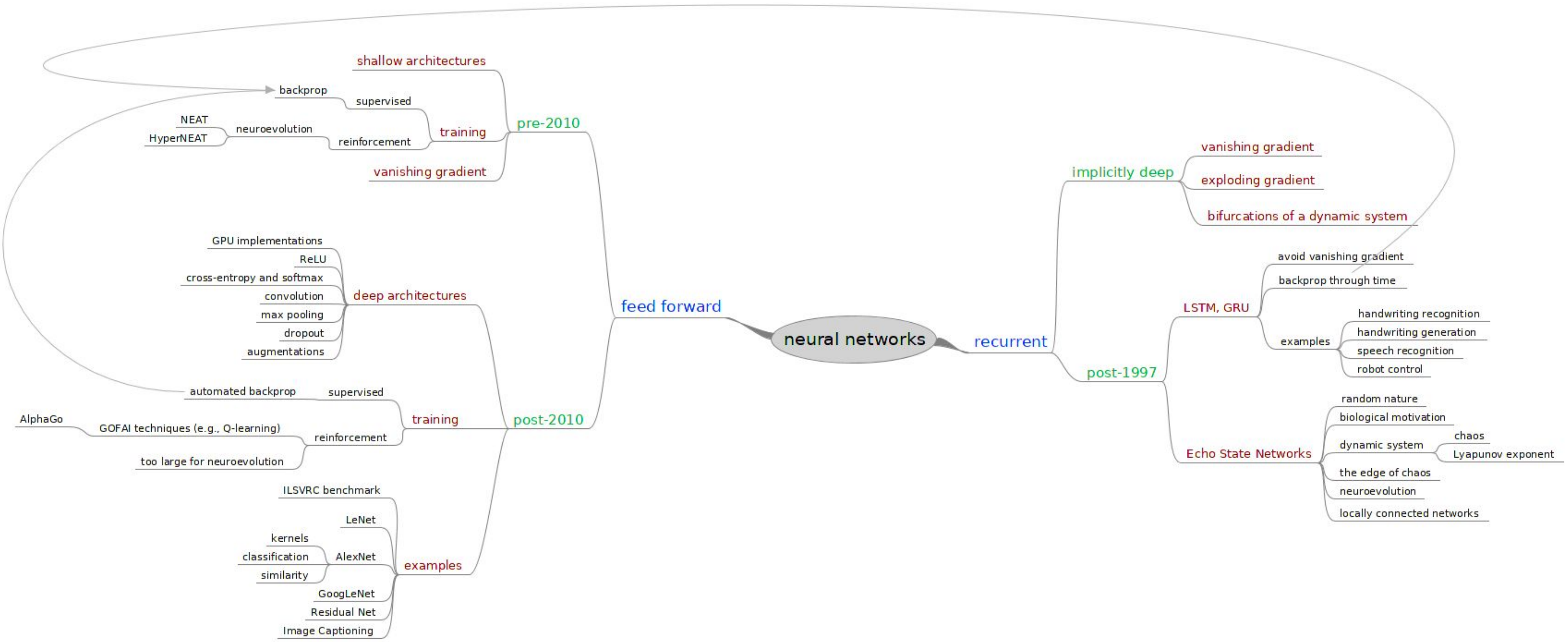
Deep Learning, Echo State Networks and the Edge of Chaos

Artificial Intelligence Seminar 1. 11. 2016

Filip Matzner, František Mráz

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

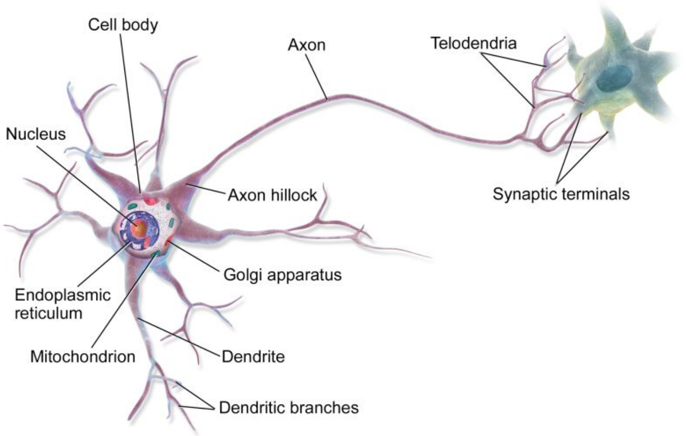
Outline


neural networks
feed forward
pre-2010
shallow architectures
training
supervised
backprop
reinforcement
neuroevolution
NEAT
HyperNEAT
vanishing gradient
post-2010
deep architectures
GPU implementations
ReLU
cross-entropy and softmax
convolution
max pooling
dropout
augmentations
training
supervised
automated backprop
reinforcement
GOFAI techniques (e.g., Q-learning)
AlphaGo
too large for neuroevolution
examples
ILSVRC benchmark
LeNet
AlexNet
kernels
classification
similarity
GoogLeNet
Residual Net
Image Captioning
recurrent
implicitly deep
vanishing gradient
exploding gradient
bifurcations of a dynamic system
post-1997
LSTM, GRU
avoid vanishing gradient
backprop through time
examples
handwriting recognition
handwriting generation
speech recognition
robot control
Echo State Networks
random nature
biological motivation
dynamic system
chaos
Lyapunov exponent
the edge of chaos
neuroevolution
locally connected networks

Outline

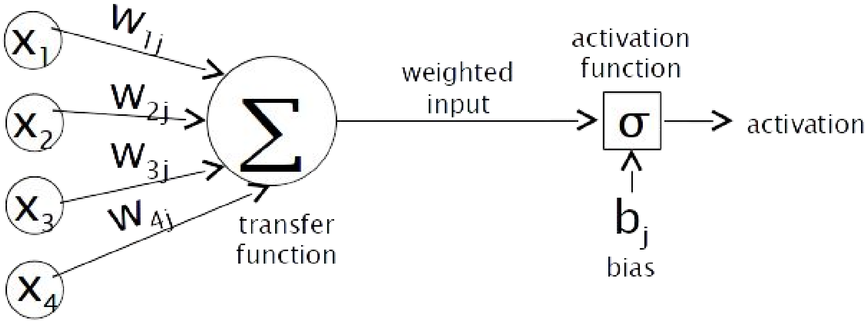
Neuron

biological



artificial

$$y_j = \varphi\left(\sum_{i=1}^{N} w_{ij} x_i + b_j\right)$$



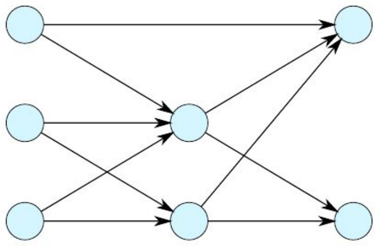
Network Architectures

biological neural networks are recurrent

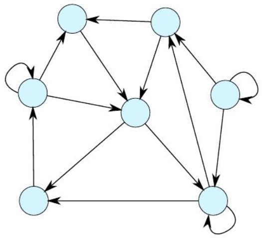
artificial recurrent networks are hard to train

$\Longrightarrow$ feed-forward networks receive more attention

feed-forward



recurrent



Outline

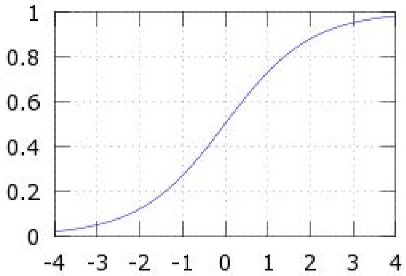
shallow architectures
backprop
supervised
training
reinforcement
vanishing gradient
pre-2010
feed forward
neural networks
recurrent
post-2010

Feed-Forward Networks (Pre-2010)

shallow = no more than a single hidden layer

sigmoidal activation function

$$y = \frac{1}{1+e^{-x}}$$



mean squared error cost function

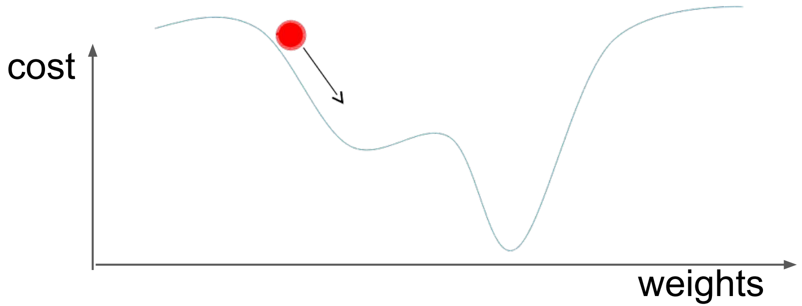
$$C_x(w,b) = \frac{\|y(x)-a\|^2}{2}$$

Feed-Forward Supervised Training

gradient descent

a method for function minimization

idea: put a ball on the cost function surface and let it roll down



it is too simple, where is the hatch?

in calculating the derivatives for each weight $\nabla C = \left(\frac{\delta C}{\delta w_{11}^{(1)}}, \frac{\delta C}{\delta w_{21}^{(2)}}, \cdots, \frac{\delta C}{\delta w_{jk}^{L}} \right)$

Backpropagation

a gradient descent method specialized on neural networks

efficiently propagates the error gradient from the output layer to the input layer

calculates all the derivatives $\nabla C = \left(\frac{\delta C}{\delta w_{11}^{(1)}}, \frac{\delta C}{\delta w_{21}^{(2)}}, \cdots, \frac{\delta C}{\delta w_{jk}^{L}} \right)$ in a single pass

Summary: the equations of backpropagation

$$\delta^L = \nabla_a C \odot \sigma'(z^L) \tag{BP1}$$

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \tag{BP2}$$

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l \tag{BP3}$$

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l \tag{BP4}$$

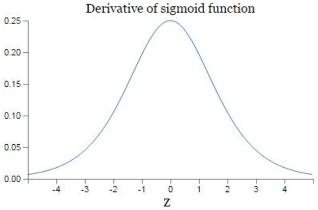
Vanishing Gradient

why the networks cannot not be deeper?

imagine a simple network with three hidden layers

$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \times w_2 \times \sigma'(z_2) \times w_3 \times \sigma'(z_3) \times w_4 \times \sigma'(z_4) \times \frac{\partial C}{\partial a_4}$$

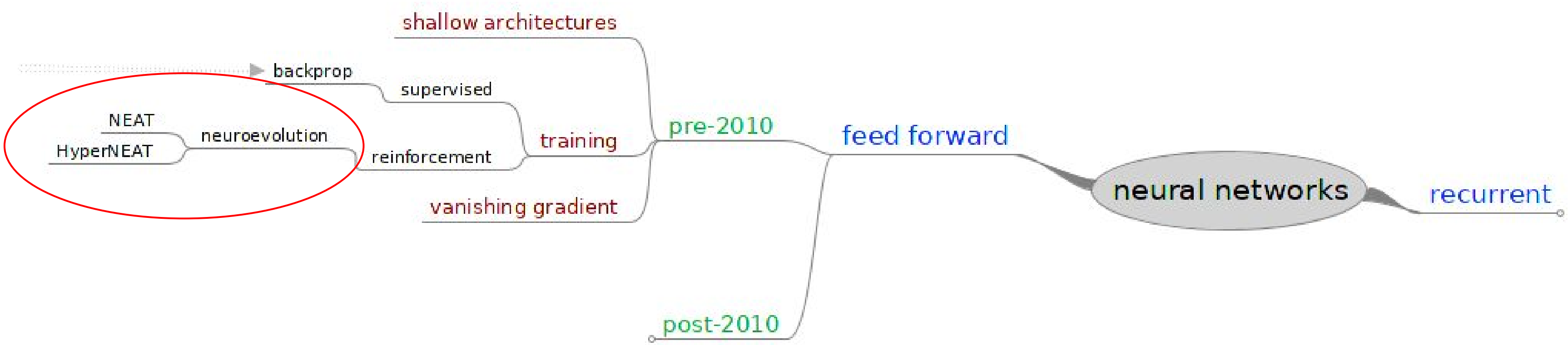
now plot the derivative of the sigmoid function



usually $|w_j \sigma(z_j)| < \frac{1}{4}$,thus with each layer, the gradient exponentially decreases

problem first described by Bengio et al. [1994]
this simple description taken from Nielsen [2015]

Outline



shallow architectures
backprop
supervised
NEAT
neuroevolution
HyperNEAT
reinforcement
training
pre-2010
feed forward
neural networks
recurrent
vanishing gradient
post-2010

Feed-Forward Neuroevolution

in the simplest possible scenario

- consider the network weights to be a vector of real numbers
- evolve the weights by the basic genetic algorithm

this does not work particularly well, specialized evolutionary algorithms exist

NEAT

one of the many existing neuroevolutionary algorithms

good for smaller networks, not as much for large ones

can evolve both feed-forward networks and recurrent networks

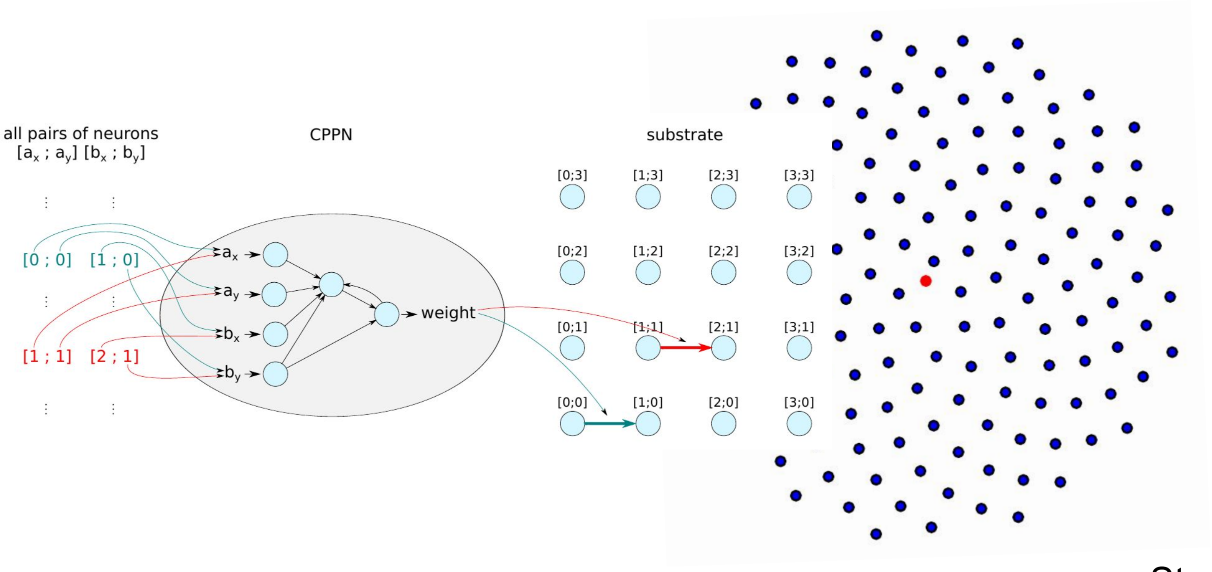
even though it is quite old, it provides a suitable baseline

Stanley and Miikkulainen [2002]

HyperNEAT

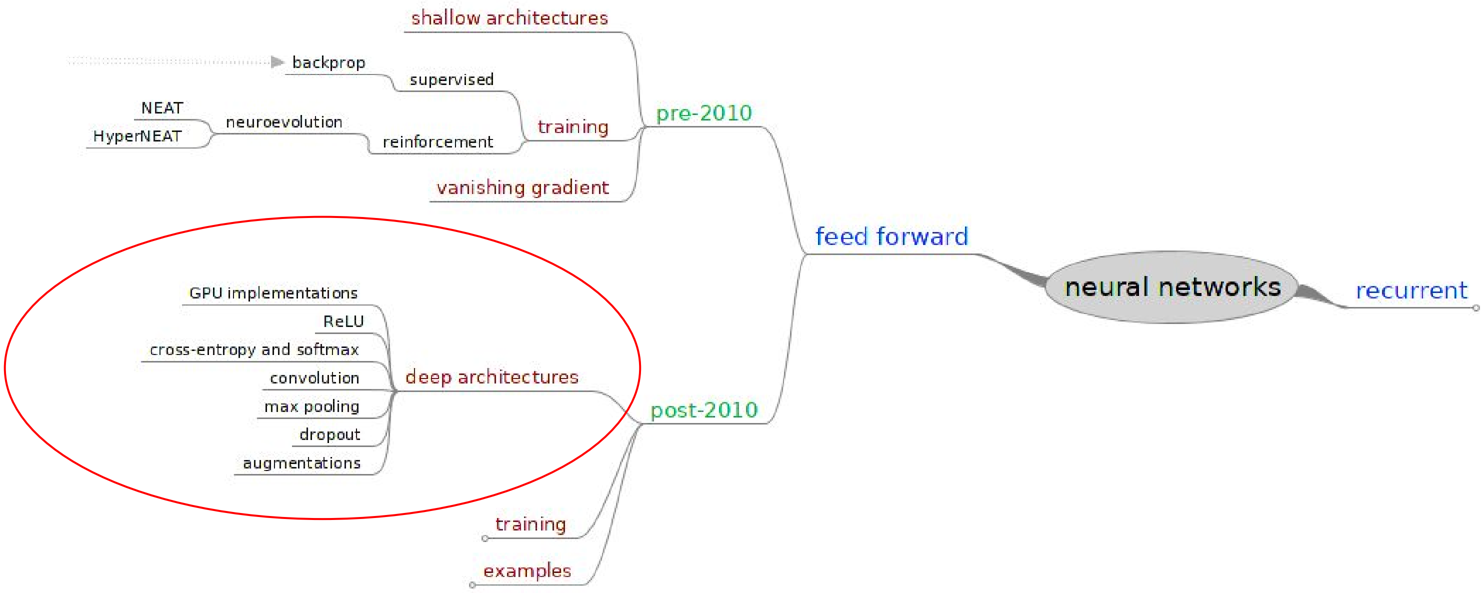
extension of NEAT, can evolve larger networks

good when the network contains a lot of regularities



Stanley et al. [2009]

Outline


shallow architectures
backprop
supervised
NEAT
neuroevolution
HyperNEAT
reinforcement
training
pre-2010
vanishing gradient
feed forward
neural networks
recurrent
GPU implementations
ReLU
cross-entropy and softmax
convolution
deep architectures
max pooling
dropout
post-2010
augmentations
training
examples

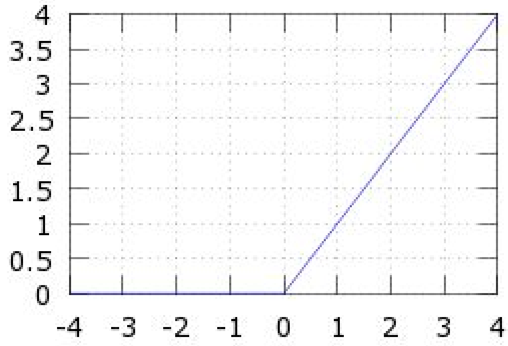
Feed-Forward Networks (Post-2010)

deep = more than two hidden layers

GPU implementations (100x speedup)

ReLU activation function

$$y = \max(0, x)$$

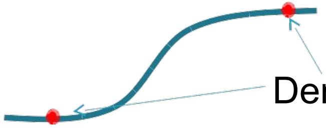


Cross-Entropy Cost Function

saturated neurons learn slowly with MSE even though their error might be the largest one

why? because of the shape of the sigmoid function

$$\frac{\partial C}{\partial w} = (a - y)\sigma'(z)$$



Derivation $\frac{\partial C}{\partial w^l_{jk}}$ at this point is small

it can be partially avoided using cross-entropy cost function

$$\mathrm{C} = -\frac{1}{\mathrm{n}}\sum_{x_j}[y_j \ln a^L_j + (1 - y_j)\ln(1 - a^L_j)]$$

Softmax Activation

another way to address the learning slowdown, especially when combined with the cross-entropy cost function

emphasizes the neuron with the maximum activation, however, does not ignore the other neurons

$$a_j^L = \frac{e^{z_j^L}}{\sum_k e^{z_k^L}}$$

can be thought of as a probabilistic distribution, because

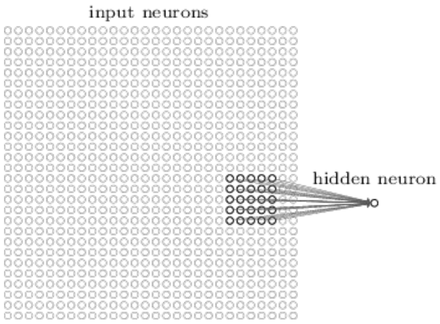
$$\sum_j a_j^L = \frac{\sum_j e^{z_j^L}}{\sum_k e^{z_k^L}} = 1$$

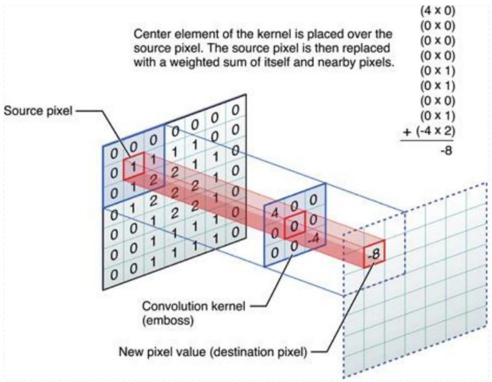
Convolutional Layers

apply the same convolutional kernel to all the pixels in the image

a way to share the weights between neurons ⟹ regularization

inspired by nature



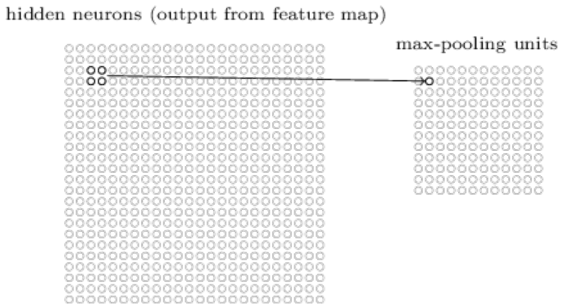


Max Pooling

similar to the convolution but only takes the maximum of the perception field

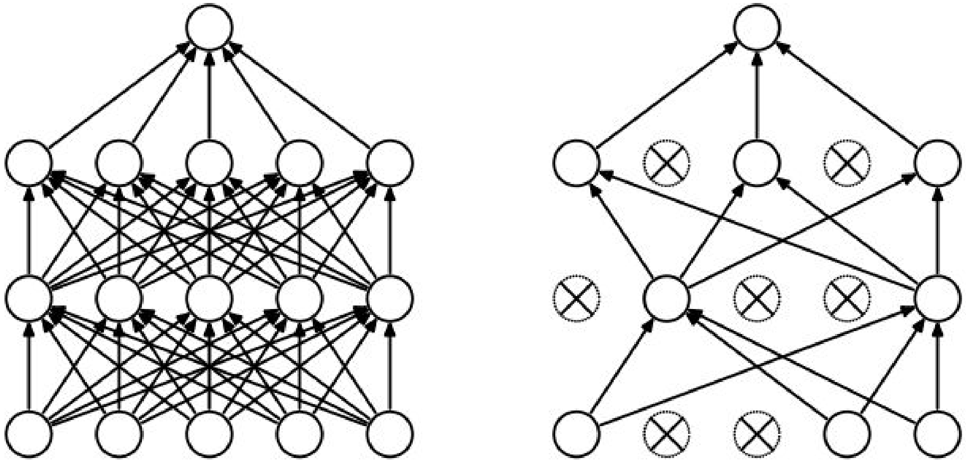
a good way to subsample the image (i.e., fewer parameters to train)

forces more succinct image representation (i.e., compression)



Dropout

randomly disable some of the neurons in each backprop step



(a) Standard Neural Net　　(b) After applying dropout.

Hinton et al. [2012]

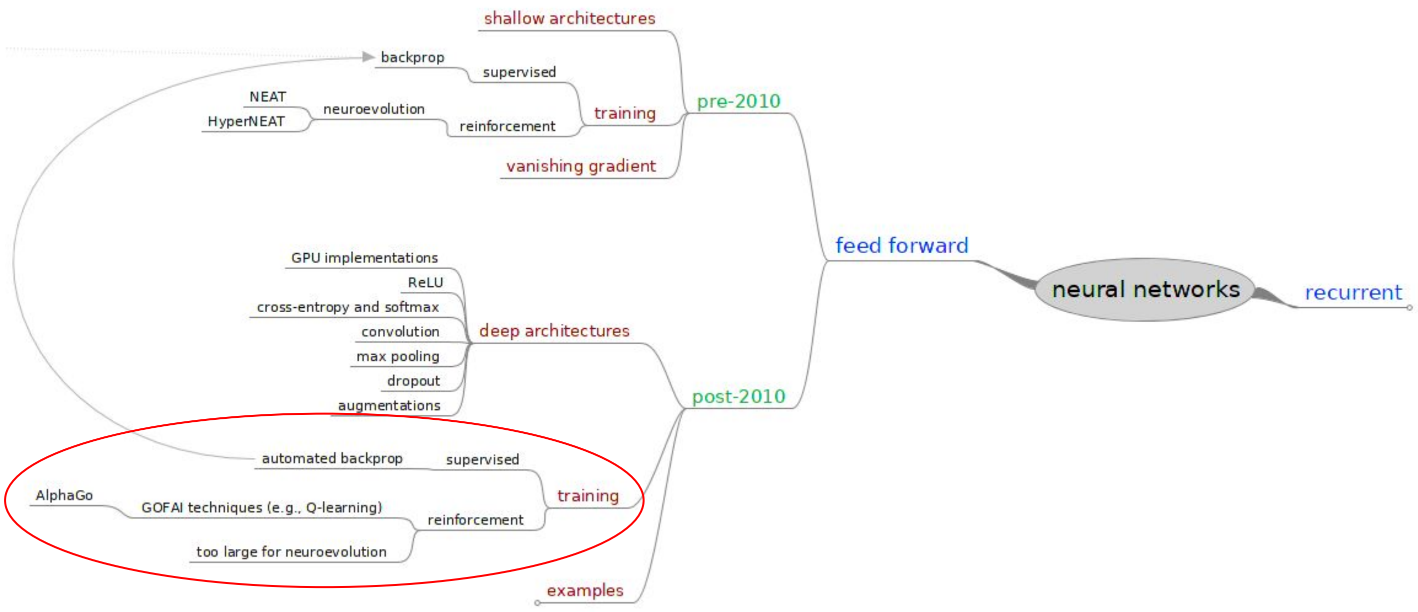
Data Augmentation

enlarge the dataset by transforming the data so that they still make sense

horizontal flip, rotation, scale, color inversion, ...

reduces overfitting

Outline



neural networks
feed forward
recurrent
pre-2010
shallow architectures
training
supervised
backprop
reinforcement
neuroevolution
NEAT
HyperNEAT
vanishing gradient
post-2010
deep architectures
GPU implementations
ReLU
cross-entropy and softmax
convolution
max pooling
dropout
augmentations
training
supervised
automated backprop
reinforcement
GOFAI techniques (e.g., Q-learning)
AlphaGo
too large for neuroevolution
examples

Supervised Training

backprop variants, e.g.,

- RMSProp (unpublished, see Hinton Neural Networks on Coursera [2012])
- Adam (Kingma and Ba [2015])

the derivatives do not have to be calculated manually, your favourite deep learning framework does it for you

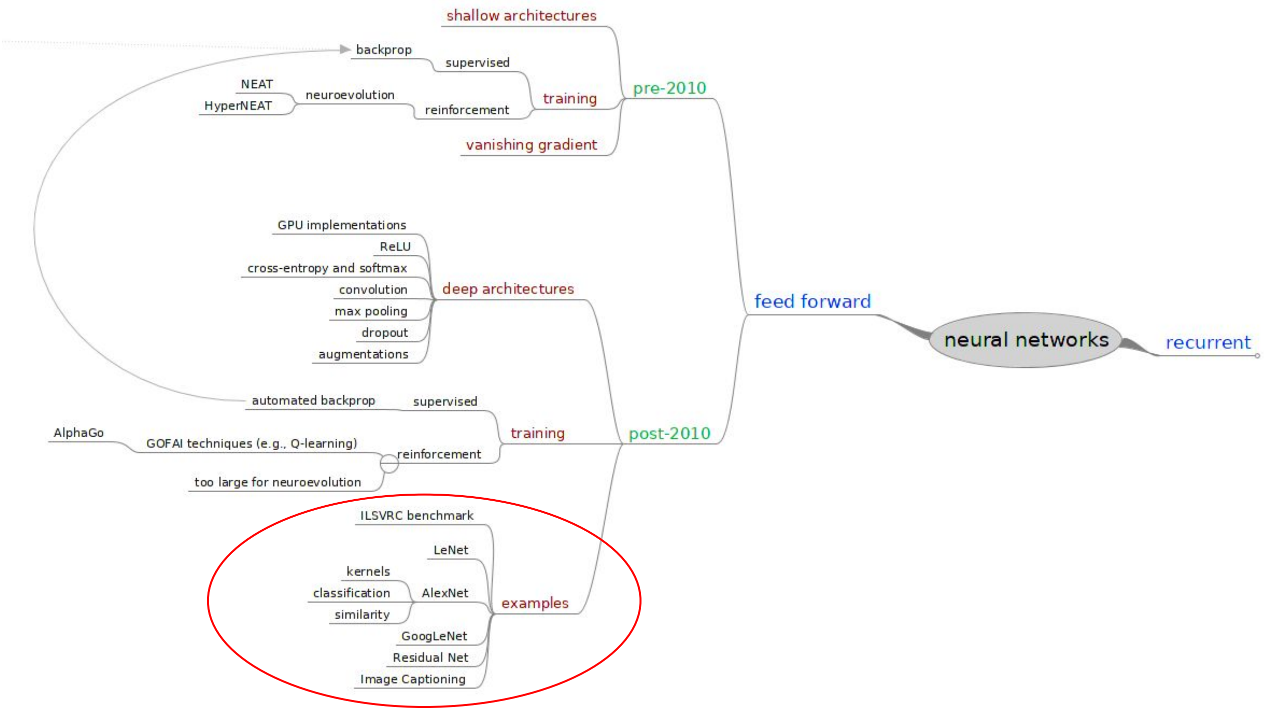
Neuroevolution

the networks are usually too large to be evolved purely by neuroevolution

not even a combination of neuroevolution and supervised training is usual

- a single training on the GPU usually takes hours or days

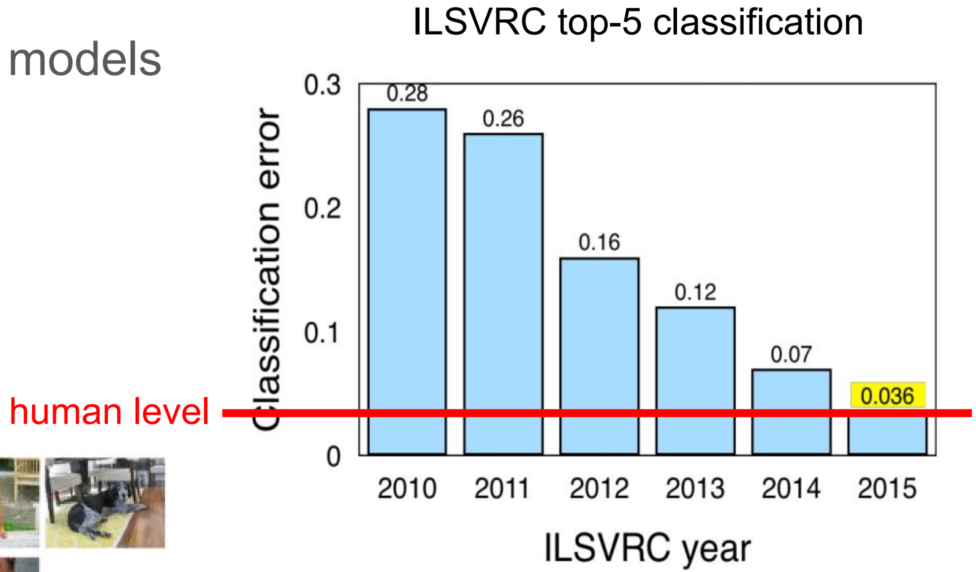
Outline

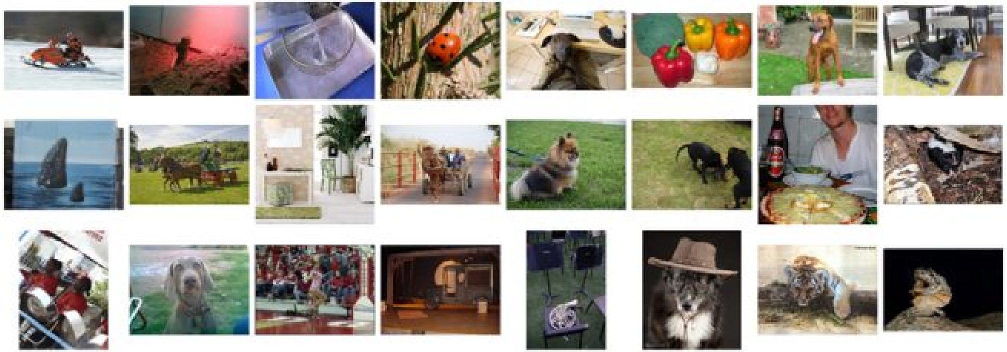


shallow architectures
backprop
supervised
NEAT
neuroevolution
HyperNEAT
reinforcement
training
pre-2010
vanishing gradient
GPU implementations
ReLU
cross-entropy and softmax
convolution
deep architectures
max pooling
dropout
augmentations
feed forward
neural networks
recurrent
automated backprop
supervised
training
post-2010
AlphaGo
GOFAI techniques (e.g., Q-learning)
reinforcement
too large for neuroevolution
ILSVRC benchmark
LeNet
kernels
classification
AlexNet
similarity
examples
GoogLeNet
Residual Net
Image Captioning

ILSVRC

- benchmark of image classification models
- ~1.5 million images
- 1000 classes

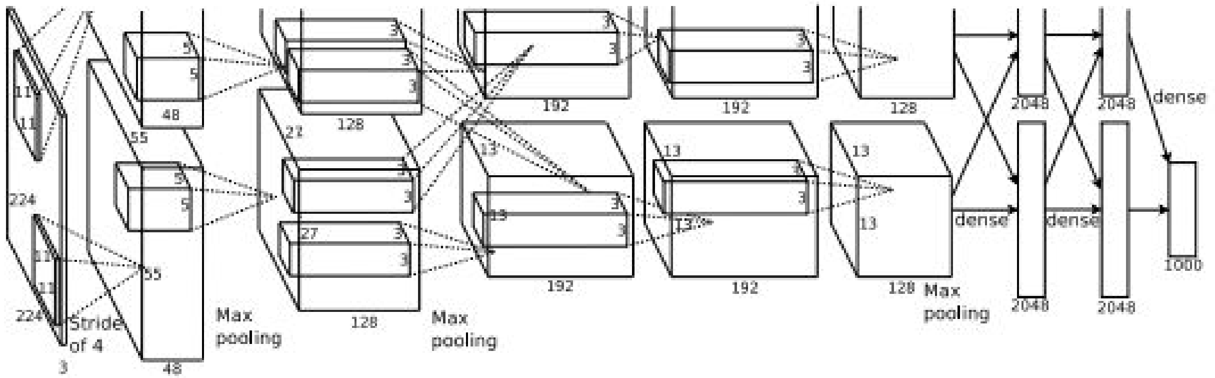




Russakovsky et al. [2015]

AlexNet

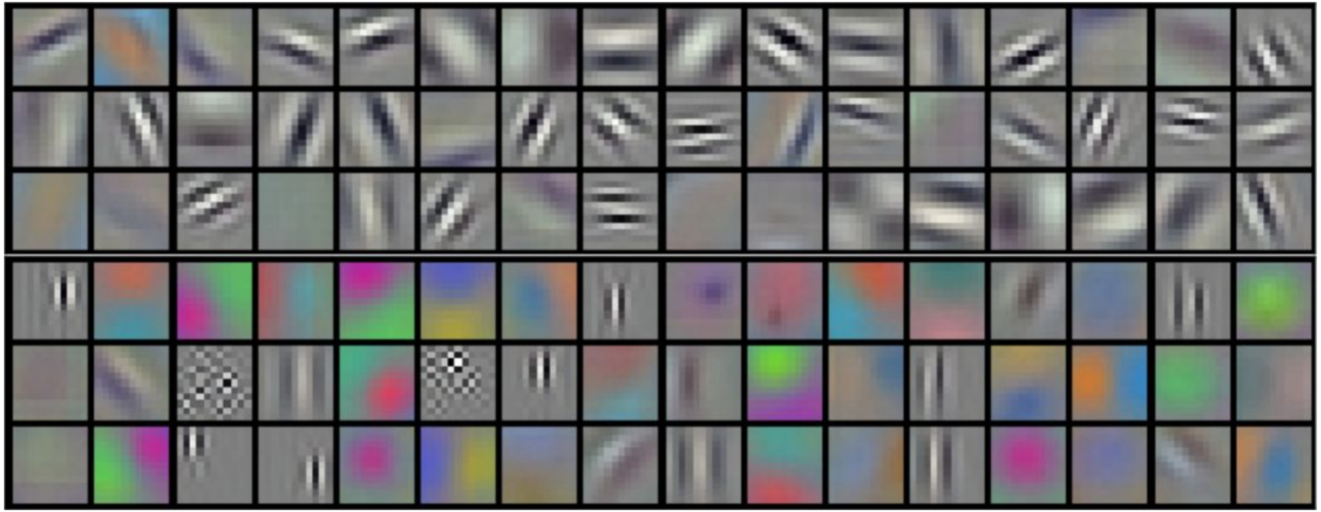
- the winner of ILSVRC image classification 2012
- 8 layers
- 15.3% top-5 error
- 60 million parameters, 2 GPUs
- the beginning of the neural-network-computer-vision era



Krizhevsky et al. [2012]

AlexNet

- One GPU evolved color filters, the other evolved black and white filters



Krizhevsky et al. [2012]

AlexNet - Classification Results

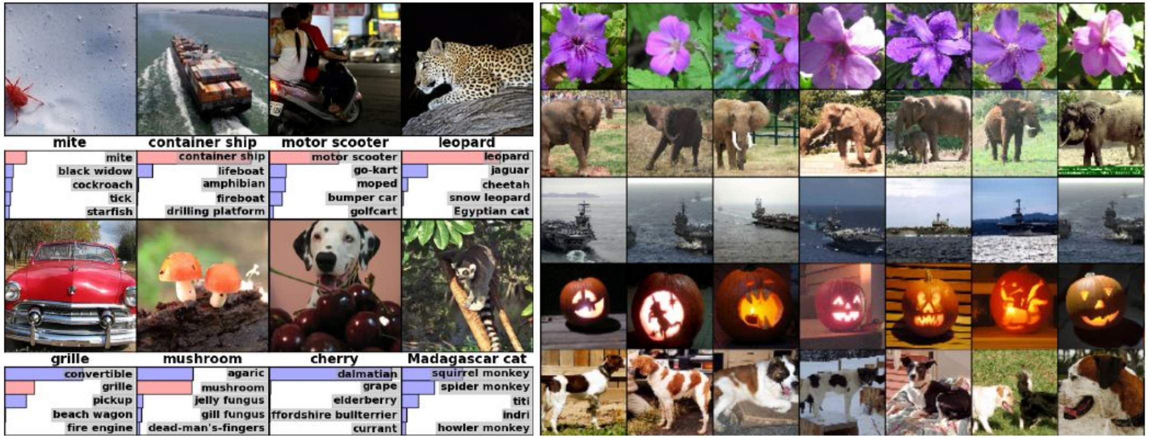
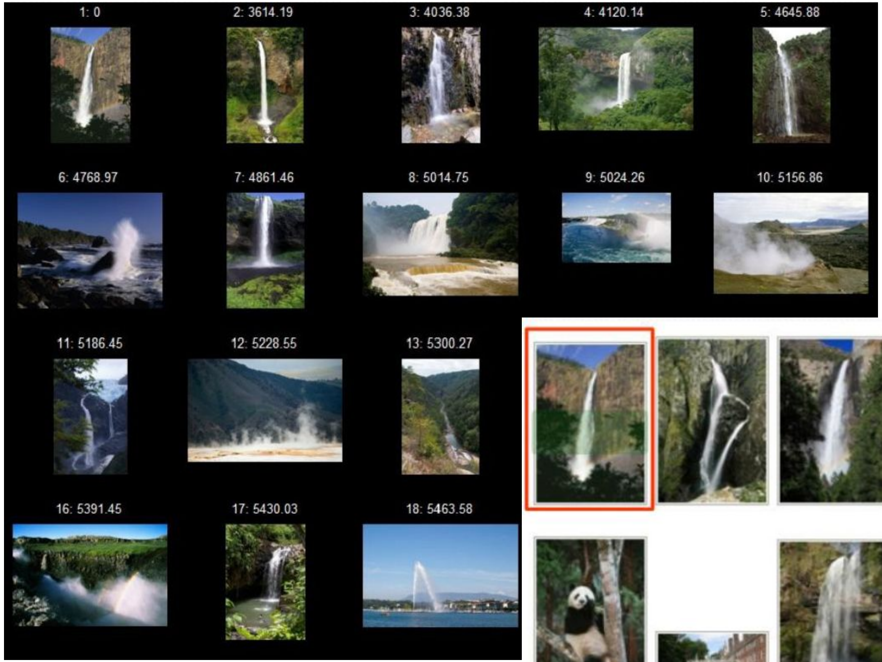


Figure 4: **(Left)** Eight ILSVRC-2010 test images and the five labels considered most probable by our model. The correct label is written under each image, and the probability assigned to the correct label is also shown with a red bar (if it happens to be in the top 5). **(Right)** Five ILSVRC-2010 test images in the first column. The remaining columns show the six training images that produce feature vectors in the last hidden layer with the smallest Euclidean distance from the feature vector for the test image.

Krizhevsky et al. [2012]

AlexNet - Image Similarity



Krizhevsky et al. [2012]

Previous best by Zezula et al. [2005]

GoogLeNet

- the winner of ILSVRC image classification 2014
- 22 layers
- 6.67% top-5 error
- Google

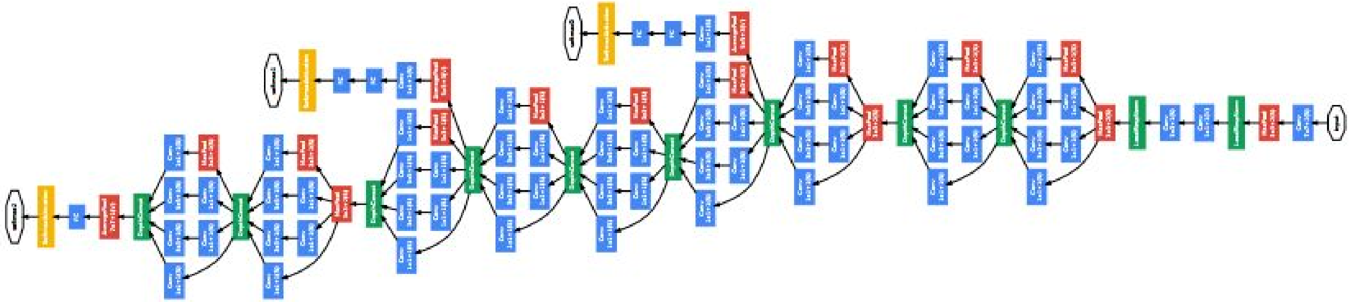


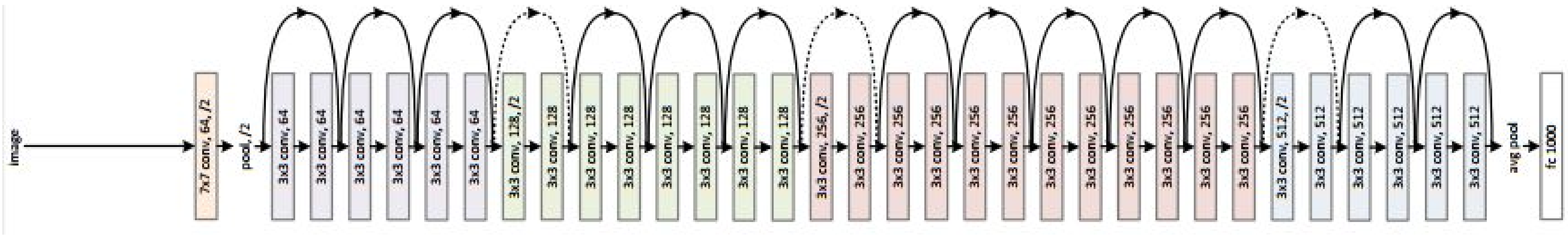
Figure 3: GoogLeNet network with all the bells and whistles

Szegedy et al. [2014]

Residual Networks

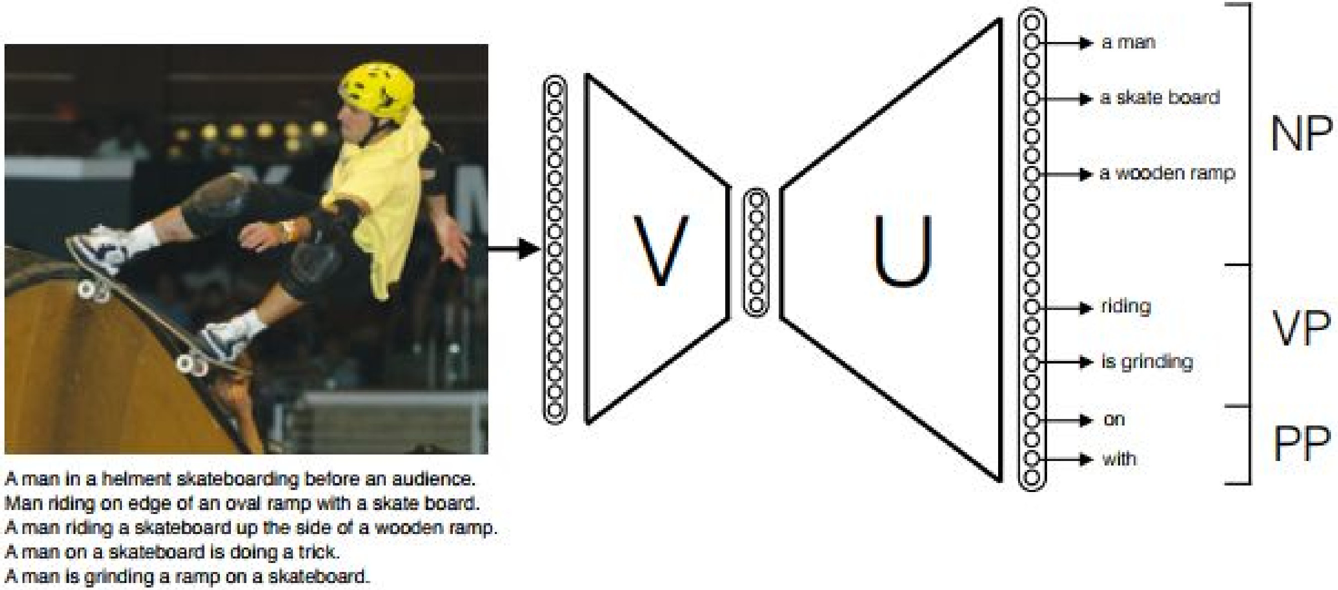
- the winner of ILSVRC image classification 2015
- 152 layers
- 4.49% top-5 error
- Microsoft research

example of a 34-layers residual network



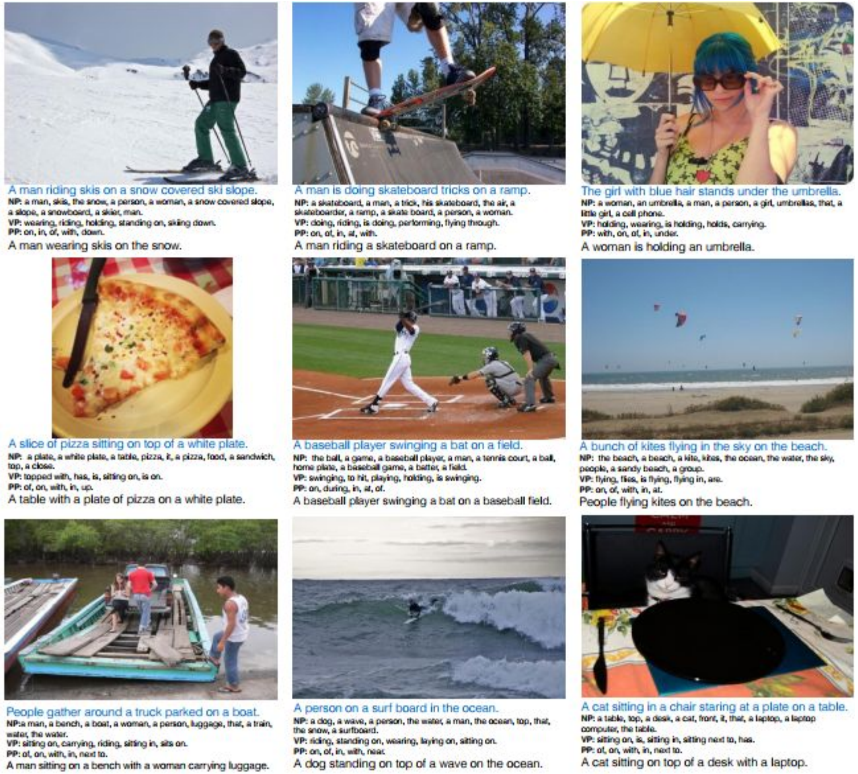
Kaiming He et al. [2015]

Image Captioning



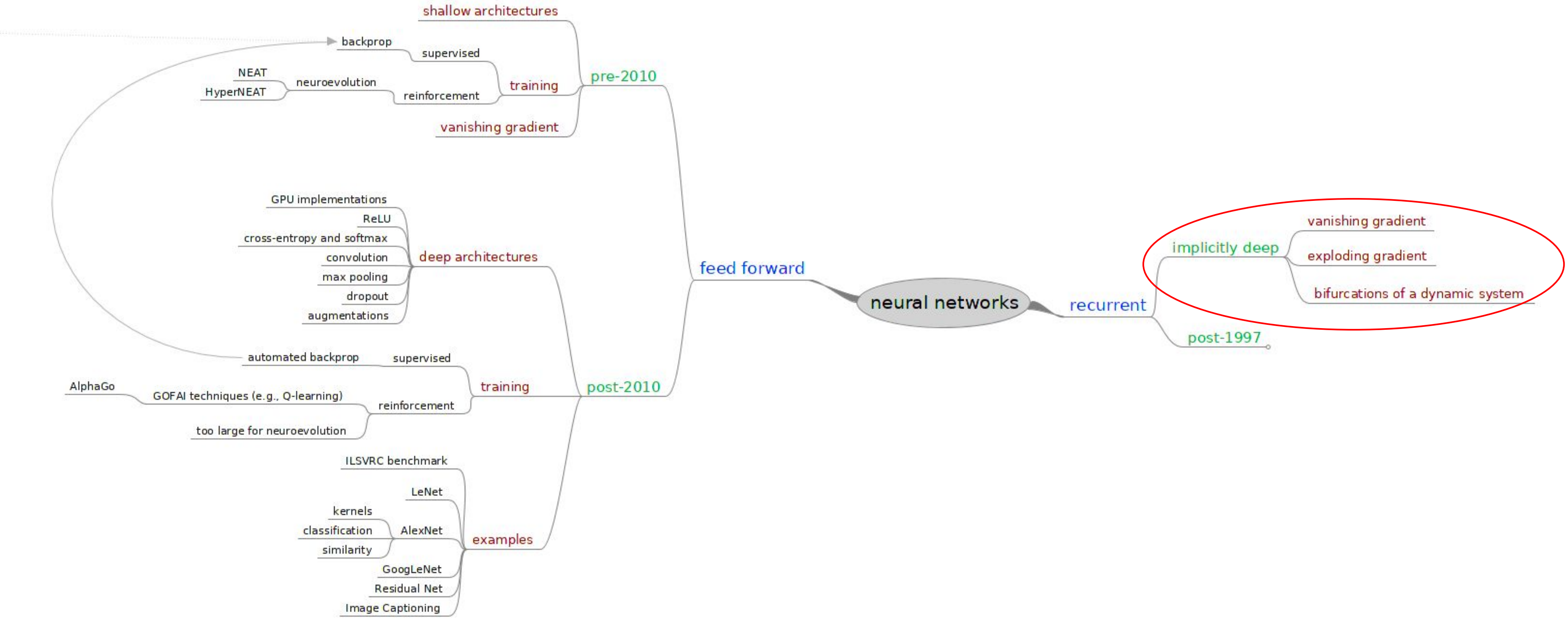
Lebret et al. [2015]

Image Captioning



Lebret et al. [2015]

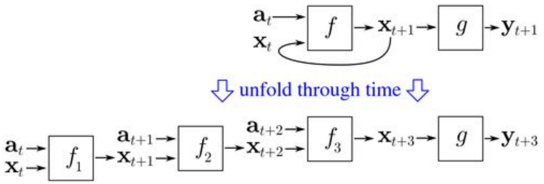
Outline



shallow architectures
backprop
supervised
NEAT
neuroevolution
training
pre-2010
HyperNEAT
reinforcement
vanishing gradient
GPU implementations
ReLU
cross-entropy and softmax
convolution
deep architectures
max pooling
dropout
augmentations
automated backprop
supervised
AlphaGo
GOFAI techniques (e.g., Q-learning)
training
post-2010
reinforcement
too large for neuroevolution
ILSVRC benchmark
LeNet
kernels
classification
AlexNet
similarity
examples
GoogLeNet
Residual Net
Image Captioning
feed forward
neural networks
recurrent
implicitly deep
vanishing gradient
exploding gradient
bifurcations of a dynamic system
post-1997

Recurrent Networks

can be unfolded through time and reduced to feed-forward networks



$\Longrightarrow$ training by backpropagation (through time)

problem: implicitly infinitely deep

$\Longrightarrow$ the vanishing gradient is even more significant

partial solution - LSTM, GRU, Echo State Networks

Geometrical Representation of Exploding Gradient

consider the dynamical system

$x_t = w\sigma(x_t-1) + b$

Fig. 6 illustrates the error surface

$E_{50} = (\sigma(x_{50}) - 0.7)^2$

possible solution: limit the gradient norm

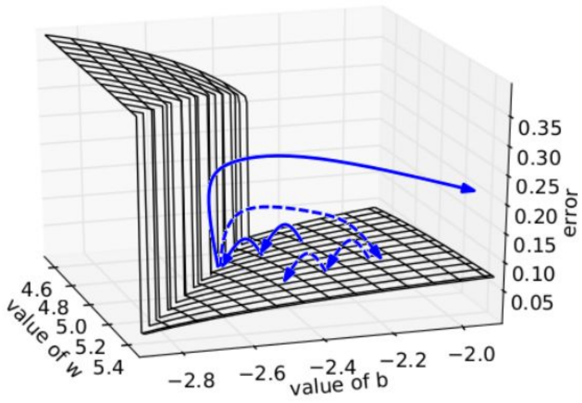
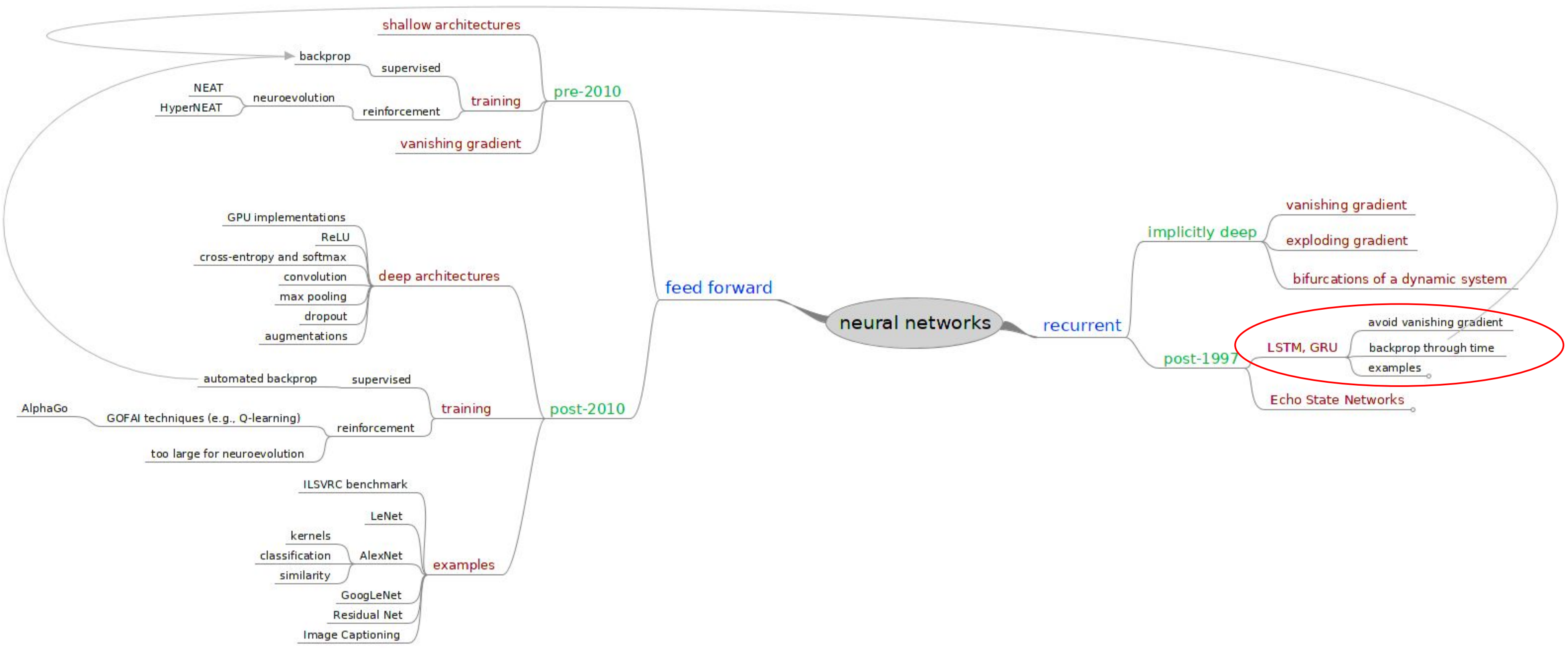


Figure 6. We plot the error surface of a single hidden unit recurrent network, highlighting the existence of high curvature walls. The solid lines depicts standard trajectories that gradient descent might follow. Using dashed arrow the diagram shows what would happen if the gradients is rescaled to a fixed size when its norm is above a threshold.

Pascanu et al. [2013]

Outline



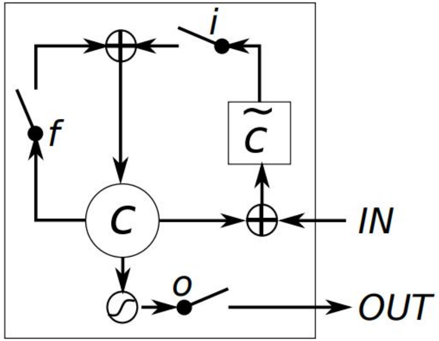
neural networks
feed forward
pre-2010
shallow architectures
training
supervised
backprop
reinforcement
neuroevolution
NEAT
HyperNEAT
vanishing gradient
post-2010
deep architectures
GPU implementations
ReLU
cross-entropy and softmax
convolution
max pooling
dropout
augmentations
training
supervised
automated backprop
reinforcement
GOFAI techniques (e.g., Q-learning)
AlphaGo
too large for neuroevolution
examples
ILSVRC benchmark
LeNet
AlexNet
kernels
classification
similarity
GoogLeNet
Residual Net
Image Captioning
recurrent
implicitly deep
vanishing gradient
exploding gradient
bifurcations of a dynamic system
post-1997
LSTM, GRU
avoid vanishing gradient
backprop through time
examples
Echo State Networks

Long-Short Term Memory (LSTM)

avoid vanishing gradient

use a special neuron with a “memory”
⟹ able to capture long-term dependencies

de-facto standard in recurrent neural networks



(a) Long Short-Term Memory

Hochreiter and Schmidhuber [1997]

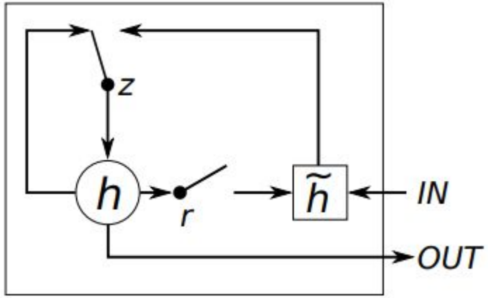
Long-Short Term Memory (LSTM)

Hochreiter and Schmidhuber [1997]

Gated Recurrent Unit (GRU)

slightly "simplified" version of LSTM

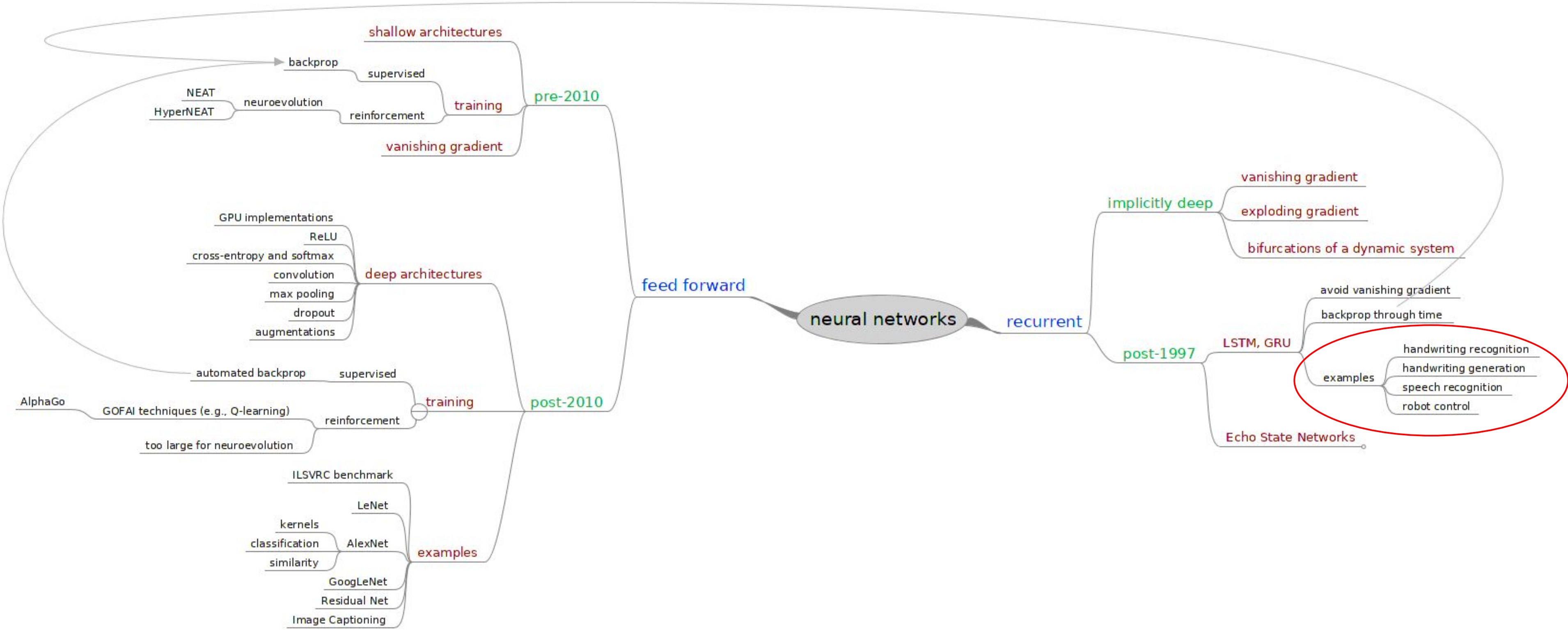
performance is comparable with LSTM



(b) Gated Recurrent Unit

Cho et al. [2014], comparison with LSTM by Chung et al. [2014]

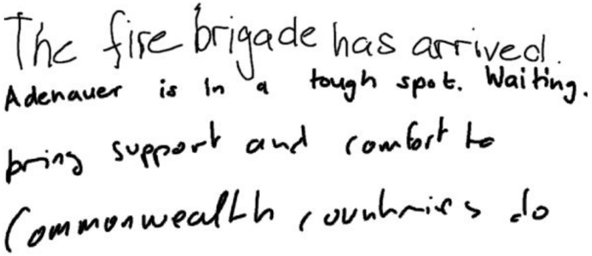
Outline



neural networks
feed forward
pre-2010
shallow architectures
training
supervised
backprop
reinforcement
neuroevolution
NEAT
HyperNEAT
vanishing gradient
post-2010
deep architectures
GPU implementations
ReLU
cross-entropy and softmax
convolution
max pooling
dropout
augmentations
training
supervised
automated backprop
reinforcement
GOFAI techniques (e.g., Q-learning)
AlphaGo
too large for neuroevolution
examples
ILSVRC benchmark
LeNet
AlexNet
kernels
classification
similarity
GoogLeNet
Residual Net
Image Captioning
recurrent
implicitly deep
vanishing gradient
exploding gradient
bifurcations of a dynamic system
post-1997
LSTM, GRU
avoid vanishing gradient
backprop through time
examples
handwriting recognition
handwriting generation
speech recognition
robot control
Echo State Networks

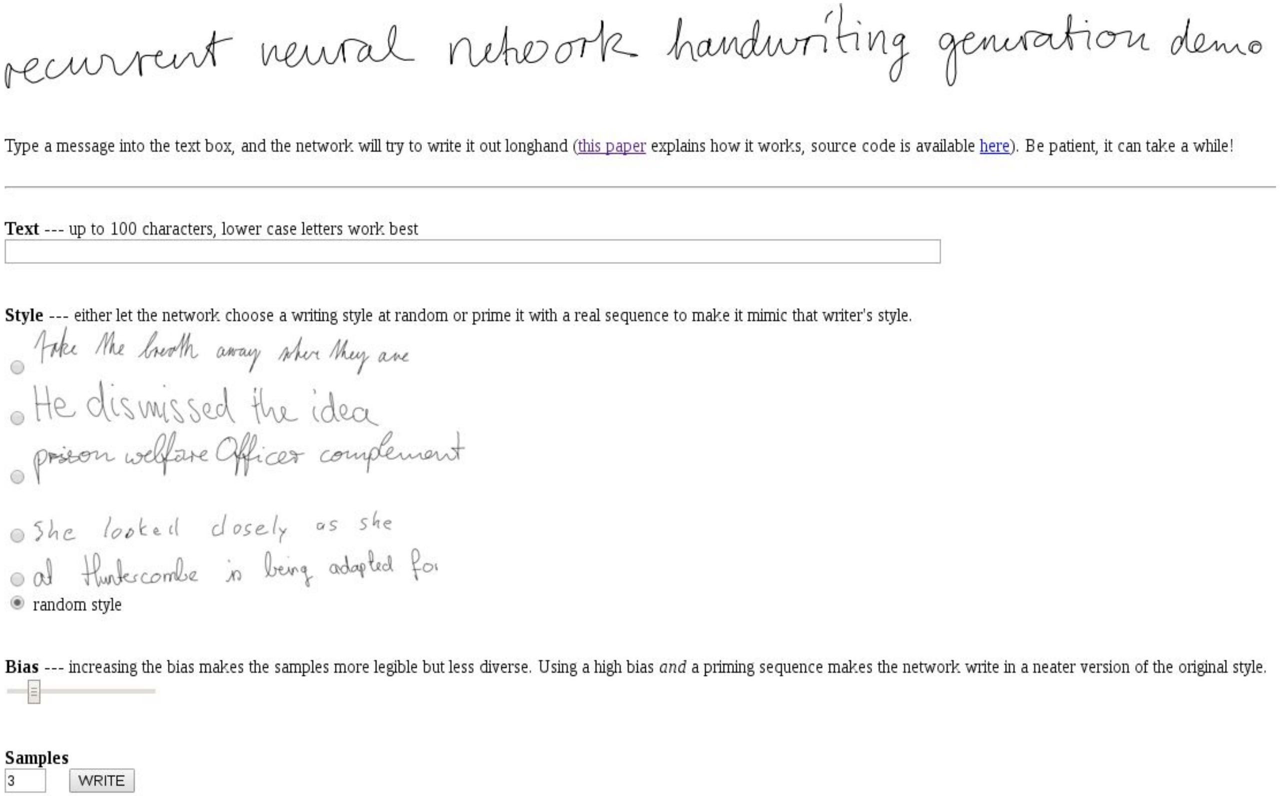
Handwriting Recognition

Fig. 1. Illustration of the recording



Graves et al. [2009]

Handwriting Generation

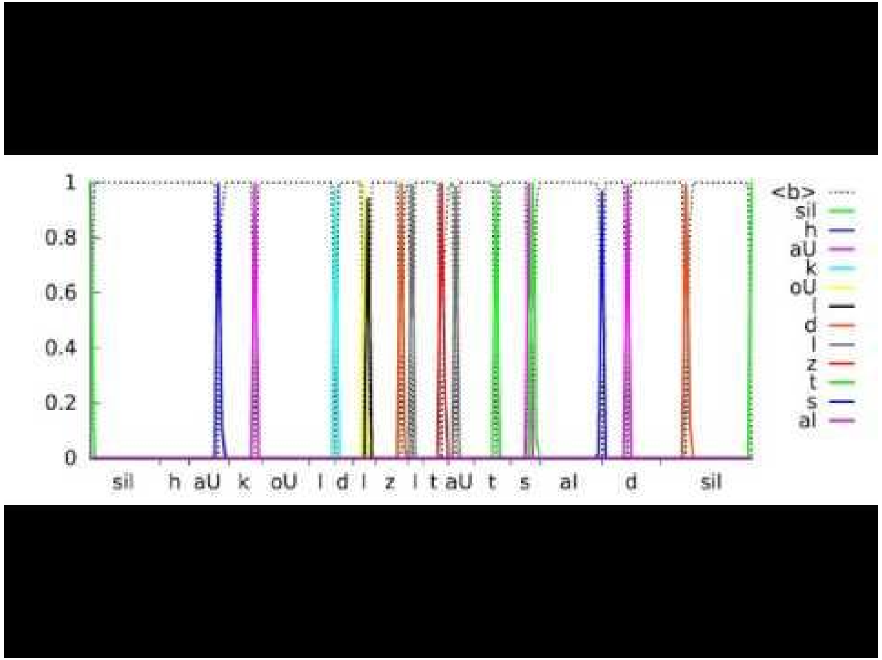


http://www.cs.toronto.edu/~graves/handwriting.html, Graves [2014]

Speech Recognition

"Using a LSTM, we cut our transcription errors by 49%."

-- Google Voice Blogspot [2015]



https://googleblog.blogspot.cz/2015/07/neon-prescription-or-rather-new.html

Graves et al. [2013]

Robot Control

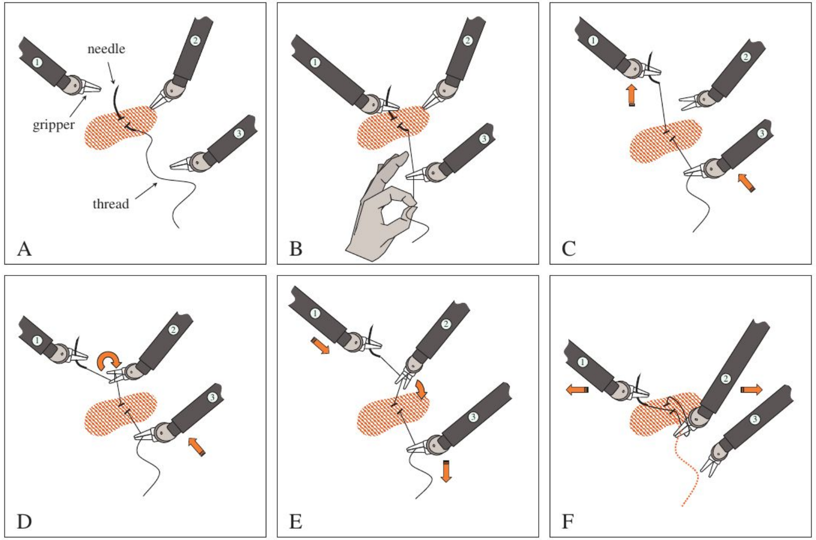


Fig. 3. **Minimally invasive knot-tying**. (A) The knot-typing procedure starts with the needle and three grippers in this configuration. (B) Gripper 1 takes the needle, and the thread is fed manually to gripper 3. (C) The thread is pulled through the puncture, and (D) wound around gripper 2. (E) Gripper 2 grabs the thread between the puncture and gripper 3. (F) The knot is finished by pulling the end of the thread through the loop.

Mayer et al. [2008]

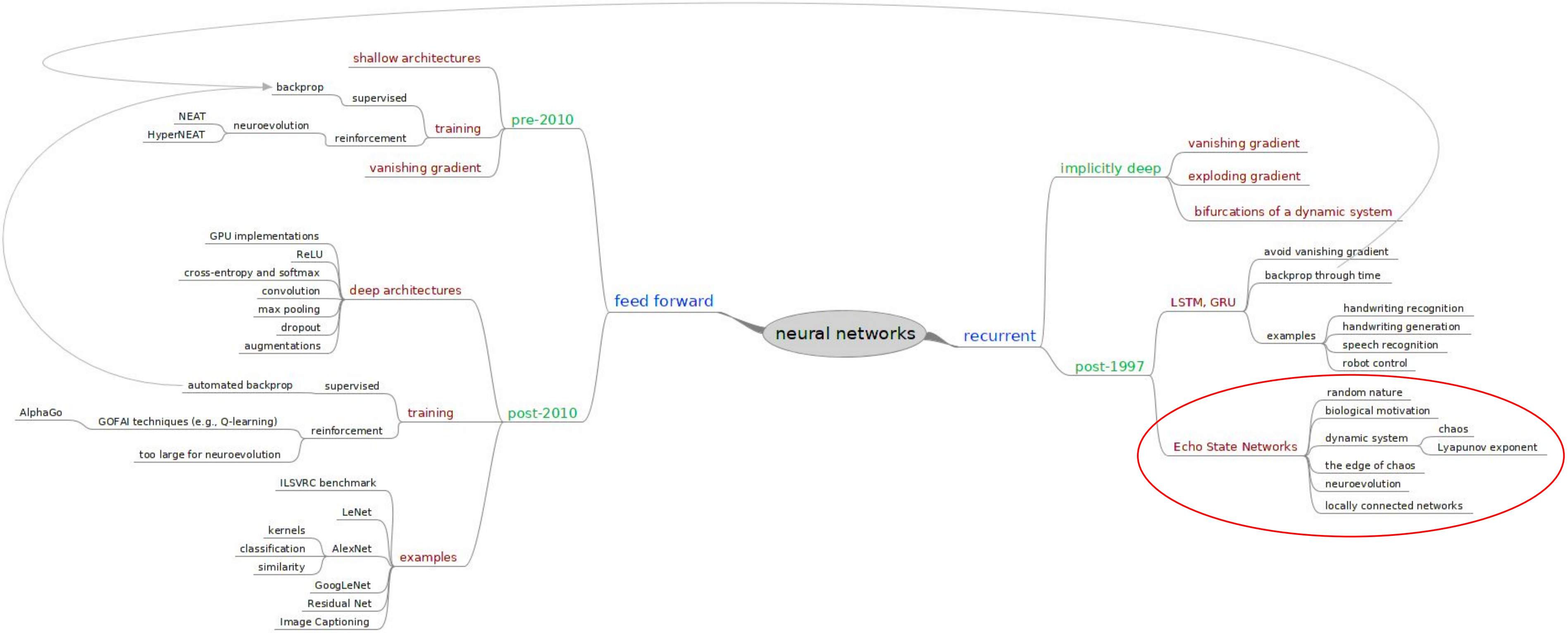
And more...

machine translation

image caption generation

...

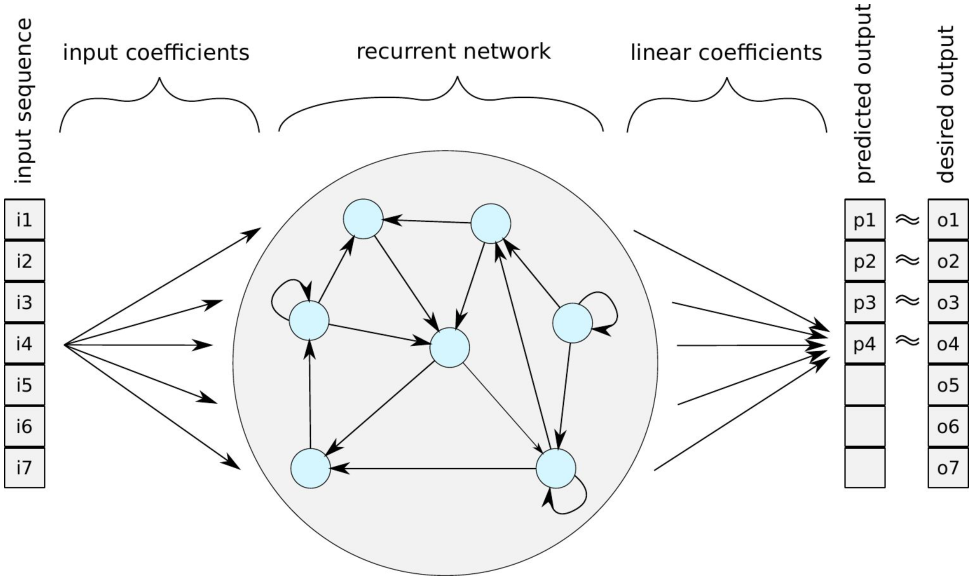
Outline



neural networks
feed forward
pre-2010
shallow architectures
training
supervised
backprop
reinforcement
neuroevolution
NEAT
HyperNEAT
vanishing gradient
post-2010
deep architectures
GPU implementations
ReLU
cross-entropy and softmax
convolution
max pooling
dropout
augmentations
training
supervised
automated backprop
reinforcement
GOFAI techniques (e.g., Q-learning)
AlphaGo
too large for neuroevolution
examples
ILSVRC benchmark
LeNet
AlexNet
kernels
classification
similarity
GoogLeNet
Residual Net
Image Captioning
recurrent
implicitly deep
vanishing gradient
exploding gradient
bifurcations of a dynamic system
post-1997
LSTM, GRU
avoid vanishing gradient
backprop through time
examples
handwriting recognition
handwriting generation
speech recognition
robot control
Echo State Networks
random nature
biological motivation
dynamic system
chaos
Lyapunov exponent
the edge of chaos
neuroevolution
locally connected networks

Echo State Networks

what about a totally random network?



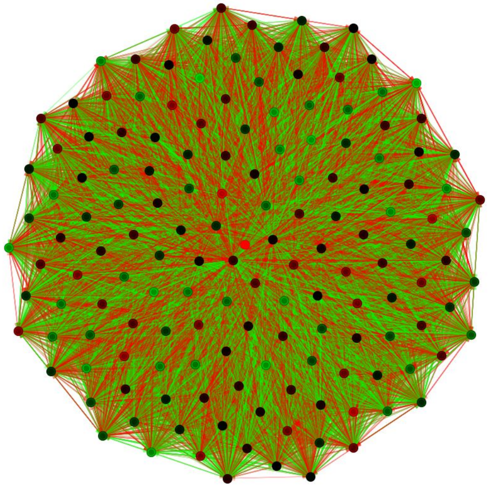
Jaeger [2001]

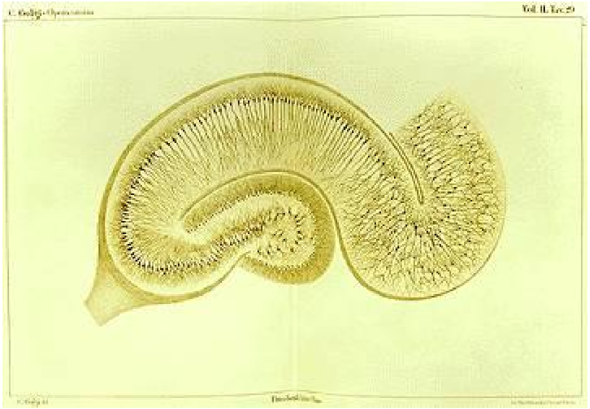
Biological Motivation

ESN's do not seem to be biologically plausible

improve them by a different topology?

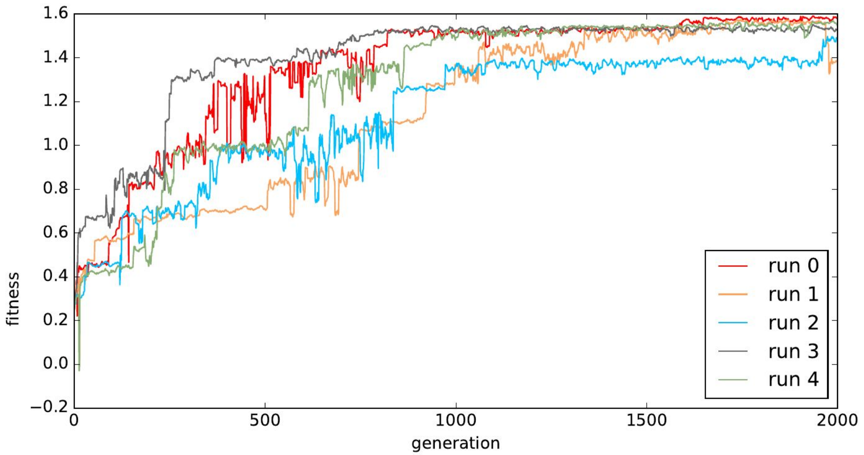
⟹ *neuroevolution (HyperNEAT)*

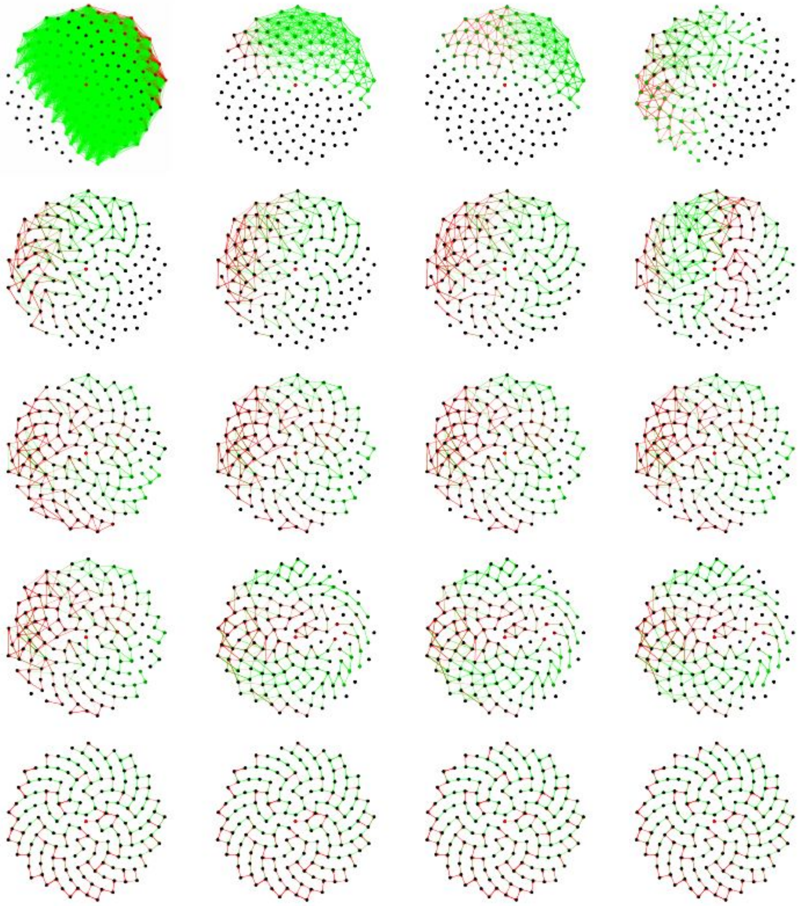




Neuroevolution

5 runs, 2000 generations, 150 individuals



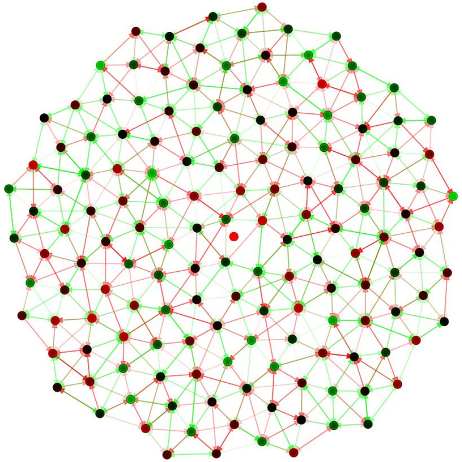


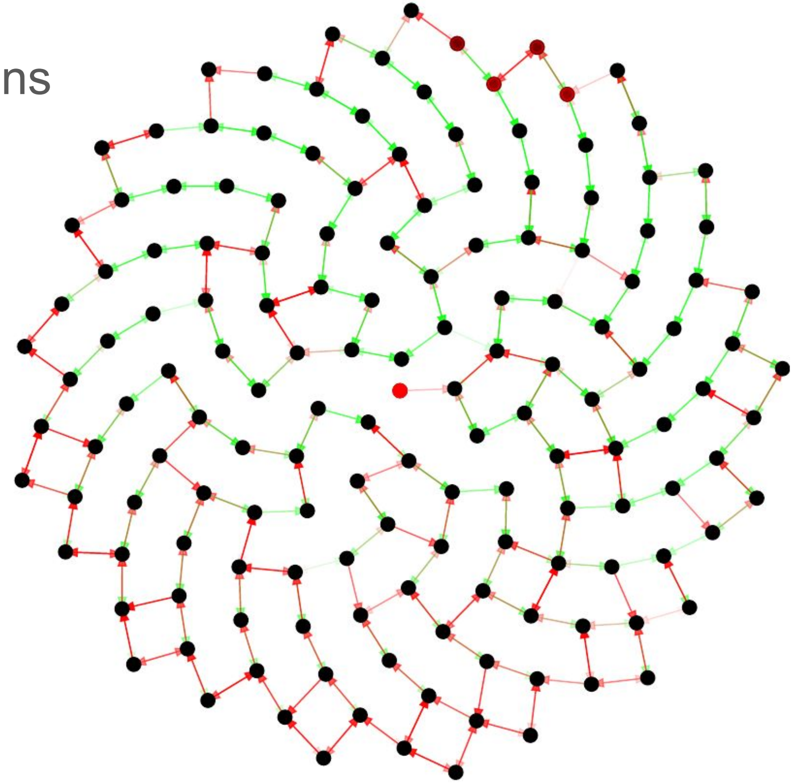
Neuroevolution

the best network has only local connections

what about a locally connected random network?

⟹ fast neuroevolution alternative





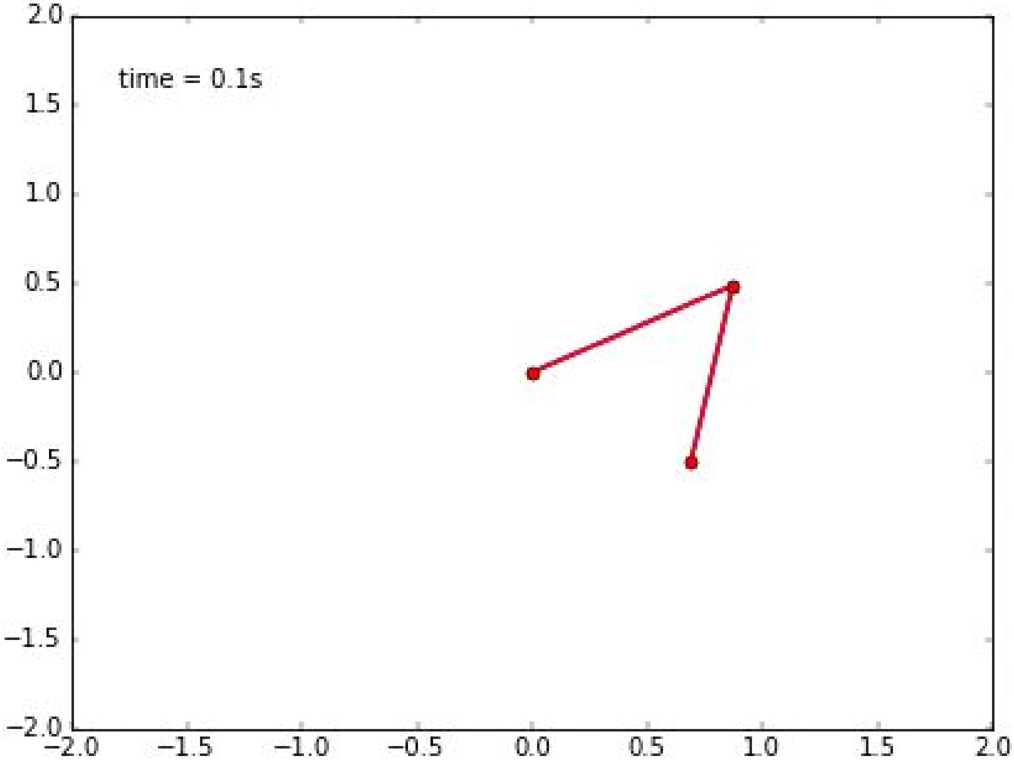
Dynamic Systems

recurrent networks are dynamic systems

$\Longrightarrow$ we can measure the amount of chaos (*Lyapunov exponent*)

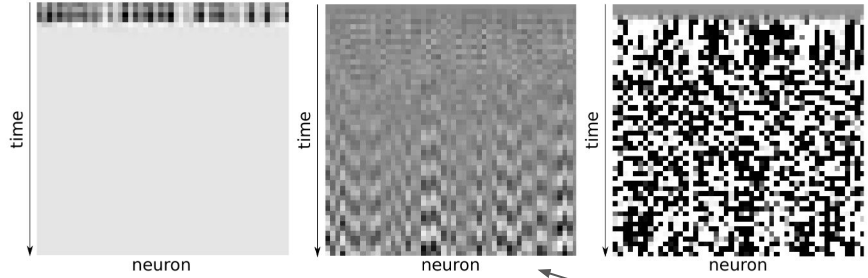
Sprott [2015]

Chaos

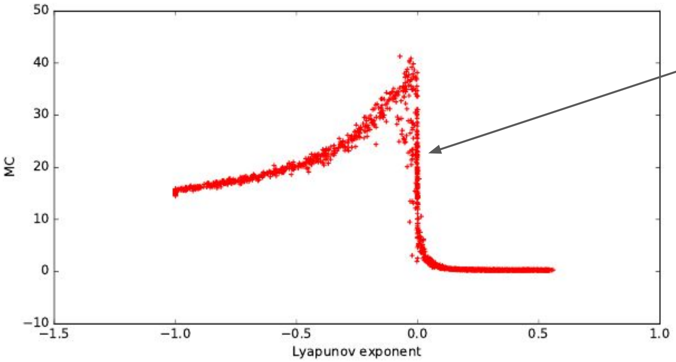


Chaos

recurrent networks have multiple levels of chaos



relation between dynamics and performance

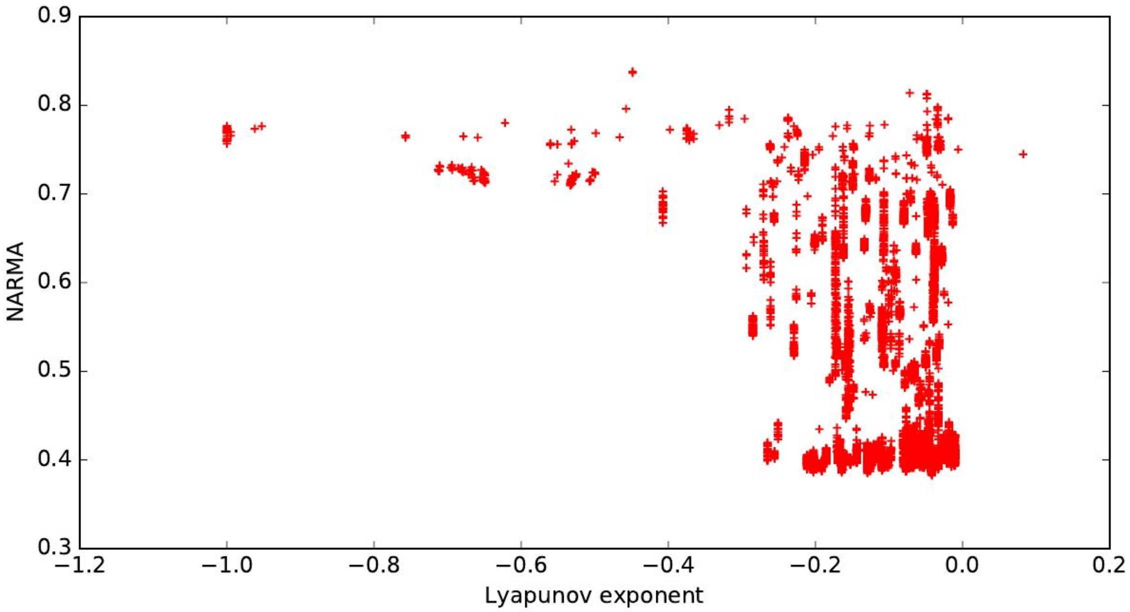


the edge of chaos

Bertschinger and Natschläger [2004], Boedecker et al. [2012]

Chaos

neuroevolution prefers to be close to the edge of chaos



Conclusion

neural networks represent a very strong artificial intelligence model

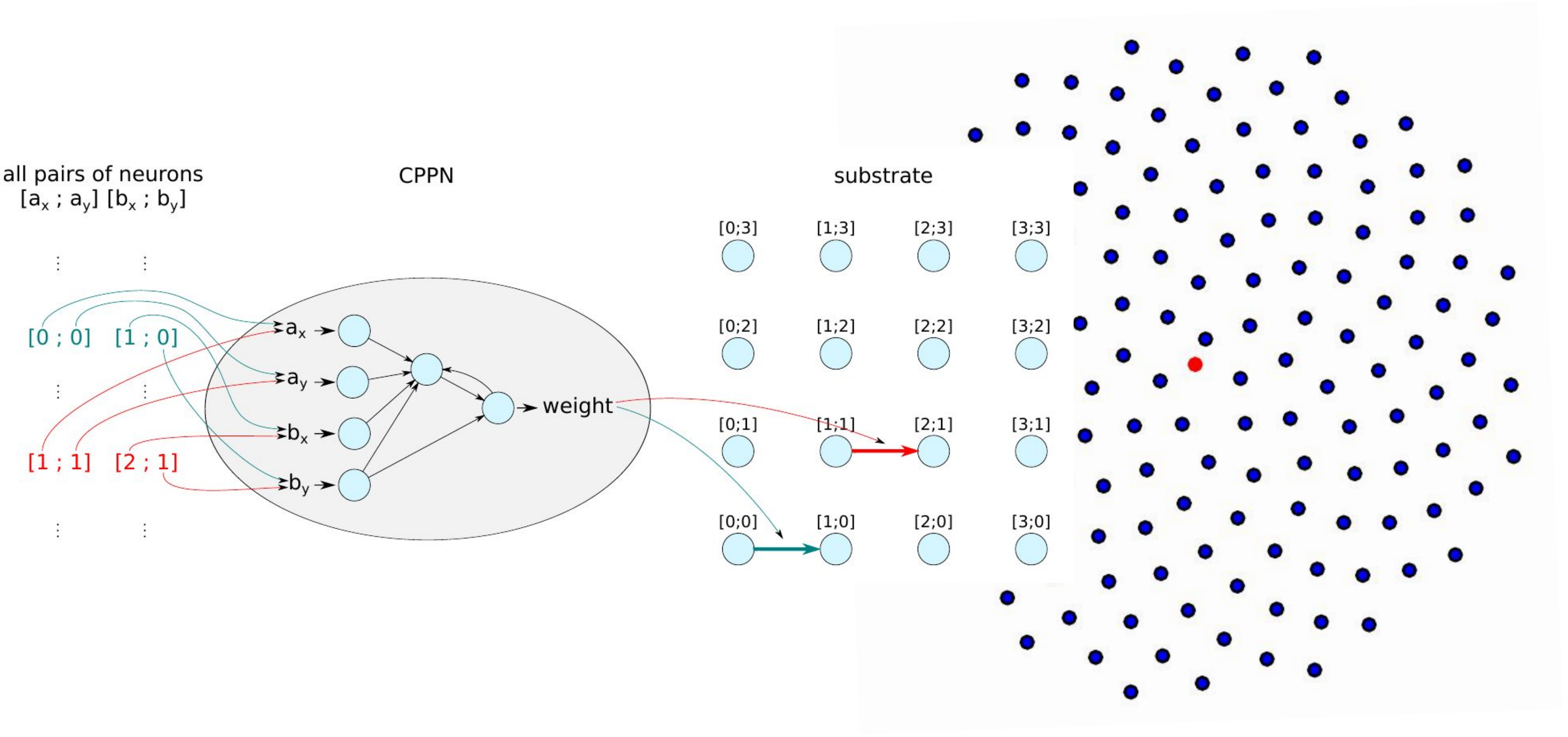
there has been a huge step forward in the last decade

recurrent networks still have a long way to go

neural networks are good in the same things as humans

neural networks are bad in the same things as humans

Questions?


all pairs of neurons
[a_x ; a_y] [b_x ; b_y]
CPPN
substrate
[0 ; 0] [1 ; 0]
[1 ; 1] [2 ; 1]
a_x
a_y
b_x
b_y
weight
[0;3] [1;3] [2;3] [3;3]
[0;2] [1;2] [2;2] [3;2]
[0;1] [1;1] [2;1] [3;1]
[0;0] [1;0] [2;0] [3;0]

Recommended Reading

1) Neural Networks and Deep Learning - Michael Nielsen
A well written online book covering the basic topics of feed-forward neural networks. Freely available.
http://neuralnetworksanddeeplearning.com/

2) ImageNet Classification with Deep Convolutional Neural Networks - Krizhevsky et al. [2012].
One of the first great successes of deep convolutional networks. Short and clear paper, however, it assumes the knowledge of backprop, max-pooling, dropout, etc.
http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

3) Deep Learning - Ian Goodfellow and Yoshua Bengio and Aaron Courville [in preparation, 2016]
A book in preparation, which is available on-line. It is written by a few of the best deep learning scientists and describes most of the up-to-date techniques.
http://www.deeplearningbook.org/

4) Deep Learning in Neural Networks: An Overview - Schmidhuber et al. [2014]
A huge survey of neural networks.
http://arxiv.org/pdf/1404.7828v4