Deep Learning for Natural Language Processing

AN OVERVIEW OF THE POSSIBILITIES

BASED ON COURSES GIVEN BY A. NG (STANFORD)

Organisation

- Deep Learning comes from Machine Learning!
  - (Un)supervised learning, Gradient descent, polynomial regression…
- Neural Network
  - From biology to logic
- Application of Deep Learning to NLP problems
  - Language Models, Statistical Machine Translation, Word Embeddings…

Deep Learning comes from Machine Learning!

SOME MACHINE LEARNING BASES TO UNDERSTAND DEEP LEARNING

Deep Learning comes from Machine Learning!

- What is Machine Learning?
- Supervised learning
- Unsupervised learning
- Non-linear classification problem

What is Machine Learning?

A FIRST STEP TO DEEP LEARNING

What is Machine Learning?

No definition accepted by everyone but some references:

- Arthur Samuel (1959): “the field of study that gives computers the ability to learn without being explicitly programmed”
  - Checker program
- Tom Mitchell (1998): “a computer program is said to learn from experience E, with respect to some task T, and some performance measure P, if its performance on T as measured by P improves with experience E.”

Example of ML

According to Mitchell's definition:

"a computer program is said to learn from experience E, with respect to some task T, and some performance measure P, if its performance on T as measured by P improves with experience E."

- Suppose your email program watches which email you do or do not mark as spam. Based on that it learns how to better filter your email. What is the task T in this setting?
  - Classifying emails as spam or not spam:
  - Watching you label email as spam or not spam:
  - The number of emails correctly classified as spam or not spam:
  - None of the above, this is not a ML problem!

Example of ML

According to Mitchell's definition:

"a computer program is said to learn from experience E, with respect to some task T, and some performance measure P, if its performance on T as measured by P improves with experience E."

- Suppose your email program watches which email you do or do not mark as spam. Based on that it learns how to better filter your email. What is the task T in this setting?
  - Classifying emails as spam or not spam: T
  - Watching you label email as spam or not spam: E
  - The number of emails correctly classified as spam or not spam: P
  - ~~None of the above, this is not a ML problem~~!

Supervised learning

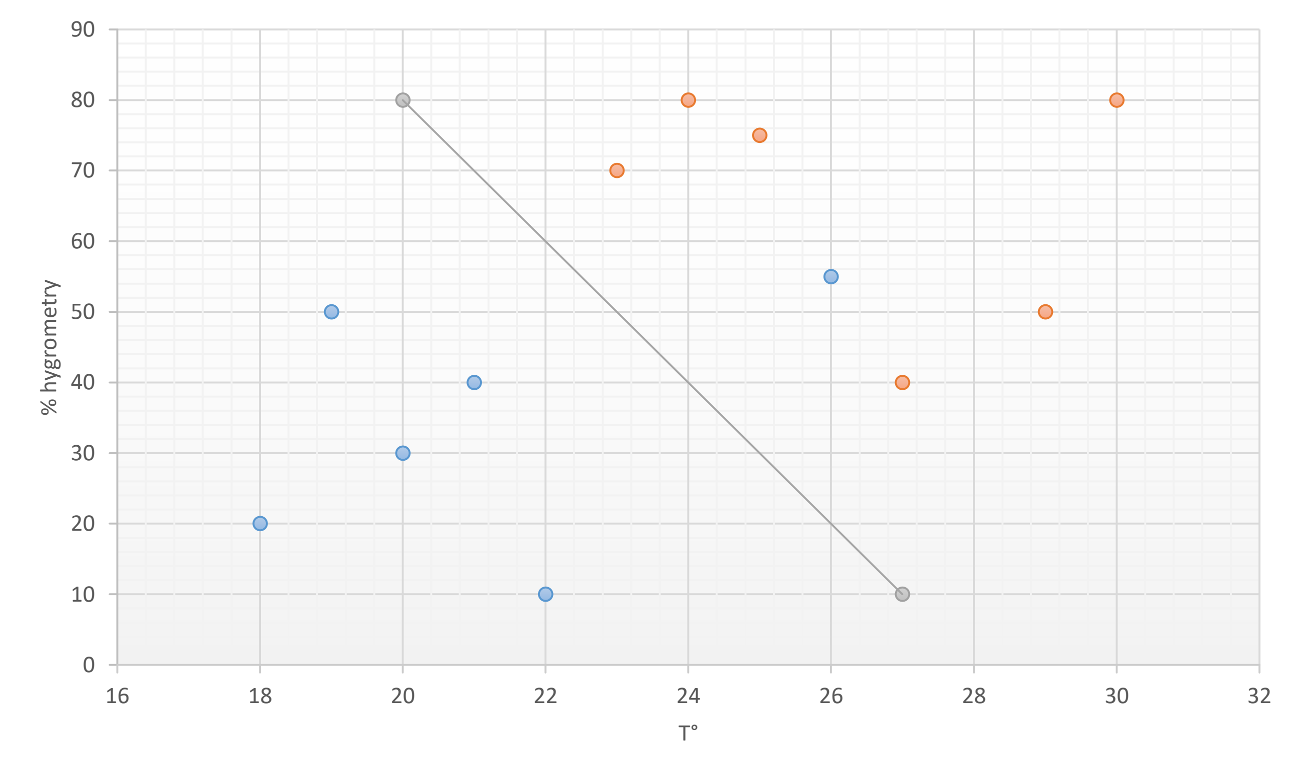
YOU KNOW WHAT YOU LEARN IS TRUE!

Supervised learning

- Example of supervised learning: house prices
  - For each sample of data, the right answer is given
  - Introduction of the linear regression:
    predict continuous valued output (price)

Supervised learning

- Example of supervised learning: healthy habitat
  - For each sample of data, the right answer is given
  - Introduction of the classification problem:
    predict a discrete value



Supervised learning

- Every example in the data set are « correct answer »
- When we want to predict a continuous value : <u>regression problem</u>
  - Example of prediction of house pricing
- When we want to predict a discrete value : <u>classification problem</u>
  - Example of prediction if the air is healthy or not

Supervised learning

- Problem 1: You have a large inventory of identical items. You want to predict how many of them you will sell in the next 3 months
- Problem 2: You have large set of customer files and you want to determine which of them have been compromised or not
- Should you treat them?
  - Both as classification problem
  - Both as regression problem
  - Pbm 1 as classification problem and Pbm 2 as regression problem
  - Pbm 1 as regression problem and Pbm 2 as classification problem

Supervised learning

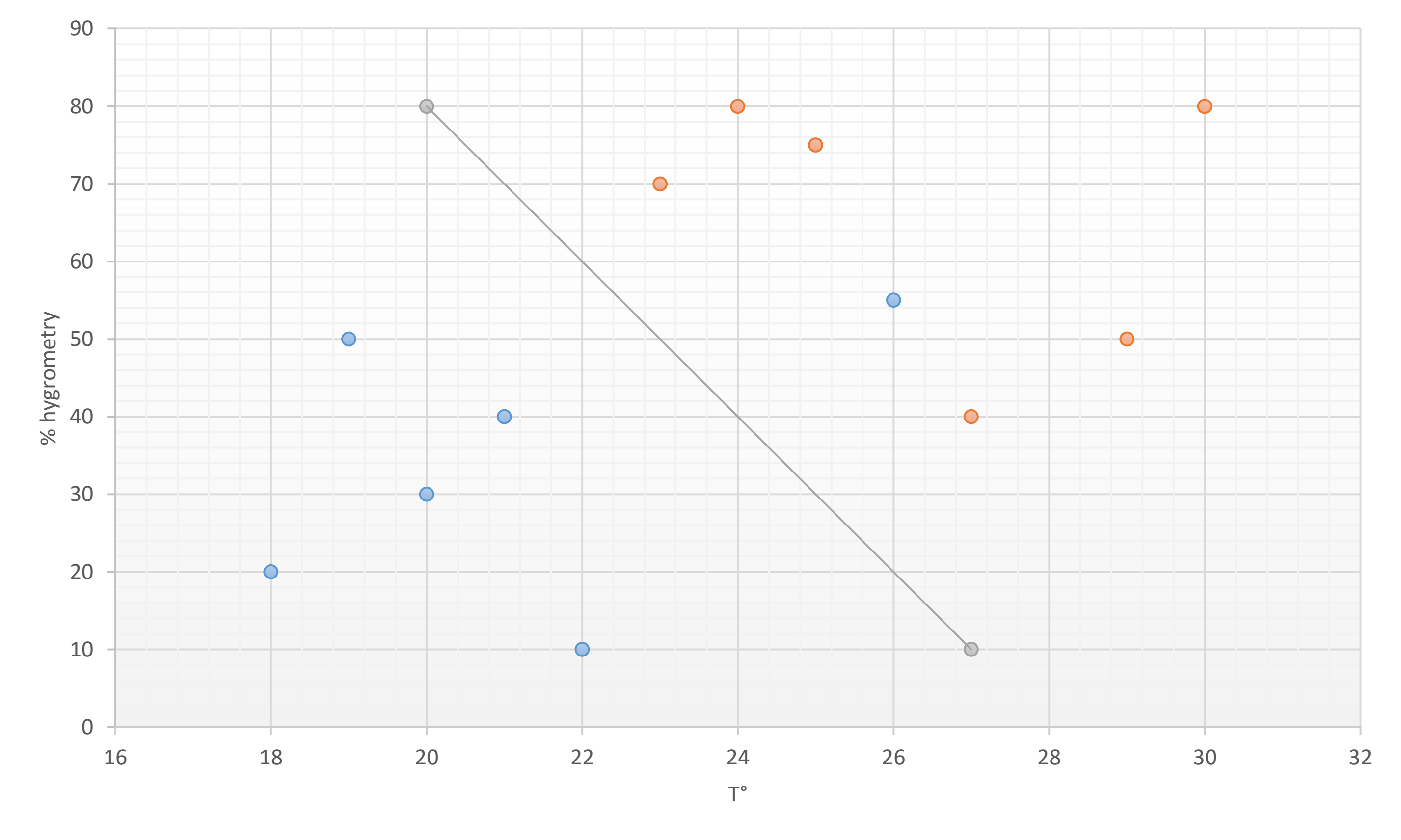
- Problem 1: You have a large inventory of identical items. You want to predict how many of them you will sell in the next 3 months
- Problem 2: You have large set of customer files and you want to determine which of them have been compromised or not
- Should you treat them?
  - ~~Both as classification problem~~
  - ~~Both as regression problem~~
  - ~~Pbm 1 as classification problem and Pbm 2 as regression problem~~
  - <u>Pbm 1 as regression problem and Pbm 2 as classification problem</u>

Unsupervised learning

YOU DON'T KNOW WHAT YOU LEARN!

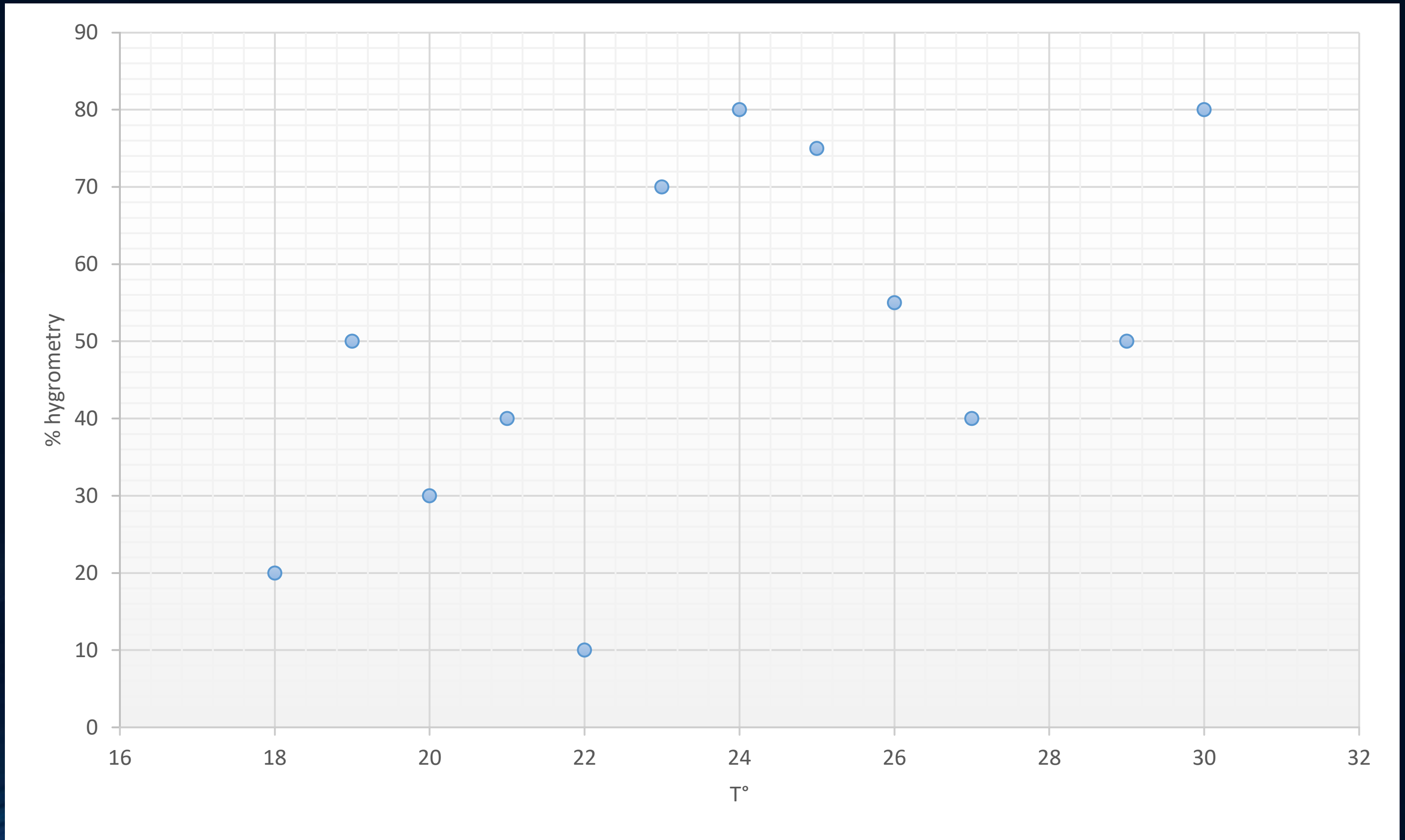
Unsupervised learning

- Retake the example of supervised learning: healthy habitat



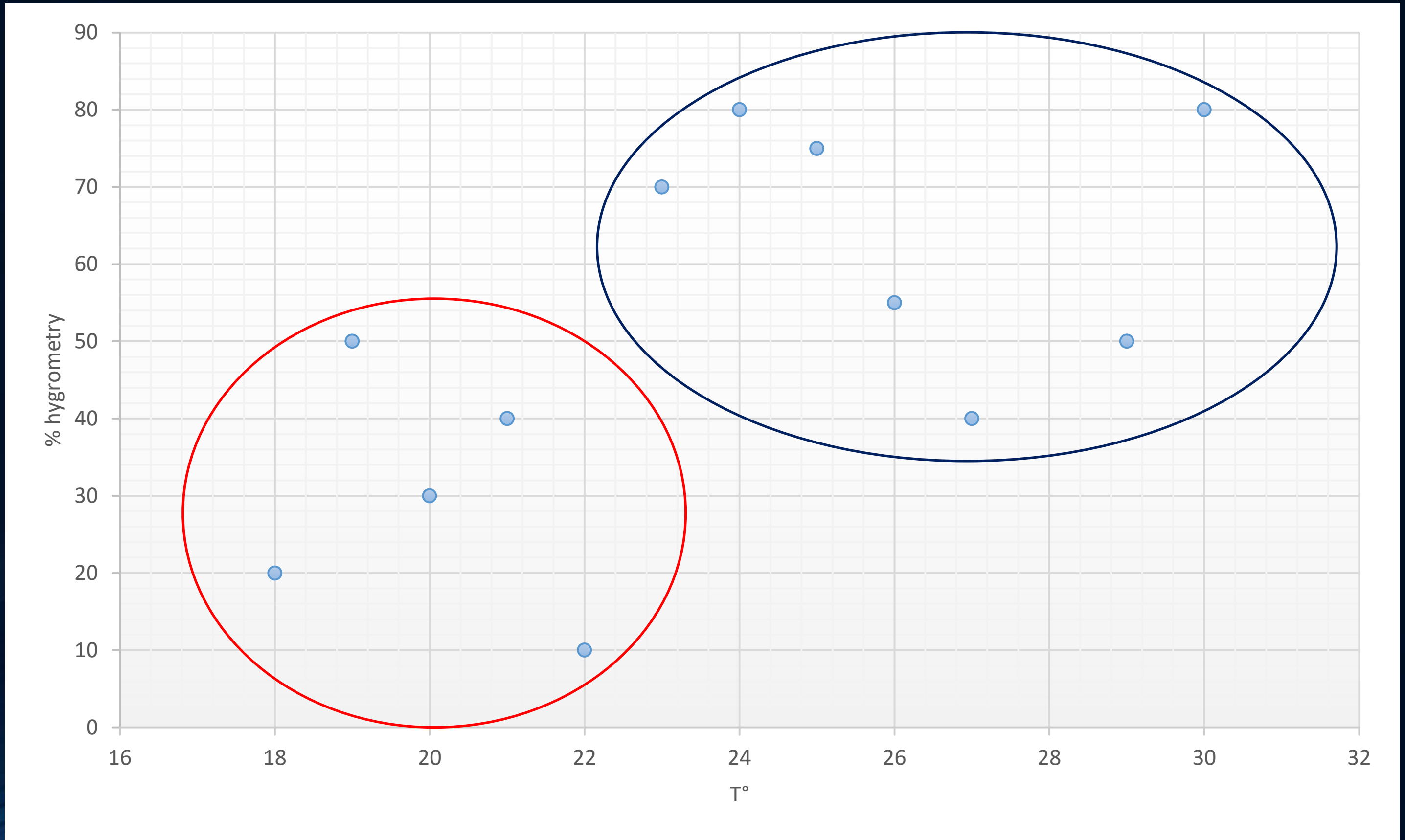
Unsupervised learning

- Retake the example of supervised learning: healthy habitat
  - For each sample of data, we don't have any labels



Unsupervised learning

- Retake the example of supervised learning: healthy habitat
  - For each sample of data, we don't have any labels => Clustering

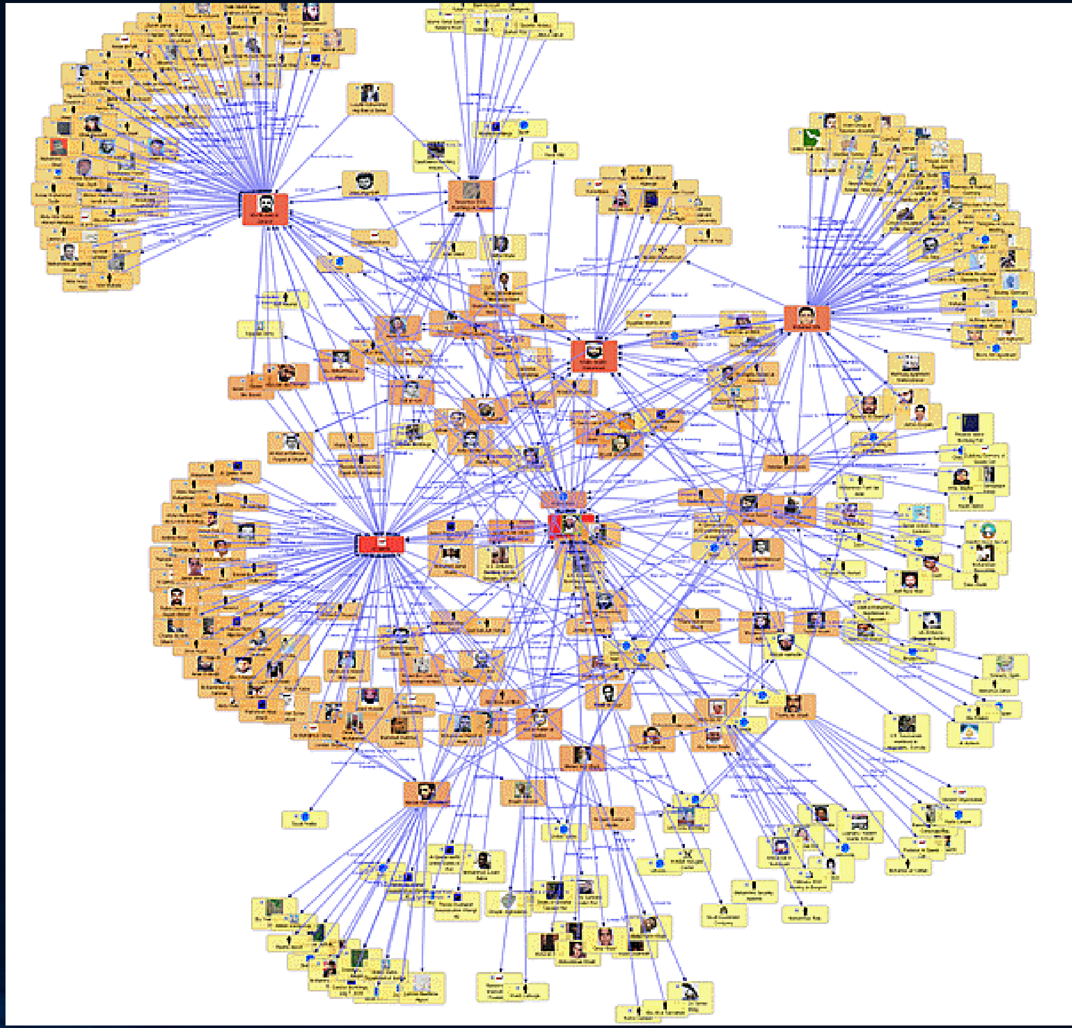


Unsupervised learning

- Clustering algorithm
  - No label: you don’t tell the algorithm which story is similar or the same to another
  - Cluster or gather data which seems to be similar
  - It also means separate data which seems to be too different

Unsupervised learning: Clustering algorithm

- Examples:
  - Google news (https://news.google.fr/)
  - Social network analysis (Facebook, Google+…)



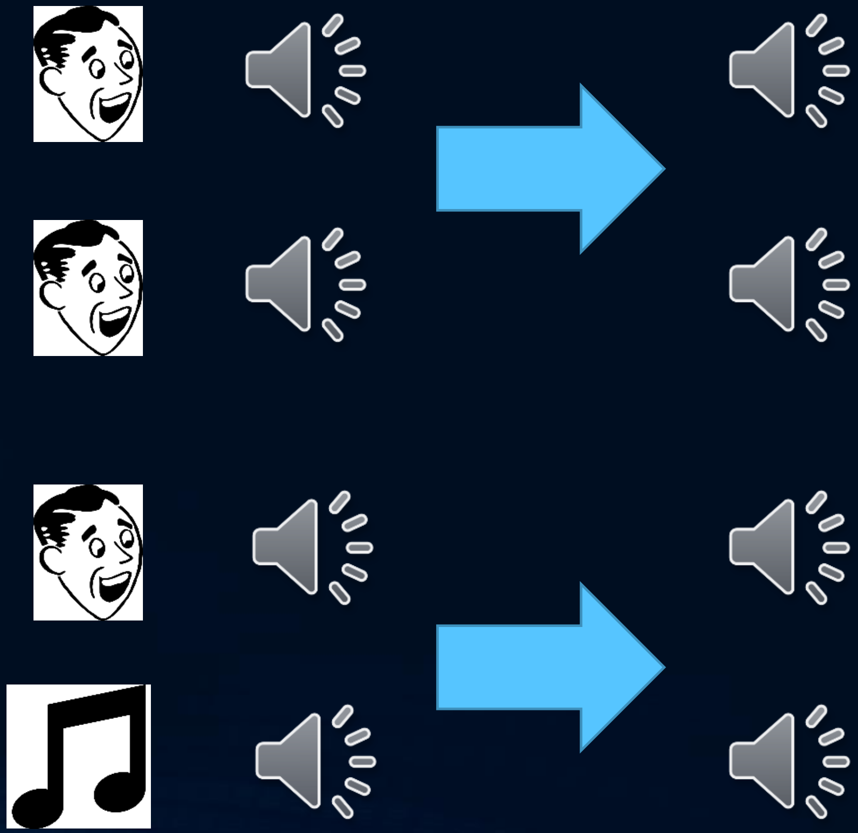
Unsupervised learning: Clustering algorithm

- Example of blind audio sources separation
  - 2 voices:

(courtesy of Te-Won Lee)

  - Voice + music:

(courtesy of Lucas Parra)



Unsupervised learning

- Of the following examples which would you address using an unsupervised algorithm?
  - Given a set of emails labelled as spam/not spam, learn a spam filter
  - Given a set of news article, group them into set of articles about the same story
  - Given a database of customer data, automatically discovers target market segments and group customers into them
  - Given a dataset of patients diagnoses having diabetes or not, learn an algorithm to group patients as having diabetes or not

Unsupervised learning

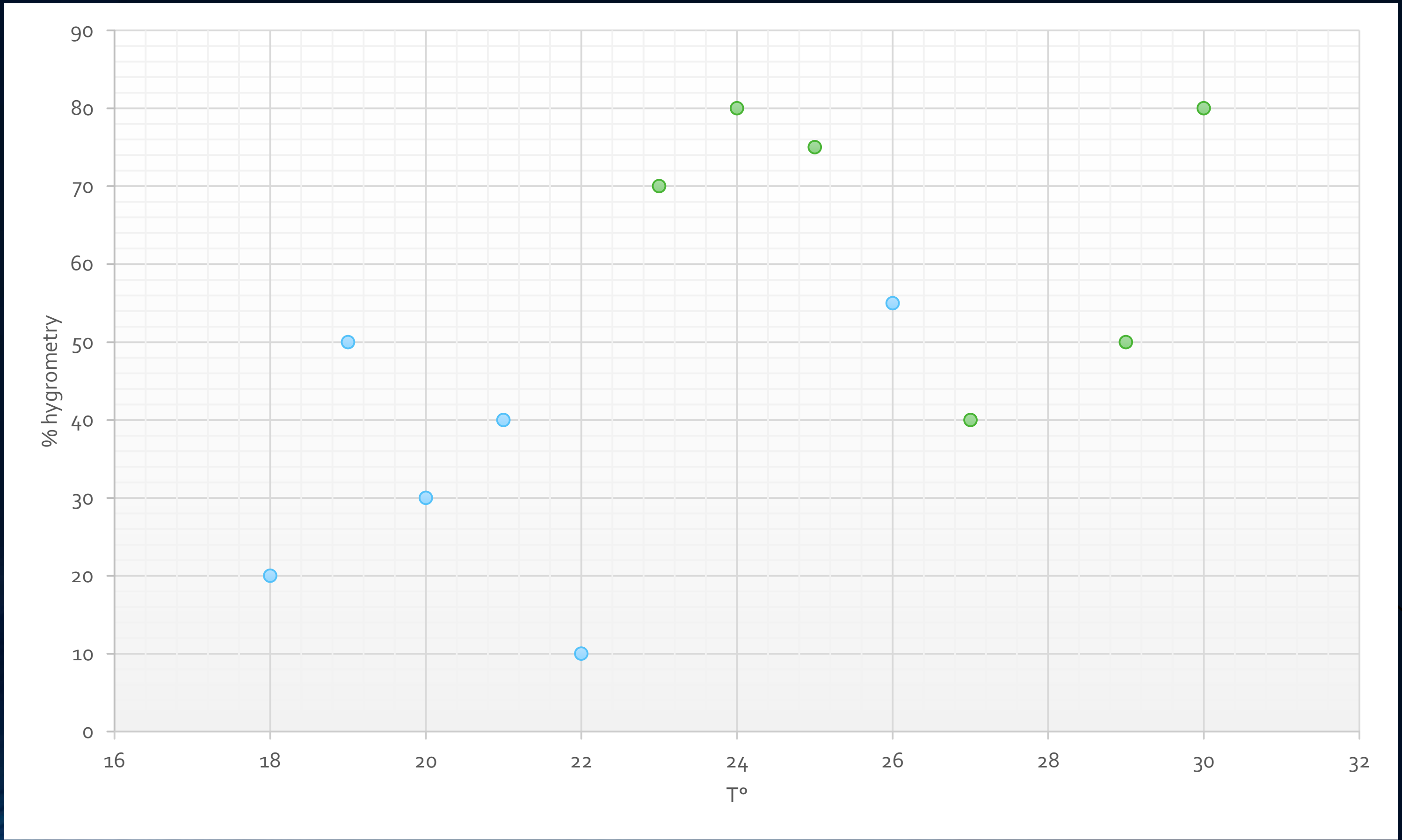
- Of the following examples which would you address using an unsupervised algorithm?
  - ~~Given a set of emails labelled as spam/not spam, learn a spam filter~~
  - Given a set of news article, group them into set of articles about the same story
  - Given a database of customer data, automatically discovers target market segments and group customers into them
  - ~~Given a dataset of patients diagnoses having diabetes or not, learn an algorithm to group patients as having diabetes or not~~

Non-linear classification problem

LET’S FIND SOMETHING SOMEWHERE!

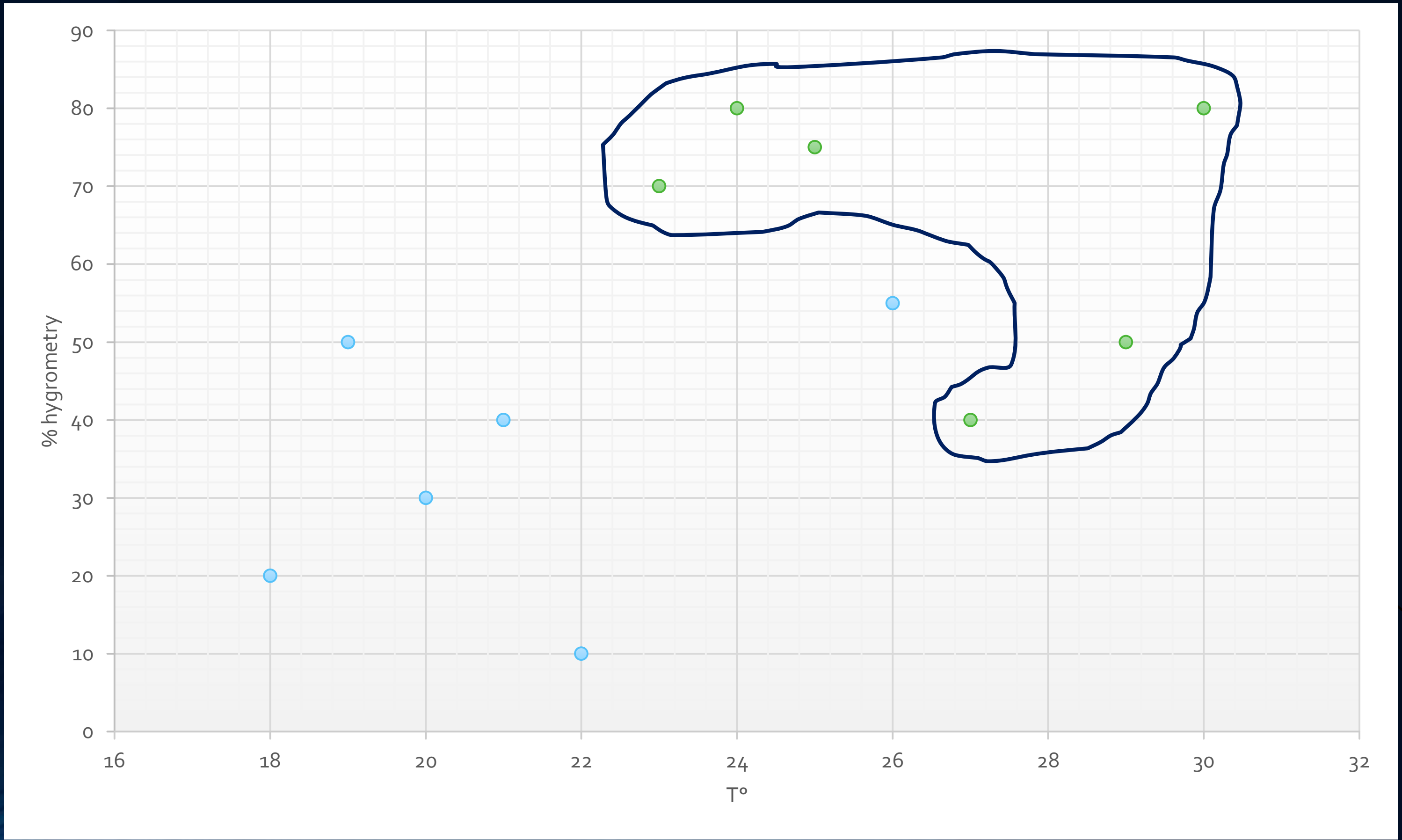
Non-linear classification problem

Non-linear Clustering



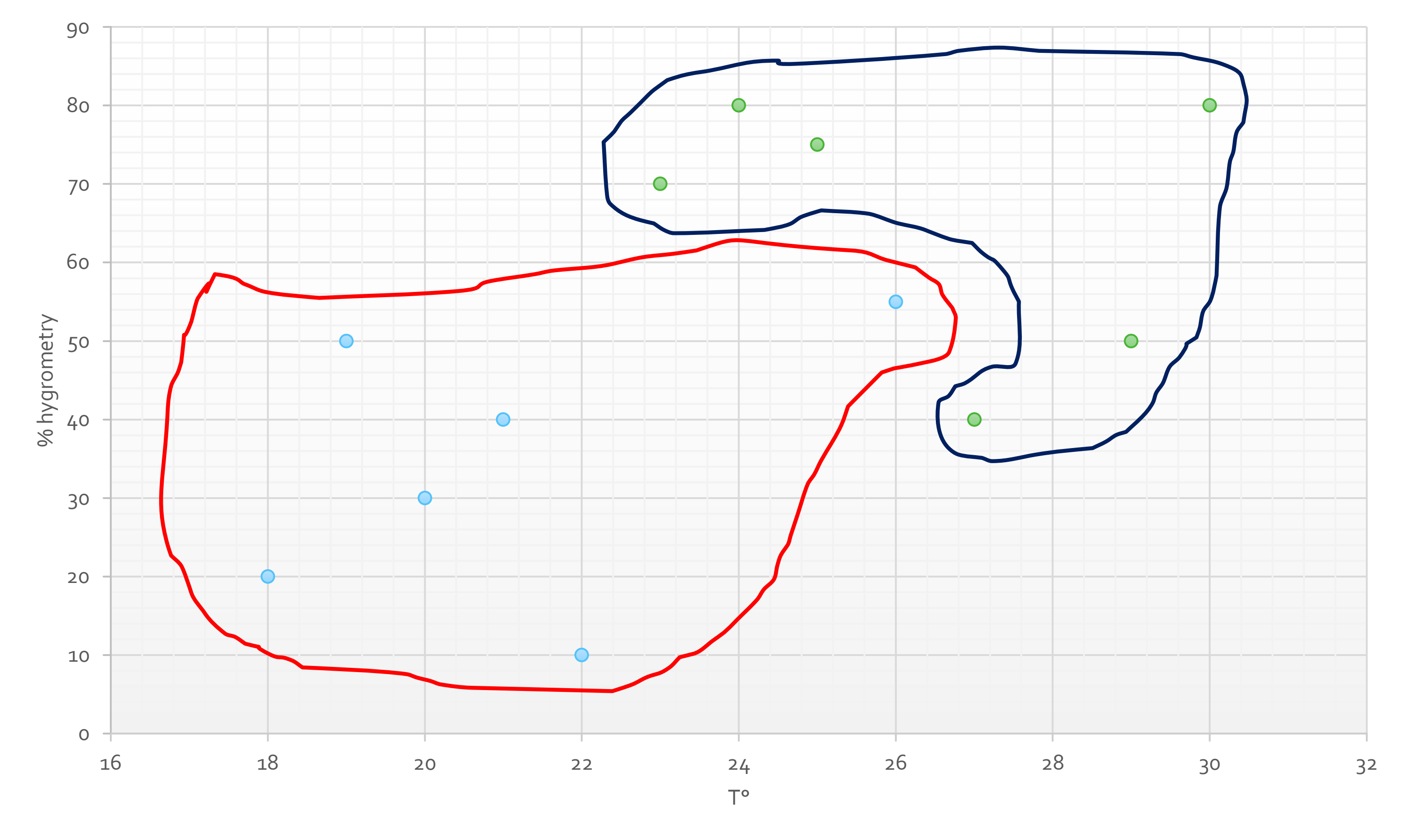
Non-linear classification problem

Non-linear Clustering



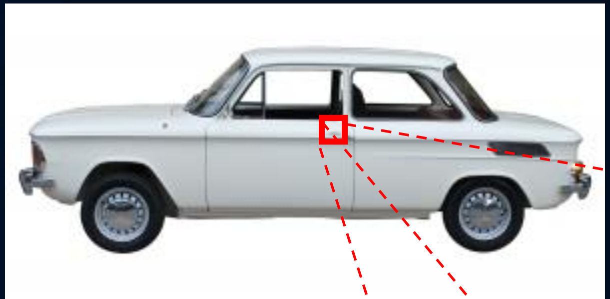
Non-linear classification problem

Non-linear Clustering



Non-linear classification problem

Computer vision: Car detection

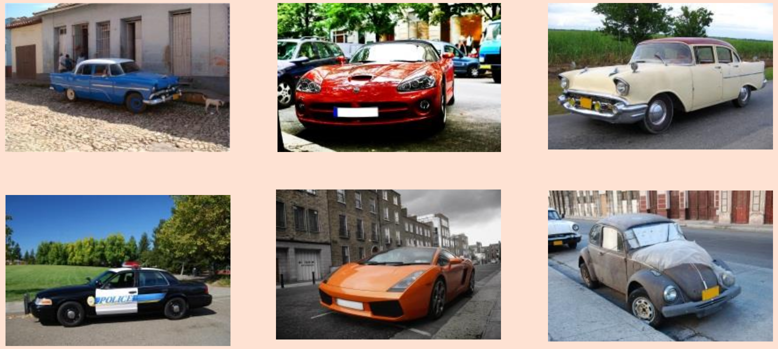


But the camera sees this:

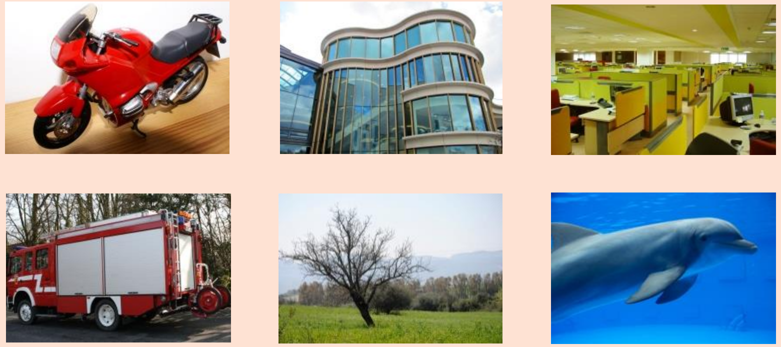
194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

Non-linear classification problem

Computer vision: Car detection



Cars



Not a car

Non-linear classification problem

Solve that kind of problems:

- Support Vector Machines
  - Well known
  - Efficient
- Neural Networks
  - New trend
  - Very efficient (new State of the Art)

Neural Network

A FIRST STEP TO DEEP LEARNING

Neural Network

- From biology to logic
- Multi-class classification
- Propagations
- Recurrent Model
- Long-Short Term Memory

Neural Network: from biology to logic

FROM THE HUMAN BRAIN TO THE COMPUTER!

Introduction

- Also called (multi-layer) perceptron
- Origins: algorithm that try to mimic the brain
- Widely used during the 80’s and early 90’s
- Recent resurgence: due to technical advances
- State of the Art technique for many applications

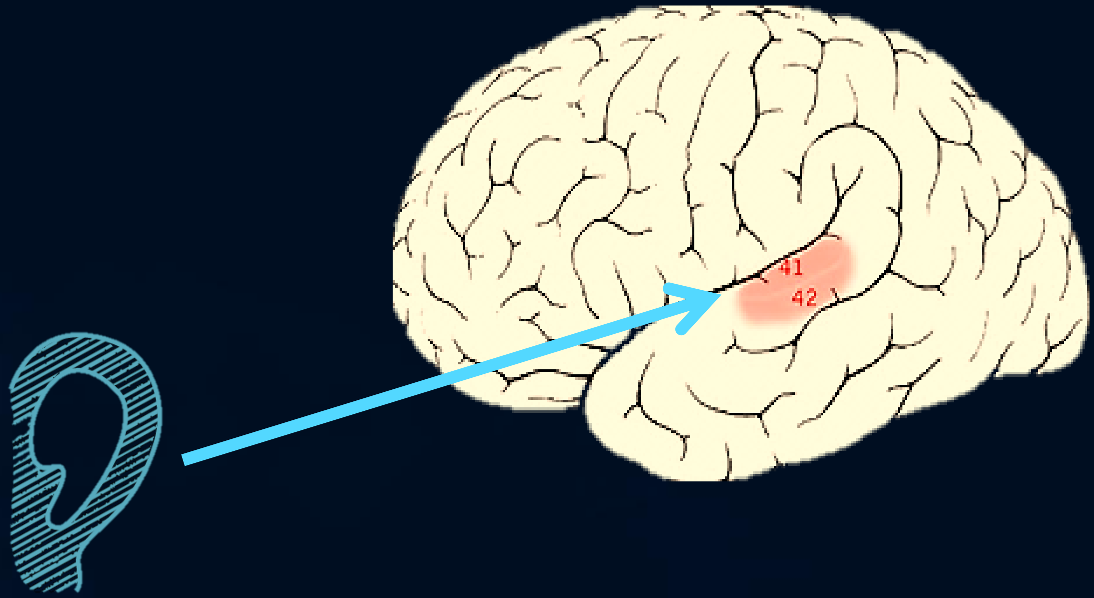
A Single algorithm to learn?

- Brain learn and process many things
  - Images, sounds, touch, taste…
  - Read, speak, maths, sciences…

$\Rightarrow$Only the same process to learn everything!

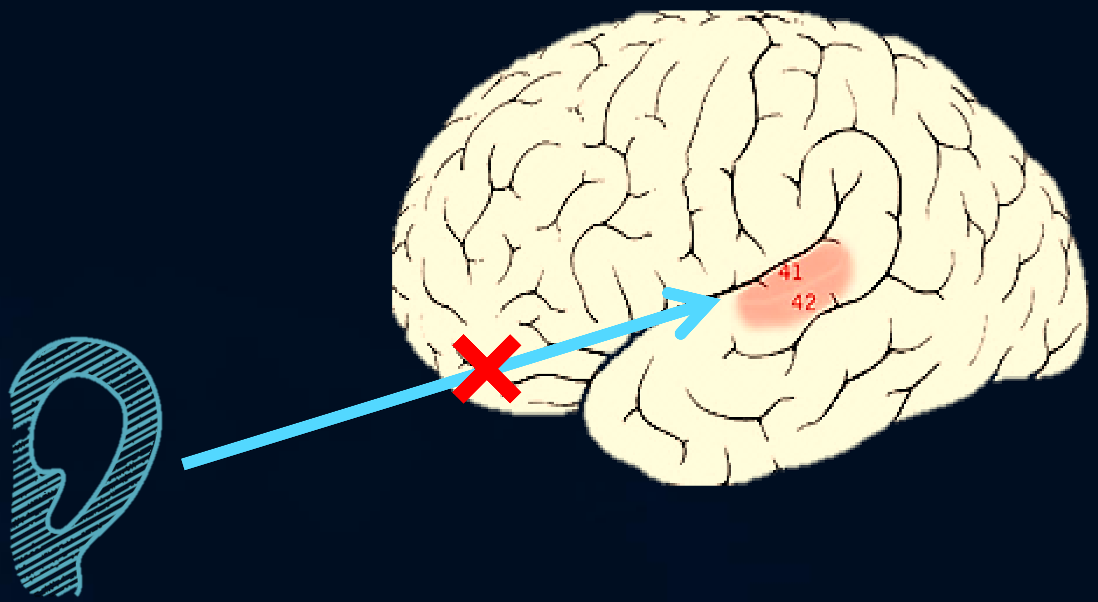
Sense replacement

Audiotory cortex learns to see



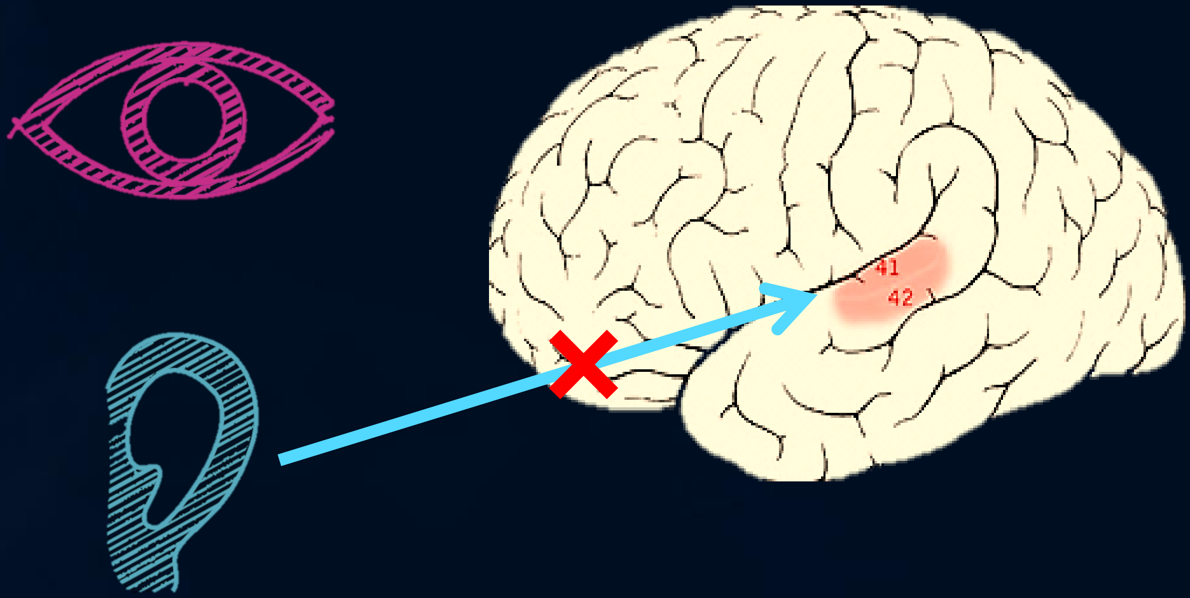
Sense replacement

Audiotory cortex learns to see



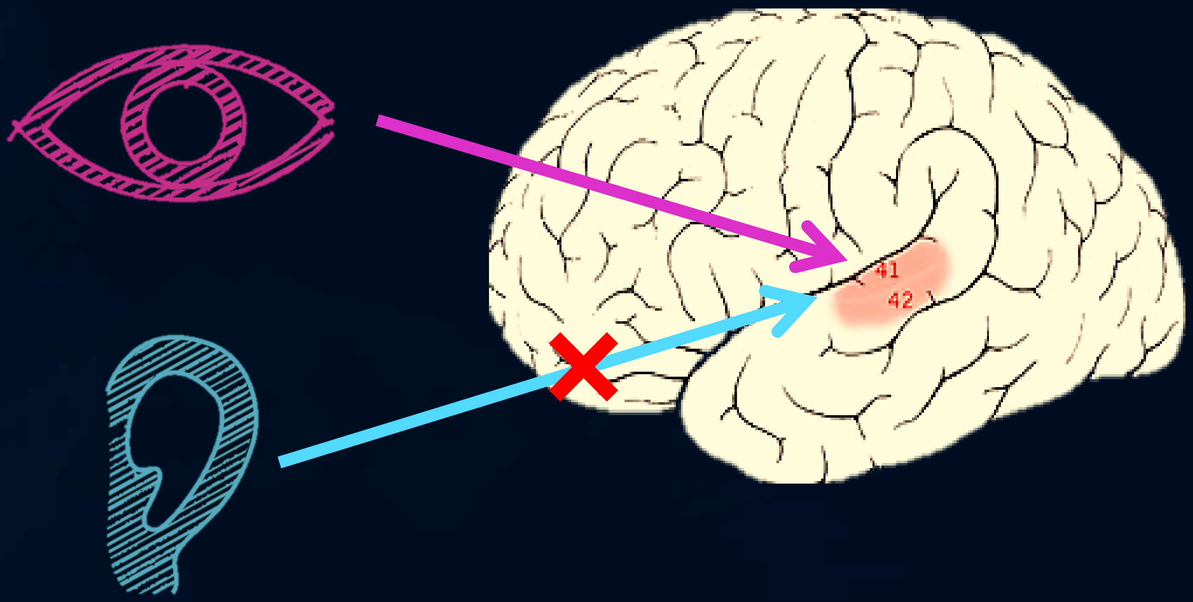
Sense replacement

Audiotory cortex learns to see



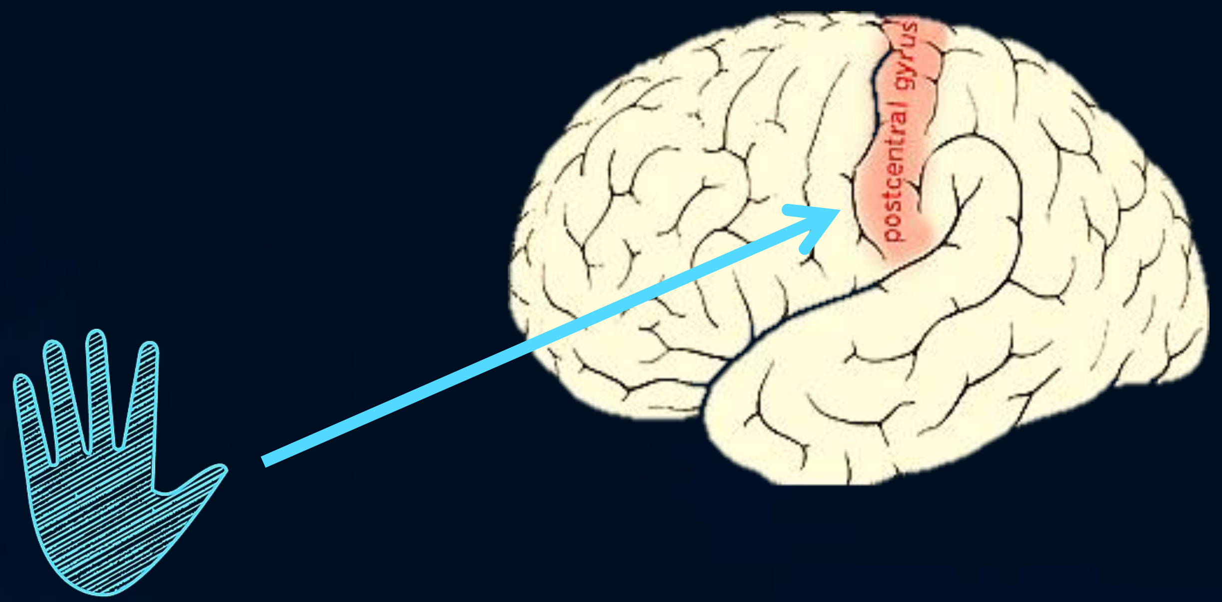
Sense replacement

Audiotory cortex learns to see



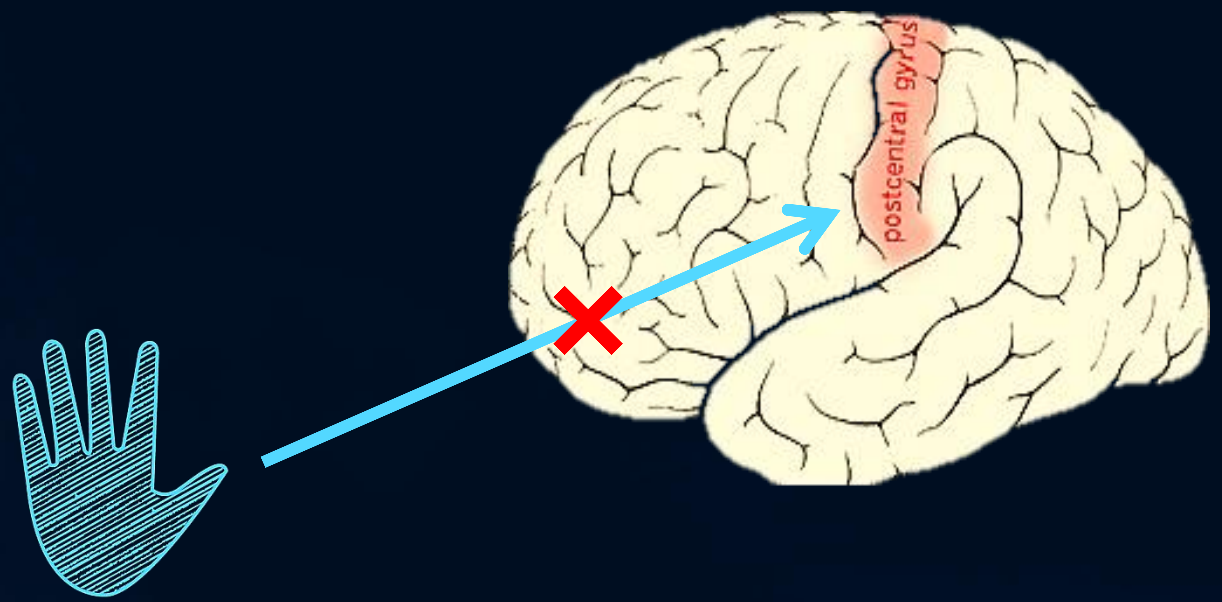
Sense replacement

Sensomatory cortex learns to see



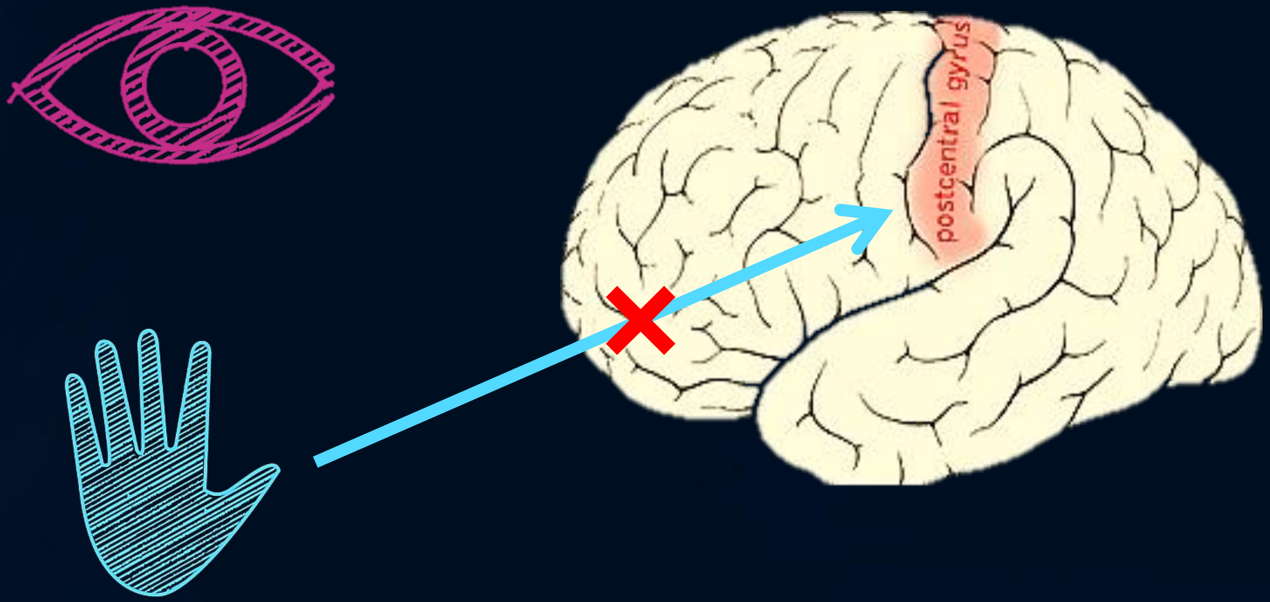
Sense replacement

Sensomatory cortex learns to see



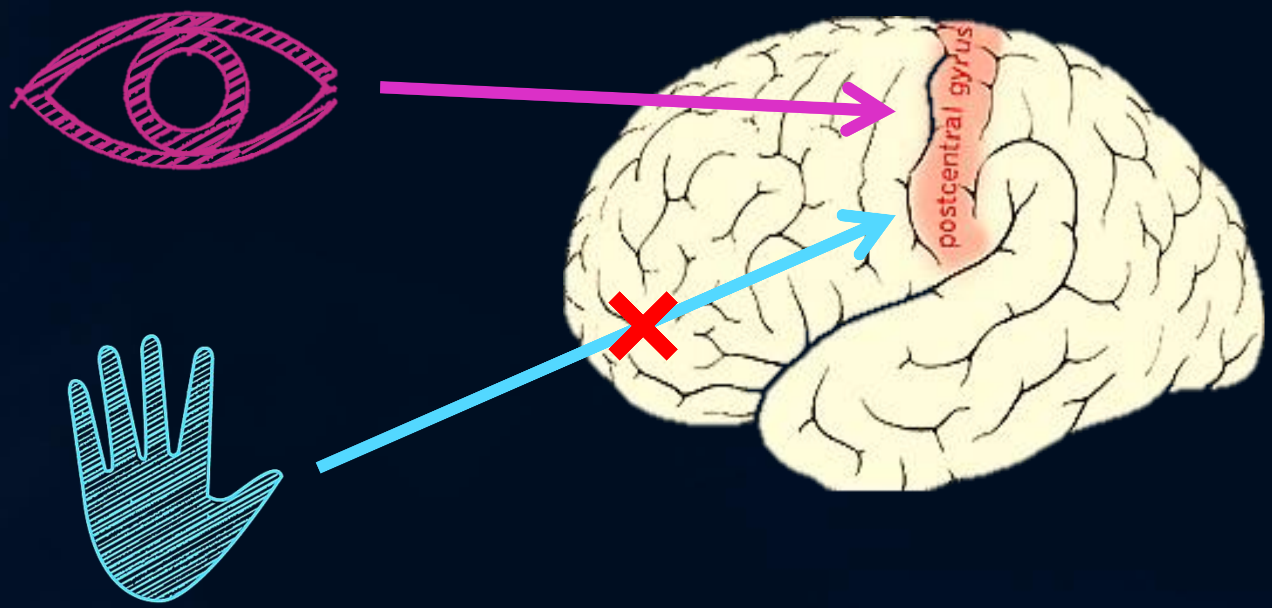
Sense replacement

Sensomatory cortex learns to see



Sense replacement

Sensomatory cortex learns to see



Sensors for the Brain

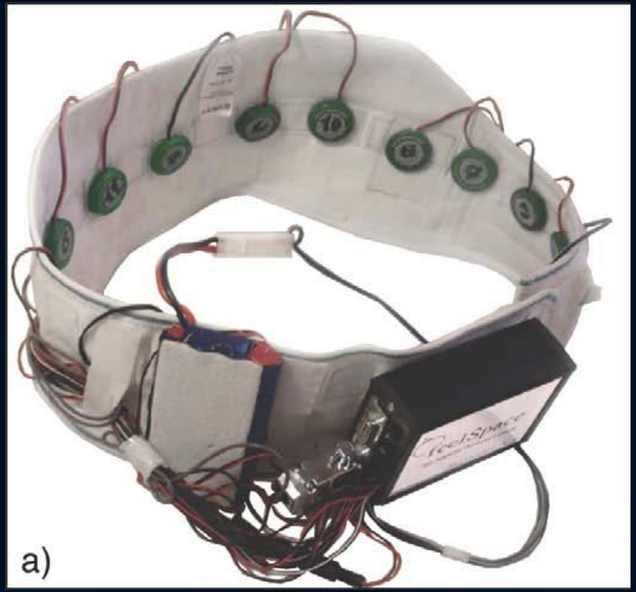
Seeing with you tongue

Sensors for the Brain

Echolocation (Sonar)

Sensors for the Brain

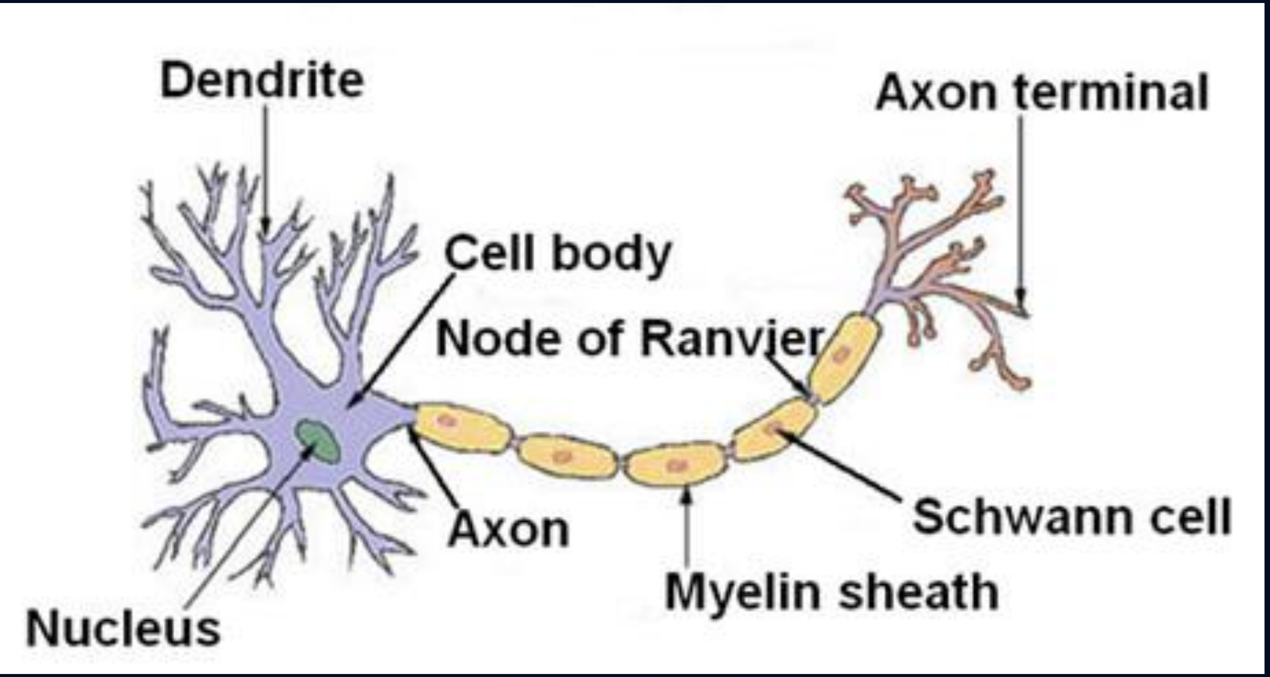
Haptic belt: Direction sense



Neurons in the brain

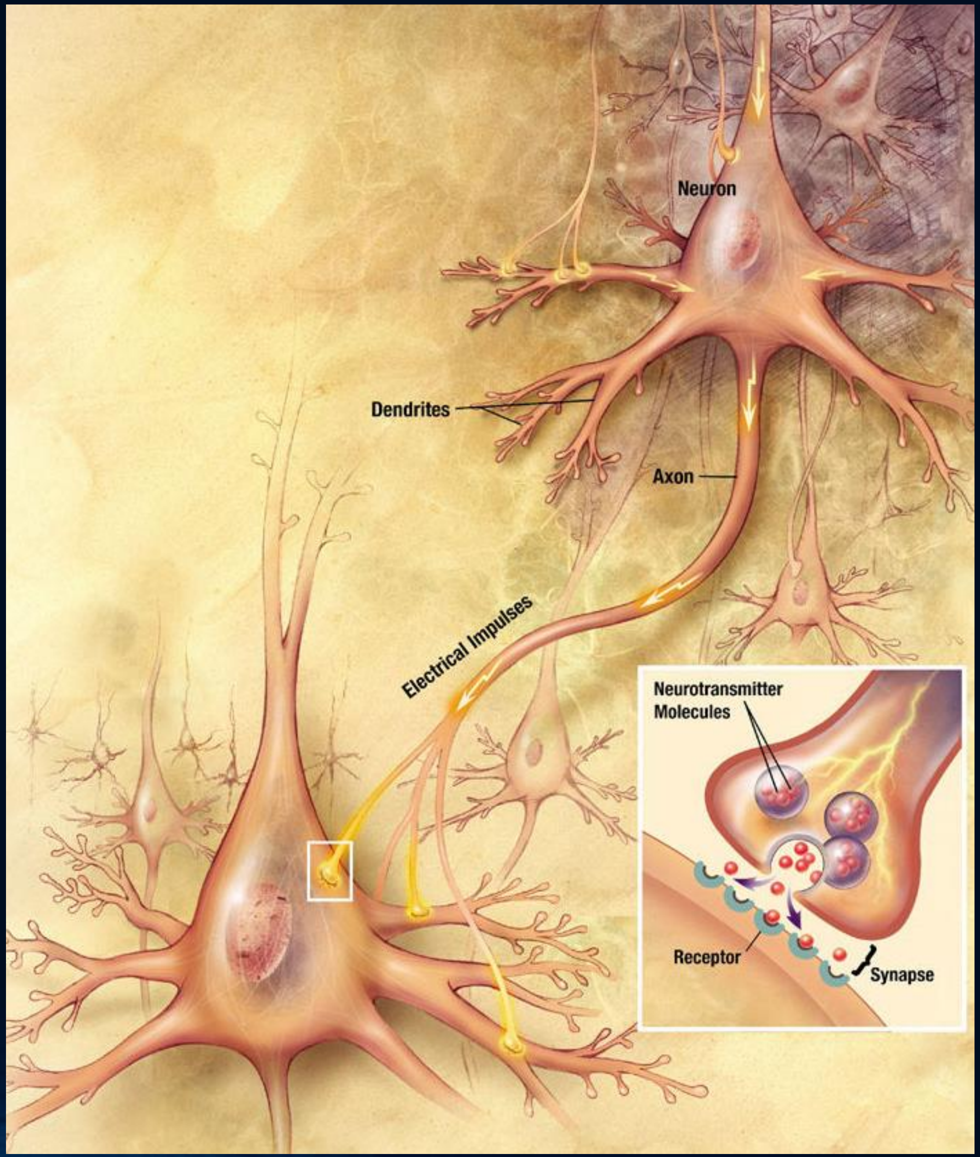
- Input: Dendrites

- Output: Axon



Neurons in the brain

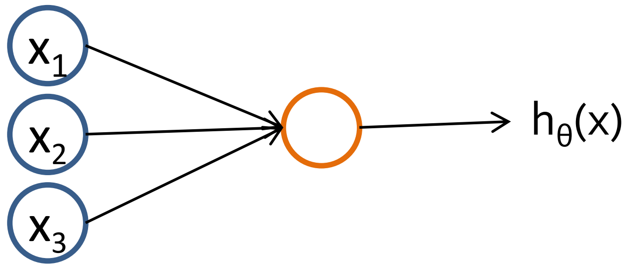
- Input: Dendrites
  - Senses (vision, audio…)
  - Other neurons
  - …
- Output: Axon
  - Other neurons
  - Muscles
  - …



Neurons model

- Input: (x_0 = bias = constant)
  - x_1
  - x_2
  - …
- Output:
  - $h_\theta(x)$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

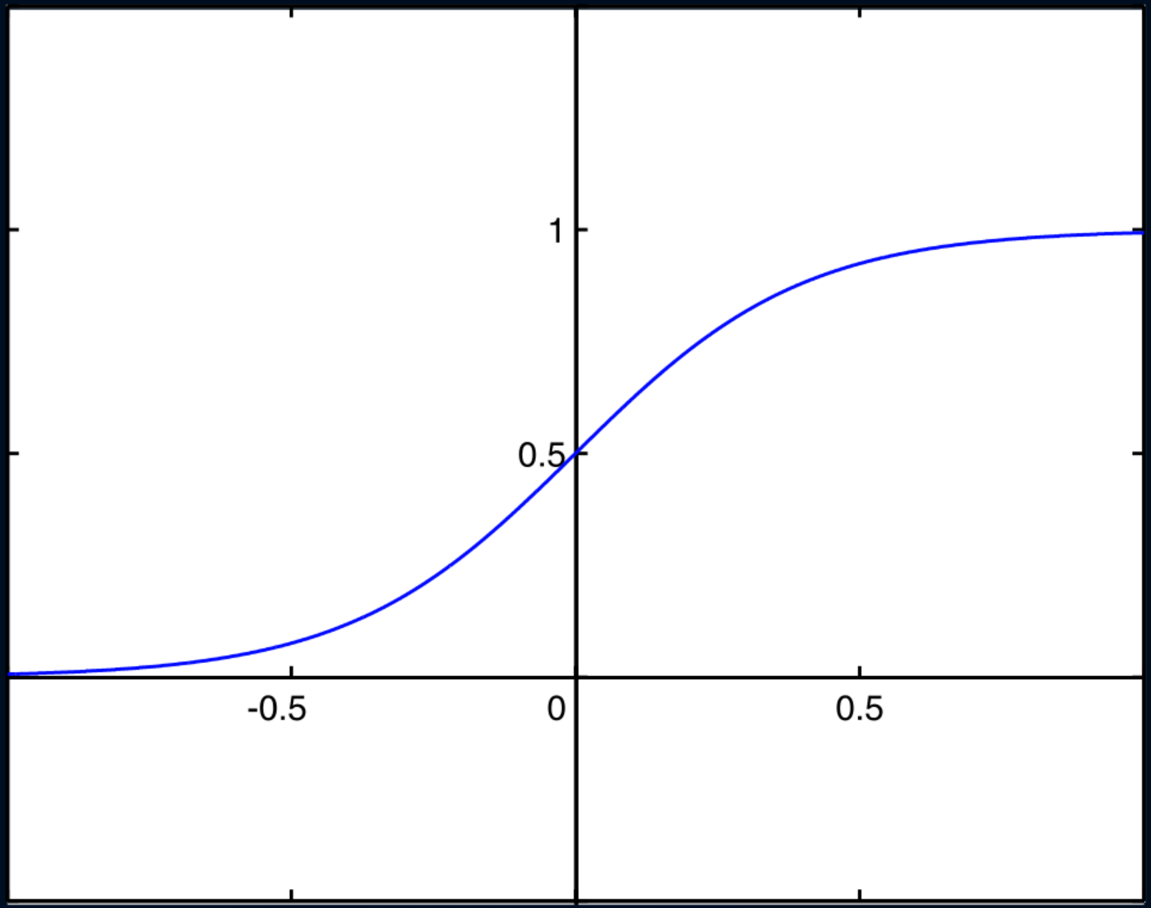


Sigmoid (logistic) activation function

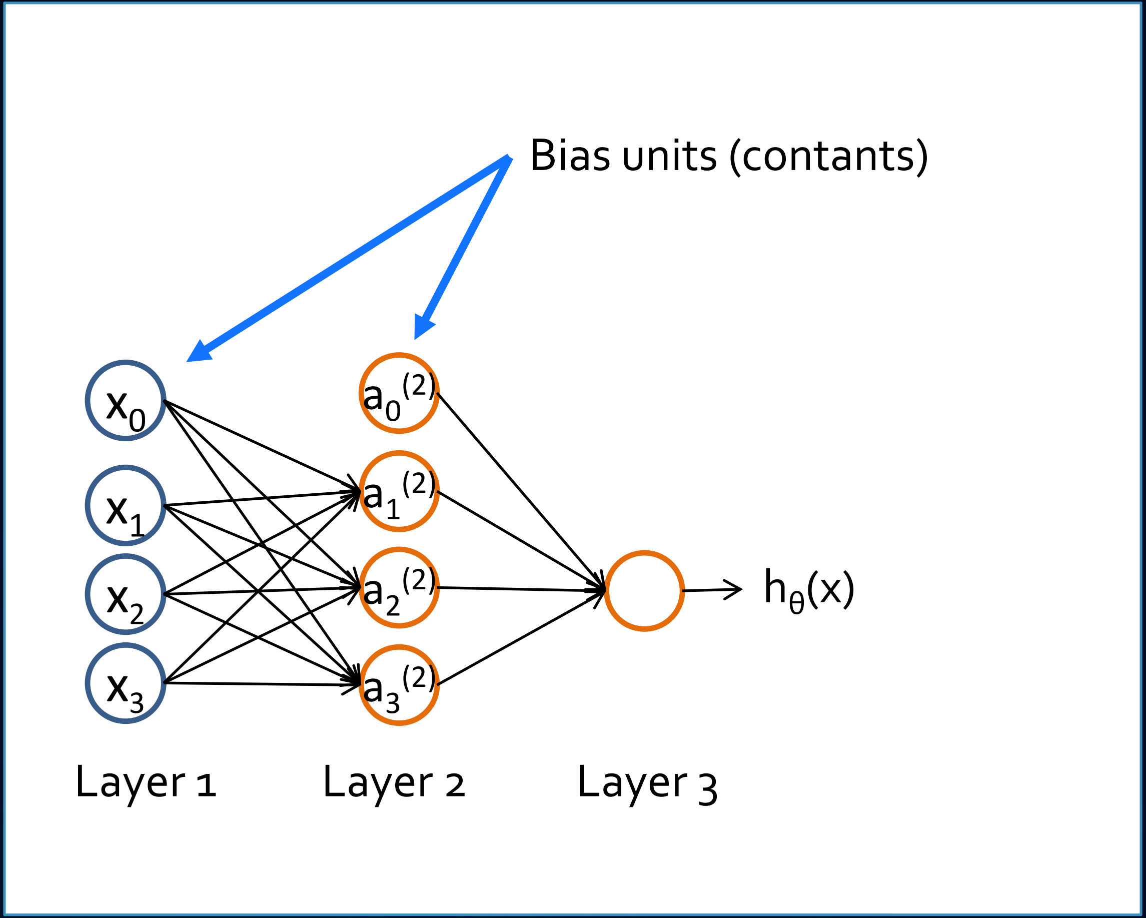
Neurons model

- Sigmoid activation function
  - Hyperbolic tangent:

$$\tanh = \frac{1}{1+e^{-\theta^T X}}$$

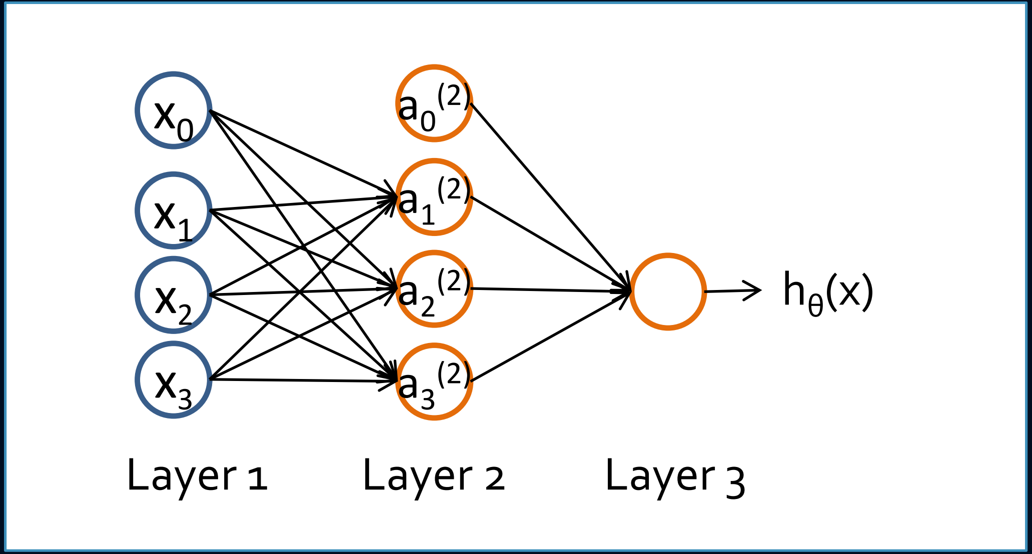


Neurons model



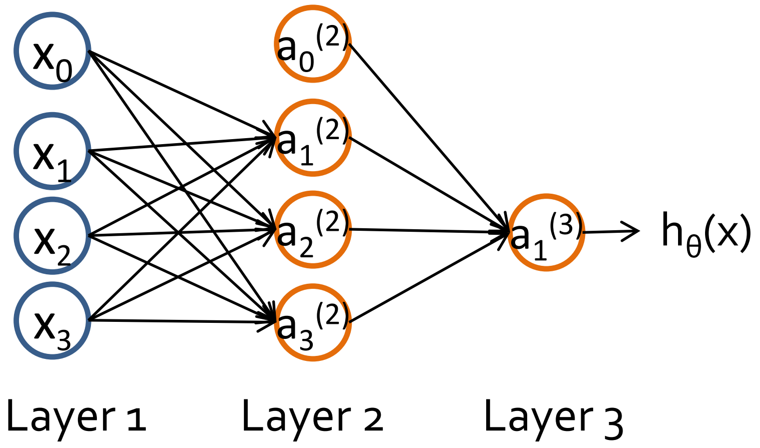
Neurons model

- $a_i^{(j)}$ = activation of unit *j* in layer *i*
- $\Theta^{(j)}$ = matrix of weights controlling function mapping from layer *j* to layer *j+1*



Neurons model

- $a_i^{(j)}$ = activation of unit *i* in layer *j*
- $\Theta^{(j)}$ = matrix of weights controlling function mapping from layer *j* to layer *j+1*
- If network has S_j units in layer *j*, S_{j+1} units in layer *j+1*, then $\Theta^{(j)}$ will be of dimension $(S_{j+1}) \times (S_j+1)$.

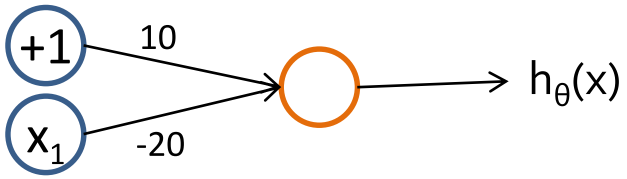


$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$
$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

Examples

- Logic function NOT:

$\Rightarrow h_\theta(x)=g(10-20x_1)$

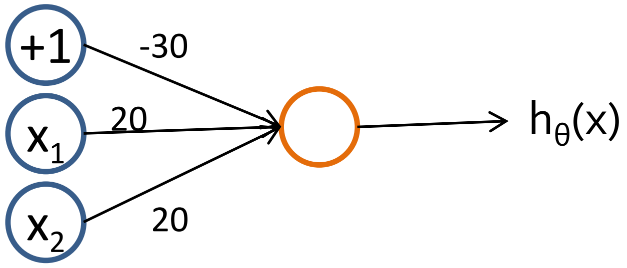


x_1	$h_\Theta(x)$
0	
1	

Examples

- Logic function AND:

$\Rightarrow h_\theta(x)=g(-30+20x_1+20x_2)$

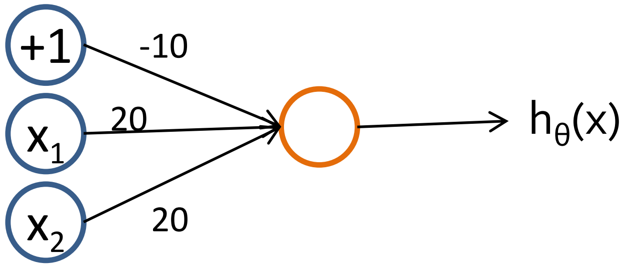


x_1	x_2	$h_\Theta(x)$
0	0	
0	1	
1	0	
1	1	

Examples

- Logic function OR:

$\Rightarrow h_\theta(x)=g(-10+20x_1+20x_2)$



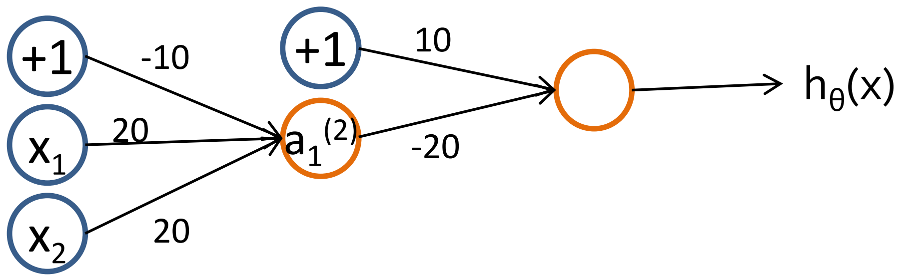
x_1	x_2	$h_\Theta(x)$
0	0	
0	1	
1	0	
1	1	

Examples

- Logic function NOR:

$\Rightarrow a_1^{(2)}=g(-10+20x_1+20x_2)$

$\Rightarrow h_\theta(x)=g(10-20(a_1^{(2)}))$

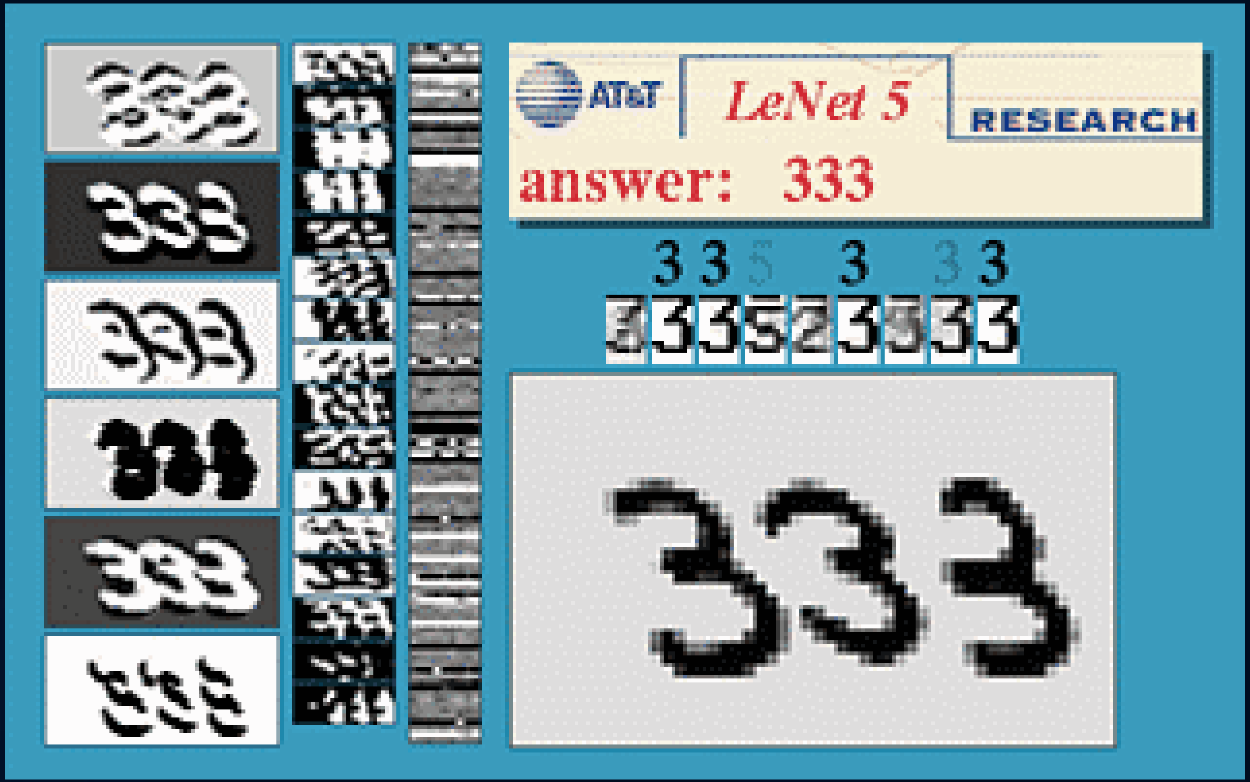


x_1	x_2	$a_1^{(2)}$	$h_\Theta(x)$
0	0		
0	1		
1	0		
1	1		

Neural Network: Multi-class classification

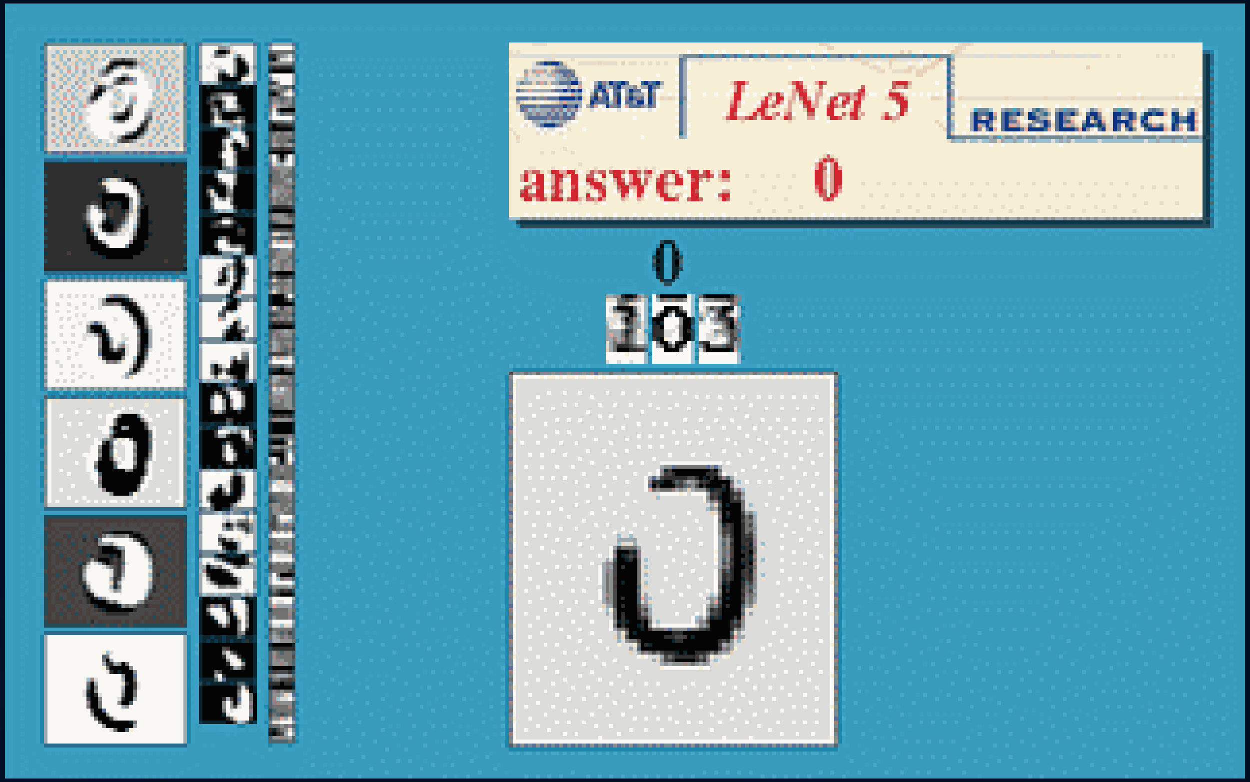
LET'S CLASSIFY LIKE THE BRAIN!

Examples of multiple output units: handwritten character



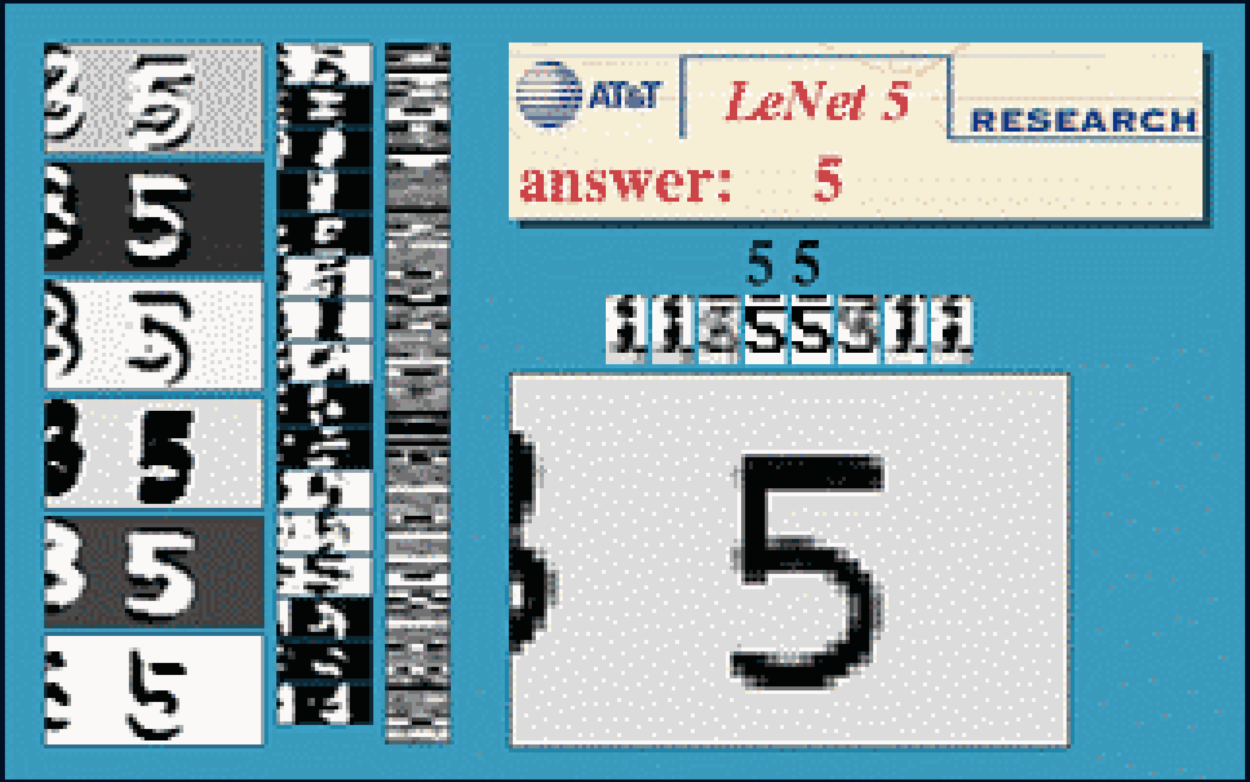
Courtesy of Yann LeCun

Examples of multiple output units: handwritten character



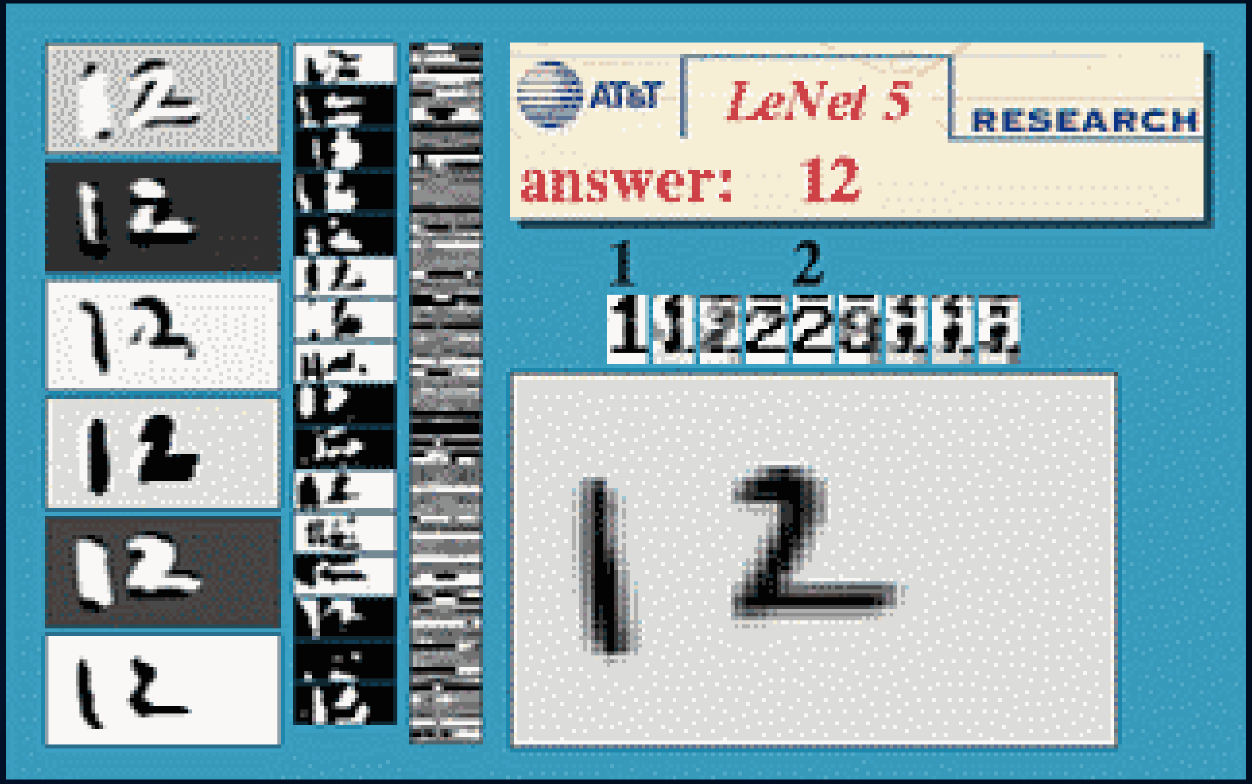
Courtesy of Yann LeCun

Examples of multiple output units: handwritten character



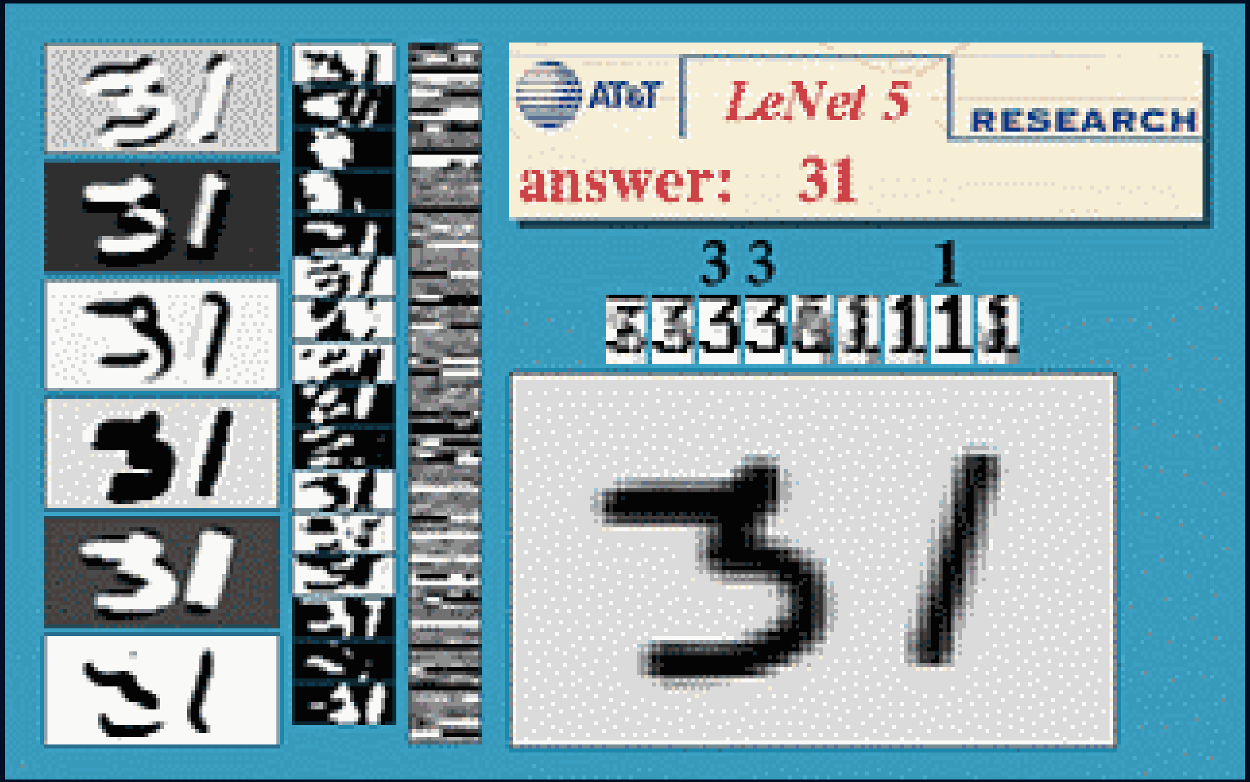
Courtesy of Yann LeCun

Examples of multiple output units: handwritten character



Courtesy of Yann LeCun

Examples of multiple output units: handwritten character

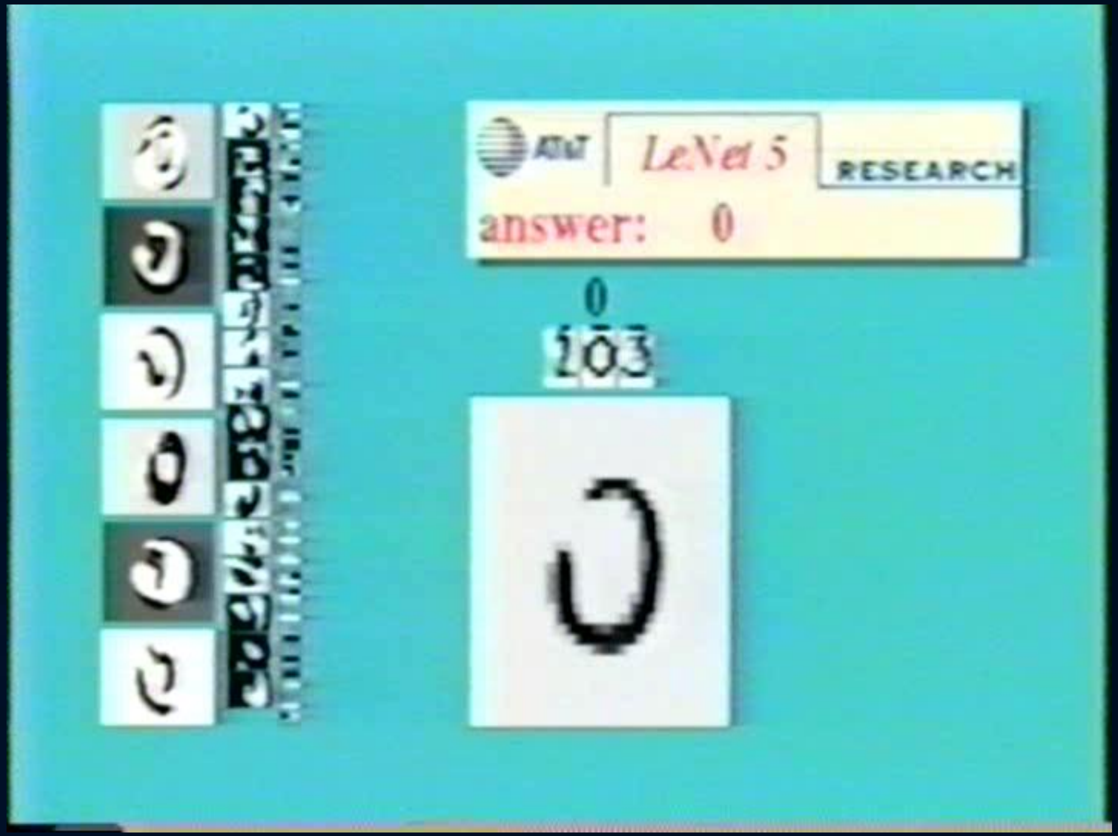


Courtesy of Yann LeCun

Examples of multiple output units: handwritten character

Courtesy of Yann LeCun

Examples of multiple output units: handwritten character



Courtesy of Yann LeCun

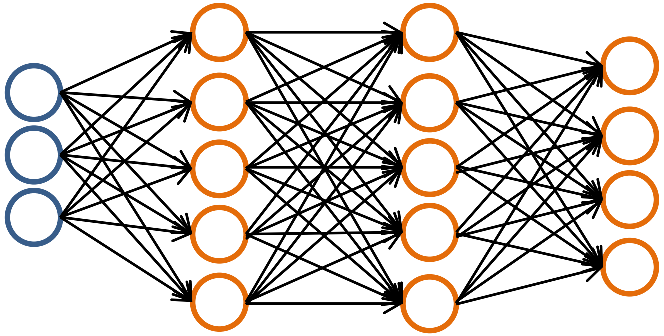
Examples of multiple output units: image classification

Pedestrian

Car

Motorcycle

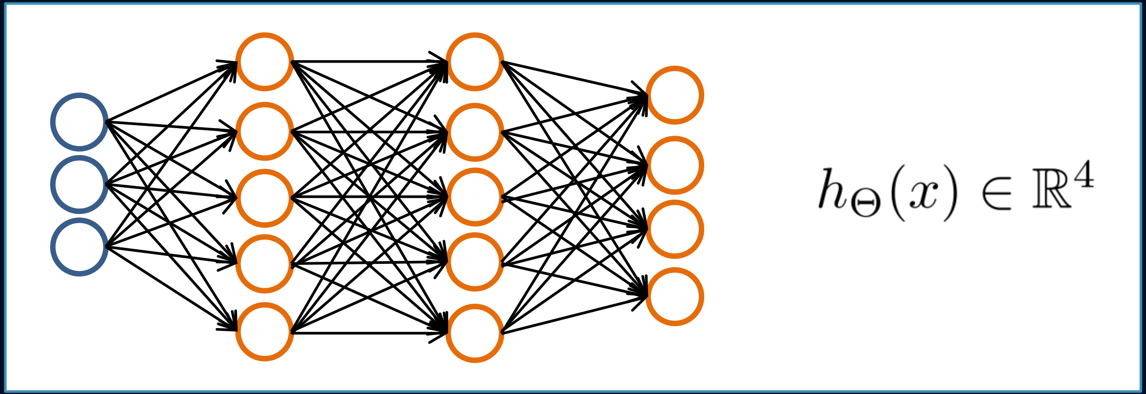
Truck



$$h_\Theta(x) \in \mathbb{R}^4$$

Examples of multiple output units: image classification

- $h_\Theta(x) \approx \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$ when pedestrian



- $h_\Theta(x) \approx \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$ when car

...

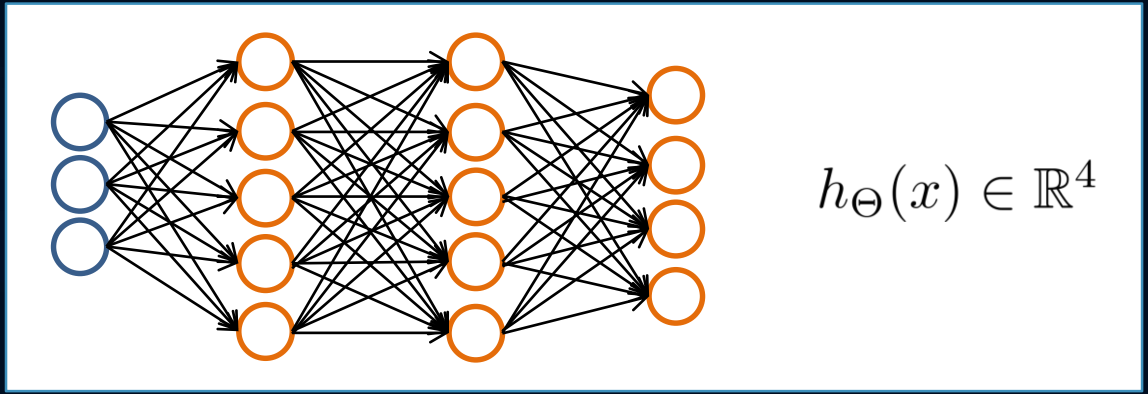

Pedestrian


Car


Motorcycle

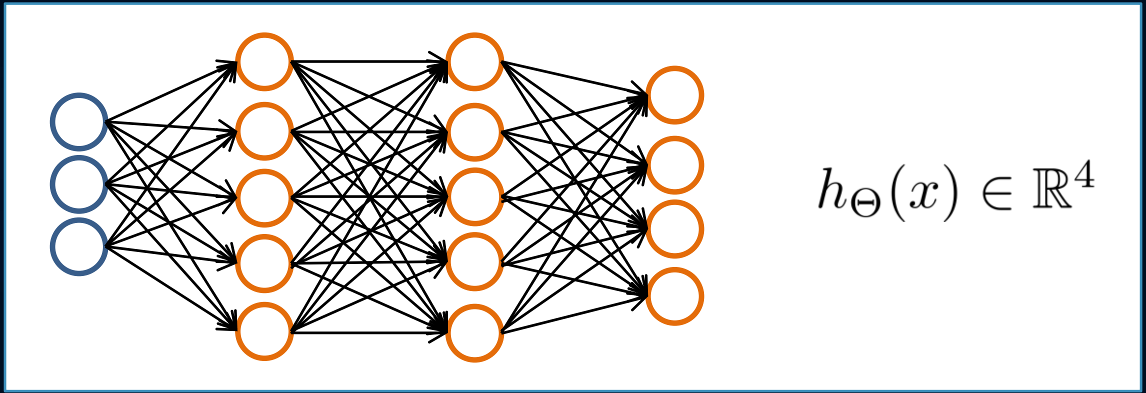

Truck

Examples of multiple output units: image classification



- Training set would be $(x^{(1)},y^{(1)})$, $(x^{(2)},y^{(2)})$, … $(x^{(m)},y^{(m)})$.
- $y^{(i)}$ one of $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

Neural network for classification



- Binary classification

y = 0 or 1

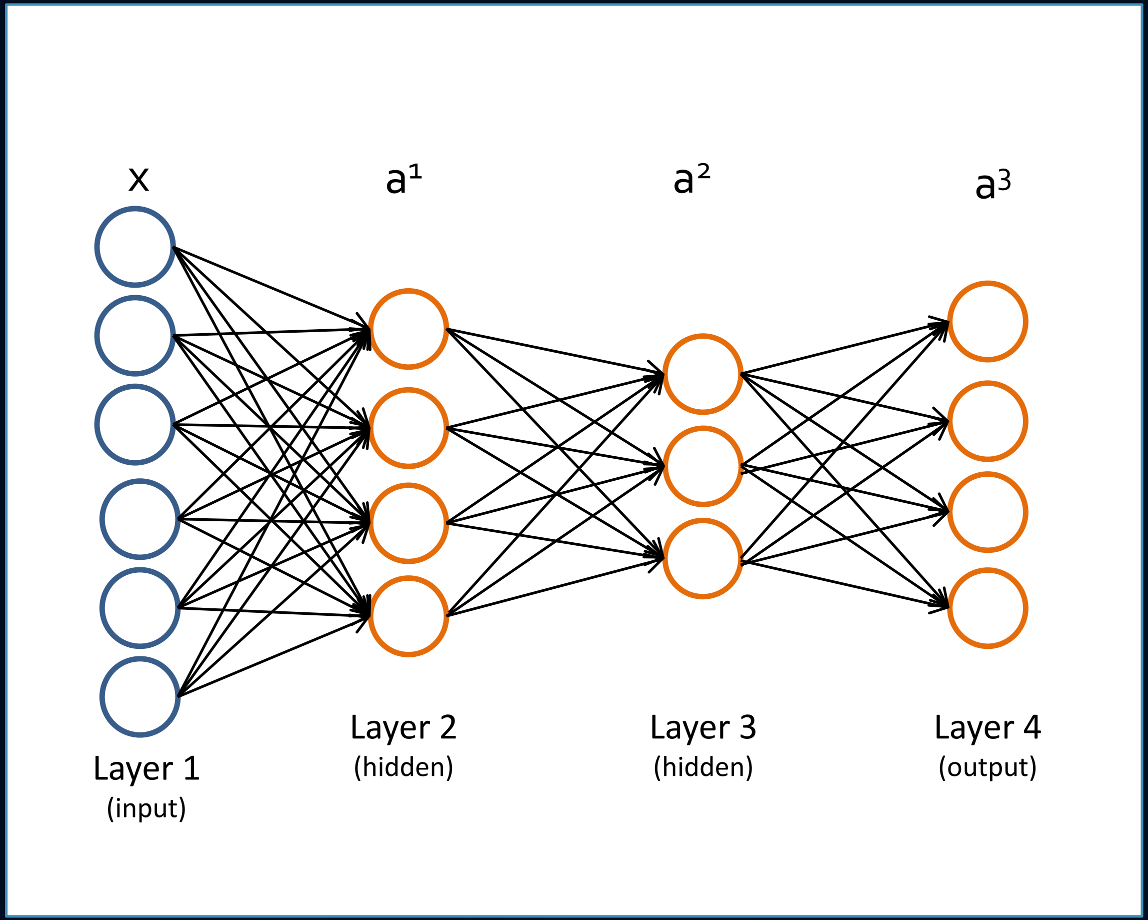
- Multi-class classification

$$y \in \mathbb{R}^K \text{; } e.g.\ K=4: \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

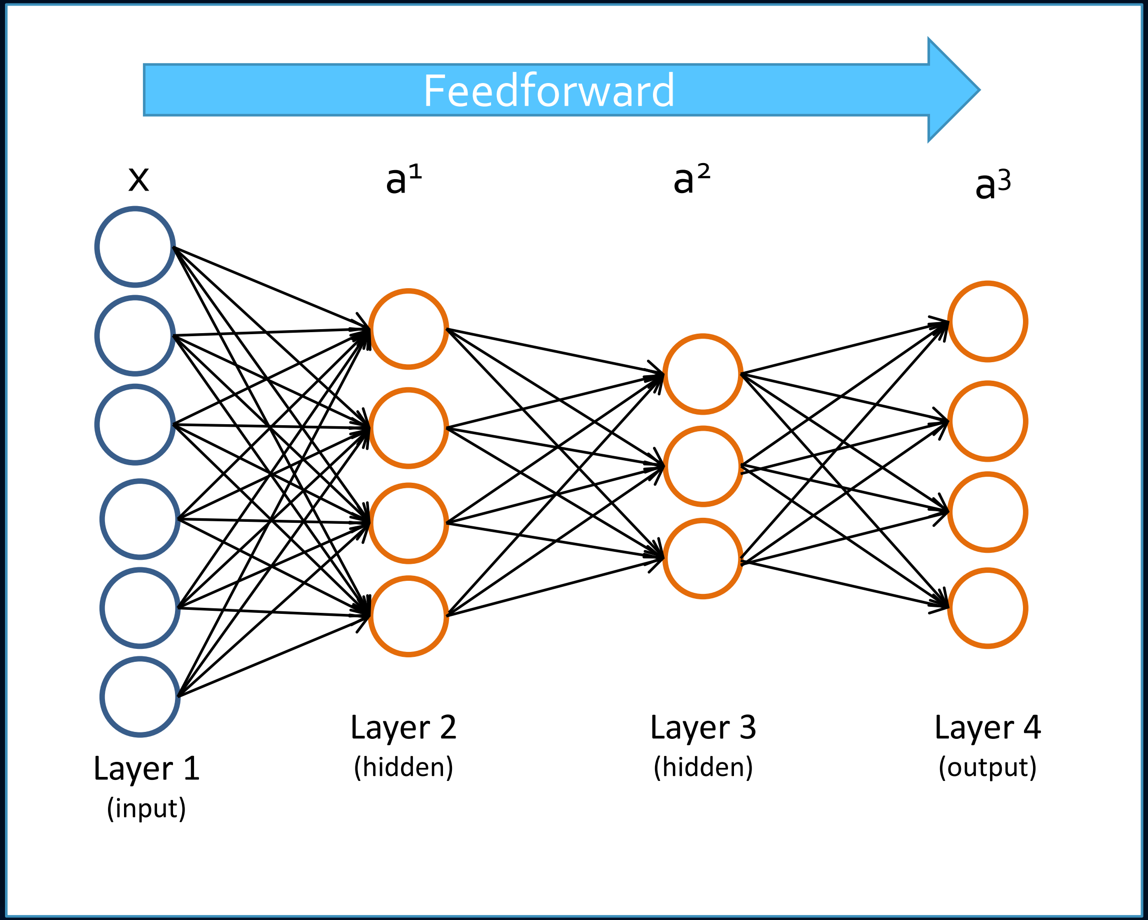
Neural Network: propagations

A FIRST STEP TO DEEP LEARNING

Feedforward

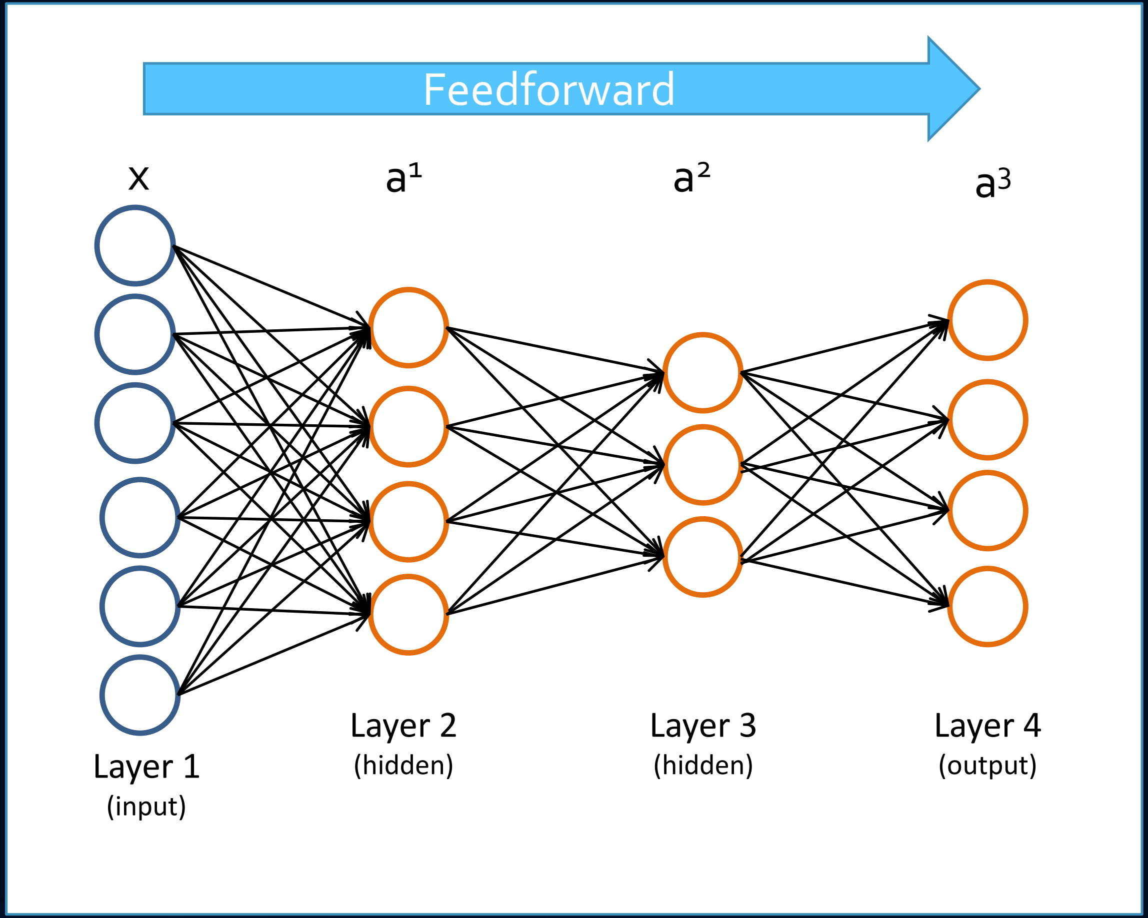


Feedforward



Feedforward

Propagate the information from the input to the output:



Feedforward

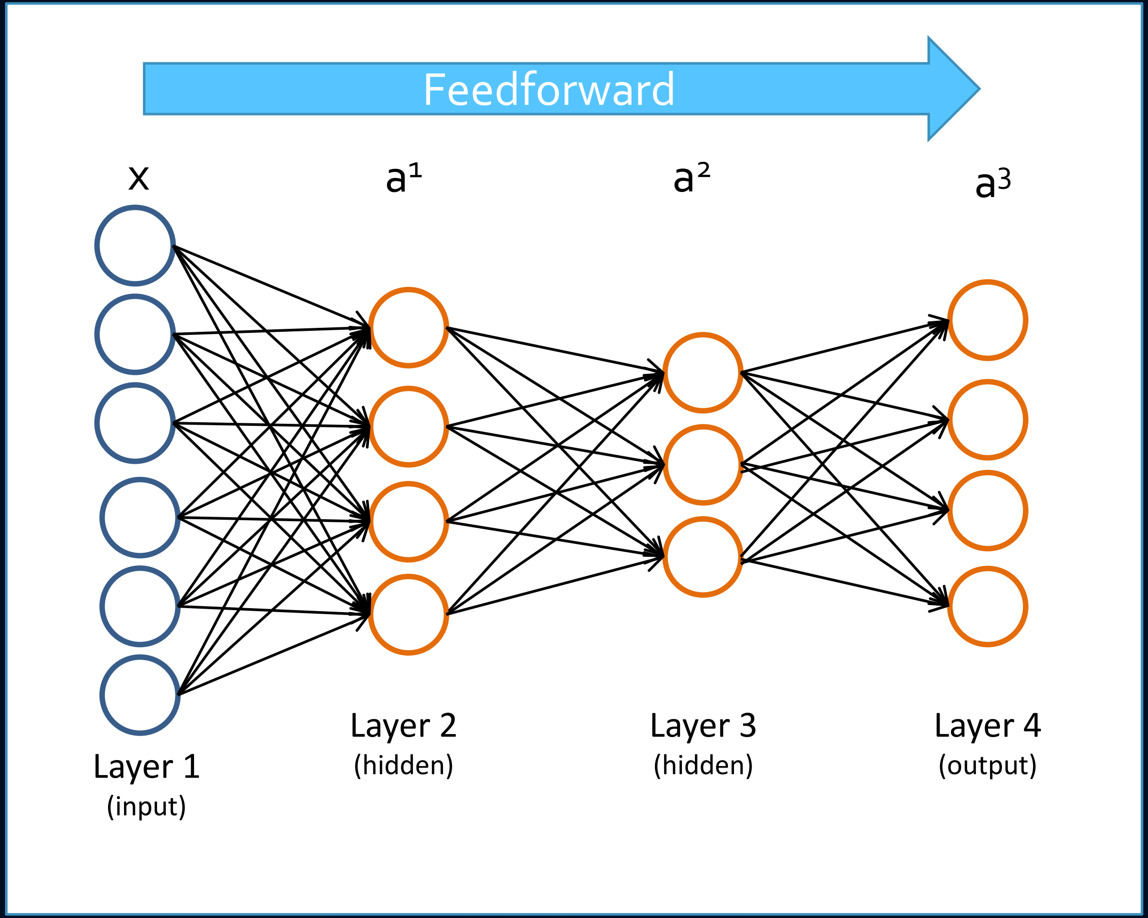
$a\ 2 = gg(z\ 2zzz\ 2\ z\ 22z\ 2);\ z\ 3\ zzz\ 3\ z\ 33z\ 3 = \theta\ 2\theta\theta\theta\ 2\ \theta\ 22\ \theta\ 2\ a\ 2aaa\ 2\ a\ 22a\ 2$; …

$a\ 2 = gg(z\ 2zzz\ 2\ z\ 22z\ 2);\ z\ 3\ zzz\ 3\ z\ 33z\ 3 = \theta\ 2\theta\theta\theta\ 2\ \theta\ 22\ \theta\ 2\ a\ 2aaa\ 2\ a\ 22a\ 2$; …

$a\ 1 = xx;\ z\ 2zzz\ 2\ z\ 22z\ 2 = \theta\ 1\ \theta\theta\theta\ 1\ \theta\ 11\theta\ 1\ a\ 1aaa\ 1\ a\ 11a\ 1$;

$a\ 2\ 2\ \ 2\ 2 = g(z^2); z^3 = \theta^2 a^2$; …

$a\ 2\ 2\ \ 2\ 2 = g(z^2); z^3 = \theta^2 a^2$; …

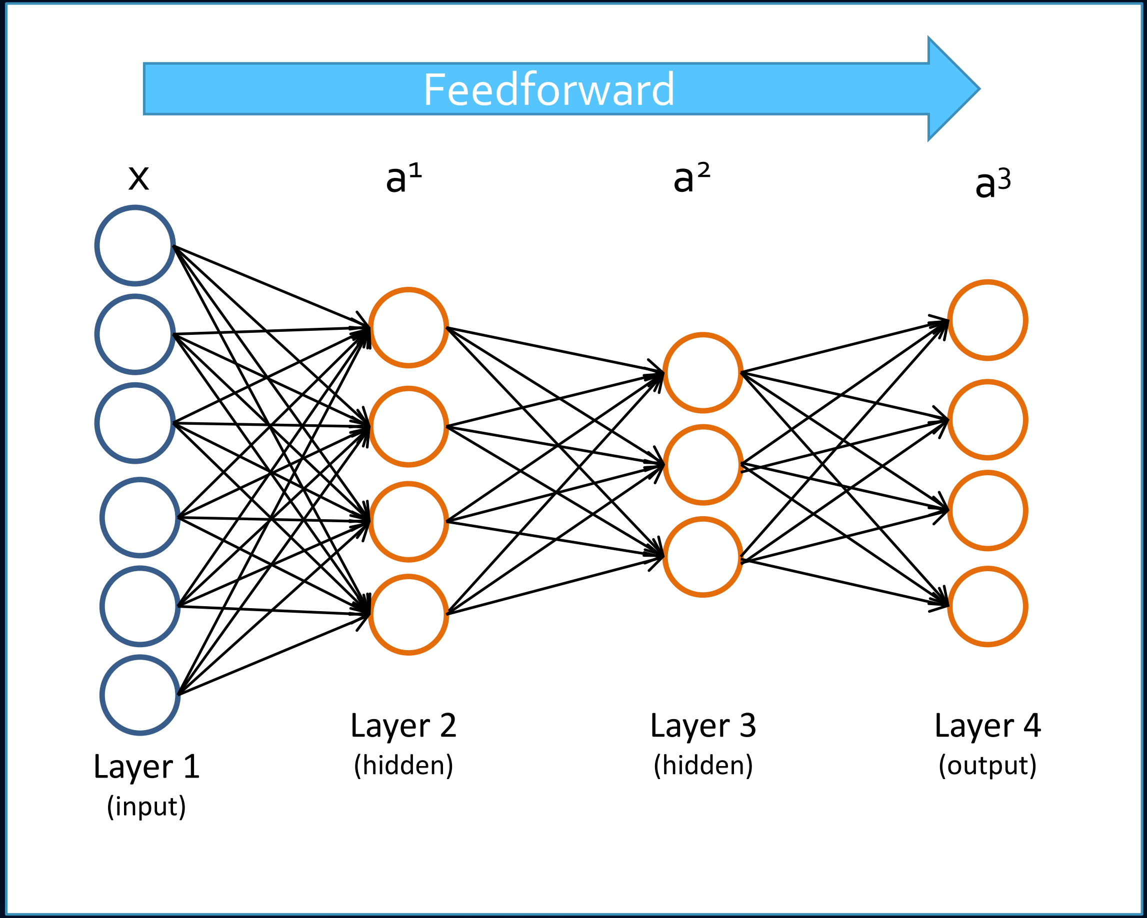


Backpropagation

$= a^3 \ - y\,;$

$= (\theta^2)^T .* \ g'(\theta^2) \ \ldots$

$=$ "error" of cost for $a_j^{(l)}$
(unit j in layer l)

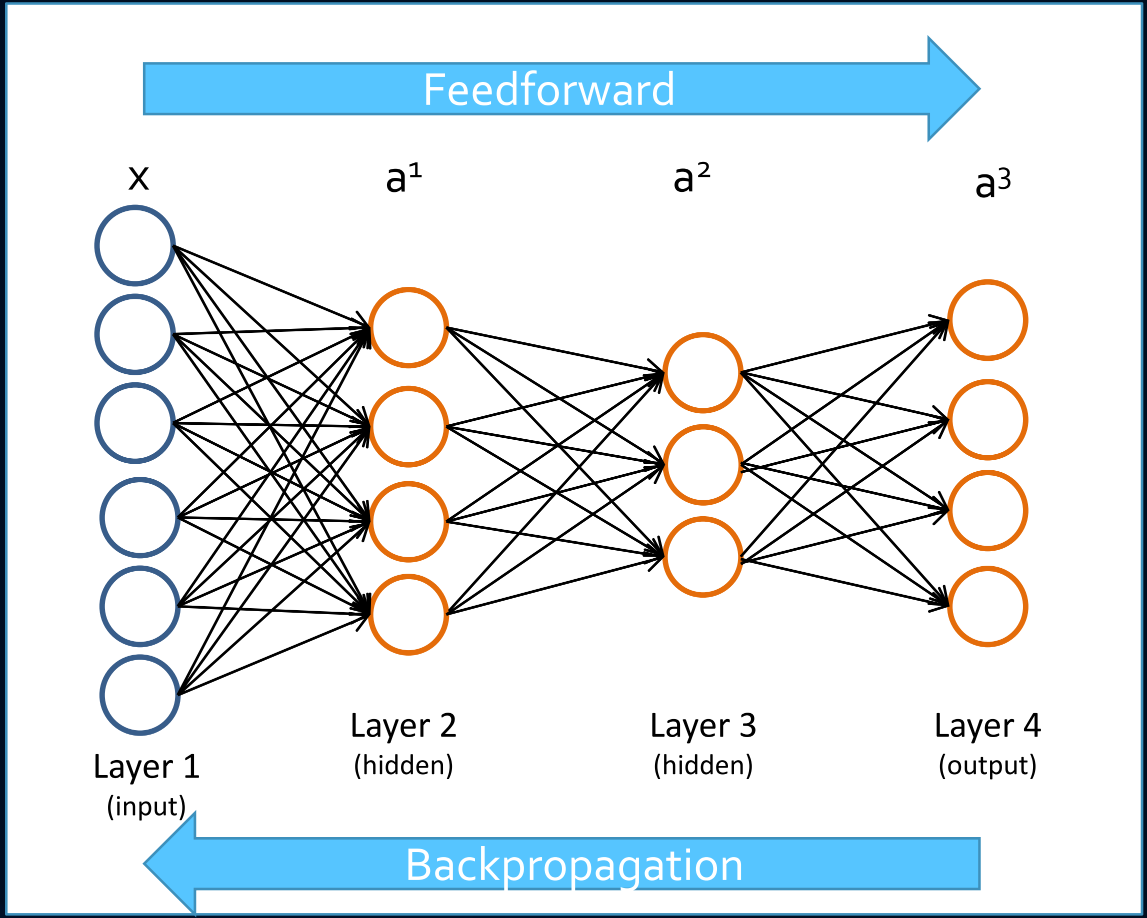


Backpropagation

$= a^3 \; - y\,;$

$= (\theta^2)^T .* \; g'(\theta^2) \;\; \ldots$

$=$ "error" of cost for $a_j^{(l)}$
(unit j in layer l)



Backpropagation

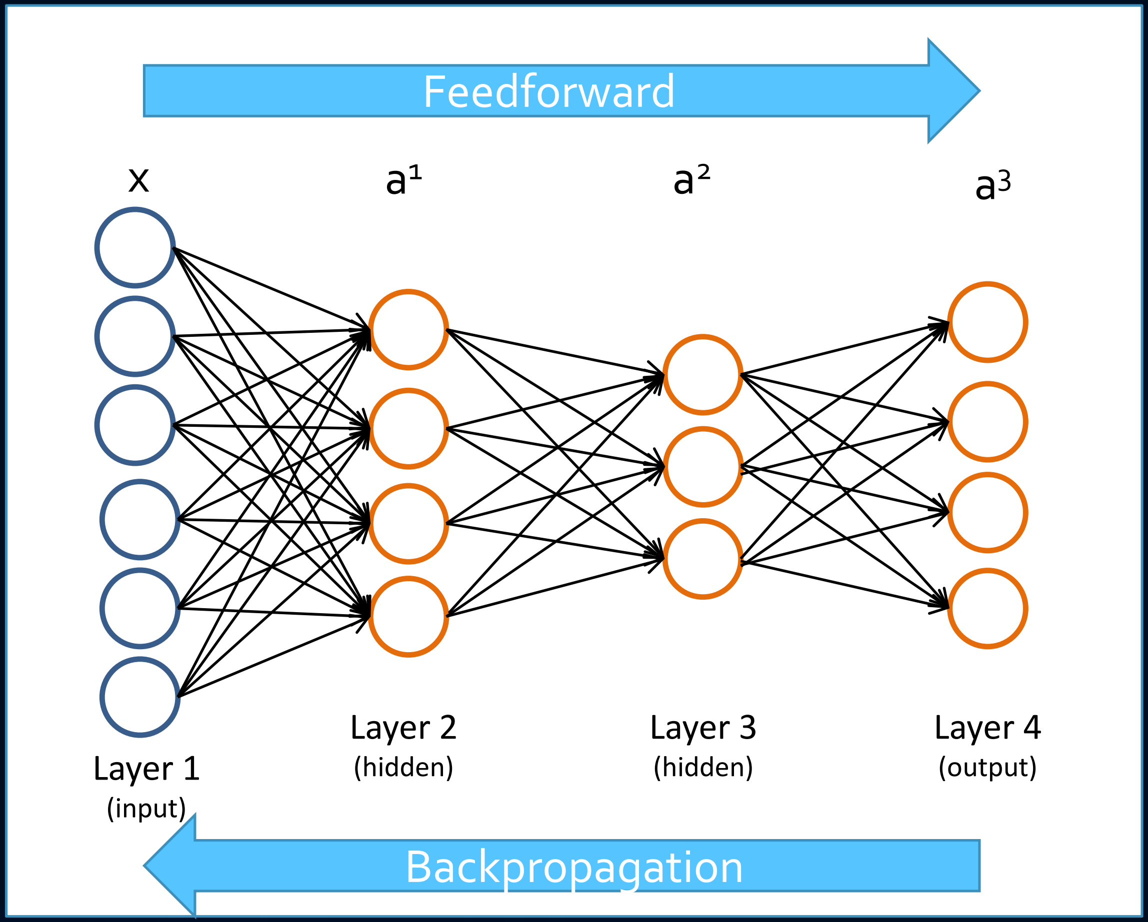
Propagate the error from the output to all the hidden layers:

$= a^3 \ - y \ ;$

$= (\theta^2)^T .* \ g'(\theta^2) \ \ldots$

= “error” of cost for $a_j^{(l)}$

(unit j in layer l)



Backpropagation

$\delta\ 3 = a\ 3\ aa\ a\ 3\ a\ 3\ 3\ a\ 3 - yy\ ;$

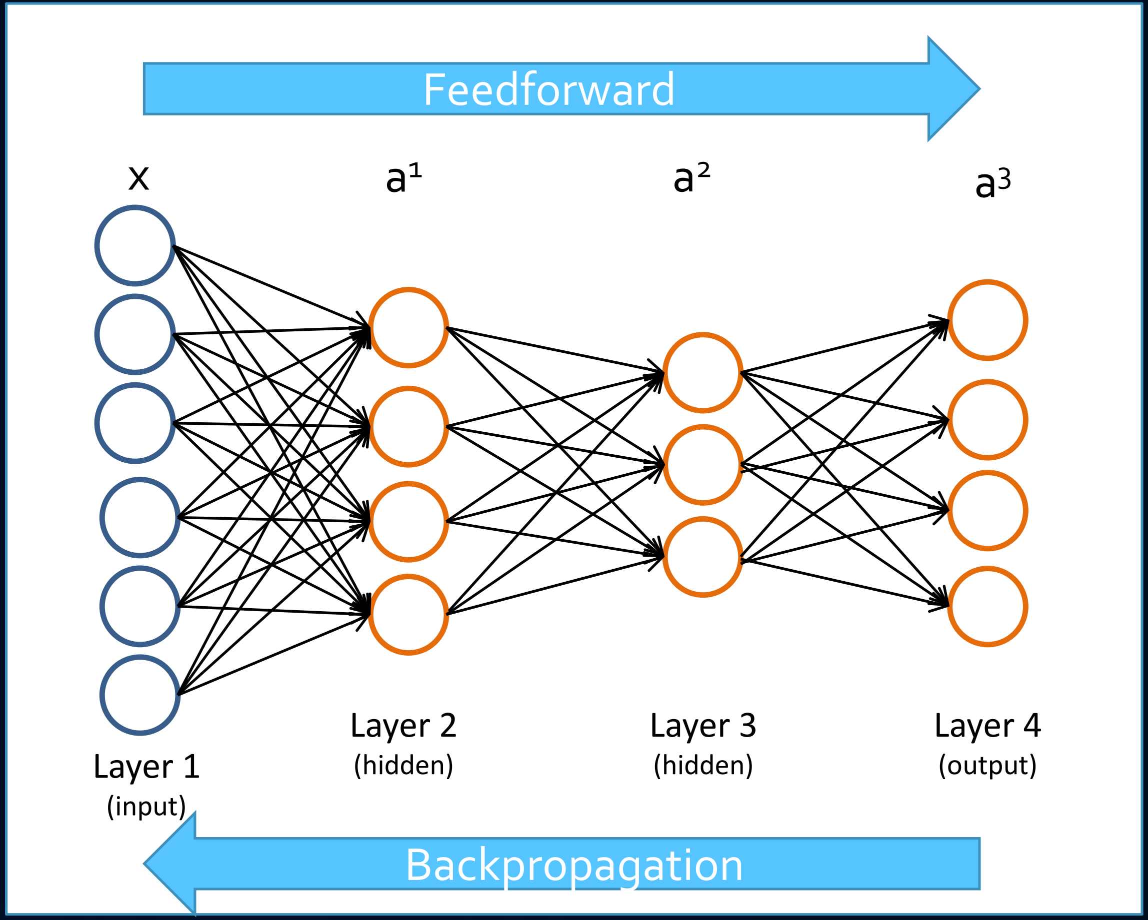
Propagate the error from the output to all the hidden layers:

$\delta\ 3\ 3\ \ 3\ 3\ = a^3 - y\ ;$

$= (\theta^2)^T .* \ g'(\theta^2)\ \ldots$

= “error” of cost for $a_j^{(l)}$

(unit *j* in layer *l*)



Backpropagation

$\delta\ 2 = (\theta\ 2\ )\ T\ (\ \theta\ 2\ \theta\theta\ \theta\ 2\ 2\ \theta\ 2\ )\ (\ \theta\ 2$
$)\ T\ TT\ (\ \theta\ 2\ )\ T\ .\ *\ g\ '\ gg\ g\ ''\ g\ '\ \ \theta\ 2$
$\theta\ 2\ \theta\theta\ \theta\ 2\ 2\ \theta\ 2\ \ \ \theta\ 2\ \ \ \ldots$

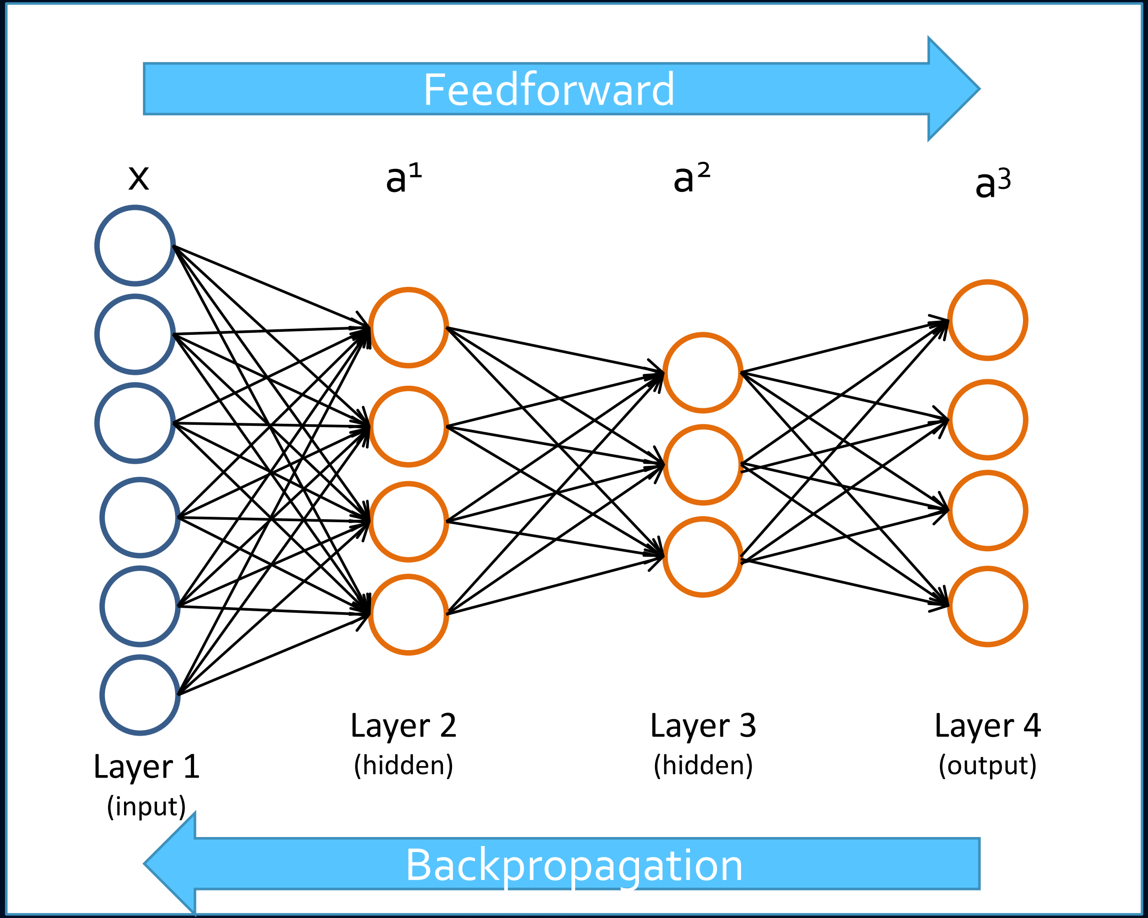
$\delta\ 3 = a\ \ 3\ aa\ a\ \ 3\ \ a\ \ 3\ 3\ a\ \ 3 - yy\ ;$

Propagate the error from the output to all the hidden layers:

$\delta\ \ 2\ \ 2\ \ \ 2\ \ 2\ \ = (\theta^2)^T .*\ g'(\theta^2)\ \ldots$

$= (\theta^2)^T .*\ g'(\theta^2)\ \ \ldots$

= "error" of cost for $a_j^{(l)}$
(unit j in layer l)



Backpropagation

$ll)\ \delta\ j\ (l)$ = "error" of cost for $a\ j\ (l)\ aa\ a\ j\ (l)\ jj\ a\ j\ (l)\ (ll)\ a\ j\ (l)$

(unit j in layer l)

$\delta\ 2 = (\theta\ 2)\ T\ (\theta\ 2\ \theta\theta\ \theta\ 2\ 2\ \theta\ 2)\ (\theta\ 2)\ T\ TT\ (\theta\ 2)\ T\ .\ *\ g\ '\ gg\ g\ '\ '\ g\ '\ \theta\ 2\ \ \theta\ 2\ \theta\theta\ \theta\ 2\ 2\ \theta\ 2\ \ \theta\ 2$
…

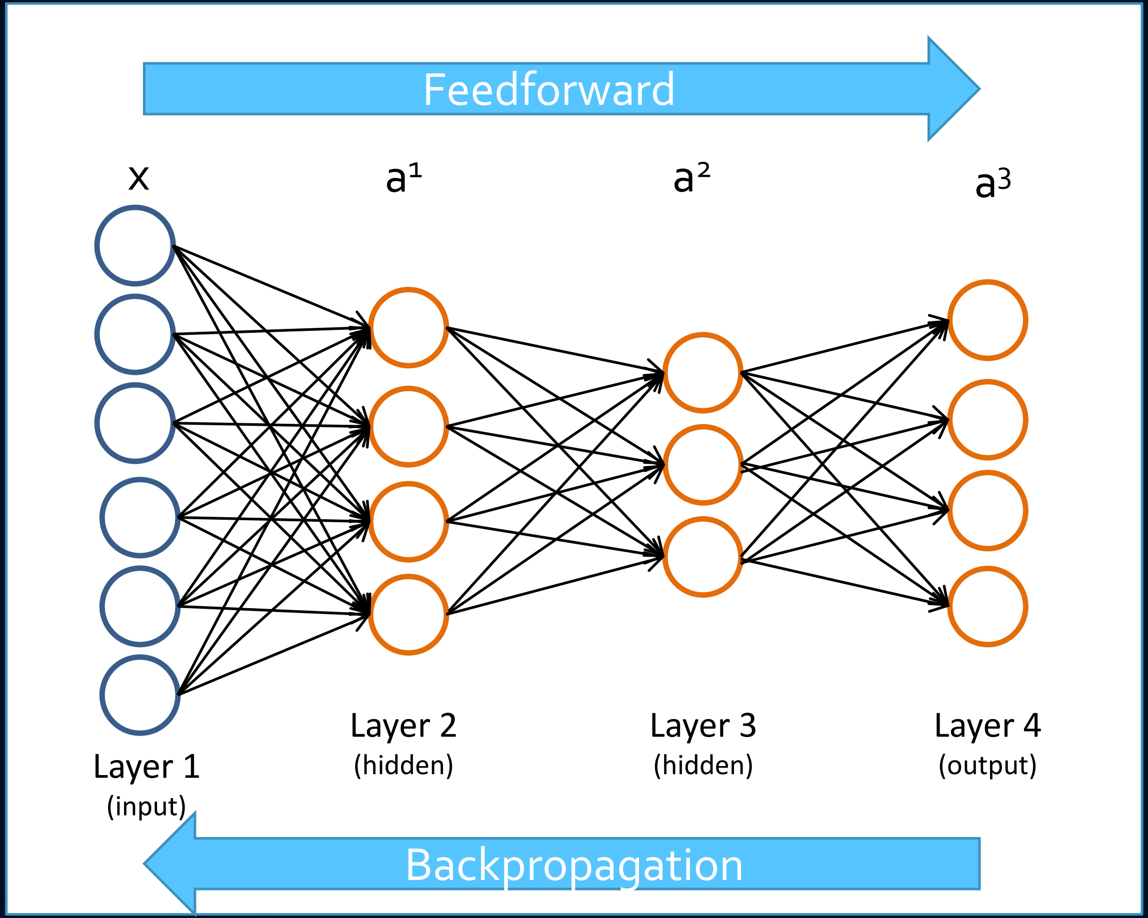
$\delta\ 3 = a\ \ 3\ aa\ a\ \ 3\ \ a\ \ 3\ 3\ a\ \ 3 - yy\ ;$

Propagate the error from the output to all the hidden layers:

$\delta\ \ 2\ \ 2\ \ \ 2\ \ 2\ = (\theta^2)^T .* \ g'(\theta^2)$ …

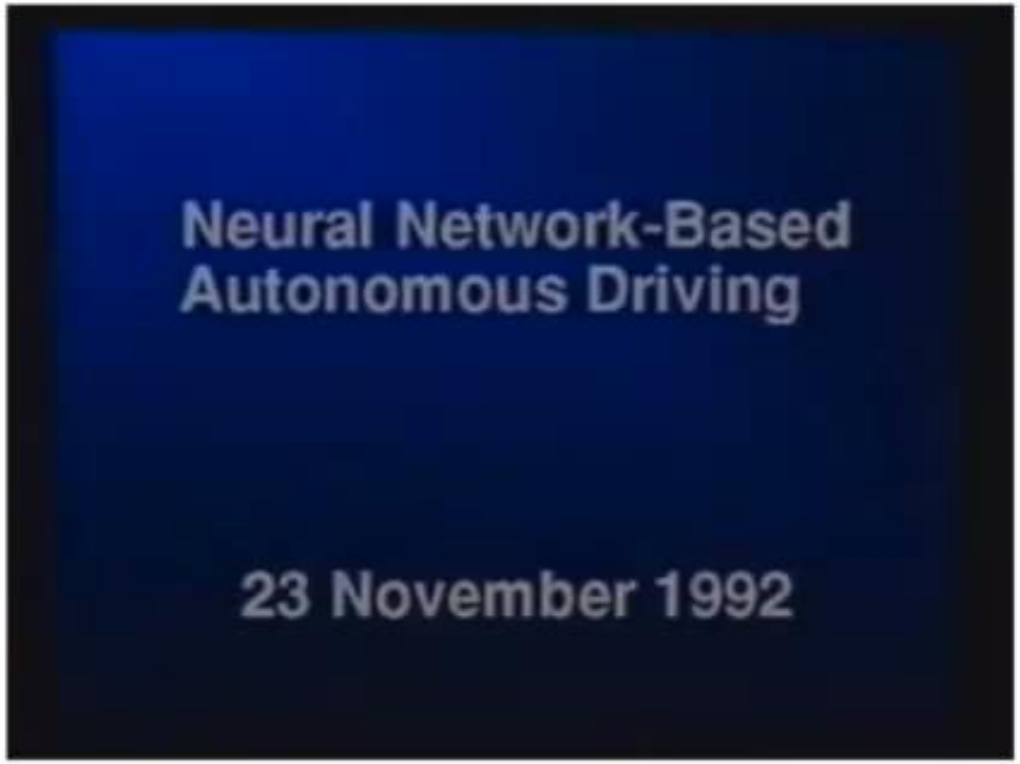
$\delta\ j\ (\ j\ (l)\ j\ (l)\ j\ (l)$ = "error" of cost for $a_j^{(l)}$
(unit *j* in layer *l*)

= "error" of cost for $a_j^{(l)}$
(unit *j* in layer *l*)



Example of backpropagation: autonomous driving (1992!)

Example of backpropagation: autonomous driving (1992!)



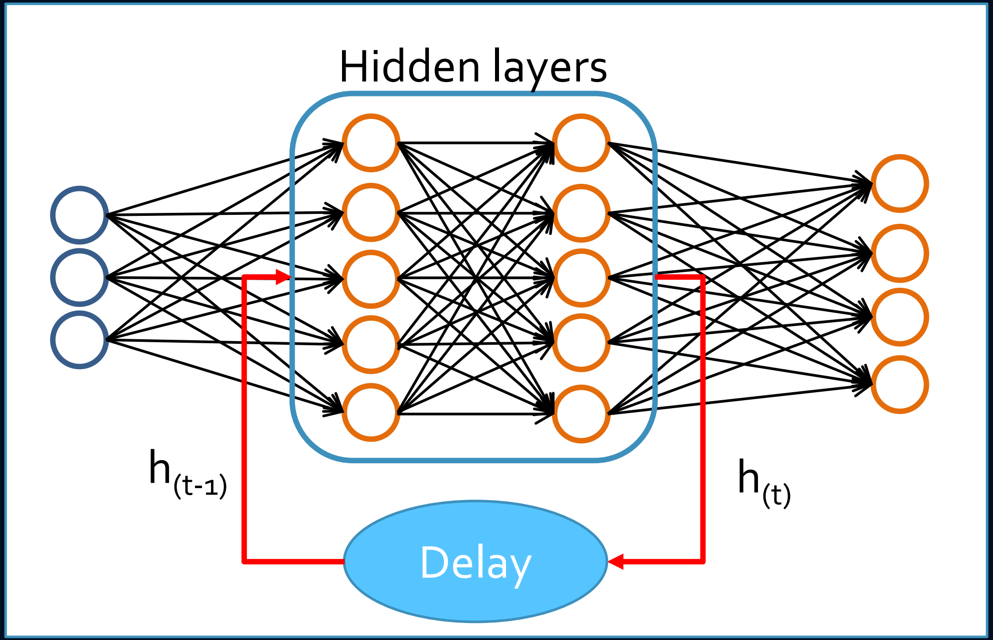
(Courtesy of Dean Pomerleau)

Neural Network: Recurrent Model

RNN MODEL

Intuition of recurrence in NN

- Take the output at *t-1* and feedback the hidden layers of the NN
- *t* is discretized with the activation update at each time step
- The output values summarize all the information previously given (*i.e.:* it keeps all the history)



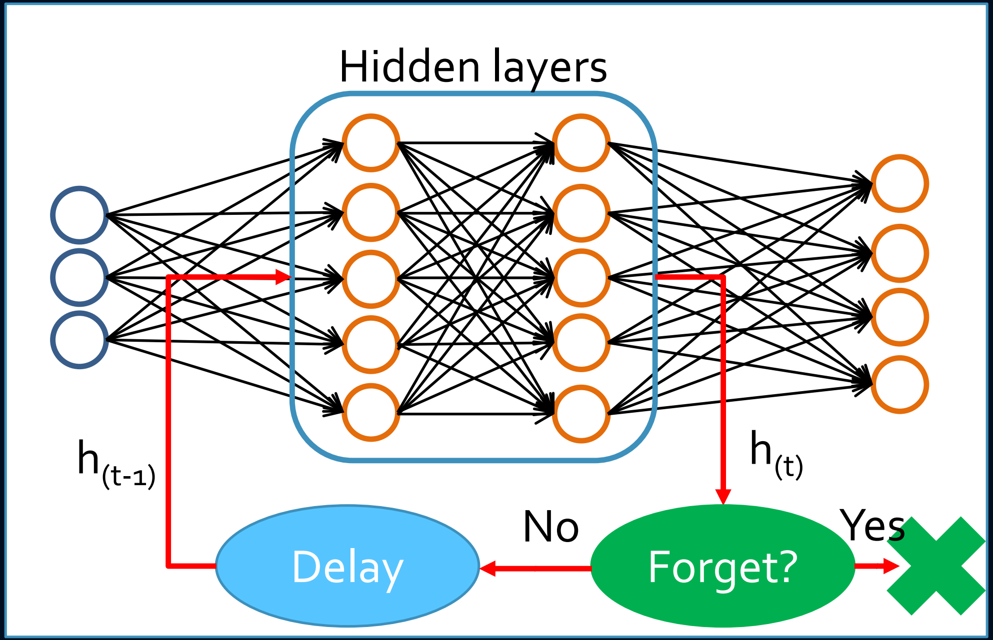
Neural Network: Long-Short Term Memory

EXTENSION OF RNN MODEL: THE LSTM!

Intuition of LSTM

- RNN keeps all the history unlike the human memory
- Based on human brain memory process: we forgot souvenirs
- Forgot old-dated things to keep the highlight on recent memories
- Revised things are more important

⇒This is the idea of LSTM



Application of Deep Learning to NLP problems

ARE NLP PROBLEMS SOLVED?

Application of Deep Learning to NLP problems

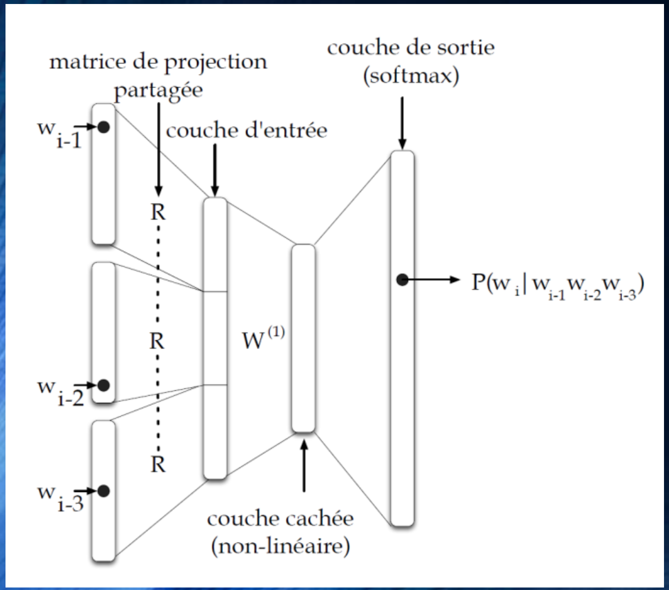
- Language Models
- Statistical Machine Translation
- Parsing
- Spoken and Natural Language Understanding
- Word Embeddings

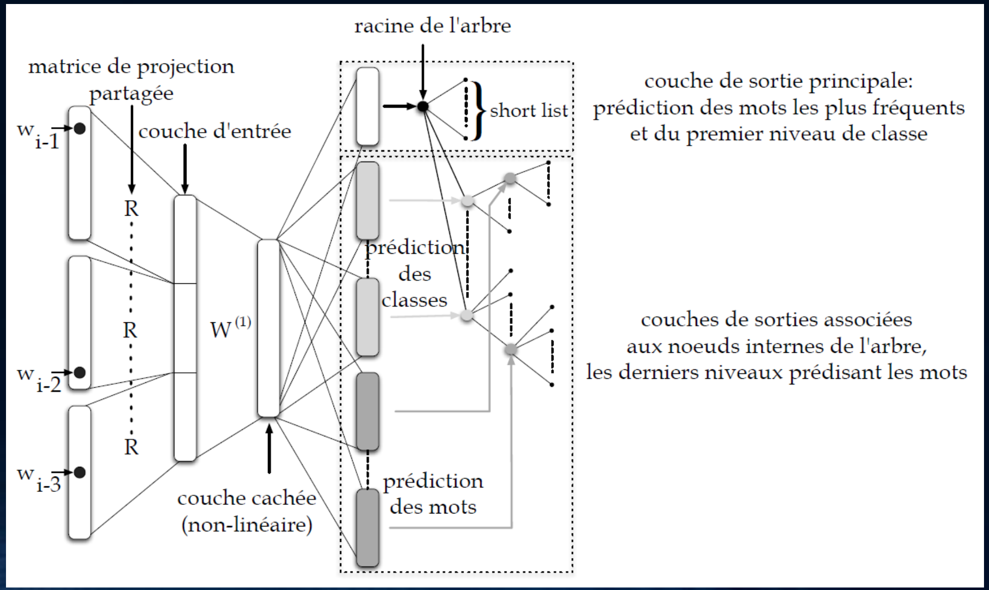
Application: Language Model

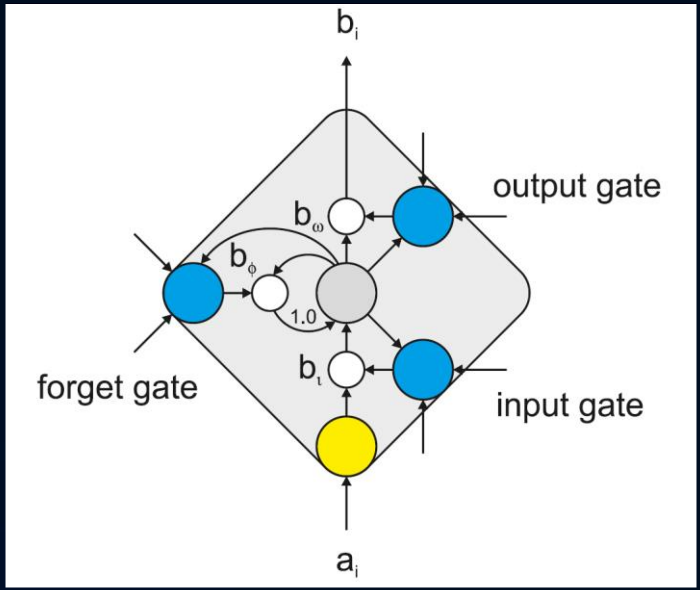
LANGUAGE MODELLED WITH DNN

Example of use for Language Models

- Language models
  - RNN:
    - Continuous Space Language Model (CSLM) [Schwenk, 2007]
    - Structured OUt- put Layer Neural Network language models (SOUL NNLM) [Le and al.,2011]
  - LSTM:
    - LSTM LM [Sundermeyer and al, 2010]







Application: SMT

TRANSLATE WITH DL APPROACHES

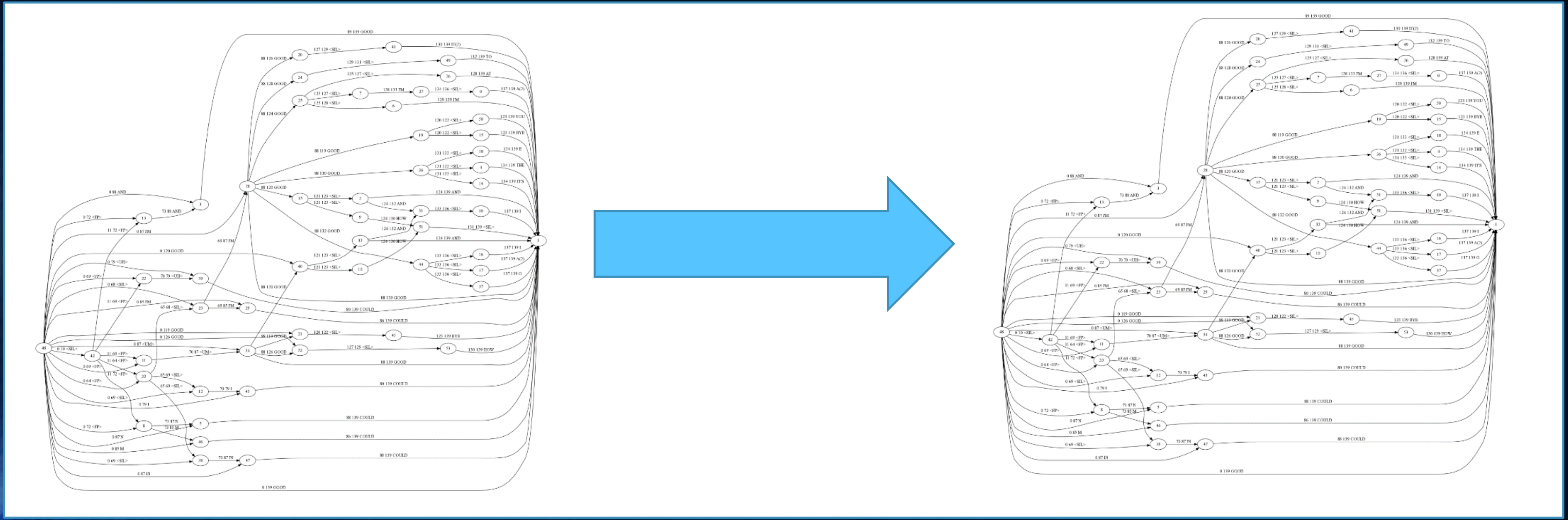
Deep Learning in SMT

- Mainly the use for rescoring (LM and Translation Model (TM))
- N-best rescoring [Schwenk, 2010]
  - SMT outputs N-best, then the DNN LM is used to re-rank the N-best

Hypothesis 1 Hypothesis 2 Hypothesis 3 Hypothesis 4 .... Hypothesis 99 Hypothesis 100	→	Hypothesis 5 Hypothesis 6 Hypothesis 1 Hypothesis 3 .... Hypothesis 87 Hypothesis 70

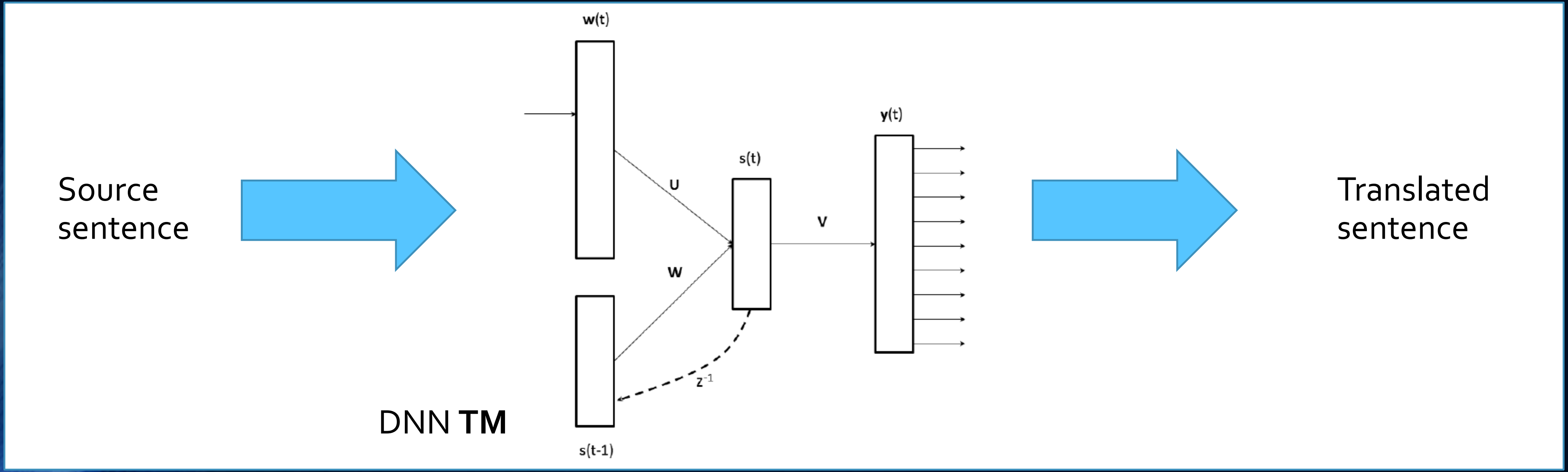
Deep Learning in SMT

- Rescoring Translation Model lattice using DNN in the SMT process [Schwenk and al.,2012]
  - Decoder outputs lattice, then the DNN TM is used to rescore the N-best



Deep Learning in SMT

- Replace the TM and LM in the translation process [Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015; Jean et al., 2015]



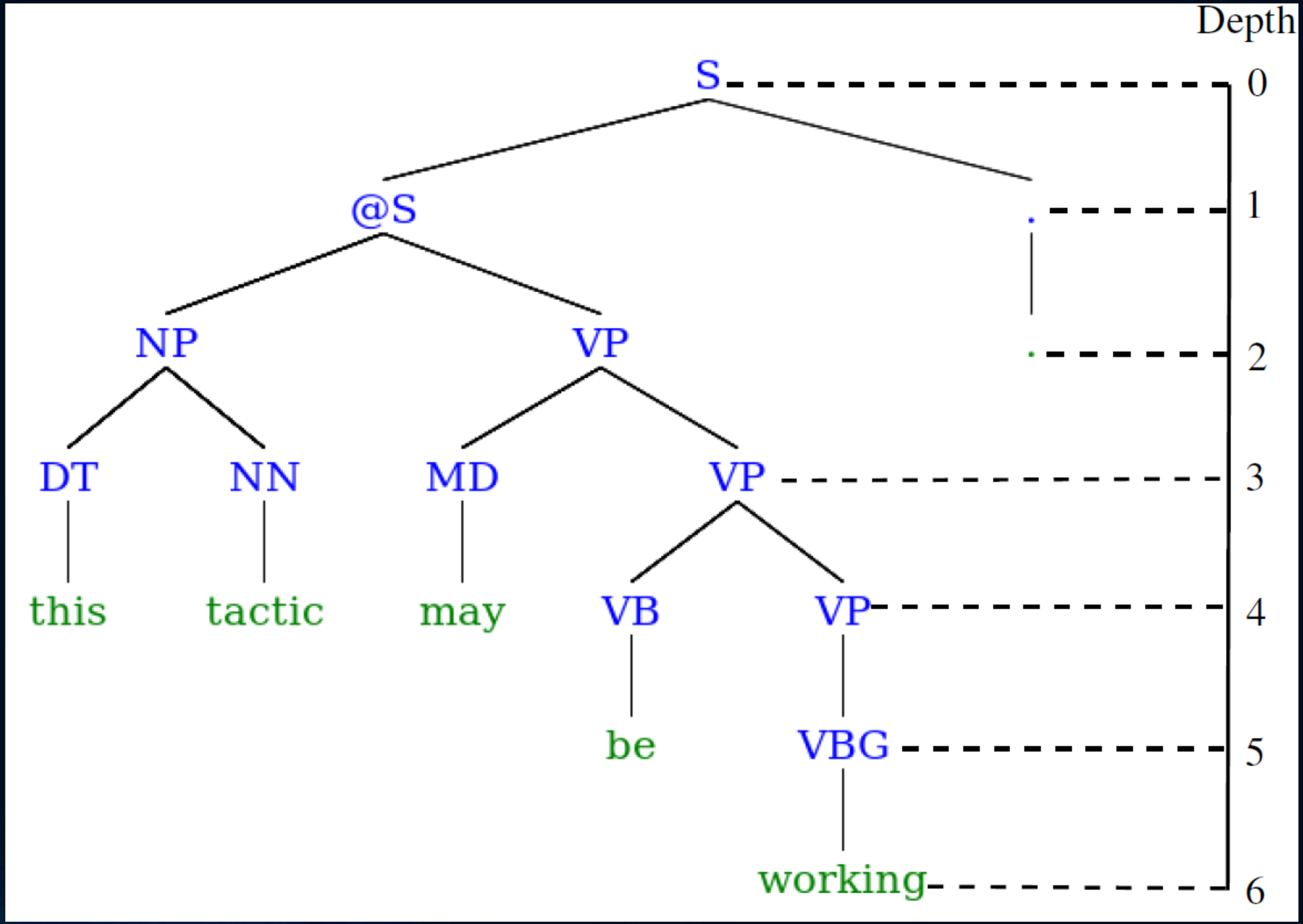
Application: Parsing

DL AND THE ANALYSIS OF A SENTENCE

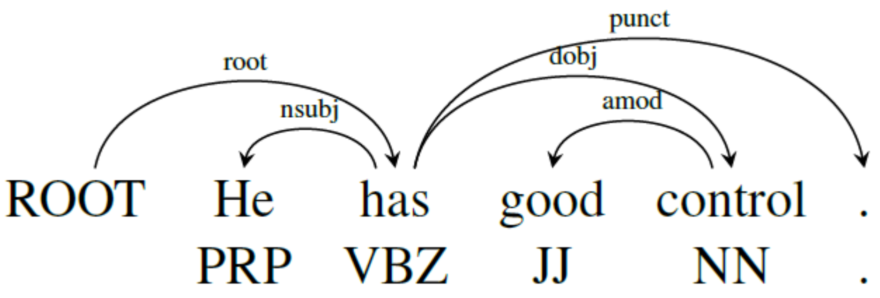
Parsing with Deep Learning

- Parsing:
  - Syntactic analysis of a sentence
- Example:

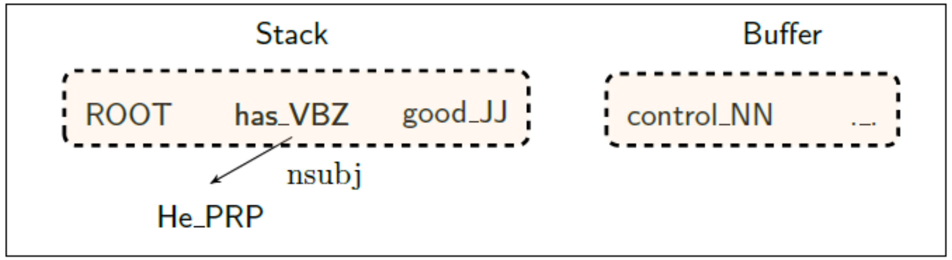
  "This tactic may be working"



Parsing with Deep Learning [Chen and Manning, 2014]



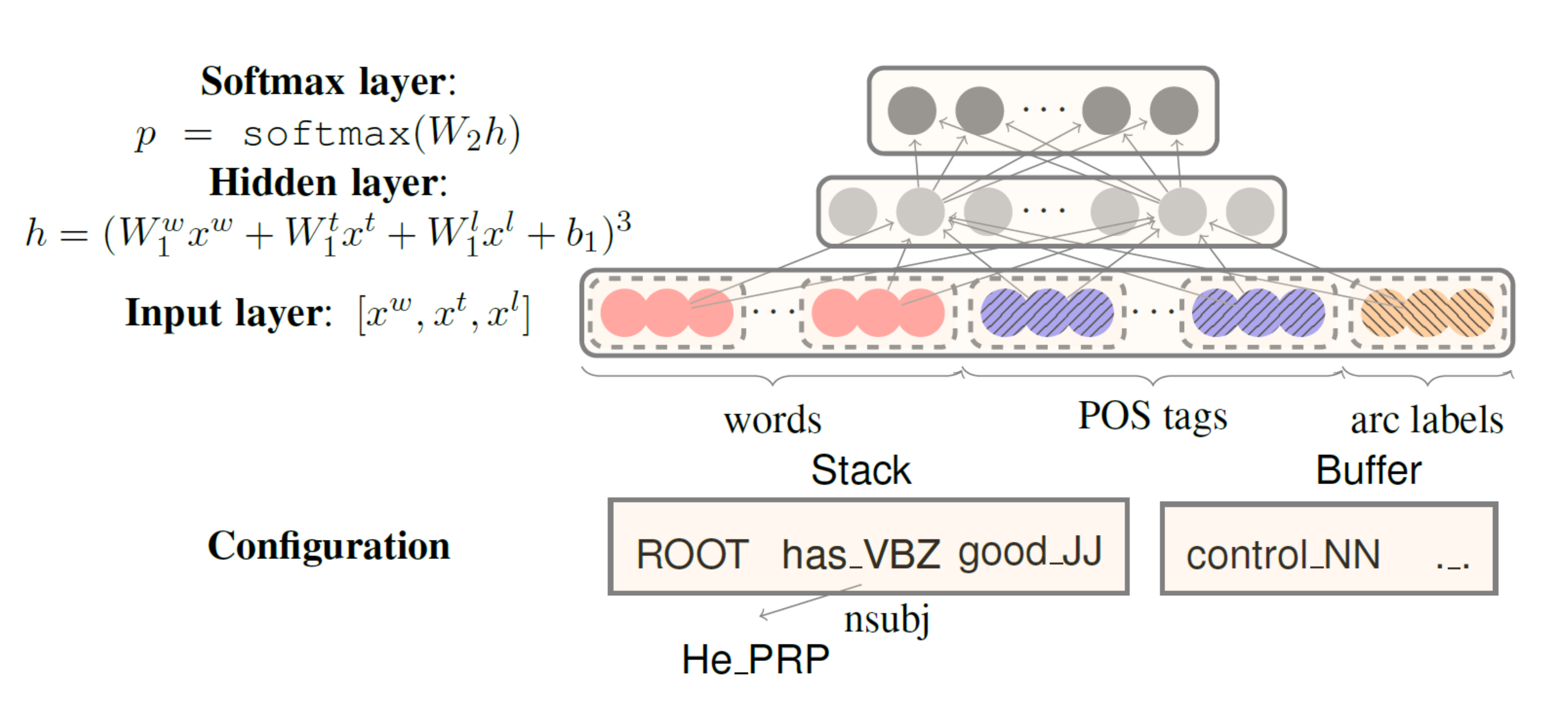
Correct transition: SHIFT



Transition	Stack	Buffer	A
	[ROOT]	[He has good control .]	$\emptyset$
SHIFT	[ROOT He]	[has good control .]	
SHIFT	[ROOT He has]	[good control .]	
LEFT-ARC(nsubj)	[ROOT has]	[good control .]	$A\cup$ nsubj(has,He)
SHIFT	[ROOT has good]	[control .]	
SHIFT	[ROOT has good control]	[.]	
LEFT-ARC(amod)	[ROOT has control]	[.]	$A\cup$amod(control,good)
RIGHT-ARC(dobj)	[ROOT has]	[.]	$A\cup$ dobj(has,control)
...	...	...	...
RIGHT-ARC(root)	[ROOT]	[]	$A\cup$ root(ROOT,has)

Parsing with Deep Learning [Chen and Manning, 2014]

⇒ Greedy decoding (POS + labels) then evaluation using a DNN



Application: Spoken and Natural Language Understanding

CAN WE CAPTURE THE MEANING OF A SENTENCE?

Spoken Language Understanding

- Mainly a POS tagging task
- Example: “yes the hotel which price is below fifty five euros”

n	Wc	c	value
1	yes	answer	yes
2	the	RefLink	singular
3	hotel	BDObject	hotel
4	which	null	
5	price	object	payment-amount
6	is below	comparative-payment	below
7	fifty five	payment-amount-int	55
8	euros	payment-currency	euro

Spoken Language Understanding

- Comparison of several approaches [Deng and al., 2012 ; Vukotic and al., 2015]

Algorithm	Parameter	Representation	Precision	Recall	F-measure	Training Time
Task: ATIS						
Bonzaiboost	100 iter	numeric (word2vec)	93.50%	94.54%	94.02%	˜20m
Bonzaiboost	100 iter	symbolic	93.12%	92.82%	92.97%	˜3m
CRF		symbolic	95.53%	94.92%	95.23%	˜6m
Elman RNN	**100 hdn**	**numeric (joint)**	**96.20%**	**96.12%**	**96.16%**	**˜1.5h**
Task: MEDIA						
Bonzaiboost	500 iter.	numeric (word2vec)	73.61%	78.85%	76.14%	˜2.5h
Bonzaiboost	500 iter.	symbolic	71.09%	75.48%	73.22%	˜34m
CRF		**symbolic**	**87.70%**	**84.35%**	**86.00%**	**˜15m**
Elman RNN	500 hdn	numeric (joint)	83.36%	80.22%	81.76%	˜31h
Elman RNN	500 hdn	numeric (word2vec)	80.48%	83.46%	81.94%	˜22h
Jordan RNN	500 hdn	numeric (joint)	82.76%	83.75%	83.25%	˜3.5h
Jordan RNN	500 hdn	numeric (word2vec)	83.40%	82.90%	83.15%	˜3h

Application: Word Embeddings

CAN WE MODEL THE WORDS IN SOME WAY?

Declination of NN: word embeddings

- Word representation in a continuous space as a vector
- Highly studied these past years [Bengio et al., 2003 ; Turian et al., 2010 ; Collobert et al.,2011 ; Huang et al., 2012]
- It follows the idea that the meaning of a word can be determined by 'the company it keeps' [Baroni and Zamparelli, 2010].
- Most famous and used is word2vec [Mikolov and al.,2013]

⇒ **Based on auto-encoders**

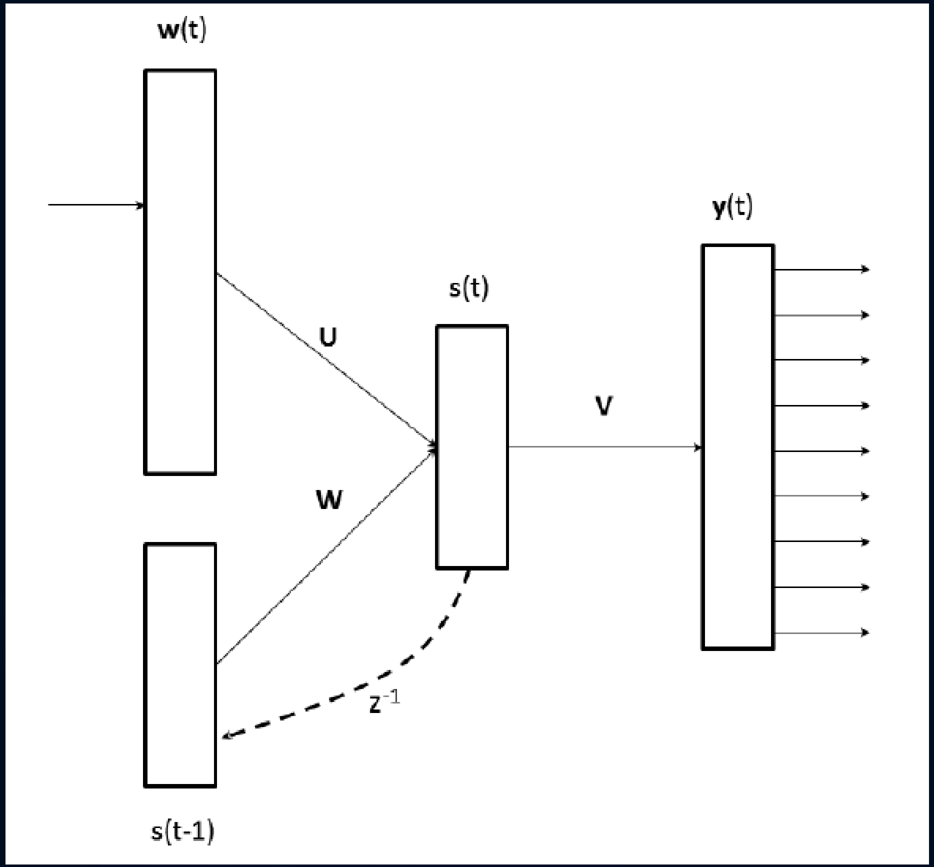
Word embedding

- Words as continuous vectors
  - $\Rightarrow$ High level of representation
  - $\Rightarrow$ Words as input
  - $\Rightarrow$ Distribution of probabilities over words

s(t) = f (U.w(t) + W.s(t−1))

y(t) = g (V.s(t))

$$f(z) = \frac{1}{1+e^{-z}}, g(z_m) = \frac{e^{z_m}}{\sum_k e^{z_k}}$$



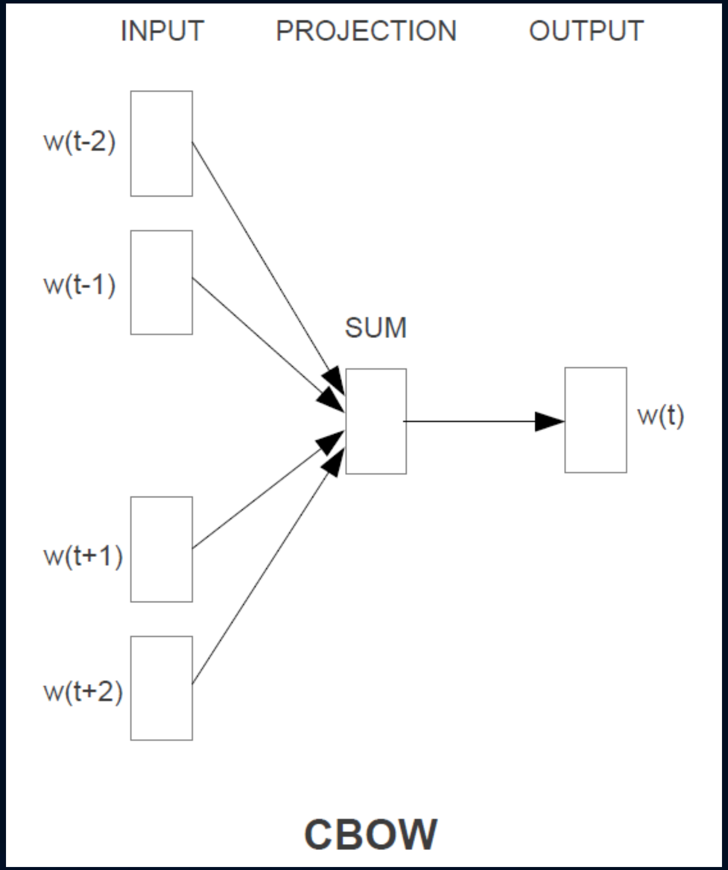
Continuous Bag of Words model

- Remove the hidden layer

  ⇒Projection layer shared by all words

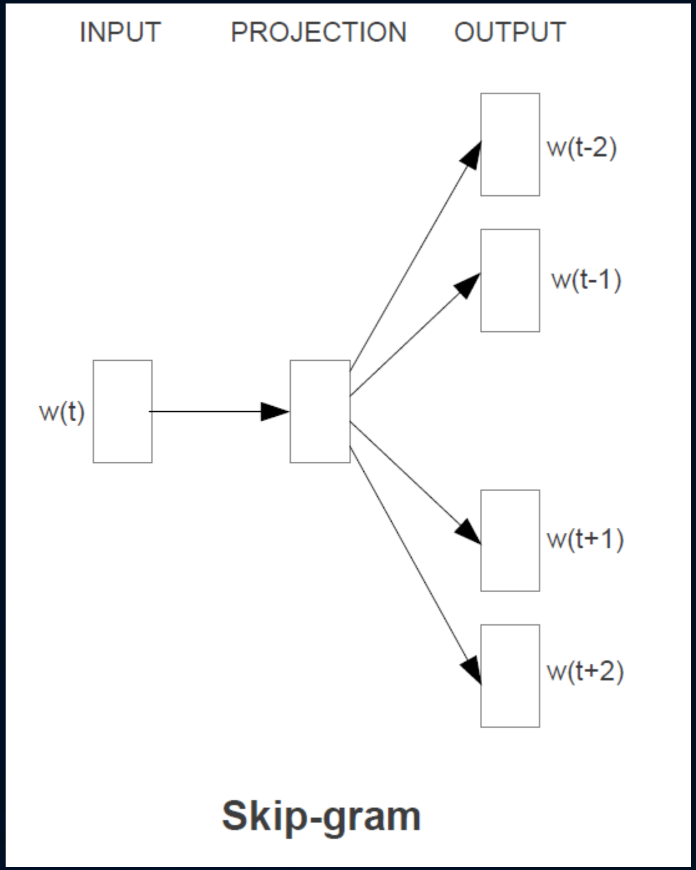
- No history / order information

  ⇒Predict current word regarding the context (other words)



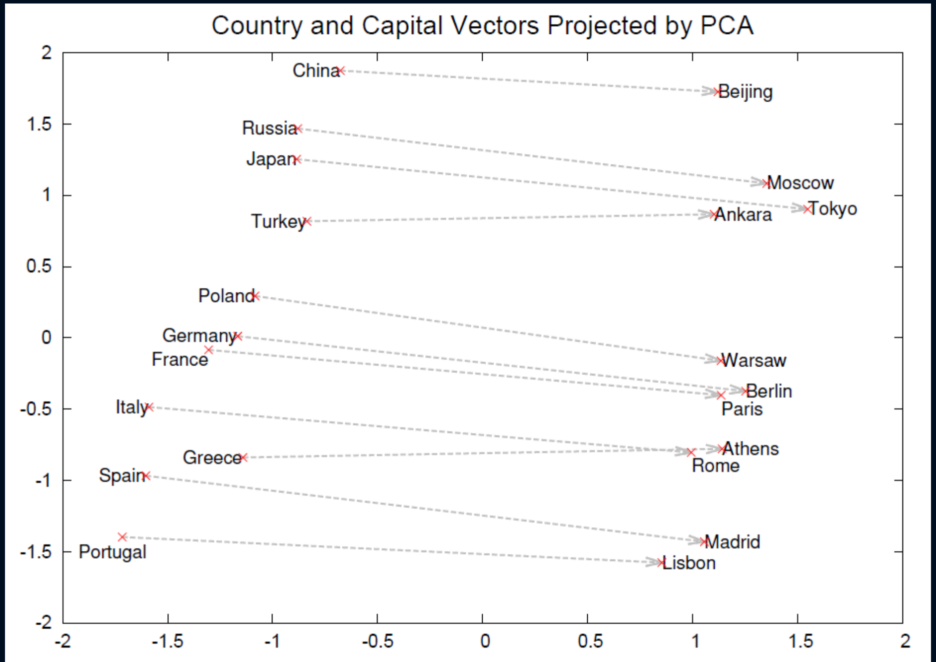
Skip-gram model

- Similar to CBOW
  - ⇒ Predict other words regarding the current word
- Negative Sampling
  - ⇒ Propose a noise distribution for counter-examples
- Subsampling of Frequent Words
  - ⇒ Remove randomly samples of frequent words
- Hierarchical Softmax
  - ⇒ Evaluate $\log_2(W)$
  - ⇒ Binary tree representation



Analogical reasoning task

- Germany, Berlin -> France, x
  - vec(x) ≈ vec("Berlin") - vec("Germany") + vec("France") => "Paris"
- quick, quickly -> slow, x
  - vec(x) ≈ vec("quickly") - vec("quick") + vec("slow") => "Slowly"
- Newspaper names:
  - New York => New York Times
  - San Jose => San Jose Mercury News
  - …
- Accuracy measured



Method	Dim	No Subsampling	subsampling
NEG-5	300	24	27
NEG-15	300	27	42
HS-Huffman	300	19	47
HS-Huffman*	1000	--	**72**

Evaluate the distance between words

- The closest words to “France”:

Words	Cosine distance
spain	0.678515
belgium	0.665923
netherlands	0.652428
italy	0.633130
switzerland	0.622323
luxembourg	0.610033
portugal	0.577154
russia	0.571507
germany	0.563291
catalonia	0.534176

- The closest words to “San Francisco”:

Words	Cosine distance
los angeles	0.666175
golden gate	0.571522
oakland	0.557521
california	0.554623
san diego	0.534939
pasadena	0.519115
seattle	0.512098
taiko	0.507570
houston	0.499762
chicago illinois	0.491598

NLP tasks that uses word embeddings

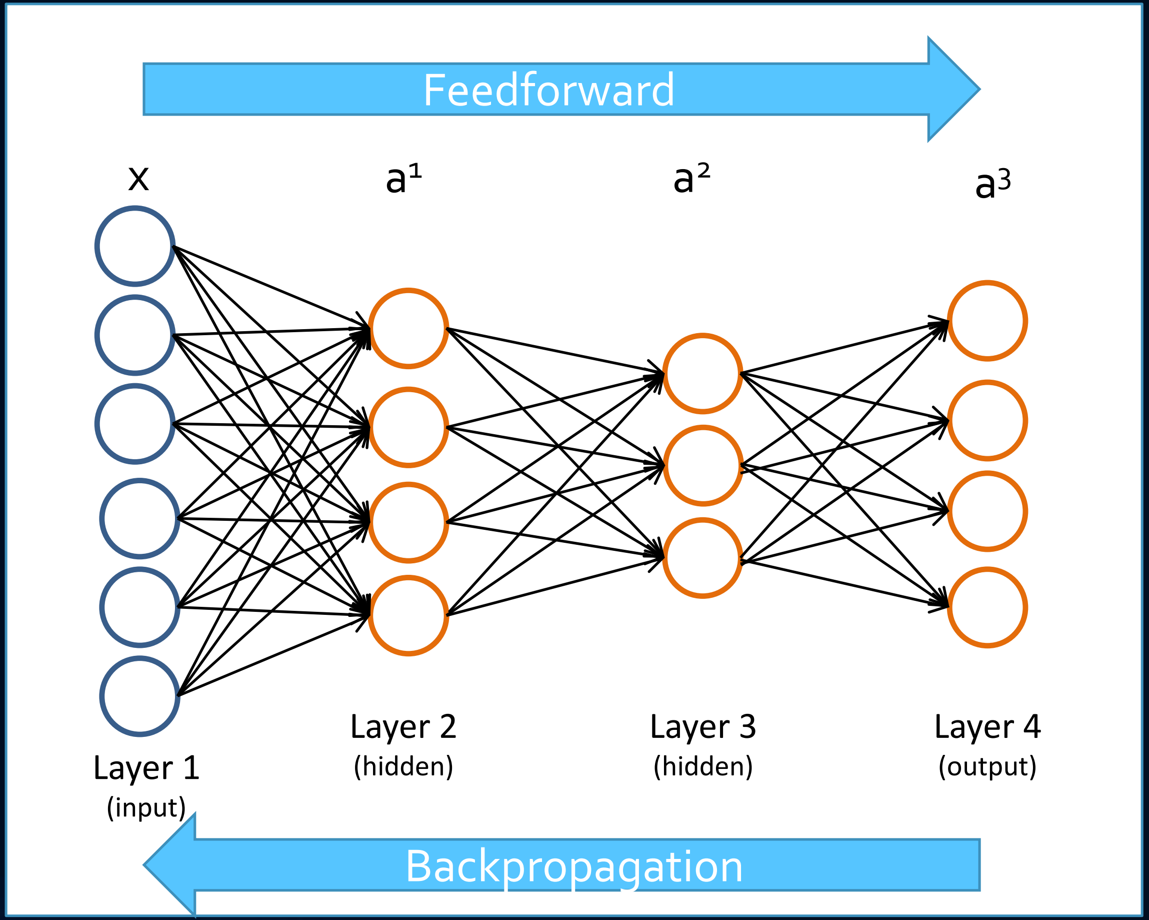
- Word Sens Disambiguation [Reisinger and Mooney, 2010 ; Huang and al, 2012 ; Neelakantan and al,2014]
- Named Entity Recognition [Neelakantan and Collins, 2014; Passos et al, 2014; Turian et al, 2010]
- Dependency Parsing [Bansal et al, 2014]
- Chunking [Turian et al, 2010; Dhillon and al., 2011]
- Sentiment Analysis [Maas et al, 2011]
- Paraphrase detection [Socher et al, 2011] and learning representations of paragraphs and documents [Le and Mikolov, 2014].
- Word clustering (from Brown corpus [Brown et al, 1992]) have similarly been successfully used as features in named entity recognition [Miller et al, 2004; Ratinov and Roth, 2009]

Conclusion

AND ADDITIONAL REMARKS

What is Deep Learning?

- A fancy expression to define Deep Neural Networks
- Can approximate non-linear functions
- « The learning becomes deep when it is composed of multiples non-linear transformations » Yann LeCun
- Propose high level of representation from raw data (speech, text, *etc.*)



Why it works?

- A lot of data (« Big Data »)
- Improvement in computational/memory power (GPU)
- Propose automatically a high level of representation
- Allow to do something without domain expertise knowledge

Applications of Deep Learning in NLP

- Language Models
- Statistical Machine Translation
- Parsing
- Spoken and Natural Language Understanding
- Word Embeddings
- ...and many others!

Who's who in Deep Learning

- Pr. Goeffrey Hinton, University of Toronto
- Pr. Yoshua Bengio, Université de Montréal
- Pr. Yann LeCun, University of New York & Director of Facebook AI
- Pr. Holger Schwenk, University of Le Mans & Facebook AI
- Pr. Christopher Manning, University of Standford
- Andrew Ng, University of Stanford & Chief Scientist of Baidu Research

The end!

Questions?

References

- [Schwenk, 2007]: Holger Schwenk, *Continuous Space Language Models* , in Computer Speech and Language, volume 21, pages 492-518, 2007.
- [Le and al., 2011]: Hai-Son Le, Ilya Oparin, Alexandre Allauzen, Jean-Luc Gauvain, and François Yvon. 2011a. Structured output layer neural network language model. In Proceedings of ICASSP'11, pages 5524–5527.
- [Sundermeyer and al., 2010]: M. Sundermeyer, R. Schluter, and H. Ney. *LSTM neural networks for language modeling*. In INTERSPEECH, 2010.
- [Schwenk 2010]: Holger Schwenk, *Continuous Space Language Models For Statistical Machine Translation*, The Prague Bulletin of Mathematical Linguistics, number 83, pages 137-146, 2010.
- [Schwenk 2012]: Holger Schwenk, *Continuous Space Translation Models for Phrase-Based Statistical Machine Translation*, in Coling, Dec 2012
- [Kalchbrenner and Blunsom, 2013]: Nal Kalchbrenner and Phil Blunsom. « Two recurrent Continuous Translation Models." *EMNLP*. 2013.
- [Sutskever and al., 2014]: Ilya Sutskever, Oriol Vinyals, and Quoc Le. *Sequence to sequence learning with neural networks.* In Advances in Neural Information Processing Systems (NIPS 2014), December, 2014
- [Bahdanau and al., 2015]: D. Bahdanau, K. Cho, and Y. Bengio. *Neural machine translation by jointly learning to align and translate*. In ICLR. 2015

References

- [Bahdanau and al., 2015]: D. Bahdanau, K. Cho, and Y. Bengio. *Neural machine translation by jointly learning to align and translate*. In ICLR. 2015
- [Jean and al.,2015]: Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. *On using very large target vocabulary for neural machine translation*. In ACL. 2015.
- [Luong and al., 2015]: Luong, Minh-Thang, et al. "*Addressing the rare word problem in neural machine translation.*" Proceedings of ACL. 2015.
- [Vukotic and al., 2015]: Vukotic, Vedran, Christian Raymond, and Guillaume Gravier. "*Is it time to switch to Word Embedding and Recurrent Neural Networks for Spoken Language Understanding?*." InterSpeech. 2015.
- [Deng & al.,2012]: Deng, L., Tur, G., He, X., & Hakkani-Tur, D. (2012, December). Use of kernel deep convex networks and end-to-end learning for spoken language understanding. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*(pp. 210-215). IEEE.
- Joseph Reisinger and Raymond J. Mooney. "Multi-prototype vector-space models of word meaning." *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010.
- Huang, Eric H., et al. "Improving word representations via global context and multiple word prototypes." *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012.

References

- [Neelakantan and al,2014]: Arvind Neelakantan, Jeevan Shankar, Alexandre Passos and Andrew McCallum "*Efficient non-parametric estimation of multiple embeddings per word in vector space*." Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Quatar, 2014
- [Neelakantan and Collins, 2014]: Arvind Neelakantan and Michael Collins. *Learning dictionaries for named entity recognition using minimal supervision*. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 452–461, Gothenburg, Sweden, April. Association for Computational Linguistics. 2014
- [Passos et al, 2014]: Alexandre Passos, Kumar Vineet and Andrew McCallum. "*Lexicon Infused Phrase Embeddings for Named Entity Resolution*." CoNLL-2014 (2014)
- [Turian et al, 2010]: Joseph Turian, Lev Ratinov, and Yoshua Bengio. *Word representations: a simple and general method for semi-supervised learning.* In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pages 384–394. Association for Computational Linguistics. 2010.
- [Bansal et al, 2014]: Mohit Bansal, Kevin Gimpel and Karen Livescu. "*Tailoring continuous word representations for dependency parsing.*" Proceedings of the Annual Meeting of the Association for Computational Linguistics. 2014.
- [Dhillon and al., 2011]: Paramveer S. Dhillon, Dean Foster, and Lyle Ungar. *Multi-View Learning of Word Embeddings via CCA*. Advances in Neural Information Processing Systems (NIPS). 2011.
- [Maas et al, 2011]: Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011, June). *Learning word vectors for sentiment analysis*. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1 (pp. 142-150). Association for Computational Linguistics.

References

- [Neelakantan and al,2014]: Arvind Neelakantan, Jeevan Shankar, Alexandre Passos and Andrew McCallum "*Efficient non-parametric estimation of multiple embeddings per word in vector space*." Proceedings

- [Socher et al, 2011]: R. Socher, E.H. Huang, J. Pennington, A.Y. Ng, and C.D. Manning. *Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection*. In NIPS, 2011.

- [Le and Mikolov, 2014]: Quoc Le and Tomas Mikolov. "*Distributed Representations of Sentences and Documents.*" Proceedings of the 31st International Conference on Machine Learning (ICML-14). 2014.

- [Mikolav and al., 2013]: Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. *Efficient Estimation of Word Representations in Vector Space*. Workshop at International Conference on Learning Representations (ICLR).

- [Mikolav and al., 2013b]: Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. *Distributed Representations of Words and Phrases and their Compositionality*. Advances in Neural Information Processing Systems (NIPS).

Other interesting references:

- [Hinton & al.,2012]: Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modelling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, *29*(6), 82-97.

- [Krizhevsky & al.,2012]: Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

- [Cho & al.,2014]: Cho, K., Gulcehre, B. V. M. C., Bahdanau, D., Schwenk, F. B. H., & Bengio, Y. (2014)Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *the proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP2014)*

- [Deng & al.,2012]: Deng, L., Tur, G., He, X., & Hakkani-Tur, D. (2012, December). Use of kernel deep convex networks and end-to-end learning for spoken language understanding. In *Spoken Language Technology Workshop (SLT), 2012 IEEE*(pp. 210-215). IEEE.

- [Kim & al.,2003]: Kim, H. J., Jordan, M. I., Sastry, S., & Ng, A. Y. (2003). Autonomous helicopter flight via reinforcement learning. In Advances in neural information processing systems.

- [Erhan and al., 2010]: Erhan, D., Bengio, Y., Courville, A., Manzagol, P. A., Vincent, P., & Bengio, S. (2010). Why does unsupervised pre-training help deep learning?. The Journal of Machine Learning Research, 11, 625-660.