Neural Networks IV

CMSC 422

SOHEIL FEIZI

sfeizi@cs.umd.edu

Why Neural Networks?

Perceptron

- Proposed by Frank Rosenblatt in 1957
- Real inputs/outputs, threshold activation function

Revival in the 1980's

Backpropagation discovered in 1970's but popularized in 1986

- David E. Rumelhart, Geoffrey E. Hinton, Ronald J. Williams. "Learning representations by back-propagating errors." In Nature, 1986.

MLP is a universal approximator

- Can approximate any non-linear function in theory, given enough neurons, data
- Kurt Hornik, Maxwell Stinchcombe, Halbert White. "Multilayer feedforward networks are universal approximators." Neural Networks, 1989

Generated lots of excitement and applications

http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning/

Neural Networks Applied to Vision

LeNet – vision application

- LeCun, Y; Boser, B; Denker, J; Henderson, D; Howard, R; Hubbard, W; Jackel, L, "Backpropagation Applied to Handwritten Zip Code Recognition," in Neural Computation, 1989
- USPS digit recognition, later check reading

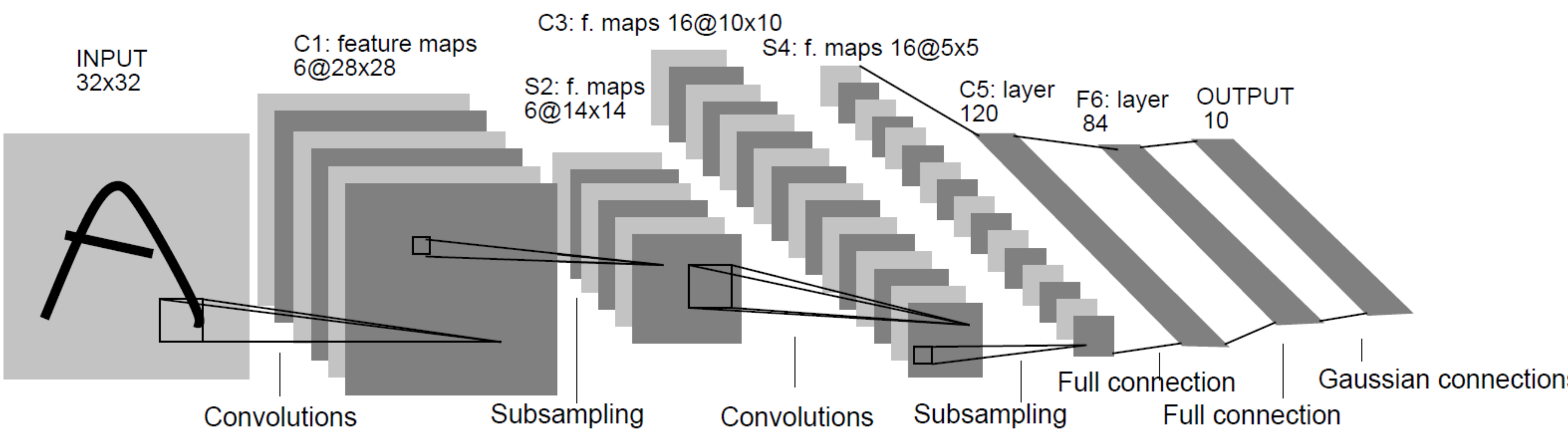


Image credit: LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 1998.

New “winter” and revival in early 2000’s

New “winter” in the early 2000’s due to

- problems with training NNs
- Support Vector Machines (SVMs), Random Forests (RF) – easy to train, nice theory

Revival again by 2011-2012

- Name change (“neural networks” -> “deep learning”)
- + Algorithmic developments that made training somewhat easier
- + Big data + GPU computing
- = performance gains on many tasks (esp Computer Vision)

http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning-part-4/

Big Data

- ImageNet Large Scale Visual Recognition Challenge
  - 1000 categories w/ 1000 images per category
  - 1.2 million training images, 50,000 validation, 150,000 testing

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei. **ImageNet Large Scale Visual Recognition Challenge**. *IJCV,* 2015.

AlexNet Architecture

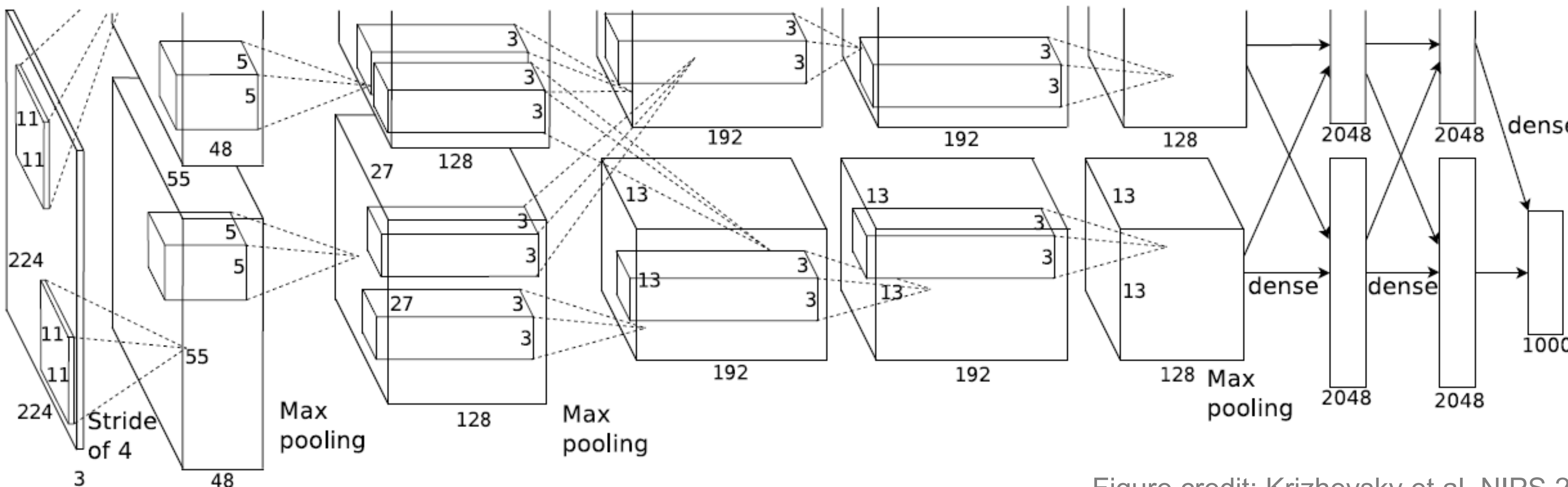


Figure credit: Krizhevsky et al, NIPS 2012.

60 million parameters!

Various tricks

- ReLU nonlinearity
- Overlapping pooling
- Local response normalization
- Dropout – set hidden neuron output to 0 with probability .5
- Data augmentation
- Training on GPUs

GPU Computing

- Big data and big models require lots of computational power

- GPUs
  - thousands of cores for parallel operations
  - multiple GPUs
  - still took about 5-6 days to train AlexNet on two NVIDIA GTX 580 3GB GPUs (much faster today)

Image Classification Performance

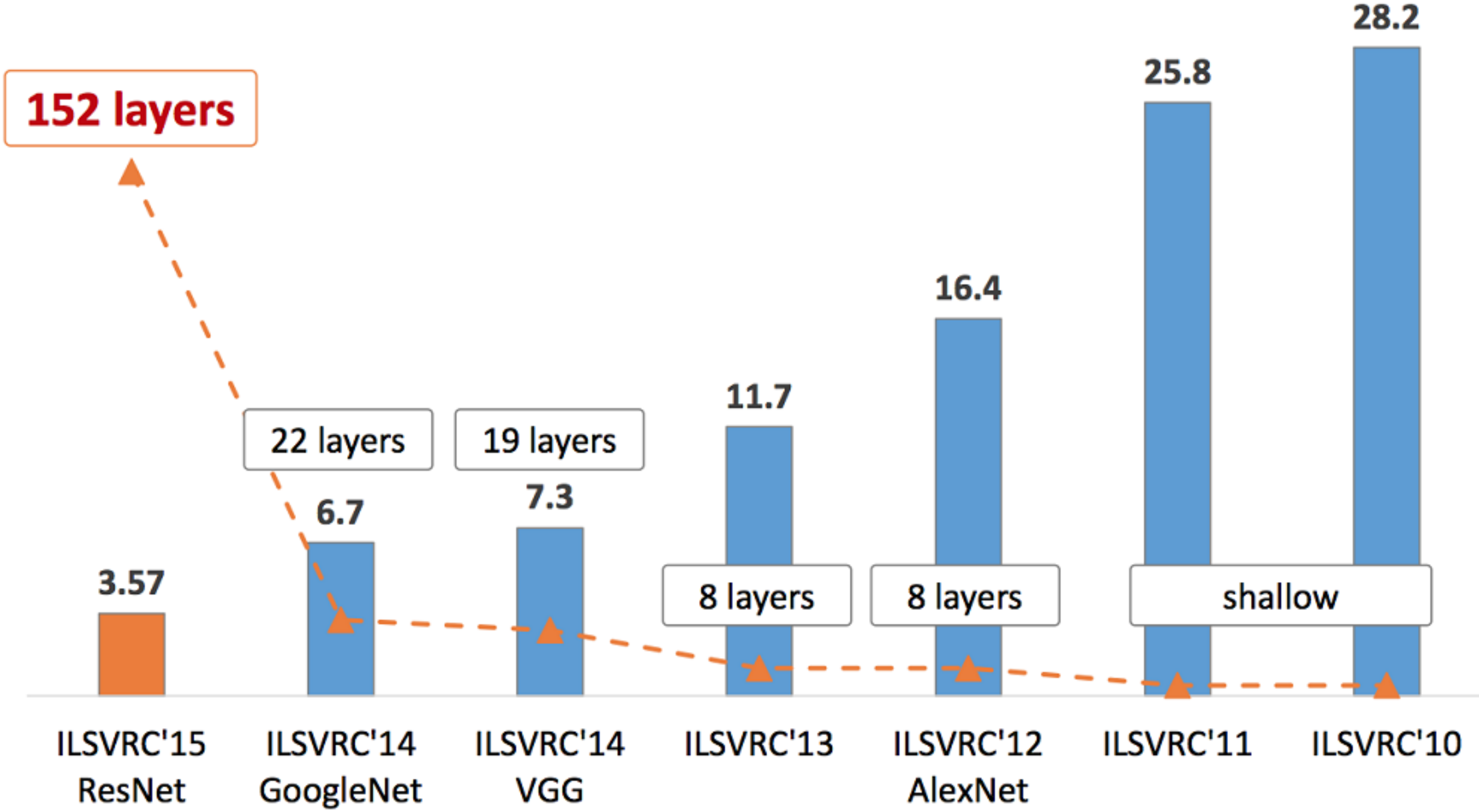


Image Classification Top-5 Errors (%)

Figure from: K. He, X. Zhang, S. Ren, J. Sun. "Deep Residual Learning for Image Recognition". arXiv 2015. (slides)

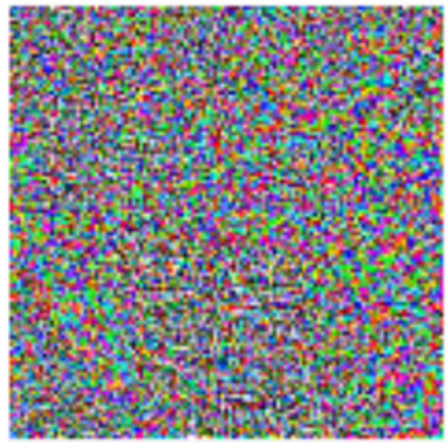
Robustness of Classifiers

"Egyptian Cat"

$\mathbf{x}$

Input Image

$+$



δ

Adversarial Perturbation

$=$

"Traffic Light"

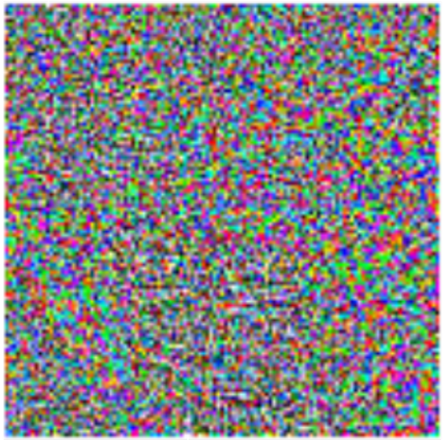
$\tilde{\mathbf{x}}$

Adversarial Example

- Adversarial Examples (Szegedy et al.'14, Biggio et al.'13, Goodfellow et al.'14)

Additive L_p Attack Threat

"Egyptian Cat" "Traffic Light"

 $+$  $=$ 

$\mathbf{x}$ δ $\tilde{\mathbf{x}}$

Optimization:

$$\max_{\delta} \ \ell_{cls}\left(f_\theta(\mathbf{x}+\delta), y\right)$$

$$\delta \in \boldsymbol{\Delta} := \{\delta \in \mathbb{R}^n : \|\delta\|_p \leq \rho\}$$

Solve using Projected Gradient Descent (Madry et al.' 17, Goodfellow et al.'15, Carlini & Wagner '16)

Adversarial Training

- Standard ERM training:

$$\min_{\theta} \; \mathbb{E}_{(\mathbf{x},y)} \left[\ell_{cls}\left(f_\theta(\mathbf{x}), y\right)\right]$$

- Adversarial training for additive attacks (Madry et al.'17):

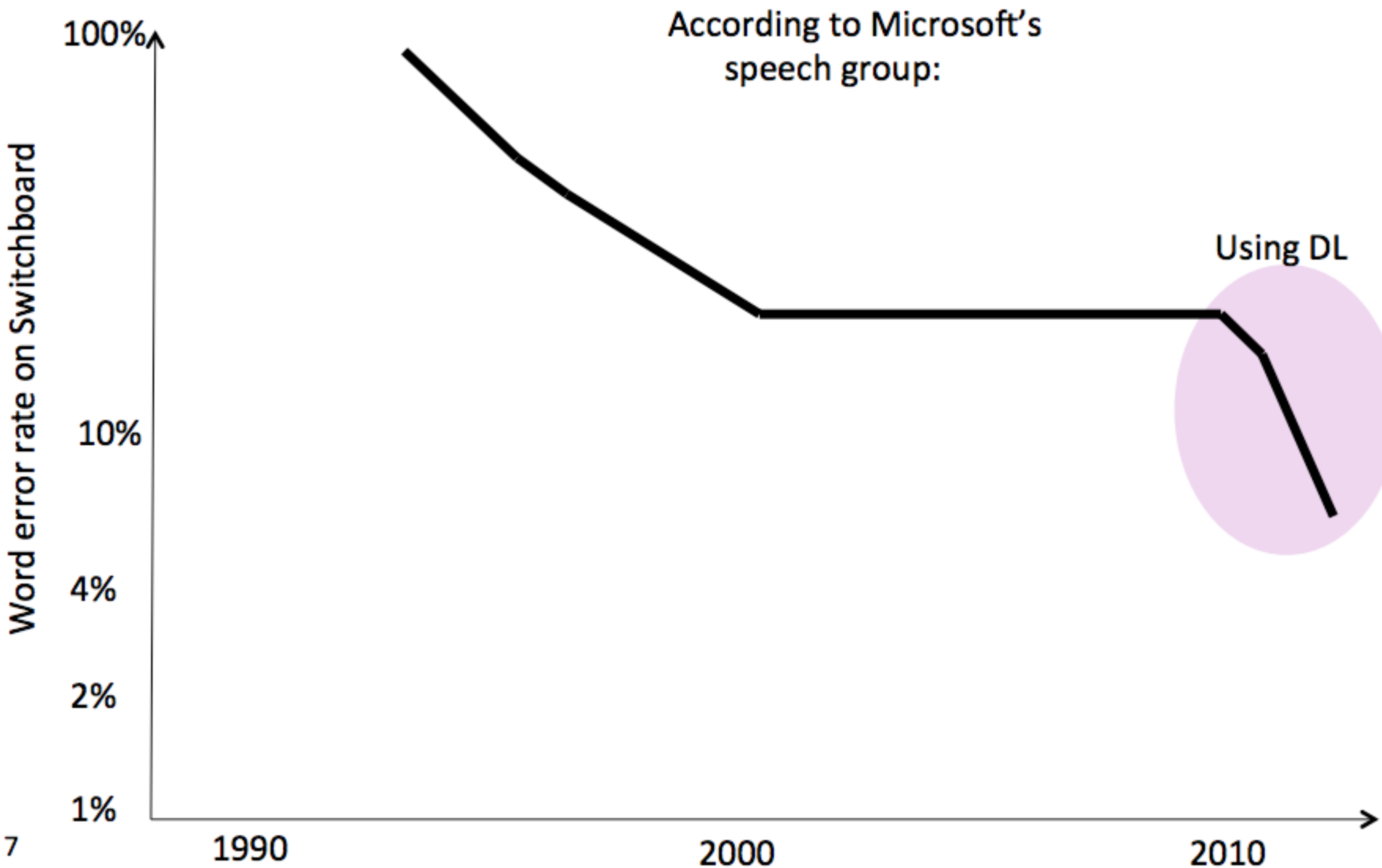
$$\min_{\theta} \; \mathbb{E}_{(\mathbf{x},y)} \left[\max_{\delta} \ell_{cls}\left(f_\theta(\mathbf{x}+\delta), y\right)\right]$$

$$\delta \in \mathbf{\Delta} := \{\delta \in \mathbb{R}^n : \|\delta\|_p \leq \rho\}$$

- Solve using alternative SGD+PGD


Certifiable Defense Against
Adversarial Attacks

Speech Recognition



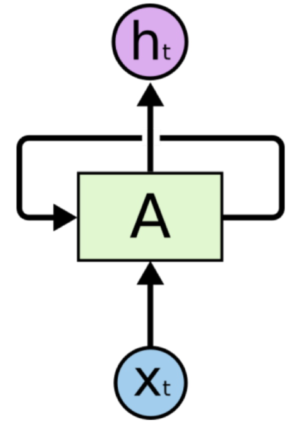
Slide credit: Bohyung Han

Recurrent Neural Networks for Language Modeling

- Speech recognition is difficult due to ambiguity
  - “how to recognize speech”
  - or “how to wreck a nice beach“?

- Language model gives probability of next word given history
  - P(“speech”|”how to recognize”)?

Recurrent Neural Networks



Networks with loops

- The output of a layer is used as input for the same (or lower) layer
- Can model dynamics (e.g. in space or time)

Loops are unrolled

- Now a standard feed-forward network with many layers
- Suffers from vanishing gradient problem
- In theory, can learn long term memory, in practice not (Bengio et al, 1994)

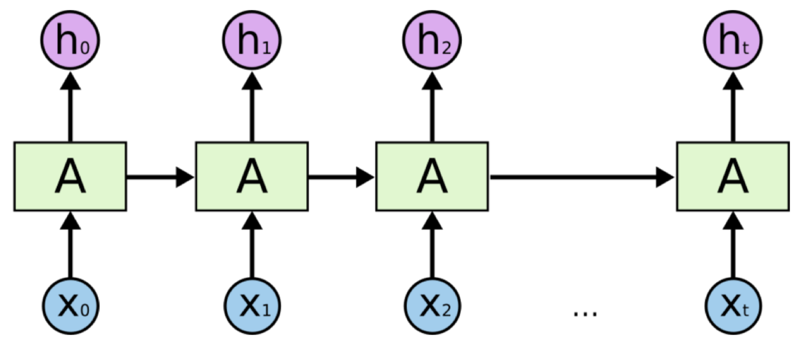


Image credit: Chritopher Olah's blog http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Sepp Hochreiter (1991), Untersuchungen zu dynamischen neuronalen Netzen, Diploma thesis. Institut f. Informatik, Technische Univ. Munich. Advisor: J. Schmidhuber.
Y. Bengio, P. Simard, P. Frasconi. Learning Long-Term Dependencies with Gradient Descent is Difficult. In TNN 1994.

Long Short Term Memory (LSTM)

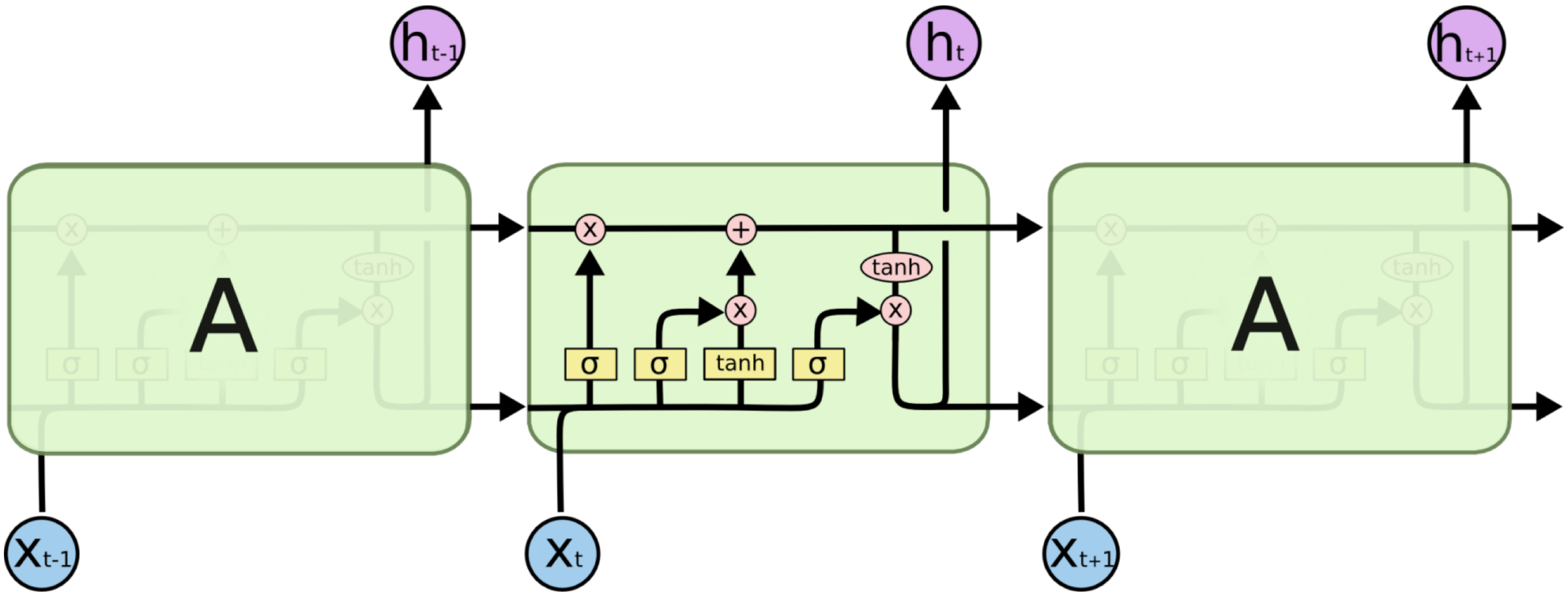


Image credit: Christopher Colah's blog, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

- A type of RNN explicitly designed not to have the vanishing or exploding gradient problem
- Models long-term dependencies
- Memory is propagated and accessed by gates
- Used for speech recognition, language modeling …

Hochreiter, Sepp; and Schmidhuber, Jürgen. "Long Short-Term Memory." Neural Computation, 1997.

Long Short Term Memory (LSTM)

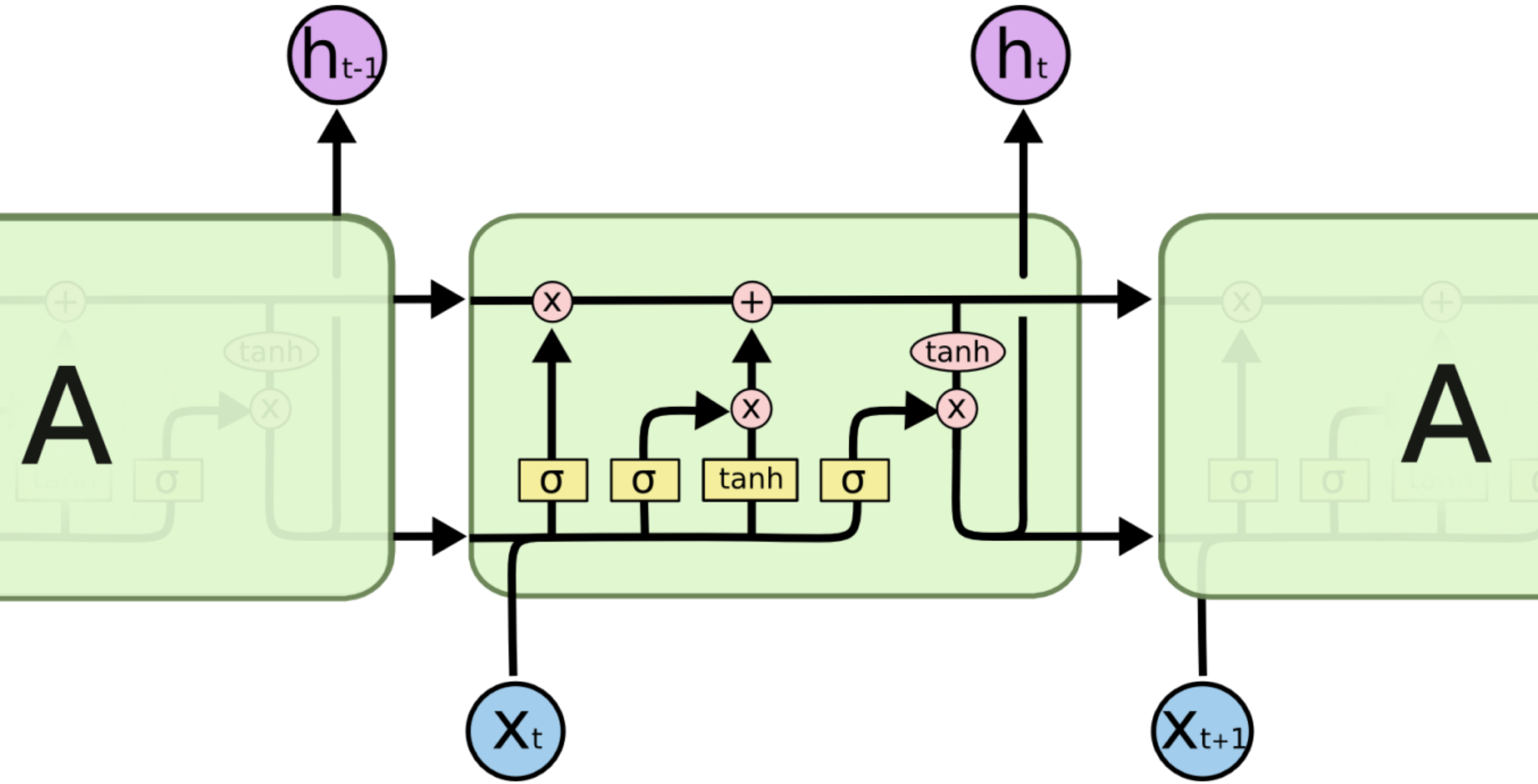


Image credit: Christopher Colah's blog, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

What you should know about deep neural networks

- Why they are difficult to train
  - Initialization
  - Overfitting
  - Vanishing gradient
  - Require large number of training examples

- What can be done about it
  - Improvements to gradient descent
  - Stochastic gradient descent
  - Momentum
  - Weight decay
  - Alternate non-linearities and new architectures

References (& great tutorials) if you want to explore further:

http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning-part-1/

http://cs231n.github.io/neural-networks-1/

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

Keeping things in perspective…

In 1958, the New York Times reported the perceptron to be "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence."