# EECS 16A Spring 2020 — Designing Information Devices and Systems I — Discussion 14A

## 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 "≈" symbol means that the system may or may not have an exact solution.

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

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

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

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

**Answer:**

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

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

**Answer:**

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

(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 $\vec{e}$, $c_i$: $\mathbf{A} = \begin{bmatrix} \mathbf{A} & | & \vec{m}_i \end{bmatrix}$
  iii. Update the error vector: $\vec{e} = \vec{y} - \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\vec{y}$
  iv. Update the counter: $j = j + 1$

At the end of the procedure, we have the matrix $\mathbf{A} = [\vec{m}_{i_1}, \ldots, \vec{m}_{i_k}]$. We return our solution vector with nonzero entries $i_1, \ldots, i_k$ by defining its nonzero entries $x_{i_1}, \ldots, x_{i_k}$ according to $[x_{i_1}, \ldots, x_{i_k}]^T = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\vec{y}$.

**Answer:**

i. Calculate the column that has the largest inner product with $\vec{b}$.

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

ii. Obtain the error vector $\vec{e}_1 = \vec{b} - \text{proj}_{\vec{m}_2}(\vec{b})$.

$\vec{e}_1 = \vec{b} - \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix} = \begin{bmatrix} 0.5 \\ -0.5 \end{bmatrix}$

iii. 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 $\mathbf{A} = \begin{bmatrix} \frac{\sqrt{2}}{2} & 1 \\ \frac{\sqrt{2}}{2} & 0 \end{bmatrix}$.

iv. Again, we update the error vector through least squares.

$\vec{e}_2 = \vec{b} - \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\vec{b} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

We see that the error vector is 0, which means the two vectors perfectly describe $\vec{b}$. To find the solution vector, we apply least squares,

$$\hat{\vec{x}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\vec{b} = \begin{bmatrix} \sqrt{2} \\ 1 \end{bmatrix}$$

Note that the solution vector is defined in the coordinates set by the columns of $\mathbf{A}$. In order to get to the solution vector in terms of the coordinates set by the columns of $\mathbf{M}$, we must map the columns of $\mathbf{A}$ to the columns of $\mathbf{M}$, which in this case they are simply the reverse of each other. Therefore, the final solution is $\hat{\vec{x}} = \begin{bmatrix} 1 \\ \sqrt{2} \end{bmatrix}$

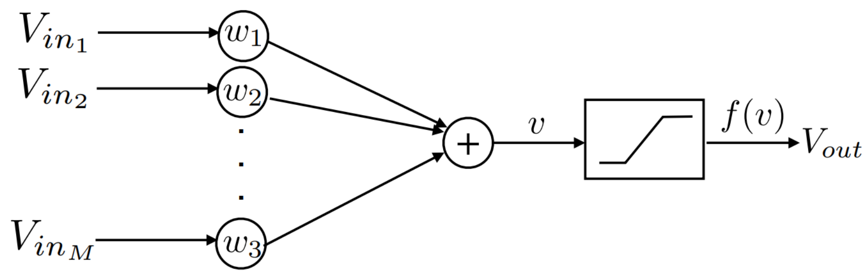
 

**2. (Optional) Orthogonal Matching Pursuit Demo**

Follow along with your discussion TA.

**3. 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.

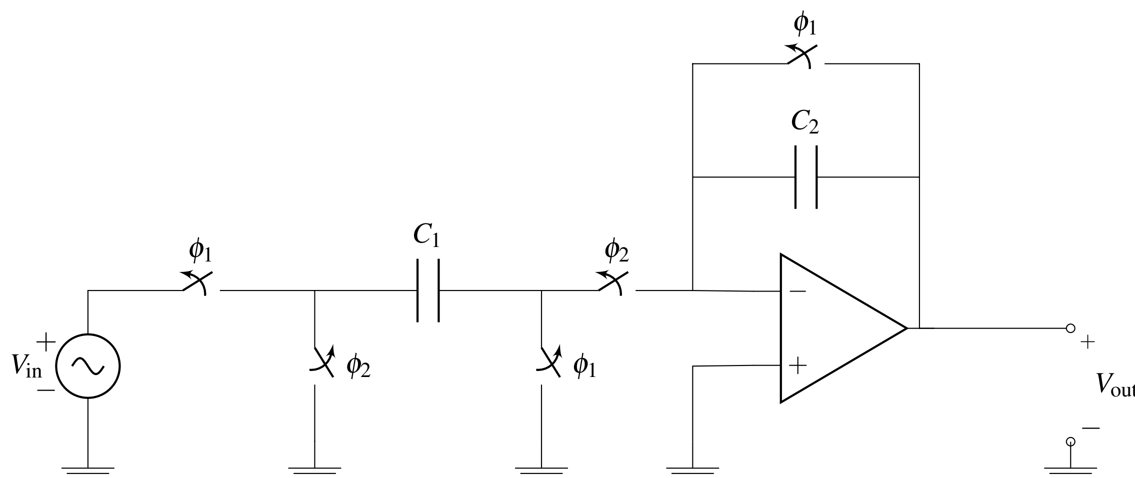



Input signals $V_{in_i}$ are voltages from other neurons, which are multiplied by a constant weight $w_i$ in each synapse and summed in the neuron. Each neuron also contains a nonlinear function (called a sigmoid) which is defined as

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

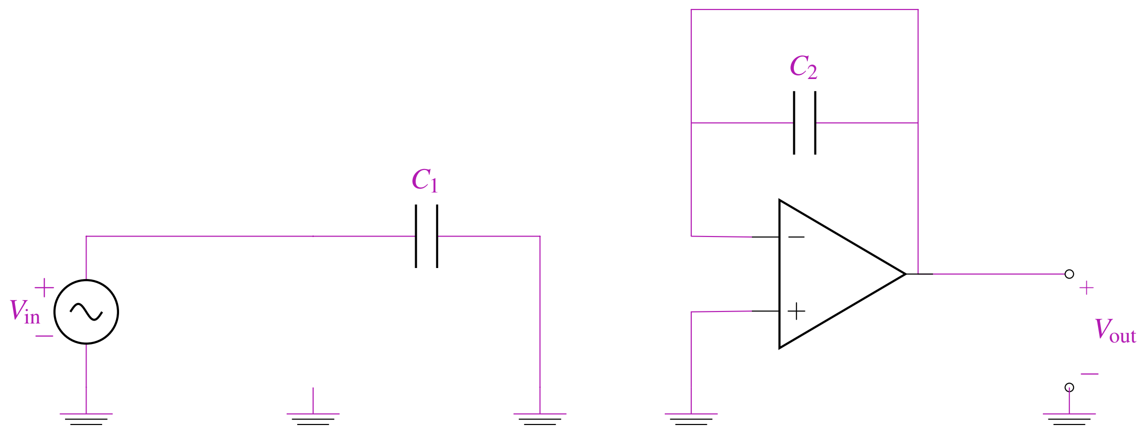
where $v$ is the internal neuron voltage after the synapse summer and $f(v)$ is the neuron voltage output.

(a) Your mentor suggests that you warm-up first by analyzing the circuit below to use as neuron with a single synapse. $\phi_1$ and $\phi_2$ are non-overlapping clock phases that control the circuit switches.




i. Draw an equivalent circuit during $\phi_1$ and write an expression for $V_{\text{out}}$ as a function of $V_{\text{in}}$, $C_1$ and $C_2$.

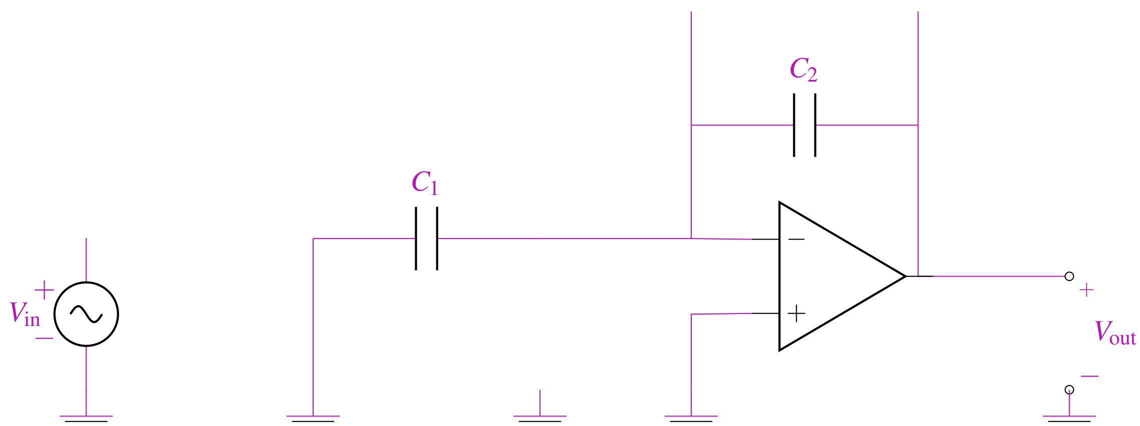
**Answer:** $V_{\text{out}} = 0$




ii. Draw an equivalent circuit during $\phi_2$ and write an expression for $V_{out}$ as a function of $V_{in}$, $C_1$ and $C_2$.

**Answer:**

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

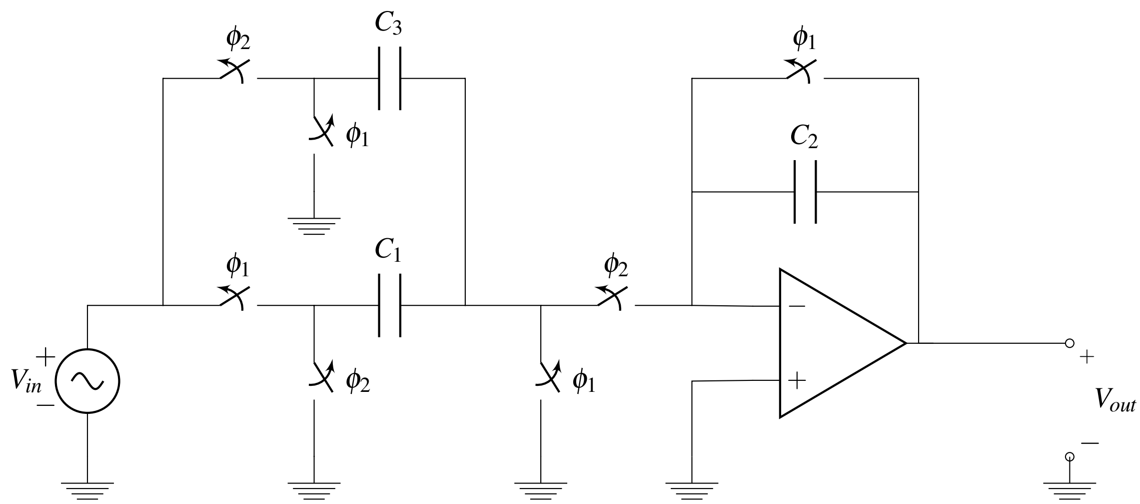



(b) Write an equation for $V_{out}$ during $\phi_2$ as a function of $V_{in}$ for $C_1 = C_2$ and op-amp supply voltages of $\pm 1$ V. Briefly explain how this circuit implements the sigmoid function.

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

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

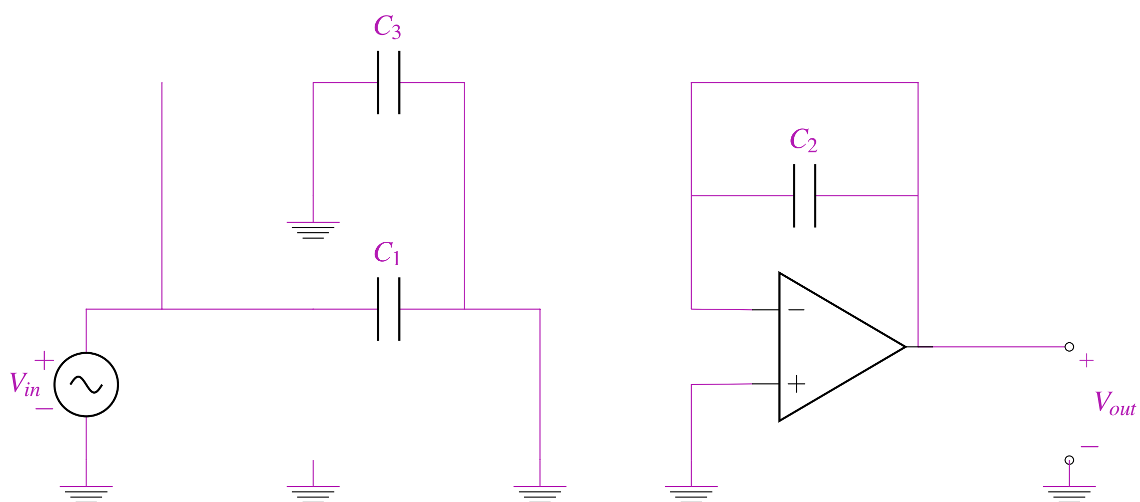
(c) Then, your mentor shows you the following neuron circuit, which can realize both positive and negative synapse weight and create $V_{out} = w_1 V_{in}$ in $\phi_2$.




i. Draw an equivalent circuit during $\phi_1$ and write an expression for $V_{\text{out}}$ as a function of $V_{\text{in}}$, $C_1$, $C_2$, and $C_3$.

**Answer:**

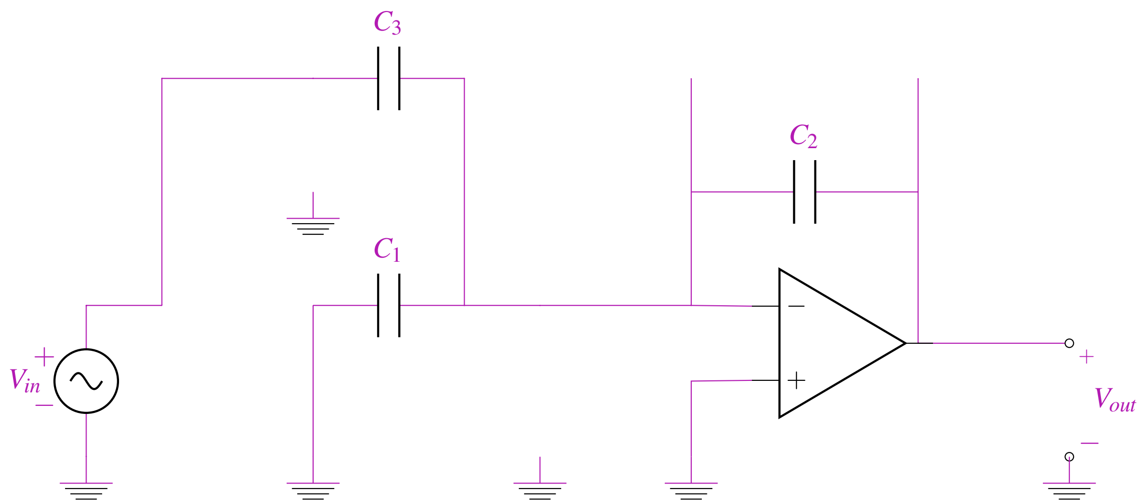
In phase 1, $V_{\text{out}} = 0$.




ii. Draw an equivalent circuit during $\phi_2$ 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 $\pm 1$V (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 ($\phi_2$), and reset in $\phi_1$.

**Answer:**