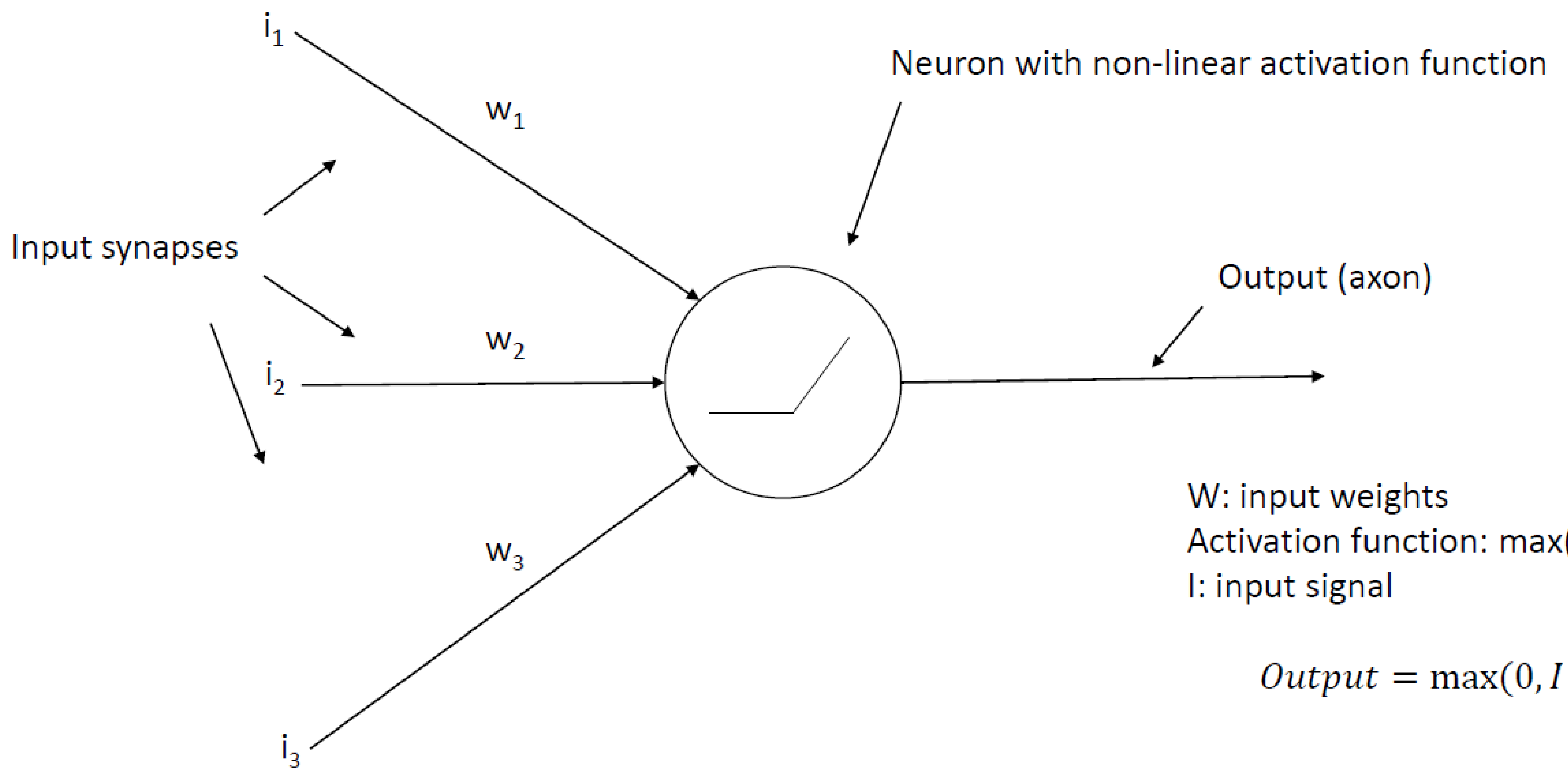
Introduction to Deep Learning

Prepared by Diana Inkpen, University of Ottawa, 2021

Some slides from Mikolov 2014 Tutorial

Artificial Neuron

Neuron (perceptron)



W: input weights
Activation function: max(0, value)
I: input signal

$$Output = \max(0, I \cdot W)$$

Activation Function

- In the previous example, we used max(0, value), the “rectified activation function”
- Other functions can be used, such as: sigmoid, tanh
- Sigmoid: $s(x) = 1/(1+e^{-x})$

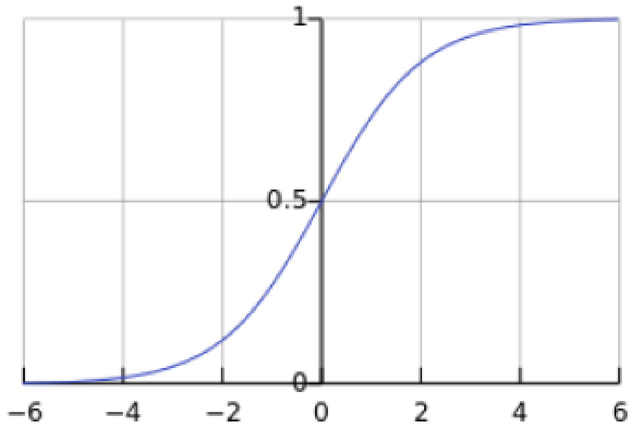
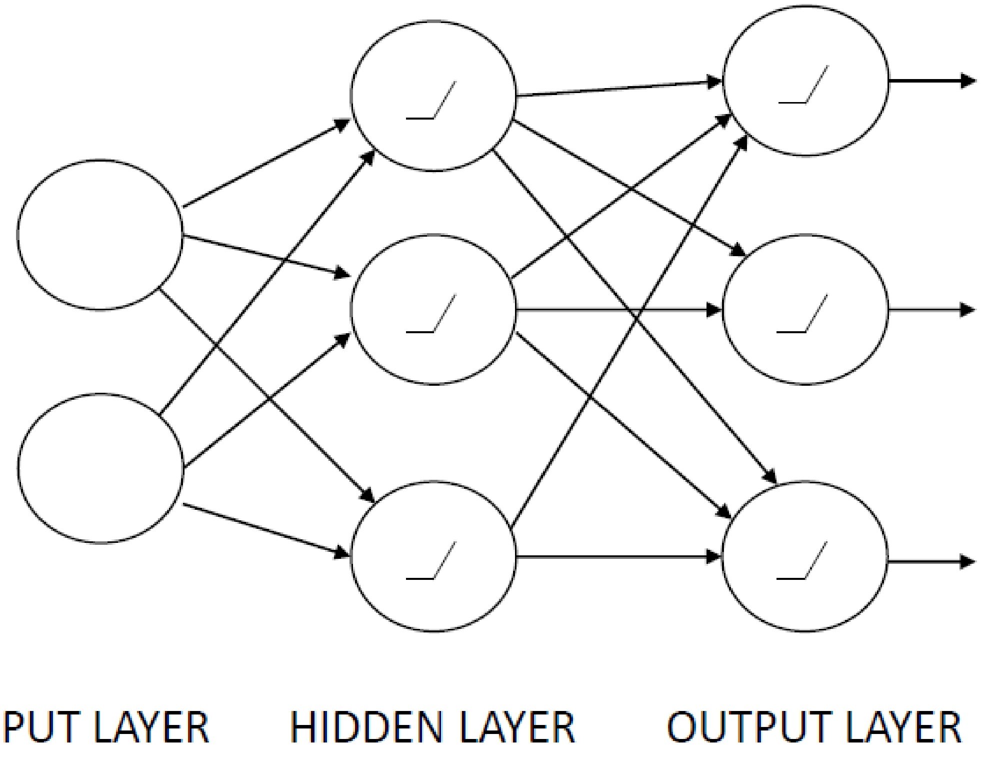


Figure from Wikipedia

Neural Network



Hidden Layers

- A neural network with one hidden layer is universal approximator: it can represent any function.
- But not all functions can be represented efficiently with a single hidden layer. Several layers are needed: deep learning.

Objective function

- The objective function defines how well the neural network performs the task.
- The goal of training is to adapt the weights so that the objective function is maximized / minimized.
- Example: classification accuracy, reconstruction error.

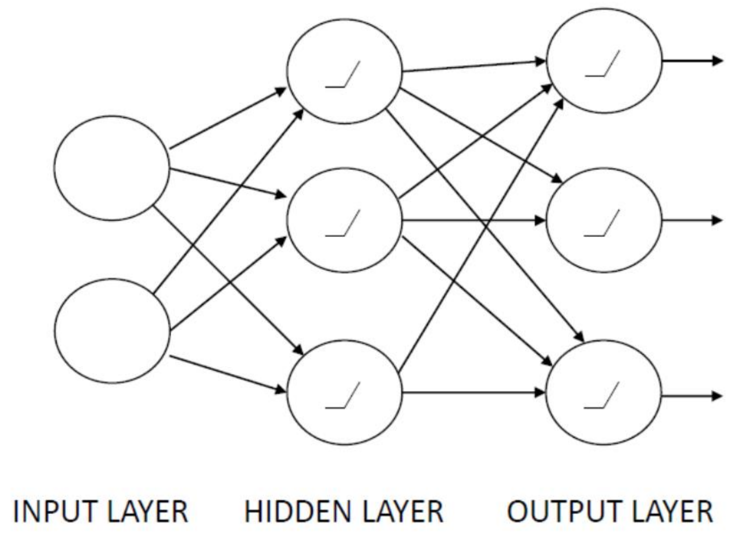
Unsupervised/supervised training

- When the goal is to model the input data, the training is called unsupervised.
- An example is auto-encoder: the objective function is to reconstruct the input data at the output layer (by performing some kind of compression when going through the hidden layers).
- Supervised training usually means that we have additional labels for the input vectors, and the goal is to perform classification.
- Stop training when performance on validation data decreases.

Training of Neural Networks

Forward pass:

- Input signal is presented.
- Hidden layer state is computed (vector times matrix operation and non-linear activation).
- Outputs are computed (vectors times matrix operation and usually non-linear activation).

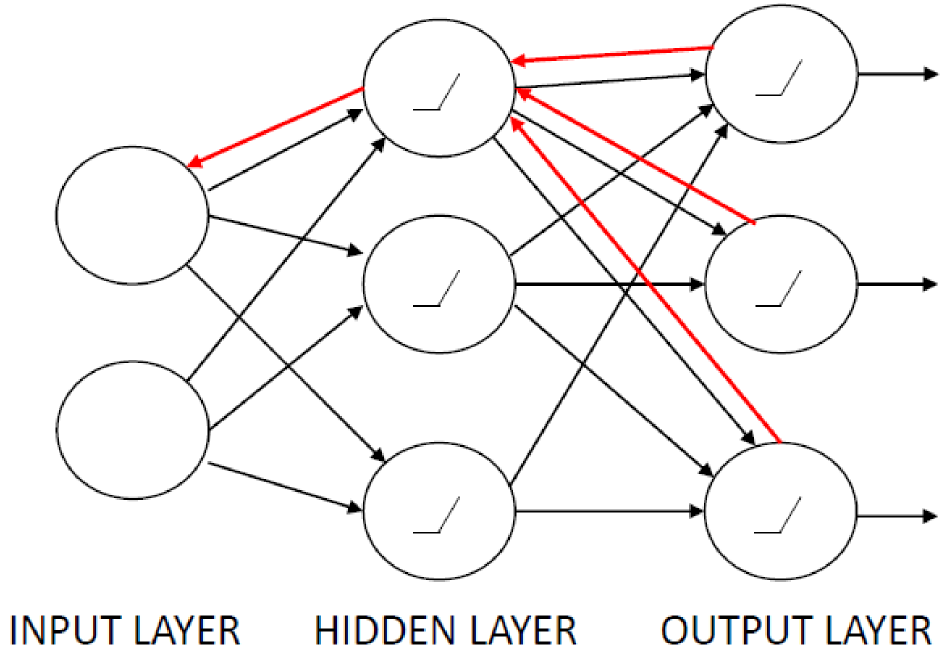


W: input weights
Activation function: max(0, value)
I: input signal

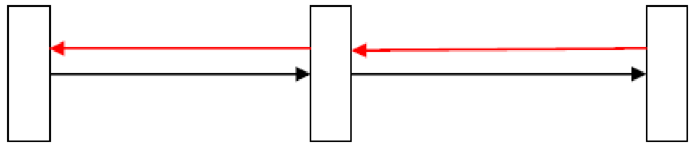
$$Output = \max(0, I \cdot W)$$

Backpropagation

- To train the network, we need to compute gradient of the error.
- The gradients are sent back using the same weights that were used in the forward pass.



Simplified graphical representation:



Stochastic Gradient Descent

- We update the weights after each training example is presented to the network.
- The input feature vector is used to compute the output vector during the forward pass.
- The target vector represents the desired output vector.
- We change the weights a little bit so that next time the same input vector is presented, the output vector will be closer to the target vector.

Training Epochs

- Several training epochs over the training data are often performed
- Usually, the training is finished when performance on held-out (validation) data does not improve
- The starting learning rate and how quickly it gets reduced can affect the resulting performance in a great way: you have to tune this.

Learning Rate

- Learning rate controls how much we change the weights: too little value will result in long training time, too high value will erase previously learned patterns.
- In practice, we start with high learning rate and reduce it during training.

Regularization

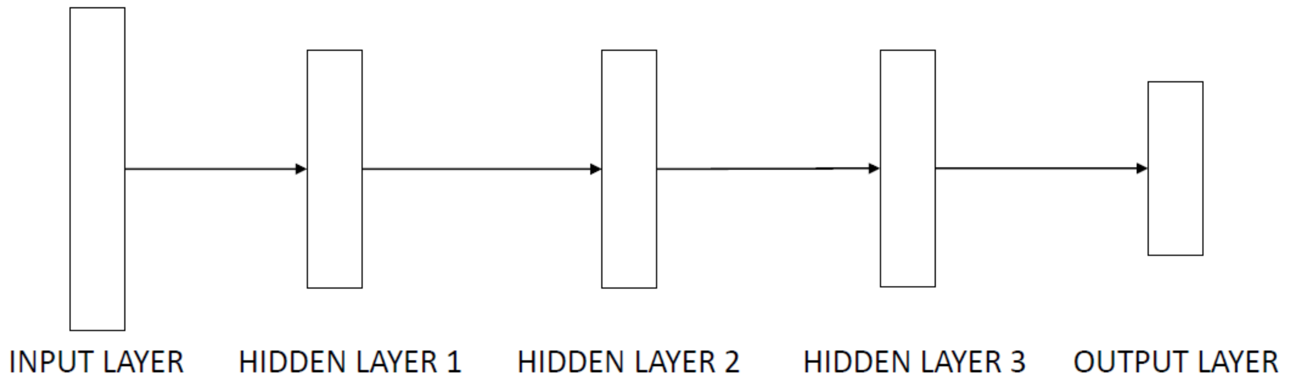
- As the network is trained, it often overfits the training data: it has very good performance during training, but fails to generalize in test data.
- The network “memorizes” the training data: often, it will contain high weights that are used to model only some small subset of data.
- We can try to force the weights to stay small during training to avoid this problem (L1 & L2 regularization).

Summary for training the neural networks

- Stochastic gradient descent and backpropagation are usually good enough.
- The power of neural networks comes from non-linear hidden layers.
- Choice of the hyper-parameters has to be done manually:
    - Type of activation function
    - Choice of architecture (how many hidden layers, their sizes)
    - Learning rate, number of training epochs
    - What features are presented at the input layer
    - How to regularize

Deep Learning

- Deep model architectures have more computational steps (hidden layers) in the model.
- Deep learning aims to learn complex patterns that cannot be learned efficiently with shallow models.

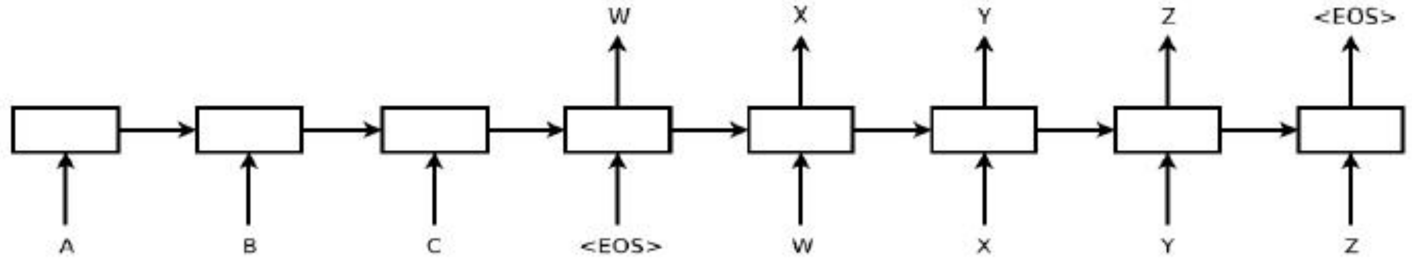


Recurrent Neural Networks (RNN)

- An RNN is a type of artificial neural network where connections between nodes form a directed graph along a temporal sequence.
- Temporal dynamic behavior.
- Backpropagation through time.
- Derived from feedforward neural networks.
- Can use their internal state (memory) to process variable length sequences of inputs.
- Applicable to tasks such as handwriting recognition and speech recognition.

Long short-term memory (LSTM) networks

- LSTM [Hochreiter, '97] has showed to be effective in a wide range of problems.
  - Machine translation [Sutskever, '14; Cho, '14]

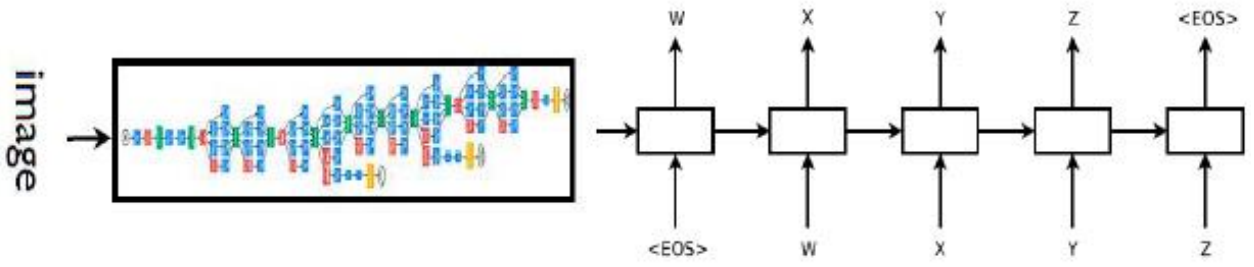


  - Image-to-text conversion [Vinyals, '14]

→

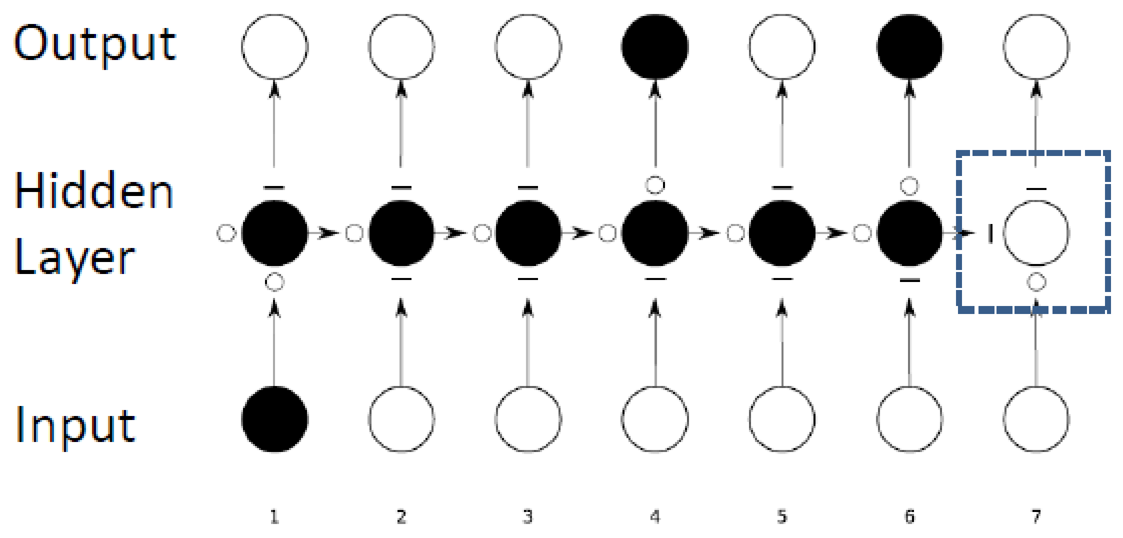
A group of people shopping at an outdoor market.

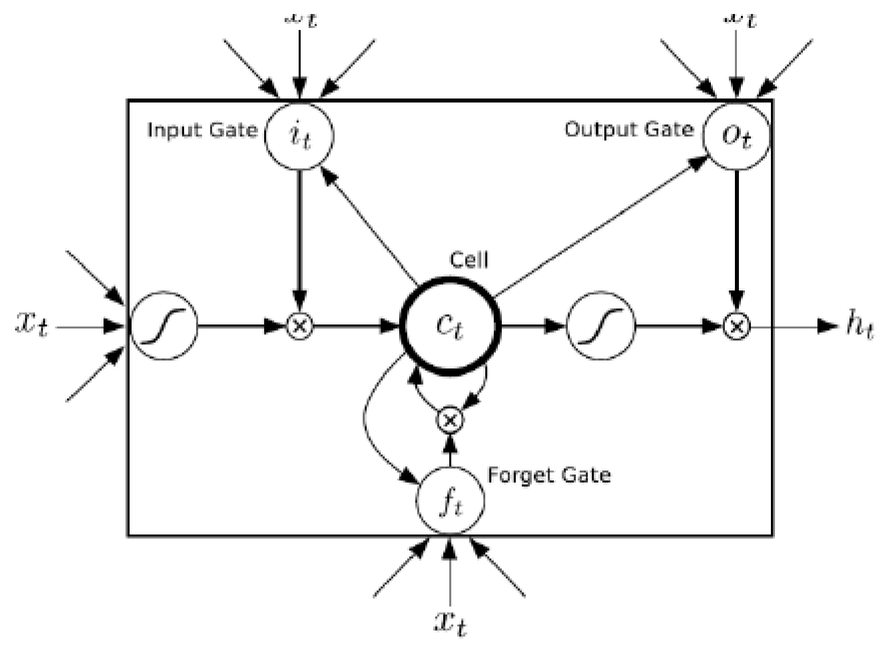
There are many vegetables at the fruit stand.



LSTM are a type of RNN

- Composed of cells that have: an input gate, an output gate and a forget gate.
- The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell.
- Useful for time series data.
- Can model long distance dependencies in text.



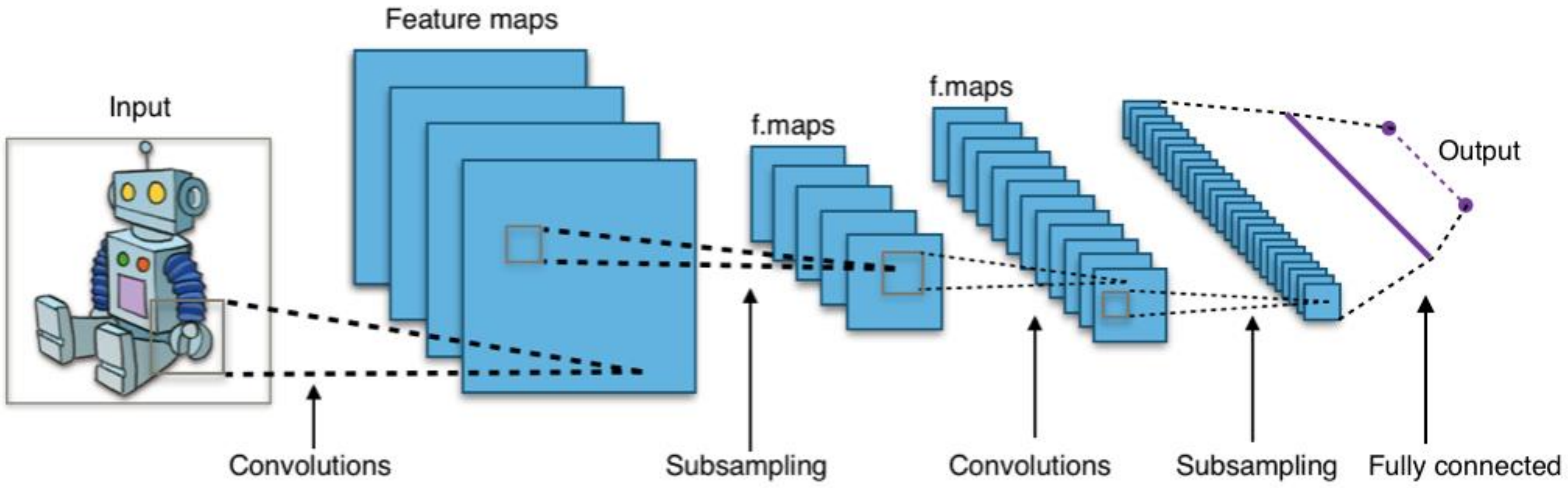


Problems when training RNN/LSTM

- Vanishing gradients:
  - As we propagate the gradients back in time, usually their magnitude decreases, and quickly approaches very small values: this is called vanishing gradient.
  - In practice this means that learning long term dependencies is difficult (LSTMs address this issue).
- Exploding gradients:
  - Sometimes, the gradients start to increase exponentially during backpropagation through the recurrent weights: this is the exploding gradient
  - Large gradients lead to big change of weights, and the network forgets what it has learned so far.
  - Simple solution: clip the values of the gradients.

Convolutional neural networks (CNN)

Developed for image processing (2D convolution filters for parts of the image).
Can be adapted for text classification with 1D convolution filters over windows of consecutive words. Can add linguistic features).



https://en.wikipedia.org/wiki/Convolutional_neural_network

Deep Learning in Image Processing

Large-Scale Visual Recognition Challenge

(1000 classes, 1.2M training images, 150K testing images)

Siberian husky

Eskimo dog

GT: sunscreen
1: hair spray
2: ice lolly
3: sunscreen
4: water bottle
5: lotion

GT: flute
1: flute
2: oboe
3: panpipe
4: trombone
5: bassoon

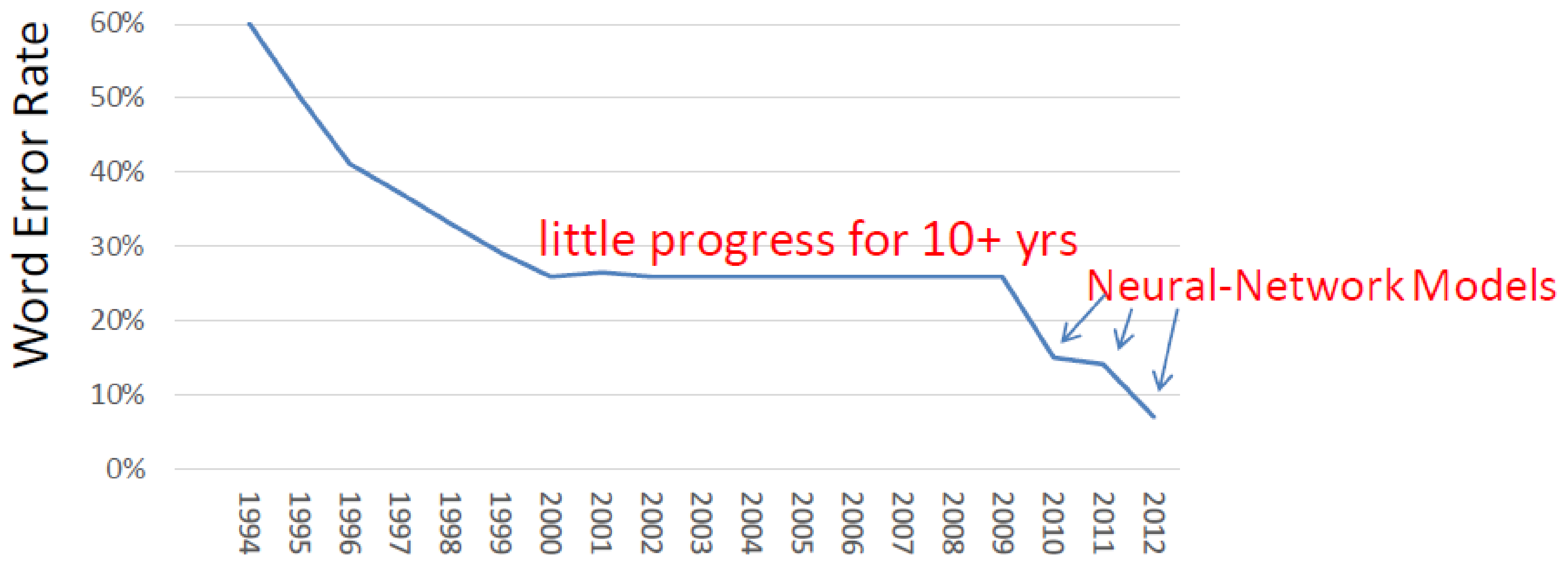
System	Year	Error
SIFT-based	2012	26.2%
SuperVision	2012	**16.4%**
Clarifai	2013	11.7%
GoogLeNet	2014	6.67%
Baidu	2015	5.98%
Microsoft	2015	**4.94%**
Google	2015	**4.90%**

Human: **5.10%**

Applications: automation for vehicles, surveillance or patrolling , image understanding, etc.

Deep Learning in Speech Recognition

Automatic Speech Recognition (speech-to-text)
(Switchboard data)



Brought ASR to more real-life use.

Applications: smart phones/watches, home appliances, cars, speech translation, etc.

Deep Learning for Natural Language Processing
Example: Neural Machine Translation

Translating texts from one language to another

System	Arabic-English	Chinese-English
OpenMT12 – 3rd Place	47.4	30.8
OpenMT12 – 2nd Place	47.5	32.2
OpenMT12 – 1st Place	49.5	32.6
BBN Neural Network Joint Model	52.8	34.7

[1] Evaluation matric: BLEU; larger is better

[2] NRC has implemented the BBN method

More recent work from Univ. of Montreal and Google.

Deep Learning for NLP and IR

- Recent progress in many NLP tasks.
- Multi-task learning helps by sharing layers/linguistic knowledge.
- Improved results for classification tasks via fine-tuning pre-trained models. Example: sentiment classification.
- Progress in word, sentence, and document embedding vectors.
- New neural information retrieval models use sentence and documents embeddings to better match queries to documents.