Computing with Networks

Total points 21

1. Let's design some feedforward networks that can do some basic operations on their inputs. This could mean lowering their intensity, looking for strong changes, or one of many other possibilities. One nice way to build intuition for this sort of processing is to think of these networks as operating on images. Even though our networks will operate over only 5 pixels of image data, we can still build the same basic operations that we would for a regular image. **2 points**

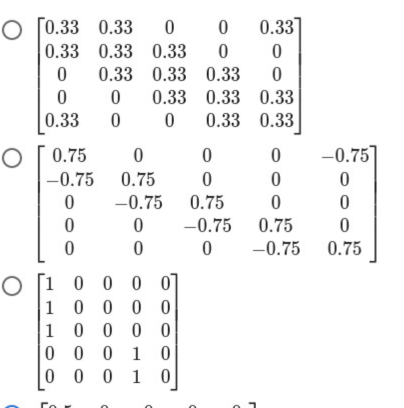
For the next four questions, we will start with the following image as input:

Suppose we processed the image and it looked like this:

Which of the following weight matrices W would give us a feedforward network most closely approximating this image processing operation?

- ( ) $\begin{bmatrix} 0.75 & 0 & 0 & 0 & -0.75 \\ -0.75 & 0.75 & 0 & 0 & 0 \\ 0 & -0.75 & 0.75 & 0 & 0 \\ 0 & 0 & -0.75 & 0.75 & 0 \\ 0 & 0 & 0 & -0.75 & 0.75 \end{bmatrix}$
- (•) $\begin{bmatrix} 0.33 & 0.33 & 0 & 0 & 0.33 \\ 0.33 & 0.33 & 0.33 & 0 & 0 \\ 0 & 0.33 & 0.33 & 0.33 & 0 \\ 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0.33 & 0 & 0 & 0.33 & 0.33 \end{bmatrix}$
- ( ) $\begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}$
- ( ) $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$

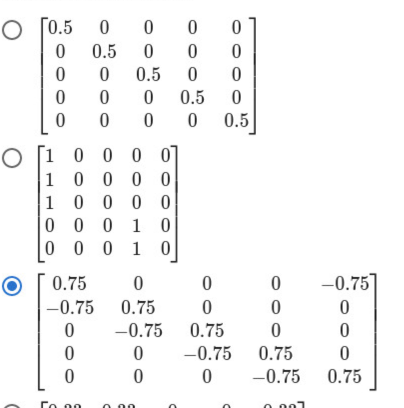
2. Suppose we processed the image and it looked like this: **2 points**



Which of the following weight matrices W would give us a feedforward network most closely approximating this image processing operation?

- ( ) $\begin{bmatrix} 0.33 & 0.33 & 0 & 0 & 0.33 \\ 0.33 & 0.33 & 0.33 & 0 & 0 \\ 0 & 0.33 & 0.33 & 0.33 & 0 \\ 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0.33 & 0 & 0 & 0.33 & 0.33 \end{bmatrix}$
- ( ) $\begin{bmatrix} 0.75 & 0 & 0 & 0 & -0.75 \\ -0.75 & 0.75 & 0 & 0 & 0 \\ 0 & -0.75 & 0.75 & 0 & 0 \\ 0 & 0 & -0.75 & 0.75 & 0 \\ 0 & 0 & 0 & -0.75 & 0.75 \end{bmatrix}$
- ( ) $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
- (•) $\begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}$

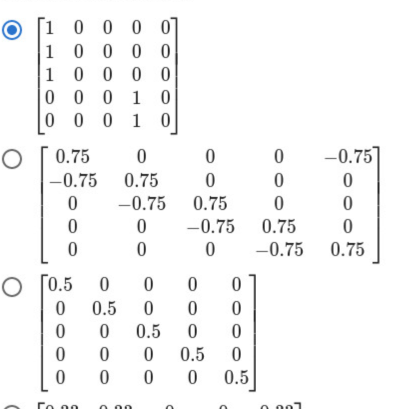
3. Suppose we processed the image and it looked like this: **2 points**



Which of the following weight matrices W would give us a feedforward network most closely approximating this image processing operation?

- ( ) $\begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}$
- ( ) $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
- (•) $\begin{bmatrix} 0.75 & 0 & 0 & 0 & -0.75 \\ -0.75 & 0.75 & 0 & 0 & 0 \\ 0 & -0.75 & 0.75 & 0 & 0 \\ 0 & 0 & -0.75 & 0.75 & 0 \\ 0 & 0 & 0 & -0.75 & 0.75 \end{bmatrix}$
- ( ) $\begin{bmatrix} 0.33 & 0.33 & 0 & 0 & 0.33 \\ 0.33 & 0.33 & 0.33 & 0 & 0 \\ 0 & 0.33 & 0.33 & 0.33 & 0 \\ 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0.33 & 0 & 0 & 0.33 & 0.33 \end{bmatrix}$

4. Suppose we processed the image and it looked like this: **2 points**



Which of the following weight matrices W would give us a feedforward network most closely approximating this image processing operation?

- (•) $\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$
- ( ) $\begin{bmatrix} 0.75 & 0 & 0 & 0 & -0.75 \\ -0.75 & 0.75 & 0 & 0 & 0 \\ 0 & -0.75 & 0.75 & 0 & 0 \\ 0 & 0 & -0.75 & 0.75 & 0 \\ 0 & 0 & 0 & -0.75 & 0.75 \end{bmatrix}$
- ( ) $\begin{bmatrix} 0.5 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}$
- ( ) $\begin{bmatrix} 0.33 & 0.33 & 0 & 0 & 0.33 \\ 0.33 & 0.33 & 0.33 & 0 & 0 \\ 0 & 0.33 & 0.33 & 0.33 & 0 \\ 0 & 0 & 0.33 & 0.33 & 0.33 \\ 0.33 & 0 & 0 & 0.33 & 0.33 \end{bmatrix}$

5. In lecture 6.2, we encountered a process of conceptual abstraction, taking us from modeling individual neurons to modeling whole networks. By the middle of the lecture, we had abstracted away many of the interesting time dynamics of feedforward neural networks and arrived at a simple equation: **3 points**

$\mathbf{v_{ss}} = F(W\mathbf{u})$

We have to make a number of assumptions to get to this equation. Necessarily we lose some interesting information when we make these assumptions from the beginning of lecture 6.2 to the point where we first see this equation (roughly 10 min in). Which of the following holds true? (Check all that are true)

- [x] This model shows us the outputs the system will converge to, but no longer describes the time dynamics of that convergence.
- [ ] This model ignores the firing rates of the output cells.
- [ ] This model ignores the firing rates of the input cells.
- [x] This model cannot adequately represent many of the dynamics of individual cells, such as the effect of the refractory period.
- [ ] This model does not allow differences in firing rates between different inputs.
- [ ] This model does not allow for synaptic connections where an input cell is inhibiting the firing of the output cell.
- [ ] Since this model does not use a detailed description of individual cells, it does not account for the strength of individual synaptic connections.
- [x] This model ignores patterns in the input and output spike timings, opting instead to simply look at firing rates.
- [x] This model ignores the effects of synchrony and correlation between the input neurons.
- [x] This model assumes static, unchanging inputs, so it does not account for the dynamics resulting from constantly changing inputs.

6. The next three questions utilize the following code to model an integrate-and-fire neuron receiving input spikes through an alpha synapse: **5 points**

Matlab:

alpha_neuron
M File

Python:

alpha_neuron
PY File

(NOTE ON DOWNLOADING CODE AND DATA: Currently, downloaded files are automatically renamed to begin with a long string of random characters (we hope to have this fixed soon). Sometimes the file type is also changed. In order to ensure that all of the files in the quizzes work correctly, make sure that after downloading each file you rename it to the file name shown in the original quiz question. If you still have trouble getting any of the files to open feel free to search or inquire on the class Discussion Forums.)

The parameter "t_{peak}" controls when the alpha function peaks after an input spike occurs (and hence how long the effects of an input spike linger on in the postsynaptic neuron). "t_{peak}" for excitatory synapses in the brain may vary from 0.5 ms (AMPA or non-NMDA) to 40 ms (NMDA synapse).

Vary the value of t_{peak} from 0.5 ms to 10 ms in steps of 0.5 ms and observe how this influences the output of the neuron for the fixed input spike train used in this code.

Plot the output spike count as a function of t_{peak} for the given input spike train.

Which of the following answers best describes the relationship between the value of t_{peak} and the firing rate of the neuron?

- (•) The neuron's firing rate increases sublinearly as a function of the value of t_{peak}.
- ( ) The neuron's firing rate increases linearly as a function of the value of t_{peak}.
- ( ) The neuron's firing rate increases superlinearly as a function of the value of t_{peak}.
- ( ) The neuron's firing rate decreases sublinearly as a function of the value of t_{peak}.
- ( ) The neuron's firing rate decreases linearly as a function of the value of t_{peak}.
- ( ) The neuron's firing rate decreases superlinearly as a function of the value of t_{peak}.

7. Continued from Question 6: **2 points**

Which of the following explanations best explains how the value of t_{peak} influences the firing rate of the neuron?

- (•) Increasing the value of t_{peak} decreases the decay rate of the synaptic conductance, allowing more charge to flow per spike and increasing the extent of the summation between input spikes.
- ( ) Increasing the value of t_{peak} increases the decay rate of the synaptic conductance, allowing less charge to flow per spike and decreasing the extent of the summation between input spikes.
- ( ) Increasing the value of t_{peak} decreases the decay rate of the neuron's voltage, increasing the extent of the summation between the neuron's voltage response to spikes.
- ( ) Increasing the value of t_{peak} increases the decay rate of the neuron's voltage, decreasing the extent of the summation between the neuron's voltage response to spikes.

8. Continued from Question 6: **2 points**

How would you turn this synapse into an inhibitory synapse? (Check all that apply)

- [ ] Changing the value of E_{syn} to be greater than 0.
- [ ] Changing the value of E_{syn} to be less than 0.
- [ ] Changing the value of E_{syn} to be less than that of V_{th}.
- [x] Changing the value of E_{syn} to be less than that of E_{leak}.

9. (Question variation 4: Different variations have different numbers) Suppose that we had a linear recurrent network of 5 input nodes and 5 output nodes. Let us say that our network's weight matrix W is: **1 point**

$$W = \begin{bmatrix} 0.6 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.6 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.6 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.6 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.6 \end{bmatrix}$$

Suppose that we have a static input vector $\mathbf{u}$:

$$\mathbf{u} = \begin{bmatrix} 0.6 \\ 0.5 \\ 0.6 \\ 0.2 \\ 0.1 \end{bmatrix}$$

Finally, suppose that we have a recurrent weight matrix M:

$$M = \begin{bmatrix} -0.5 & 0 & 0.5 & 0.5 & 0 \\ 0 & -0.5 & 0 & 0.5 & 0.5 \\ 0.5 & 0 & -0.5 & 0 & 0.5 \\ 0.5 & 0.5 & 0.0 & -0.5 & 0 \\ 0 & 0.5 & 0.5 & 0 & -0.5 \end{bmatrix}$$

Which of the following is the steady state output $\mathbf{v_{ss}}$ of the network?

(Hint: See the lecture on recurrent networks, and consider writing some Matlab/Octave/Python code to handle the eigenvectors/values (you may use the "eig" function in Matlab/Octave or "np.linalg.eig" in Python))

- ( ) $\mathbf{v_{ss}} = \begin{bmatrix} 0.873 \\ 0.791 \\ 0.864 \\ 0.755 \\ 0.718 \end{bmatrix}$
- ( ) $\mathbf{v_{ss}} = \begin{bmatrix} 0.547 \\ 0.480 \\ 0.543 \\ 0.381 \\ 0.336 \end{bmatrix}$
- (•) $\mathbf{v_{ss}} = \begin{bmatrix} 0.616 \\ 0.540 \\ 0.609 \\ 0.471 \\ 0.430 \end{bmatrix}$
- ( ) $\mathbf{v_{ss}} = \begin{bmatrix} 1.67 \\ 1.58 \\ 1.66 \\ 1.56 \\ 1.53 \end{bmatrix}$

Coursera Honor Code Learn more

- [x] I understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

Daniel Engbert

Use the name on your government issued ID

Submit | Save draft

Like | Dislike | Report an issue