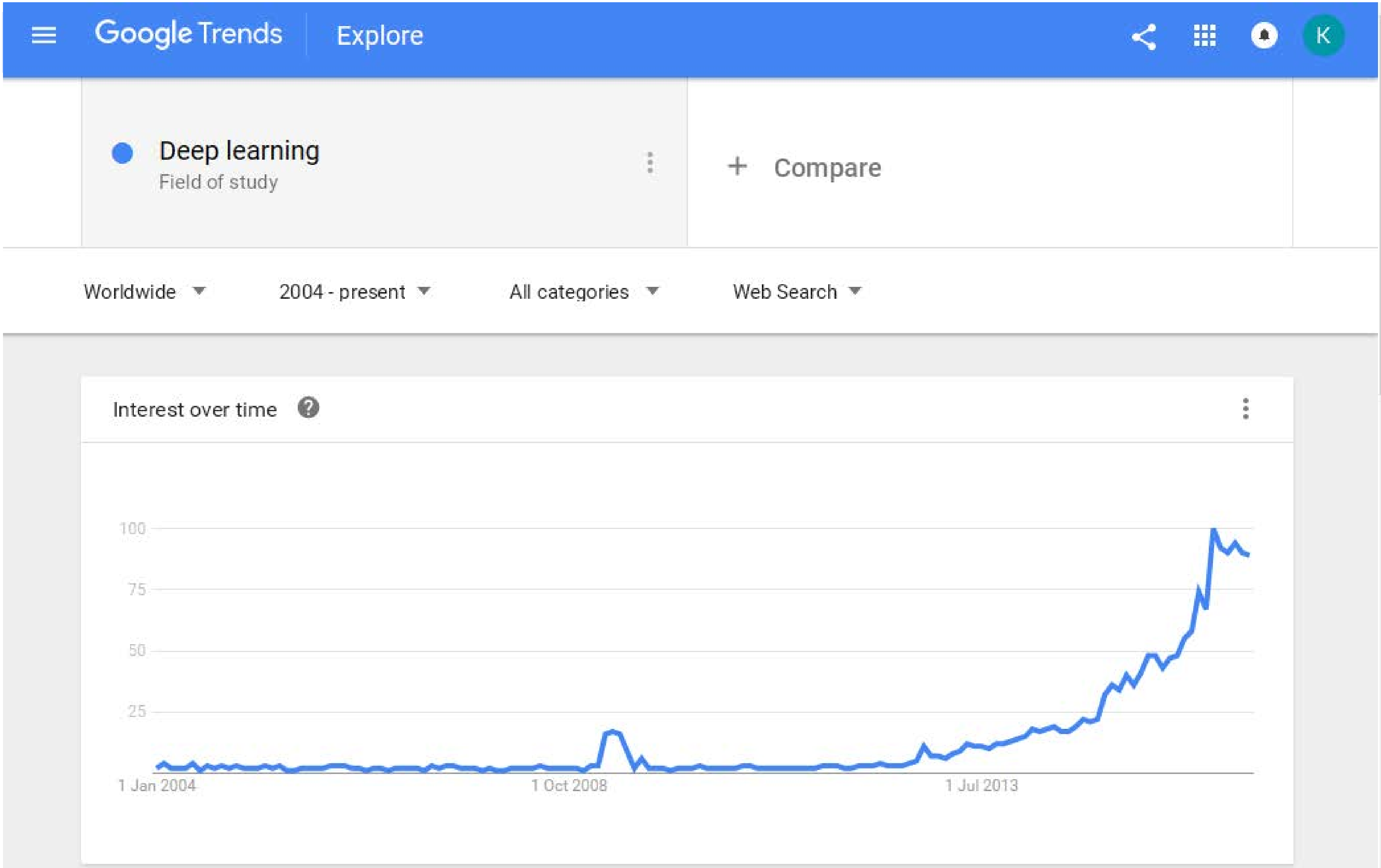
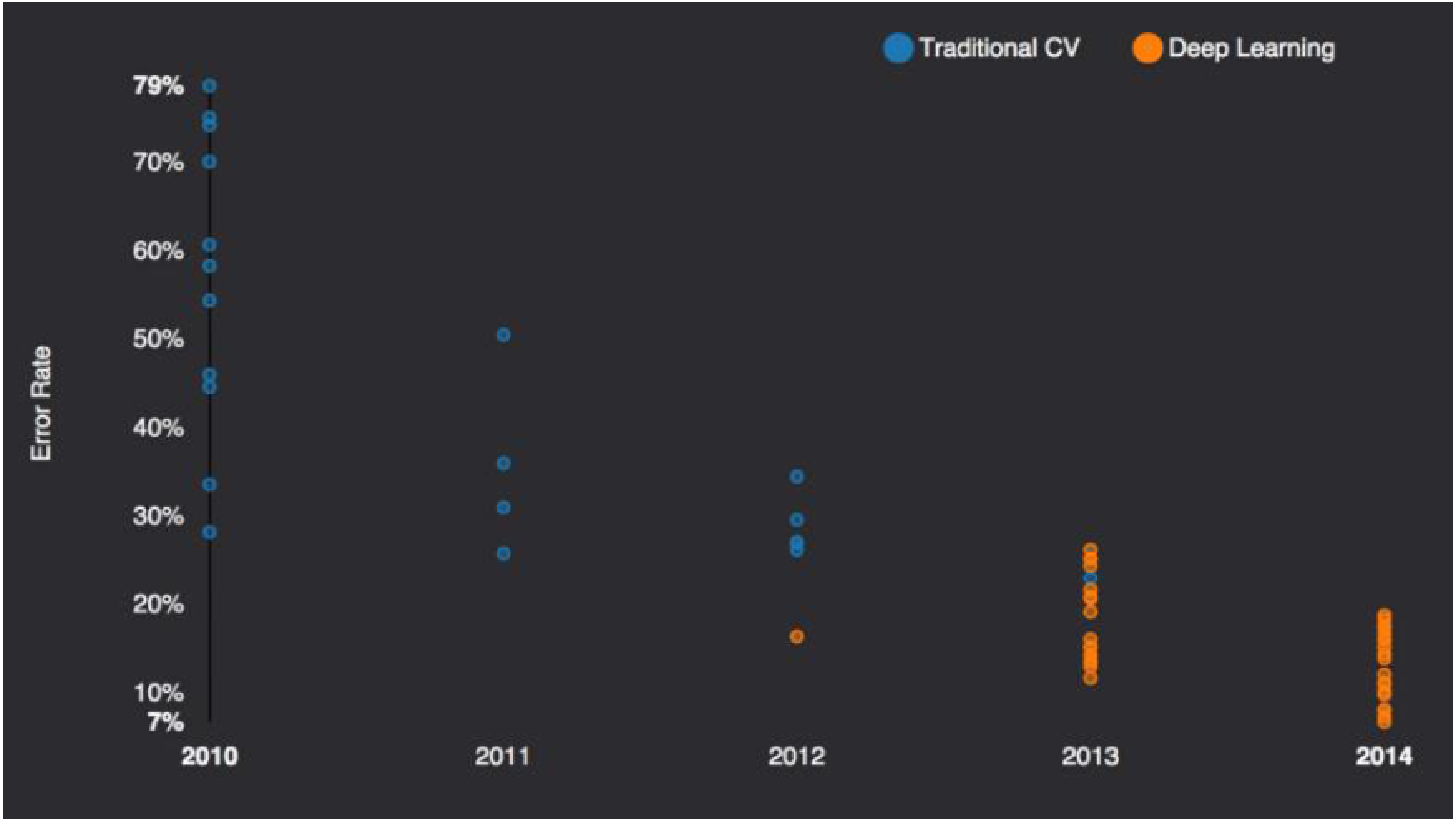
Text Mining with Deep Learning


Google Trends
Explore
K
Deep learning
Field of study
+ Compare
Worldwide
2004 - present
All categories
Web Search
Interest over time
100
75
50
25
1 Jan 2004
1 Oct 2008
1 Jul 2013



ImageNet: The "computer vision World Cup"

Beats state of the art in many areas

- Language Modeling (2012, Mikolov et al)
- Image Recognition (Krizhevsky won 2012 ImageNet competition)
- Sentiment Classification (2011, Socher et al)
- Speech Recognition (2010, Dahl et al)
- MNIST hand-written digit recognition (Ciresan et al, 2010)

Other Successful Applications

- Automatic summarization
- Coreference resolution
- Discourse analysis
- Machine translation
- Morphological segmentation
- Named entity recognition (NER)
- Natural language generation
- Word sense disambiguation
- Relationship extraction
- Speech processing
- Part-of-speech tagging
- sentence boundary disambiguation
- Sentiment analysis
- Optical character recognition (OCR)
- Question answering
- Parsing
- Word segmentation
- Natural language understanding
- Information retrieval (IR)
- Speech recognition
- Topic segmentation and recognition
- Speech segmentation
- Information extraction (IE)

What's so great about DL?

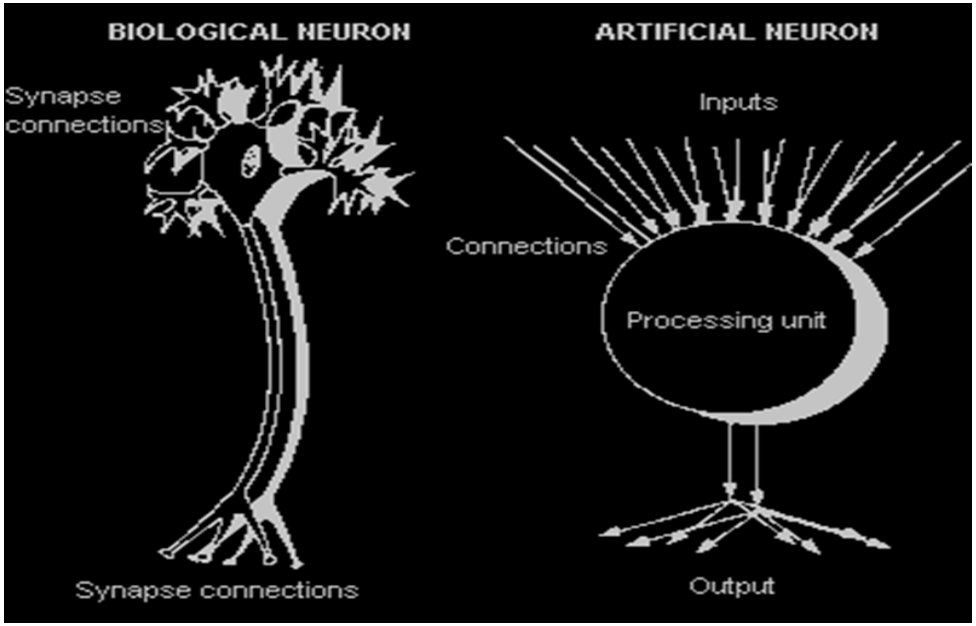
- You probably have read (or heard) more..
- In this lecture, I will attempt to
  - » Explain precisely why everyone is so excited about DL (Part 1)
  - » Show you how it's able to perform so well (Part 2: Case studies)
  - » And, let you see (a bit of it) for yourself..! (Part 3: Workshop)

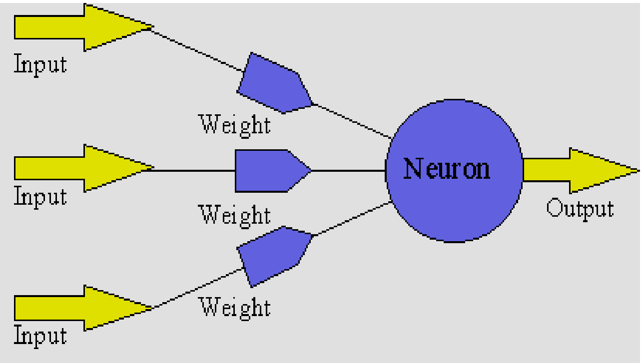
Background

Biological Inspiration

- Human brain has ten billion (10^{10}) neurons
- Neuron switching time $>10^{-3}$secs
- Face Recognition ~0.1secs
- On average, each neuron has several thousand connections
- Hundreds of operations per second
- High degree of parallel computation
- Distributed representations

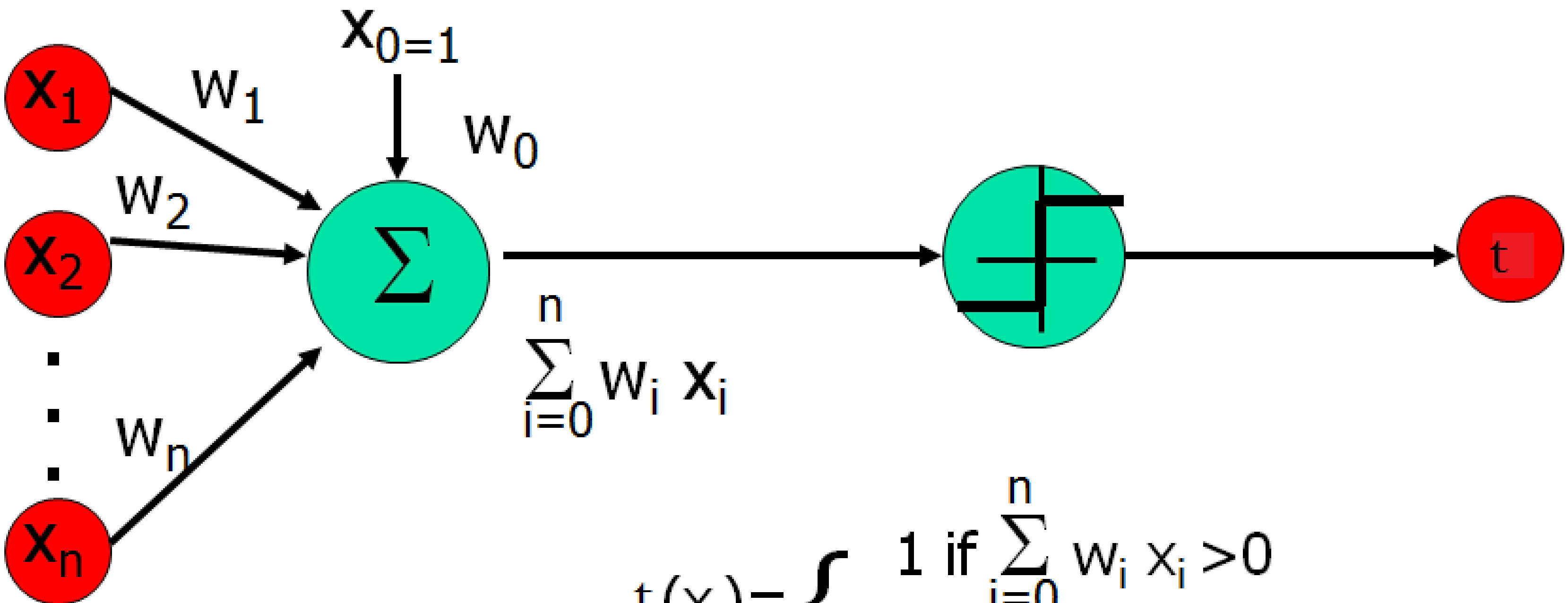
Biological Neuron-Artificial Neuron





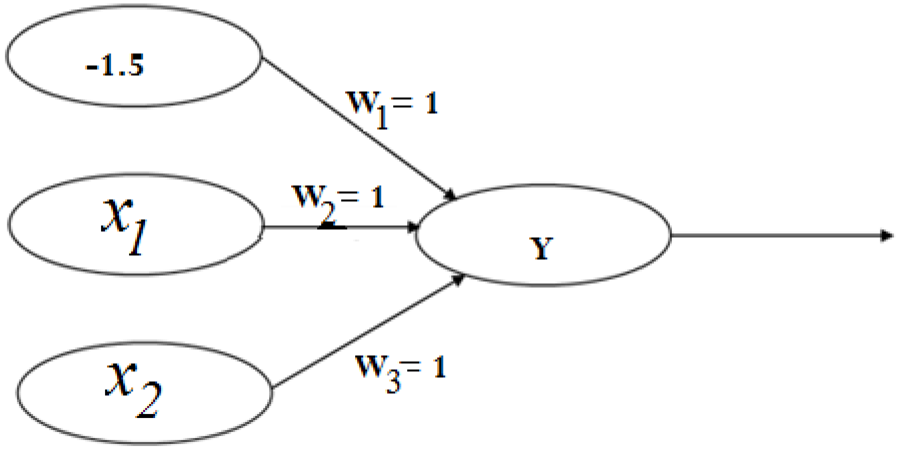
- Many simple neuron-like threshold switching units
- Many weighted interconnections among units
- Highly parallel, distributed processing
- Learning by tuning the connection weights

Perceptron: Linear Threshold Unit (1950's)



$$t(x_i) = \begin{cases} 1 \text{ if } \sum_{i=0}^{n} w_i x_i > 0 \\ -1 \text{ otherwise} \end{cases}$$

Perceptrons for Logical AND



Training data set
for Logical AND

x_1	x_2	t
0	0	0
0	1	0
1	0	0
1	1	1

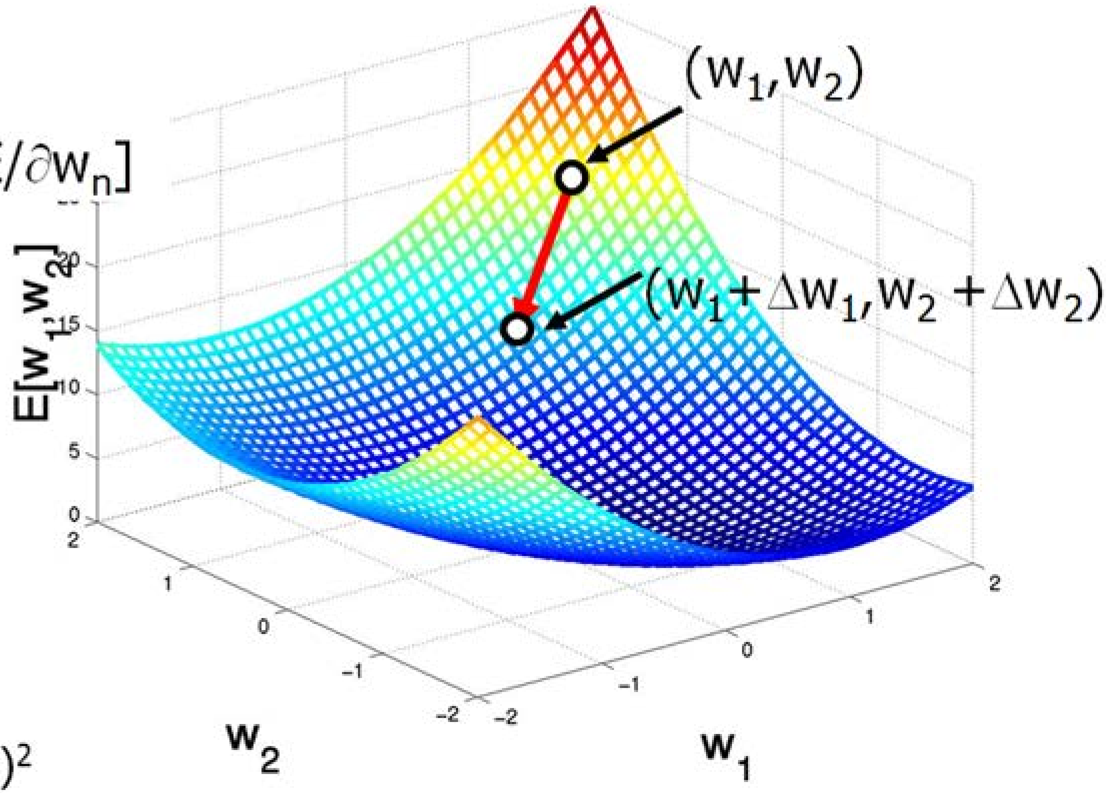
	x_1	x_2	Summation	Output (t)
-1.5	0	0	(0*1)+(0*1)-1.5 = -1.5	0
-1.5	0	1	(0*1)+(1*1)-1.5 = -0.5	0
-1.5	1	0	(1*1)+(0*1)-1.5 = -0.5	0
-1.5	1	1	(1*1)+(1*1) -1.5 = 0.5	1

Gradient Descent Learning

Gradient:

$\nabla E[w]=[\partial E/\partial w_0, \ldots \ \partial E/\partial w_n]$

$\Delta w = -\eta \ \nabla E[w]$



$\Delta w_i = -\eta \ \partial E/\partial w_i$

$= \partial/\partial w_i \ 1/2 \sum (t_n - y_n)^2$

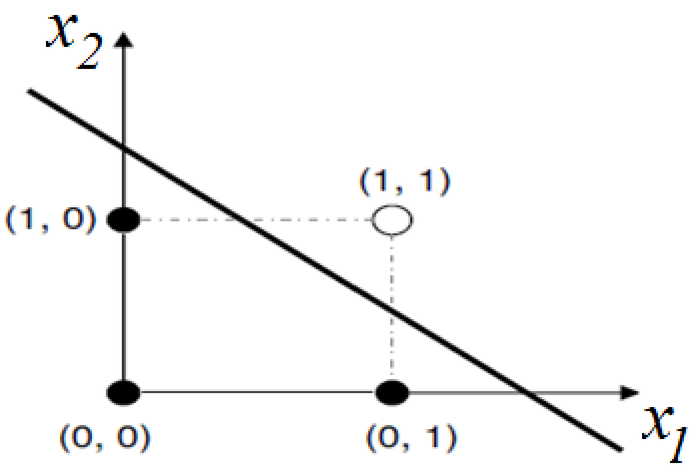
$= \partial/\partial w_i \ 1/2 \sum_d (t_n - \sum_i w_i x_i)^2$

$= \sum_n (t_n - y_n)(-x_i)$

Linearly Separable Property

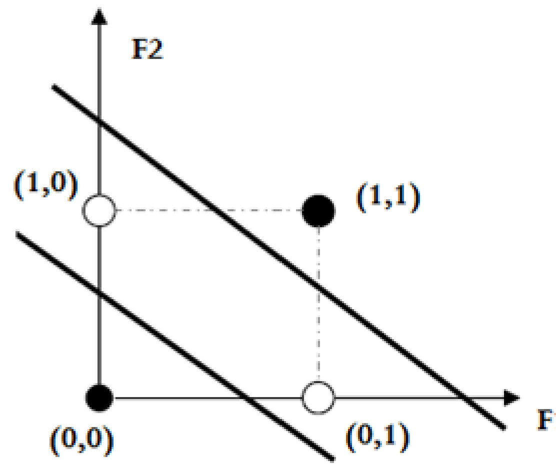
Logical AND

AND		
x_1	x_2	t
0	0	0
0	1	0
1	0	0
1	1	1

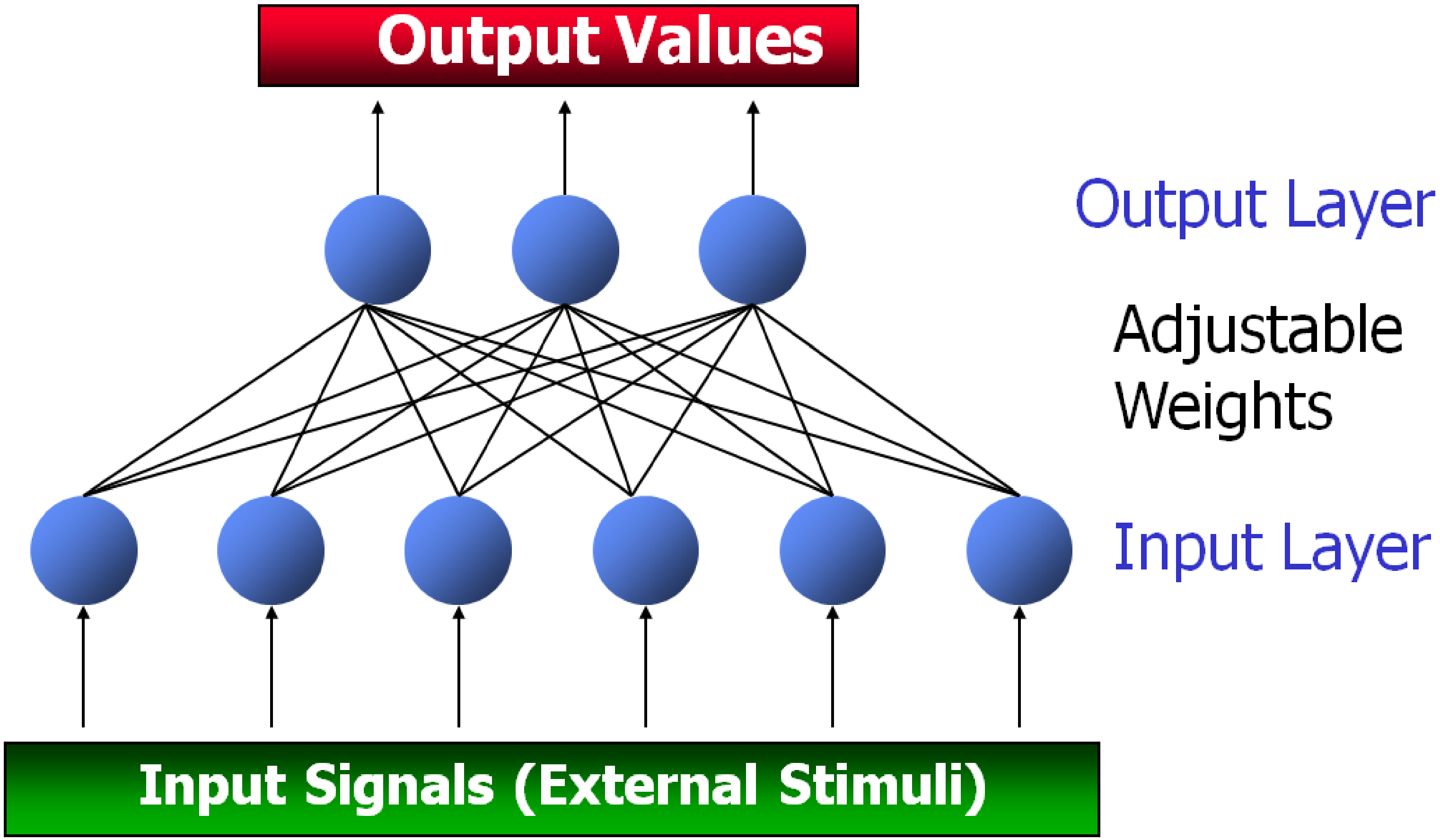


Logical XOR

XOR		
x_1	x_2	t
0	0	0
0	1	1
1	0	1
1	1	0



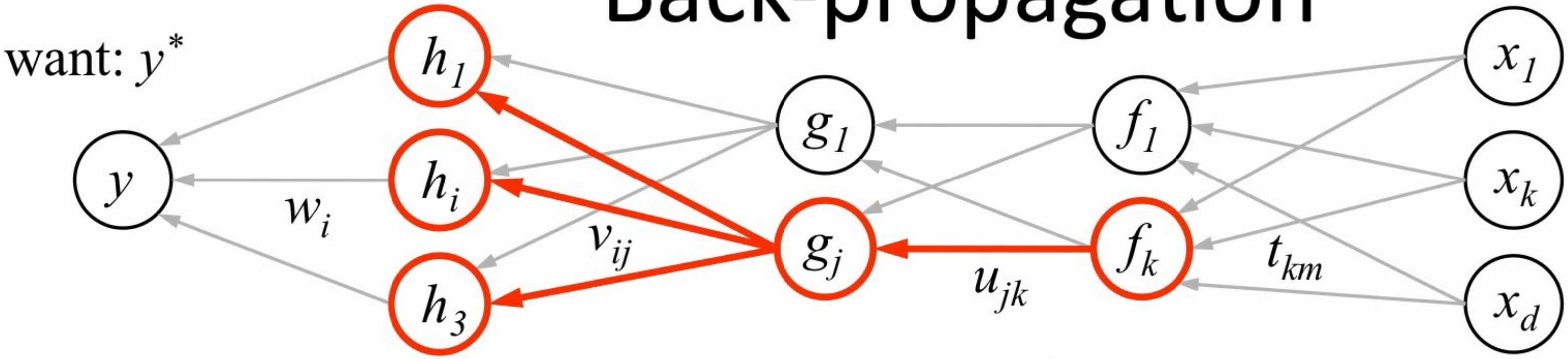
Multi-layer Perceptron Network (1960-80's)



Layers of a MLP

- The input layer
  - Introduces input values into the network.
- The hidden layer(s)
  - Perform classification of features
  - Often one or two hidden layers
- The output layer
  - Functionally just like the hidden layers
  - Outputs are passed on to the world outside the neural network.

Back-propagation



1. receive new observation $\boldsymbol{x} = [x_1 \ldots x_d]$ and target y^*
2. **feed forward:** for each unit g_j in each layer 1…L
   compute g_j based on units f_k from previous layer: $g_j = \sigma\left(u_{j0} + \sum_k u_{jk} f_k\right)$
3. get prediction y and error $(y\text{-}y^*)$
4. **back-propagate error:** for each unit g_j in each layer L…1

(a) compute error on g_j

$$\underbrace{\frac{\partial E}{\partial g_j}}_{\text{should } g_j \text{ be higher or lower?}} = \sum_i \underbrace{\sigma'(h_i) v_{ij}}_{\text{how } h_i \text{ will change as } g_j \text{ changes}} \underbrace{\frac{\partial E}{\partial h_i}}_{\text{was } h_i \text{ too high or too low?}}$$

(b) for each u_{jk} that affects g_j

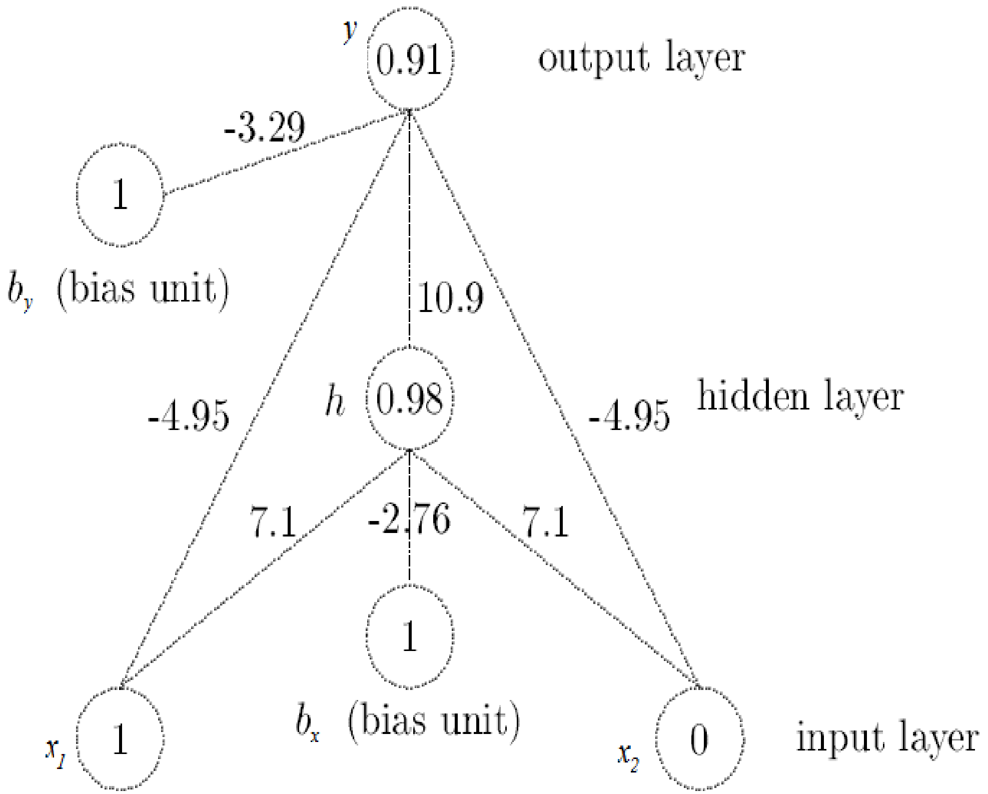
(i) compute error on u_{jk}

$$\frac{\partial E}{\partial u_{jk}} = \underbrace{\frac{\partial E}{\partial g_j}}_{\text{do we want } g_j \text{ to be higher/lower}} \underbrace{\sigma'(g_j) f_k}_{\text{how } g_j \text{ will change if } u_{jk} \text{ is higher/lower}}$$

(ii) update the weight

$$u_{jk} \leftarrow u_{jk} - \eta \frac{\partial E}{\partial u_{jk}}$$

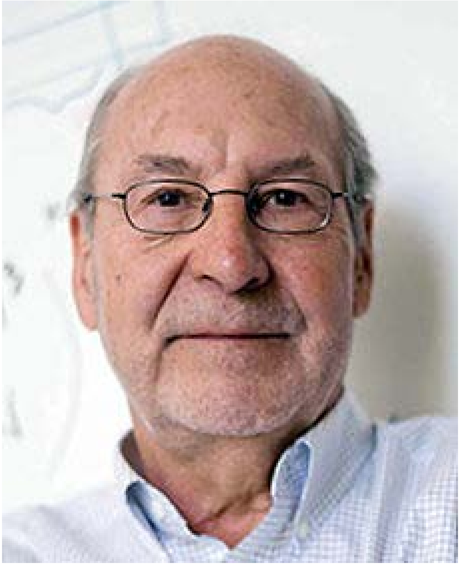
XOR with MLP



y
0.91
output layer
-3.29
1
b_y (bias unit)
10.9
-4.95
h
0.98
-4.95
hidden layer
7.1
-2.76
7.1
1
b_x (bias unit)
x_1
1
x_2
0
input layer

Power of MLP's

- MLP with 1 hidden layer can represent
  - » Any boolean function
  - » Any bounded continuous function



Cybenko 1989

But..

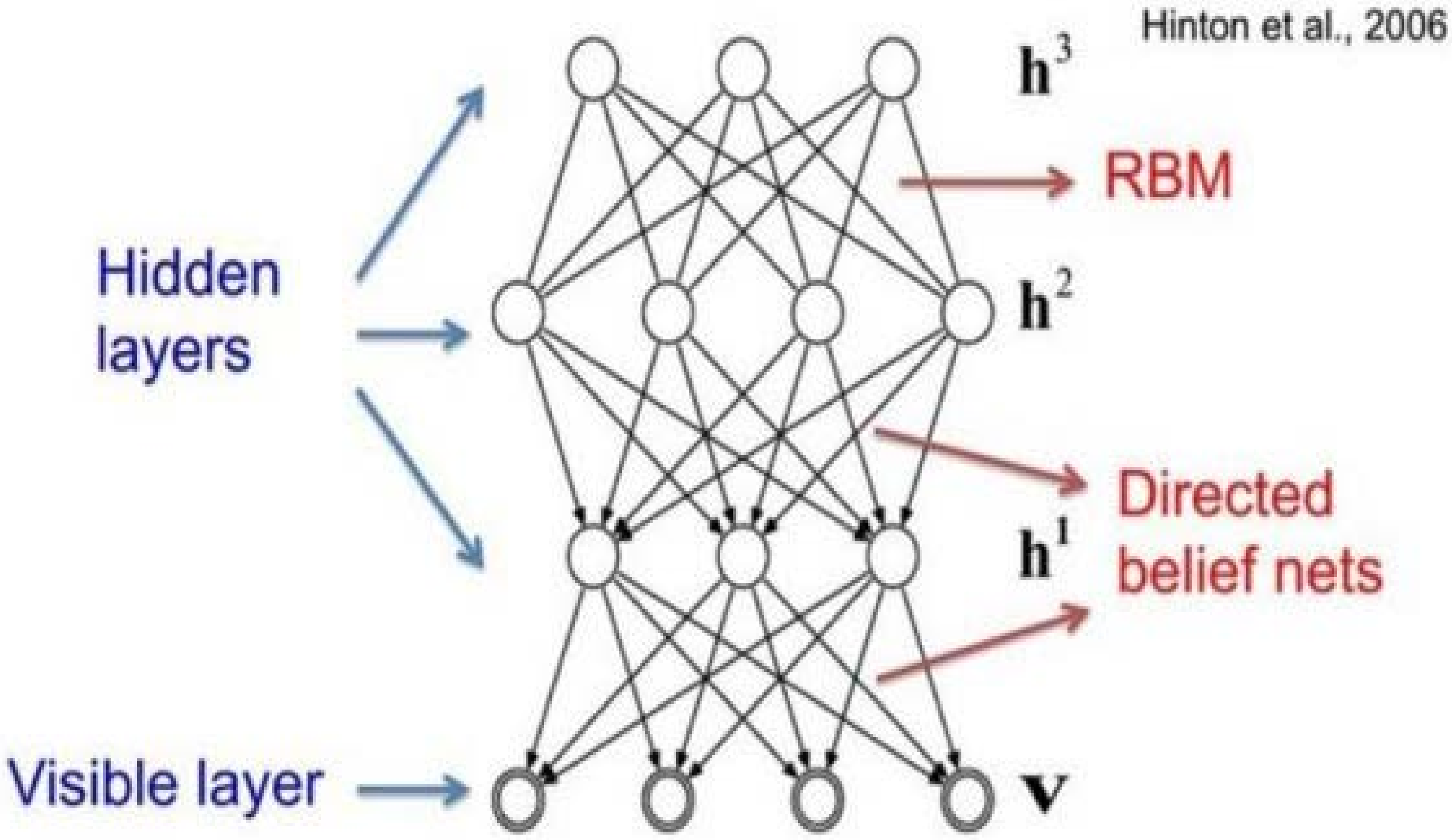
- Requires labeled data
  - » Most data is unlabeled
- Backpropagation solutions may be trapped in local minima
- Large networks (> 2 hidden layers) are harder to train
  - » Vanishing gradients
- Overfitting becomes a serious issue

Fast forward to 2006…

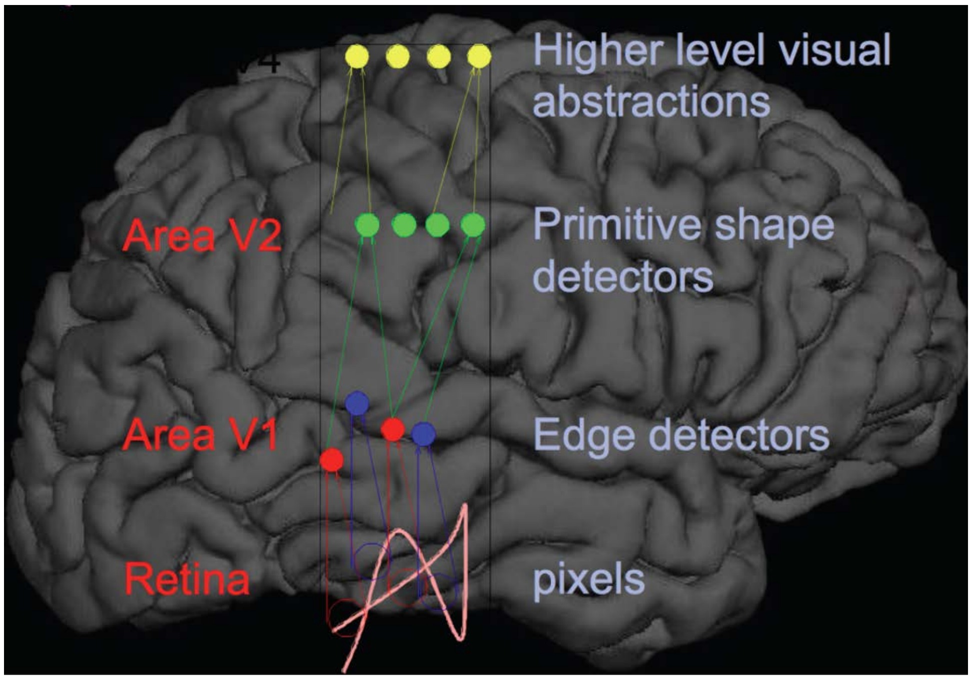
Breakthrough work by GE Hinton's group

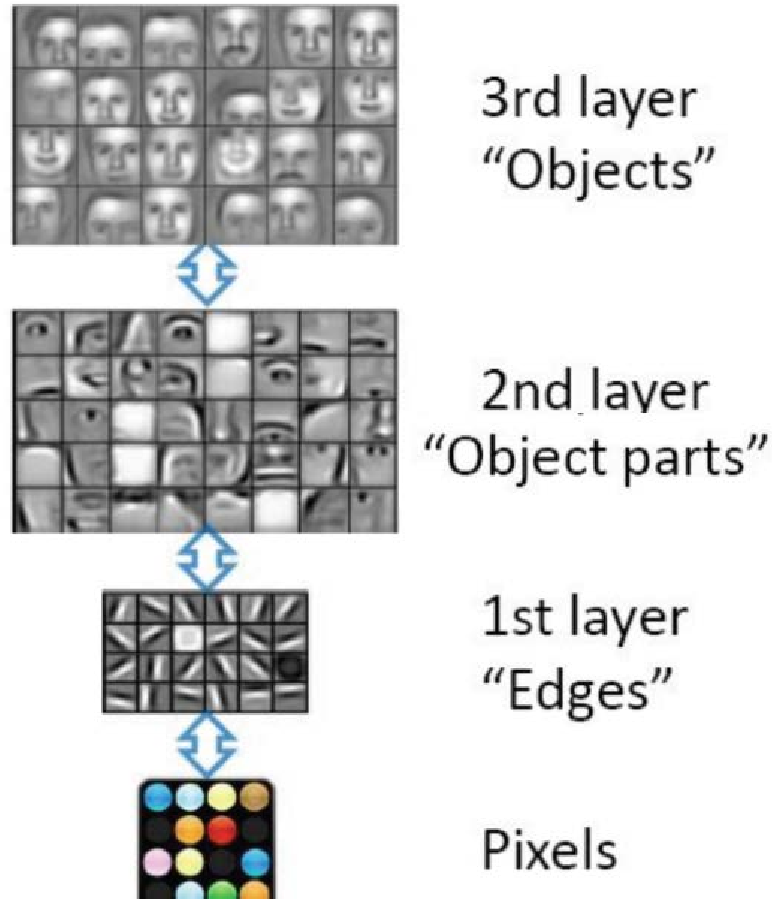
- Proposed simple neural nets called Restricted Boltzmann Machines (RBM), and stacked them to develop Deep Belief Networks (DBN's)
- Came up with a fast algorithm for training them
- Demonstrated DBN beats state-of-the-art significantly

Stacked RBM's

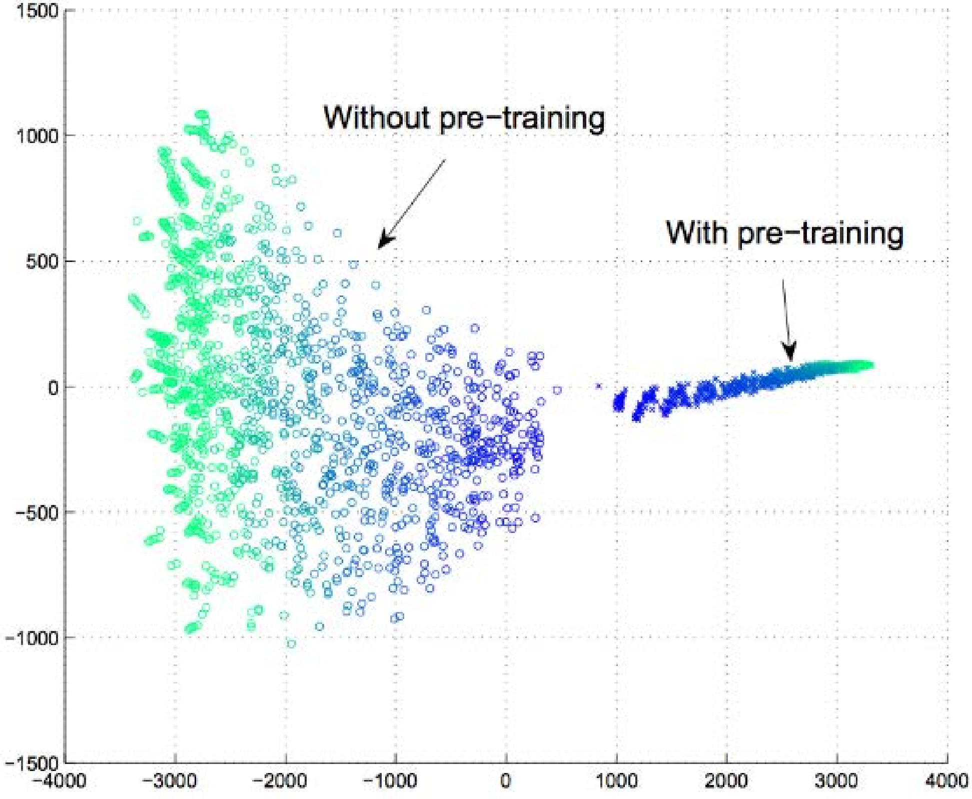


Learning Representations



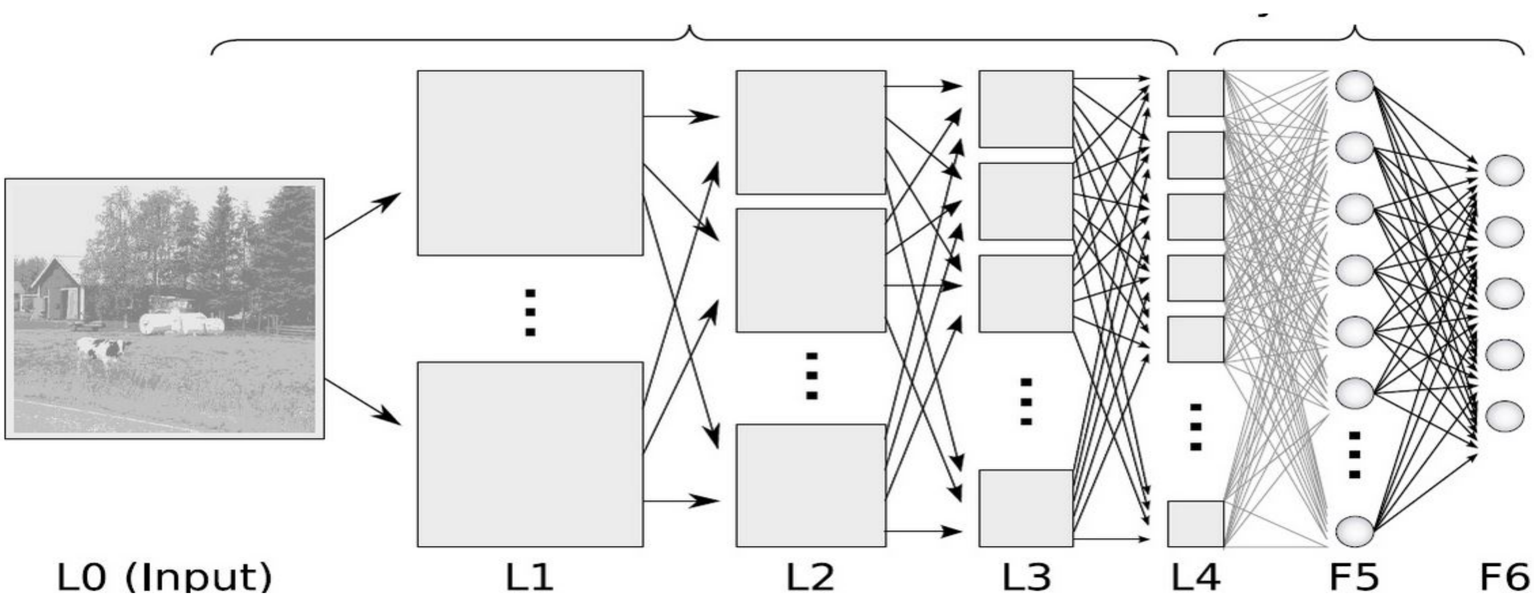


Key Insight – Unsupervised Pre-training



Deep Learning: Key Aspects

- Multiple layers of processing units
- Supervised or unsupervised learning of feature representations in each layer (*pre-training*)
- Layers form a hierarchy (low-level to high-level features)



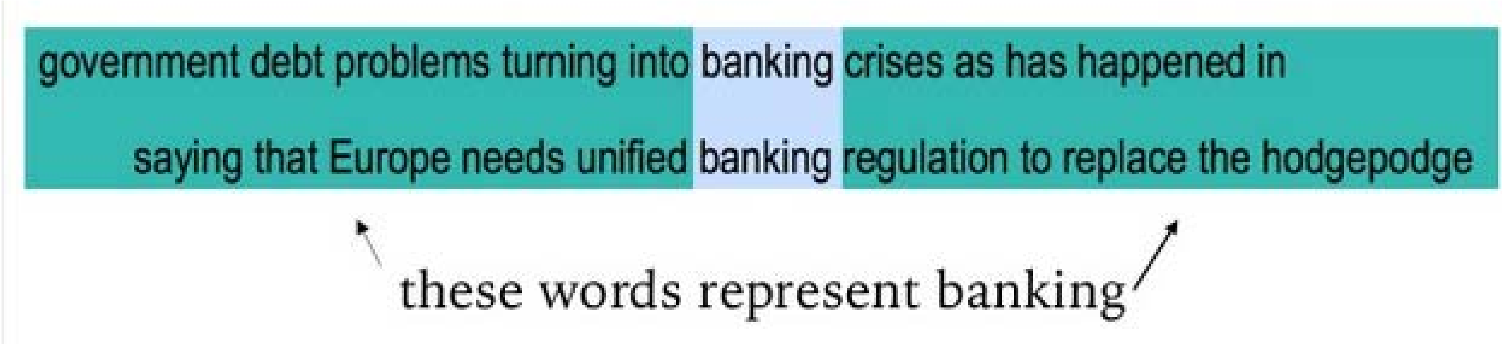
Deep Learning Approaches to Text Mining

Background

- Recap: the fundamental issue in Text mining
  - » The Semantic Gap!
- Text Semantics can appear via..
  - » Words
  - » Compositions of words (sentence, para)
  - » More complex compositions (Documents, Collections)
- Hard to represent computationally!

Distributional Representation

- "*You shall know a word by the company it keeps*" (JR Firth, 1957)
- Most successful idea in modern statistical NLP



- Also useful in DL

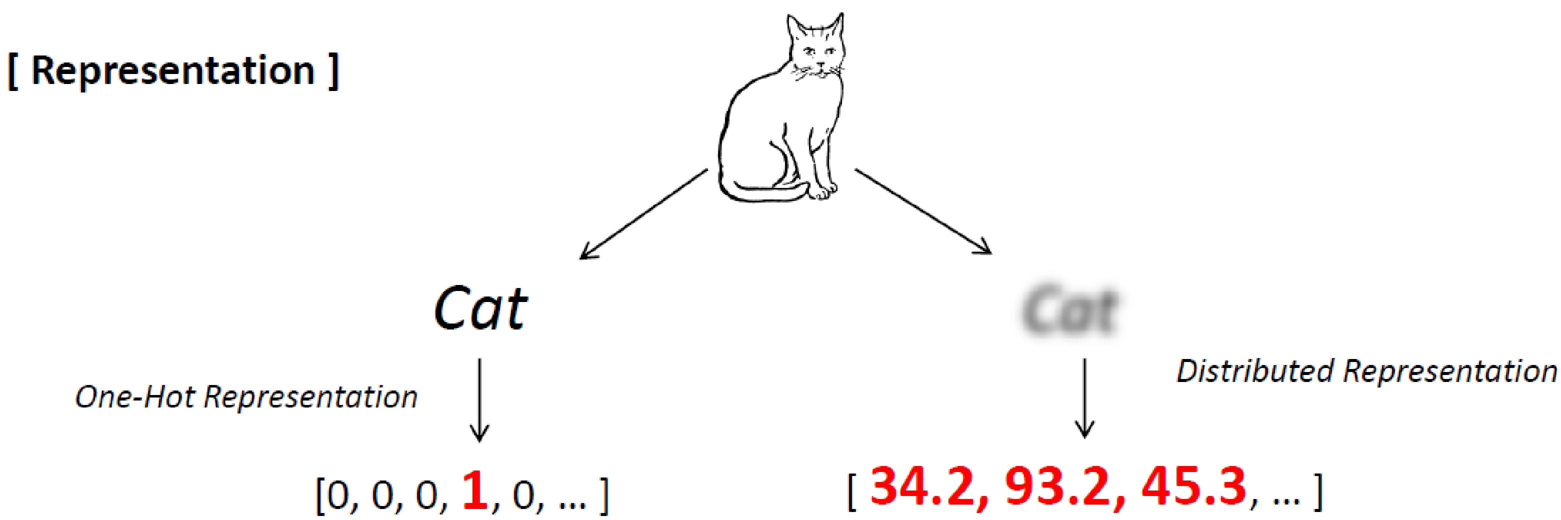
Deep Learning Approaches

- Word Embeddings (Word2Vec)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)

Word2Vec

- By Tomas Mikolov & team at Google (2013)
- Simple and a very successful approach to construct *word embeddings*
- Empirically showed that the model has better syntactic and semantic representation than previous models

Word Embedding: Example



Think of it as a ‘*word context vector*’

Word2Vec Philosophy

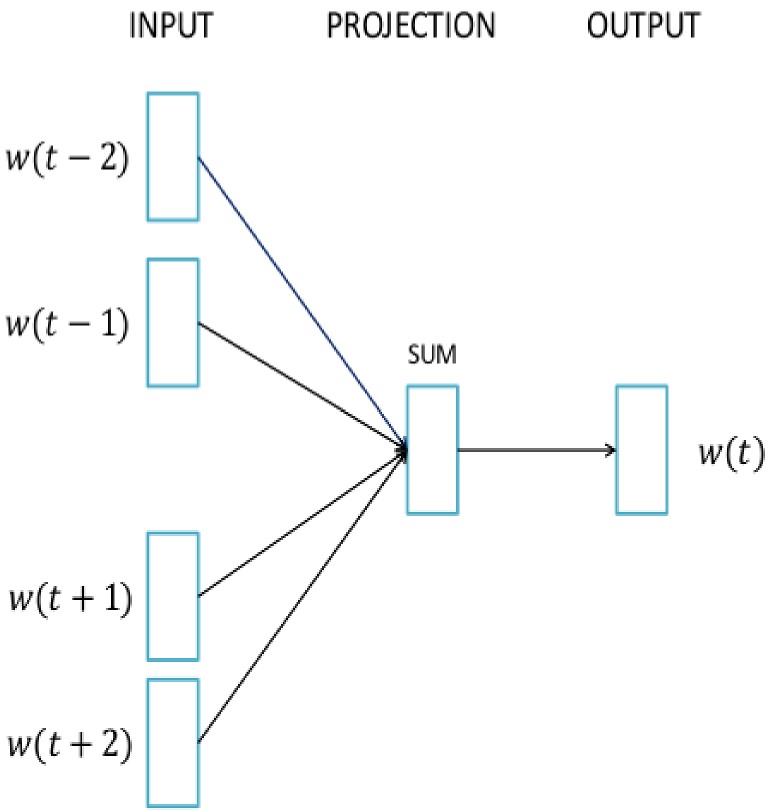
- Gather lots and lots of texts
- Apply DL to learn word embeddings
  - » Continuous bag of word model
  - » Skip gram model
- Use word embeddings for text mining (classification, clustering, etc.)

Word2Vec is basically a *pre-training strategy*

Continuous-Bag-of-word (CBOW) model

- Idea: *Given context words, can we predict center word*

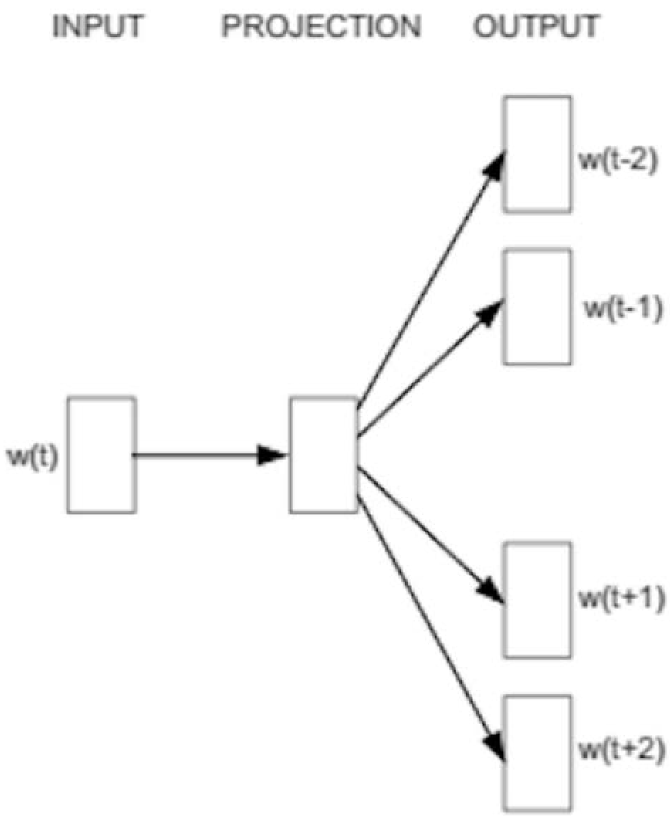
i.e. Probability(“It is (**?**) to finish” → “**time**”)



Skip-Gram model

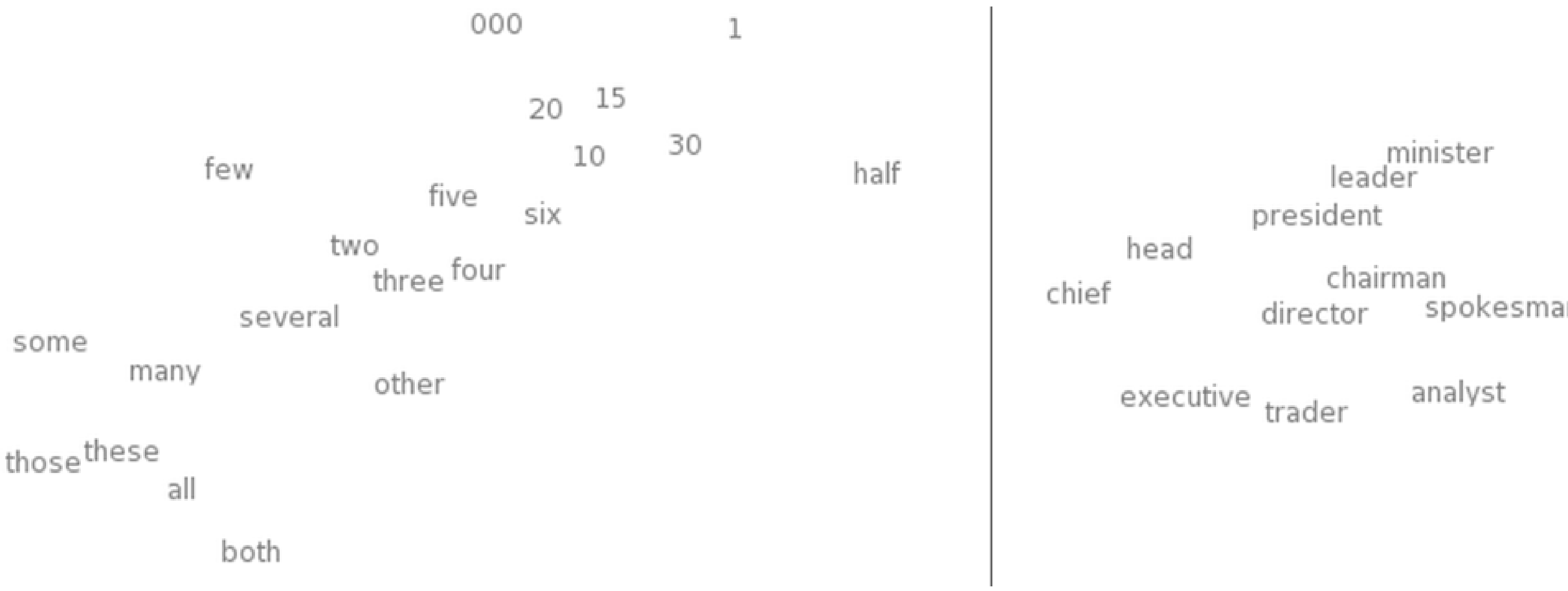
- Idea: *Given center word, can we predict context words*
- Mirror of CBOW (vice versa)

i.e. Probability("**time**" → "It is (**?**) to finish")



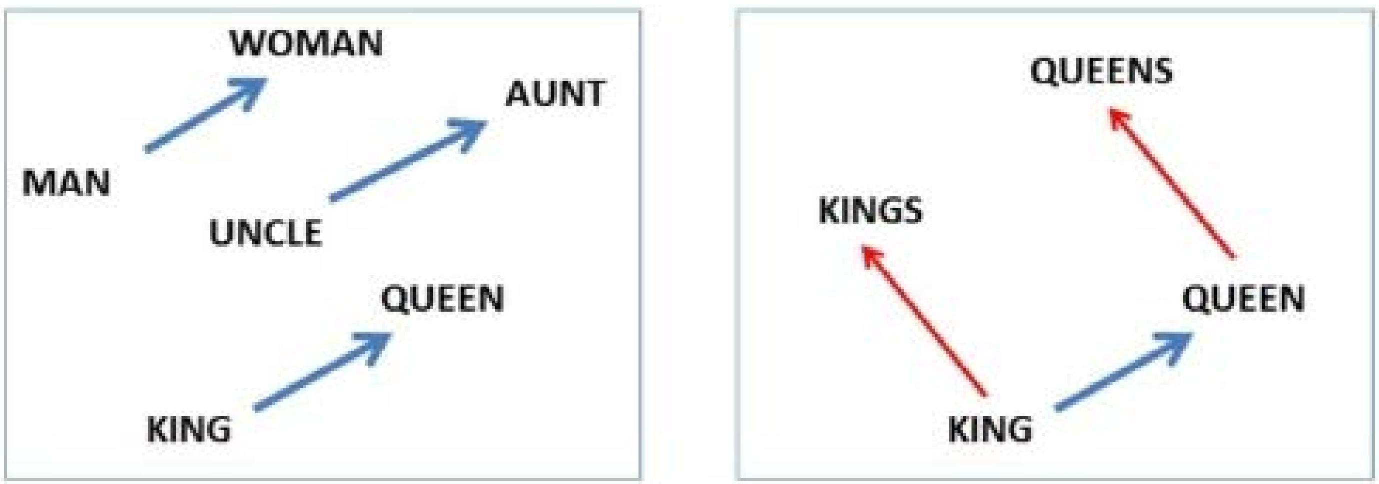
Skip-gram

Example of Representations Learnt



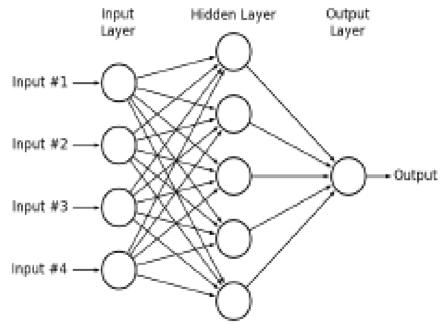
Exhibits Compositional Semantics

vec("man") – vec("king") + vec("woman") = vec("queen")



Real-life Usage

word	vector
The	(0.12, 0.23, 0.56)
Cardinals	(0.24, 0.65, 0.72)
will	(0.38, 0.42, 0.12)
win	(0.57, 0.01, 0.02)
the	(0.53, 0.68, 0.91)
world	(0.11, 0.27, 0.45)
series	(0.01, 0.05, 0.62)



Average Pooling

word	vector
The	(0.12, 0.23, 0.56)
Cardinals	(0.24, 0.65, 0.72)
will	(0.38, 0.42, 0.12)
win	(0.57, 0.01, 0.02)
the	(0.53, 0.68, 0.91)
world	(0.11, 0.27, 0.45)
series	(0.01, 0.05, 0.62)

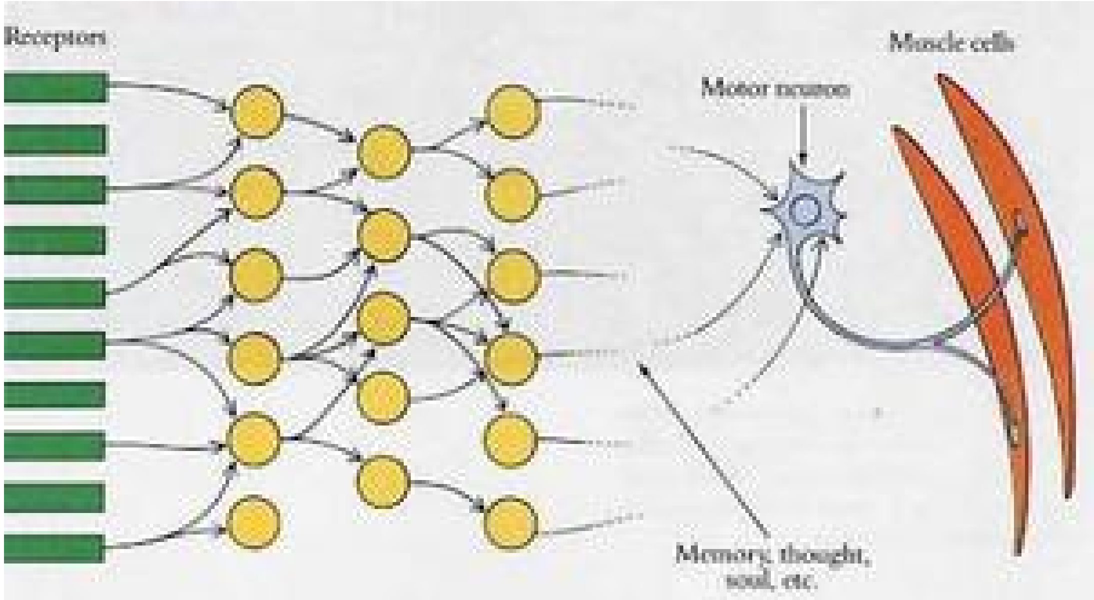
$\Rightarrow$

sentence vector

(0.28, 0.33, 0.49)

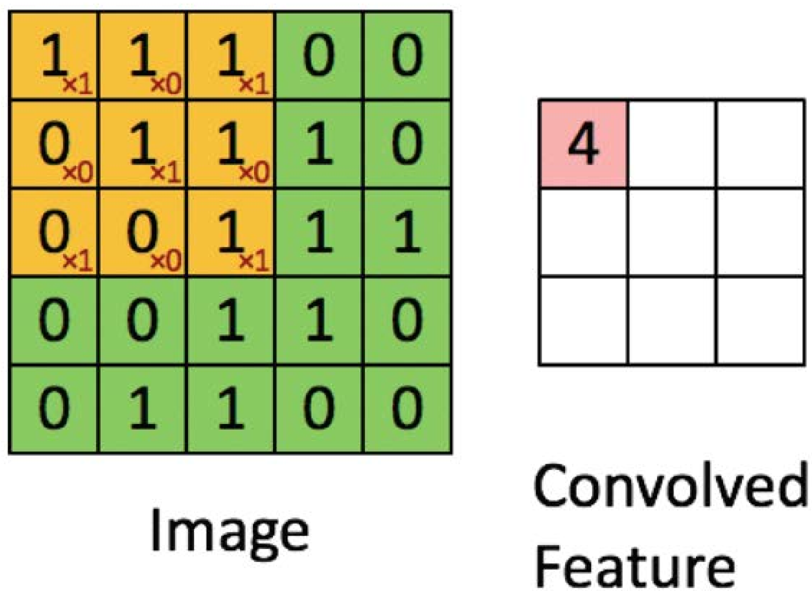
Convolutional Neural Networks (CNN's)

- Biological plausibility (Hubel & Wiesel 1959)

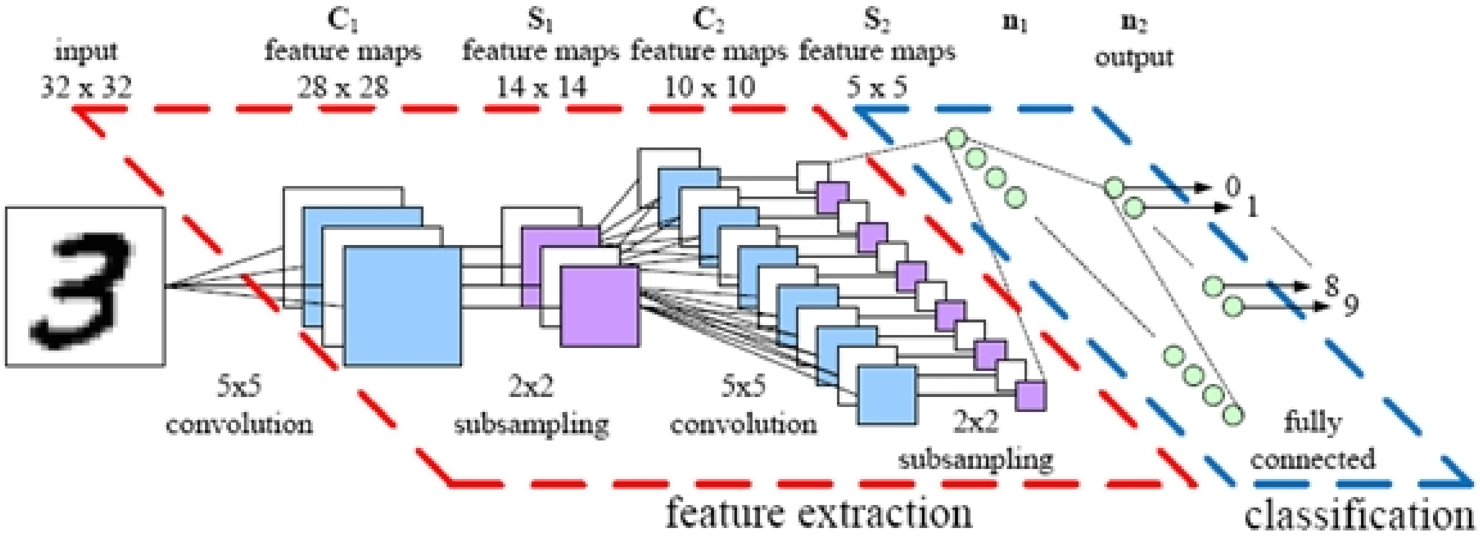


Convolutional Neural Networks (CNN's)

- Convolution is a kind of *blending* operator that is *slided* over to generate complex representations of data



Example CNN Architecture



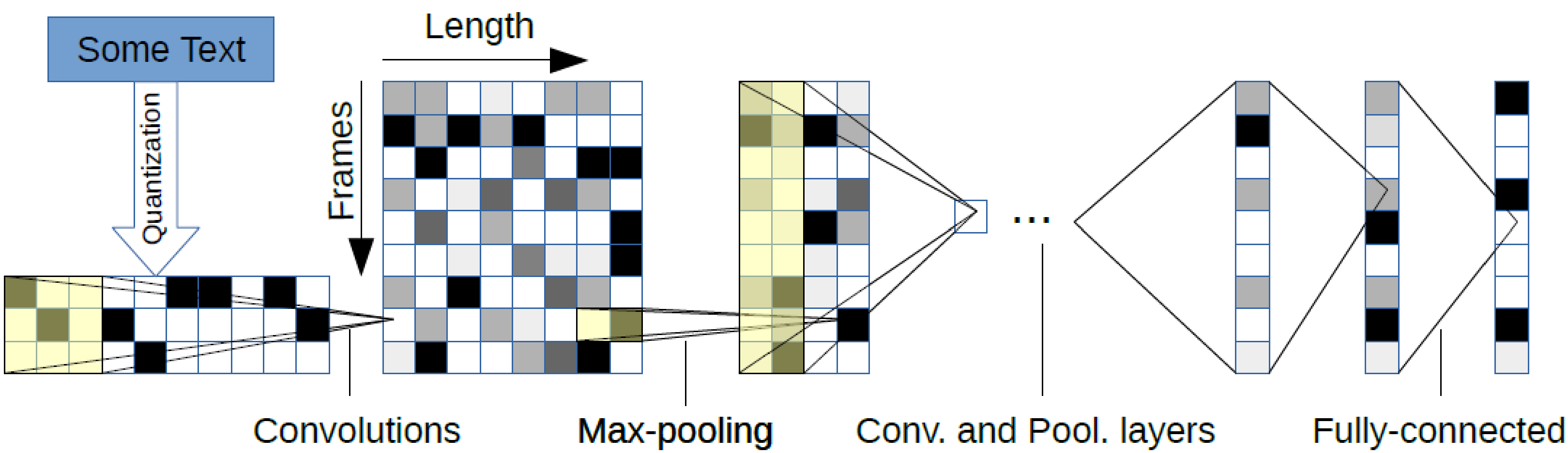
CNN - Key Principles

- Local receptive fields
  - » Ensures local connectivity
- Shared weights
  - » For parameter reduction
- Pooling
  - » For down-sampling
- Dropout
  - » Helps to avoid overfitting

CNN Approach for Text Mining

Text pre-processing

- Character Quantization
  - 69 characters - 26 English letters, 10 digits and 33 other characters
  - Convert each to a 69 size vector with 1 for the character and 0 for others
- Or, Use Word2Vec as input to CNN



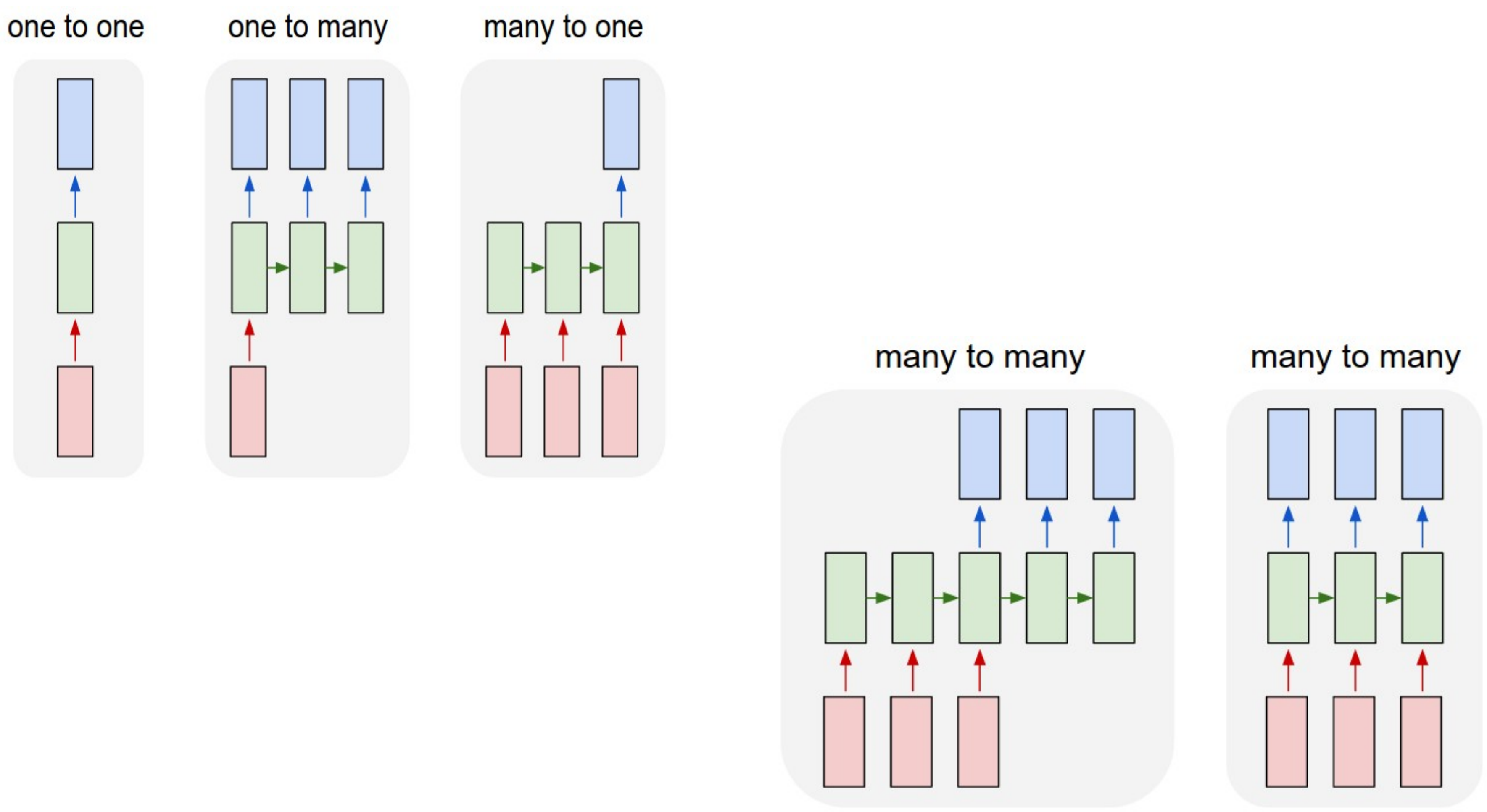
Performance on News Categorization

Model	Thesaurus	Train	Test
Large ConvNet	No	99.00%	91.12%
Large ConvNet	Yes	**99.00%**	**91.64%**
Small ConvNet	No	98.94%	89.32%
Small ConvNet	Yes	98.97%	90.39%
Bag of Words	No	88.35%	88.29%

Recurrent Neural Networks

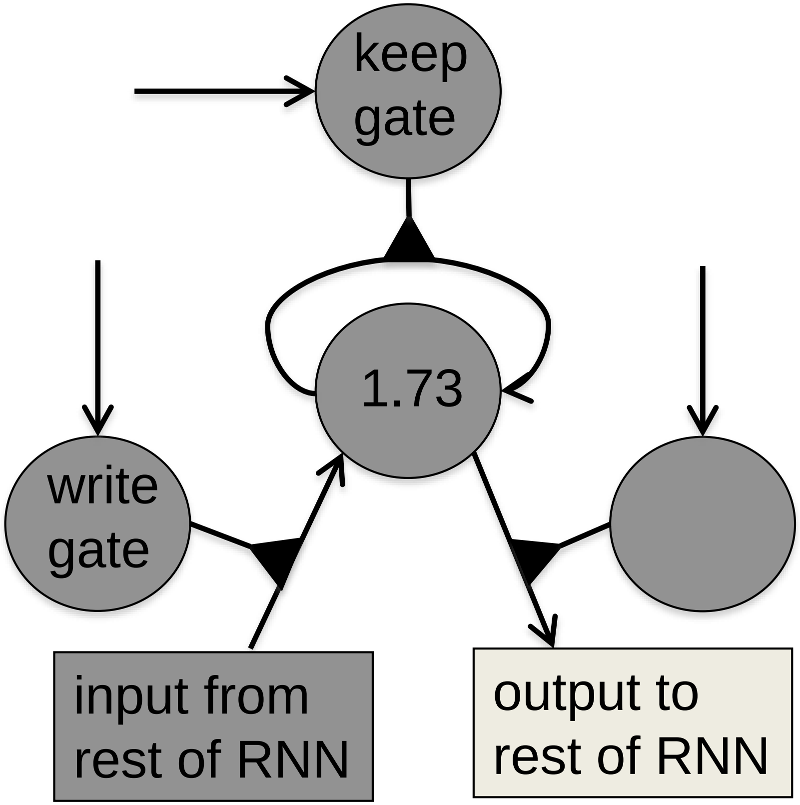
- Texts are typically varying in length, compared to standard structured data
- Word sequences in a text contribute to the semantics
- Word2Vec and CNN force fixed length representations
  - » E.g. word2vec average pooling
  - » Vector padding in CNN
- Recurrent neural networks can model texts more naturally

RNN Architectures

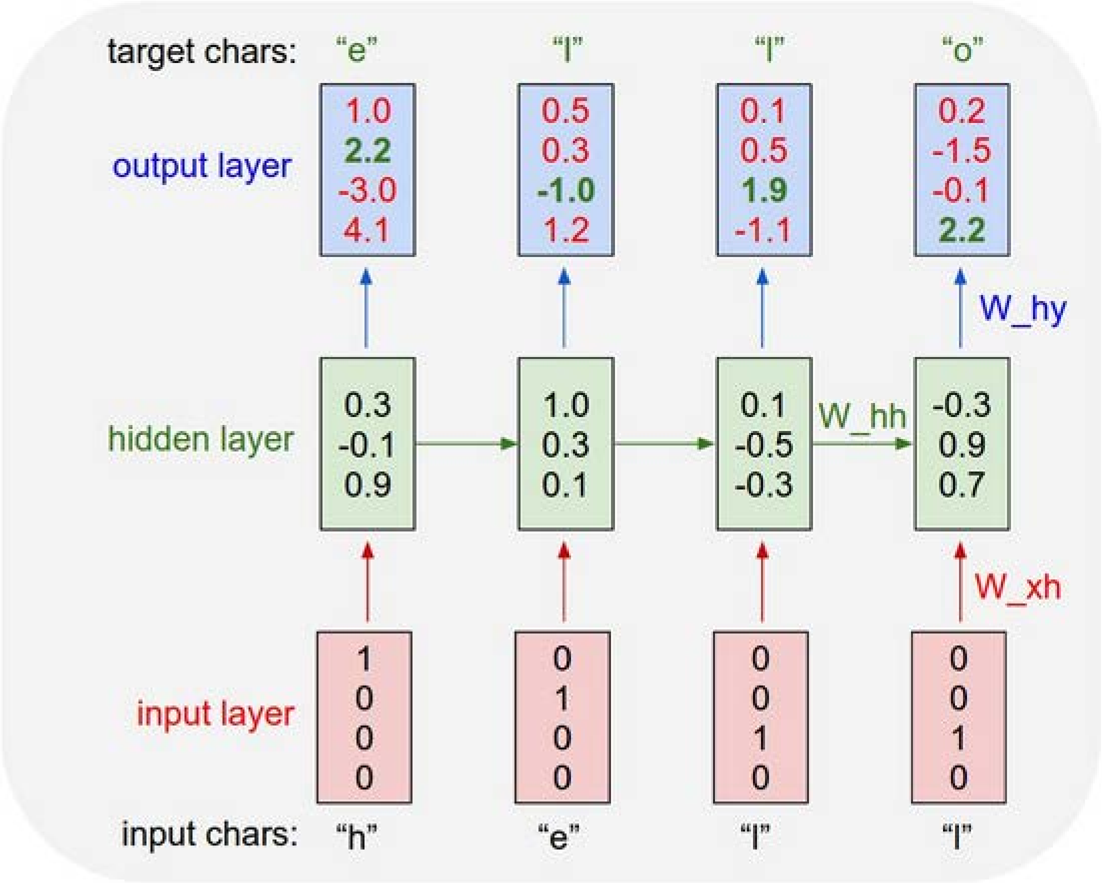


Long Short Term Memory (LSTM) Units

- Hochreiter & Schmidhuber proposed to solve the problem of getting an RNN to remember things for a long time (like hundreds of time steps).
- They designed a memory cell using logistic and linear units with multiplicative interactions.



Long Short-Term Memory (LSTM) based RNN



After passing Shakespeare Texts..

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

Source : Andrej Karpathy Blog

After passing LaTex Texts..

For $\bigoplus_{n=1,\dots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, **??** and the fact that any U affine, see Morphisms, Lemma **??**. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma **??** we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example **??**. It may replace S by $X_{spaces,\acute{e}tale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma **??**. Namely, by Lemma **??** we see that R is geometrically regular over S.

Lemma 0.1. *Assume (3) and (3) by the construction in the description.*

Suppose $X = \lim |X|$ (by the formal open covering X and a single map $\underline{Proj}_X(\mathcal{A}) = \mathrm{Spec}(B)$ over U compatible with the complex

$$Set(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X,\mathcal{O}_X}).$$

*When in this case of to show that $\mathcal{Q} \to \mathcal{C}_{Z/X}$ is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition **??** (without element is when the closed subschemes are catenary. If T is surjective we may assume that T is connected with residue fields of S. Moreover there exists a closed subspace $Z \subset X$ of X where U in X' is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem*

(1) *f is locally of finite type. Since $S = \mathrm{Spec}(R)$ and $Y = \mathrm{Spec}(R)$.*

Proof. This is form all sheaves of sheaves on X. But given a scheme U and a surjective étale morphism $U \to X$. Let $U \cap U = \coprod_{i=1,\dots,n} U_i$ be the scheme X over S at the schemes $X_i \to X$ and $U = \lim_i X_i$. □

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{X,\dots,0}$.

Lemma 0.2. *Let X be a locally Noetherian scheme over S, $E = \mathcal{F}_{X/S}$. Set $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$. Since $\mathcal{I}^n \subset \mathcal{I}^n$ are nonzero over $i_0 \leq \mathfrak{p}$ is a subset of $\mathcal{J}_{n,0} \circ \overline{A}_2$ works.*

Lemma 0.3. *In Situation **??**. Hence we may assume $\mathfrak{q}' = 0$.*

Proof. We will use the property we see that $\mathfrak{p}$ is the mext functor (**??**). On the other hand, by Lemma **??** we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where K is an F-algebra where δ_{n+1} is a scheme over S. □

After passing C sources..

```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
  rw->name = "Getjbbregs";
  bprm_self_clearl(&iv->version);
  regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
  return segtable;
}
```

Summary

- Deep learning provides new approaches for training deep neural architectures with multiple hidden layers
- DL beats the state-of-the-art in many data analytics problems
- Covered three major DL approaches to Text mining
  - » Word2Vec
  - » CNN
  - » LSTM based RNN

Where do we go from here..

Reading Materials

- Several good tutorials on Youtube; Also on slideshare
- Deep Learning for NLP – Stanford Course by Richard Socher
  - » Videos & Course material available
- Deep Learning Book – I. Goodfellow, Y. Bengio, A. Courville
  - » Chapter 12.4 – Applications to NLP

API's / Tools

- Python
  - » Theano/Keras
  - » Gensim (for word2vec)
  - » Pylearn2
- Java
  - » DeepLearning4j
- Others
  - » Torch
  - » Lasagne
  - » More: http://deeplearning.net/software_links/

Thank you