epsilon (y - t) y (1 - y) w_i x_i$
- $\epsilon (t - y) y (1 - y) w_i x_i$

$\epsilon\,(t-y)\,x_i$

Lecture 3 Quiz

$\epsilon\,(t-y)\,y\,(1-y)\,x_i$

Quiz, 6 questions

1 point

3.

Suppose we have a set of examples and Brian comes in and duplicates every example, then randomly reorders the examples. We now have twice as many examples, but no more information about the problem than we had before. If we do not remove the duplicate entries, which one of the following methods will *not* be affected by this change, in terms of the computer time (time in seconds, for example) it takes to come close to convergence?

- [x] Online learning, where for every iteration we randomly pick a training case.
- [ ] Full-batch learning.
- [ ] Mini-batch learning, where for every iteration we randomly pick 100 training cases.

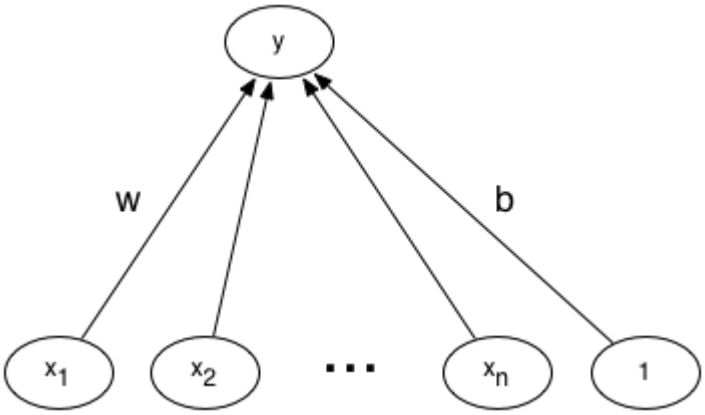
1 point

4.

Lecture 3 Quiz

Quiz, 6 questions

Consider a linear output unit versus a logistic output unit for a feed-forward network with *no hidden layer* shown below. The network has a set of inputs x and an output neuron y connected to the input by weights w and bias b.



We're using the squared error cost function even though the task that we care about, in the end, is binary classification. At training time, the target output values are 1 (for one class) and 0 (for the other class). At test time we will use the classifier to make decisions in the standard way: the class of an input x according to our model **after training** is as follows:

$$\text{class of } x = \begin{cases} 1 \text{ if } w^T x + b \geq 0 \\ 0 \text{ otherwise} \end{cases}$$

Note that we will be training the network using y, but that the decision rule shown above will be the same at *test* time, regardless of the type of output neuron we use for training.

Which of the following statements is true?

- [ ] At the solution that minimizes the error, the learned weights are always the same for both types of units; they only differ in how they get to this solution.
- [x] Unlike a linear unit, using a logistic unit will not penalize is for getting things right too confidently.
- [ ] The error function (the error as a function of the weights) for both types of units will form a quadratic bowl.
- [ ] For a logistic unit, the derivatives of the error function with respect to the weights can have unbounded magnitude, while for a linear unit they will have bounded magnitude.

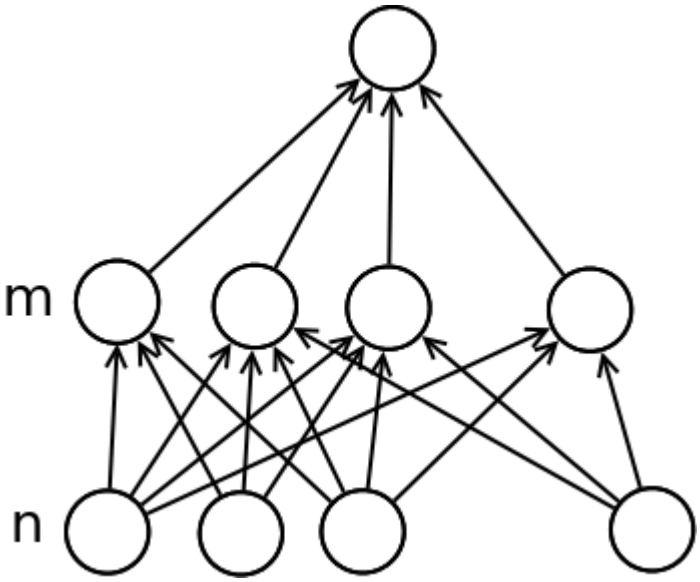
1 point

5.

Lecture 3 Quiz

Quiz, 6 questions

Consider a neural network with one layer of **logistic** hidden units (intended to be fully connected to the input units) and a linear output unit. Suppose there are n input units and m hidden units. Which of the following statements are true? Check all that apply.



- [x] A network with $m > n$ has more learnable parameters than a network without any hidden layers (with the same inputs).
- [ ] As long as $m \geq 1$, this network can learn to compute any function that can be learned by a network without any hidden layers (with the same inputs).
- [ ] Any function that can be learned by such a network can also be learned by a network without any hidden layers (with the same inputs).
- [x] If $m > n$, this network can learn more functions than if m is less than n (with n being the same).

1 point

6.

Lecture 3 Quiz

Quiz, 6 questions

Brian wants to make his feed-forward network (with no hidden units) using a **linear** output neuron more powerful. He decides to combine the predictions of two networks by averaging them. The first network has weights $\boldsymbol{w_1}$ and the second network has weights $\boldsymbol{w_2}$. The predictions of this network for an example $\boldsymbol{x}$ are therefore:

$$y = \frac{1}{2}w_1^T x + \frac{1}{2}w_2^T x$$

Can we get the exact same predictions as this combination of networks by using a single feed-forward network (again with no hidden units) using a **linear** output neuron and weights $\boldsymbol{w_3} = \frac{1}{2}(w_1 + w_2)$?

- [x] Yes
- [ ] No

I, **Alan Wright**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

Learn more about Coursera's Honor Code

Submit Quiz