

WORD REPRESENTATIONS

David Kauchak
CS158 – Fall 2016

Admin

Quiz #2

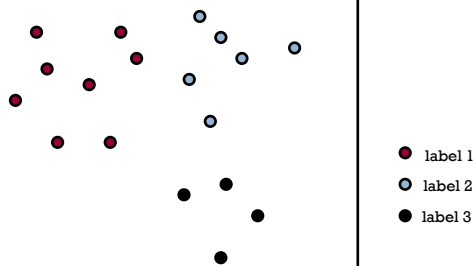
- ▣ Quartile1: 23.25 (78%)
- ▣ Median: 26 (87%)
- ▣ Quartile3: 28 (93%)
- ▣ Average: 24.8 (83%)

Assignments 3 and 5a graded (4b back soon)

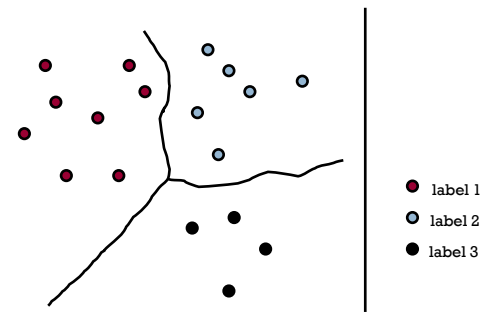
Assignment 5

Decision boundaries

The **decision boundaries** are places in the features space where the classification of a point/example changes

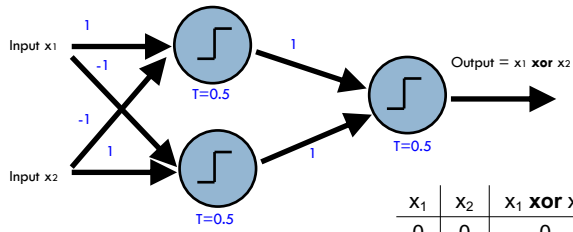


A complicated decision boundary



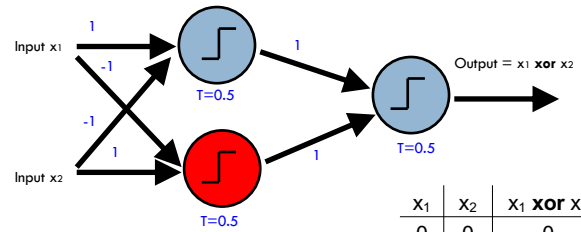
For those curious, this is the decision boundary for k-nearest neighbors

What does the decision boundary look like?



x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

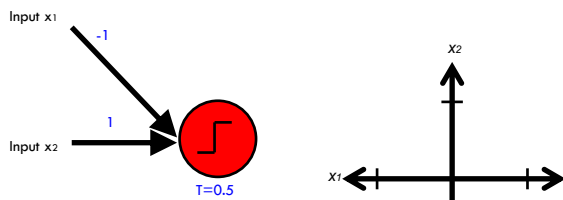
What does the decision boundary look like?



x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0

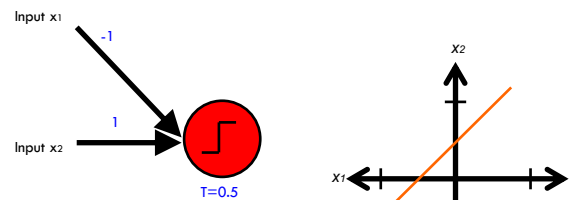
What does this perceptron's decision boundary look like?

NN decision boundary



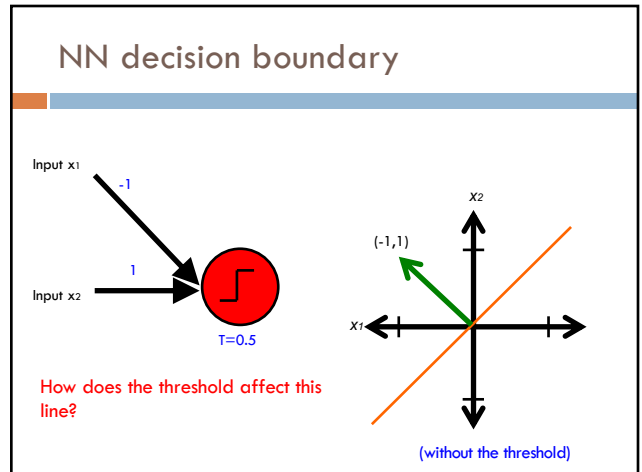
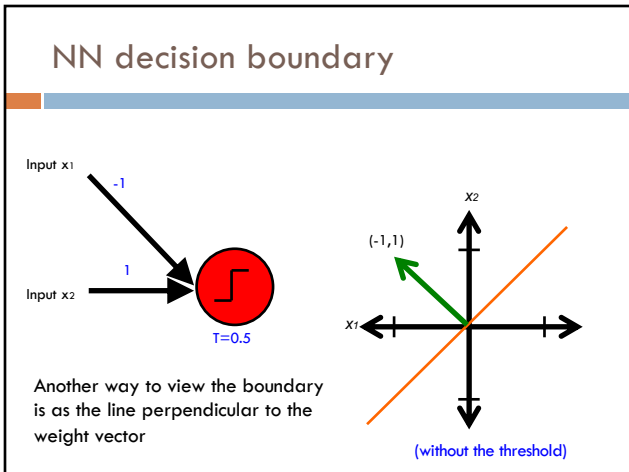
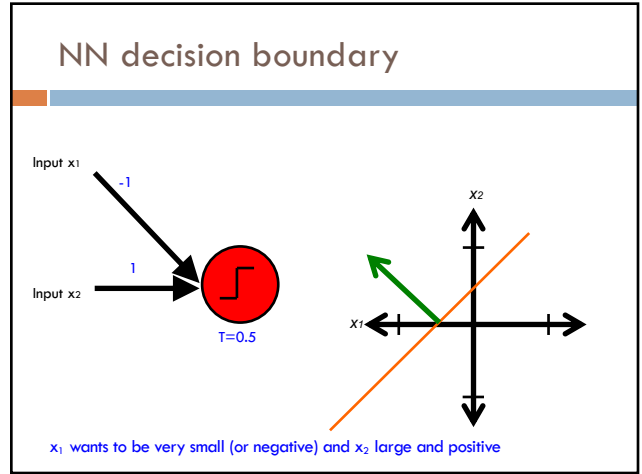
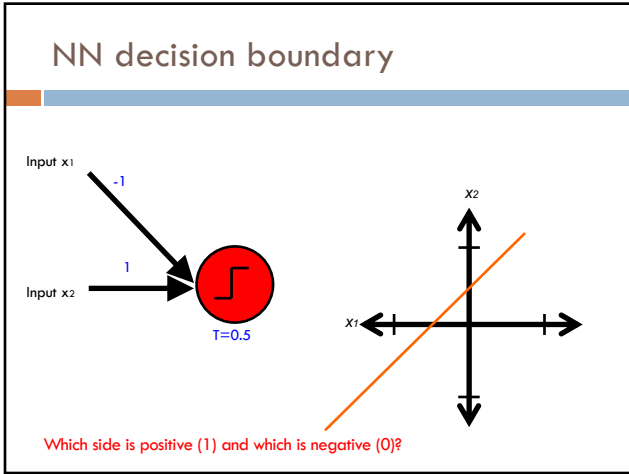
What does this perceptron's decision boundary look like?

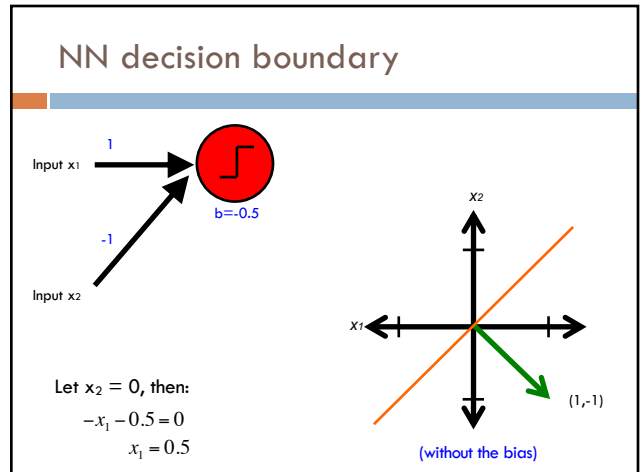
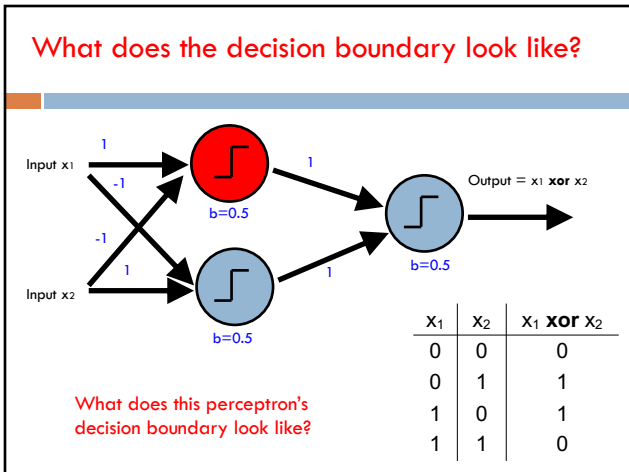
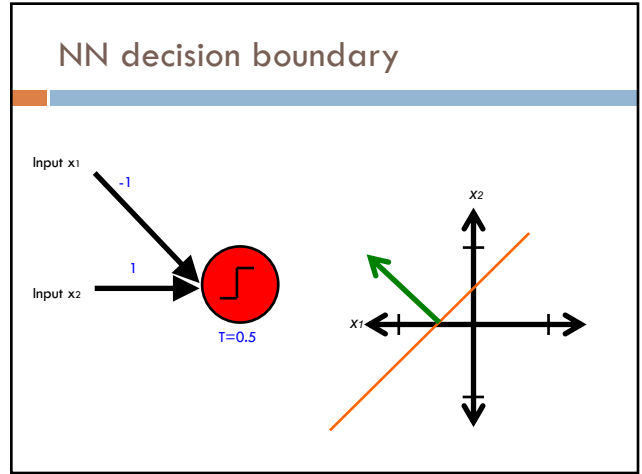
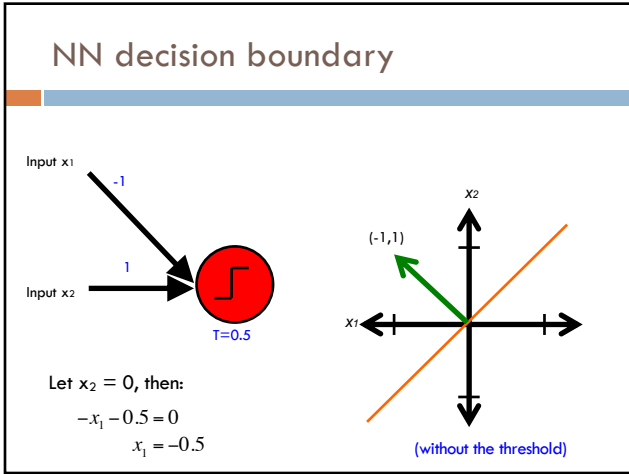
NN decision boundary

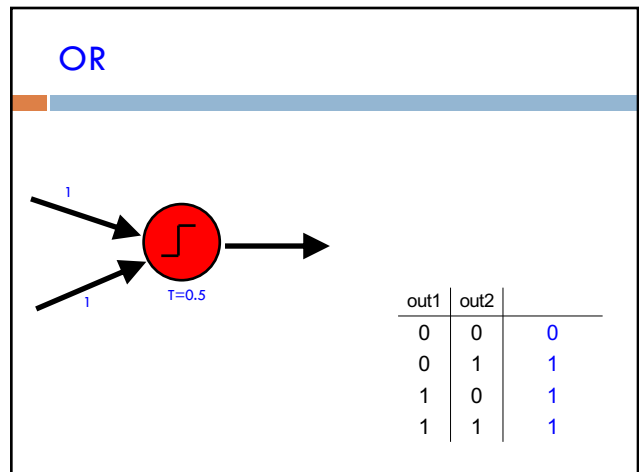
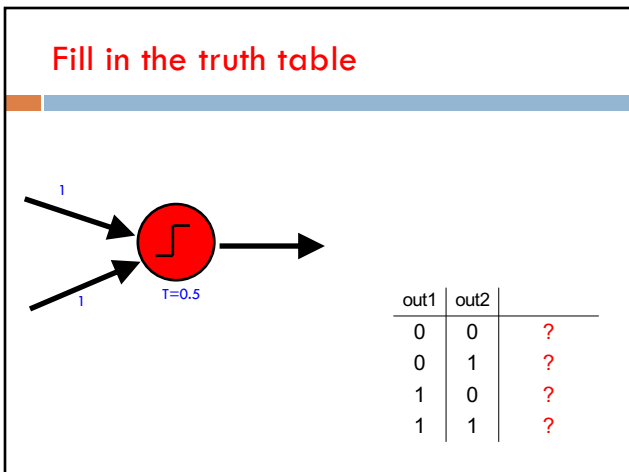
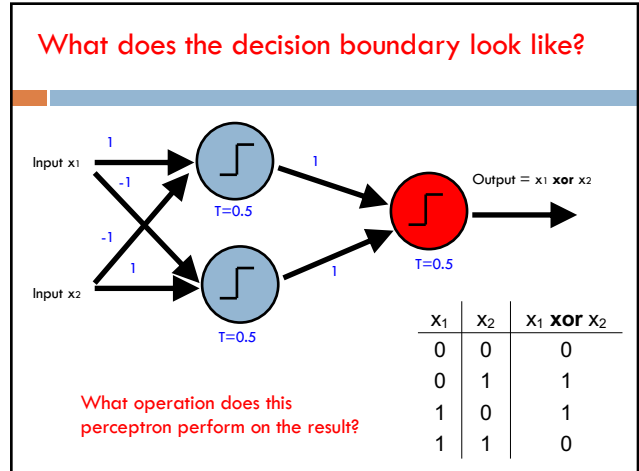
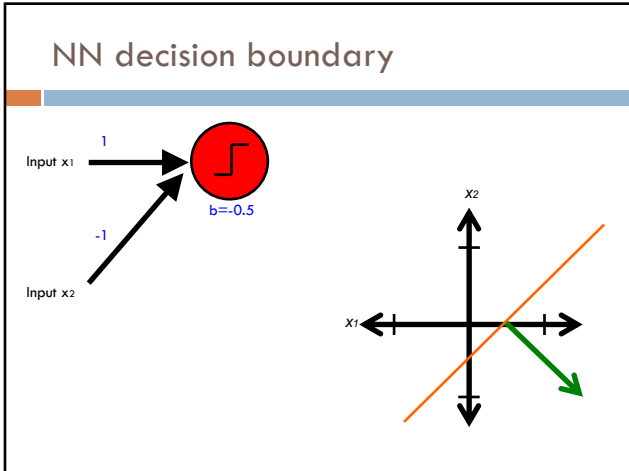


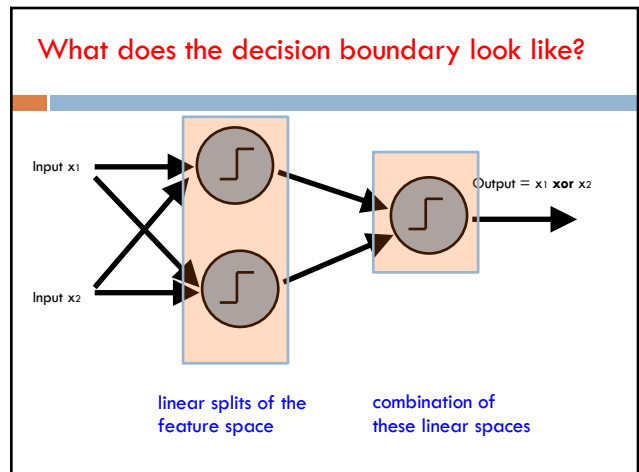
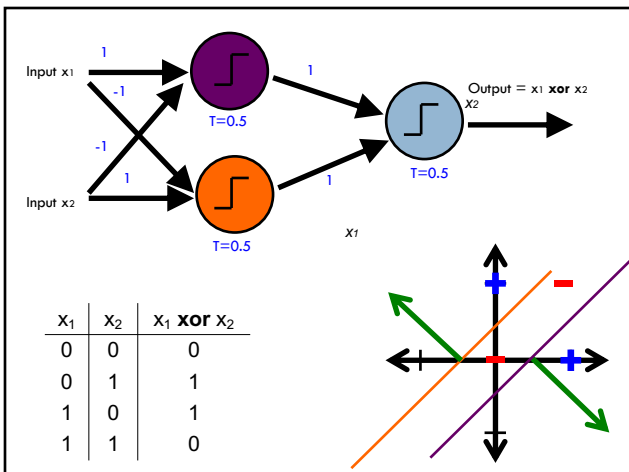
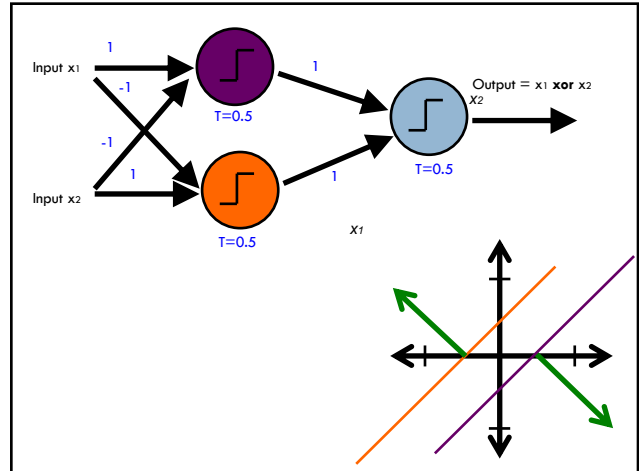
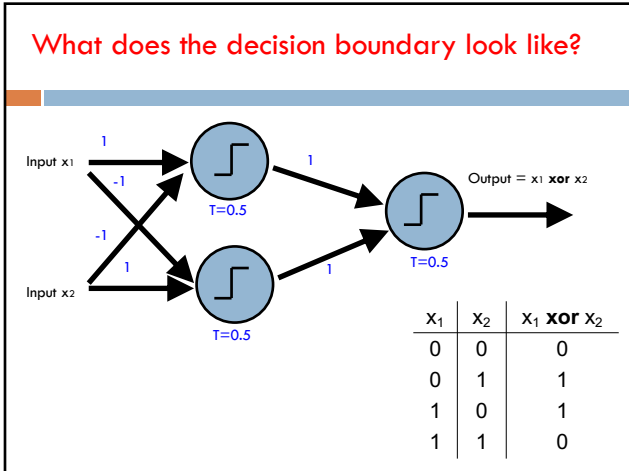
$$-x_1 + x_2 = 0.5$$

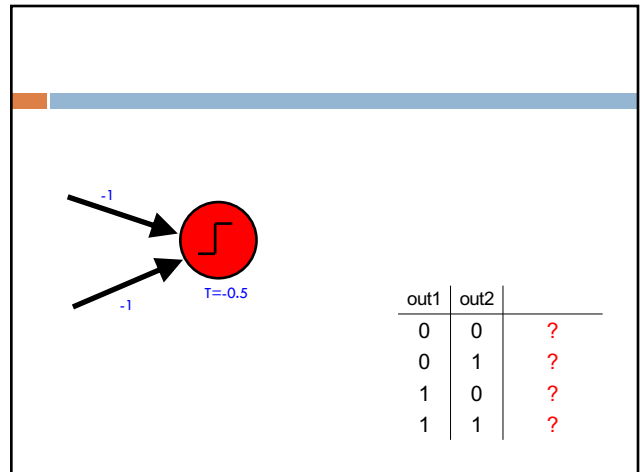
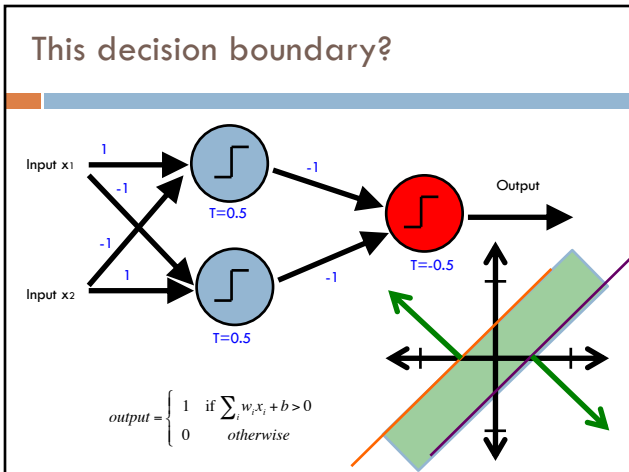
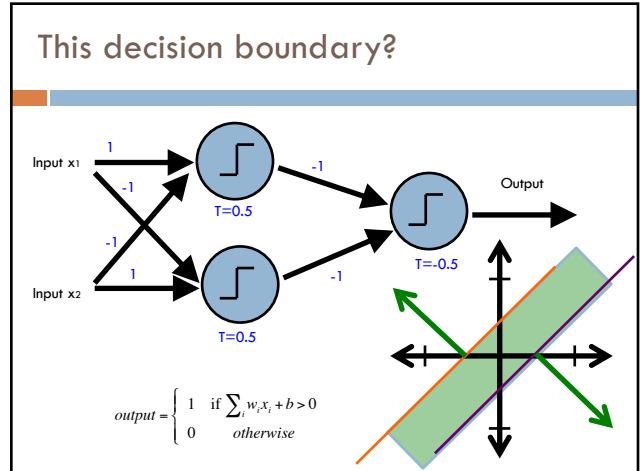
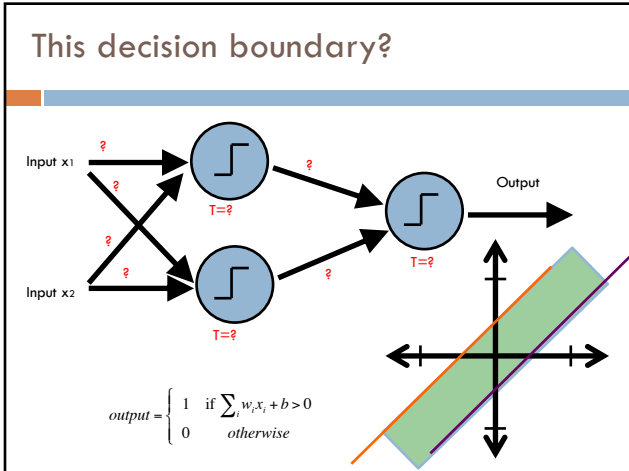
$$x_2 = x_1 + 0.5$$



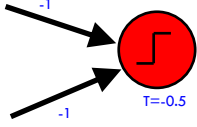






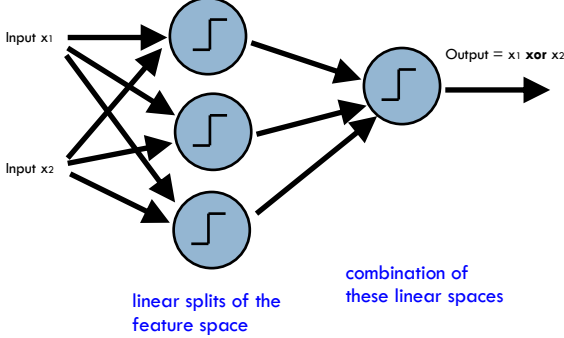


NOR

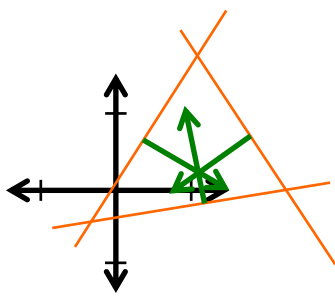


out1	out2	
0	0	1
0	1	0
1	0	0
1	1	0

What does the decision boundary look like?



Three hidden nodes



NN decision boundaries

Theorem 9 (Two-Layer Networks are Universal Function Approximators). *Let F be a continuous function on a bounded subset of D -dimensional space. Then there exists a two-layer neural network \hat{F} with a finite number of hidden units that approximate F arbitrarily well. Namely, for all x in the domain of F , $|F(x) - \hat{F}(x)| < \epsilon$.*

'Or, in colloquial terms "two-layer networks can approximate any function."

NN decision boundaries

More hidden nodes = more complexity

Adding more layers adds even more complexity (and much more quickly)

Good rule of thumb:

$$\text{number of 2-layer hidden nodes} \leq \frac{\text{number of examples}}{\text{number of dimensions}}$$



Trained a NN with 1B connections on 10M snapshots from youtube on 16,000 processors

<http://www.nytimes.com/2012/06/26/technology/in-a-big-network-of-computers-evidence-of-machine-learning.html>

Deep learning



Deep learning is a branch of machine learning based on a set of algorithms that attempt to model high level abstractions in data by using a deep graph with multiple processing layers, composed of multiple linear and non-linear transformations.

Deep learning is part of a broader family of machine learning methods based on learning representations of data.

Deep learning

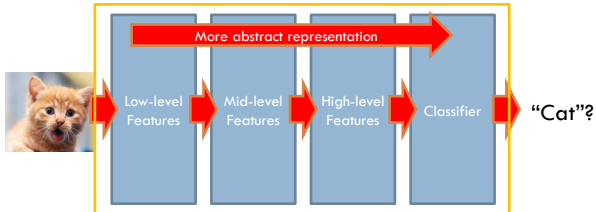
Key: learning better features that abstract from the "raw" data

Using **learned** feature representations based on large amounts of data, generally unsupervised

Using classifiers with multiple layers of learning

Deep learning

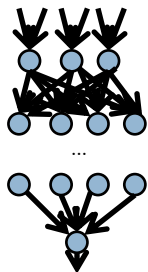
- Train *multiple layers* of features/abstractions from data.
- Try to discover *representation* that makes decisions easy.



Deep Learning: train layers of features so that classifier works well.

Slide adapted from: Adam Coates

Deep learning for neural networks

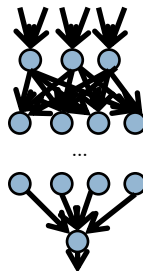


Traditional NN models: 1-2 hidden layers

Deep learning NN models: 3+ hidden layers

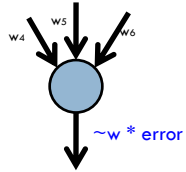
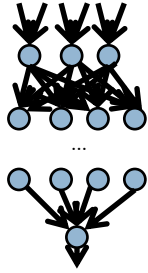
Challenges

What makes "deep learning" hard for NNs?



Challenges

What makes “deep learning” hard for NNs?



Modified errors tend to get diluted as they get combined with many layers of weight corrections

Deep learning

Growing field

Driven by:

- ▣ Increase in data availability
- ▣ Increase in computational power
- ▣ Parallelizability of many of the algorithms

Involves more than just neural networks (though, they're a very popular model)

word2vec

How many people have heard of it?

What is it?

Word representations generalized

Project words into a multi-dimensional “meaning” space

word $\rightarrow [x_1, x_2, \dots, x_d]$

What is our projection for assignment 5?

Word representations generalized

Project words into a multi-dimensional “meaning” space

word $\rightarrow [w_1, w_2, \dots, w_d]$

Each dimension is the co-occurrence of word with w_i

Word representations

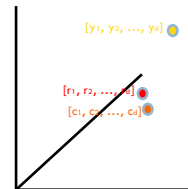
Project words into a multi-dimensional “meaning” space

word $\rightarrow [x_1, x_2, \dots, x_d]$

red $\rightarrow [r_1, r_2, \dots, r_d]$

crimson $\rightarrow [c_1, c_2, \dots, c_d]$

yellow $\rightarrow [y_1, y_2, \dots, y_d]$



Word representations

Project words into a multi-dimensional “meaning” space

word $\rightarrow [x_1, x_2, \dots, x_d]$

The idea of word representations is not new:

- Co-occurrence matrices
- Latent Semantic Analysis (LSA)

New idea: learn word representation using a task-driven approach

A prediction problem

I like to eat bananas with cream cheese

Given a context of words

Predict what words are likely to occur in that context

A prediction problem

Given text, can generate lots of **positive** examples:

I like to eat bananas with cream cheese

input

___ like to eat

I ___ to eat bananas

I like ___ eat bananas with

I like to ___ bananas with cream

...

prediction

I

like

to

eat

...

A prediction problem

Use data like this to learn a distribution:

$$p(\text{word} \mid \text{context})$$

$$p(w_i \mid w_{i-2} w_{i-1} w_{i+1} w_{i+2})$$

words before
words after

A prediction problem

Any problems with using only positive examples?

$$p(w_i \mid w_{i-2} w_{i-1} w_{i+1} w_{i+2})$$

input

___ like to eat

I ___ to eat bananas

I like ___ eat bananas with

I like to ___ bananas with cream

...

prediction

I

like

to

eat

...

A prediction problem

Want to learn a distribution over **all** words

$$p(w_i \mid w_{i-2} w_{i-1} w_{i+1} w_{i+2})$$

input

___ like to eat

I ___ to eat bananas

I like ___ eat bananas with

I like to ___ bananas with cream

...

prediction

I

like

to

eat

...

A prediction problem

Negative examples?

I like to eat bananas with cream cheese

input

___ like to eat
I ___ to eat bananas
I like ___ eat bananas with
I like to ___ bananas with cream
...

prediction

I
like
to
eat
...

A prediction problem

Any other word that didn't occur in that context

I like to eat bananas with cream cheese

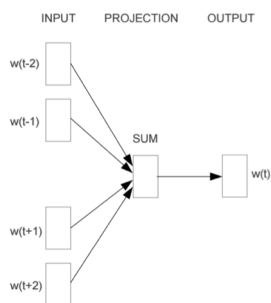
input

___ like to eat
I ___ to eat bananas
I like ___ eat bananas with
I like to ___ bananas with cream
...

prediction (negative)

car
snoopy
run
sloth
...

Train a neural network on this problem



<https://arxiv.org/pdf/1301.3781v3.pdf>

Encoding words

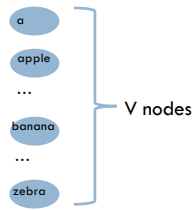
How can we input a "word" into a network?



“One-hot” encoding

For a vocabulary of V words, have V input nodes

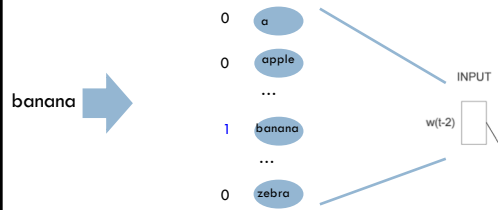
All inputs are 0 except the for the one corresponding to the word



“One-hot” encoding

For a vocabulary of V words, have V input nodes

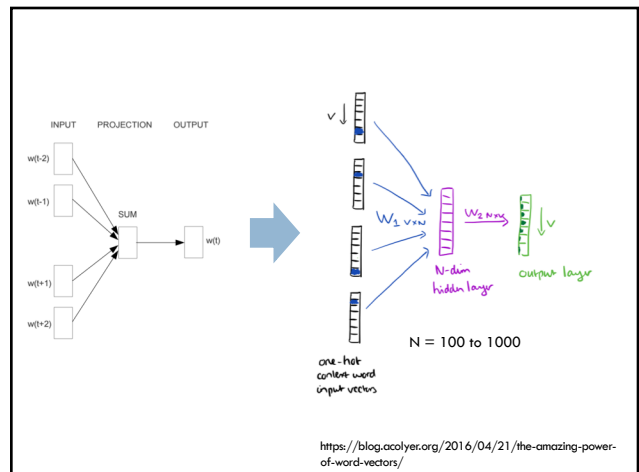
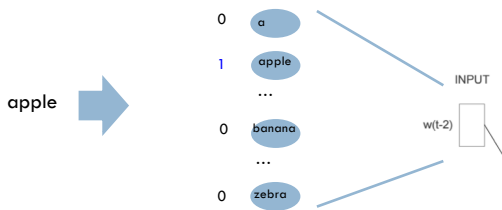
All inputs are 0 except the for the one corresponding to the word

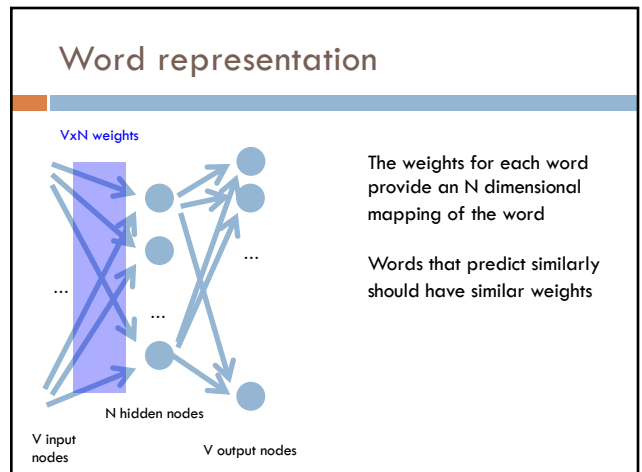
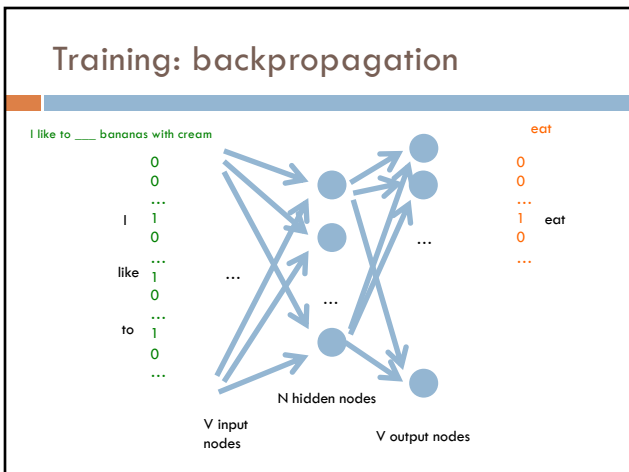
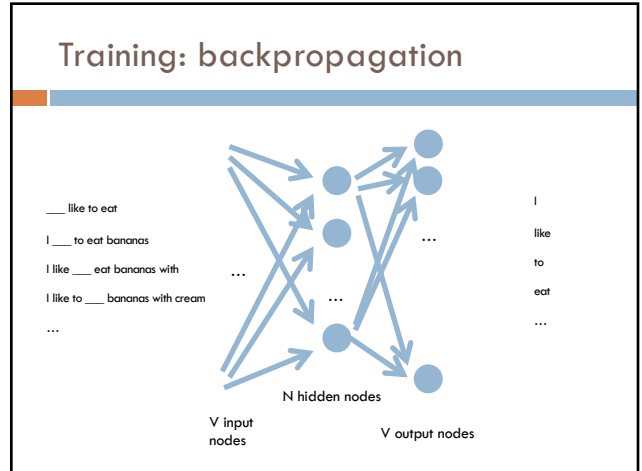
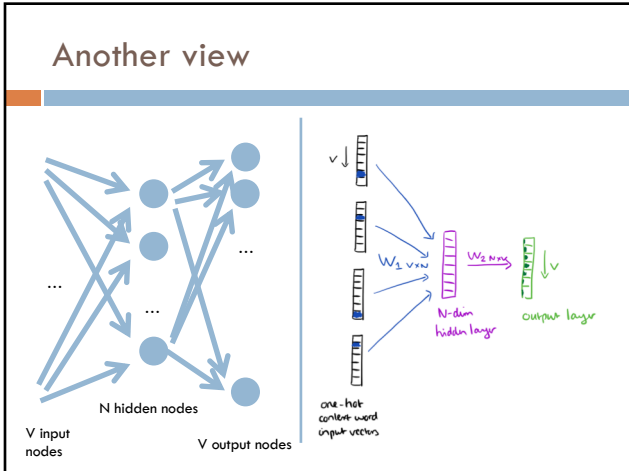


“One-hot” encoding

For a vocabulary of V words, have V input nodes

All inputs are 0 except the for the one corresponding to the word





Results

$$\text{vector}(\text{word1}) - \text{vector}(\text{word2}) = \text{vector}(\text{word3}) - X$$

word1 is to word2 as word3 is to X

Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter

Results

$$\text{vector}(\text{word1}) - \text{vector}(\text{word2}) = \text{vector}(\text{word3}) - X$$

word1 is to word2 as word3 is to X

Type of relationship	Word Pair 1		Word Pair 2	
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

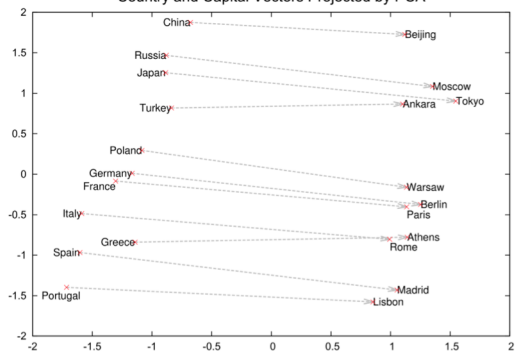
Results

$$\text{vector}(\text{word1}) - \text{vector}(\text{word2}) = \text{vector}(\text{word3}) - X$$

word1 is to word2 as word3 is to X

Newspapers			
New York	New York Times	Baltimore	Baltimore Sun
San Jose	San Jose Mercury News	Cincinnati	Cincinnati Enquirer
NHL Teams			
Boston	Boston Bruins	Montreal	Montreal Canadiens
Phoenix	Phoenix Coyotes	Nashville	Nashville Predators
NBA Teams			
Detroit	Detroit Pistons	Toronto	Toronto Raptors
Oakland	Golden State Warriors	Memphis	Memphis Grizzlies
Airlines			
Austria	Austrian Airlines	Spain	Spainair
Belgium	Brussels Airlines	Greece	Aegean Airlines
Company executives			
Steve Ballmer	Microsoft	Larry Page	Google
Samuel J. Palmisano	IBM	Werner Vogels	Amazon

Country and Capital Vectors Projected by PCA

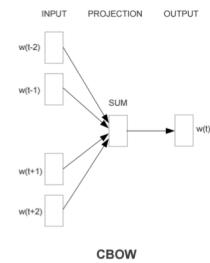


2-Dimensional projection of the N-dimensional space

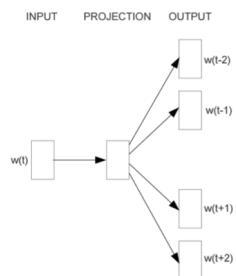
Visualized

<https://projector.tensorflow.org/>

Continuous Bag Of Words



Other models: skip-gram



word2vec

A model for learning word representations from large amounts of data

Has become a popular pre-processing step for learning a more robust feature representation

Models like word2vec have also been incorporated into other learning approaches (e.g. translation tasks)

word2vec resources

- <https://blog.acolyer.org/2016/04/21/the-amazing-power-of-word-vectors/>
- <https://code.google.com/archive/p/word2vec/>
- <https://deeplearning4j.org/word2vec>
- <https://arxiv.org/pdf/1301.3781v3.pdf>