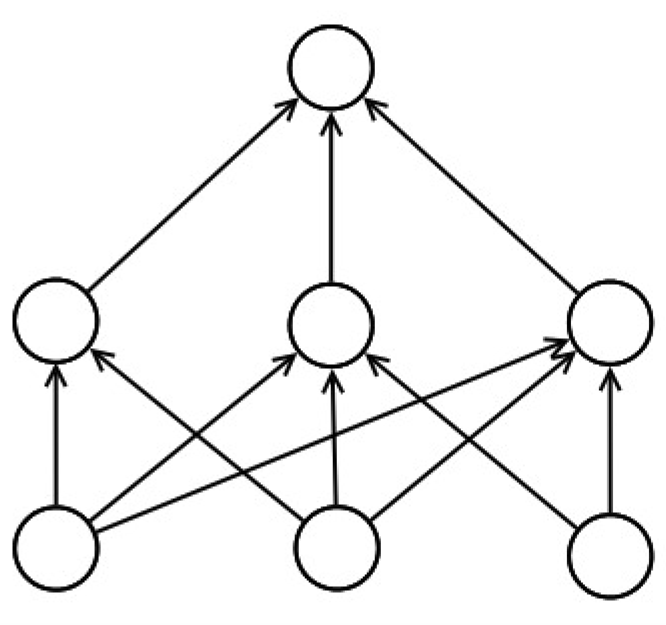
Congratulations! You passed!

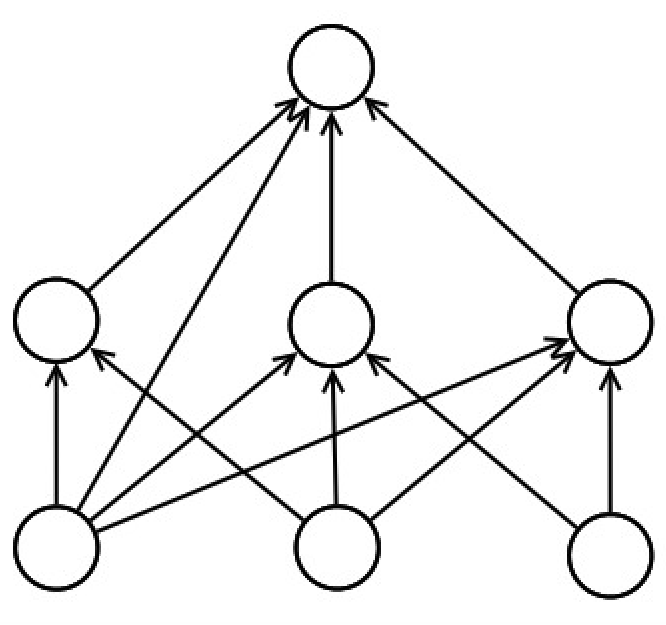
Next Item

1 / 1 points

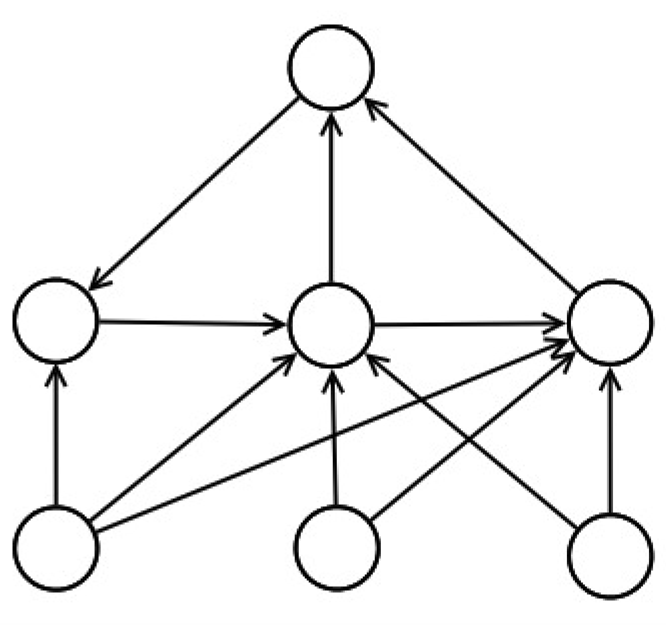
1. Which of the following neural networks are examples of a feed-forward neural network?

- [x] 

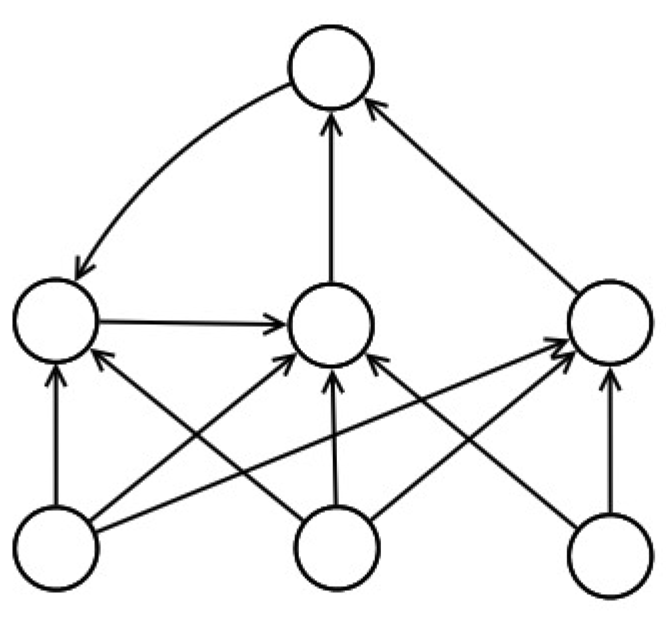
Correct
A feed-forward network does not have cycles.

- [x] 

Correct
A feed-forward network does not have cycles.

- [ ] 

Un-selected is correct

- [ ] 

Un-selected is correct

---

1 / 1 points

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 linear, i.e. $y = \mathbf{w}^\top \mathbf{x}$ (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?

- [ ] x_i

Un-selected is correct

- [x] $\epsilon(t - \mathbf{w}^\top \mathbf{x})x_i$

Correct
There are multiple components to this, all multiplied together: the learning rate, the derivative of the loss function w.r.t. the state of the output unit, and the derivative of the input to the output unit w.r.t. w_i.

- [ ] $\epsilon(\mathbf{w}^\top \mathbf{x} - t)x_i$

Un-selected is correct

- [ ] x_i if $\mathbf{w}^\top \mathbf{x} > t$

  $-x_i$ if $\mathbf{w}^\top \mathbf{x} \leq t$

Un-selected is correct

---

1 / 1 points

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.

Correct
Full-batch learning needs to look at every example before taking a step, therefore each step will be twice as expensive. Online learning only looks at one example at a time so each step has the same computational cost as before. On expectation, online learning would make the same progress after looking at half of the dataset as it would have if Brian has not intervened.

Although this example is a bit contrived, it serves to illustrate how online learning can be advantageous when there is a lot of redundancy in the data.

- [ ] Full-batch learning.

Un-selected is correct

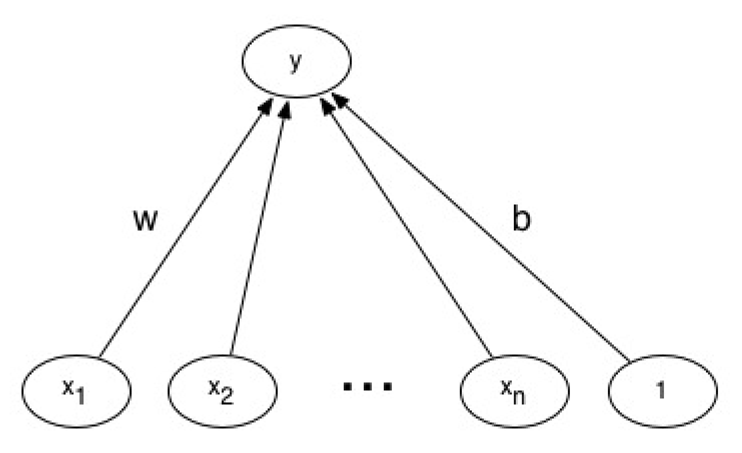
- [ ] Mini-batch learning, where for every iteration we randomly pick 100 training cases.

Un-selected is correct

---

1 / 1 points

4. 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?

- [x] Unlike a linear unit, using a logistic unit will not penalize is for getting things right too confidently.

Correct
If the target is 1 and the prediction is 100, the logistic unit will squash this down to a number very close to 1 and so we will not incur a very high cost. With a linear unit, the difference between the prediction and target will be very large and we will incur a high cost as a result, despite the fact that we get the classification decision correct.

- [ ] 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.

Un-selected is correct

- [ ] The error function (the error as a function of the weights) for both types of units will form a quadratic bowl.

Un-selected is correct

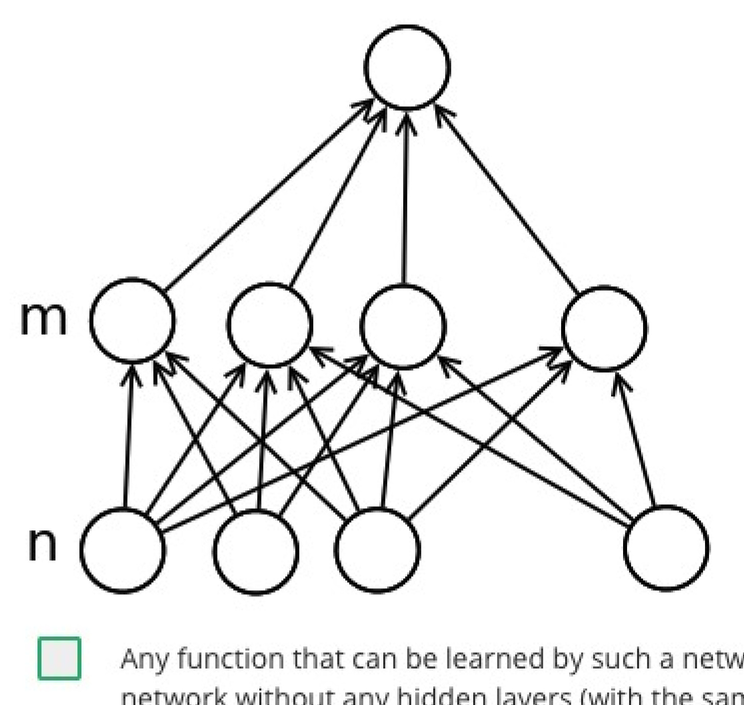
- [ ] For a logistic unit, the derivatives of the error function with respect to the weights can have unbounded magnitude, while for a linear unit they will have bounded magnitude.

Un-selected is correct

---

1 / 1 points

5. 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.



- [ ] 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).

Un-selected is correct

- [ ] 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).

Un-selected is correct

- [x] A network with $m > n$ has more learnable parameters than a network without any hidden layers (with the same inputs).

Correct
The bulk of the learnable parameters is in the connections from the input units to the hidden units. There are $m \cdot n$ learnable parameters there.

- [x] If $m > n$, this network can learn more functions than if m is less than n (with n being the same).

Correct
This is quite a flexible model. It can learn many functions that cannot be learned without the use of a hidden layer. The nonlinearity in the hidden layer is essential.

---

1 / 1 points

6. 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 w_1 and the second network has weights w_2. The predictions of this network for an example 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 $w_3 = \frac{1}{2}(w_1 + w_2)$?

- [x] Yes

Correct

- [ ] No

Un-selected is correct