EECS 16A Fall 2019 Designing Information Devices and Systems I Discussion 15A

1. Least Squares Projections in OMP

OMP picks out signatures that are present in the device (or columns of the matrix) one by one. Some students were wondering why we cannot just project the error vector onto the columns one by one as we pick them out. This problem will illustrate why the joint projection onto the span of all vectors that are present is the right thing to do.

Suppose we have the following system. In practice of course, one would never use OMP for such a small size problem with a square matrix, but this is for illustration purposes. Also, the "$\approx$" symbol means that the system may or may not have an exact solution.

$$\mathbf{M}\vec{x} \approx \vec{b} \tag{1}$$

$$\begin{bmatrix} | & | \\ \vec{m}_1 & \vec{m}_2 \\ | & | \end{bmatrix} \vec{x} \approx \vec{b} \tag{2}$$

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \approx \begin{bmatrix} 2 \\ 1 \end{bmatrix} \tag{3}$$

(a) Find $\text{proj}_{\vec{m}_1}(\vec{b})$.

Answer:

$$\text{proj}_{\vec{m}_1}(\vec{b}) = \frac{\langle \vec{m}_1, \vec{b} \rangle}{||\vec{m}_1||^2} \vec{m}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

(b) Find $\text{proj}_{\vec{m}_2}(\vec{b})$.

Answer:

$$\text{proj}_{\vec{m}_2}(\vec{b}) = \frac{\langle \vec{m}_2, \vec{b} \rangle}{||\vec{m}_2||^2} \vec{m}_2 = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}$$

(c) Find $\text{proj}_{\vec{m}_1}(\vec{b}) + \text{proj}_{\vec{m}_2}(\vec{b})$.

Answer:

$$\text{proj}_{\vec{m}_1}(\vec{b}) + \text{proj}_{\vec{m}_2}(\vec{b}) = \begin{bmatrix} 3.5 \\ 1.5 \end{bmatrix}$$

We expect to get $\vec{b}$ back after the projection because $\vec{b} \in \text{col}(\mathbf{M})$, but that is not what we calculated.

(d) Find the least squares projection of $\vec{b}$ onto $\text{col}(\mathbf{M})$. Is it equal to part (c)? Why?

Answer:

$$\mathbf{M}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\vec{b} = \vec{b} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$\mathbf{M}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\vec{b} \neq \text{proj}_{\vec{m}_1}(\vec{b}) + \text{proj}_{\vec{m}_2}(\vec{b})$ because the columns of $\mathbf{M}$ are not orthogonal.

(e) Now let us try to find $\hat{\vec{x}}$ using the OMP algorithm. The OMP procedure is included below.

Inputs:

- A matrix $\mathbf{M}$, whose columns, $\vec{m}_i$, make up a set of vectors, $\{\vec{m}_i\}$, each of length n
- A vector $\vec{y}$ of length n
- The sparsity level k of the signal

Outputs:

- A vector $\vec{x}$, that contains k non-zero entries.
- A error vector $\vec{e} = \vec{y} - \mathbf{M}\vec{x}$

Procedure:

- Initialize the following values: $\vec{e} = \vec{y}$, $j = 1$, k, $\mathbf{A} = \begin{bmatrix} \quad \end{bmatrix}$
- while ($j \leq k$):
  i. Compute the inner product for each vector in the set, $\vec{m}_i$, with $\vec{e}$: $c_i = \langle \vec{m}_i, \vec{e} \rangle$.
  ii. Column concatenate matrix $\mathbf{A}$ with the column vector that had the maximum inner product value with $\vec{e}$, c_i: $\mathbf{A} = \begin{bmatrix} \mathbf{A} & \mid & \vec{m}_i \end{bmatrix}$
  iii. Use least squares to compute $\vec{x}$ given the $\mathbf{A}$ for this iteration: $\vec{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\vec{y}$
  iv. Update the error vector: $\vec{e} = \vec{y} - \mathbf{A}\vec{x}$
  v. Update the counter: $j = j + 1$

Answer:

i. Calculate the column that has the largest inner product with $\vec{b}$.
$\langle \vec{m}_1, \vec{b} \rangle = 2$ and $\langle \vec{m}_2, \vec{b} \rangle = 3$, so $\vec{m}_2$ has the largest inner product with $\vec{b}$. Update $\mathbf{A} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.

ii. Find $\text{proj}_{\vec{m}_2}(\vec{b})$.
As calculated above, $\text{proj}_{\vec{m}_2}(\vec{b}) = \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}$.

iii. Obtain the error vector $\vec{e}_1 = \vec{b} - \text{proj}_{\vec{m}_2}(\vec{b})$.
$\vec{e}_1 = \vec{b} - \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix} = \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}$

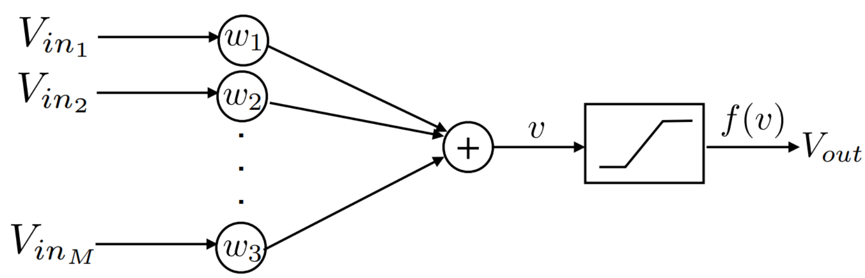
iv. Calculate the column that has the largest inner product with $\vec{e}_1$.
$\langle \vec{m}_1, \vec{e}_1 \rangle = 0.5$ and $\langle \vec{m}_2, \vec{e}_1 \rangle = 0$, so $\vec{m}_1$ has the largest inner product with $\vec{e}_1$. Note that $\vec{e}_1$ is orthogonal to $\vec{m}_2$. Update $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$.

v. **Note:** At this point we have found two columns that explain $\vec{b}$ the most. We want the error vector in the second iteration $\vec{e}_2$ to be orthogonal to **both** columns. We find the least squares projection of $\vec{b}$ onto col($\mathbf{M}$) and then subtract it from $\vec{e}_2$.
$\vec{e}_2 = \vec{b} - \mathbf{M}(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\vec{b} = \vec{b} - \mathbf{M}\hat{\vec{x}} = \vec{b} - \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
We see that $\langle \vec{e}_2, \vec{m}_1 \rangle = 0$ and $\langle \vec{e}_2, \vec{m}_2 \rangle = 0$, so we have found $\hat{\vec{x}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$.
(Aside: The following demonstrates an **incorrect** procedure. We find $\text{proj}_{\vec{m}_1}(\vec{b}) = \frac{\langle \vec{m}_1, \vec{b} \rangle}{||\vec{m}_1||^2}\vec{m}_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$.
Then we find $\vec{e}_2 = \vec{e}_1 - \text{proj}_{\vec{m}_1}(\vec{b}) = \vec{b} - \text{proj}_{\vec{m}_2}(\vec{b}) - \text{proj}_{\vec{m}_1}(\vec{b})$.
$\vec{e}_2 = \vec{b} - \text{proj}_{\vec{m}_2}(\vec{b}) - \text{proj}_{\vec{m}_1}(\vec{b}) = \vec{b} - \frac{\langle \vec{m}_2, \vec{b} \rangle}{||\vec{m}_2||^2}\vec{m}_2 - \frac{\langle \vec{m}_1, \vec{b} \rangle}{||\vec{m}_1||^2}\vec{m}_1 = \begin{bmatrix} -1.5 \\ -0.5 \end{bmatrix}$

We see that $\langle \vec{e}_2, \vec{m}_1 \rangle \neq 0$ and $\langle \vec{e}_2, \vec{m}_2 \rangle \neq 0$. This means that we have not found the closest vector in the span of $\vec{m}_1$ and $\vec{m}_2$ to $\vec{b}$!)

2. Brain-on-a-Chip with 16A Neurons (Spring 2017 Final)

Neurelic Inc, is a hot new startup building chips that emulate some of the brain functions (for example associative memory). As an intern, fresh out of 16A you get to implement the neural network circuits on this chip. The neural network consists of neurons that consist of the following blocks shown on the figure below.

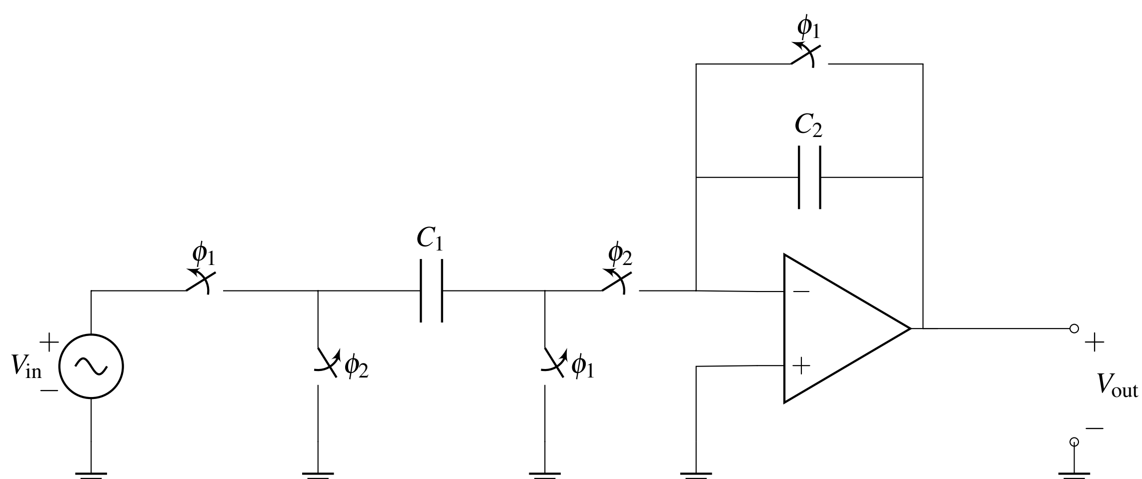


Input signals V_{in_i} are voltages from other neurons, which are multiplied by a constant weight w_i in each synapse and summed in the neuron. Each neuron also contains a nonlinear function (called a sigmoid) which is defined as

$$f(v) = \begin{cases} -1, & v \leq -1 \\ v, & -1 < v < 1 \\ +1, & v \geq +1 \end{cases}$$

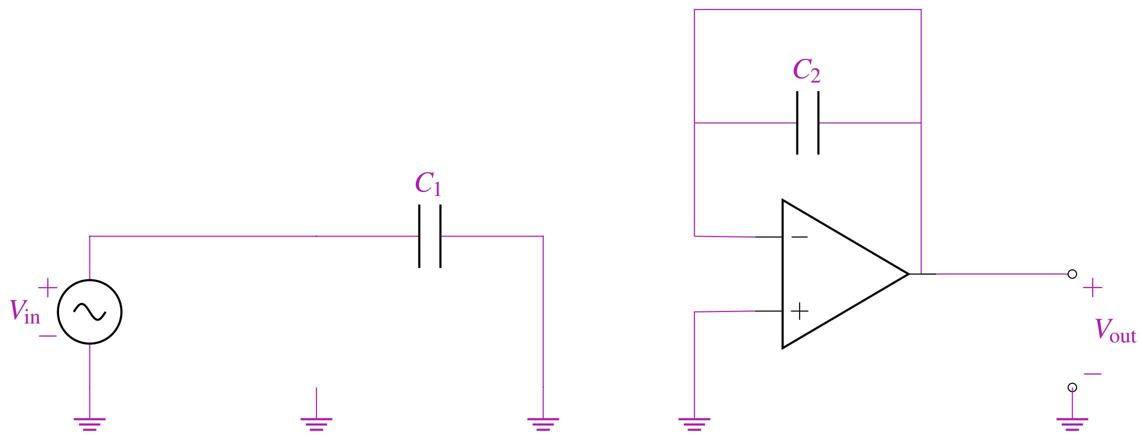
where v is the internal neuron voltage after the synapse summer and $f(v)$ is the neuron voltage output.

(a) Your mentor suggests that you warm-up first by analyzing the circuit below to use as neuron with a single synapse. ϕ_1 and ϕ_2 are non-overlapping clock phases that control the circuit switches.



i. Draw an equivalent circuit during ϕ_1 and write an expression for V_{out} as a function of V_{in}, C_1 and C_2.

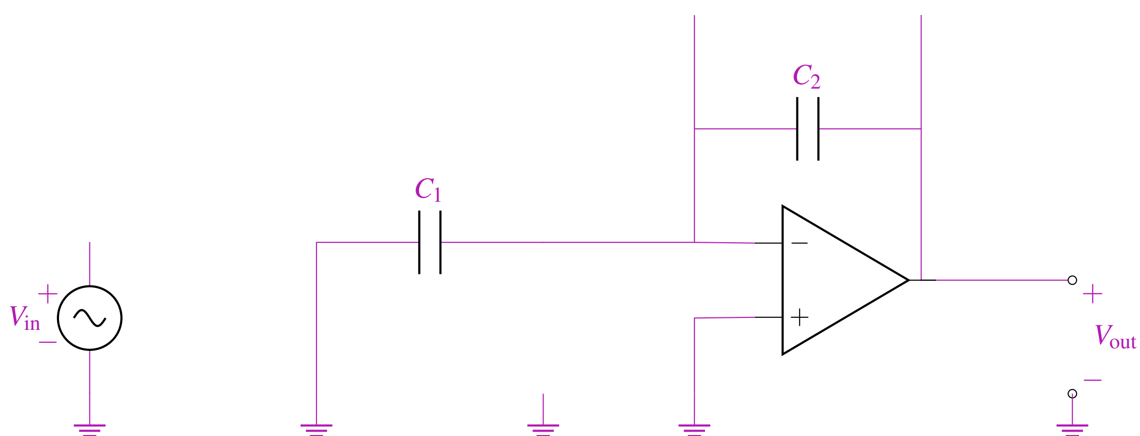
Answer: $V_{out} = 0$



ii. Draw an equivalent circuit during ϕ_2 and write an expression for V_{out} as a function of V_{in}, C_1 and C_2.

Answer:

$$V_{\text{in}}C_1 = V_{\text{out}}C_2$$
$$V_{\text{out}} = \frac{C_1}{C_2}V_{\text{in}}$$

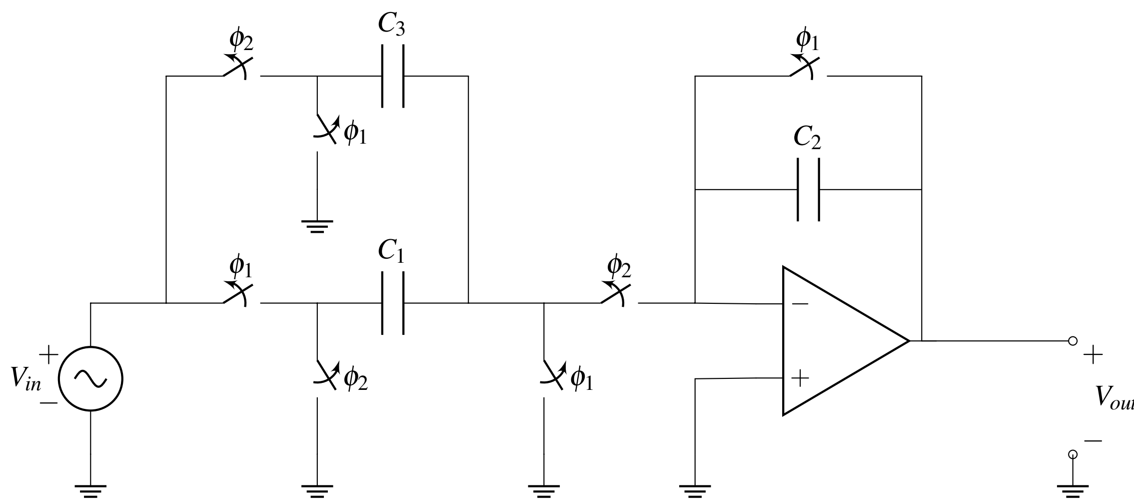


(b) **(Out of scope for this semester. Questions involving op amp supply voltages will not be in the final.)** Write an equation for V_{out} during ϕ_2 as a function of V_{in} for $C_1 = C_2$ and op-amp supply voltages of $\pm 1\,\text{V}$. Briefly explain how this circuit implements the sigmoid function.

Answer: From part (a)(ii) we know $V_{\text{out}} = \frac{C_1}{C_2}V_{\text{in}}$.Setting $C_1 = C_2$, we find $V_{\text{out}} = V_{\text{in}}$. Because of the rails of the op amp, once the V_{in} exceeds $1\,\text{V}$, the output will be $1\,\text{V}$. From this, we see the circuit implements the sigmoid function:

$$V_{\text{out}} = \begin{cases} -1, & V_{\text{in}} \leq -1 \\ V_{\text{in}}, & -1 < V_{\text{in}} < 1 \\ +1, & V_{\text{in}} \geq +1 \end{cases}$$

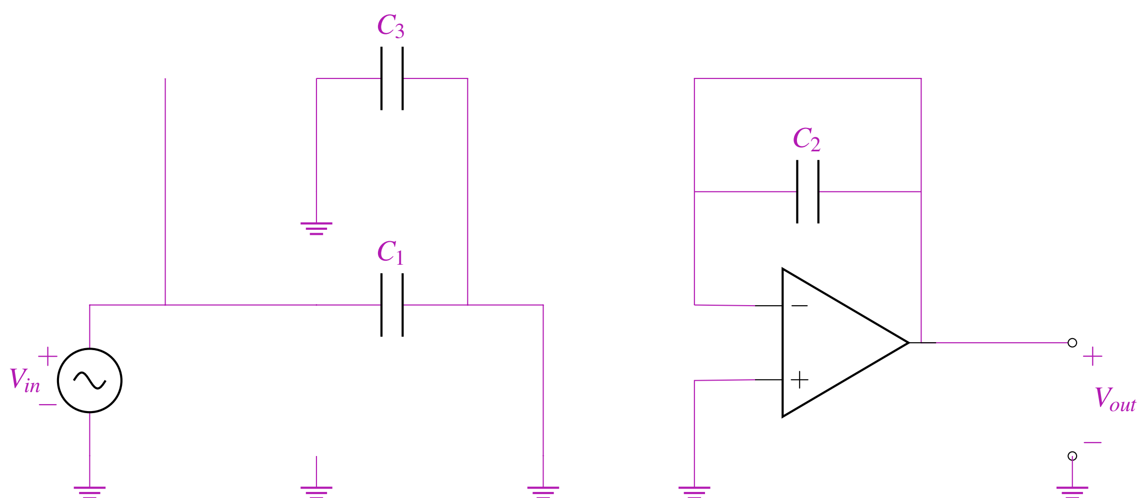
(c) Then, your mentor shows you the following neuron circuit, which can realize both positive and negative synapse weight and create $V_{out} = w_1 V_{in}$ in ϕ_2.



i. Draw an equivalent circuit during ϕ_1 and write an expression for V_{out} as a function of V_{in}, C_1, C_2, and C_3.

Answer:

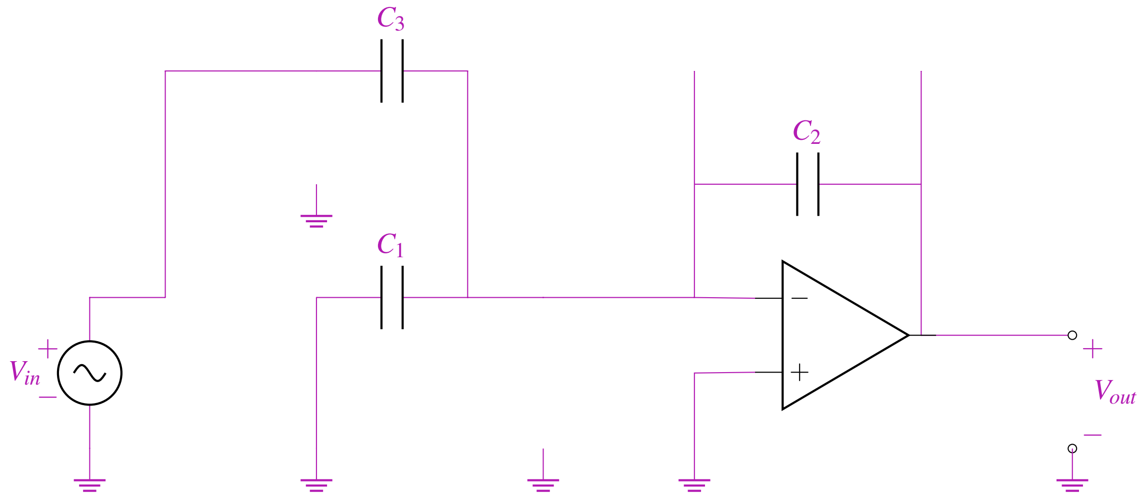
In phase 1, $V_{\text{out}} = 0$.



ii. Draw an equivalent circuit during ϕ_2 and write an expression for V_{out} as a function of V_{in}, C_1, C_2, and C_3.

Answer:

In phase 2, $V_{\text{out}} = \frac{C_1 - C_3}{C_2} V_{\text{in}}$.



(d) Now it is your turn to implement a neuron that realizes the following function $V_{out} = w_1 V_{in_1} + w_2 V_{in_2}$. Draw the circuit, such that $w_1 = 1/2$ and $w_2 = -1/4$. Label all circuit elements appropriately. You should use a single op-amp and as many capacitors and switches as you need. All capacitors must be of size C_{unit}. Assume that the op-amp power supplies are ± 1V (no need to draw them in the circuit). The circuit should operate in 2 phases, with $V_{out} = w_1 V_{in_1} + w_2 V_{in_2}$ in the second phase (ϕ_2), and reset in ϕ_1.

Answer: