The Deep Learning Revolution

Raymond J. Mooney
University of Texas at Austin

Deep Learning Revolution

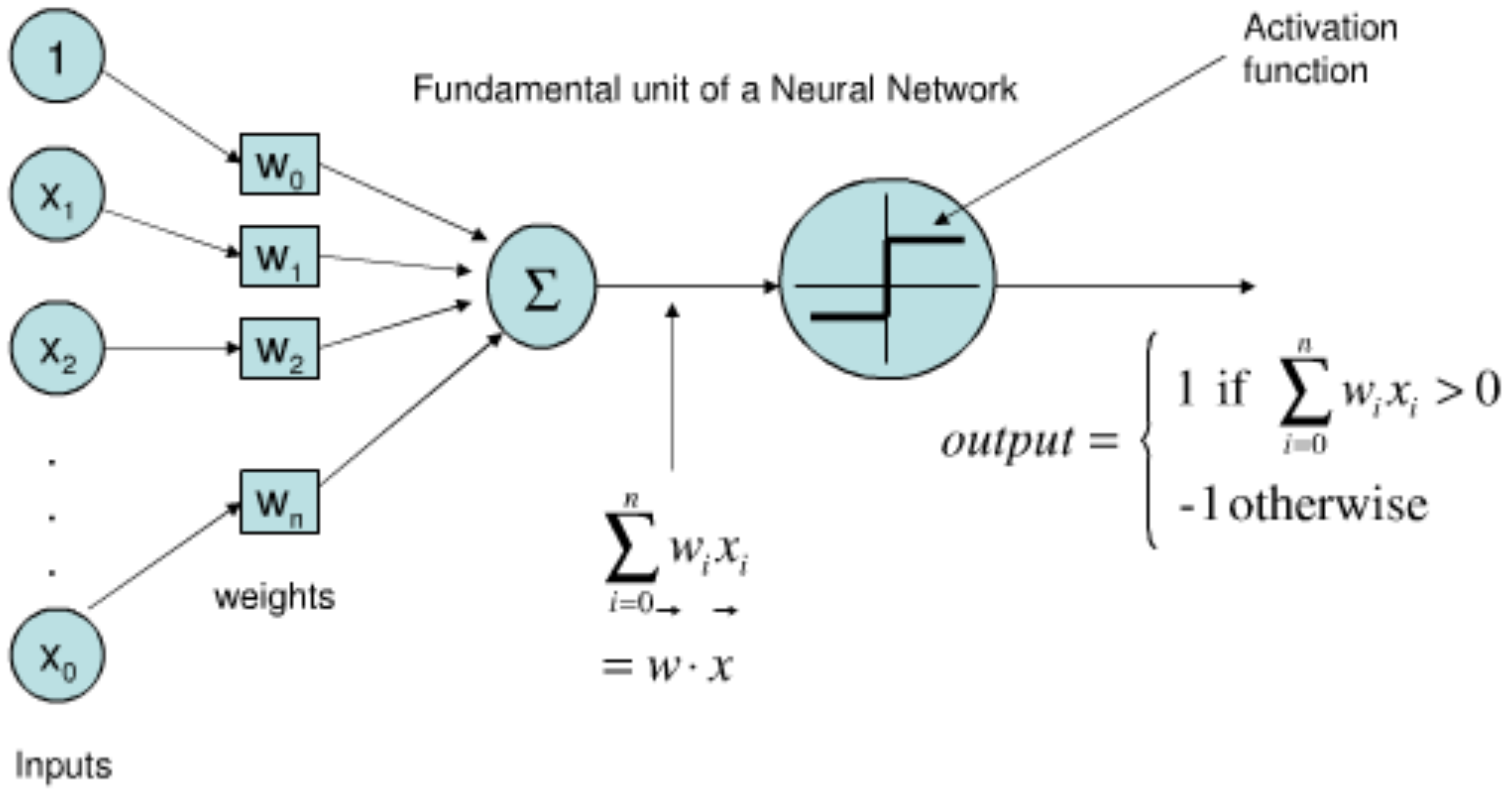
- Recent machine learning methods for training "deep" neural networks (NNs) have demonstrated remarkable progress on many challenging AI problems (e.g. speech recognition, visual object recognition, machine translation, game playing).
- However, their capabilities are prone to "hype."
- Deep learning has not "solved" AI and current methods have clear limitations.

Very Brief History of Machine Learning

- Single-layer neural networks (1957-1969)
- Symbolic AI & knowledge engineering (1970-1985)
- Multi-layer NNs and symbolic learning (1985-1995)
- Statistical (Bayesian) learning and kernel methods (1995-2010)
- Deep learning (CNNs and RNNs) (2010-?)

Single-Layer Neural Network
(Linear Threshold Unit)

- Mathematical model of an individual neuron.



Perceptron

- Rosenblatt (1957) developed an iterative, hill-climbing algorithm for learning the weights of single-layer NN to try to fit a set of training examples.
- Unable to learn or represent many classification functions (e.g. XOR), only the "linearly separable" ones are learnable.

Perceptron Learning Rule

- Update weights by:

$$w_i = w_i + \eta(t - o)x_i$$

  where η is the "learning rate," *t* is the teacher output, and *o* is the network output.
- Equivalent to rules:
  - If output is correct do nothing.
  - If output is high, lower weights on active inputs
  - If output is low, increase weights on active inputs

Perceptron Learning Algorithm

- Iteratively update weights until convergence.

Initialize weights to random values
Until outputs of all training examples are correct
    For each training pair, *E*, do:
        Compute current output *o* for *E* given its inputs
        Compare current output to target value, *t* , for *E*
        Update weights using learning rule

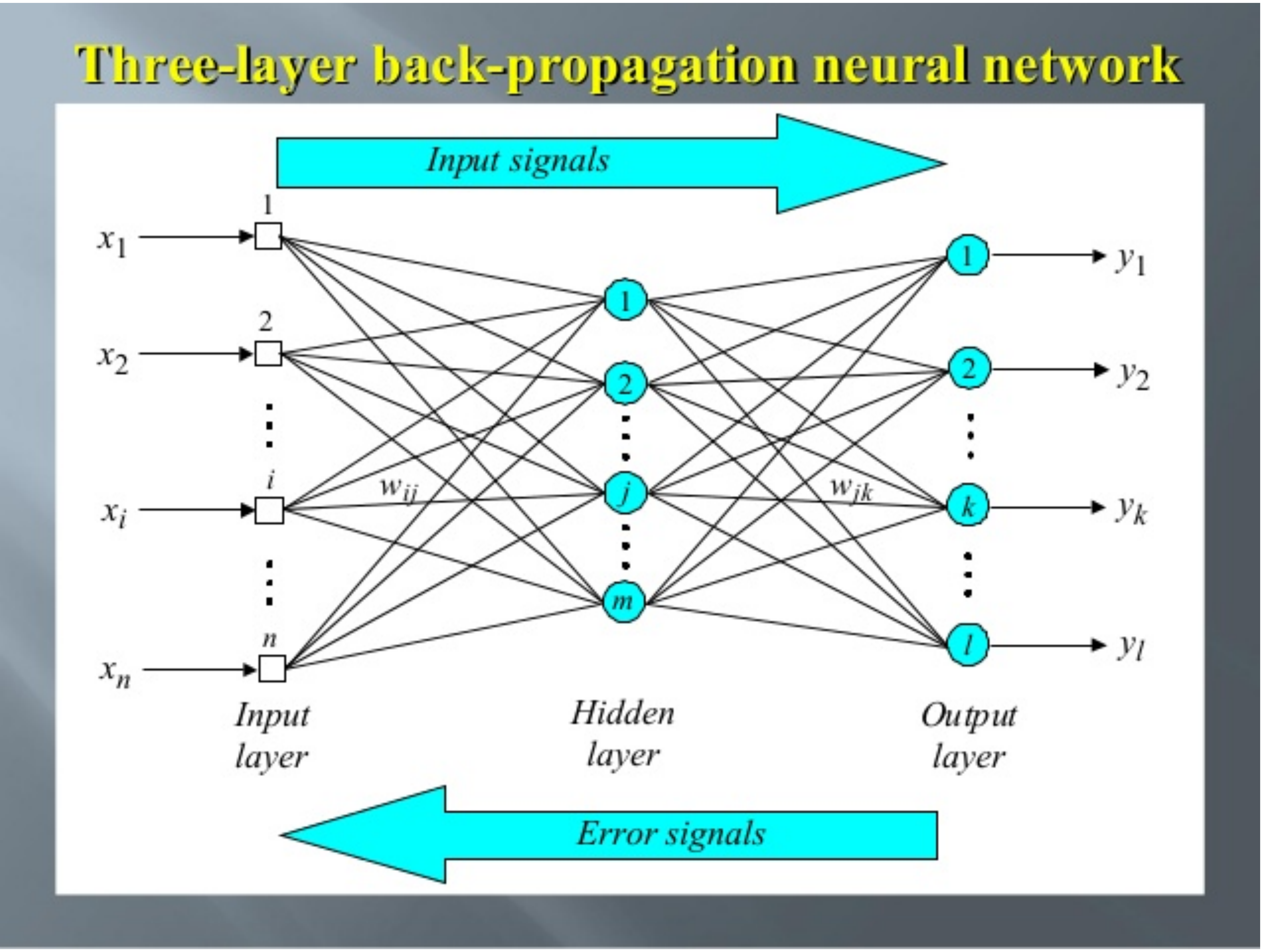
Perceptron Demise

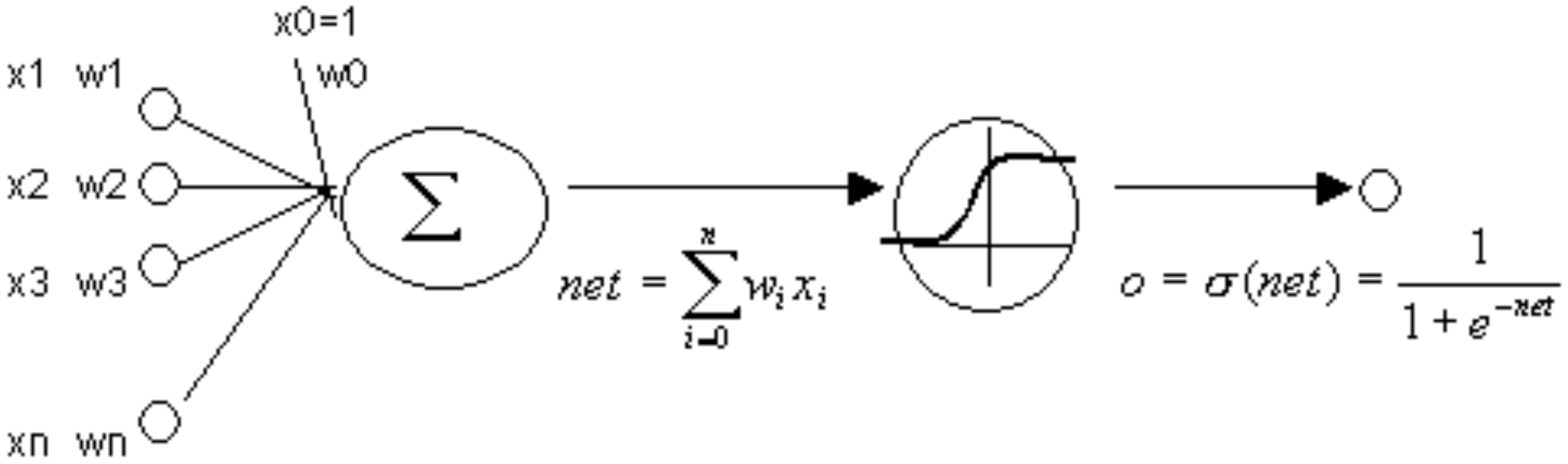
- *Perceptons* (1969) by Minksy and Papert illuminated the limitations of the perceptron.
- Work on neural-networks dissipated during the 70's and early 80's.

Neural Net Resurgence (1986)

- Interest in NNs revived in the mid 1980's due to the rise of "connectionism."
- Backpropagation algorithm popularized for training three-layer NN's.
- Generalized the iterative "hill climbing" method to approximate fitting two layers of synaptic connections, but no convergence guarantees.

3-Layer NN Backpropagation





[The sigmoid threshold unit]

Second NN Demise (1995-2010)

- Generic backpropagation did not generalize that well to training deeper networks.
- Little theoretical justification for underlying methods.
- Machine learning research moved to graphical models and kernel methods.

Deep Learning Revolution (2010…)

- Improved methods developed for training deep neural works.
- Particular successes with:
  - Convolutional neural nets (CNNs) for vision.
  - Recurrent neural nets (RNNs) for machine translation and speech recognition.
  - Deep reinforcement learning for game playing.

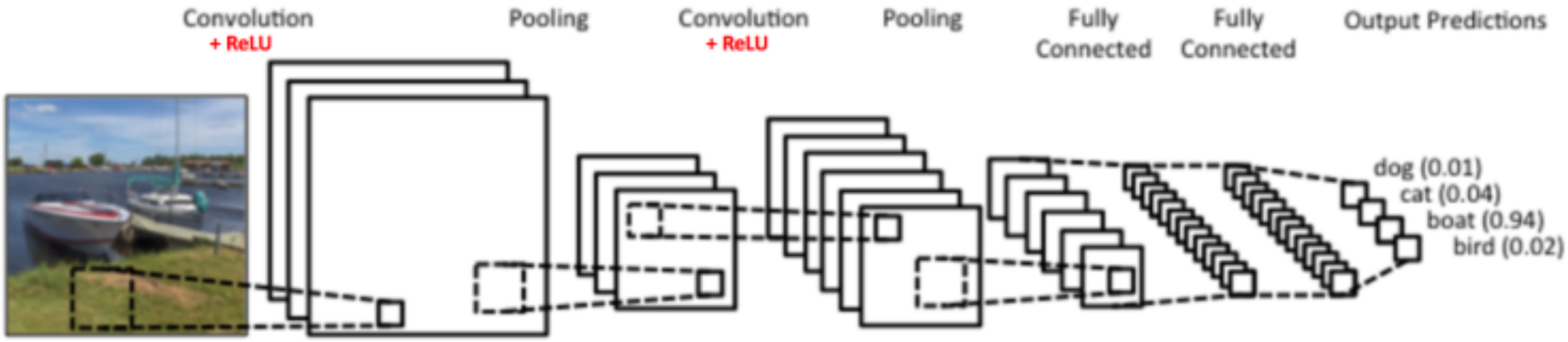
Massive Data and Specialized Hardware

- Large collections of supervised (crowdsourced) training data has been critical.
- Efficient processing of this big data using specialized hardware (Graphics Processing Units, GPUs) has been critical.

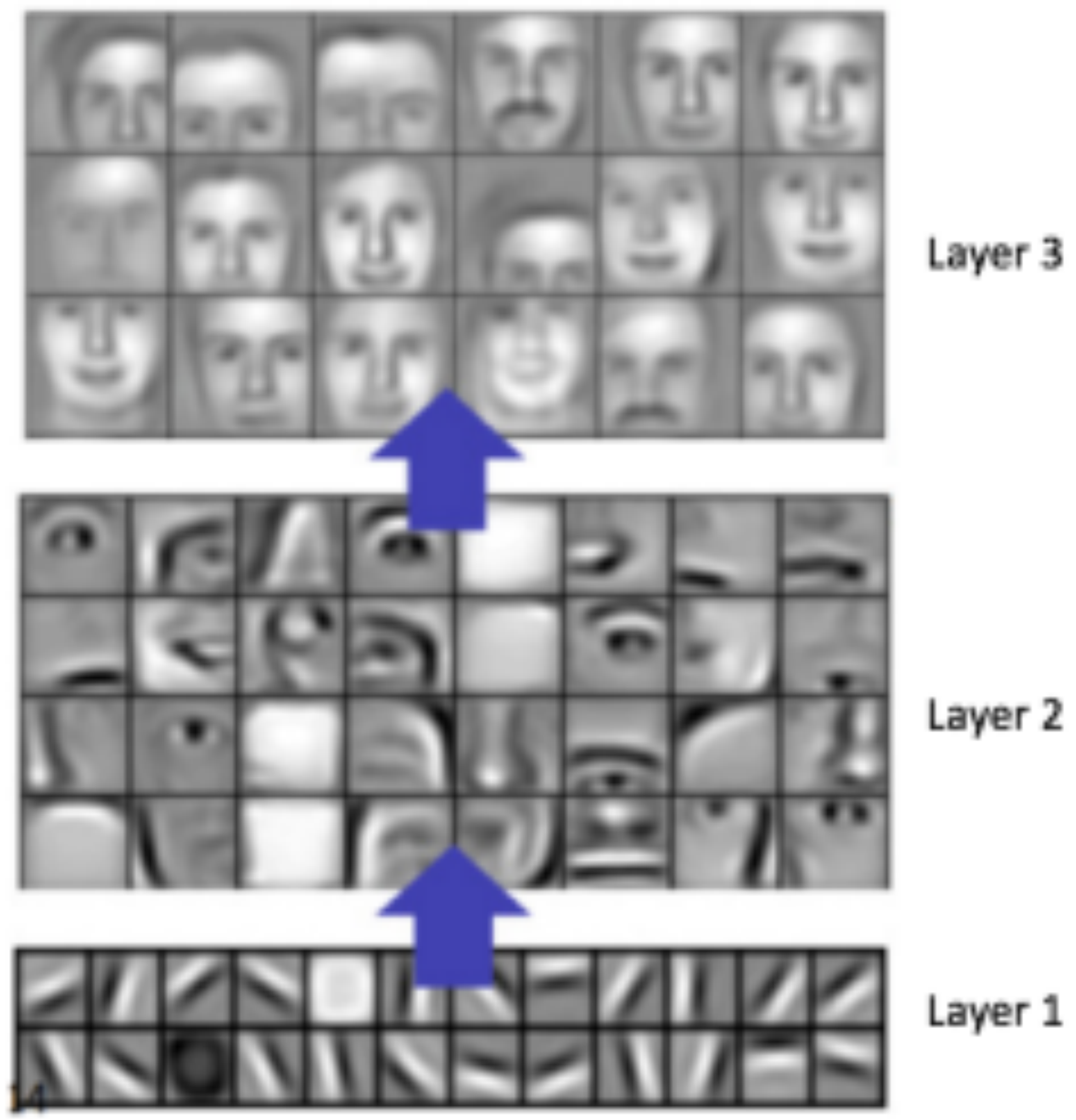
CNNs

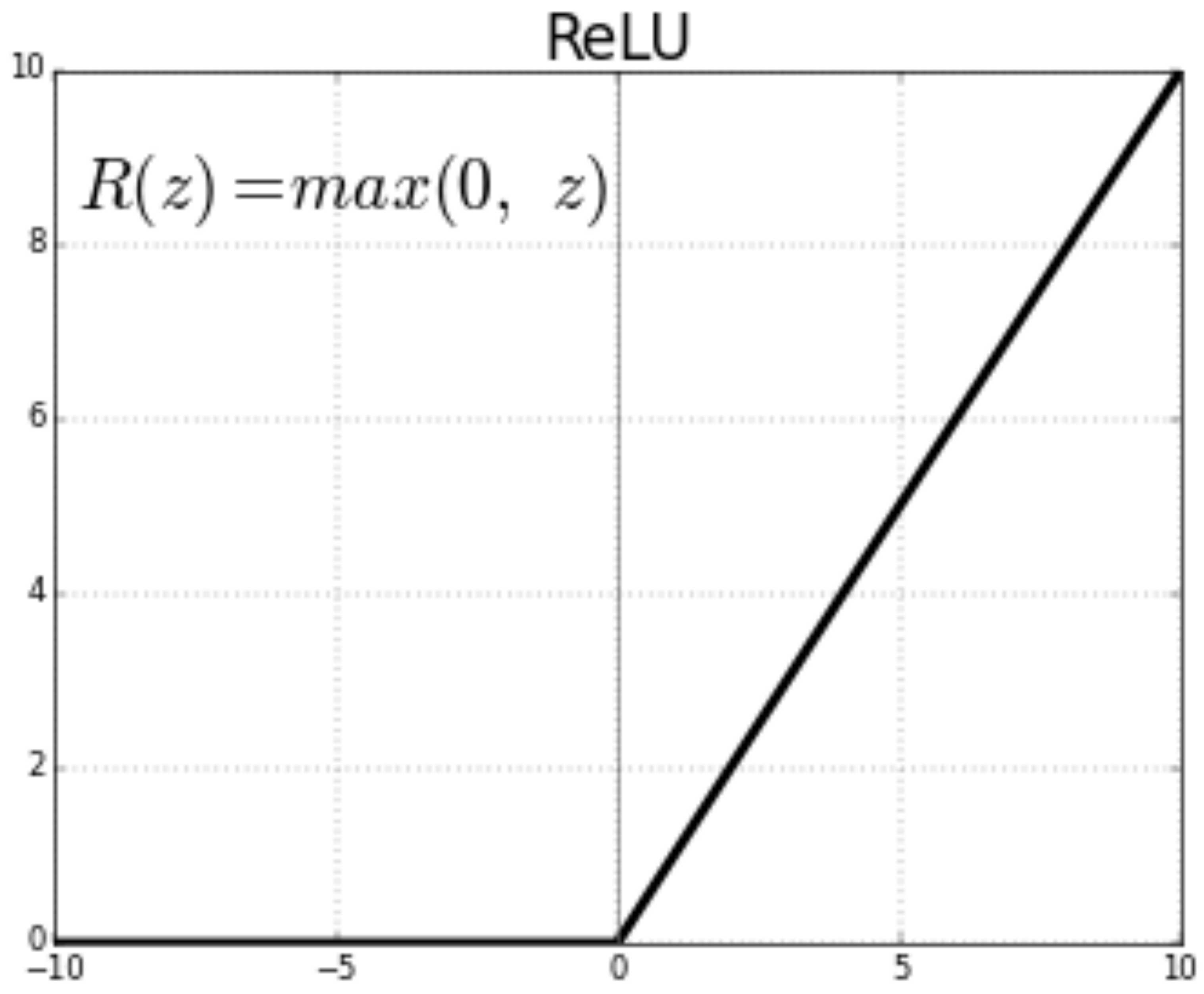
- Convolutional layers learn to extract local features from image regions (receptive fields) analogous to human vision (LeCun, et al., 1998).
- Deeper layers extract higher-level features.
- Pool activity of multiple neurons into one at the next layer using max or mean.
- Nonlinear processing with Rectified Linear Units (ReLUs)
- Decision made using final fully connected layers.

CNNs



Increasingly broader local features extracted from image regions





ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- Recognize 1,000 categories of objects in 150K test images (given 1.2M training images).

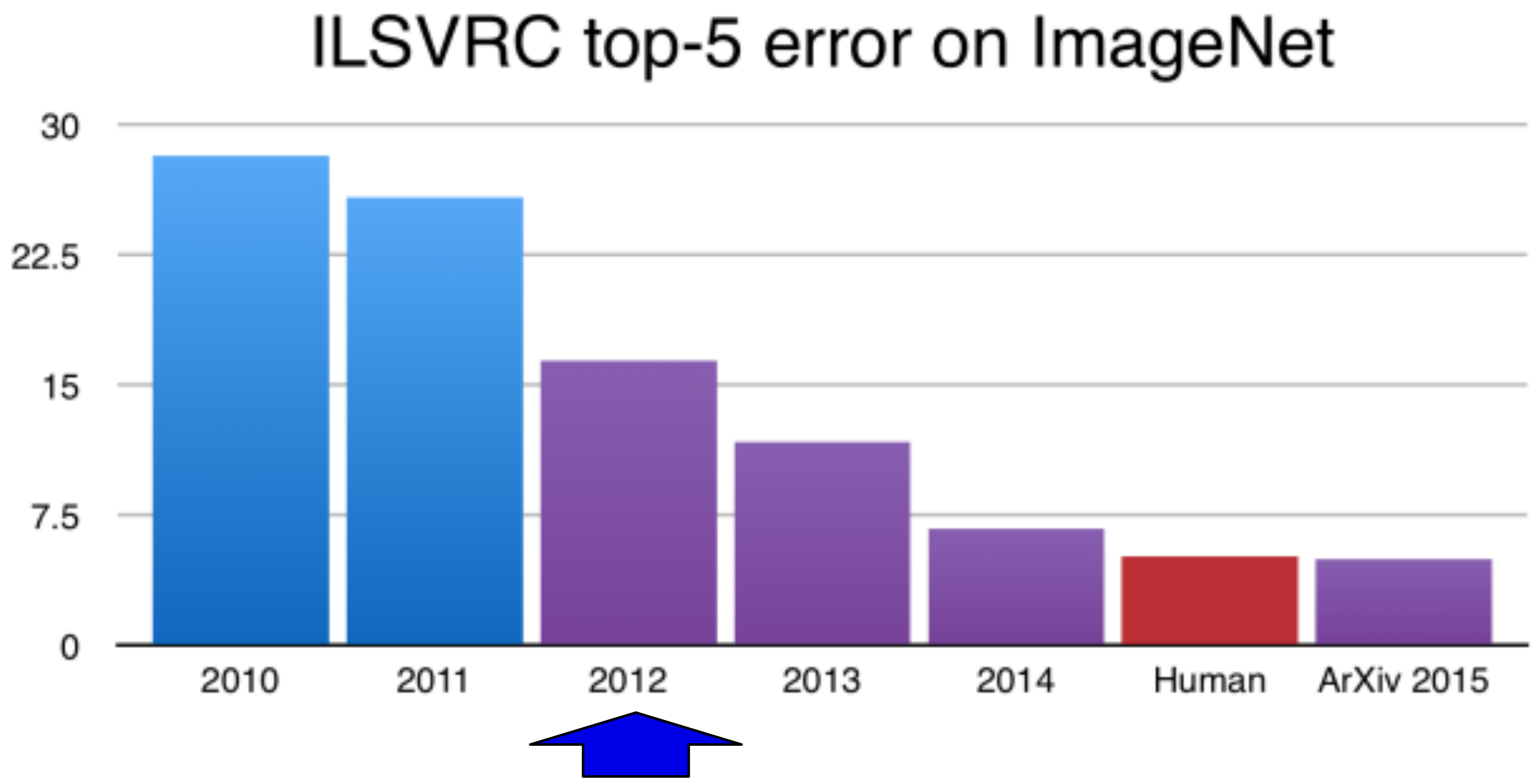
Mongoose

Canoe

Missile

Trombone

ImageNet Performance Over Time



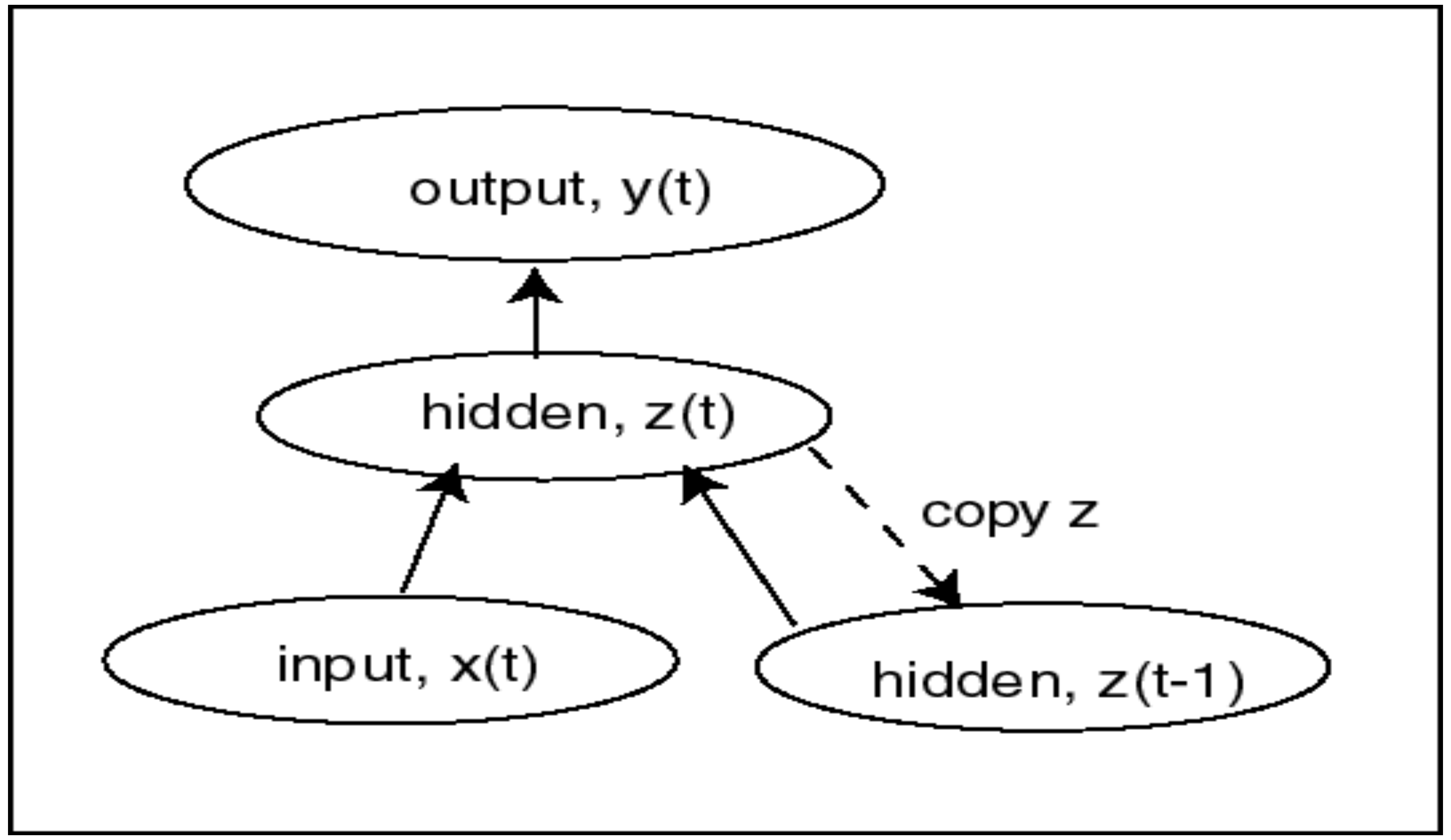
CNNs
introduced

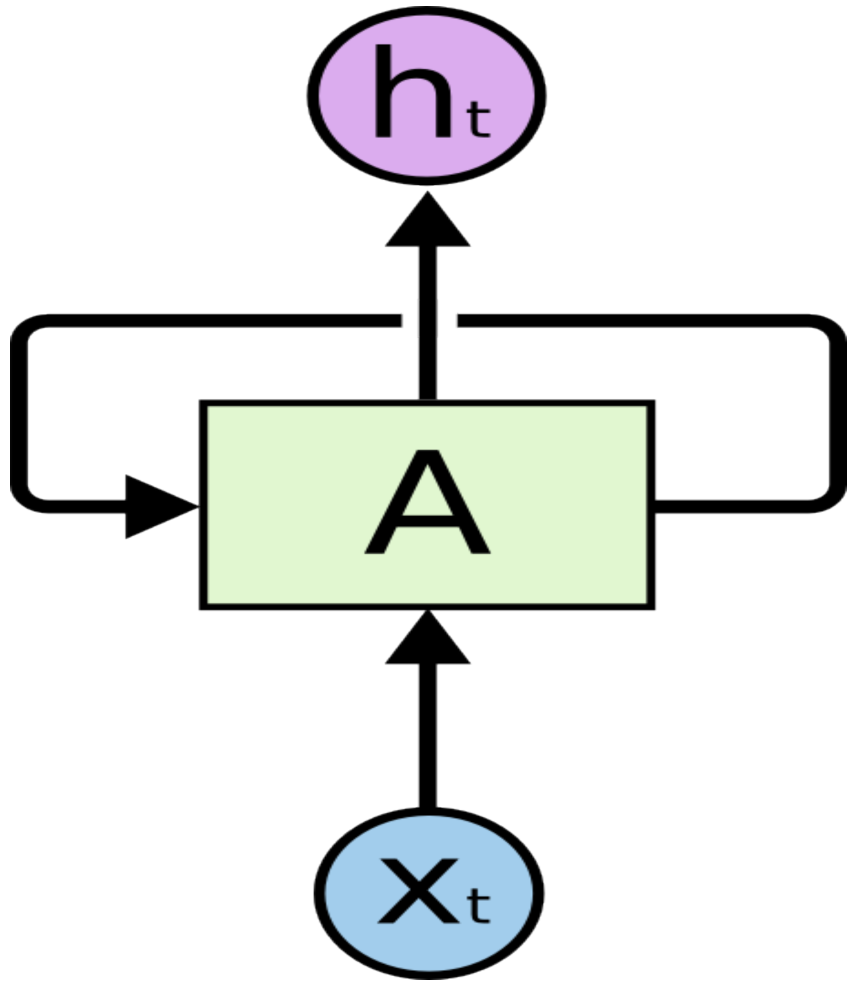
Recurrent Neural Networks (RNNs)

- Add feedback loops where some units' current outputs determine some future network inputs.
- RNNs can model dynamic finite-state machines, beyond the static combinatorial circuits modeled by feed-forward networks.

Simple Recurrent Network (SRN)

- Initially developed by Jeff Elman ("*Finding structure in time*," 1990).
- Additional input to hidden layer is the state of the hidden layer in the previous time step.

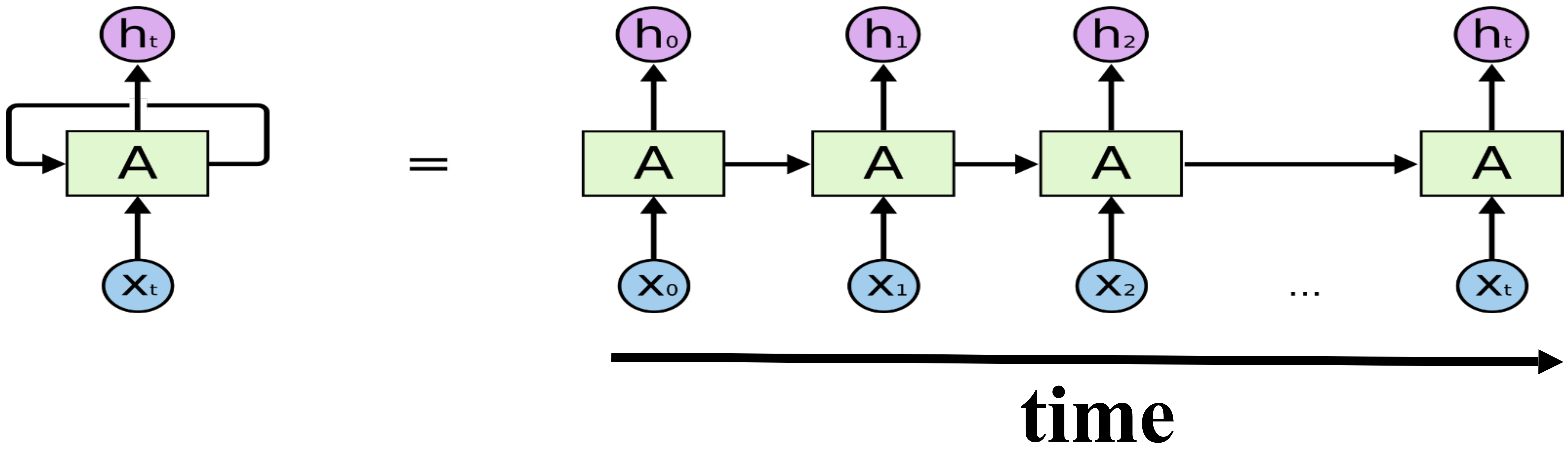




http://colah.github.io/posts/2015-08-Understanding-LSTMs/

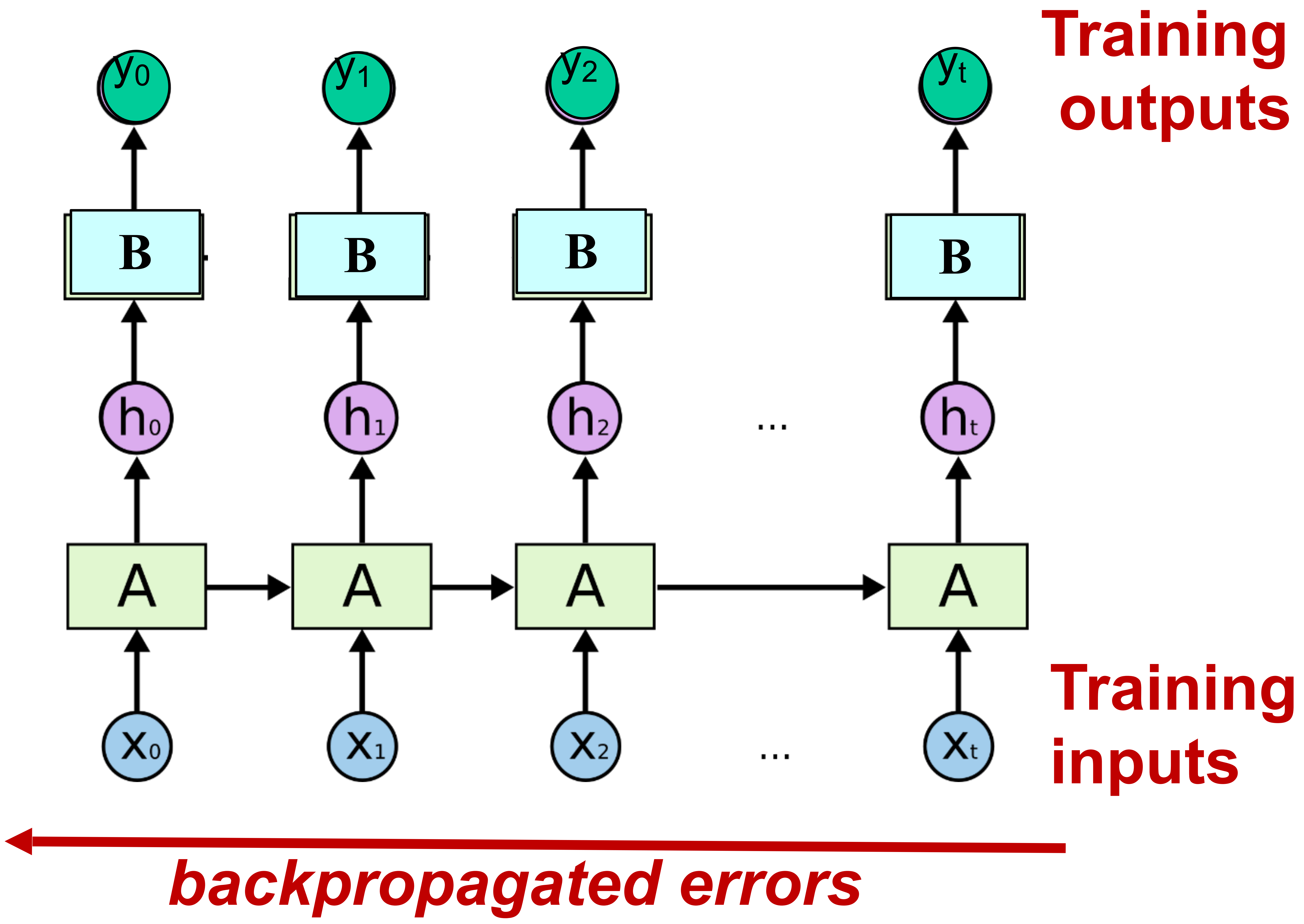
Unrolled RNN

- Behavior of RNN is perhaps best viewed by "unrolling" the network over time.



Training RNN's

- RNNs can be trained using "backpropagation through time."
- Can viewed as applying normal backprop to the unrolled network.



Vanishing/Exploding Gradient Problem

- Backpropagated errors multiply at each layer, resulting in exponential decay (if derivative is small) or growth (if derivative is large).
- Makes it very difficult train deep networks, or simple recurrent networks over many time steps.

Long Distance Dependencies

- It is very difficult to train SRNs to retain information over many time steps.
- This make is very difficult to learn SRNs that handle long-distance dependencies, such as subject-verb agreement.