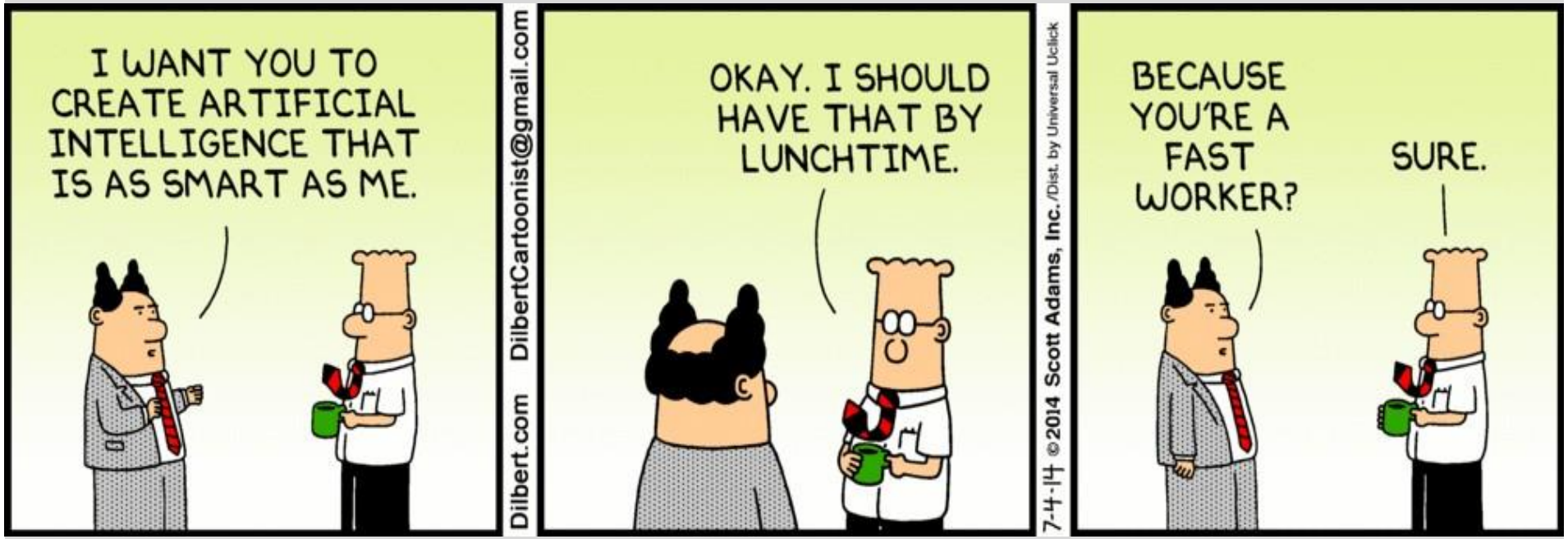
Introduction to Deep Learning

- Natural Language Processing

Swayam Mittal - Data Scientist,
Indegene

)


I WANT YOU TO CREATE ARTIFICIAL INTELLIGENCE THAT IS AS SMART AS ME.
OKAY. I SHOULD HAVE THAT BY LUNCHTIME.
BECAUSE YOU'RE A FAST WORKER?
SURE.
Dilbert.com DilbertCartoonist@gmail.com
7-4-14 ©2014 Scott Adams, Inc./Dist. by Universal Uclick

AGENDA/SCHEDULE

40-50% Theory + 60-50% Hands-On

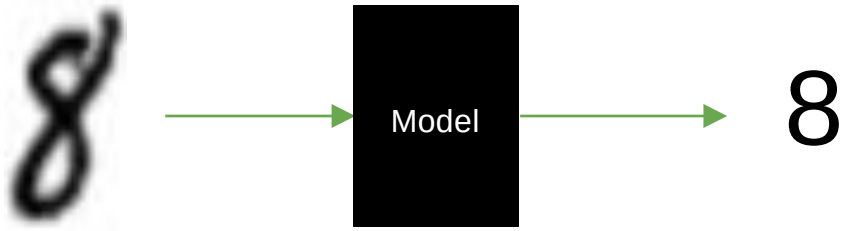
1) Motivation to Machine Learning/Deep Learning
    i. Biological Motivation, Hierarchical/Representation Learning
2) Introduction to Artificial Neural Networks/Deep Learning
    i. Neuron, Perceptron, Logistic, MLP, Rectified Linear Units
    ii. Backpropagation Algorithm, Gradient Descent(including SGD), Mini-batch
3) Word Embedding
    i. CBOW, skip-gram, word2vec
4) Convolution Neural Networks
    i. Convolution
    ii. Sub-sampling, Pooling
    iii. Dropout
    iv. Architecture
5) Recurrent Neural Network
    ii. LSTM
6) Challenges in Deep Learning
    i. Vanishing Gradients & Local Minima
    ii. Overfitting


LEARNING

What is Learning?

How do we recognize the digits?

Machine Learning Framework



Inputs | Computation | Outputs

source: http://www.slideshare.net/indicods/deep-learning-with-python-and-the-theano-library

Recognizing Digit – An algorithm

How do we recognize the digits?

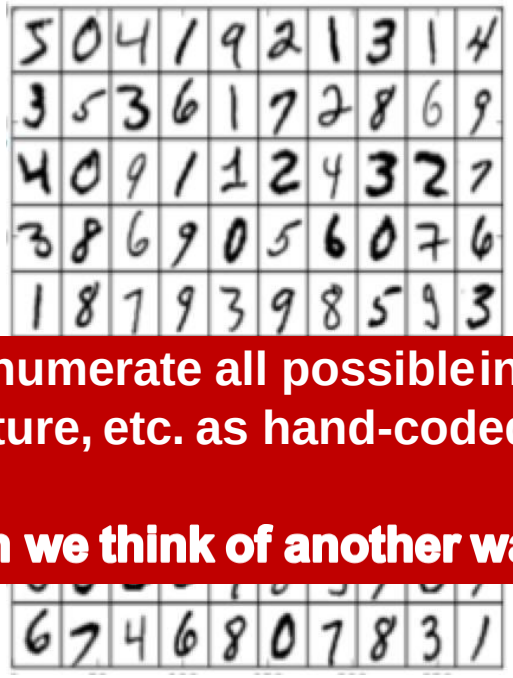
Use functionsthat compute relevant information to solve the problem

k Nearest-Neighbors

For each image, find“most similar” image. Guess that as the label.

Recognizing Digit – An algorithm

How do we recognize the digits?



Difficult to enumerate all possible interactions, spatial structure, etc. as hand-coded features.

Can we think of another way?


LEARNING –
Biological Inspiration

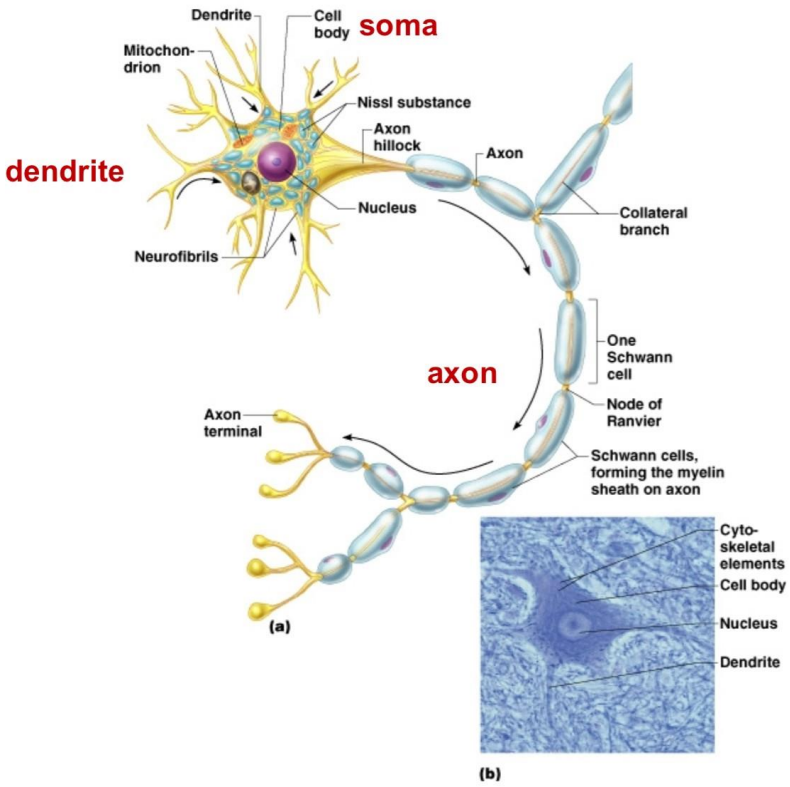
Questions about Learning

How is information detected?

How is it stored?

How does it influence recognition?

Brain



- Connected network of neurons.
- Communicate by electric and chemical signals

~ 10^{11} neurons

~1000 synapses per neuron

- Signals come in via dendrites into soma
- Signal goes out via axon to other neurons through synapses

Learnings from Neuro & Cognitive Science

Kids talk grammatically correct sentences even before they are taught formal langauge.

Kids learn after listening to a lot of sentences

→ Associations and Structural inferences. Understand context. Eg: Drinking water Vs River Vs Ocean

→ See/hear/feel first. Assimilate. Build the context hierarchically. Recognize. Respond.

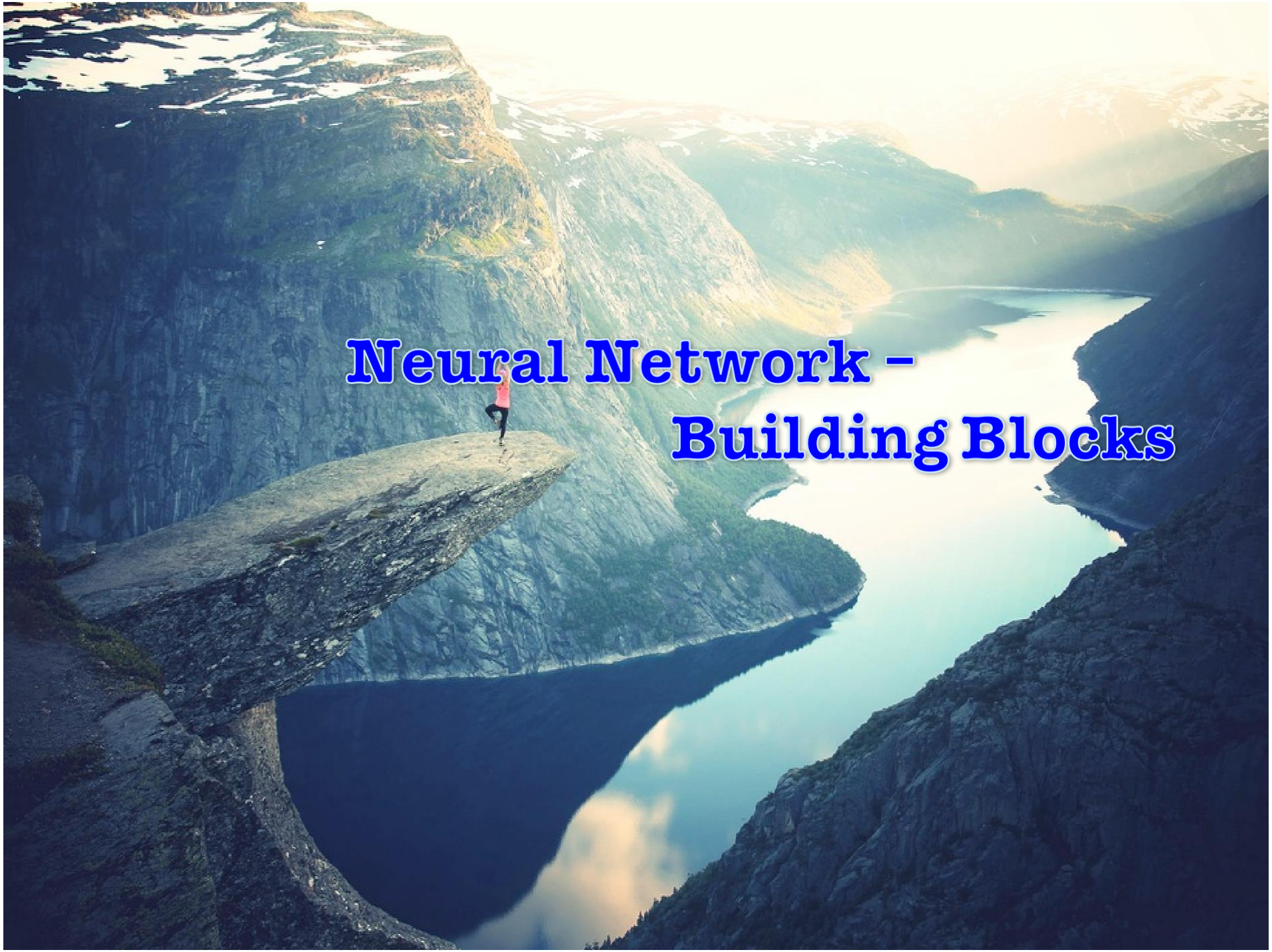
Lessons from Biological Learning

Importance of Connectionism
- Simple units interacting in a complex network
- Distributed representation of knowledge
- Parallelism

Threshold mechanism for robust classification

Mechanism for learning – adjusting synaptic weights

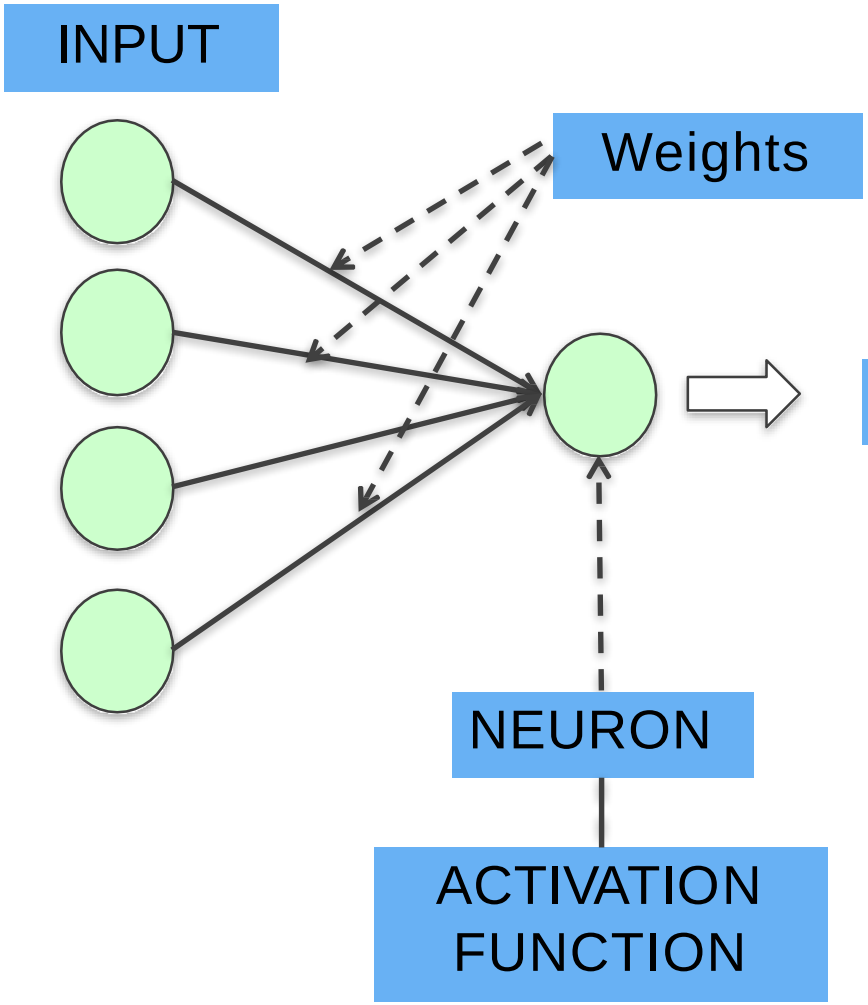
Comprehend inner structure of observed data


Neural Network –
Building Blocks

DID YOU KNOW?

ZEBRAS ARE ACTUALLY BLACK WITH WHITE STRIPES, NOT WHITE WITH BLACK STRIPES.

Neuron, Activation Function

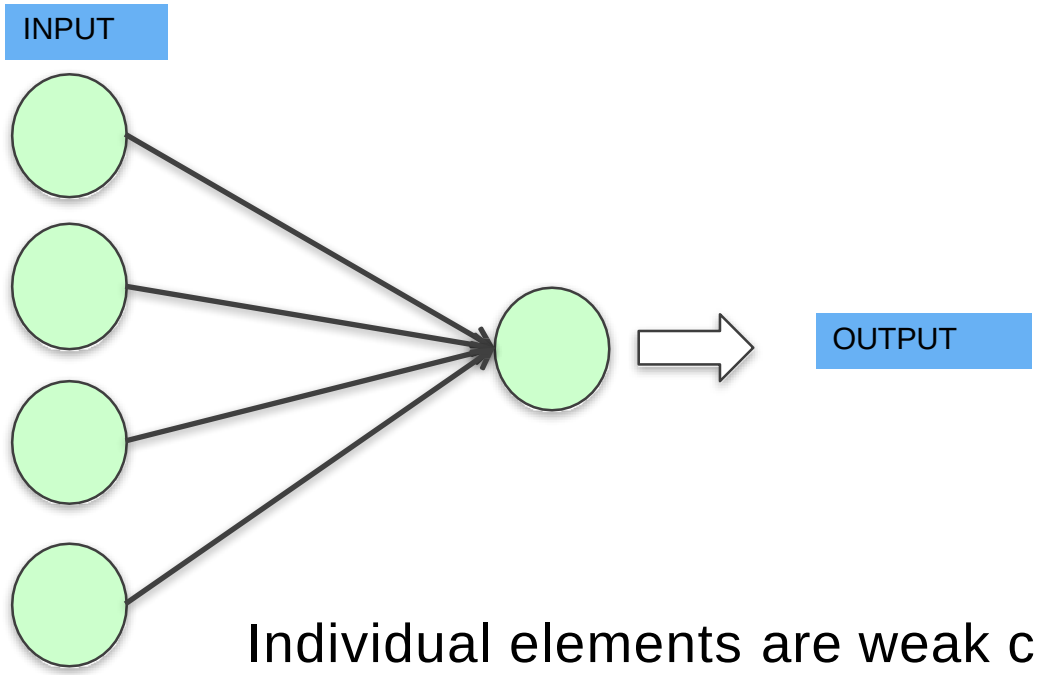


$$\text{Local field} : \sum_{d=0}^{D} w_d x_d$$

$$\text{Activation function} : \varphi(\cdot)$$

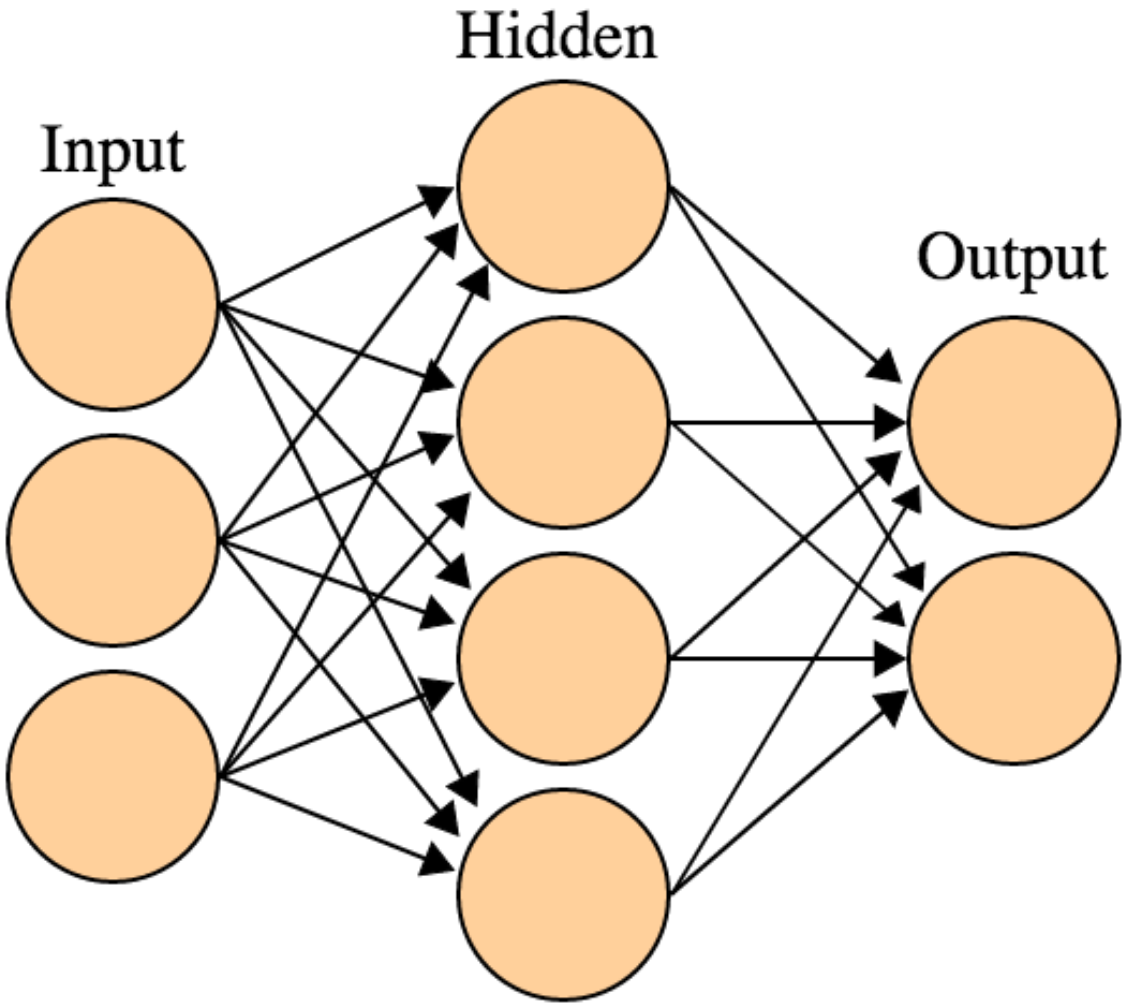
$$\text{Output} : h(\mathbf{x}) = \varphi\left(\sum_{d=0}^{D} w_d x_d \right)$$

Need: Networked Units



Individual elements are weak computational elements

A Simple Neural Network



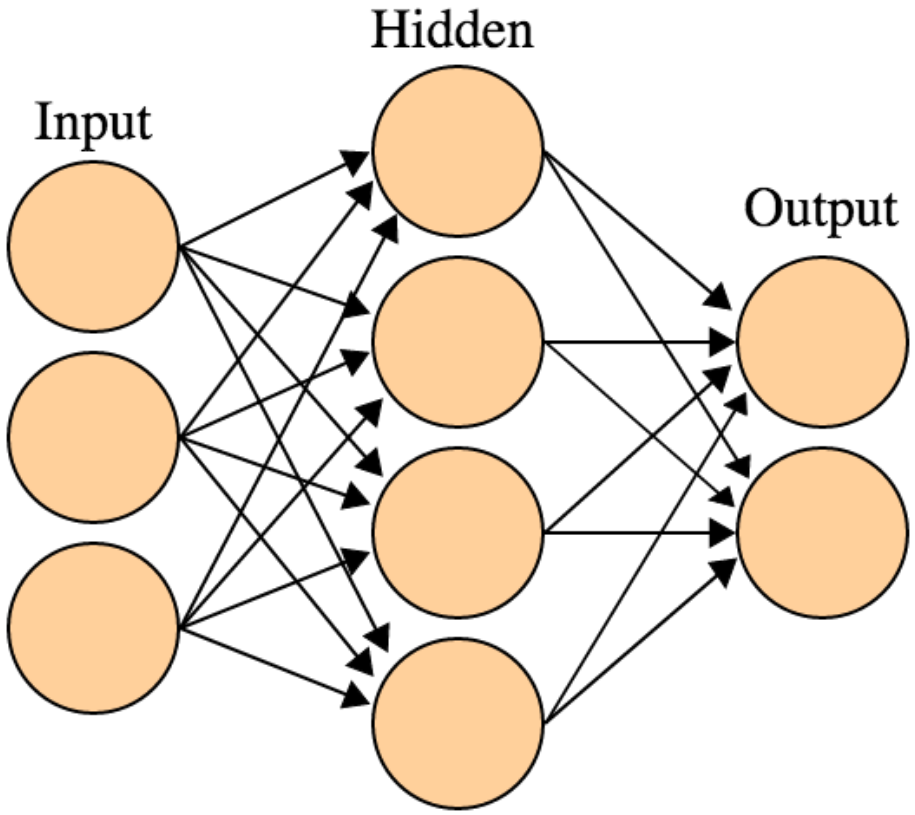
$$\text{Local field} : \sum_{d=0}^{D} w_d x_d$$

$$\text{Activation function} : \varphi(\cdot)$$

$$\text{Output} : h(\mathbf{x}) = \varphi\left(\sum_{d=0}^{D} w_d x_d \right)$$

source: https://en.wikipedia.org/wiki/Artificial_neural_network

Feed-forward Neural Network



» The network is formed by
—Input layer of source nodes
—One or several hidden layers of processing neurons
—Output layer of processing neuron(s)

» Connections only between adjacent layers

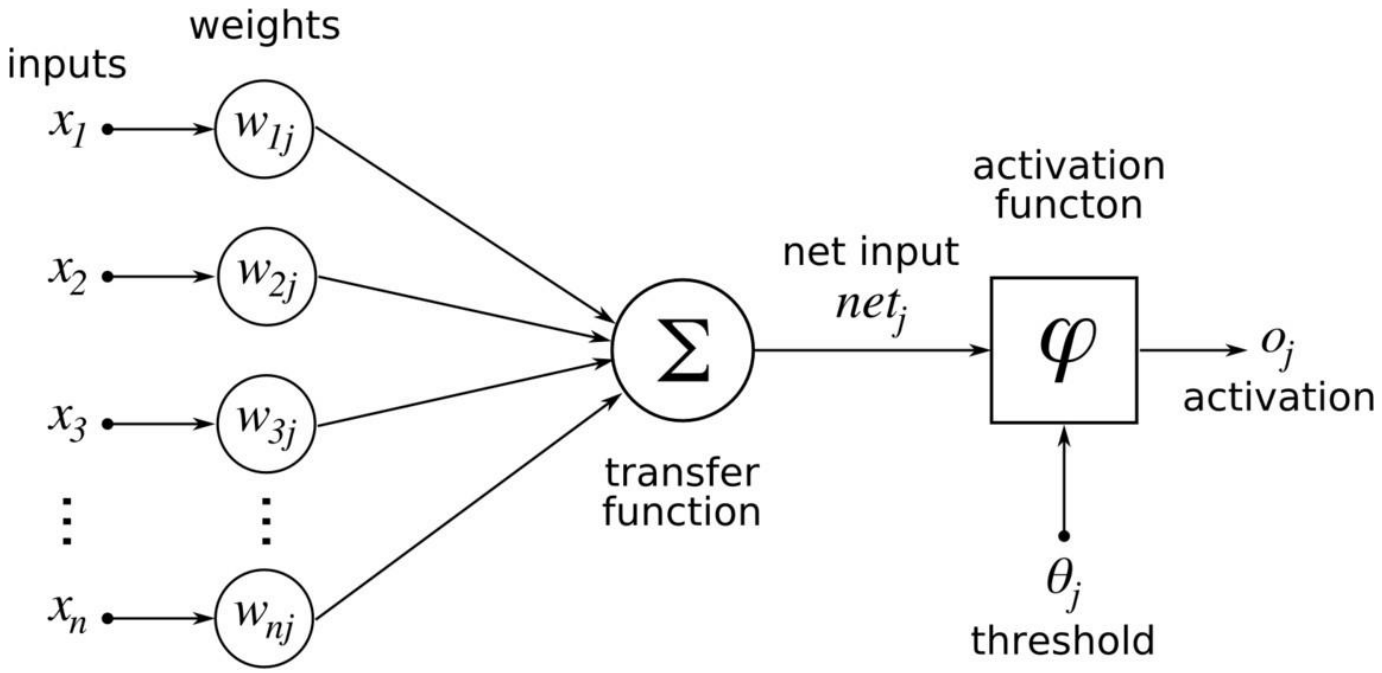
» There are no feedback connections

source: https://en.wikipedia.org/wiki/Artificial_neural_network


Activation Functions

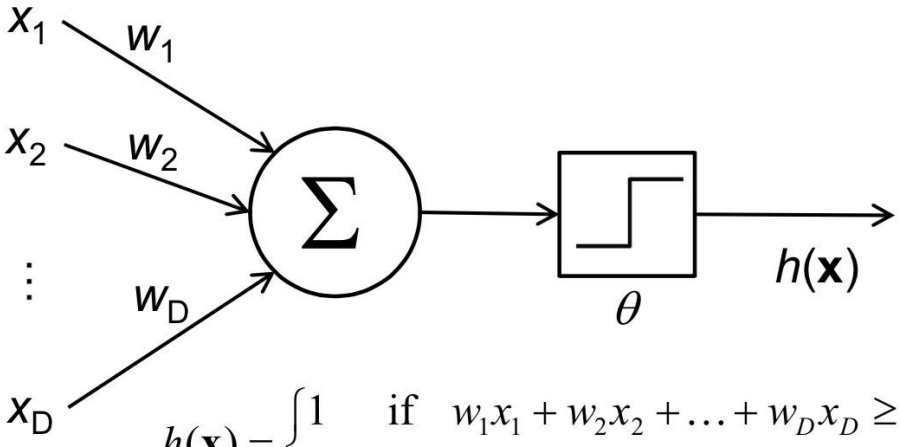
Activation Function

activation function of a node defines the output of that node given input(s)



source: https://en.wikibooks.org/wiki/Artificial_Neural_Networks/Activation_Functions

Perceptron

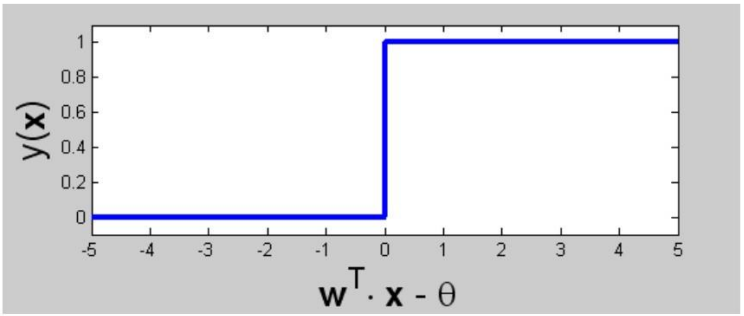


$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } w_1 x_1 + w_2 x_2 + \ldots + w_D x_D \geq \theta \\ 0 & \text{if } w_1 x_1 + w_2 x_2 + \ldots + w_D x_D < \theta \end{cases}$$

$$\text{threshold}: \theta$$

In vector notation:

$$h(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w}^T \cdot \mathbf{x} \geq \theta \\ 0 & \text{if } \mathbf{w}^T \cdot \mathbf{x} < \theta \end{cases}$$



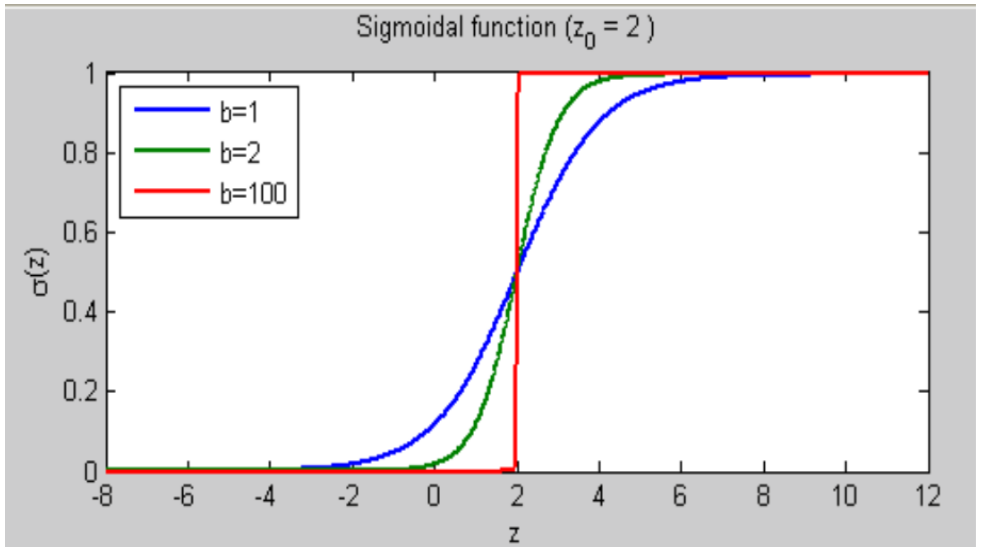
- Invented in 1957.
- Classifies input data into one of the output classes.
- Online learning possible

If the weighted input is more than the threshold, classify as 1. Else 0

McCulloch, W. S., and Pitts, W. "A Logical Calculus of Ideas Immanent in Nervous Activity." Bulletin of mathematical biophysics, 5, pp. 115-133 (1943). Reprinted in McCulloch, W. S., *Embodiments of mind*. Cambridge, MA: MIT Press.

source: ASDM Summer School on Deep Learning 2014

Sigmoid/Logistic



$$\sigma(z) \equiv \frac{1}{1+e^{-z}}; \quad \textbf{[logistic sigmoid]}$$

Local field : $\mathbf{w}^T \cdot \mathbf{x} = \sum_{d=0}^{D} w_d x_d$

Activation function : $\sigma(z) = \frac{1}{1+e^{-z}}$

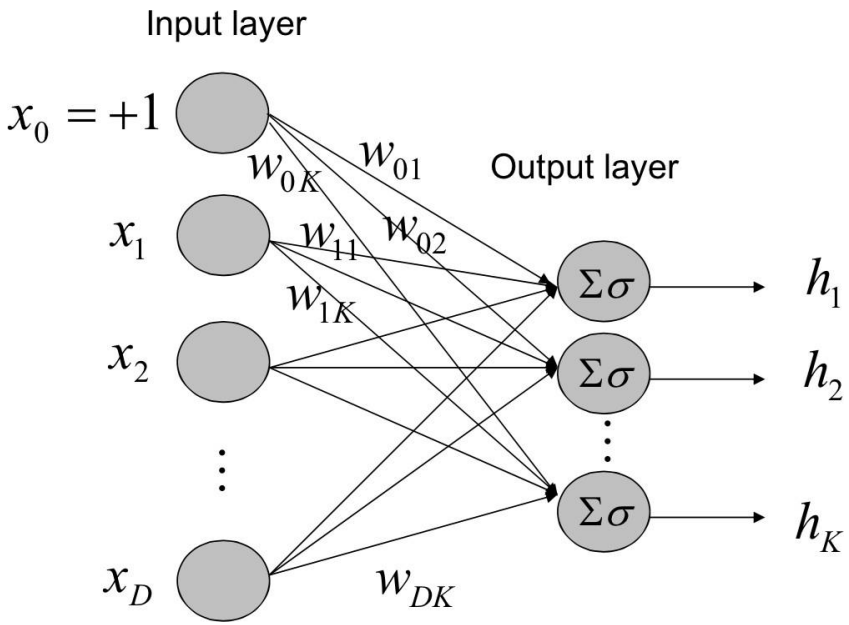
Output : $h(\mathbf{x}) = \sigma\left(\mathbf{w}^T \cdot \mathbf{x}\right)$

- Output is bounded between 0 & 1
- Domain: Complete set of Real numbers
- Smooth and continuous function.

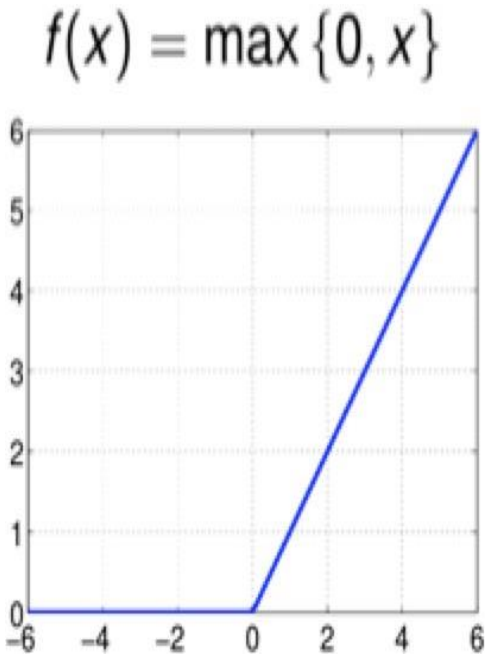
- Symmetric
- Derivative can be quickly calculated
- Positive, Bounded, strictly positive

Softmax

Generalization of logistic regression for Multi-class classification

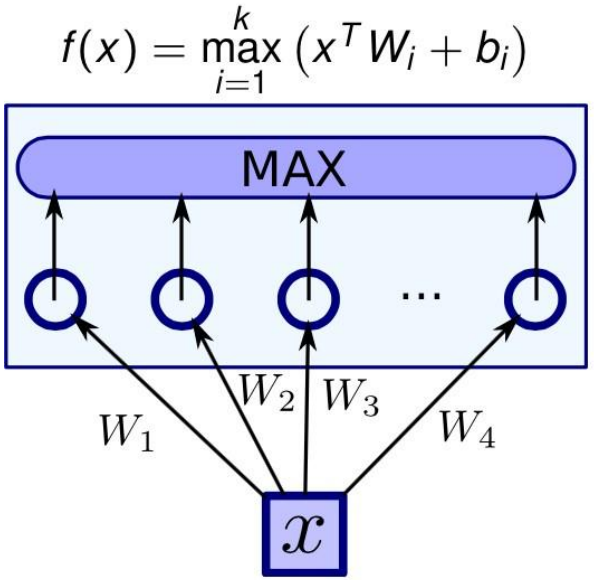


Rectified Linear Units

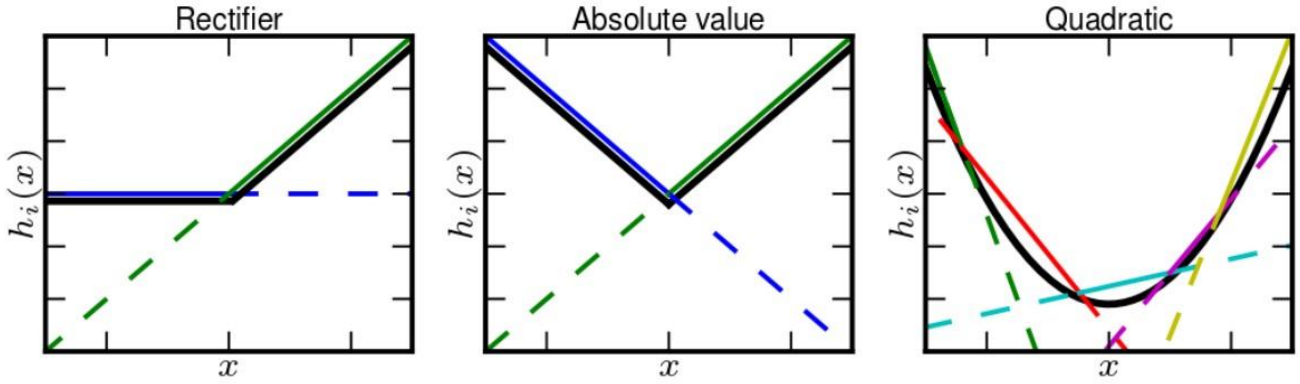


- Cheap to compute (no products/exponentials)
- Faster training
- Sparser networks
- Bounded below 0
- Strictly increasing

Max-out



- Generalization of Rectified Linear
- Max of k linear functions -> piece-wise linear
- At large k, can approximate a non-linear function

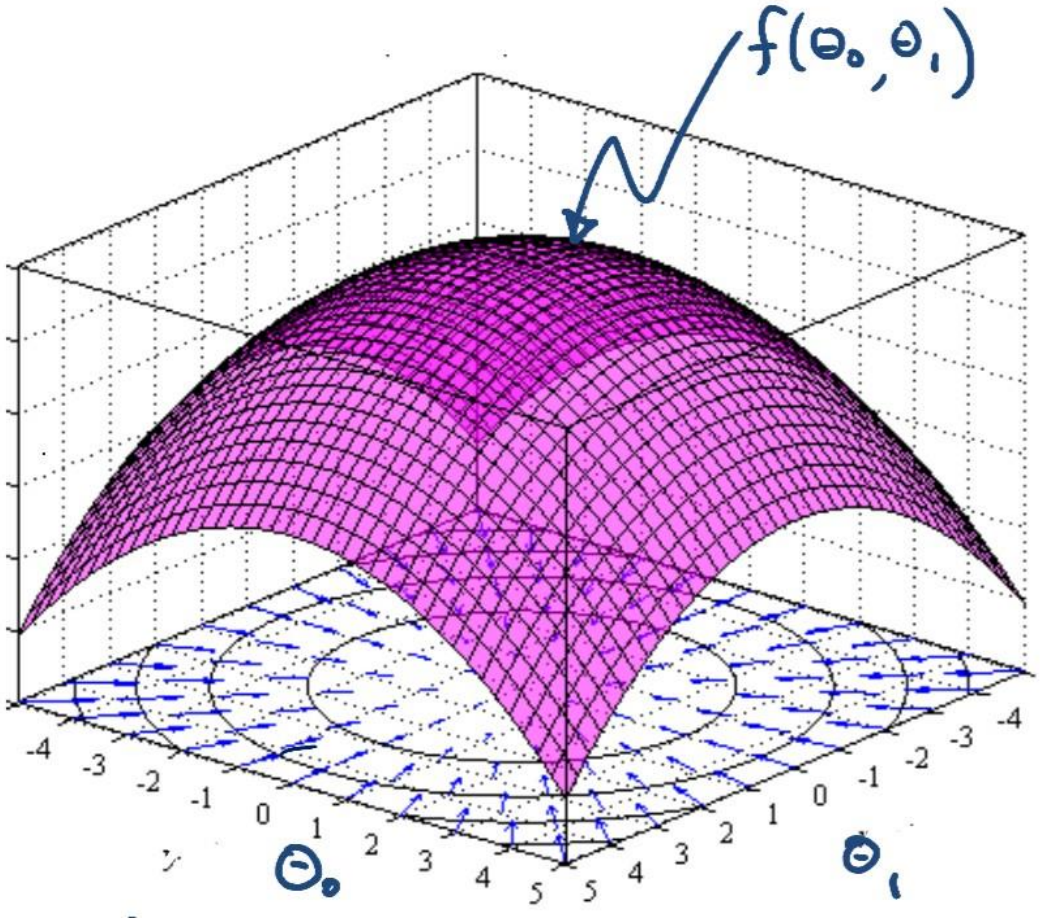


source: ASDM Summer School on Deep Learning 2014


Learning in ANN

Learning in ANN – Gradient Descent

Goal: To find minimum of the loss function (minimize error of the model)

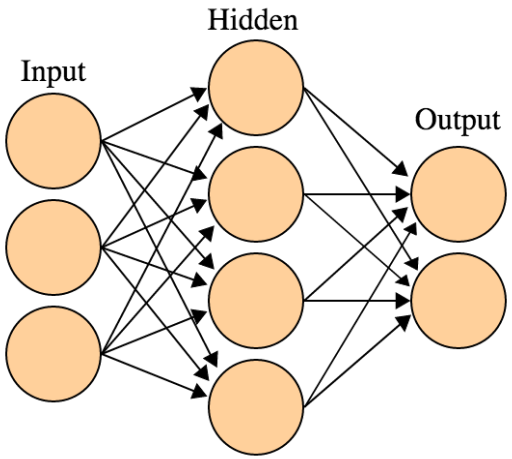


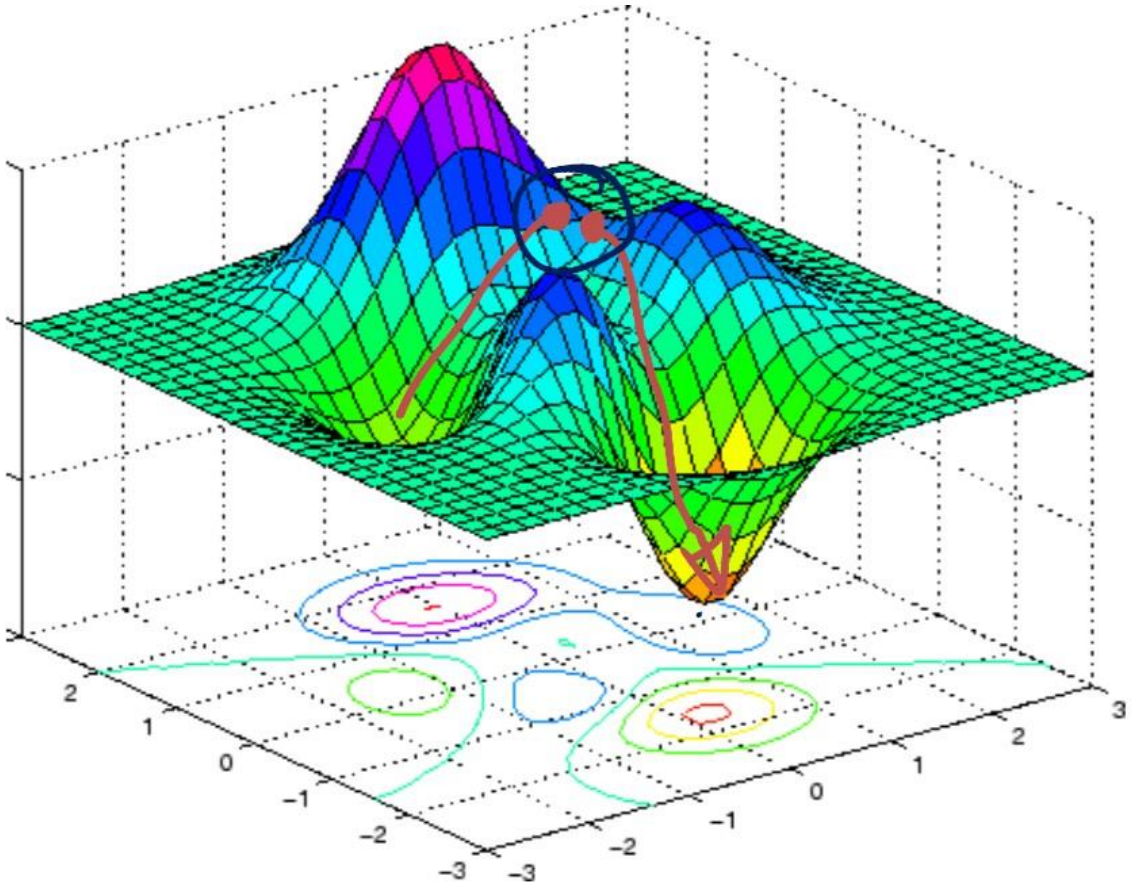
Repeat until convergence {

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

}

Er.... How to compute gradient?





Backpropagati…thm – Pseduo Code

Algorithm 1 Backprop…

for d in data **do**

FORWARDS PASS

Starting from the …rward pass trough the network, comput… at each layer.

BACKWARDS PASS

Compute the deriva… th respect to the output layer activities

for layer in layers **do**

Compute the derivat… th respect to the inputs of the upper layer n…

Compute the deriv… on with respect to the weights between the … below

Compute the derivati… on with respect to the activities of the layer be…

end for

Updates the weights.

end for

source: http://tex.stackexchange.com/questions/155614/description-of…

Backpropagation Algorithm – Pseduo Code

Algorithm 1 Backpropagation learning algorithm

for d in data **do**

FORWARDS PASS

Starting from the input layer, use eq. 1 to do a forward pass trough the network, computing the activities of the neurons at each layer.

BACKWARDS PASS

Compute the derivatives of the error function with respect to the output layer activities

for layer in layers **do**

Compute the derivatives of the error function with respect to the inputs of the upper layer neurons

Compute the derivatives of the error function with respect to the weights between the outer layer and the layer below

Compute the derivatives of the error function with respect to the activities of the layer below

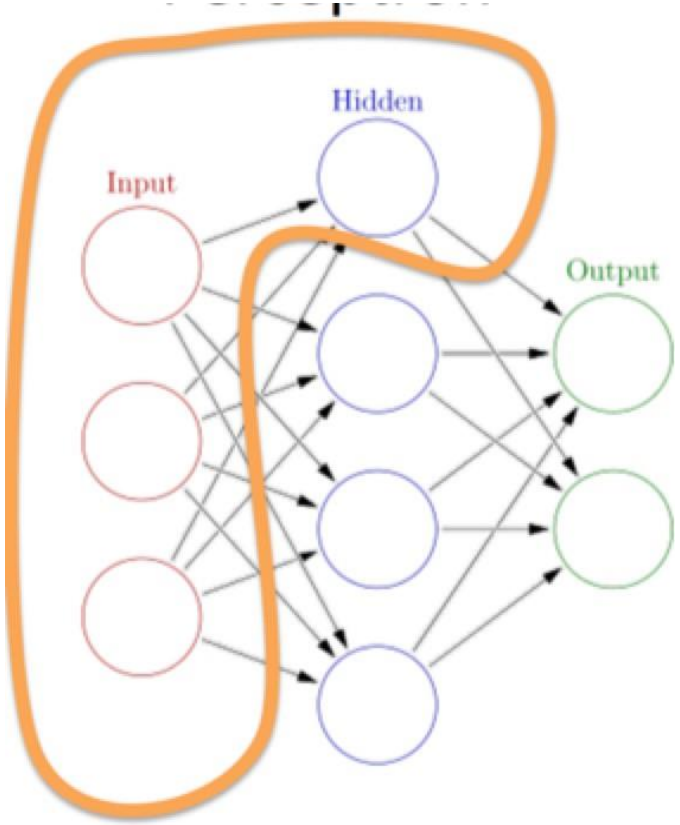
end for

Updates the weights.

end for

source: http://tex.stackexchange.com/questions/155614/description-of-algorithm-in-words

Backpropagation Algorithm



Computes gradient of the loss function w.r. to the weights.

Backpropagate training error to generate deltas of all the neurons from hidden layers to output layer

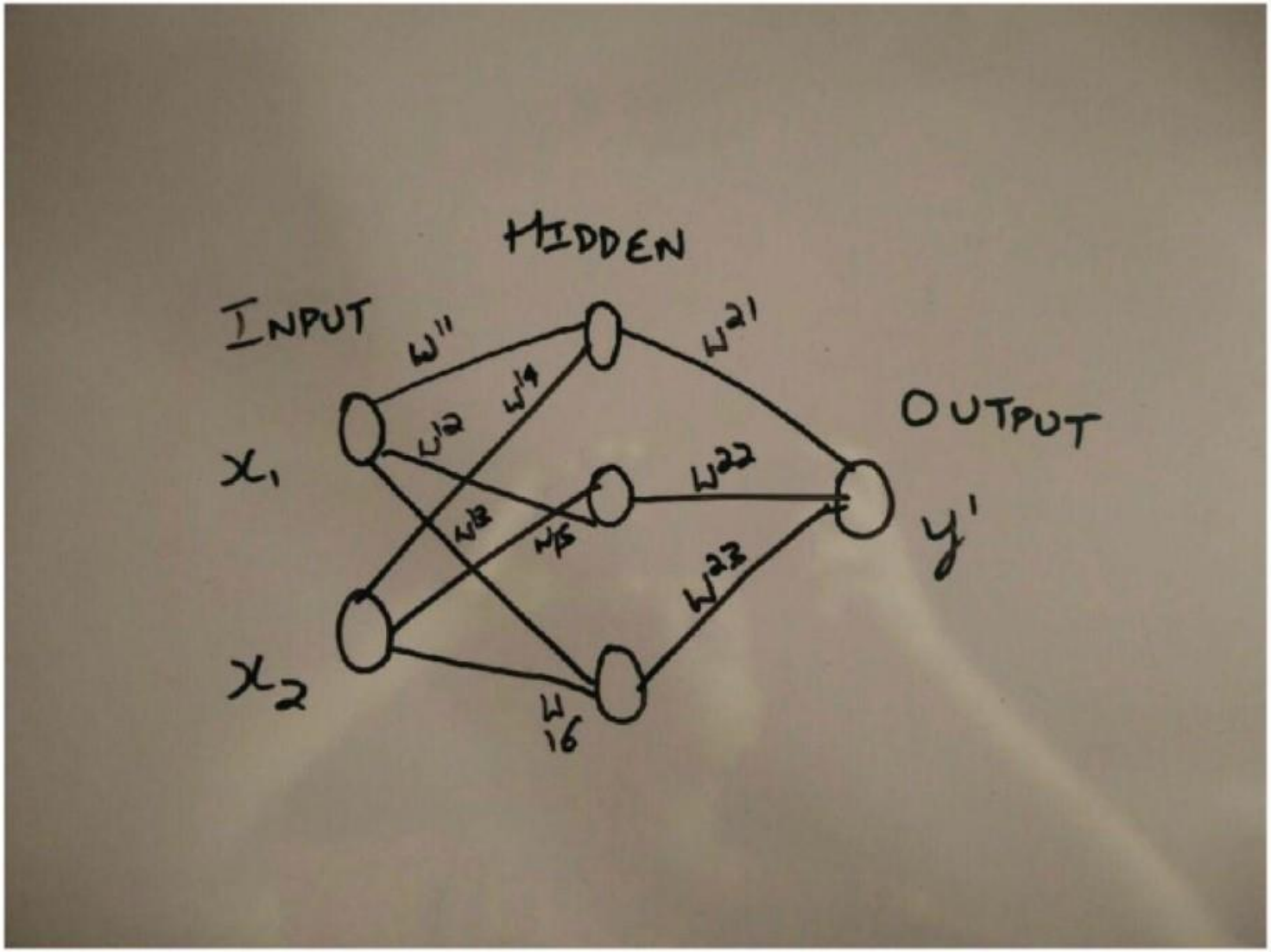
Use gradient descent to update weights

References:

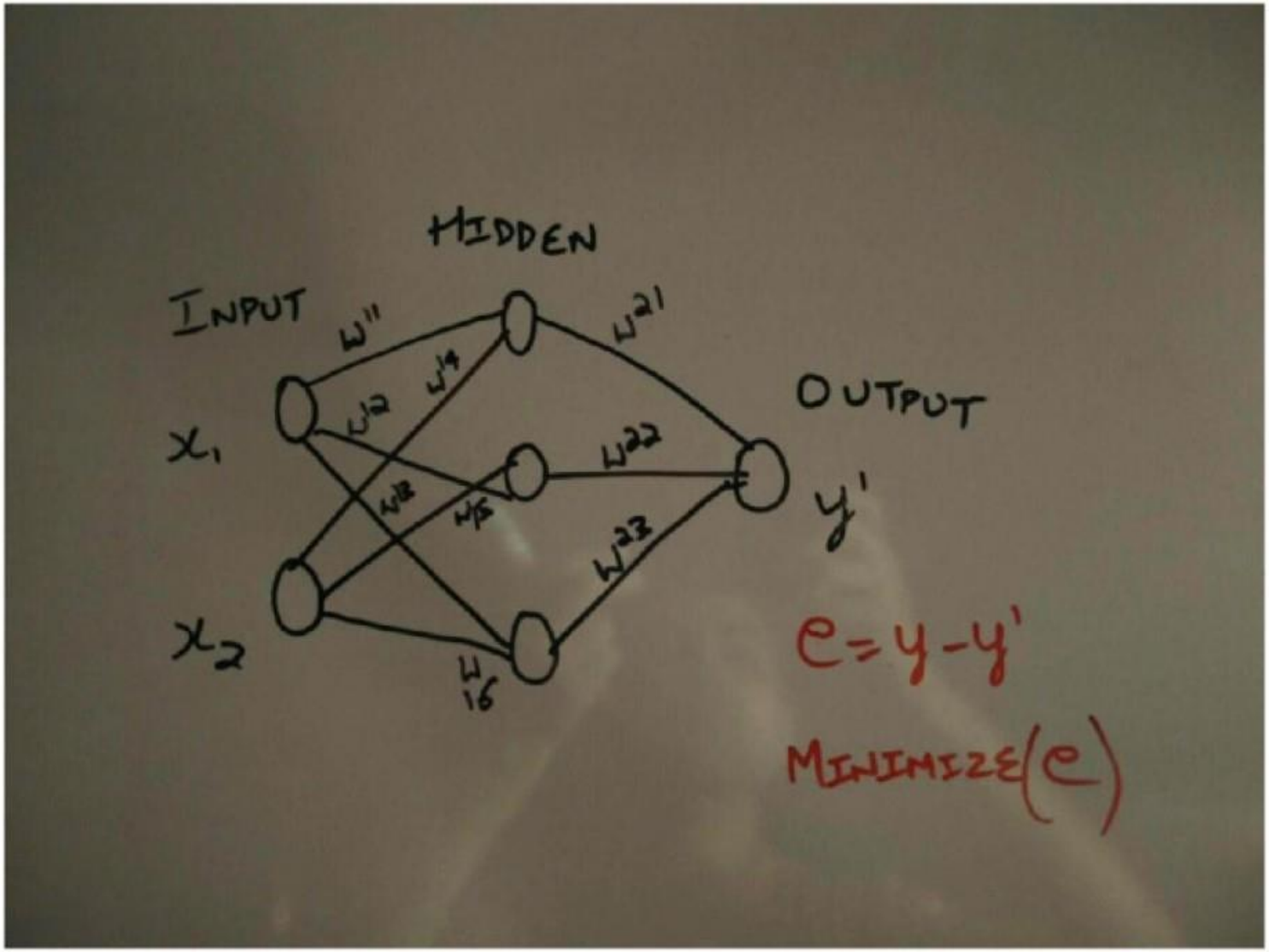
colah.github.io/posts/2015-08-Backprop/

http://neuralnetworksanddeeplearning.com/chap2.html

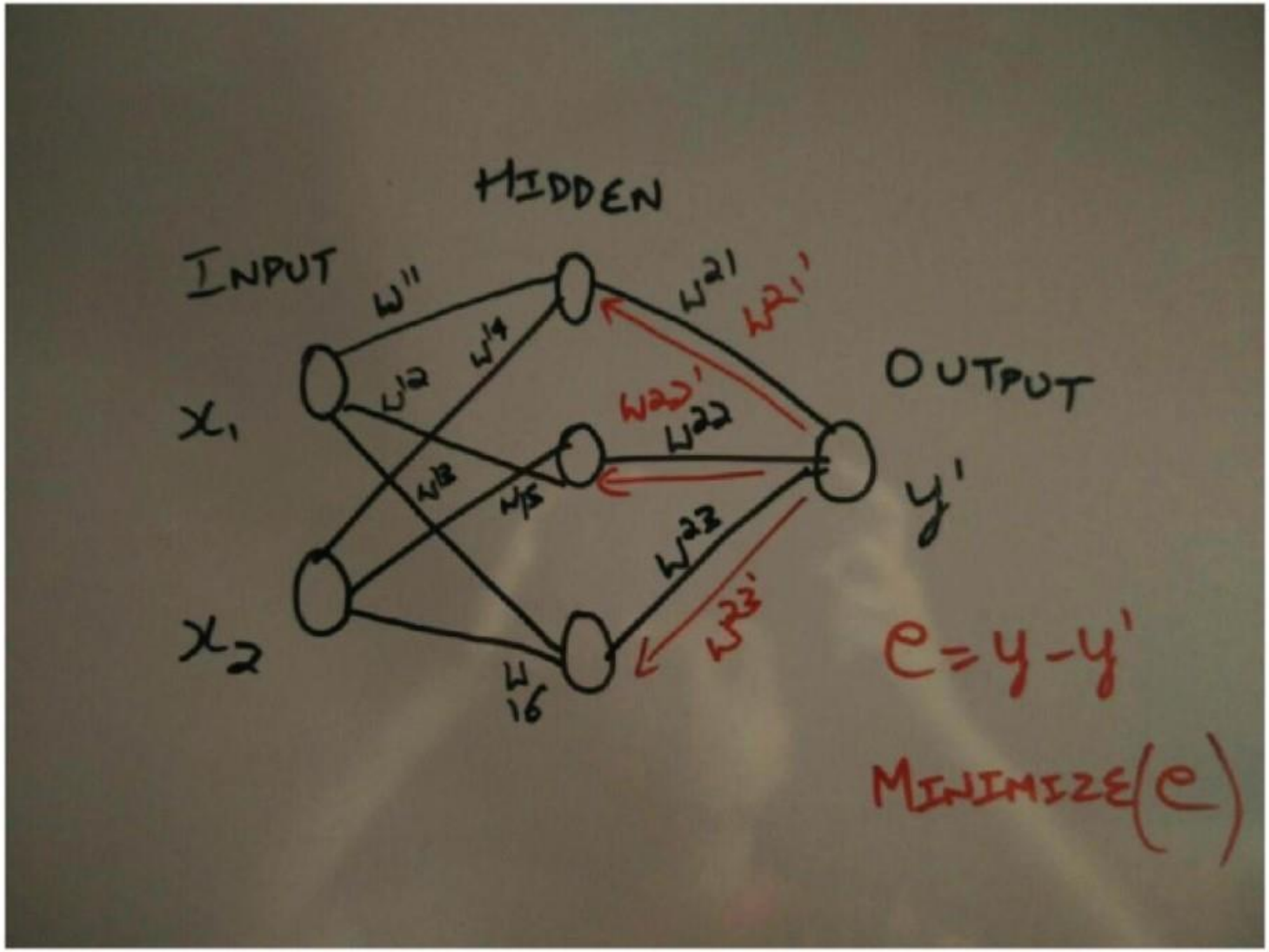
Backpropagation Algorithm



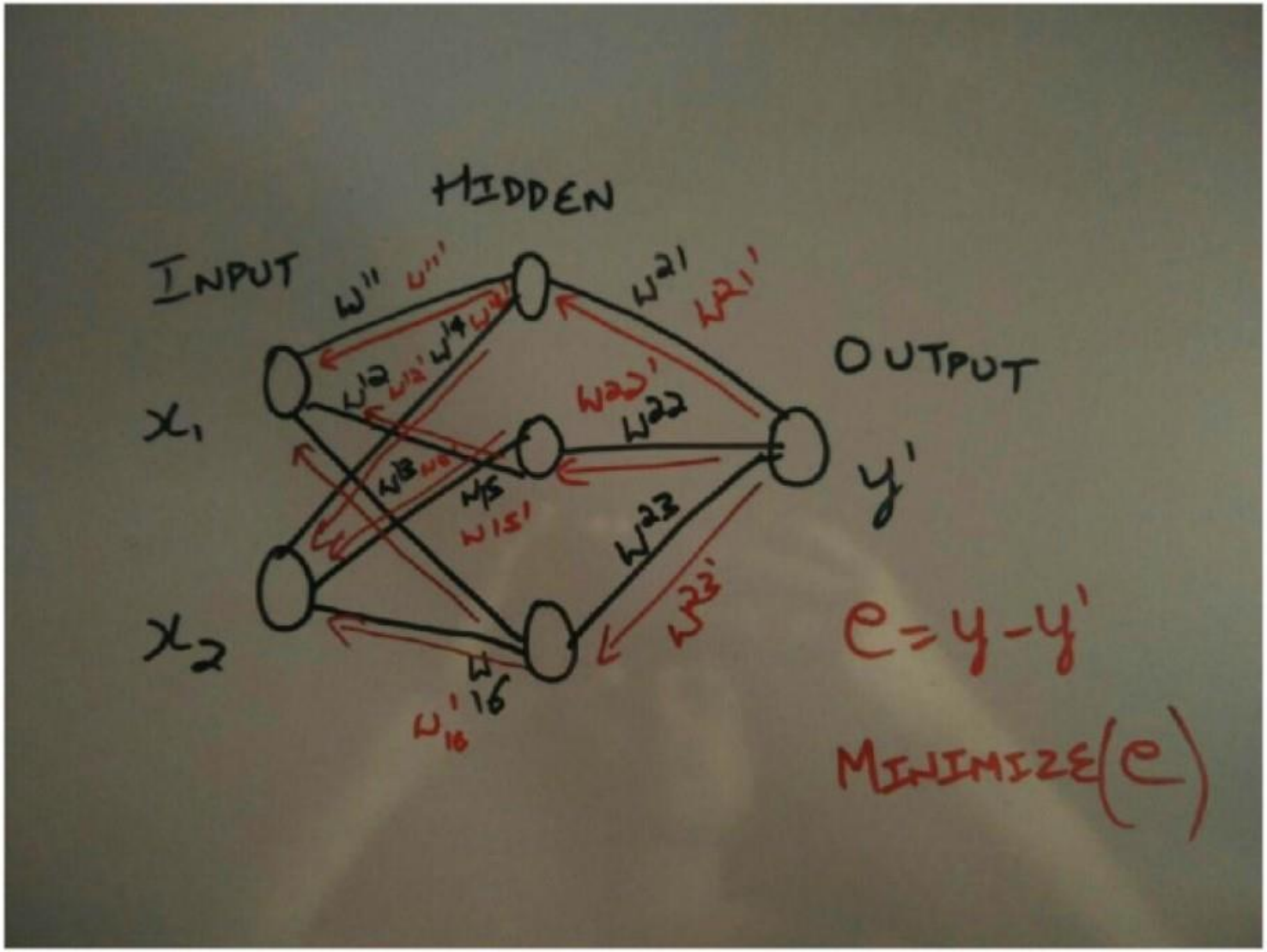
Backpropagation Algorithm



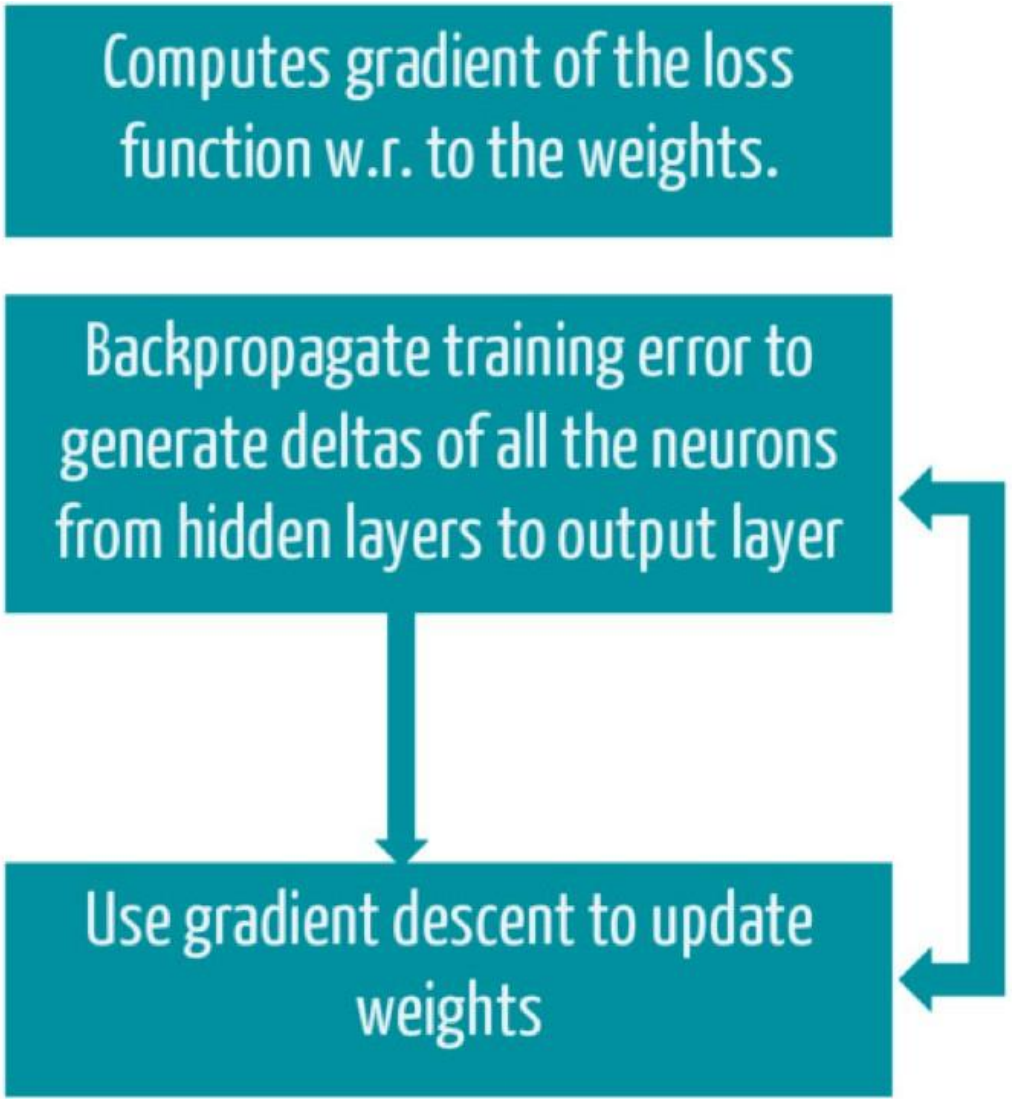
Backpropagation Algorithm



Backpropagation Algorithm



Backpropagation Algorithm



SGD/Mini-Batch/Online

- Stochastic Gradient Descent: Instead of using all of the training data, train iteratively on "mini-batches"
- Online Learning: Mini-batch size is 1. Weights are adjusted for every single data point.
- There are other variants of the loss function – with better empirical results than SGD (Not covered in the slides)

SGD/Mini-Batch/Online

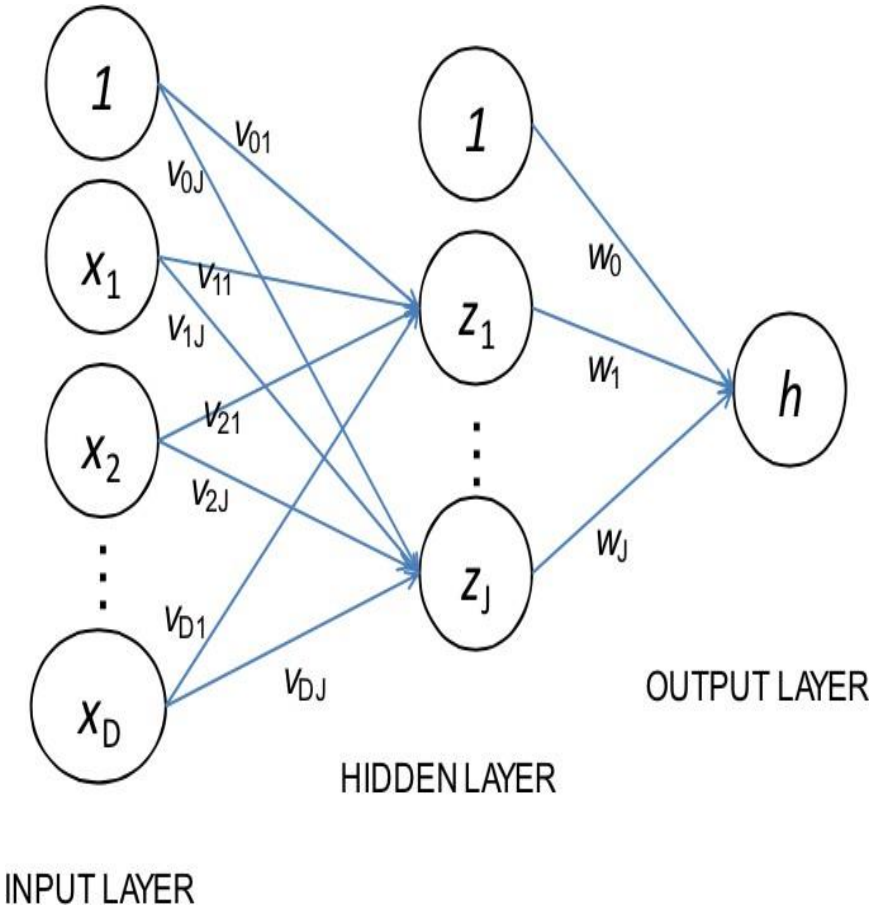
Algorithm 2 Minibatch Stochastic Gradient Descent Training

Input: Function $f(\mathbf{x};\theta)$ parameterized with parameters θ.
Input: Training set of inputs $\mathbf{x_1},\ldots,\mathbf{x_n}$ and outputs $\mathbf{y_1},\ldots,\mathbf{y_n}$.
Input: Loss function L.
while stopping criteria not met **do**
    Sample a minibatch of m examples $\{(\mathbf{x_1},\mathbf{y_1}),\ldots,(\mathbf{x_m},\mathbf{y_m})\}$
    $\hat{\mathbf{g}} \leftarrow 0$
    for $i = 1$ to m **do**
        Compute the loss $L(f(\mathbf{x_i};\theta),\mathbf{y_i})$
        $\hat{\mathbf{g}} \leftarrow \hat{\mathbf{g}} +$ gradients of $\frac{1}{m}L(f(\mathbf{x_i};\theta),\mathbf{y_i})$ w.r.t θ
    $\theta \leftarrow \theta + \eta_k\hat{\mathbf{g}}$
return θ

source: http://u.cs.biu.ac.il/~yogo/nnlp.pdf


Our First Architecture

Multi-Layer Perceptron



- Feedforward ANN
- Activation function: Mostly sigmoid
- Improvement over basic perceptron: Can classify data that aren't linearly separable

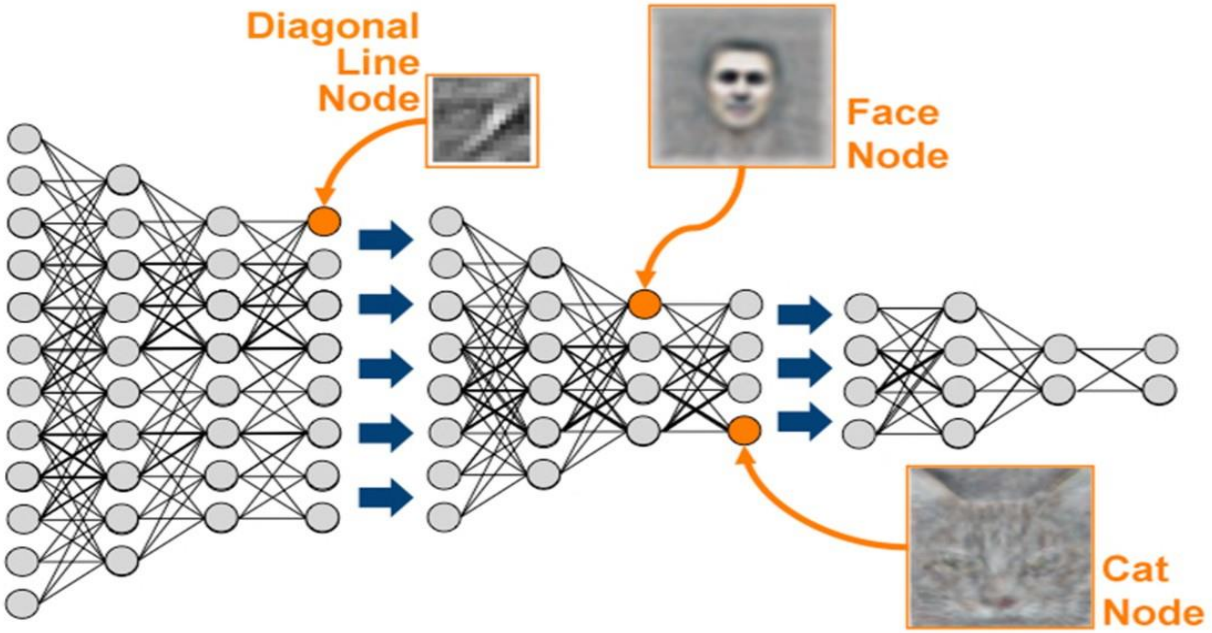

SOME DEEP LEARNING EXAMPLES

Machine Translation

source:http://blog.webcertain.com/machine-translation-technology-the-search-engine-takeover/18/02/2015/

Video Classification

Google's neural net learns just by watching youtube videos



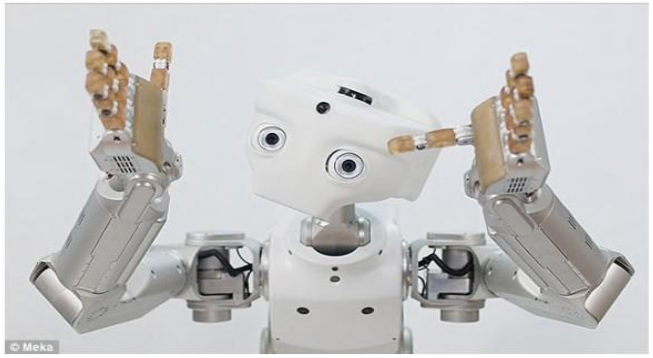


Image Recognition

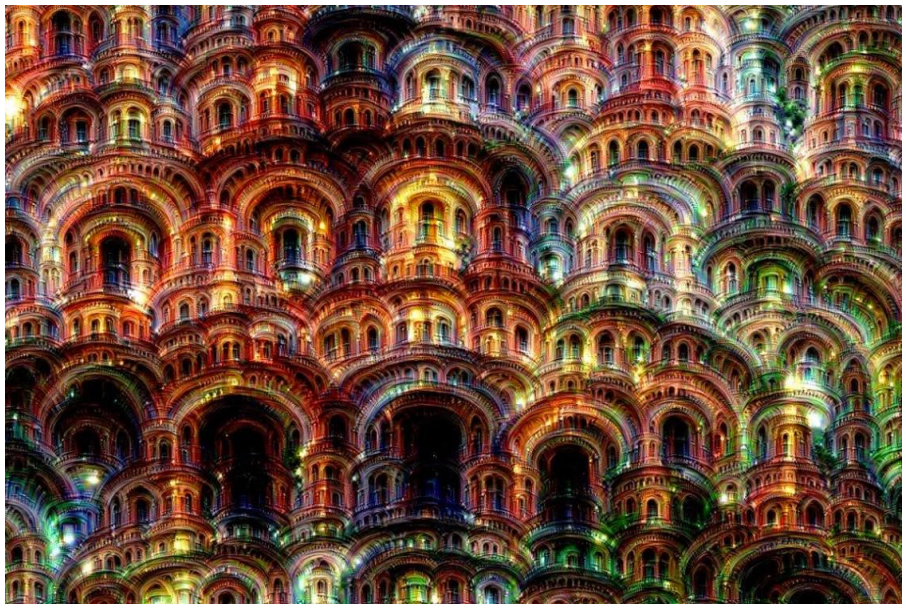
source: http://www.cs.ubc.ca/~nando/

Automatic Generation of Cooking Recipe

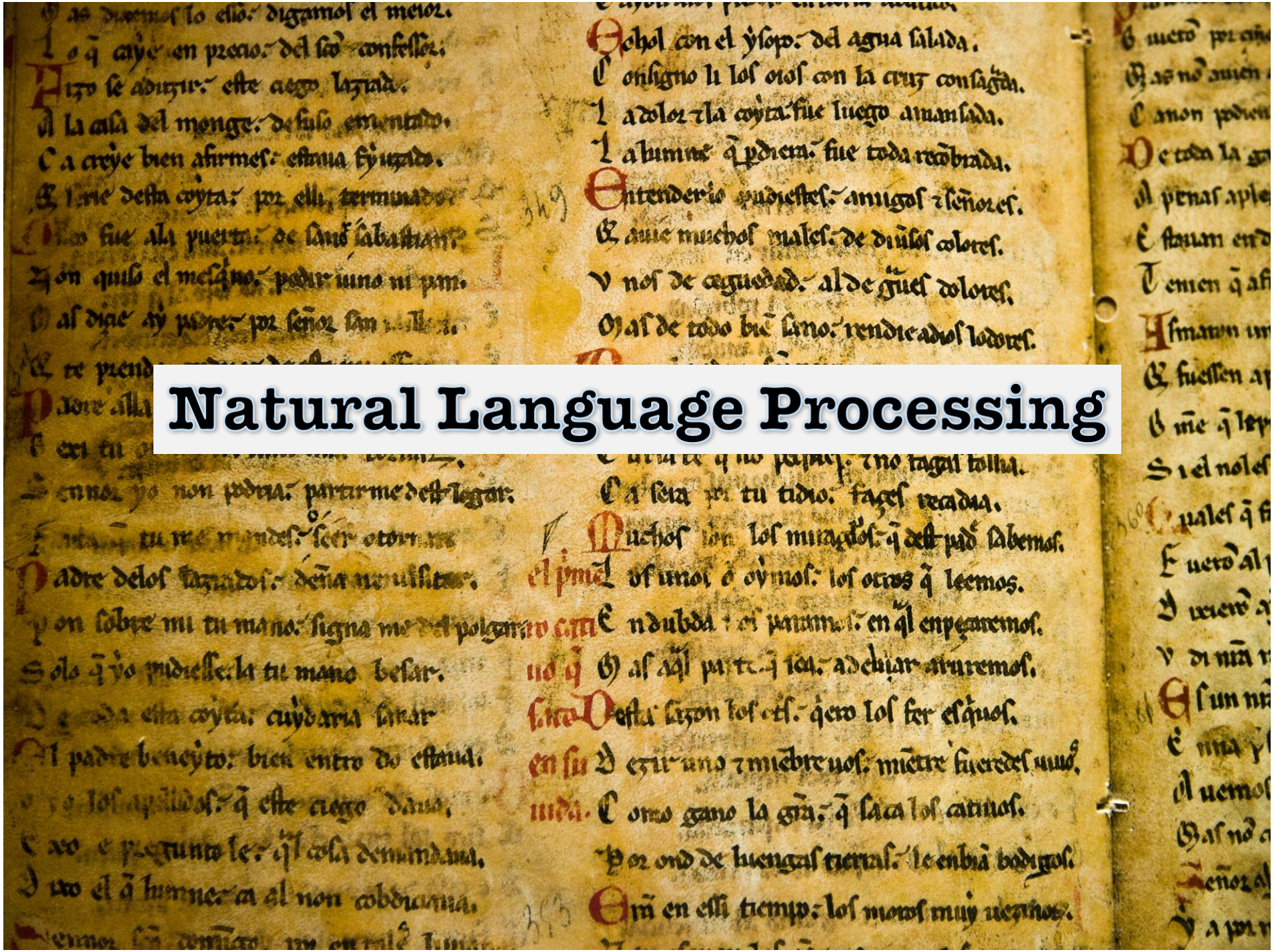
```
MMMMM

MMMMM----- Recipe via Meal-Master (tm) v8.05

      Title: BARBECUE RIBS
 Categories: Chinese, Appetizers
      Yield: 4 Servings

      1 pk Seasoned rice
      1    Beer -- cut into
           -cubes
      1 ts Sugar
    3/4 c  Water
           Chopped finels,
           -up to 4 tblsp of chopped
      2 pk Yeast Bread/over

MMMMM--------------------------FILLING-----------------------------
      2 c  Pineapple, chopped
    1/3 c  Milk
    1/2 c  Pecans
           Cream of each
      2 tb Balsamic cocoa
      2 tb Flour
      2 ts Lemon juice
           Granulated sugar
      2 tb Orange juice
      1 c  Sherry wheated curdup
      1    Onion; sliced
      1 ts Salt
      2 c  Sugar
    1/4 ts Salt
    1/2 ts White pepper, freshly ground
           Sesame seeds
      1 c  Sugar
    1/4 c  Shredded coconut
    1/4 ts Cumin seeds

  Preheat oven to 350.  In a medium bowl, combine milk, the sugar, vanilla and seasoned
  flour and water and then cornstarch. add tomatoes, oregano, and
  nutmeg; serve.
```

Source: https://gist.github.com/nylki/1efbaa36635956d35bcc

Image Generation – Google Inceptionism

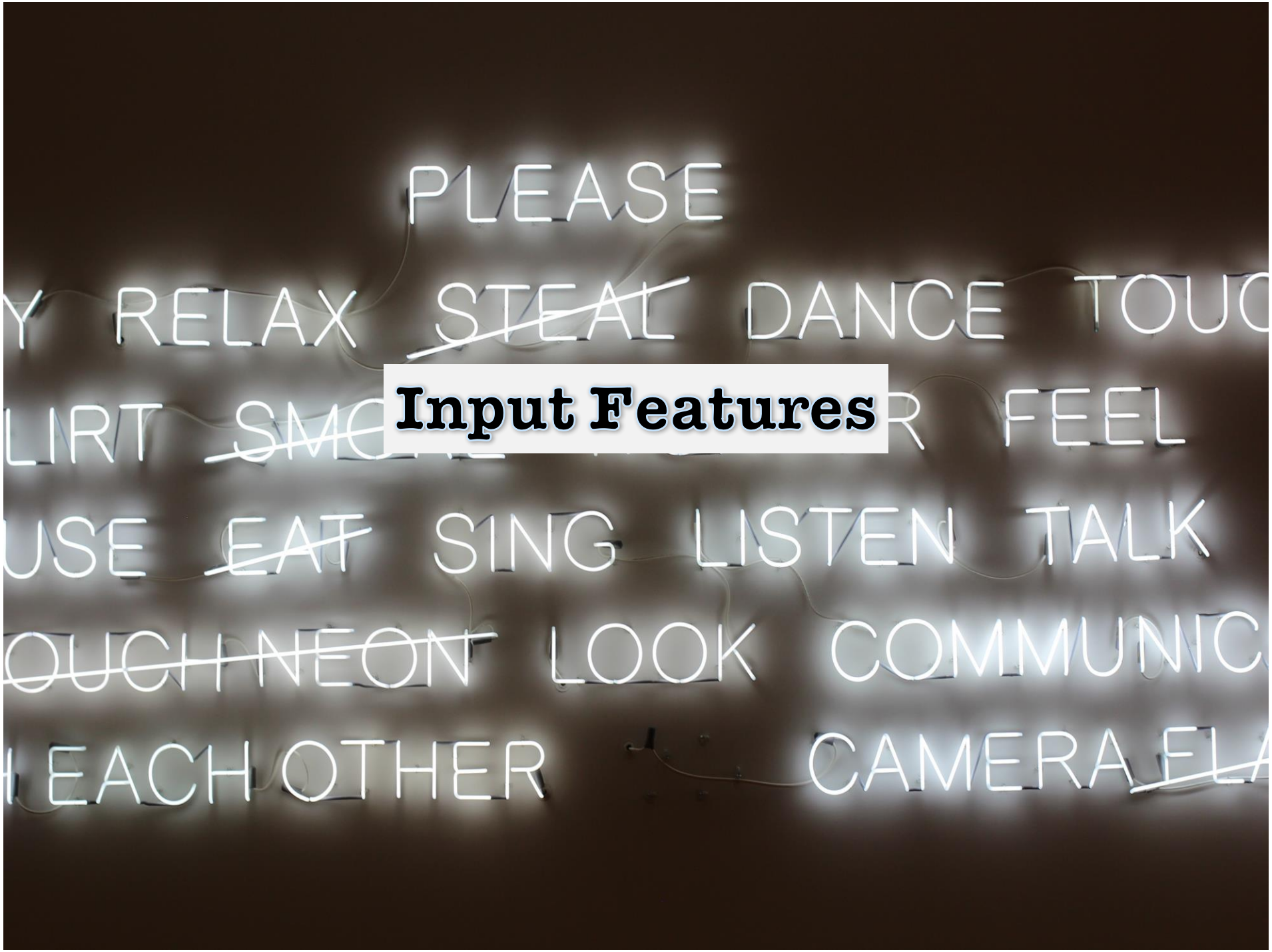


source: http://googleresearch.blogspot.in/2015/06/inceptionism-going-deeper-into-neural.html


Natural Language Processing

Some NLP Tasks

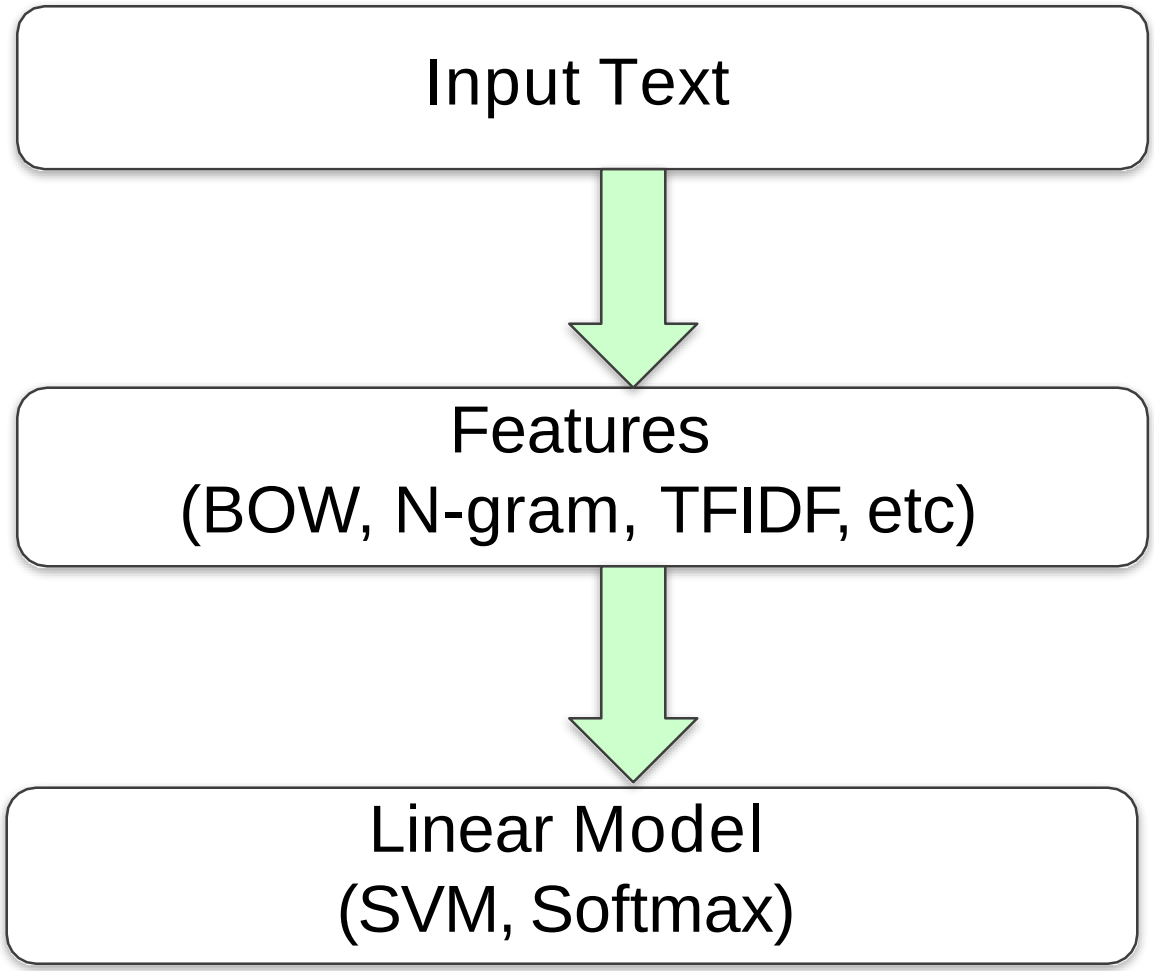
- Sentiment Analysis
- Machine Translation
- Document Classification
- Language Modeling
- Event Detection
- Question-Answering
- Chat Bots (Response Generation)


PLEASE
Y RELAX STEAL DANCE TOUC
LIRT SMOKE R FEEL
USE EAT SING LISTEN TALK
OUCH NEON LOOK COMMUNIC
EACH OTHER CAMERA FLA
Input Features

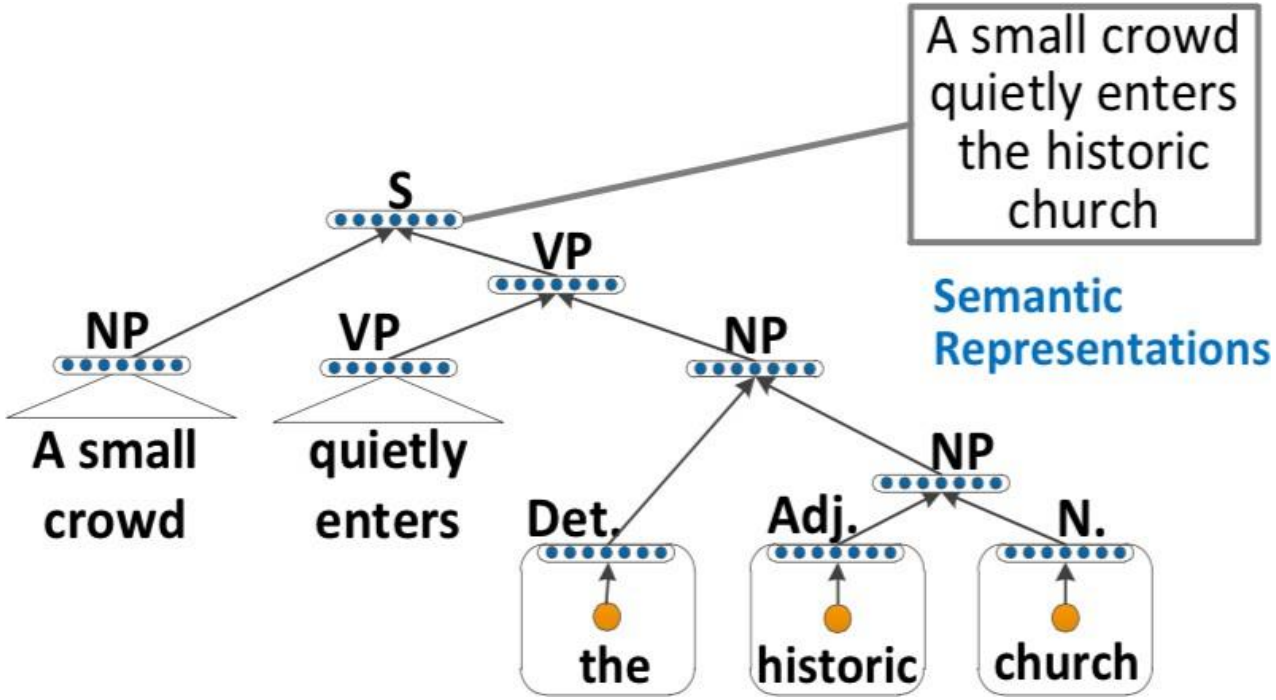
How to create input features?

- ❑ Bag of Words
- ❑ N-gram
- ❑ TF-IDF

Existing ML framework for NLP



A Sentence



source: http://web.stanford.edu/class/cs224n/handouts/CS224N_DeepNLP_Week7_lecture2.pdf

Structure is very important

A small crowd quietly enters the historic church

A Historic crowd enters the small quietly church

Structure important for:

- ❑ Humor
- ❑ Sarcasm

Representing structure is hard

n-grams

– quickly explodes

Limitation of the architectures so far

Fixed Size Input
(Eg: Image)

Fixed computational steps
(Eg: Number of layers)

Fixed Size Ouput
(Eg: probabilities of different classes)

Limitation of the architectures so far

Hierarchy captured. But context, structure ?

– mostly NOT !

NLP –Deep Learning Classification System

The general structure for an NLP classification system based on a feed-forward neural network is thus:

1. Extract a set of core linguistic features $f_1, \ldots, f_k$ that are relevant for predicting the output class.

2. For each feature f_i of interest, retrieve the corresponding vector $v(f_i)$.

3. Combine the vectors (either by concatenation, summation or a combination of both) into an input vector $\mathbf{x}$.

4. Feed $\mathbf{x}$ into a non-linear classifier (feed-forward neural network).

source: http://u.cs.biu.ac.il/~yogo/nnlp.pdf

Word Embeddings

- Vectorization
- Dense-representations on fixed-size vectors (low dimensional space)

Word Embeddings

Goal

- "learning high-quality word vectors from huge data sets with billions of words, and with millions of words in the vocabulary"
- Resulting word representations
  - Similar words tend to be close to each other
  - Words can have multiple degrees of similarity

http://www.slideshare.net/TraianRebedea/what-is-word2vec

Word Embeddings

"You shall know a word
by the company it keeps"

One of the most successful ideas of modern statistical NLP

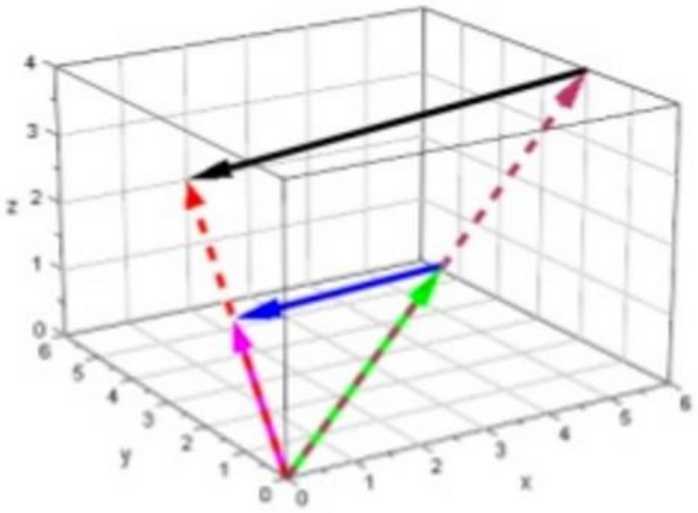
government debt problems turning into	banking	crises as has happened in
saying that Europe needs unified	banking	regulation to replace the hodgepodge

↖ These words will represent *banking* ↗

Quoted after Socher

http://www.slideshare.net/hen_drik/word2vec-from-theory-to-practice

Word Embeddings



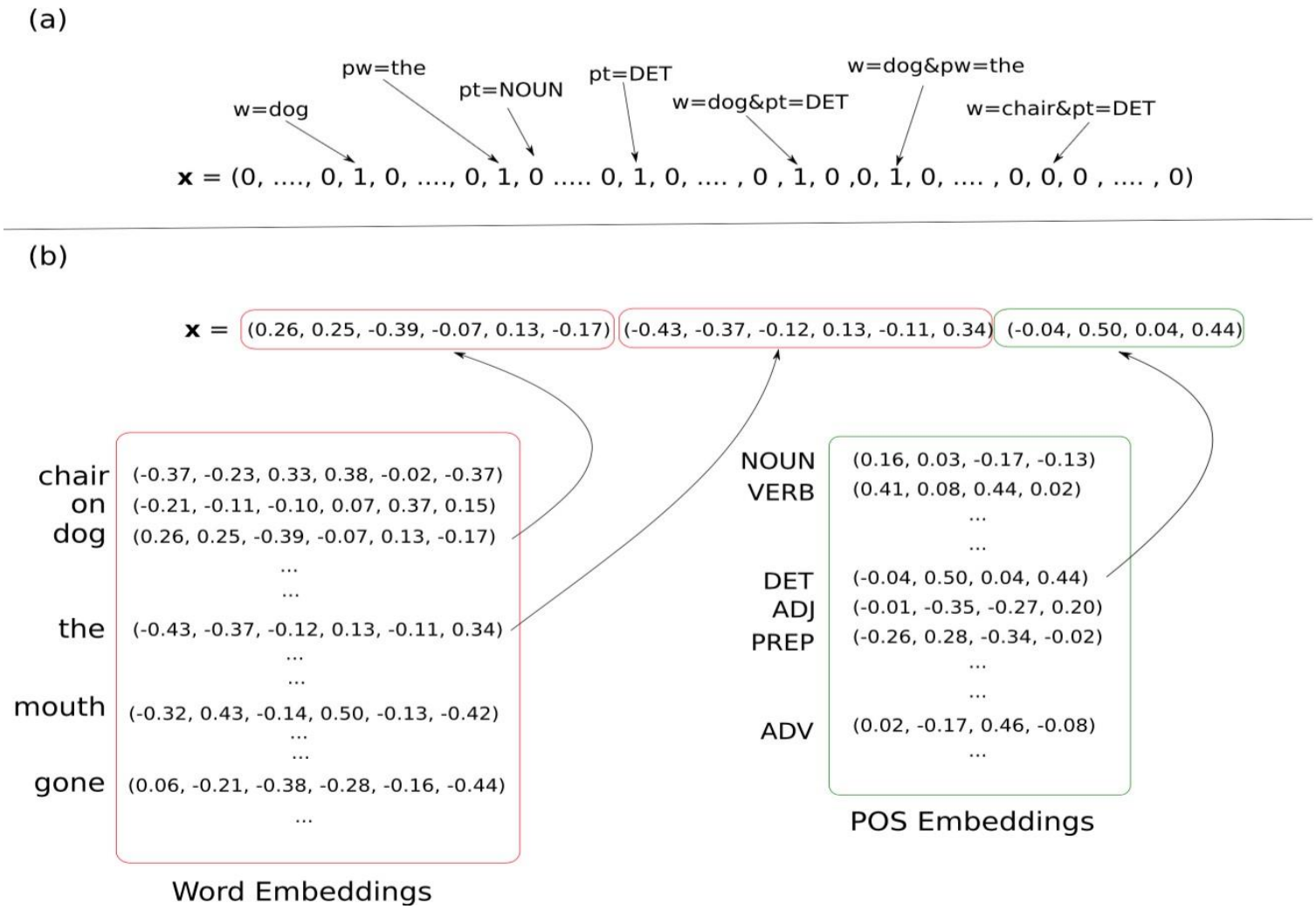
linguistics =

$$\begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

Quoted after Socher

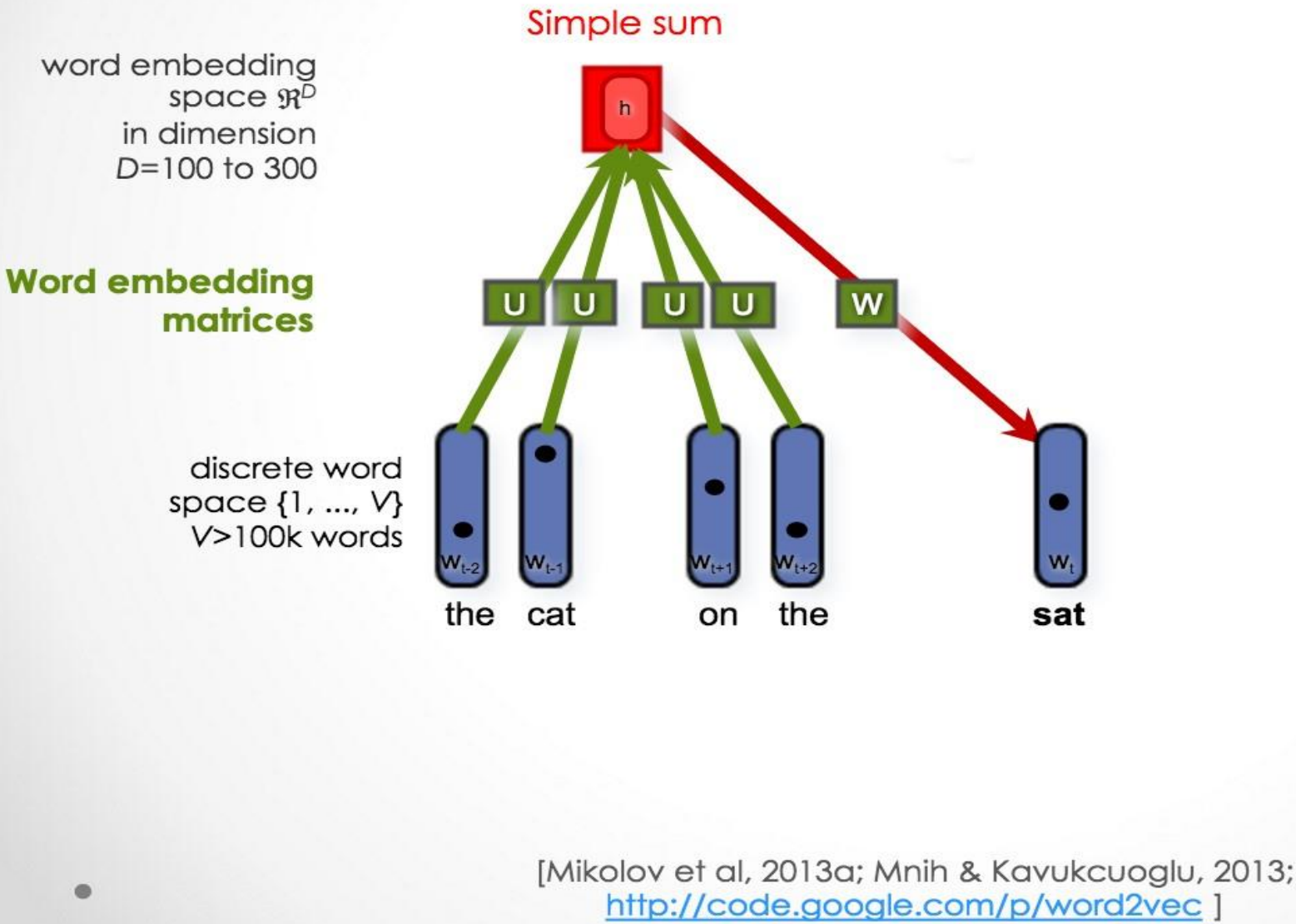
http://www.slideshare.net/hen_drik/word2vec-from-theory-to-practice

Sparse Vs Dense Feature Representation:
Example: "the dog"



source: http://u.cs.biu.ac.il/~yogo/nnlp.pdf

CBOW



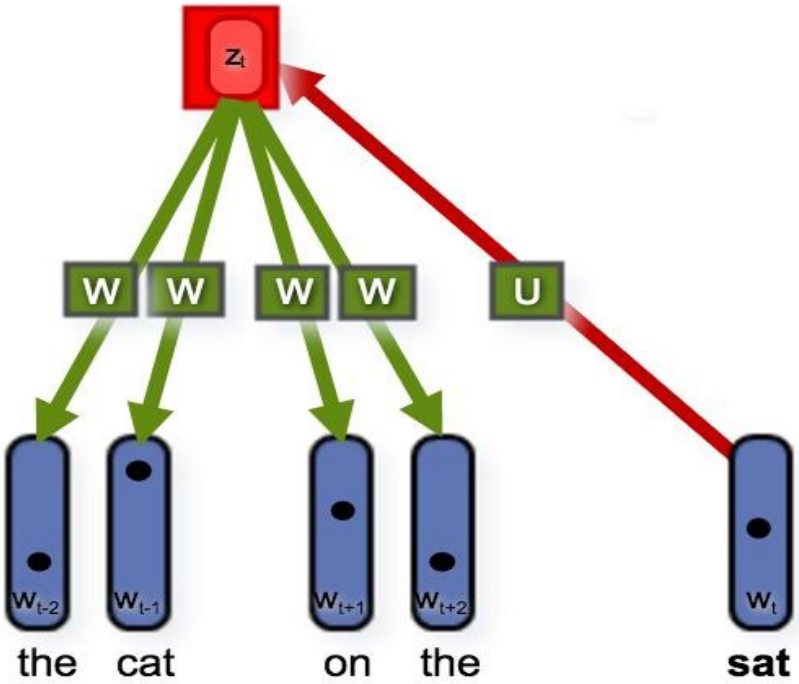
source: http://slideplayer.com/slide/4559559/

Skip Grams

word embedding space $\Re^D$ in dimension D=100 to 1000

Word embedding matrices

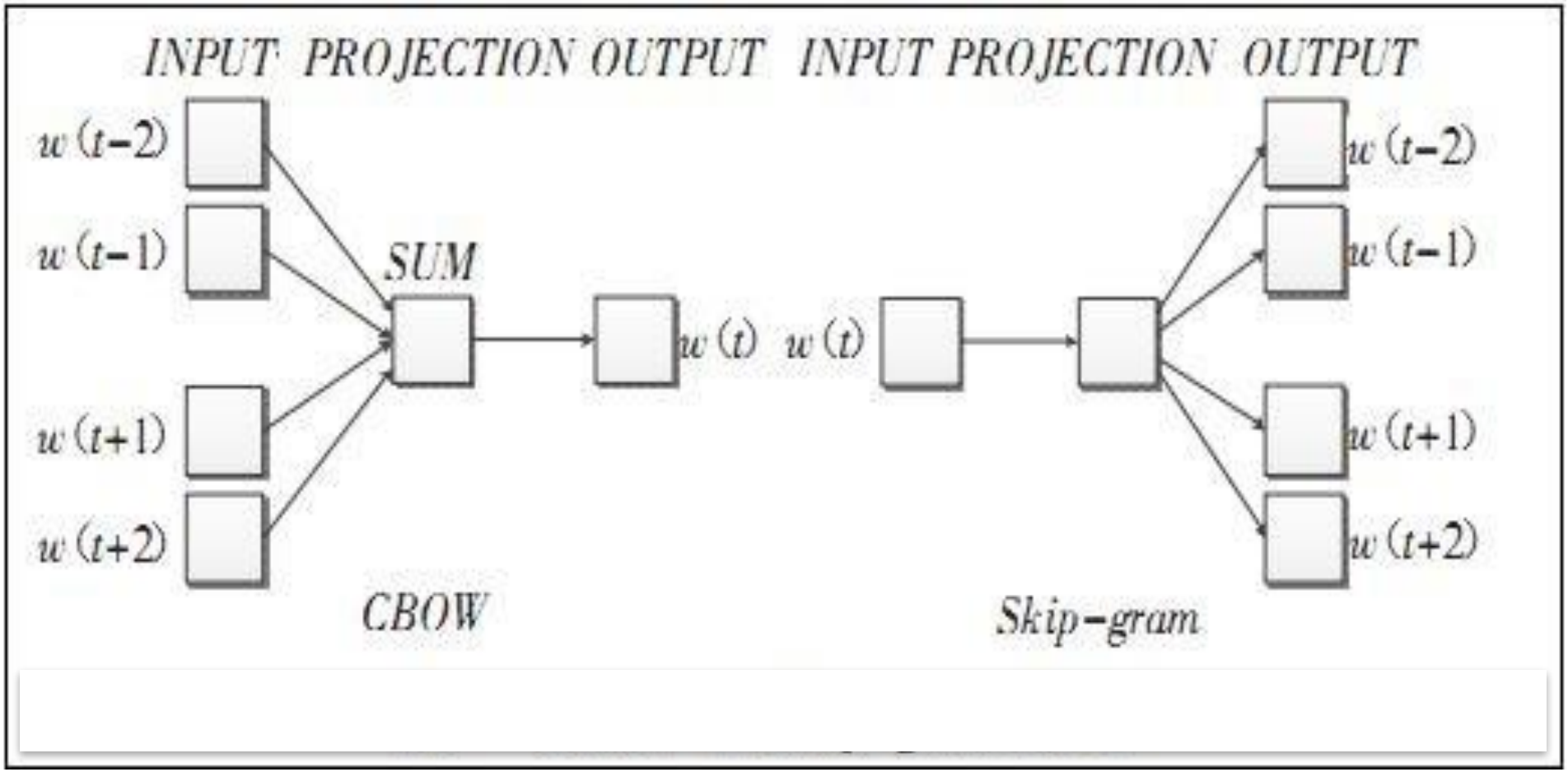
discrete word space {1, ..., V} V>100k words



[Mikolov et al, 2013a, 2013b; Mnih & Kavukcuoglu, 2013; http://code.google.com/p/word2vec]

source: http://slideplayer.com/slide/4559559/

CBOW Vs Skip-gram



source: http://blog.csdn.net/eastmount/article/details/50637476

word2vec (Google)

Use documents to train a neural network model maximizing the conditional probability of context given the word

Apply the trained model to each word to get its corresponding vector

Calculate the vector of sentences by averaging the vector of their words

Construct the similarity matrix between sentences

Use pagerank to score the sentences in graph

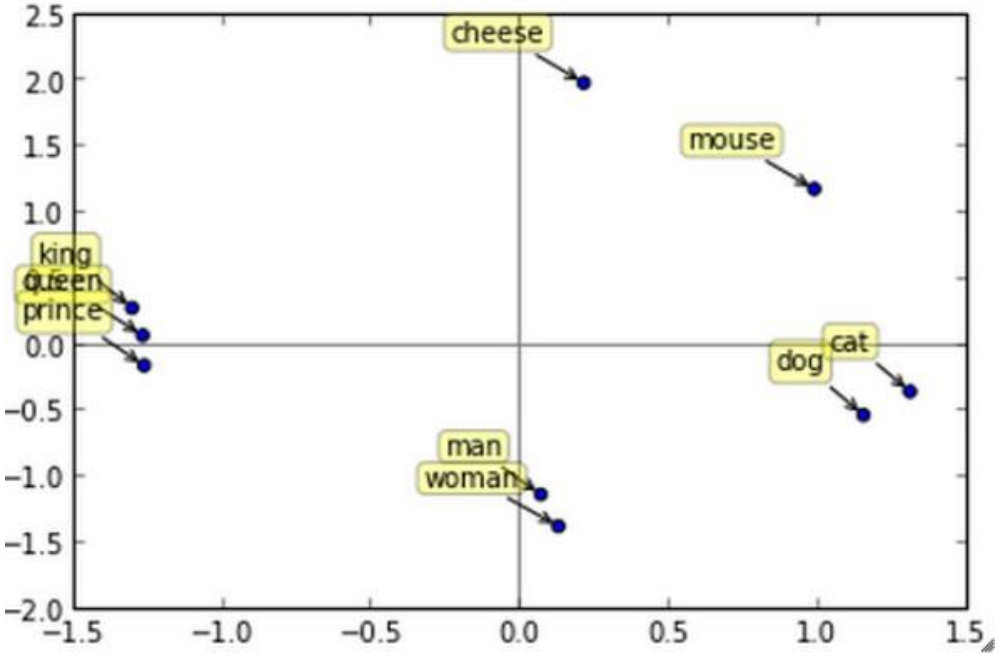
Source: http://www.slideshare.net/andrewkoo/word2vec-algorithm

word2vec

word2vec associates words to points in space

Similar words are closer together

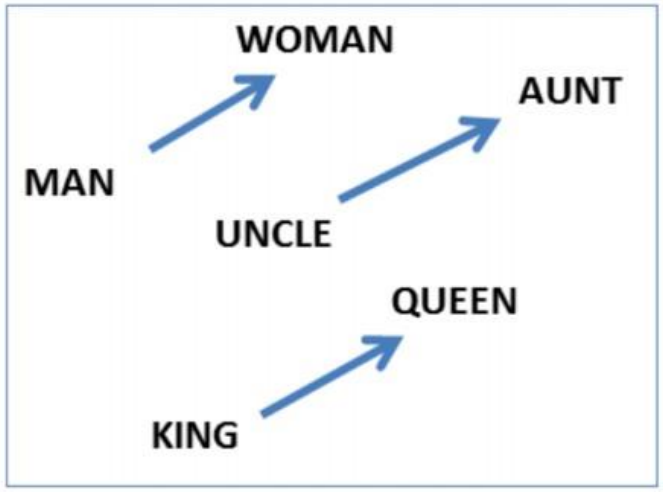
Developed by Mikolov, Sutskever, Chen, Corrado and Dean in 2013 at Google Research



source: http://files.meetup.com/12426342/5_An_overview_of_word2vec.pdf

word2vec

Word relationships are displacements



Source: *Linguistic Regularities in Continuous Space Word Representations*, Mikolov et al, 2013

vec[queen] – vec[king] = vec[woman] – vec[man]

source: http://files.meetup.com/12426342/5_An_overview_of_word2vec.pdf


HANDS-ON

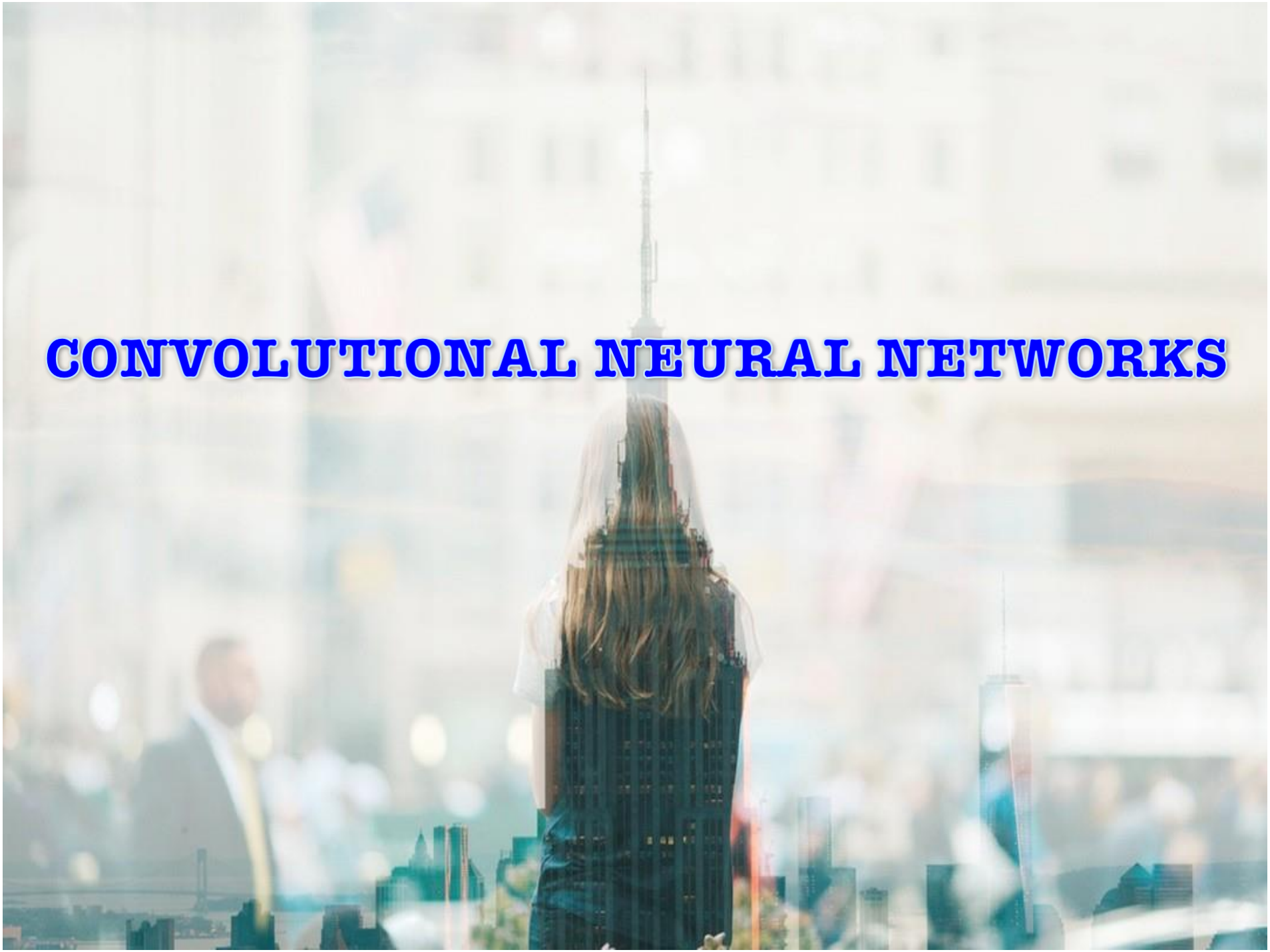
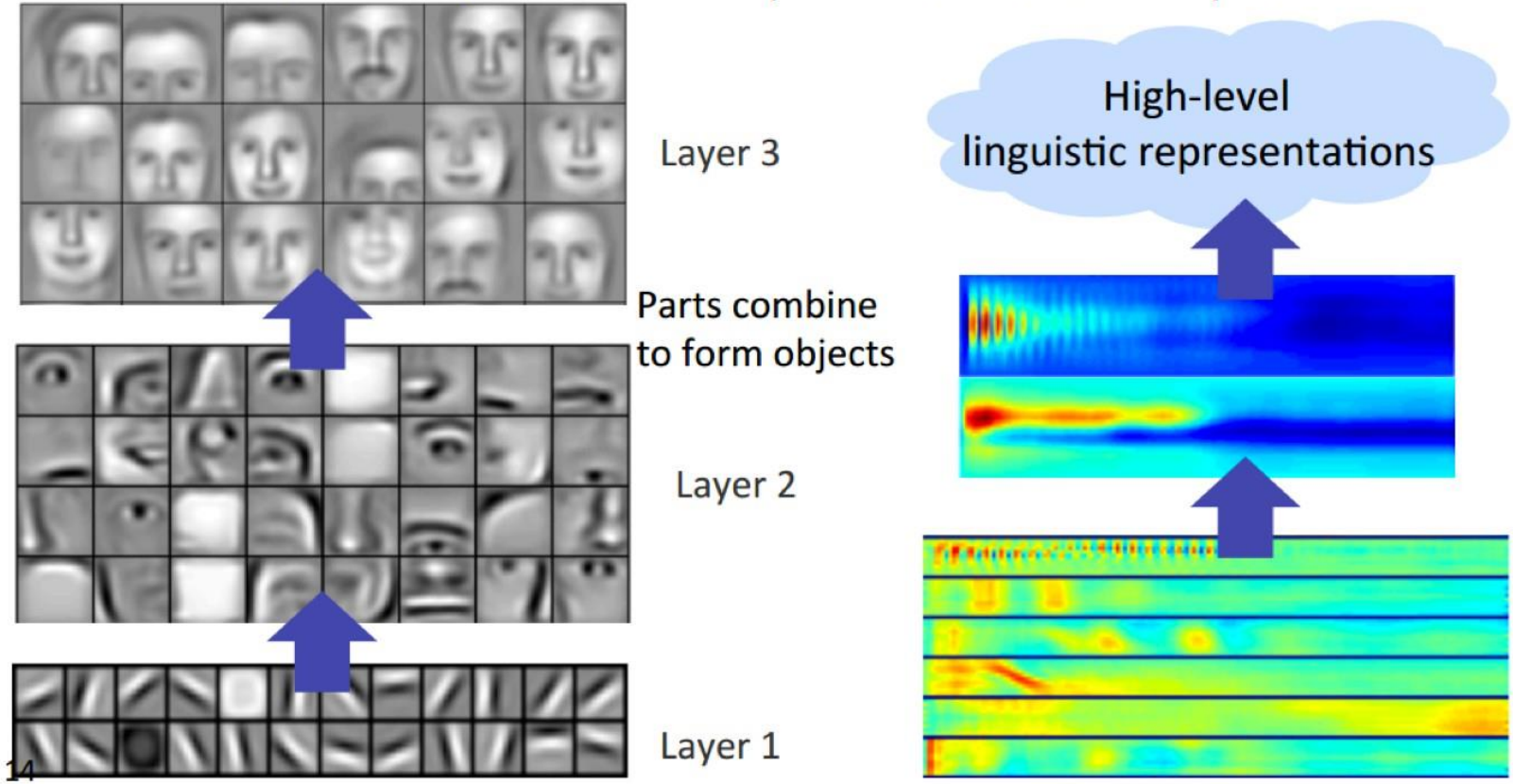

CONVOLUTIONAL NEURAL NETWORKS

Image Recognition/Classification

Successive model layers learn deeper intermediate representations



Prior: underlying factors & concepts compactly expressed w/ multiple levels of abstraction

Source: http://www.iro.umontreal.edu/~bengioy/talks

Convolution

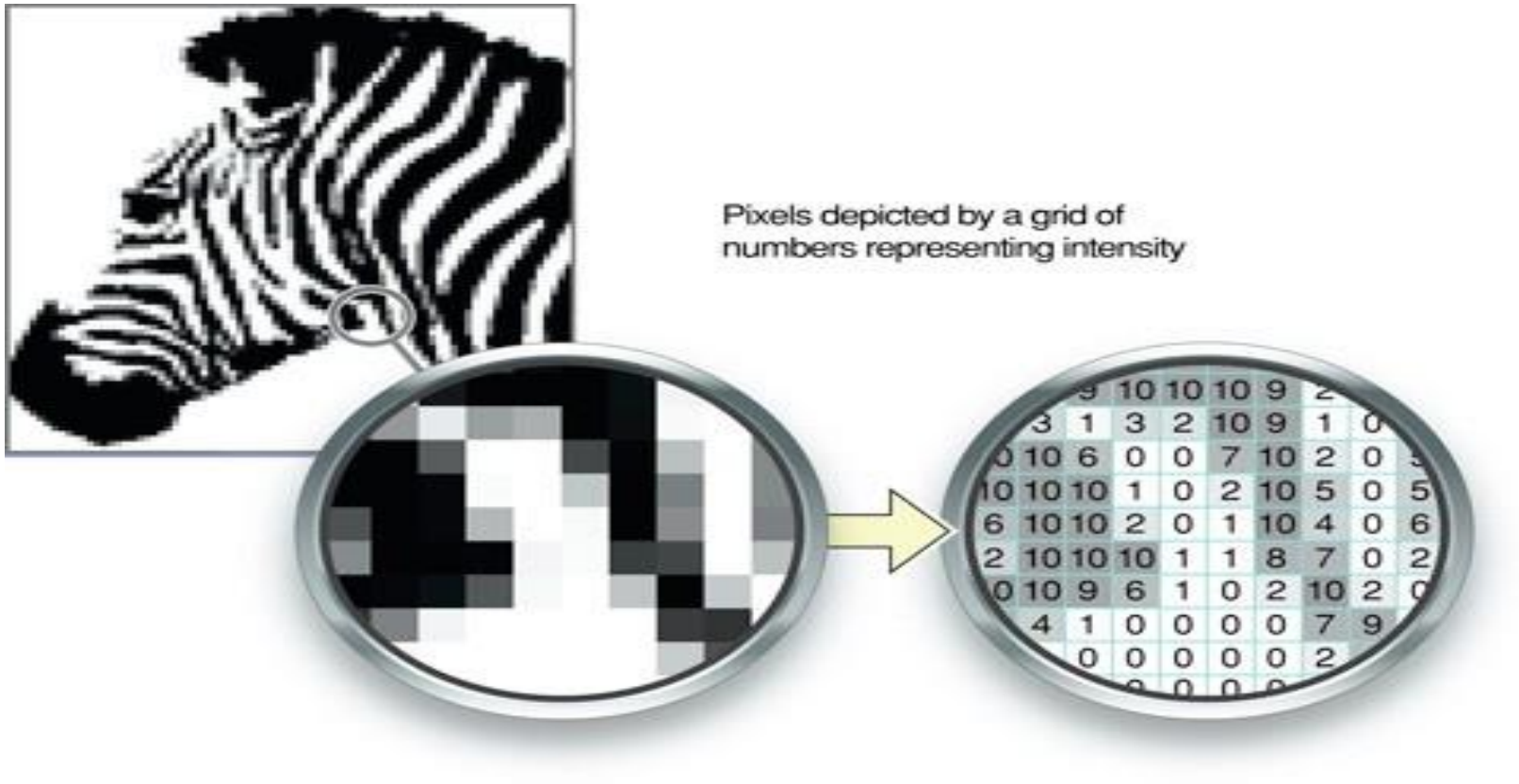
Image Processing Technique to change intensities of a pixel to reflect the intensities of the surrounding pixels. Eg: image effects like blur, sharpen, and edge detection

Original

Emboss

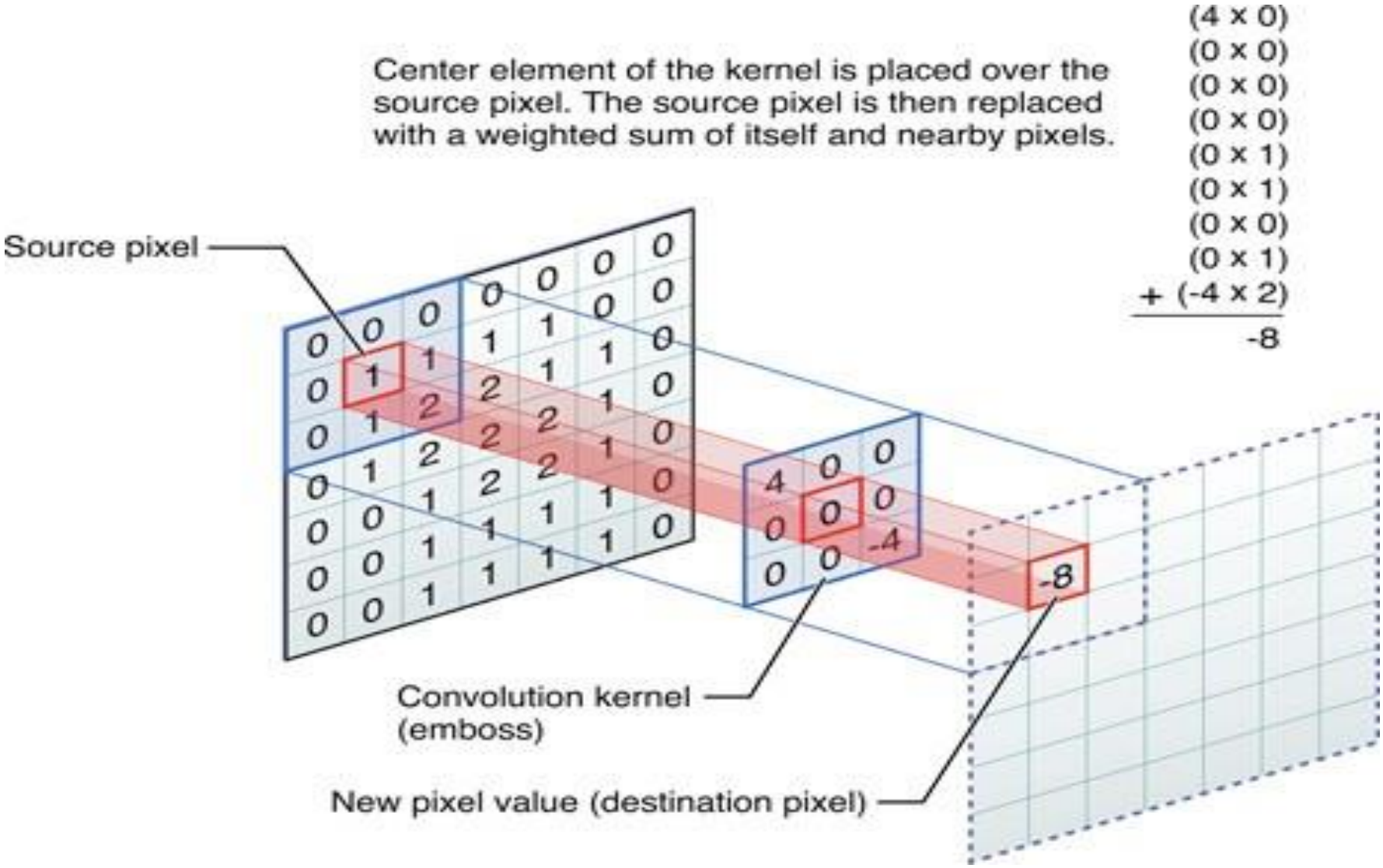
source: https://developer.apple.com/library/ios/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html

Convolution



source: https://developer.apple.com/library/ios/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html

Convolution

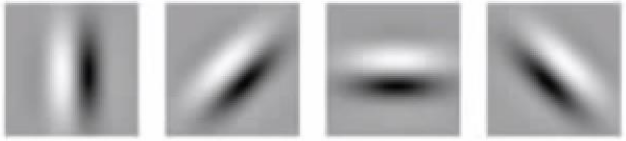


source: https://developer.apple.com/library/ios/documentation/Performance/Conceptual/vImage/ConvolutionOperations/ConvolutionOperations.html

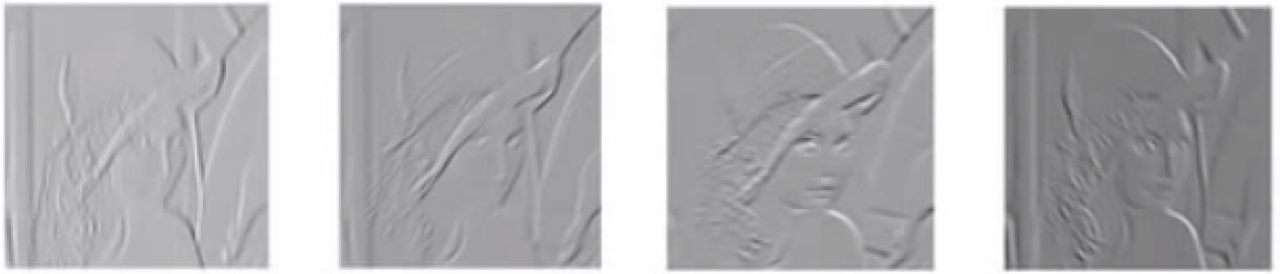
Convolution – One more example

Lena

Gabor filters



Convolved by Gabor filters lena



source: http://web.ics.purdue.edu/~adundar/teaching.html

Sub-sampling

Input 508x508 pixels Lena

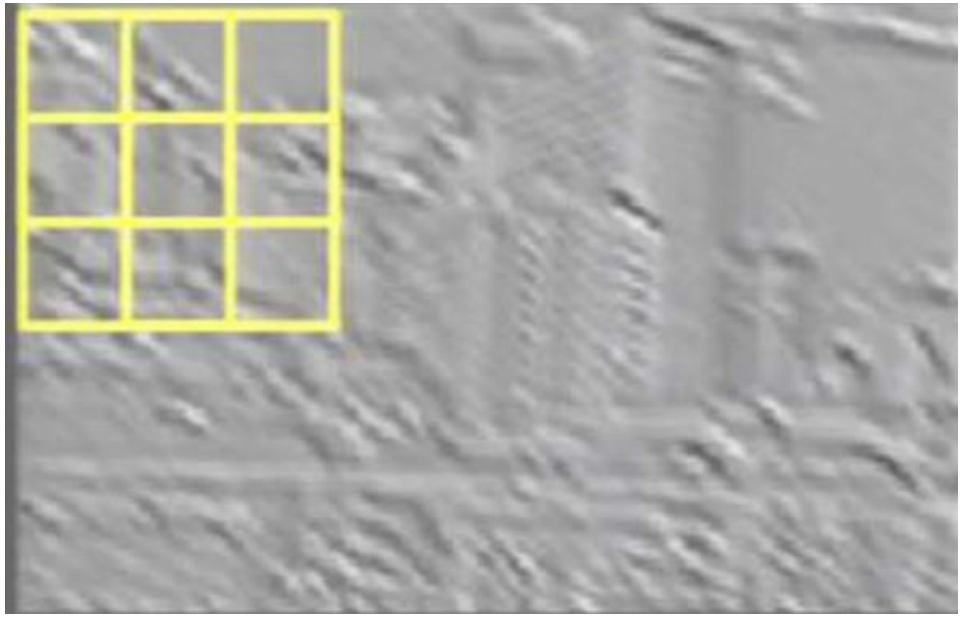
down-sampled by 2x2

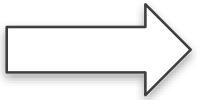
==========>

Input 254x254 pixels Lena

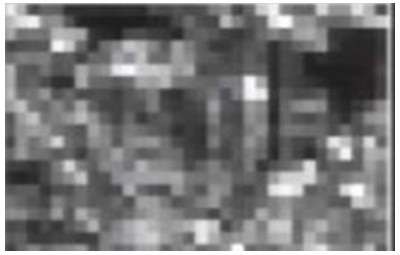
source: http://web.ics.purdue.edu/~adundar/teaching.html

Pooling

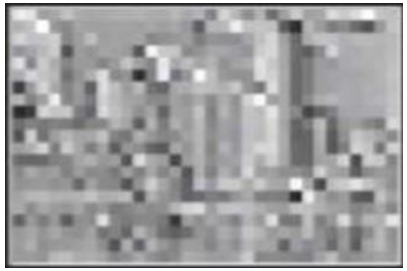




Max



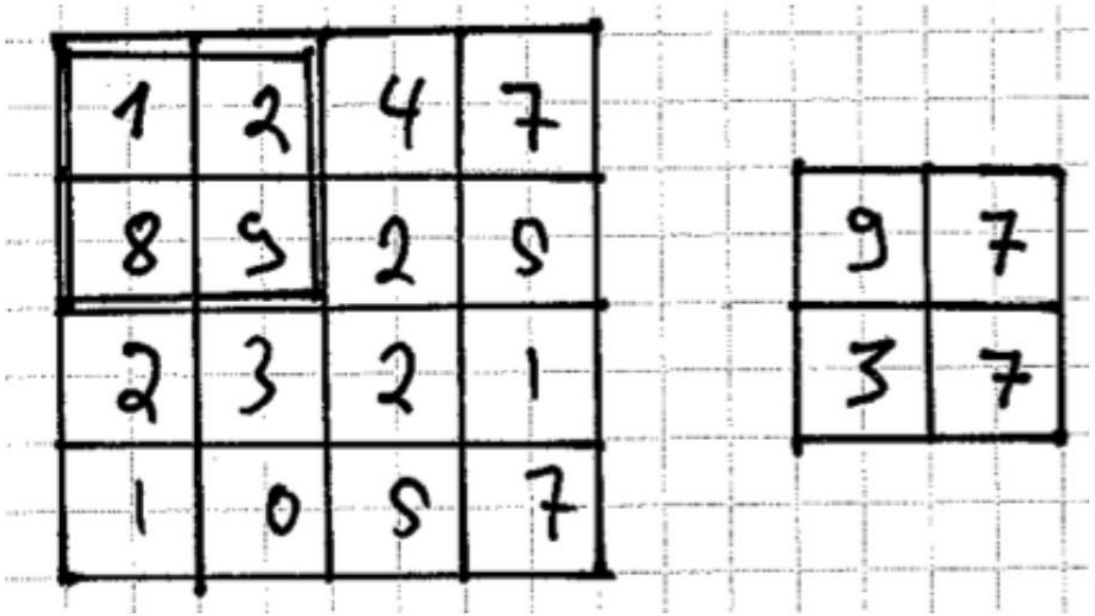
Sum



source: http://web.ics.purdue.edu/~adundar/teaching.html

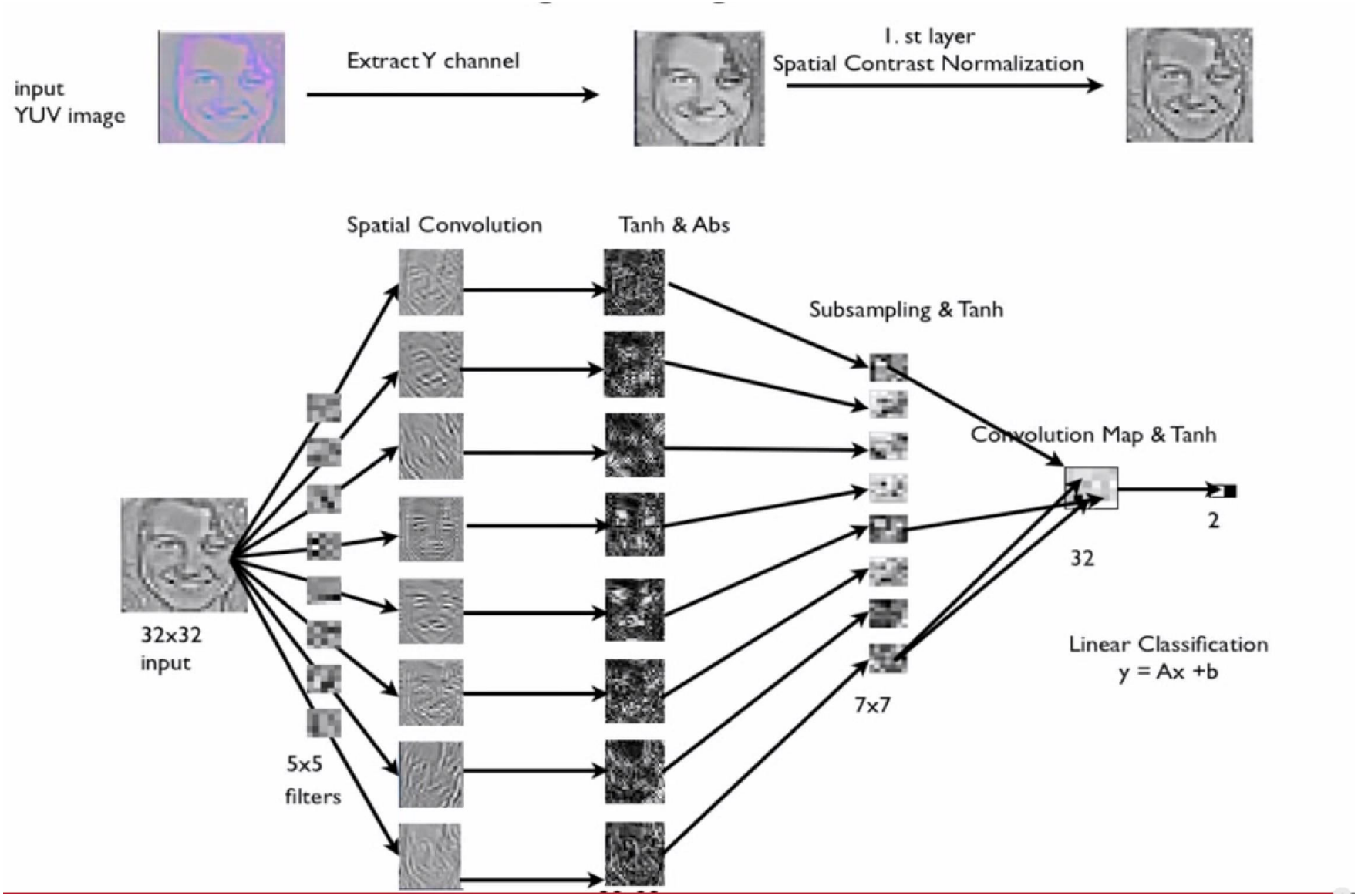
Max Pooling

Hinton: The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster“



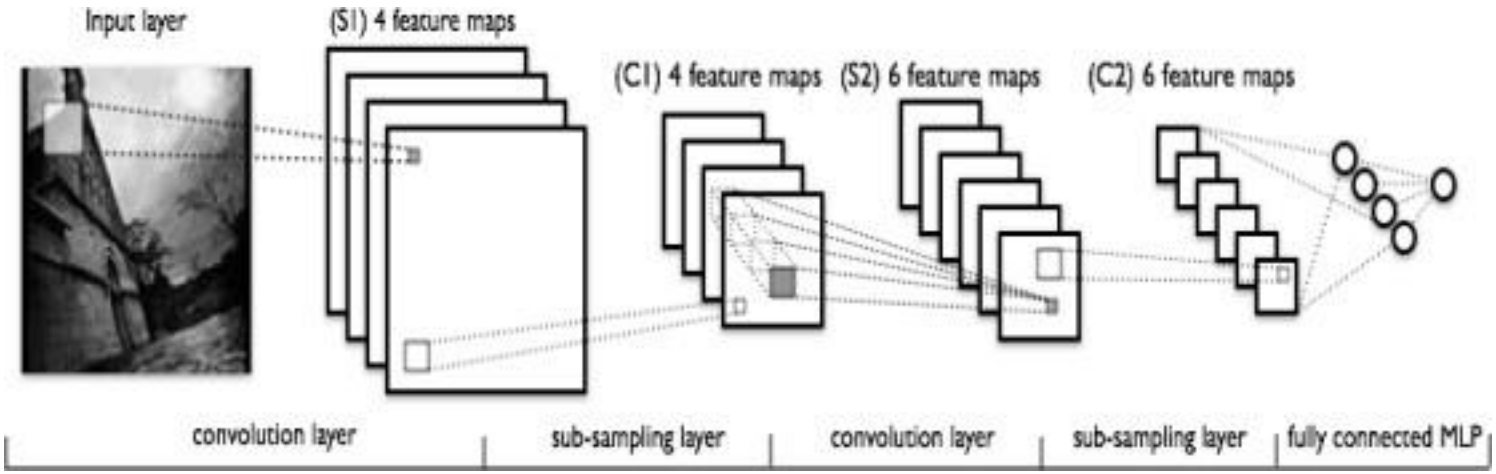
Simply join e.g. 2x2 adjacent pixels in one.

Basic Architecture



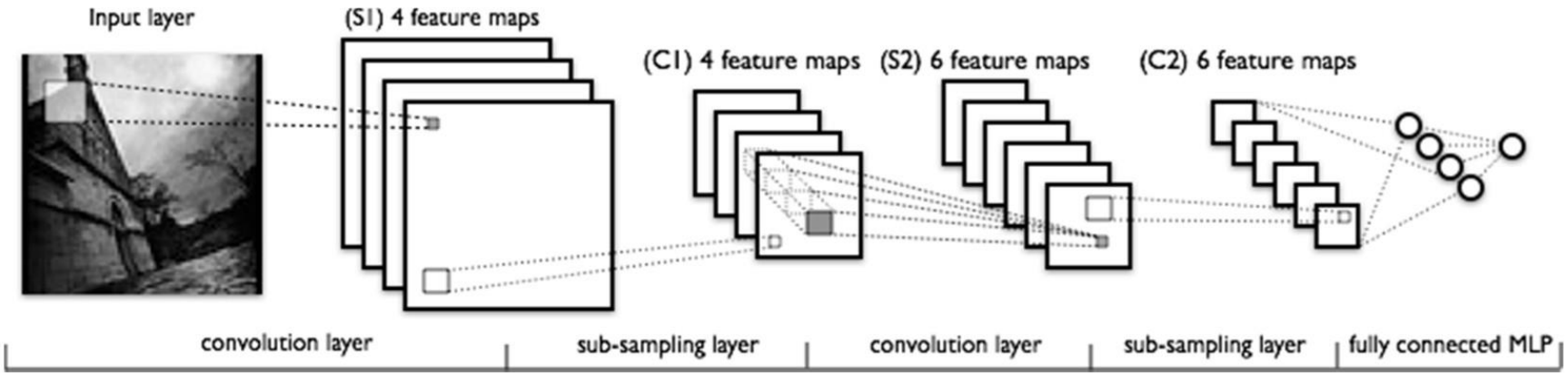
source: http://web.ics.purdue.edu/~adundar/teaching.html

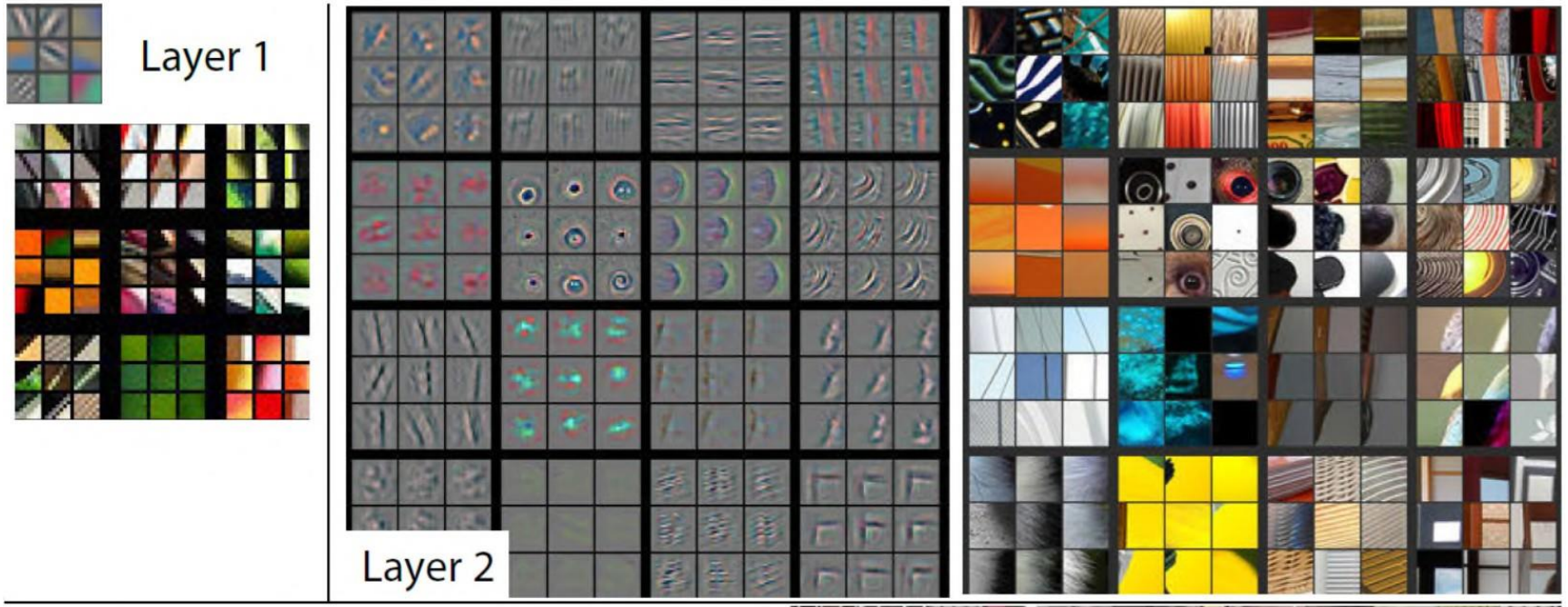
CNN Architecture



Source: http://deeplearning.net

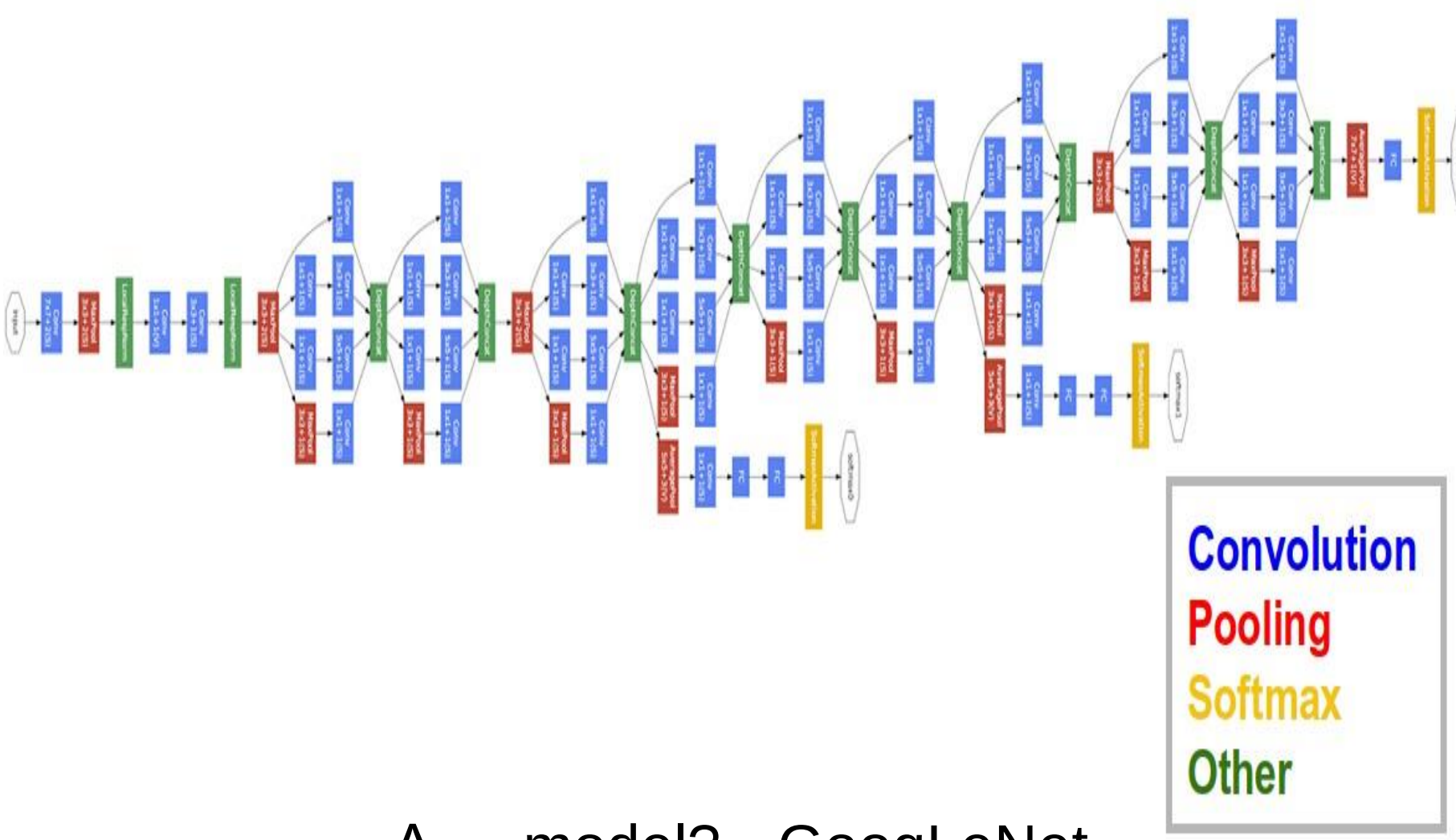
CNN Architecture





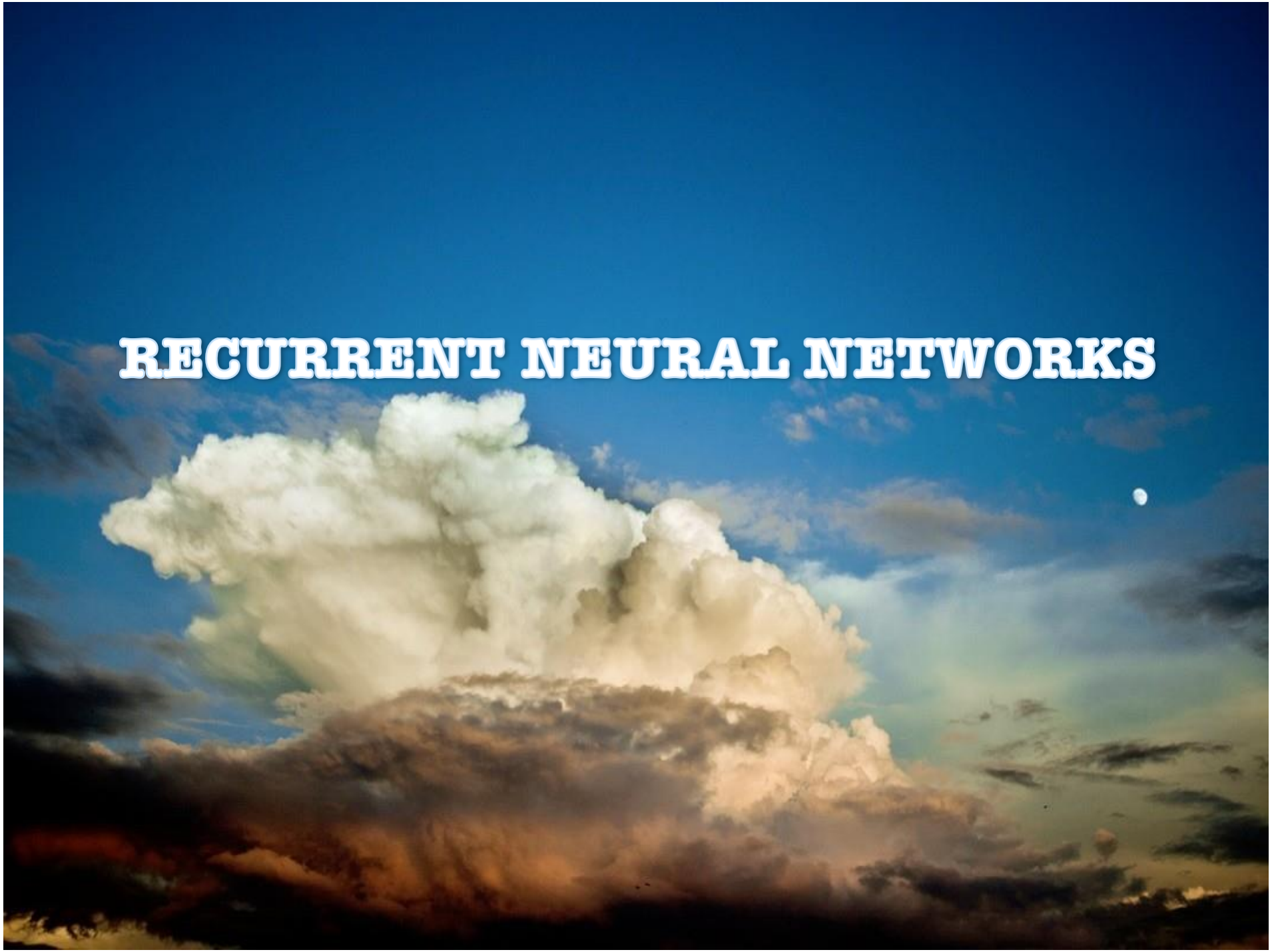
[Matthew Zeiler & Rob Fergus]

Example of a CNN model

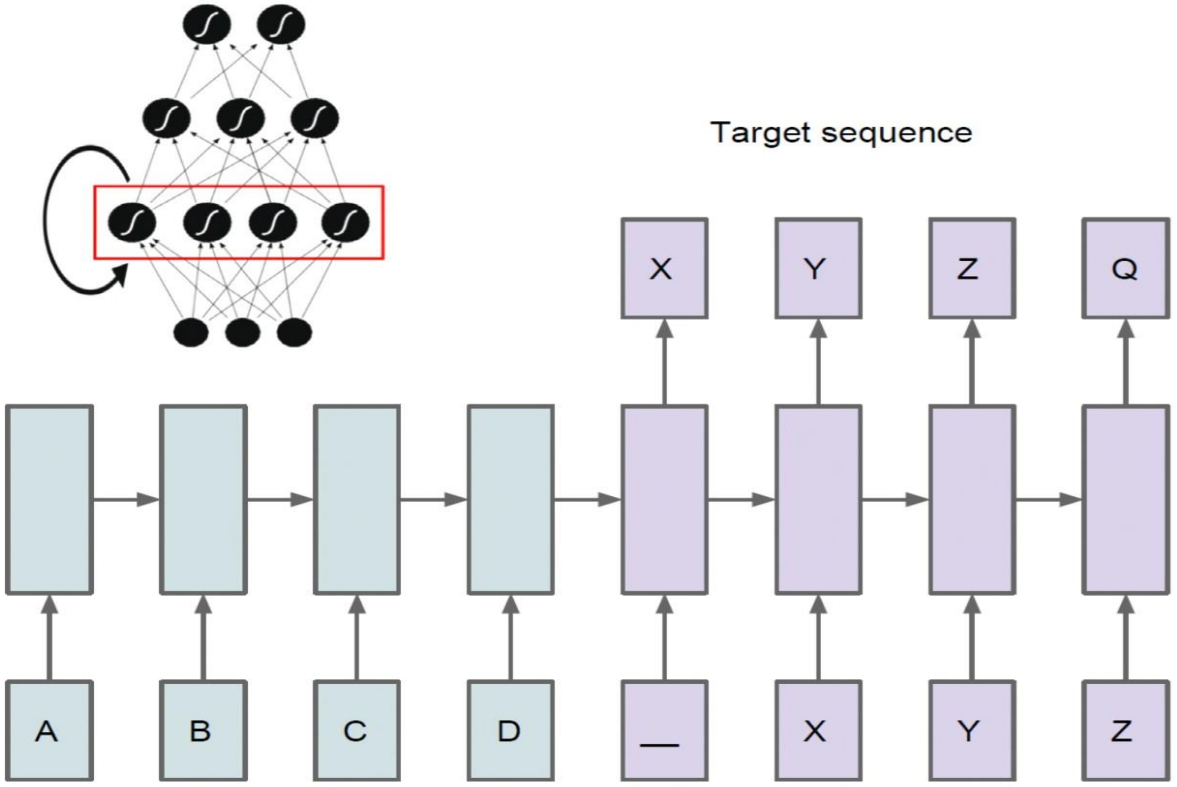


A … model? - GoogLeNet

source: arXiv:1409.4842v1 [cs.CV] 17 Sep2014

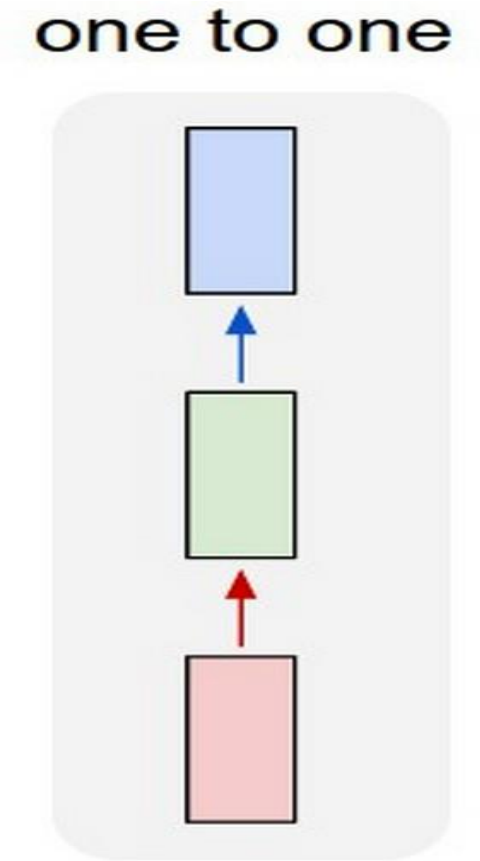

RECURRENT NEURAL NETWORKS

Recurrent Neural Network



[Sutskever et al, Graves et al]

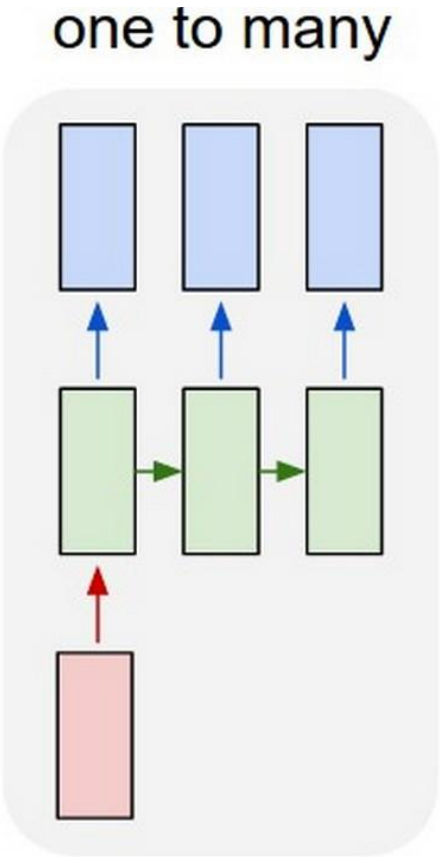
One-to-One Sequence



Vanilla-model. No RNN

source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

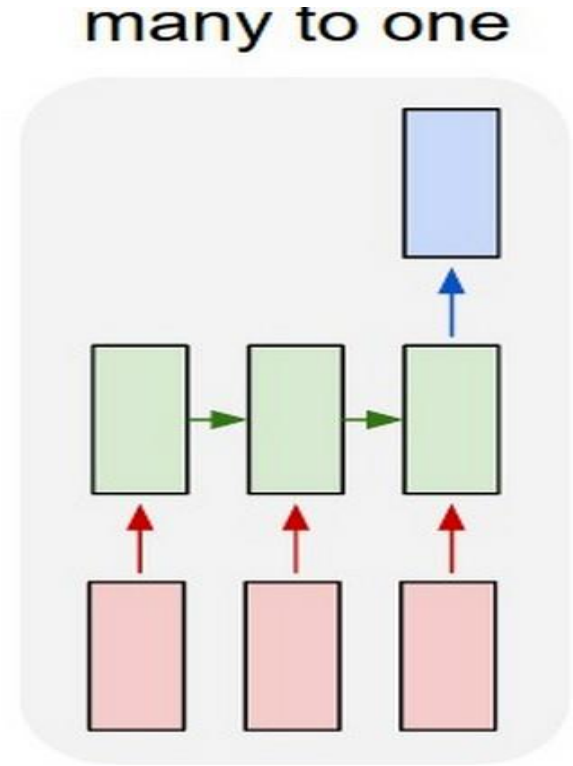
One-to-Many Sequence



Sequence output
(Eg:Recognize image and explain it in words)

source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

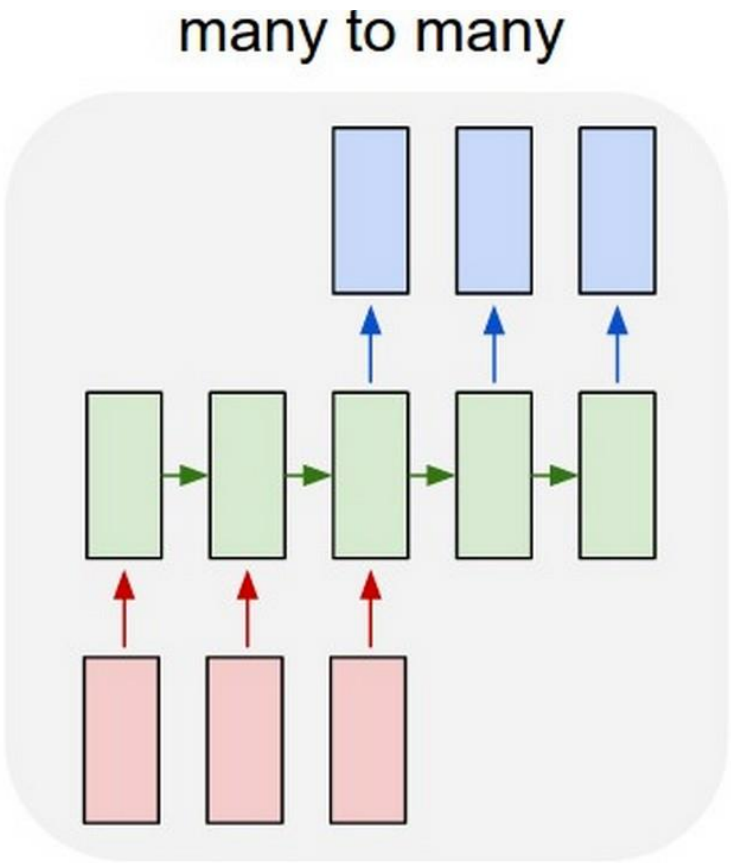
Many-to-One Sequence



Sequence input (Eg:Sentiment analysis of a sentence)

source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

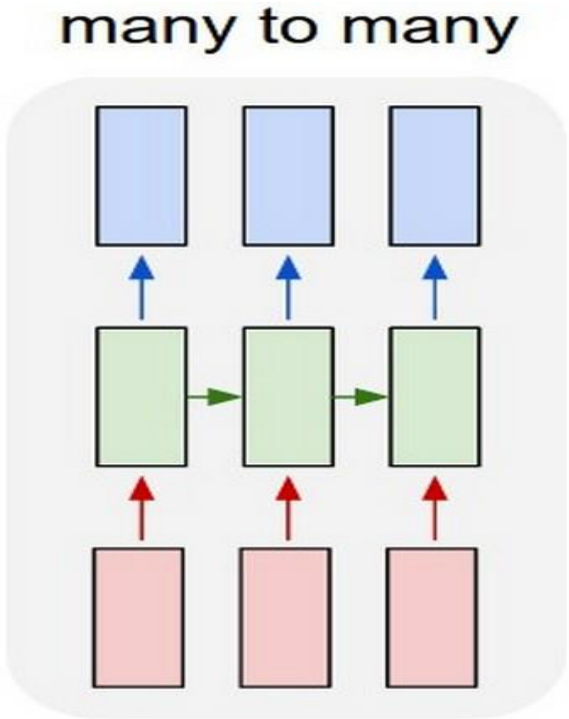
Many-to-Many Sequence



Sequence input (Eg:Machine Translation – English to Spanish)

source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

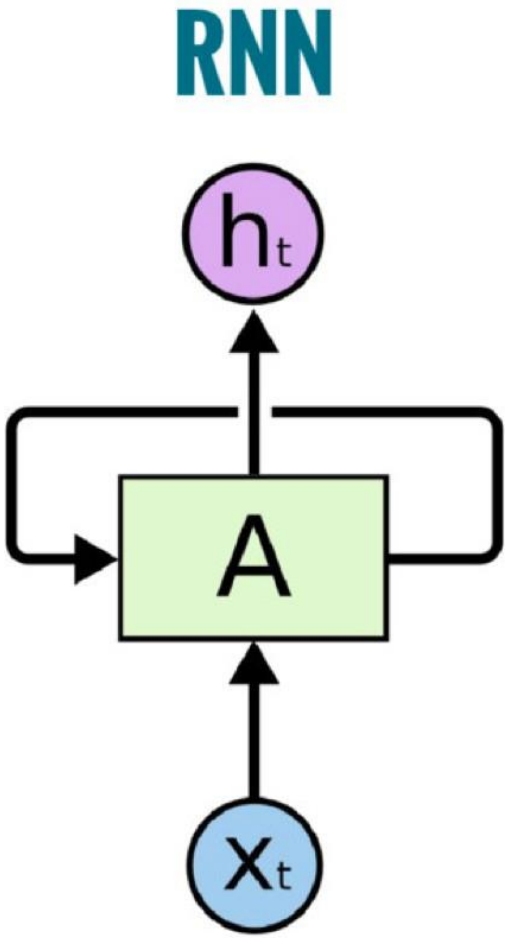
Many-to-Many Sequence



Synced Sequence input & Output (Eg:Video classification of each frame)

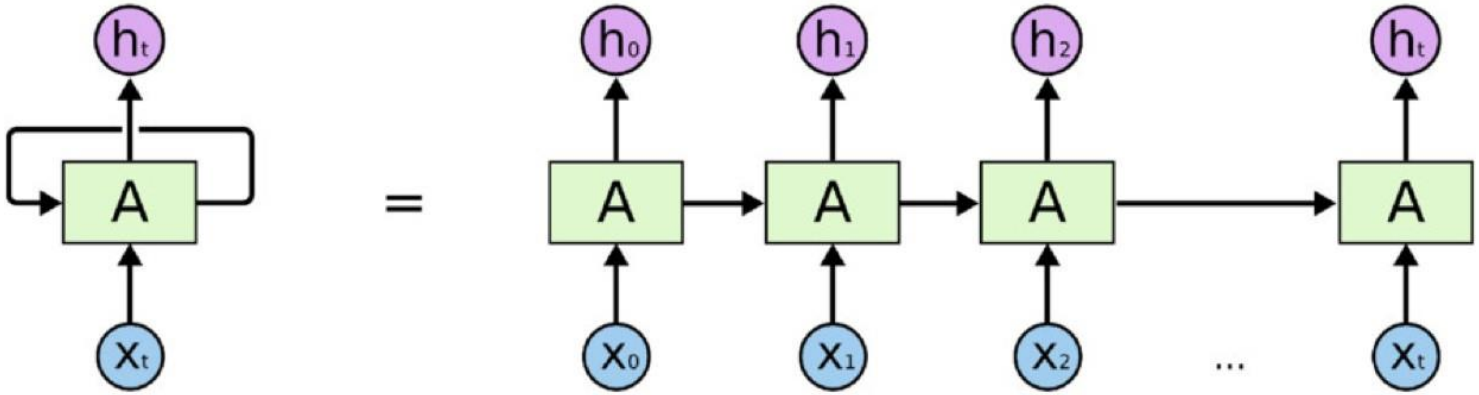
source: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Many-to-Many Sequence



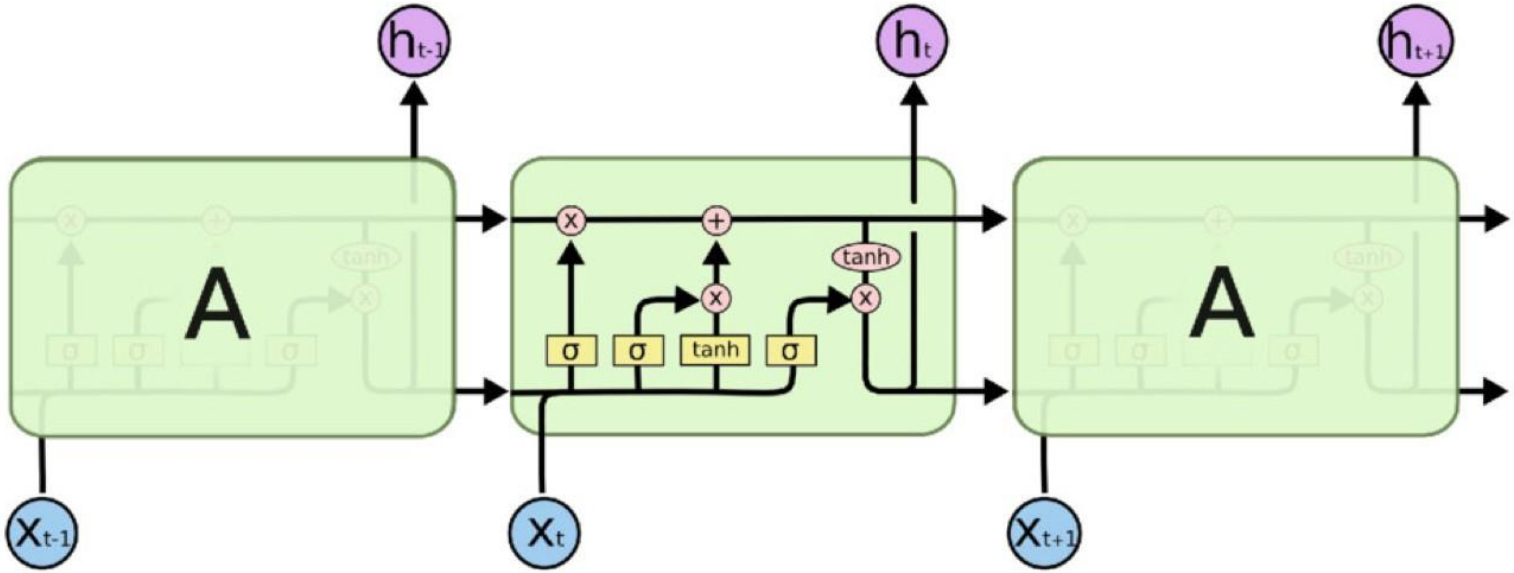
source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Many-to-Many Sequence



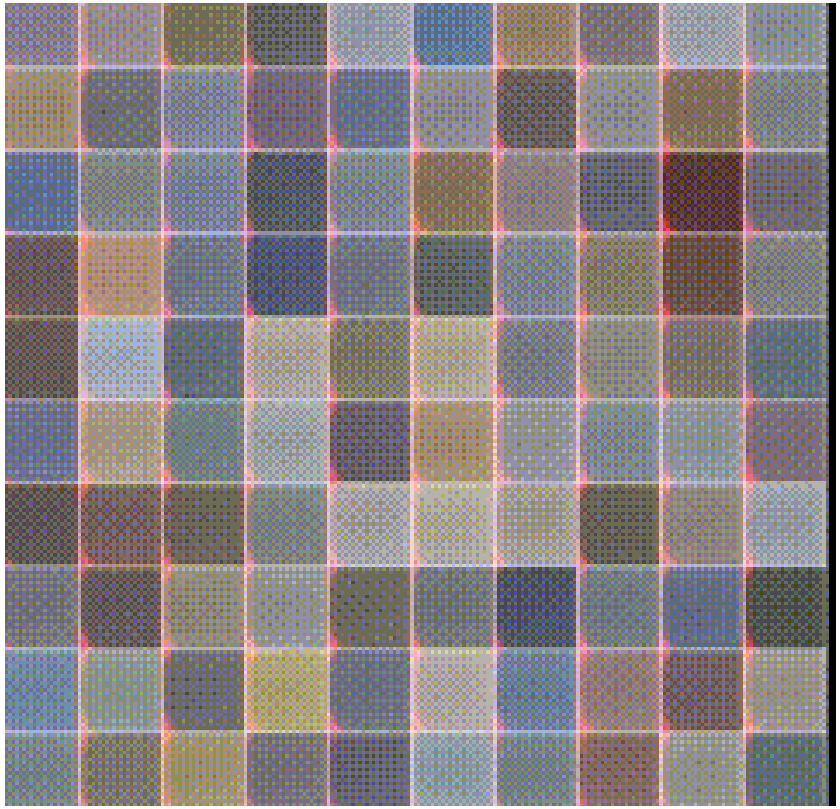
source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory



source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

RNN – Generating Images



Source: http://arxiv.org/abs/1502.04623c

Effect of Dataset Size

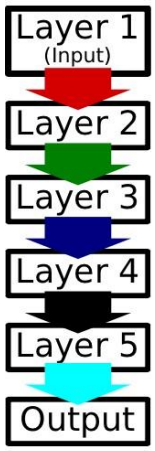
- RNNs have poor generalization properties on small datasets.
  - 1K labeled examples 25-50% worse than linear model…
- RNNs have better generalization properties on large datasets.
  - 1M labeled examples 0-30% better than linear model.
- Crossovers between 10K and 1M examples
  - Depends on dataset.

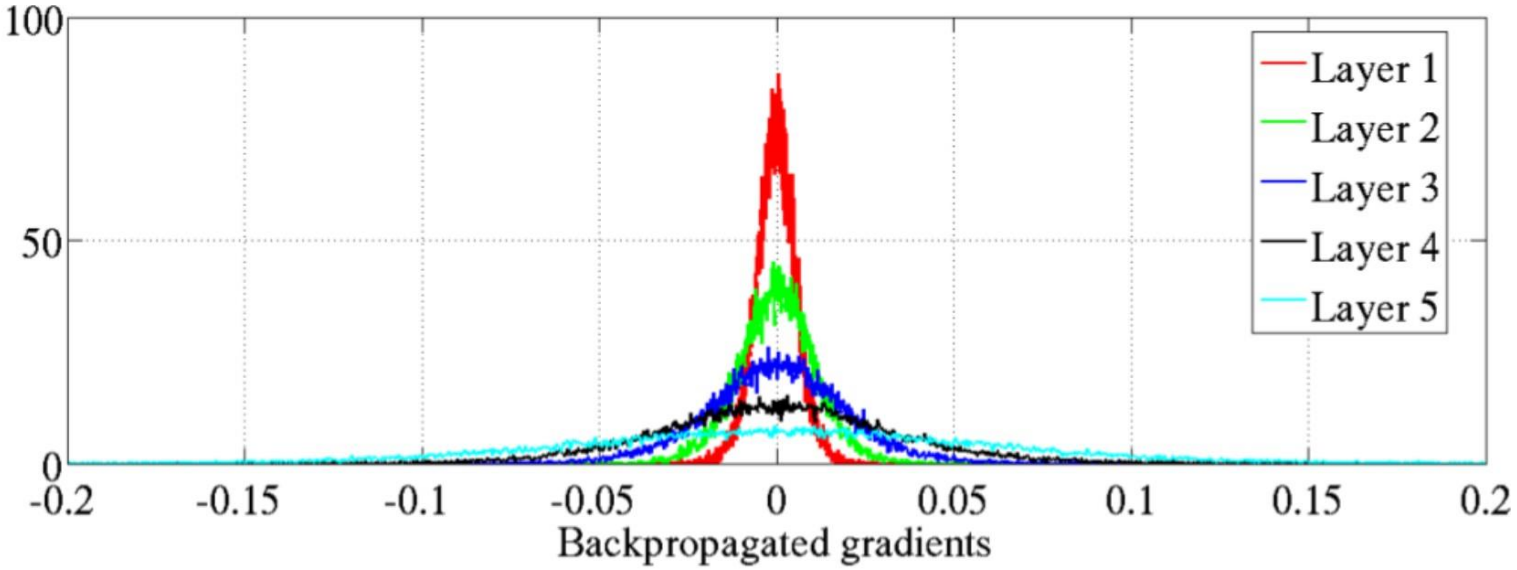
Source: http://www.slideshare.net/indicods/general-sequence-learning-with-recurrent-neural-networks-for-next-ml


CHALLENGES IN DEEP LEARNING

People who are more likeable and attractive are generally perceived as more intelligent. This is called the "Halo Effect."

Vanishing Gradient





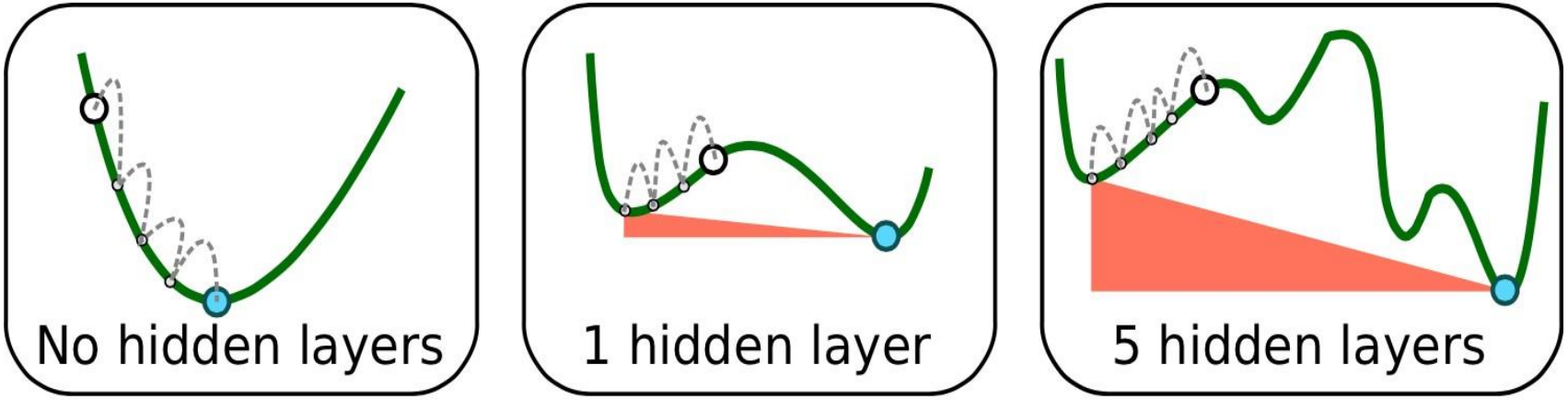
- In backpropagation, the gradient becomes zero in the layers nearest to the input.
- Intuition : layers near to the output have larger influence in the error rate.
- Training deep layers successfully is terribly slow.
- Until deep layers learn some sensible weights, subsequent layers are trying to learn mostly from noise.

(*) Glorot & Bengio. Understanding the difficulty in training deep feedforward neural networks.

source: ASDM Summer School on Deep Learning 2014

High Non-Linearity – Local Minima

- Backpropagation is a gradient descent method.
- Only guarantees finding a local minimum of the error function, which can be worse than the global minimum .
- When introducing more non–linearities in the network (more layers), local minima multiply .



- It is hard to find good weight configurations for a deep network.

source: ASDM Summer School on Deep Learning 2014

Alleviating vanishing gradients & local minima

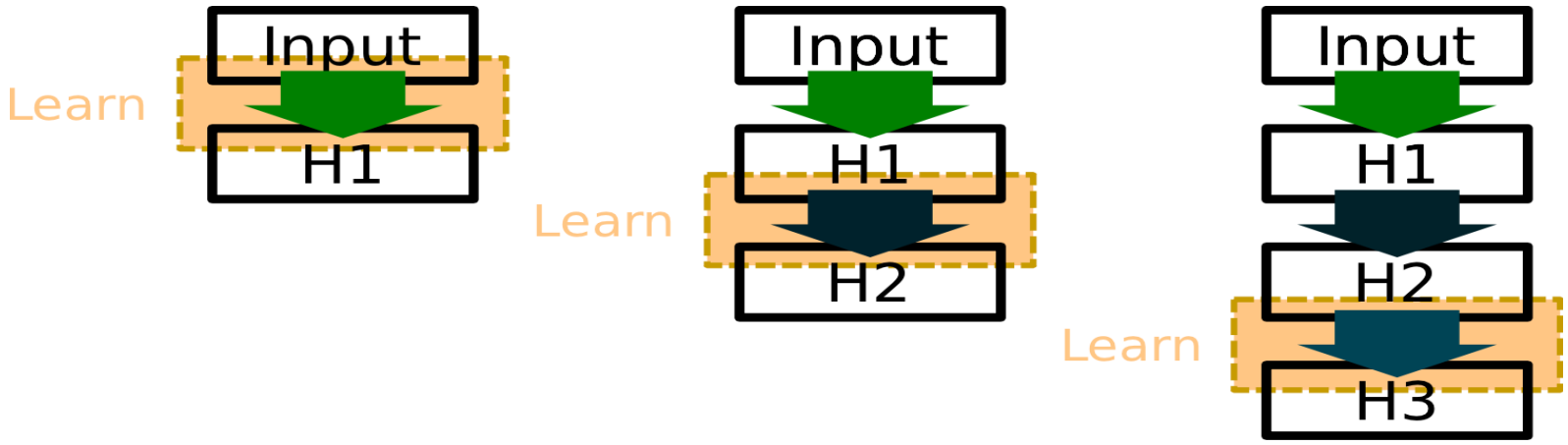
- Smart weight initialization
- Use RL and Maxout units
- Train for a longer time .
- Use restarts .

Incremental Pre-training

Instead of learning all the network weights jointly, build each layer iteratively .

Incremental Pre-training

- Start learning the layer nearest to inputs.
- proceed adding layers until last hidden layer
- When adding a new layer, weights from previous layers are kept fixed .
- One layer is learned at a time, vanishing gradients are avoided .



Incremental Pre-training

Two options for output

- Keep pre–trained weights fixed .
- Deep network ≃ feature generator : more useful features are obtained from the inputs.
- The output layer is trained as a perceptron , inputs are the features obtained.

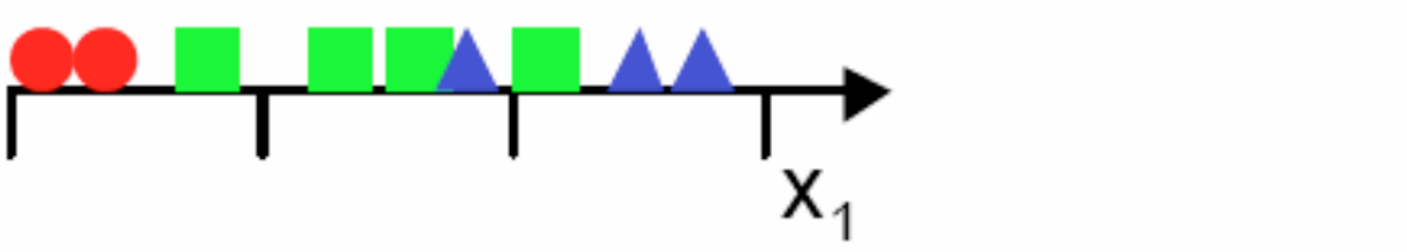
- Discriminative fine–tuning . Optimize jointly over all network weights (e.g. using BP).
- Pre–trained weights seem to be a better choice than random initialization .
- Slower Fine tuning


UNSUPERVISED LEARNING

Curse of Dimensionality

Single feature

Divide feature space into 3 simple bins

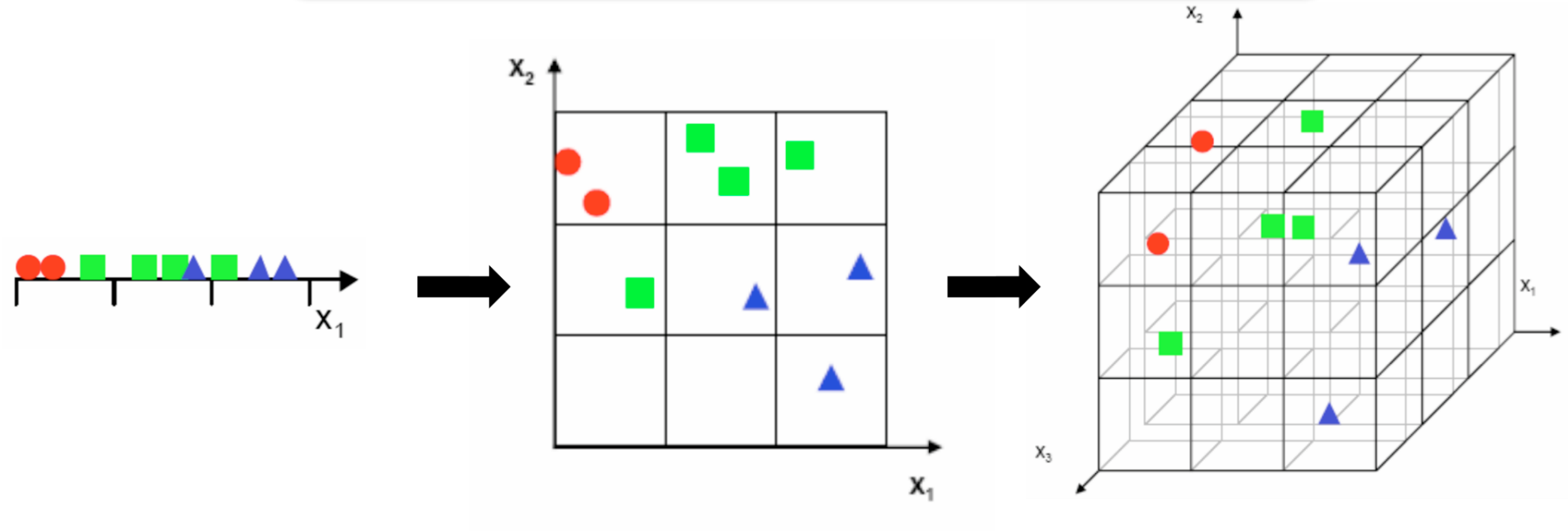


Too simplistic –
Too much overlap between classes.

source:http://nikhilbuduma.com/2015/03/10/the-curse-of-dimensionality/

Curse of Dimensionality

But by adding features to improve separability, the feature space explodes.



source: http://nikhilbuduma.com/2015/03/10/the-curse-of-dimensionality/

Feature Selection

Need: Automated feature selection

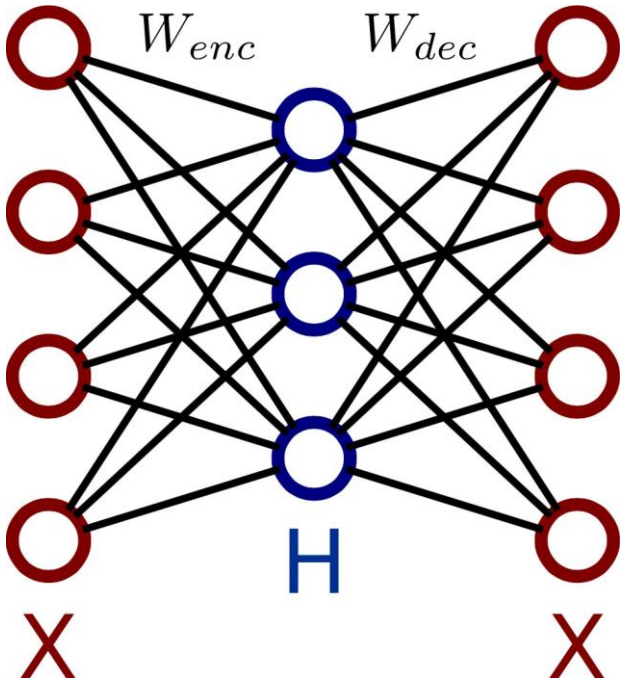
source:http://nikhilbuduma.com/2015/03/10/the-curse-of-dimensionality/

Autoencoders

GOAL: Efficient Reconstruction of Input Data

Use input as input and output of the network and train (Don't use the target)

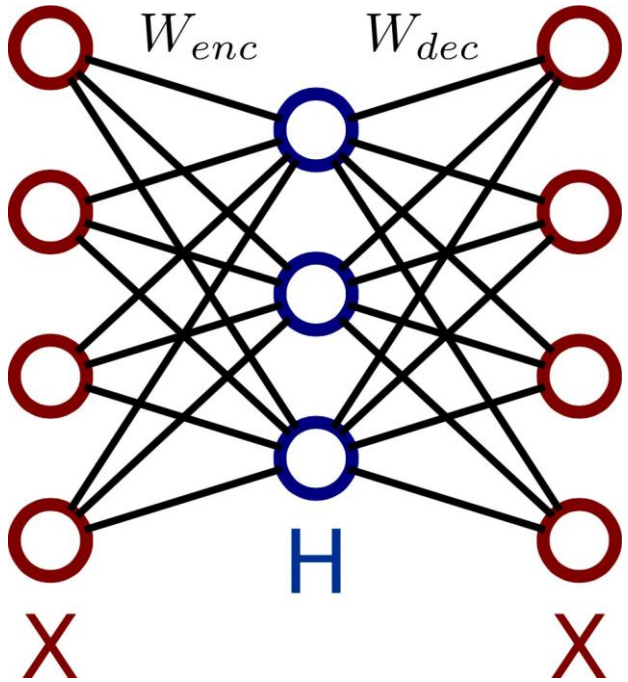
Learn the weights using a standard neural network method (e.g. backpropagation).



Autoencoders - Output

After learning,
W_{enc} : compresses the information in the data into the hidden units.

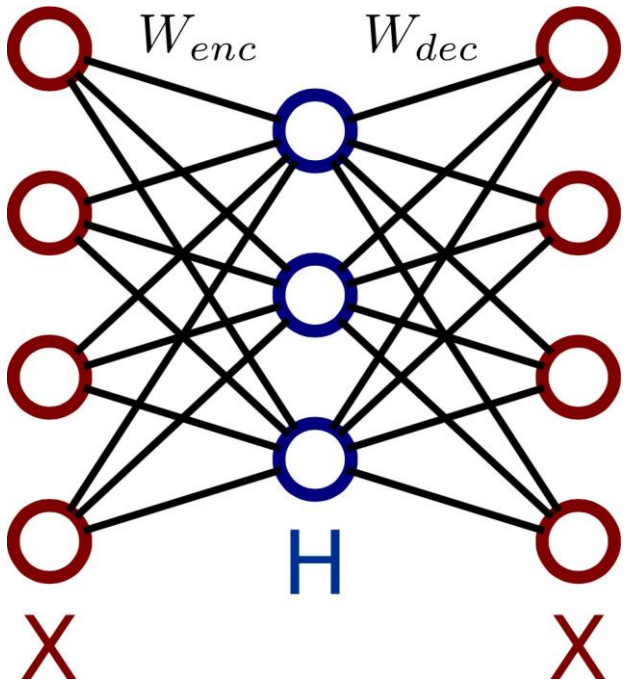
W_{dec} decompresses the info in the hidden units back to its original form *(with some loss)*.



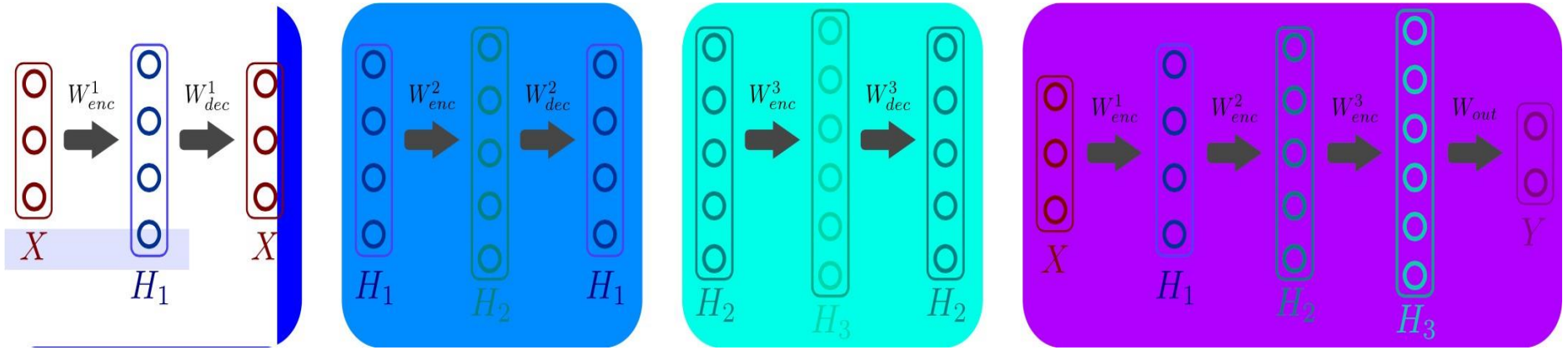
Autoencoders – To make them effective

Enforce sparsity .
(A network is sparse if only a few of its units >0 simultaneously)

Introduce random noise in the input patterns when training the network. (Only in the inputs!)



Deep Autoencoders



Train autoencoder to reconstruct the training data .

Proceed iteratively, building new autoencoders reconstructing the value of the hidden units of the previous stage.

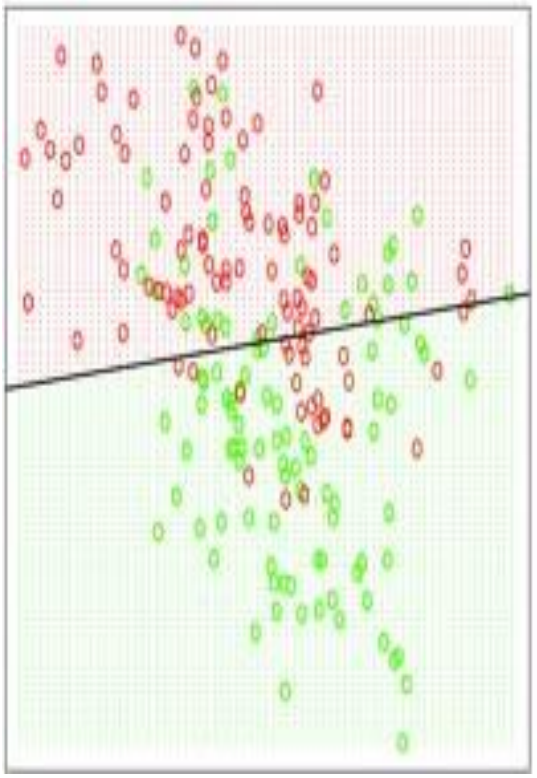
Create a feedforward network using the encoding weights W_{enc} from all the trained autoencoders. As output layer, use the training labels .

Fine–tuning : train the last layer or the full network using standard backpropagation.

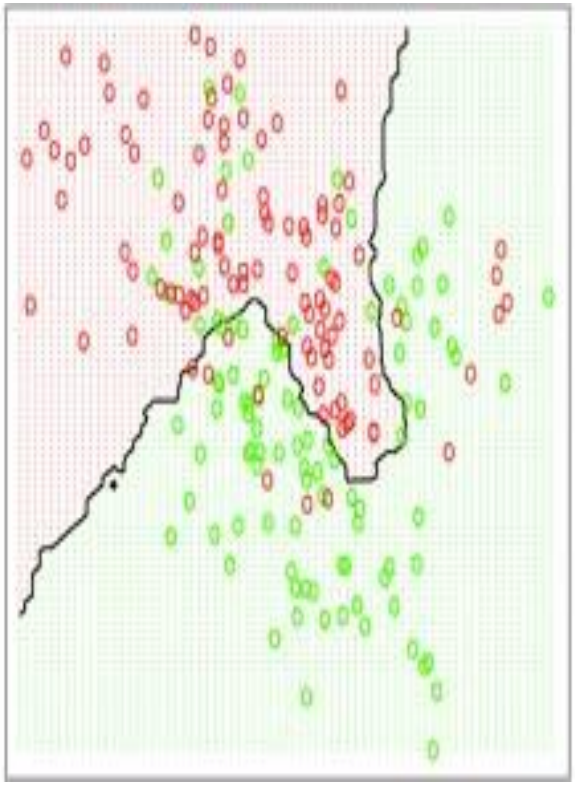

CHALLENGES IN DEEP LEARNING
- Continued

Overfitting

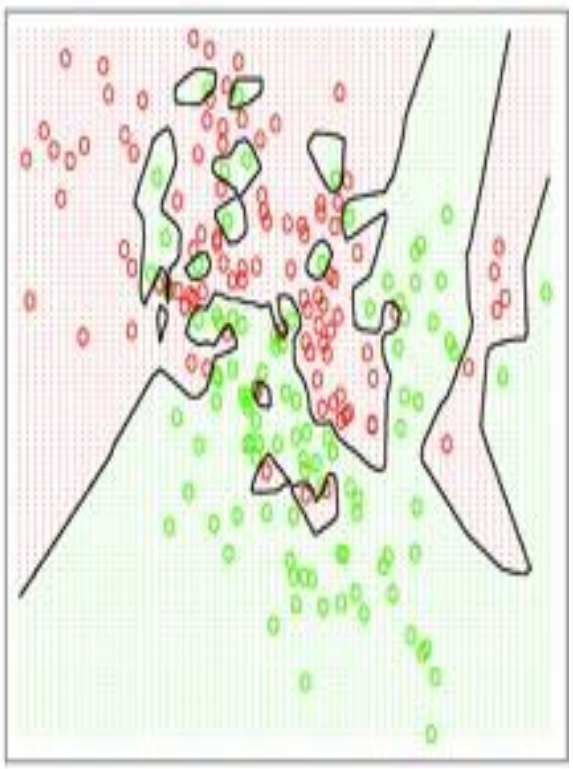
Underfitted Model



Good Model



Overfitted Model



source:http://mathbabe.org/2012/11/20/columbia-data-science-course-week-12-predictive-modeling-data-leakage-model-evaluation/

Some ways to address Overfitting

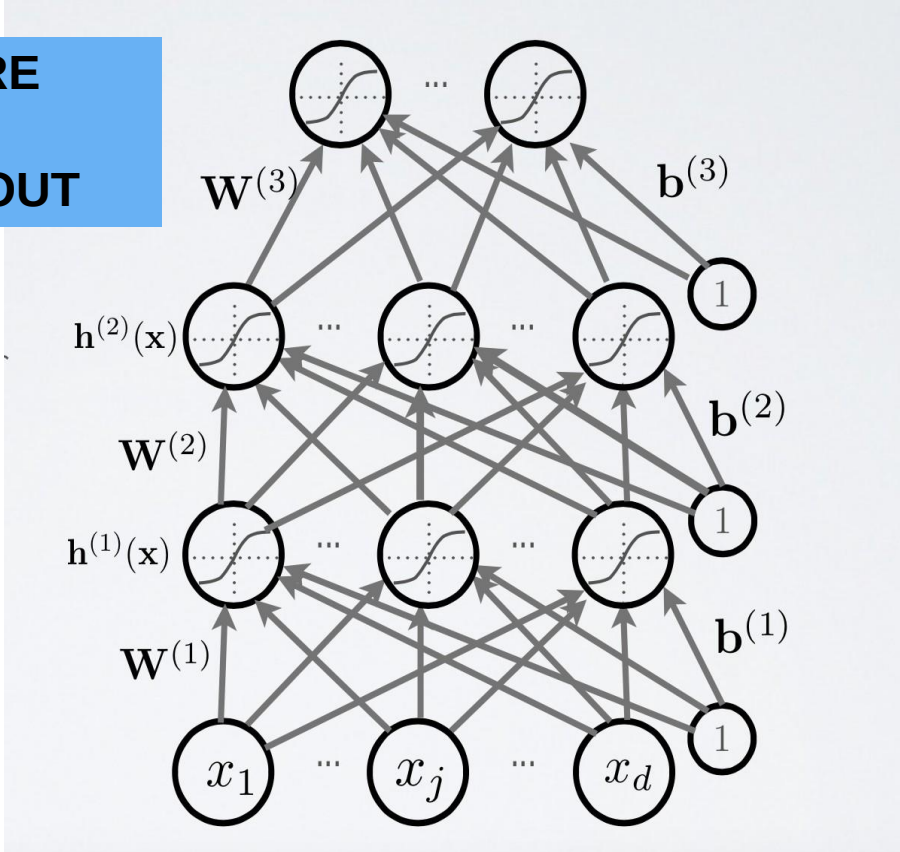
- Weight Decay
- L1/L2 Regularization
- Suitable Model Architectures (depth and width of the layers)
- Unsupervised Pre-training
- Dropout
- Data Augmentation

Dropout

- Cripple the network by removing hidden units stochastically
- In practice, probability of 0.5 works well.

BEFORE

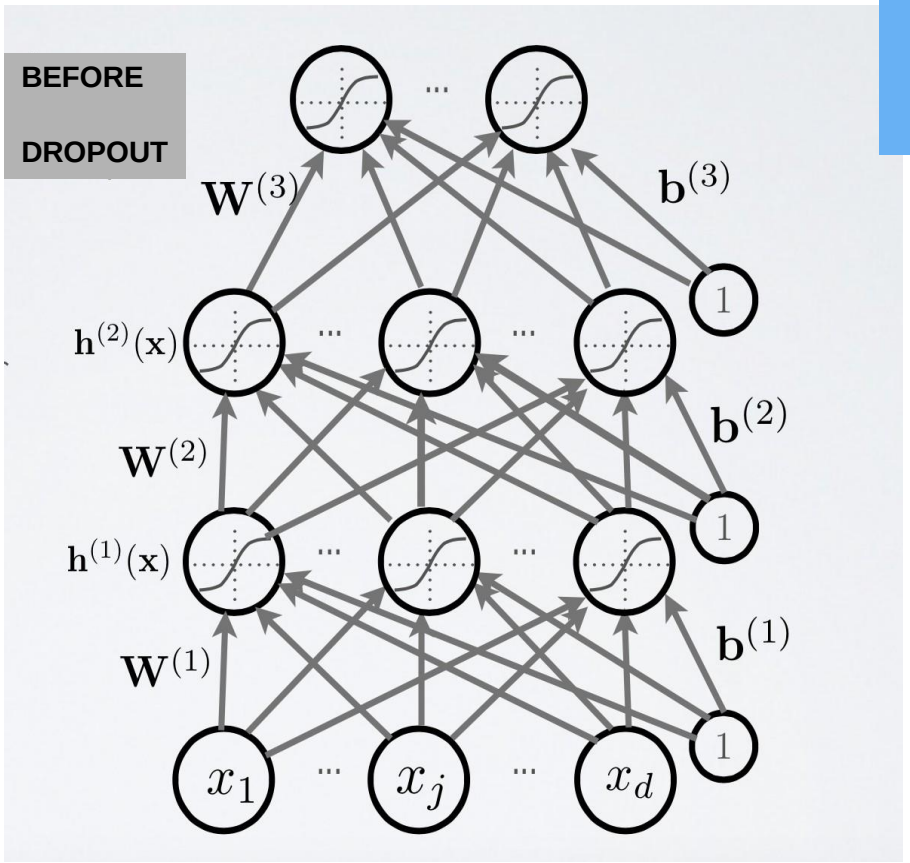
DROPOUT

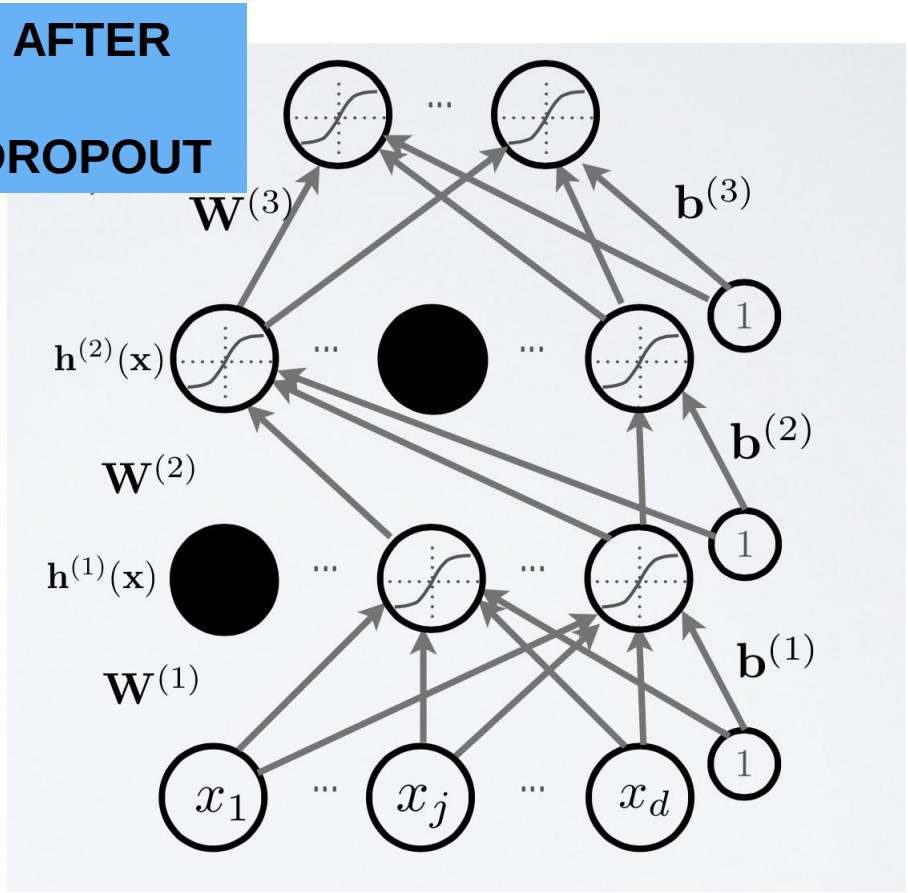


source:http://info.usherbrooke.ca/hlarochelle/neural_networks/content.html

Dropout

- Cripple the network by removing hidden units stochastically
- In practice, probability of 0.5 works well.





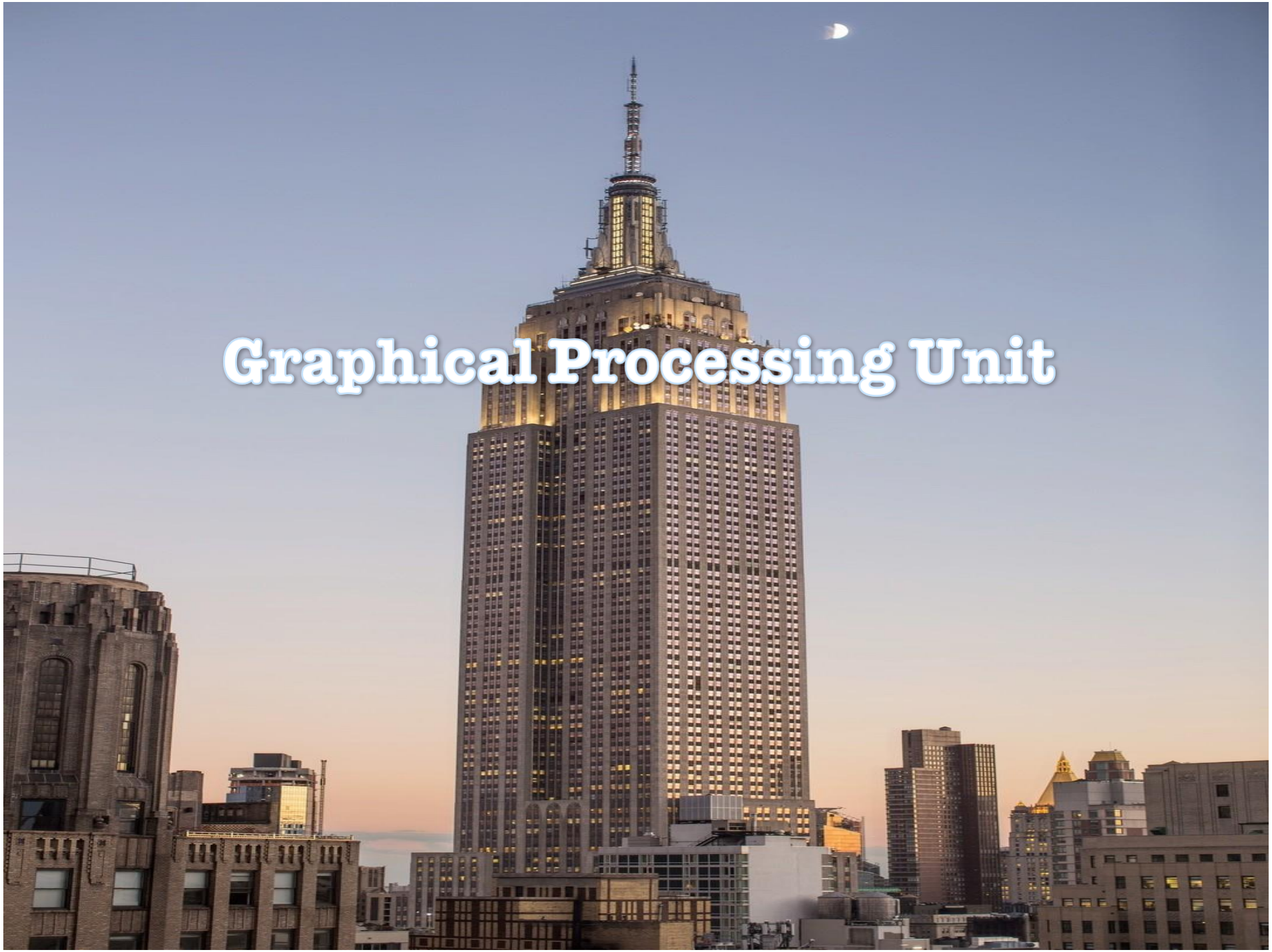
source:http://info.usherbrooke.ca/hlarochelle/neural_networks/content.html

Data Augmentation (for Images)

Some ways to augment data:

- Rotation: random angle between 0° and 360°
- Translation: random shift between -10 and 10 pixels
- Rescaling: random scale factor between 1/1.6 and 1.6
- Flipping: yes or no
- Shearing: random angle between -20° and 20°
- Stretching: random with stretch factor between 1/1.3 and 1.3

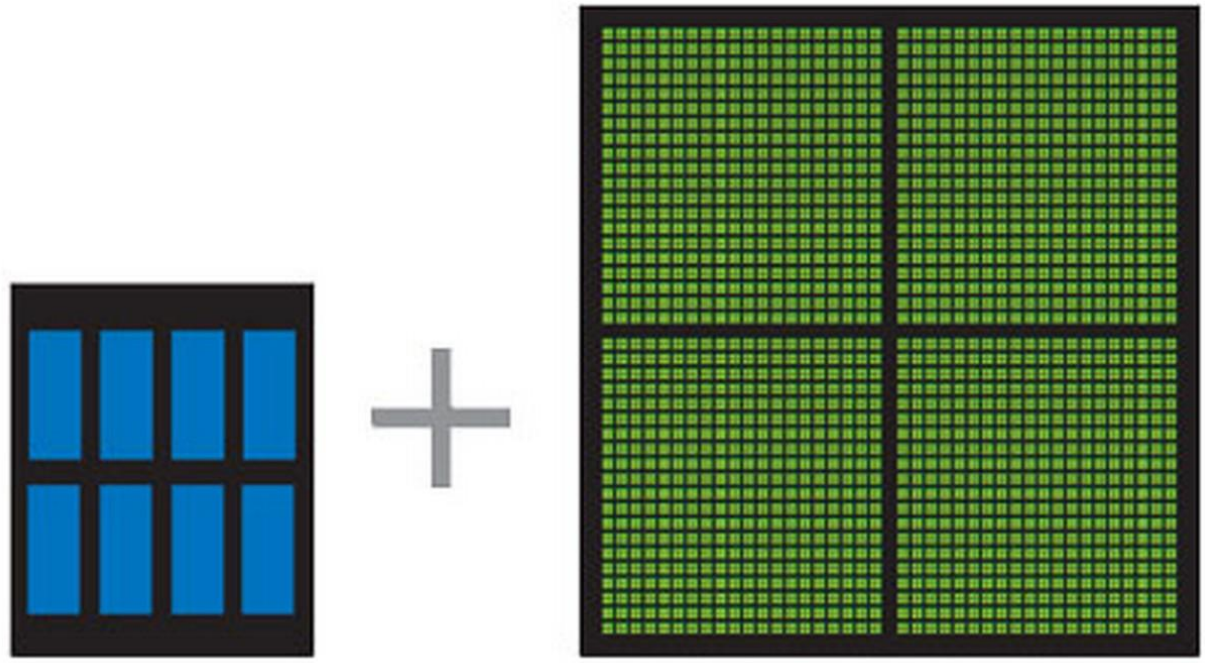

HANDS-ON


Graphical Processing Unit

Impact of GPU

Compared to CPUs 20x speedups are typical

GPUs have thousands of cores to process parallel workloads efficiently

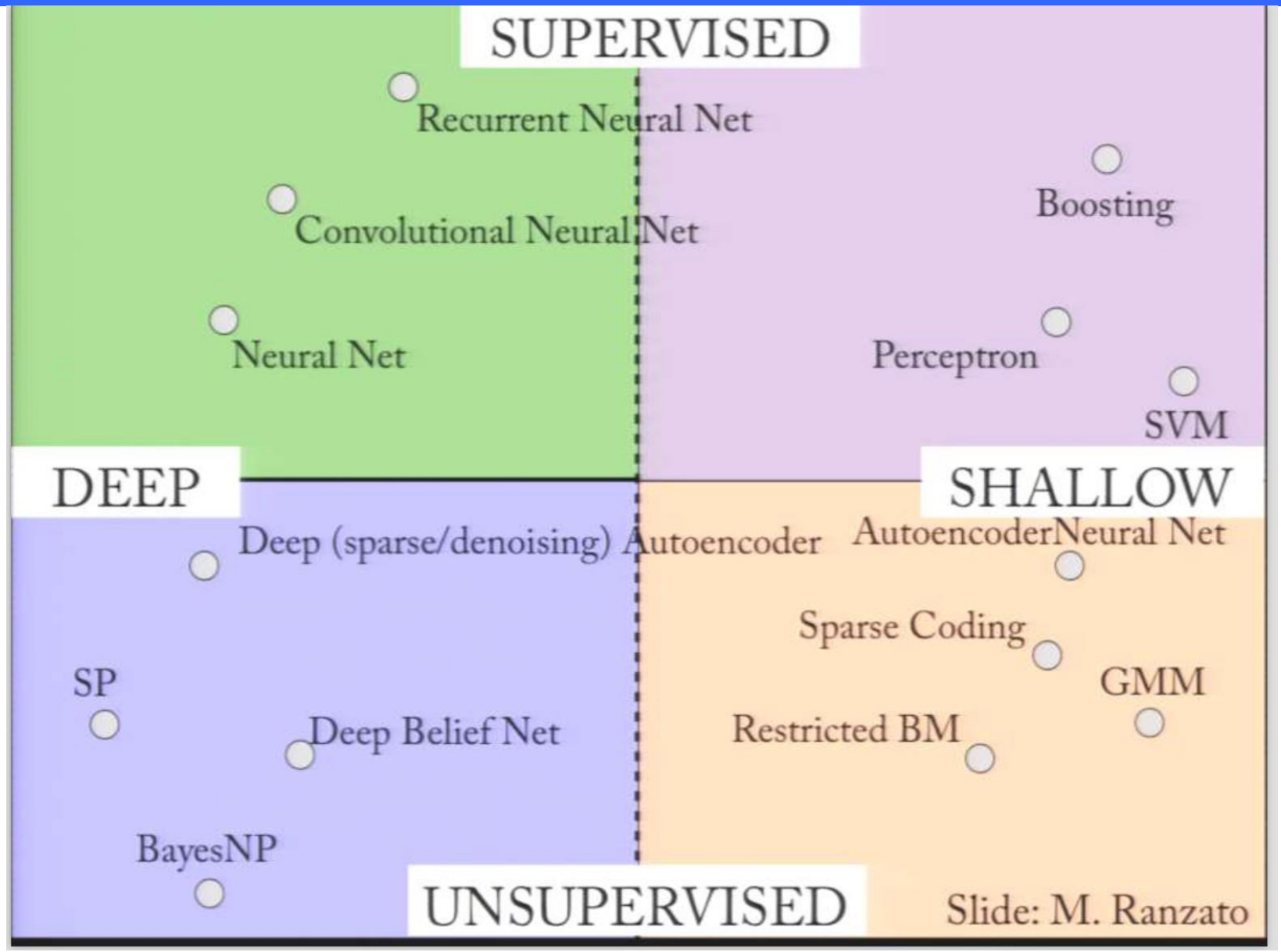


Source: http://www.nvidia.com/object/what-is-gpu-computing.html

Impact of GPU

- Accelerated computations on float32 data
- Matrix multiplication, convolution, and large element-wise operations can be accelerated a lot (5-50x)
- Difficult to parallelize dense neural networks on multiple GPU efficiently (Active area of research)
- CNN – unlike dense neural networks – can be run very efficiently on multiple GPUs. Their use of weight sharing makes data parallelism very efficient
- Copying of large quantities of data to and from a device is relatively slow.
- CUDA has released cuDNN.

Summary



Source: NIPS 2013 Tutorial

Thank you !