6.034 Quiz 3

8 November 2017

Name	
Email	

For 1 extra credit point: Circle the TA whose recitations you attend, so that we can more easily enter your score in our records and return your quiz to you promptly:

Suri Bandler	**Erin Hong**	**Samarth Mohan**
Jake Barnwell	**Nathan Landman**	**Michael Shum**
Abigail Choe	**Amanda Liu**	**Jackie Xu**
Francesca Cicileo	**Nick Matthews**	

Problem number	Maximum	Score	Grader
1 – SVMs	50		
2 – Neural Nets	50		
Total	**100**		

SRN	6		

There are 12 pages in this quiz, including this one, but not including tear-off sheets. A tear-off sheet with useful equations is located after the final page of the quiz.

This quiz is open book, open notes, open just about everything, including a calculator, but no computers.

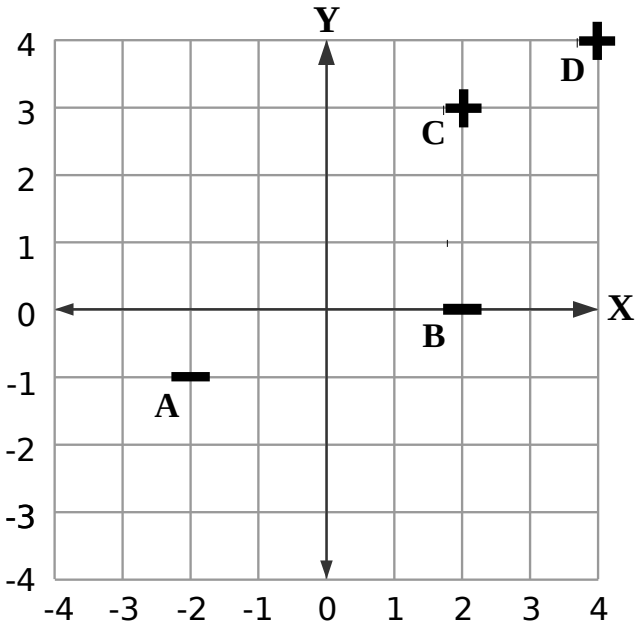
Problem 1: Support Vector Machines (50 points)

The 6034th annual Hunger Games have begun! Kyla the Golden Retriever volunteers as tribute for District 12. Kyla wants to distinguish between her friends, classified as plus (+), and her foes, classified as minus (–).

Part A: Making Good Decisions (32 points)

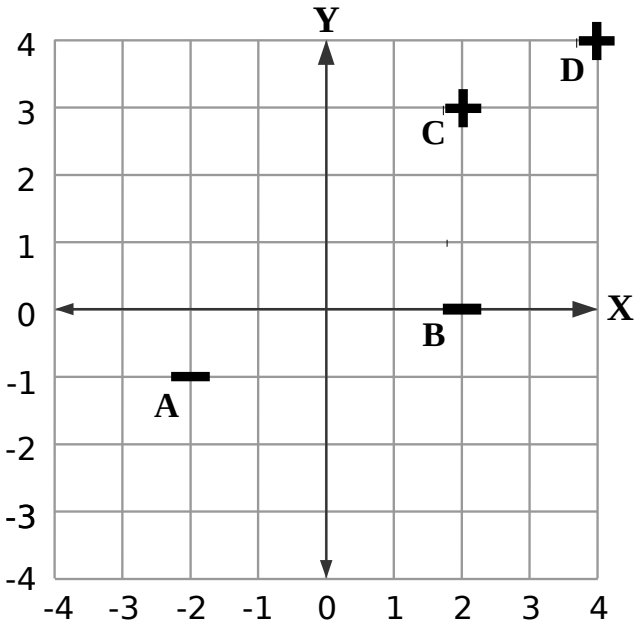
A1 (12 points) Kyla sees a mix of friends and foes in the arena but needs your help to know whom she can trust! You can help her by designing an SVM that can differentiate between friends and foes. On the graph below,

- Draw the SVM **decision boundary** with a solid line.
- Draw the SVM **gutters** with dotted lines.
- Circle the **support vectors**.
- *Draw an arrow in the **direction of the vector*** $\vec{w}$.



This is a duplicate copy of the graph; if you want this copy to be graded instead, check the box:

☐ **I want to start over; grade this copy**



A2 (8 points) Using your decision boundary and data points in part A1 above, compute the values of the normal vector $\vec{w}$ and the offset b.

For your convenience, an equation sheet for SVMs is provided on a tear-off sheet at the end of the quiz.

$\vec{w} =$ $\qquad$ $b =$

For partial credit for part A2, you can show your work here:

A3 (6 points) Of the relations written below, **circle ALL that are true.** If none of the statements are true, instead circle **NONE**. (You may use your intuition; you need not solve for the alphas.)

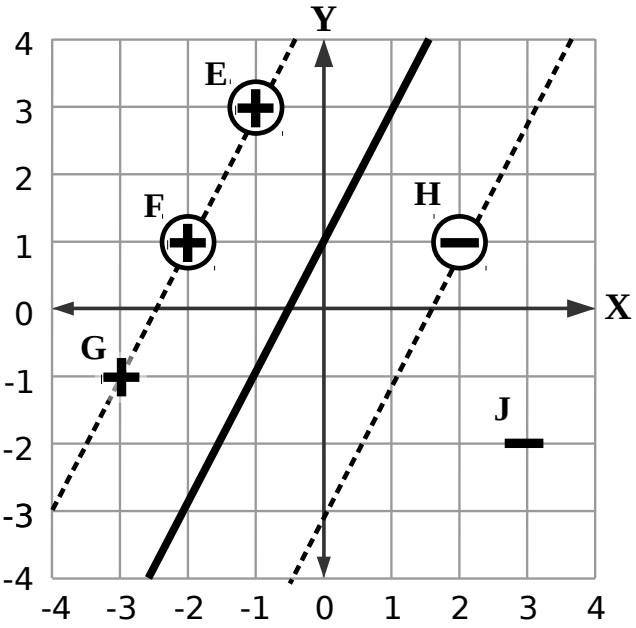
$\alpha_D < \alpha_A$ $\qquad$ $\alpha_C = \alpha_A + \alpha_B$ $\qquad$ $\alpha_C = \alpha_B$ $\qquad$ $\alpha_D > 0$ $\qquad$ NONE

A4 (6 points) Based on the decision boundary you drew in part A1, how should Kyla classify the following points? For each point below, circle the **one** best answer.

$(X=1, Y=2)$	FRIEND (+)	FOE (–)	CAN'T TELL
$(X=-1, Y=0)$	FRIEND (+)	FOE (–)	CAN'T TELL

Part B: No Margin for Error (10 points)

Kyla runs to the forest and sees a new collection of friends and foes, depicted below. (Note that all samples have integer coordinates.) Having now learned how to train SVMs, she solves the SVM problem and finds that the support vectors are **E**, **F**, and **H**. She circles the support vectors, and draws the decision boundary and gutters:



For each of the five samples **E**, **F**, **G**, **H**, and **J**, consider how the margin width would change if the individual sample were **removed** and the SVM were **retrained** with only the four other training samples. Consider each trial independently. For each row below, circle the **one** best answer.

If you...	**The margin width would...**		
Remove E	INCREASE ↑	DECREASE ↓	STAY THE SAME
Remove F	INCREASE ↑	DECREASE ↓	STAY THE SAME
Remove G	INCREASE ↑	DECREASE ↓	STAY THE SAME
Remove H	INCREASE ↑	DECREASE ↓	STAY THE SAME
Remove J	INCREASE ↑	DECREASE ↓	STAY THE SAME

Part C: Kernels (8 points)

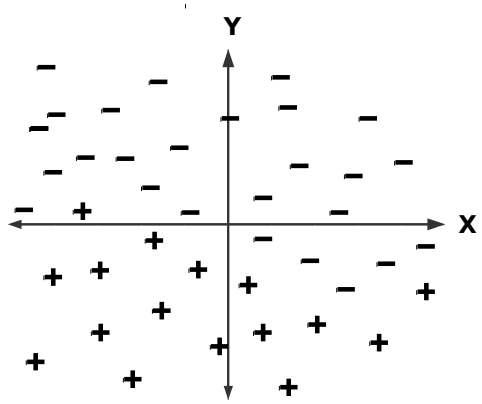
For each of the data sets below, **circle ALL kernels** that could possibly be used to perfectly classify the data with an SVM. If no kernel can be used to perfectly classify the data, instead circle **NONE OF THESE**. The three types of kernels to consider are:

QUADRATIC: A polynomial kernel of degree 2

CUBIC: A polynomial kernel of degree 3

RBF: A radial basis function kernel

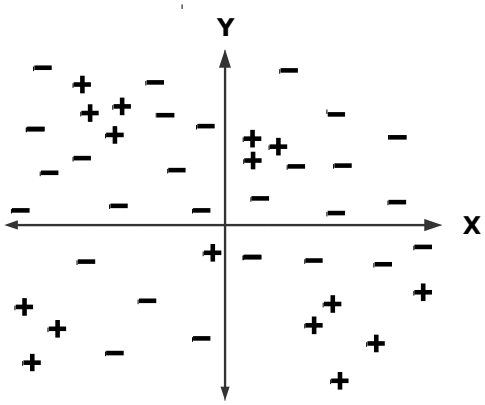
Graph 1



QUADRATIC CUBIC RBF

NONE OF THESE

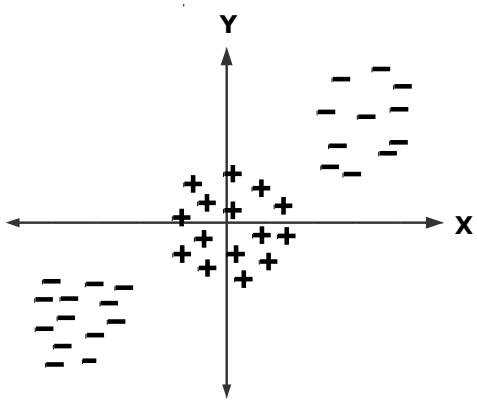
Graph 2



QUADRATIC CUBIC RBF

NONE OF THESE

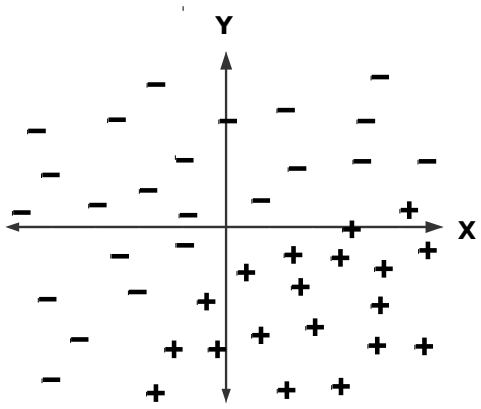
Graph 3



QUADRATIC CUBIC RBF

NONE OF THESE

Graph 4



QUADRATIC CUBIC RBF

NONE OF THESE

Problem 2: Neural Networks (50 points)

Part A: Trial by Big Head (15 points)

You've just been hired by tech-giant Hooli to lead the AI lab. A fellow team member affectionately known as Big Head decides to put your skills to the test with a series of conceptual questions. For each of the following questions, circle the **one** best answer:

A1 (3 points) Your neural net is still inaccurate after having been trained for a fixed number of iterations. You consider adjusting the learning rate. Which of the following adjustments might cause the network to converge to a more accurate solution?

A) Increasing the learning rate.
B) Decreasing the learning rate.
C) Either increasing or decreasing the learning rate.
D) None of these: the change in accuracy is independent of learning rate.

A2 (3 points) One motivation for using the sigmoid instead of the stairstep threshold function in back propagation is:

A) It's computationally inefficient to compute the stairstep gradient.
B) The stairstep function is an odd function.
C) Back propagation is only well-defined for the sigmoid function.
D) The sigmoid function is differentiable everywhere.
E) None of the above.

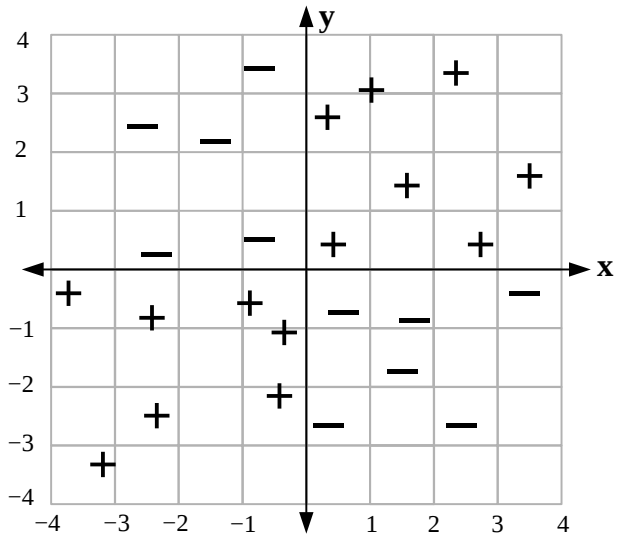
A3 (3 points) Suppose you decide to use **the identity function** as your threshold function:

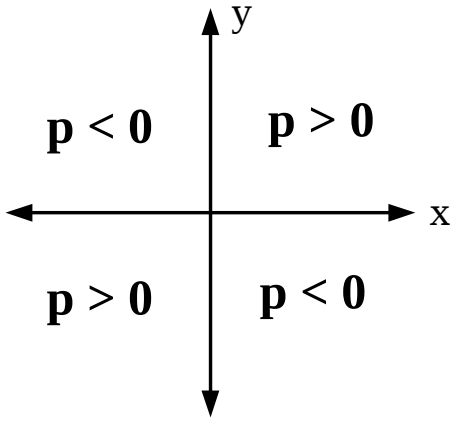
$$f(x) = x$$

In other words, the output of each neuron is just the weighted sum of its inputs. Of the following three neural networks, which one can accurately classify new inputs **that the others cannot**?

A) *Network A*: A neural net with 2 input neurons and 1 output neuron.
B) *Network B*: A neural net with 2 input neurons, 1 hidden-layer neuron, and 1 output neuron.
C) *Network C*: A neural net with 2 input neurons, 3 hidden-layer neurons, and 1 output neuron.
D) Networks A, B, and C can accurately classify the same inputs.
E) Can't tell without more information.

A4 (6 points) You're building a neural net to classify the positive and negative data samples shown below on the left: positive samples will output 1, and negative samples will output 0. Each sample has features *x* and *y*, however, you are only allowed to use the feature $p = x \cdot y$ as input to any neural network. (See below on the right for an illustration of *p*'s value as a function of *x* and *y*.)





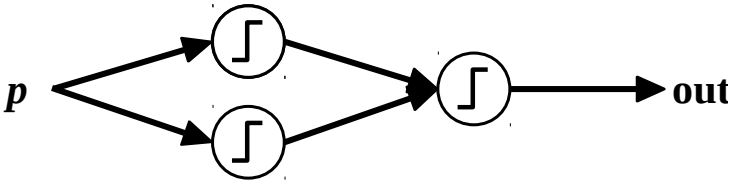
For each of the following network architectures, could a neural network of that shape correctly classify the data? (Assume all neurons use the stairstep threshold function.) In each row, circle either **YES** or **NO**:

Network 1: 

YES **NO**

Network 2: 

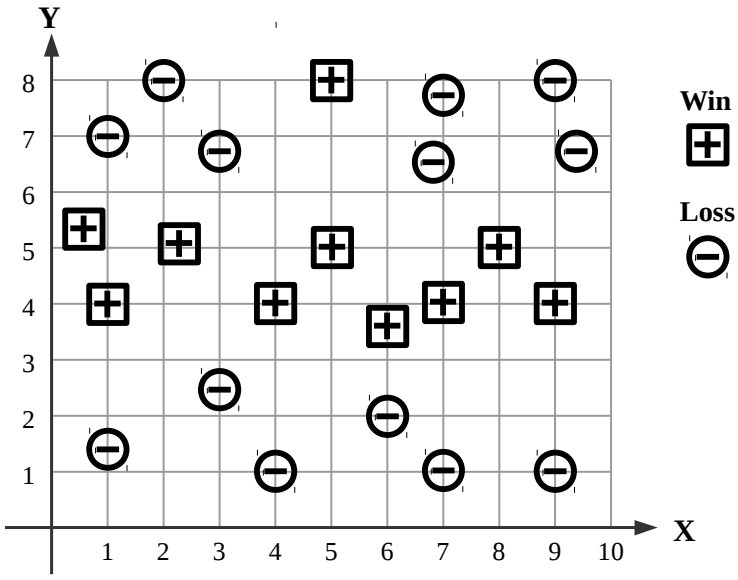
YES **NO**

Network 3: 

YES **NO**

Part B: Let's Go! (21 points)

Impressed with your knowledge, Big Head places you on a top-secret team tasked with building a Go-playing bot called BetaGo. Your job is to build a neural net that can be used to classify winning (⊞) and losing (⊖) board positions, as shown on the right.

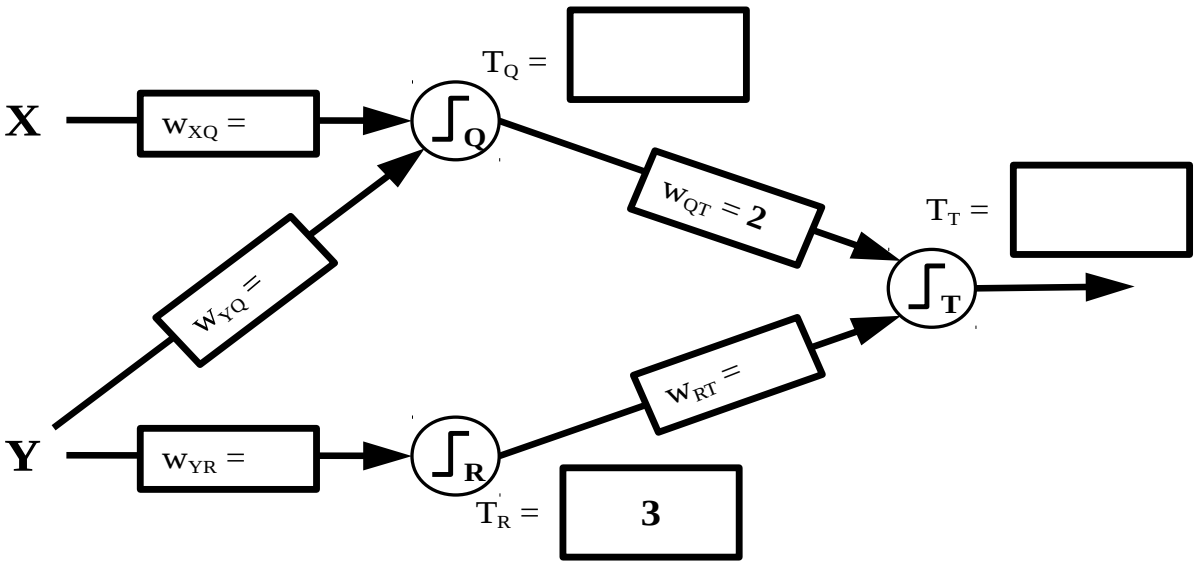


Due to budget cuts, you are limited to using a small net with only three neurons (each of which uses a stairstep threshold function).

Big Head tells you that the small network won't be able to classify *all* of the samples correctly: in fact, it will misclassify *one* sample.

In the neural net skeleton below, we have already filled in weight w_{QT} and threshold T_R for you. Fill in the remaining **four** weights and **two** thresholds with **INTEGER values** such that the net will correctly classify all but one sample. *(We have provided space on the next page to show your work for partial credit.)*

Note: A Win outputs 1; a Loss outputs 0.



For partial credit for part B, you can show your work here:

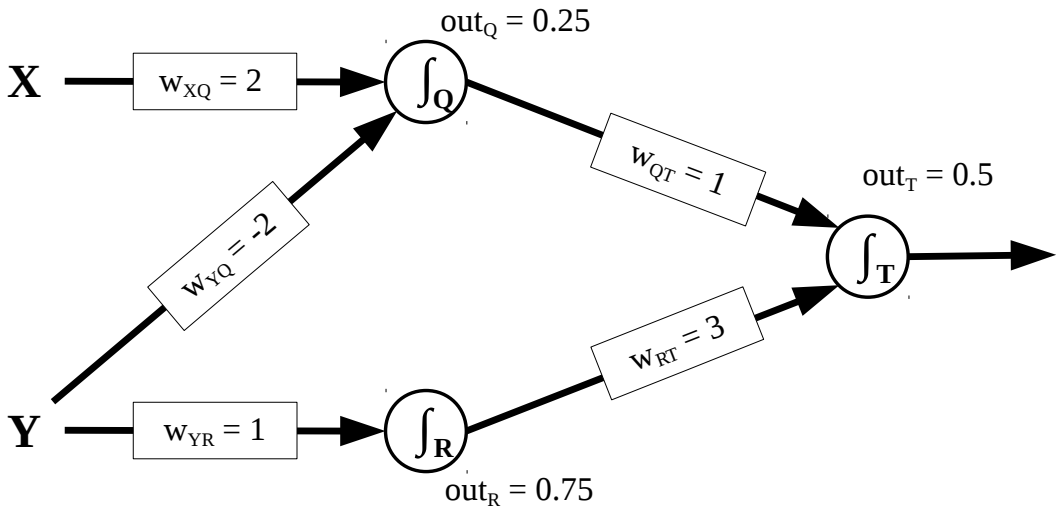
Part C: Back-handed Back-prop (14 points)

Downtrodden startup founder Richard Hendrix approaches and tells Big Head that he doesn't need to manually pick all the weights: there's an automatic way to train the weights of the network using back propagation! Big Head asks for a demonstration of back propagation.

For your convenience, an equation sheet for neural nets is provided on a tear-off sheet at the end of the quiz.

C1 (8 points) The following neural network uses the **sigmoid threshold function**. The weights have been initialized randomly, and you have just performed forward propagation to determine the current outputs of each of the network's neurons. Calculate the values of δ_T and δ_Q, assuming the following:

X = 3, Y = 7, Learning Rate = 1, Desired Output = 0



Space is provided on the next page to show your work.

$\delta_T =$

$\delta_Q =$

For partial credit for part C1, you can show your work here:

C2 (6 points) Calculate the weight update for W_{YR}^{new}, the weight between input Y and neuron R. **You can leave your answer in terms of δ_Q, δ_R, and δ_T.**

$W_{YR}^{new} =$

For partial credit for part C2, you can show your work here:

Spiritual and Right Now (6 points)

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

1. Winston explained that the new version of AlphaGo, described in October:
   (a) Evolves by way of a stochastic genetic algorithm.
   (b) Depends on training from millions of expert human moves.
   (c) Bootstraps up from random behavior by playing itself.
   (d) Still cannot beat teams composed of itself and human supervisors.
   (e) Gains strength from a large repository of opening moves.
2. In inheritance hierarchies:
   (a) Most knowledge is learned by near-miss samples.
   (b) Most knowledge is attached at intermediate levels of a classification tree.
   (c) Most knowledge is expressed in terms of transitions.
   (d) Most knowledge is inherited by offspring from parents in a genetic algorithm.
   (e) Most knowledge is expressed in new forms of formal logic.
3. In their work on learning, Sussman and Yip exploited the sparseness of:
   (a) Phonemes in a high-dimensional distinctive-feature space.
   (b) Possible gestures realizable by the human vocalization apparatus.
   (c) Meaningful sentences likely heard by a child in the first year of life.
   (d) Local maxima in the accuracy function of wide neural networks.
   (e) Syllabic combinations that exhibit the McGurk effect.
4. In near-miss learning, a model learns meaningful information from every data point with:
   (a) Positive examples leading to speculation and near-misses leading to percolation.
   (b) Positive examples leading to sporadic learning and near-misses leading to constant learning.
   (c) Positive examples leading to efficient learning and near-misses leading to generalized learning.
   (d) Positive examples leading to exploitation and near-misses leading to exploration.
   (e) Positive examples leading to generalization and near-misses leading to specialization.
5. One of Minsky's contributions associated with society of mind is:
   (a) Explaining human intelligence via multiplicities of mechanisms.
   (b) Using crowd-sourcing to develop a large data set of common sense rules.
   (c) Retaining the usefulness of humans by wiring multiple brains together.
   (d) Developing an open-source community to develop AI systems.
   (e) Recognizing that generalized groups are smarter than specialized individuals.
6. Brook's research group produced a robot that:
   (a) Played fetch with a dog.
   (b) Prepared mixed drinks in a barroom demonstration.
   (c) Picked up a soda can and got rid of it.
   (d) Produced a copy of itself.
   (e) Participated in a robot soccer tournament.

(Tear-off sheet)

Common SVM equations as seen in recitation:

1. $\vec{w} \cdot \vec{x} + b = 0$

2. $MarginWidth = \frac{2}{\|\vec{w}\|}$

3. $\vec{w} \cdot \vec{x}_i + b = y_i$

4. $\sum_i y_i \alpha_i = 0$

5. $\vec{w} = \sum_i y_i \alpha_i \vec{x}_i$

(Tear-off sheet)

Common neural network equations as seen in recitation:

1. $w_{AB,new} = w_{AB,old} + \Delta w_{AB}$

2. $\Delta w_{AB} = r \cdot out_A \cdot \delta_B$

3. $$\delta_B = \begin{cases} out_B(1-out_B)(out*-out_B), \text{ if B in final layer} \\ out_B(1-out_B)\sum_{outgoing\, C_i} w_{BC_i}\delta_{C_i}, \text{ otherwise} \end{cases}$$