6.034 Quiz 3

18 November 2015

Name	Kylo Ren
Email	Kyloren@mit.edu

Circle your TA **(for 1 extra credit point)**, so that we can more easily enter your score in our records and return your quiz to you promptly.

Ryan Alexander	**Ben Greenberg**	**Neil Gurram**
Eeway Hsu	**Brittney Johnson**	**Veronica Lane**
Robert Luo	**Jessica Noss**	**Harland Sanders**
Mycal Tucker	**Sarah Vente**	**Jess Wass**

Problem number	Maximum	Score	Grader
1	50		
2	50		
Total	100		

SRN	6		

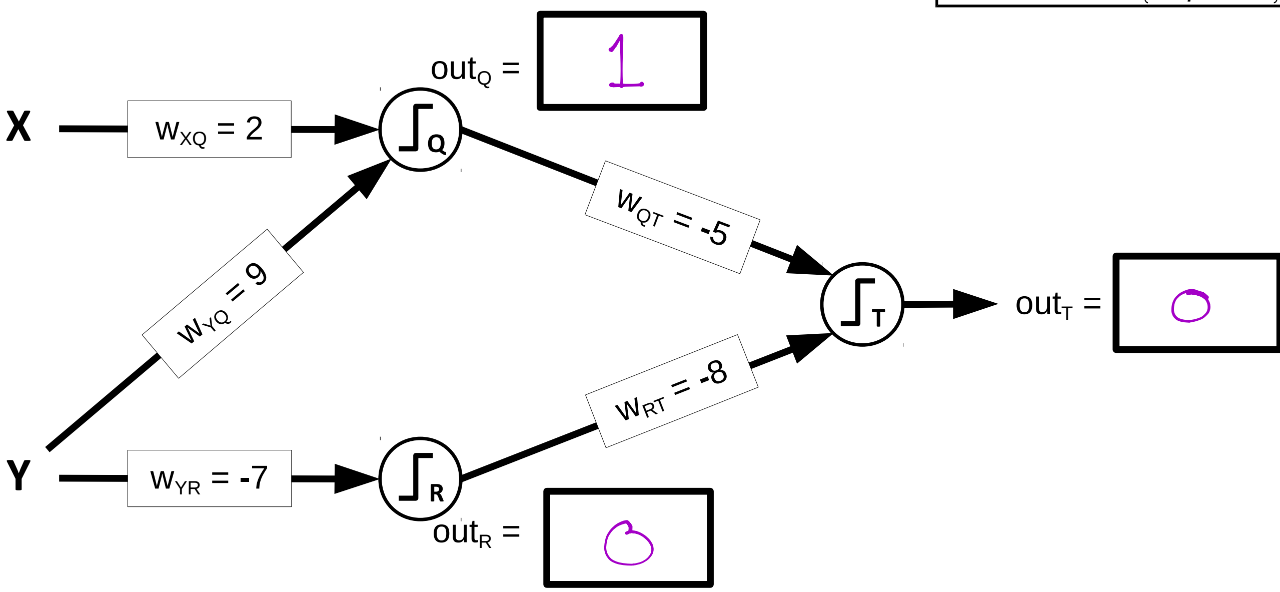
There are 12 pages in this quiz, including this one. As always, open book, open notes, open just about everything, including a calculator, but no computers.

Problem 1: Neural Nets (50 points)

Part A: Forward Propagation (18 points)

A1 (6 points) On the neural net below, perform forward propagation **using the stairstep function with a threshold of 0**, starting from these inputs: **(X = -3, Y= 2)**. Write the output of each neuron (out_Q, out_R, out_T) in the boxes provided.

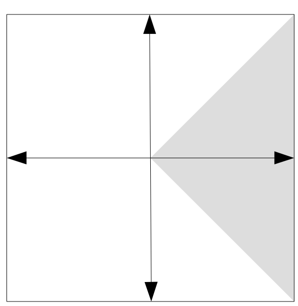
$$stairstep_T(x) = \begin{Bmatrix} 0, if\ x < T \\ 1, if\ x \geq T \end{Bmatrix}$$

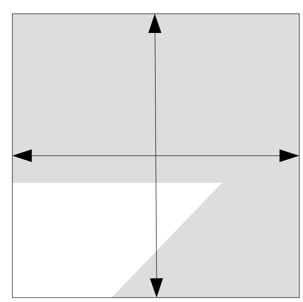


A2 (12 points) By changing the thresholds (not shown) and weights in the neural net above, which of the following pictures could this neural net be trained to draw? Next to each picture, circle YES if the neural net could draw the picture, otherwise circle NO.

YES

NO



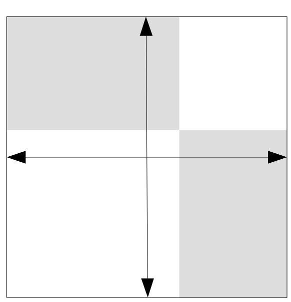


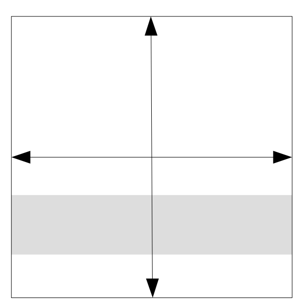
YES

NO

YES

NO





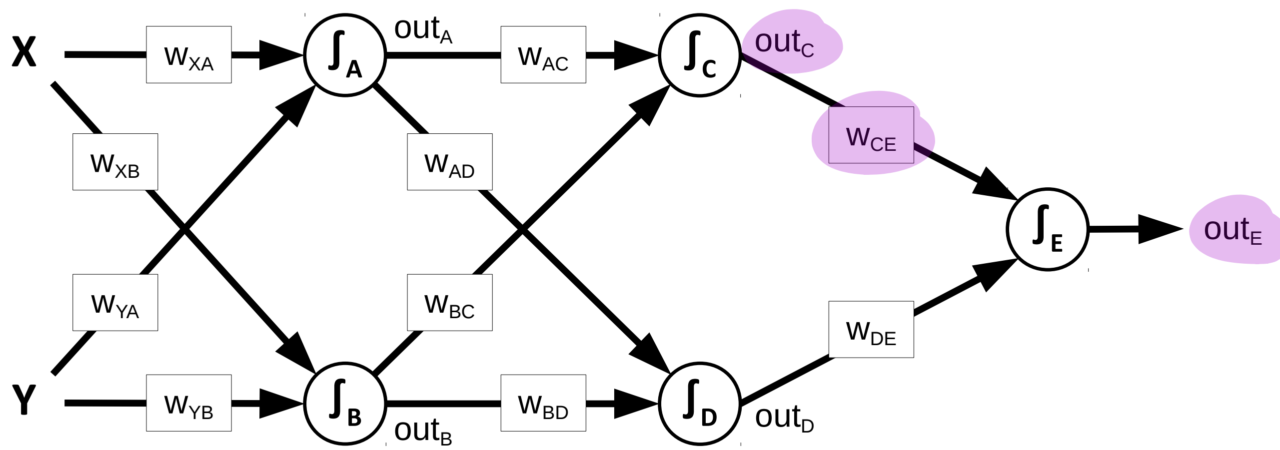
YES

NO

Part B: Backward Propagation (24 points)

Robert is training a neural net. He wants to perform a single step of back propagation.

B1 (7 points) In the neural net below, Robert has computed the output of each neuron using the sigmoid function (for example, out_A is the computed output of neuron A). Now, he will **update the weight w_{CE} going from neuron C to neuron E**. Given that he has already calculated the outputs, which inputs (e.g. Y), weights (e.g. w_{YB}), and neuron outputs (e.g. out_B) does Robert need to use in order to compute the new weight w_{CE}? On the neural net below, **circle each input, weight, and neuron output that is required**.

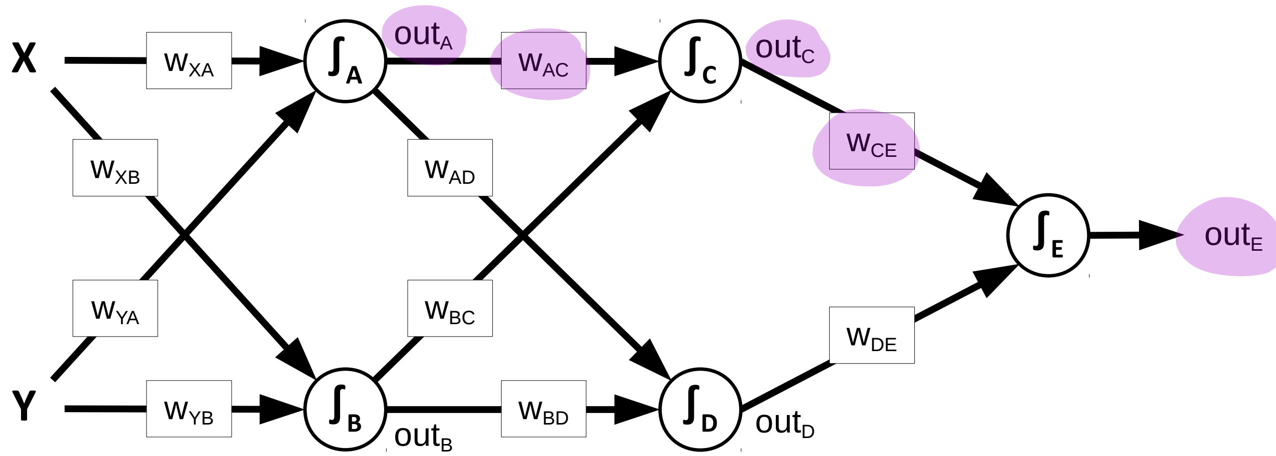


Space provided to show your work (for partial credit)

$$w'_{CE} = r \cdot out_C \cdot \delta_E + w_{CE}$$
$$= r \cdot out_C \cdot (out_E(1-out_E)(out^* - out)) + w_{CE}$$

out^* = desired output
out = overall output

B2 (7 points) Next, Robert will **update the weight w_{AC} going from neuron A to neuron C**. On the neural net below, **circle each input, weight, and neuron output that is required**.



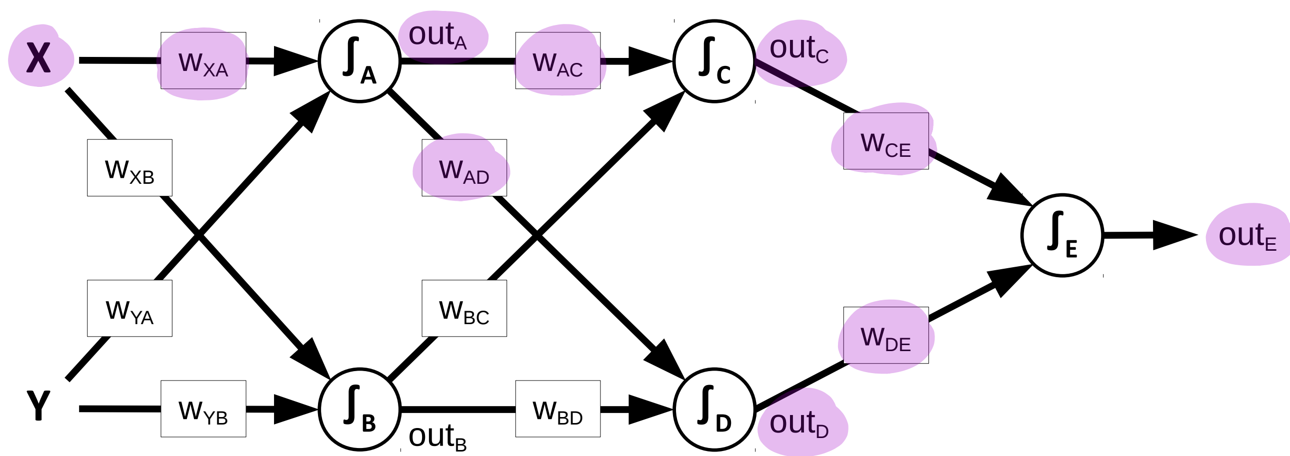
Space provided to show your work (for partial credit)

$$w'_{AC} = w_{AC} + r \cdot out_A \cdot \delta_C$$

$$= w_{AC} + r \cdot out_A \left(out_C \left(1 - out_C\right) \cdot w_{CE} \, \delta_E\right)$$

$$\delta_E = out_E \left(1 - out_E\right)\left(out^* - out\right)$$

B3 (10 points) Finally, Robert will **update the weight w_{XA} going from input X to neuron A**. On the neural net below, **circle each input, weight, and neuron output that is required**.



Space provided to show your work (for partial credit)

$$w'_{XA} = w_{XA} + r \cdot x \cdot \delta_A$$

$$= w_{XA} + r \cdot x \cdot [out_A (1-out_A) \cdot (\delta_C w_{AC} + \delta_D w_{AC})$$

$$= w_{XA} + r \cdot x [out_A (1-out_A) [out_C (1-out_C) w_{CE} \delta_E w_{AC} + out_D (1-out_D) w_{DE} \delta_E w_{AD}]]$$

$$\delta_E = out_E (1-out_E)(out^* - out)$$

Part C: Questions about Backward Propagation (8 points)

C1 (3 points) Brittney trains a neural net using back propagation, starting from some set of initial weights. Jess trains an identical neural net, but starting from a different set of initial weights. They both use the same inputs, desired output, and step size *r*. Assuming that Brittney and Jess will both reach the desired accuracy, are their weights guaranteed to converge to the same values? (Circle the **ONE BEST** answer)

A. Yes, because the performance (accuracy) function is convex and has a global maximum.

B. Yes, if Brittney's performance function gets stuck at a local maximum, then Jess's performance function will get stuck at the same local maximum.

C. No, because back propagation can be subject to overfitting.

D. No, there are multiple possible sets of weights that could produce the desired output for the given inputs.

C2 (5 points) Ben is performing back propagation on a different neural net. Ben's weights have converged on a local maximum, but the neural net's output is still not sufficiently close to the desired value. Which of the following solutions might fix Ben's problem? (Circle **ALL** that apply)

A. Increase the step size *r*. (That is, the current *r* is too small.)

B. Decrease the step size *r*. (That is, the current *r* is too large.)

C. Add more neurons to the neural net.

D. Remove some neurons from the neural net.

E. Use a different definition of “sufficiently close to the desired value.”

D is ambiguous; either answer accepted

Problem 2: Support Vector Machines (50 points)

Part A: Parameters (9 points)

In this part, assume that every set of training data is linearly separable.

A1 (3 points) Ryan trains an SVM on a particular set of training data. If Ryan then adds a new training point to the dataset and retrains the SVM, the **number of support vectors** may: (Circle **ALL** that apply)

INCREASE **DECREASE** **STAY THE SAME**

A2 (3 points) Veronica trains an SVM. Then, Veronica modifies the training data and retrains the SVM. Prof. Winston comments, "If you calculate the magnitude of $\vec{w}$ for your modified problem, you'll find that it has **increased (↑)**." What could Prof. Winston have observed to reach this conclusion? (Circle **ALL** that apply)

A. The gutters became farther apart

B. The gutters became closer together

C. A new training point was added outside the original margin

W↑, margin width↓

D. A new training point was added within the original margin (between the gutters)

E. A training point was removed from the original SVM

A3 (3 points) Eeway trains an SVM. Then, Eeway modifies the training data and retrains the SVM. Prof. Winston comments, "If you calculate the magnitude of $\vec{w}$ for your modified problem, you'll find that it has **decreased (↓)**." What could Prof. Winston have observed to reach this conclusion? (Circle **ALL** that apply)

A. The gutters became farther apart

B. The gutters became closer together

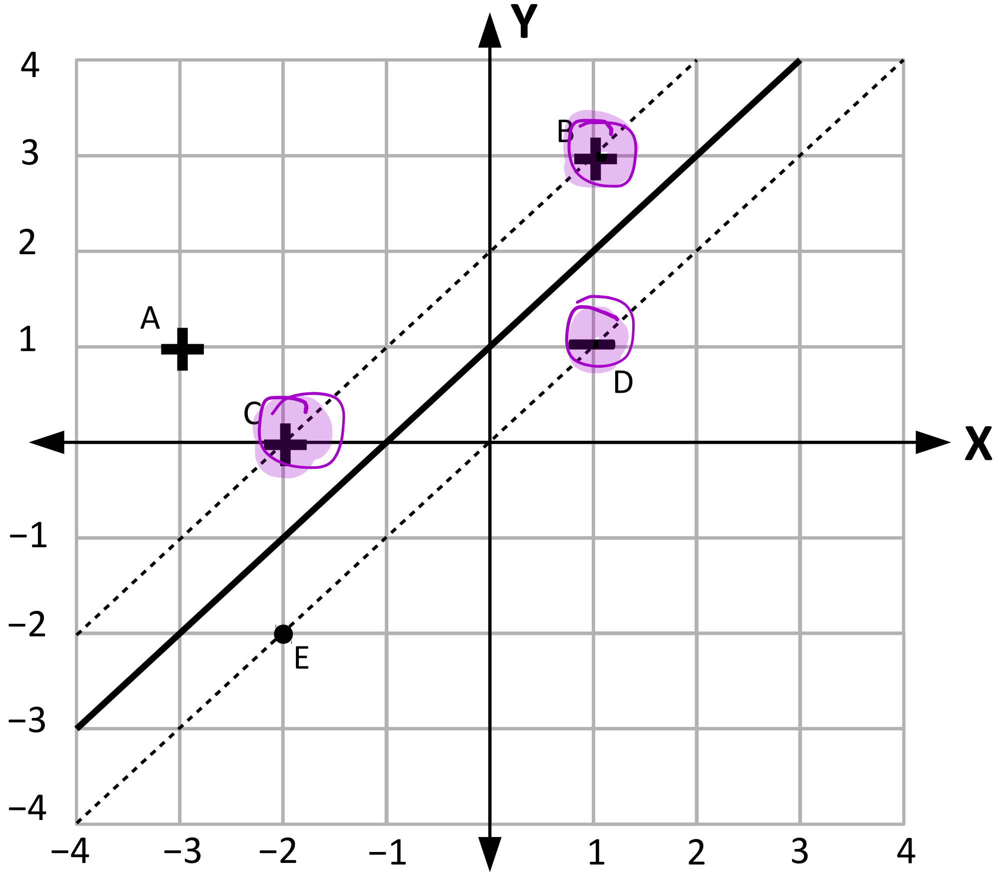
C. A new training point was added outside the original margin

D. A new training point was added within the original margin (between the gutters)

E. A training point was removed from the original SVM

Part B: Sliding Supportiveness (26 points)

B1 (12 points) Sarah trains an SVM, shown below. **A**, **B**, **C**, and **D** are training points. **E** is a test point with unknown classification. You don't need to calculate $\vec{w}$ or b.



1. On the graph above, **CIRCLE the support vectors**.

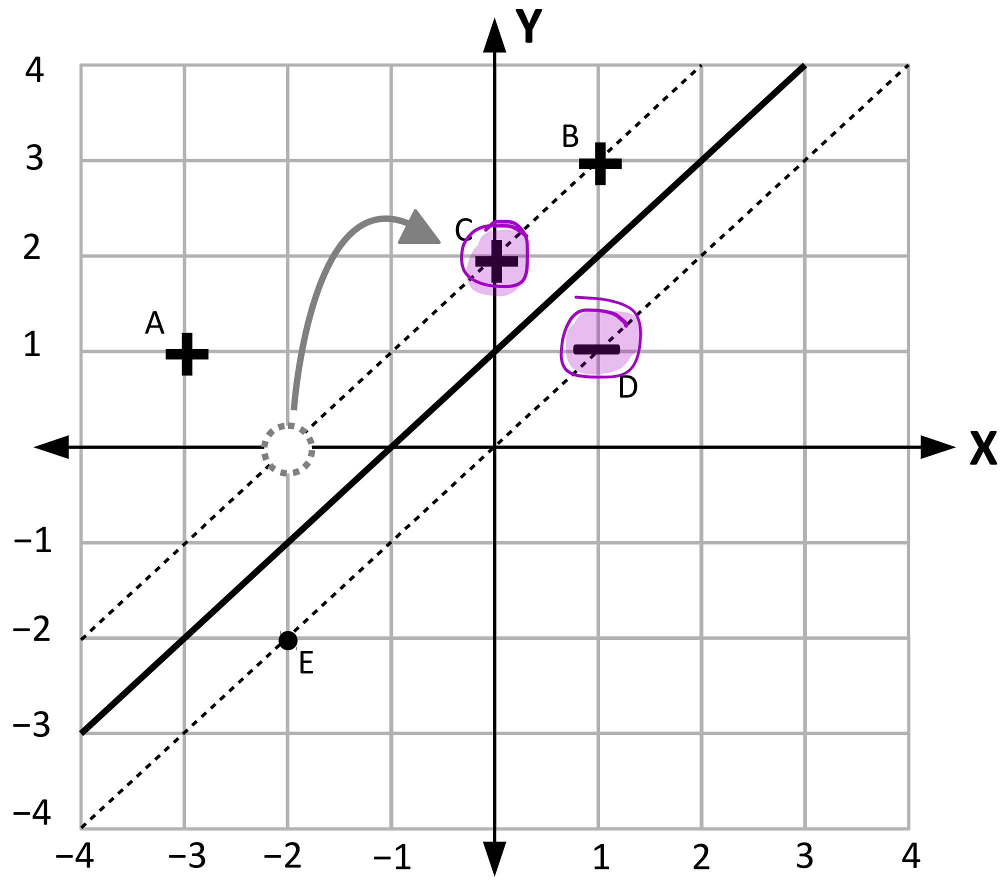
2. For each training point, indicate whether its supportiveness value (α) is positive, negative, or zero. (Circle one for each)

α_A:	**POSITIVE**	**NEGATIVE**	**ZERO**
α_B:	**POSITIVE**	**NEGATIVE**	**ZERO**
α_C:	**POSITIVE**	**NEGATIVE**	**ZERO**
α_D:	**POSITIVE**	**NEGATIVE**	**ZERO**

3. How would Sarah's trained SVM classify the test point **E**? (Circle one) (If a point is on the boundary, classify it as "CAN'T TELL".)

POSITIVE (+) **NEGATIVE (−)** **CAN'T TELL**

B2 (14 points) Mycal modifies Sarah's training data by moving training point **C** from (-2, 0) to (0, 2), then retrains the SVM. The resulting SVM is shown below.



1. On the graph above, **CIRCLE the support vectors**.

2. How does moving point **C** affect the supportiveness values relative to Sarah's SVM in Part B1? For each, indicate whether the value has increased, decreased, or stayed the same (no change). (Circle one for each)

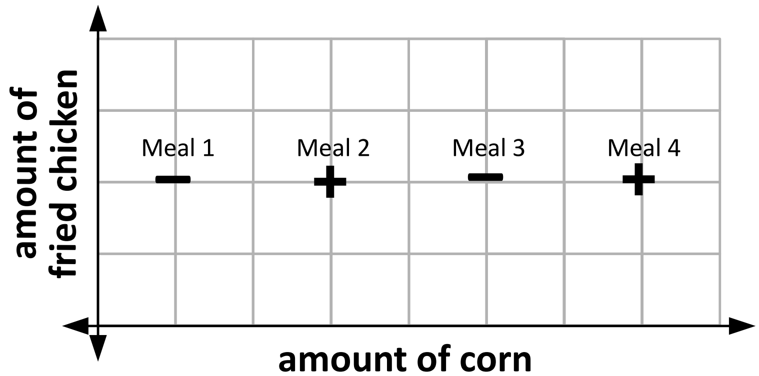
α_A:	**INCREASED (↑)**	**DECREASED (↓)**	**NO CHANGE**
α_B:	**INCREASED (↑)**	**DECREASED (↓)**	**NO CHANGE**
α_C:	**INCREASED (↑)**	**DECREASED (↓)**	**NO CHANGE**
α_D:	**INCREASED (↑)**	**DECREASED (↓)**	**NO CHANGE**

3. How would Mycal's retrained SVM classify the test point **E**? (Circle one)

POSITIVE (+) **NEGATIVE (−)** **CAN'T TELL**

Part C: Colonel Selection (15 points)

Colonel Sanders wants to classify KFC meals using SVMs. He makes four meals with varying amounts of corn (a new side dish) and asks customers to rate them as good (**+**) or bad (**−**). Each meal contains the same amount of fried chicken. He gets the following training data:



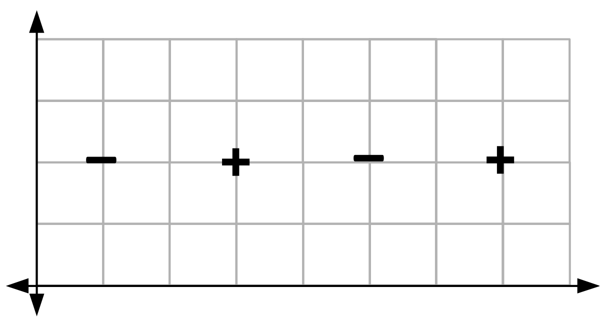
C1 (10 points) For each of the following kernels:

- If Colonel Sanders could use the kernel to produce an SVM that perfectly classifies all four training points, **circle YES**. Then, **sketch an example of a decision boundary** that the kernel might draw. (The boundary does not need to be optimal, as long as it perfectly classifies the training data and could reasonably be drawn by the kernel).
- Otherwise, **circle NO**.

Linear kernel:

YES

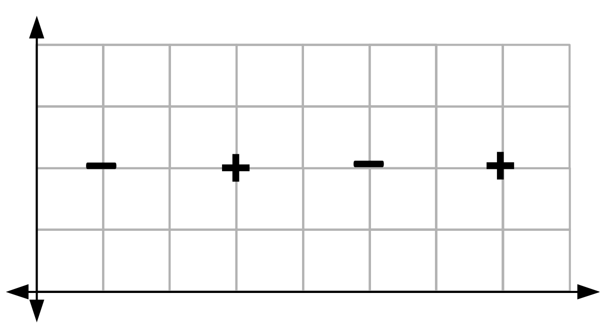
NO



Quadratic kernel (degree-2 polynomial):

YES

NO

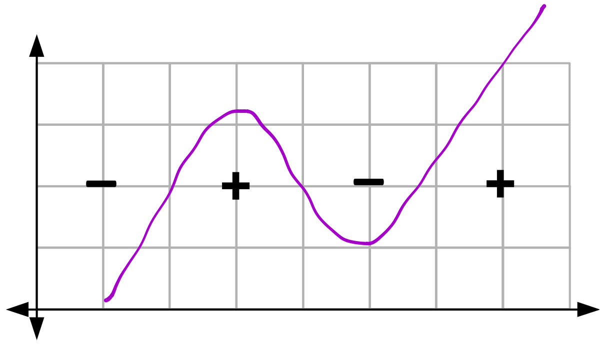


(CONTINUED ON NEXT PAGE →)

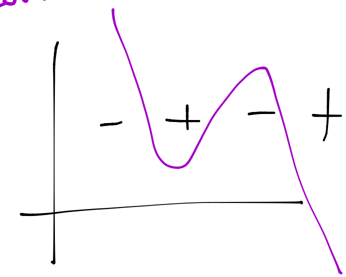
Cubic kernel
(degree-3 polynomial):

YES

NO



both correct

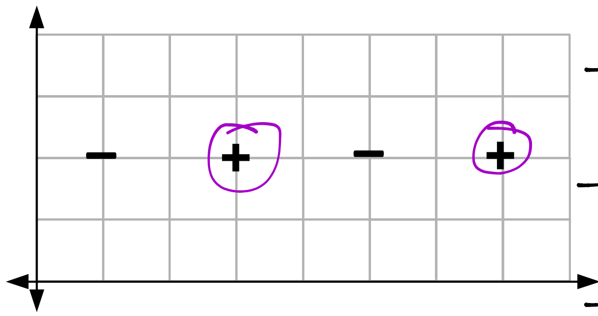


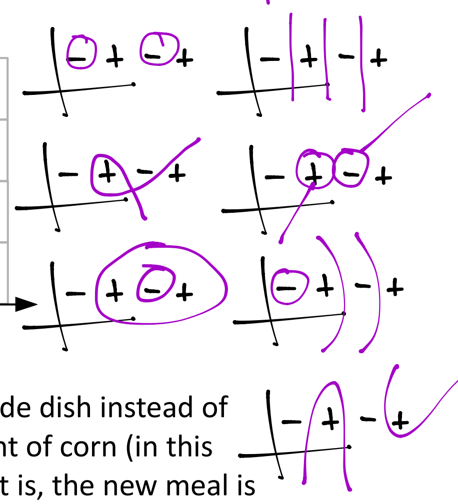
Also accepted:

Radial basis kernel:

YES

NO



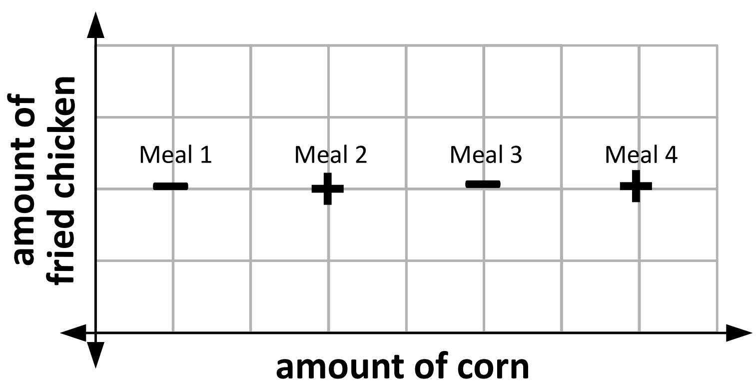


C2 (5 points) Colonel Sanders's friend Orville suggests using popcorn as a side dish instead of regular corn. Colonel Sanders creates a fifth meal that has the same amount of corn (in this case, popcorn) and fried chicken as Meal 3 (**−**). The new meal is a hit! (That is, the new meal is classified as **+**.) Which kernels can perfectly classify all **five** training points? (Circle **ALL** that apply, or circle NONE OF THESE if none apply)

LINEAR **QUADRATIC** **POLYNOMIAL** **RADIAL BASIS**

NONE OF THESE

For your convenience, here is the original data (before adding Meal 5):



(SRN QUESTIONS ON NEXT PAGE →)

Problem 3: Spiritual and Right-Now

Circle the **one best** answer for each of the following questions. There is **no penalty for wrong answers**, so it pays to guess in the absence of knowledge.

1. Winston claimed that widening a neural net by putting more neurons in each layer:

    1. Makes the net susceptible to "computational indigestion."
    2. Increases dropout problems.
    3. Increases the likelihood of getting stuck in a local maximum.
    4. Decreases the likelihood of getting stuck in a local maximum.
    5. Reduces the need for pooling.

2. The Yip-Sussman approach to learning phonological rules:

    1. Uses deep neural nets to replicate the function of the auditory cortex.
    2. Uses the McGurk effect to distinguish vowels from consonants.
    3. Uses nearest-neighbor learning to place decision boundaries in phoneme space.
    4. Uses soft-margin SVMs to place decision boundaries in phoneme space.
    5. Uses several examples to learn pluralization rules.

3. Near-miss learning:

    1. Succeeds only if there is at least one sample for each of the five heuristics.
    2. Learns concepts such as "arch" by exploiting statistical regularities in large datasets.
    3. Uses beam search to build a model that maximizes true positives and minimizes false positives.
    4. Uses a climb-tree heuristic when a positive sample has a different type from the evolving model.
    5. Uses a dot-product mechanism to identify structure type once a model is learned.

4. A transition-space representation works because:

    1. All the useful history of a dynamic system is captured in its current state.
    2. A small vocabulary of change descriptors can describe events such as car crashes.
    3. It is provably equivalent to formal logic.
    4. It connects goals like "get on plane" to subgoals like "arrive at airport."
    5. It constitutes a universal assembly language for all other representations.

5. According to Matt Wilson, researchers can influence:

    1. Rats' hearing by limiting sleep.
    2. Rats' fear of predators by inhibiting cortical function.
    3. Rats' night vision by limiting food.
    4. Rats' dreams by playing sounds.
    5. Rats' prosocial behavior by implanting hormones.

6. According to Matt Wilson, humans with surgically damaged hippocampuses:

    1. Cannot form long-term memories.
    2. Cannot relate tools to uses.
    3. Cannot do mental arithmetic.
    4. Cannot tell left from right.
    5. Cannot recognize their parents.