Deep Recurrent Neural Networks

Artem Chernodub

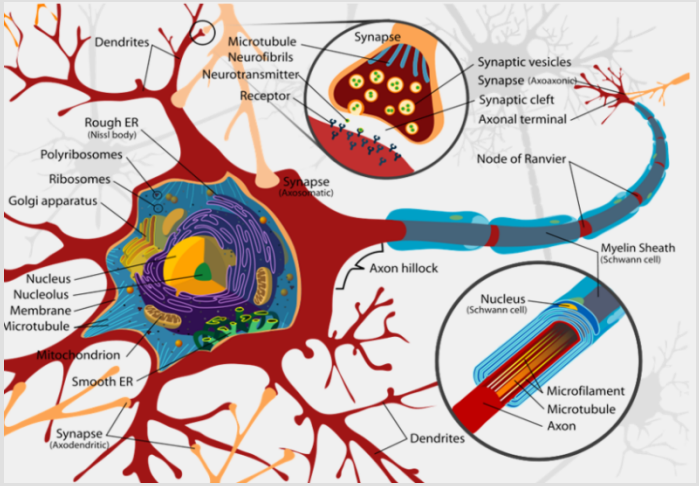
e-mail: a.chernodub@gmail.com
web: http://zzphoto.me

ZZ Photo

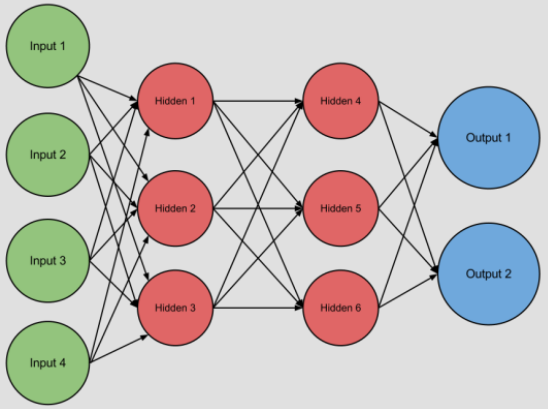
IMMSP NASU

Neuroscience

Biological-inspired models



Machine Learning



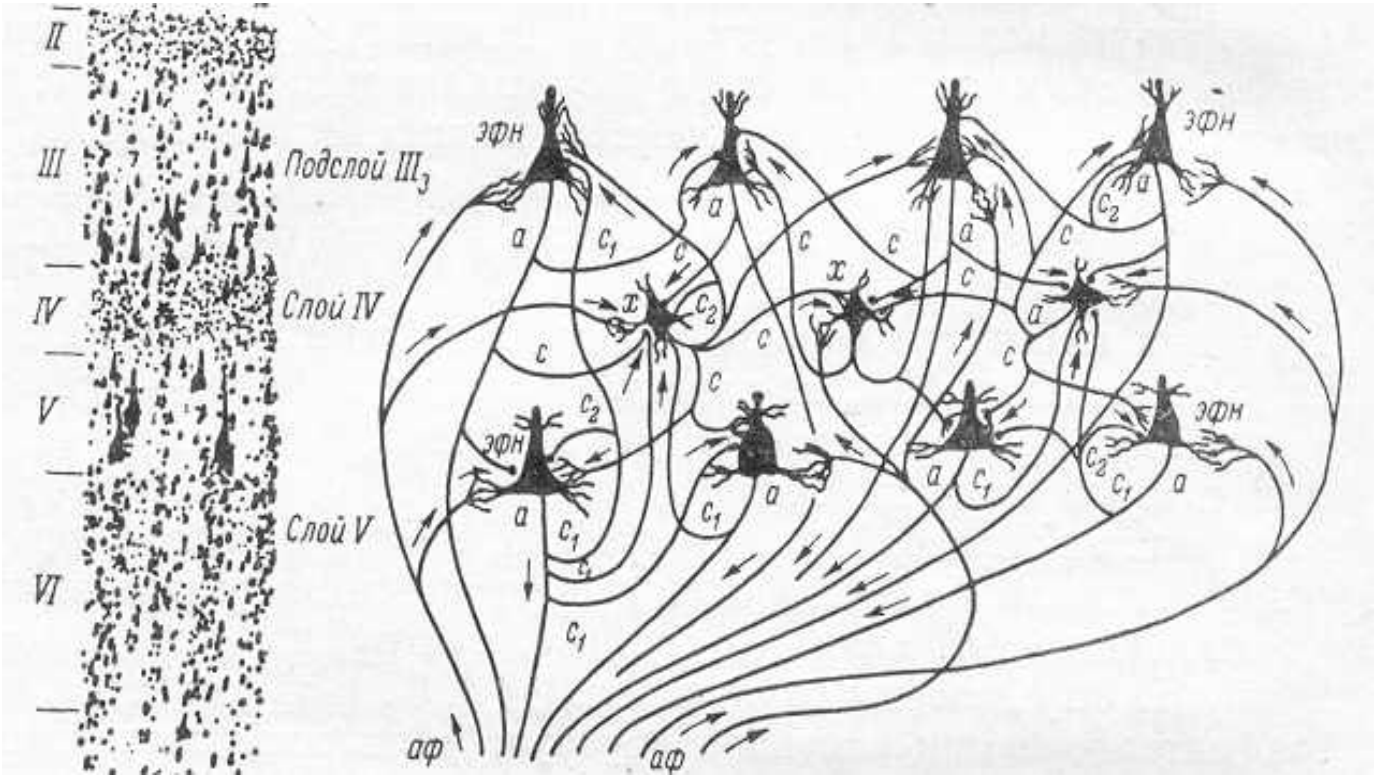
$$pxy = pyx\, p(x)/p(y)$$

Problem domain: processing the sequences

Neural Network Models under consideration

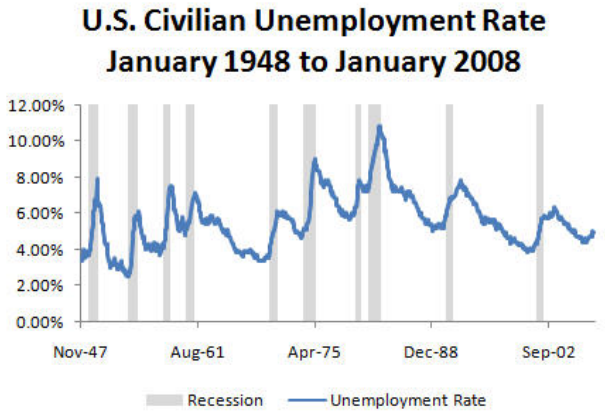
- Classic Feedforward (Shrink) Neural Networks.
- Feedforward Deep Neural Networks.
- Recurrent Neural Networks.

Biological Neural Networks

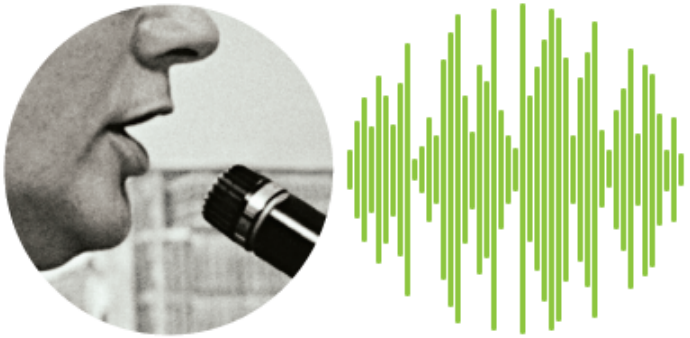


Sequence processing problems

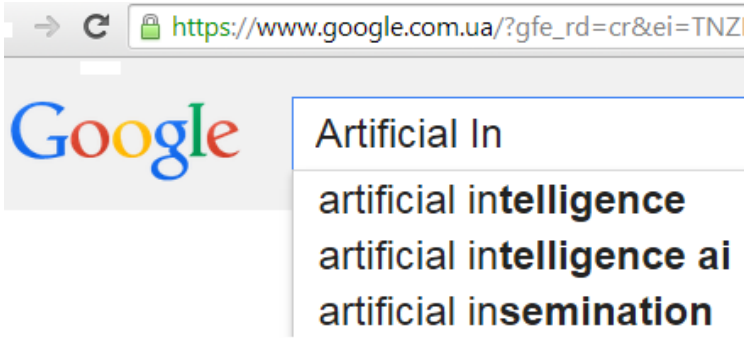
Time series predictions



Speech Recognition



Natural Language Processing



Engine control

Pen handwriting recognition (IAM-OnDB)

Model	# of parameters	Word Error Rate (WER)
HMM	-	35.5%
CTC RNN	13M	20.4%

Training data: 20, 000 most frequently occurring words written by 221 different writers.

A. Graves, S. Fernandez, M. Liwicki, H. Bunke, J. Schmidhuber. Unconstrained online handwriting recognition with recurrent neural networks // Advances in Neural Information Processing Systems 21, NIPS'21, p 577-584, 2008, MIT Press, Cambridge, MA, 2008.

TIMIT Phoneme Recognition

Model	# of parameters	Error
Hidden Markov Model, HMM	-	27,3%
Deep Belief Network, DBN	~ 4M	26,7%
Deep RNN	4,3M	17.7%

Training data: 462 speakers train / 24 speakers test, 3.16 / 0.14 hrs.

Graves, A., Mohamed, A.-R., and Hinton, G. E. (2013). Speech recognition with deep recurrent neural networks // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 6645–6649. IEEE.

Mohamed, A. and Hinton, G. E. (2010). Phone recognition using restricted Boltzmann machines // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 4354–4357.

Google Large Vocabulary Speech Recognition

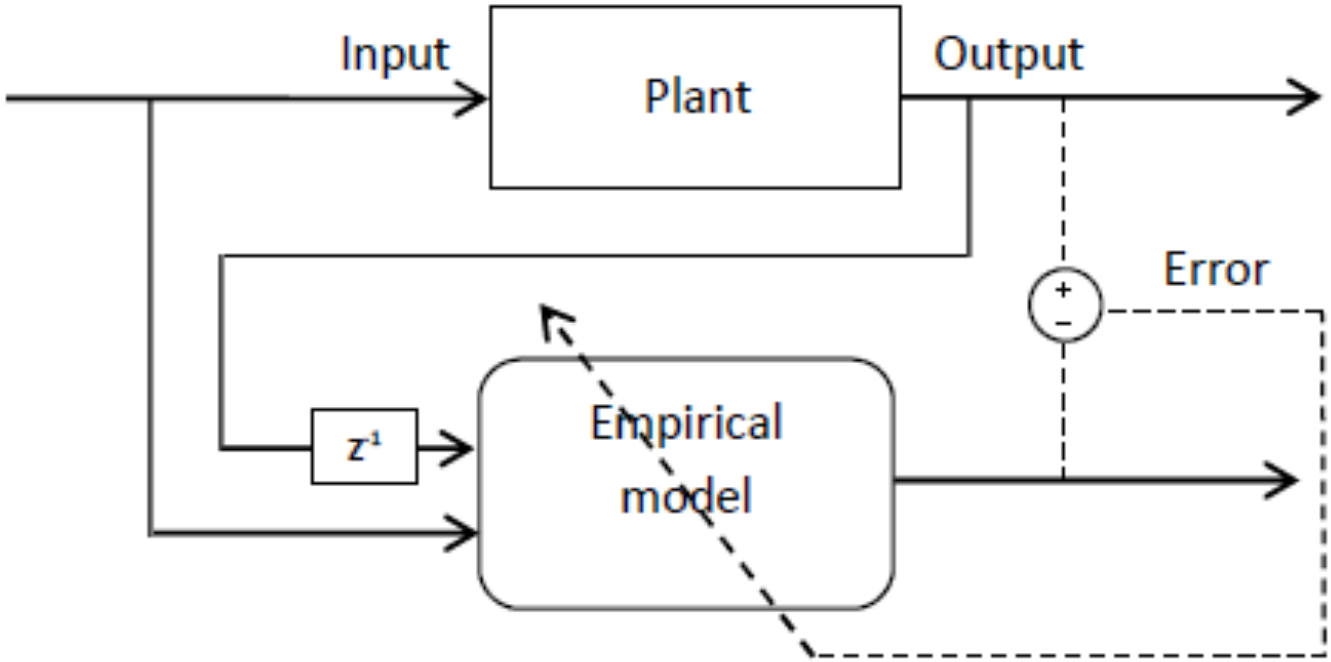
Model	# of parameters	Cross-entropy
ReLU DNN	85M	11.3
Deep Projection LSTM RNN	13M	10.7

Training data: 3M utterances (1900 hrs).

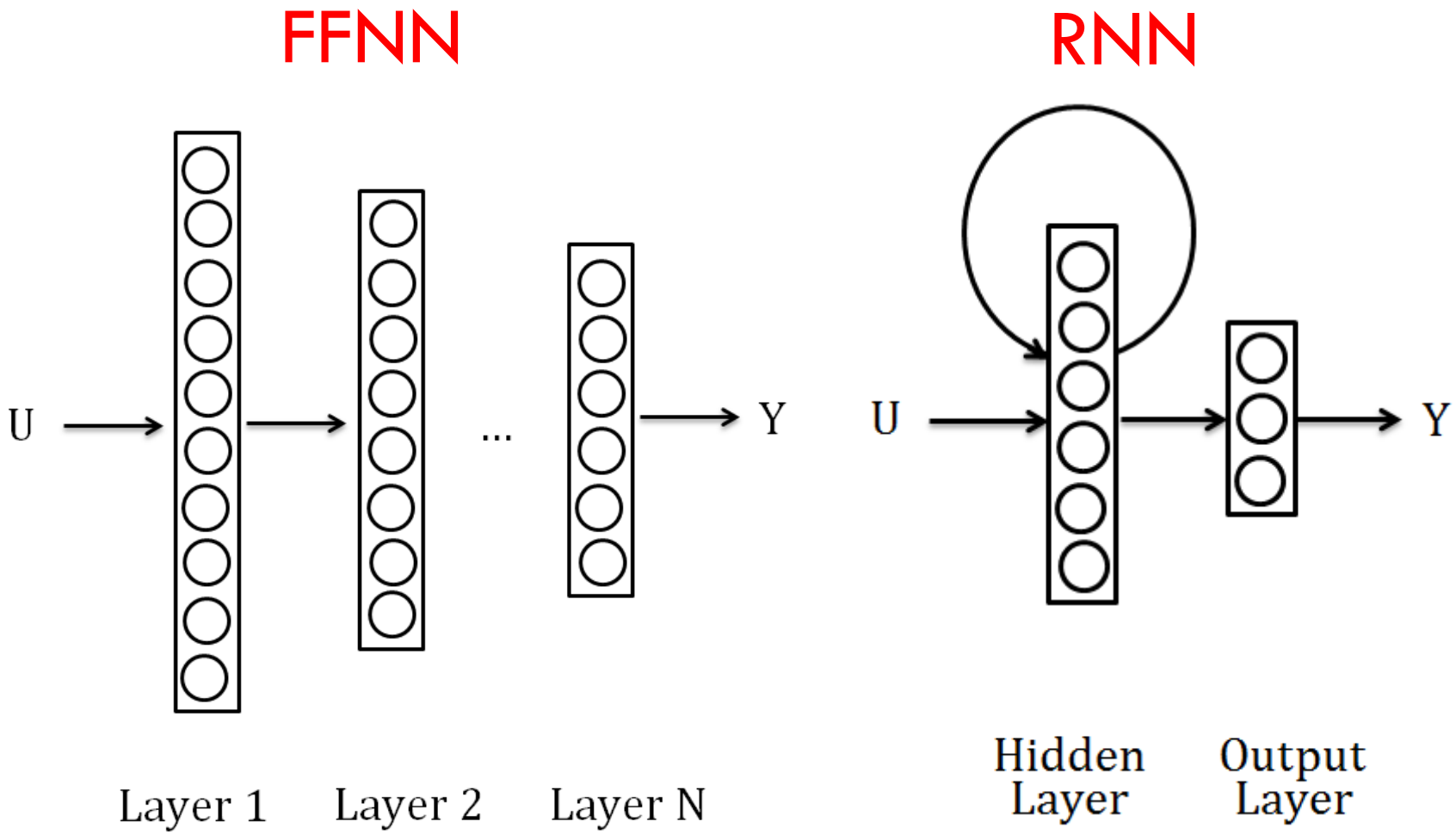
H. Sak, A. Senior, F. Beaufays. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling // INTERSPEECH'2014.

K. Vesely, A. Ghoshal, L. Burget, D. Povey. Sequence-discriminative training of deep neural networks // INTERSPEECH'2014.

Dynamic process identification scheme

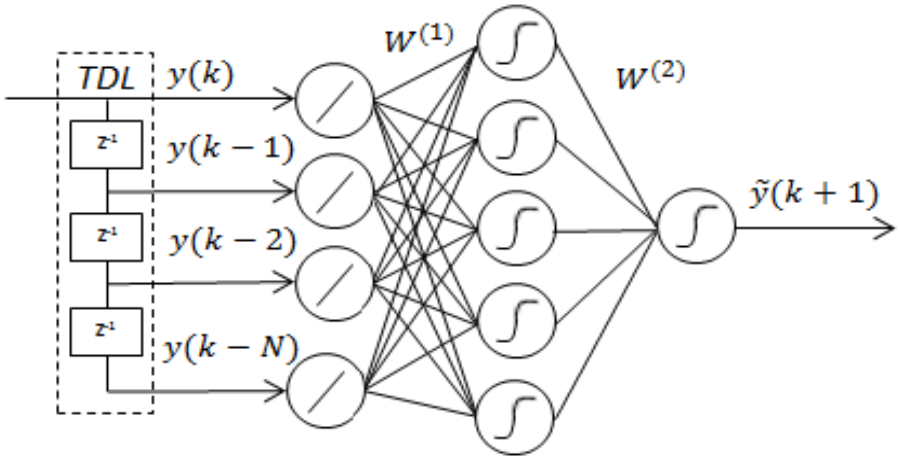


Feedforward and Recurrent Neural Networks: popularity 90% / 10%

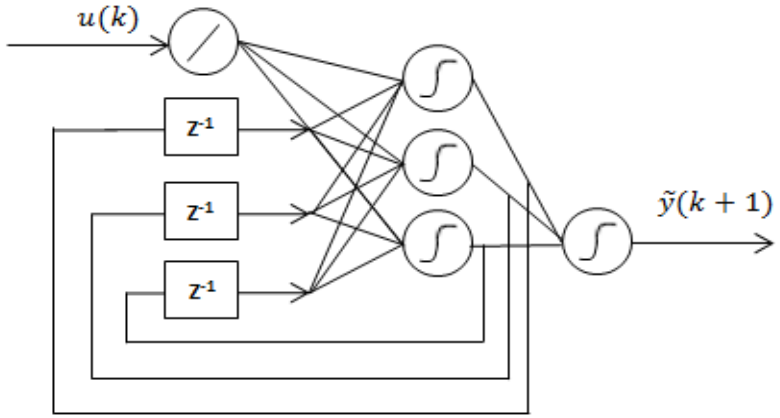


Feedforward vs Recurrent Neural Networks

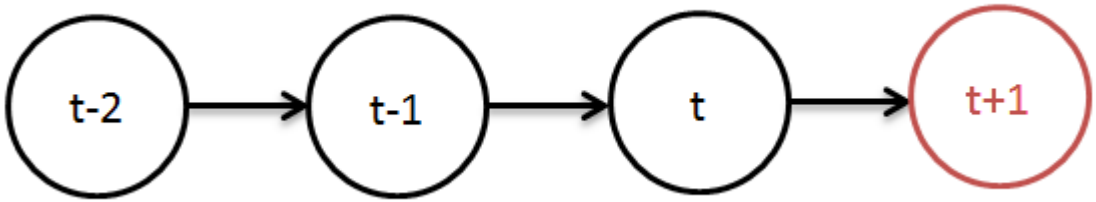
Dynamic Multilayer Perceptron, DMLP (FFNN)



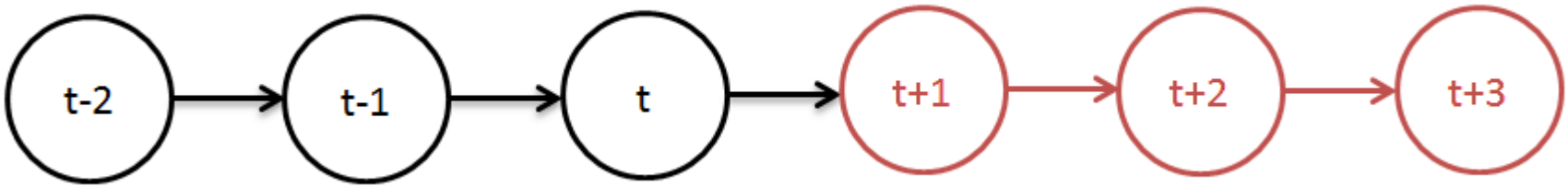
Recurrent Multilayer Perceptron, RMLP (RNN)



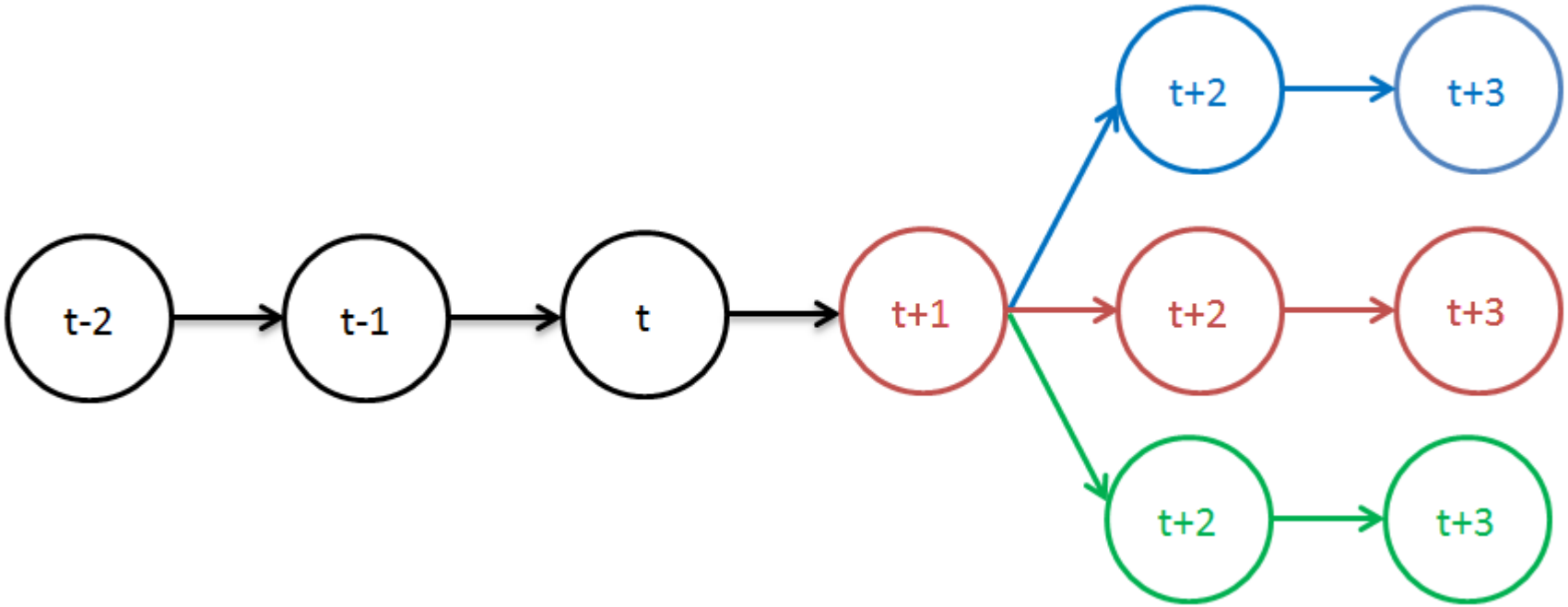
Processing the sequences: Single-Step-Ahead predictions



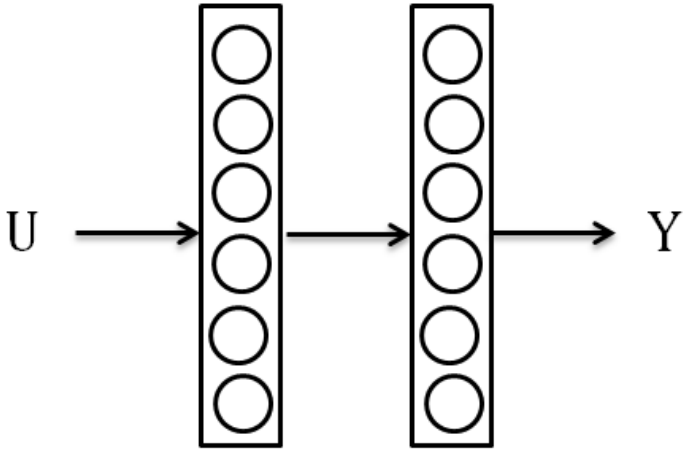
Processing the sequences: Multi-Step-Ahead predictions, autoregression



Processing the sequences: Multi-Step-Ahead predictions under control

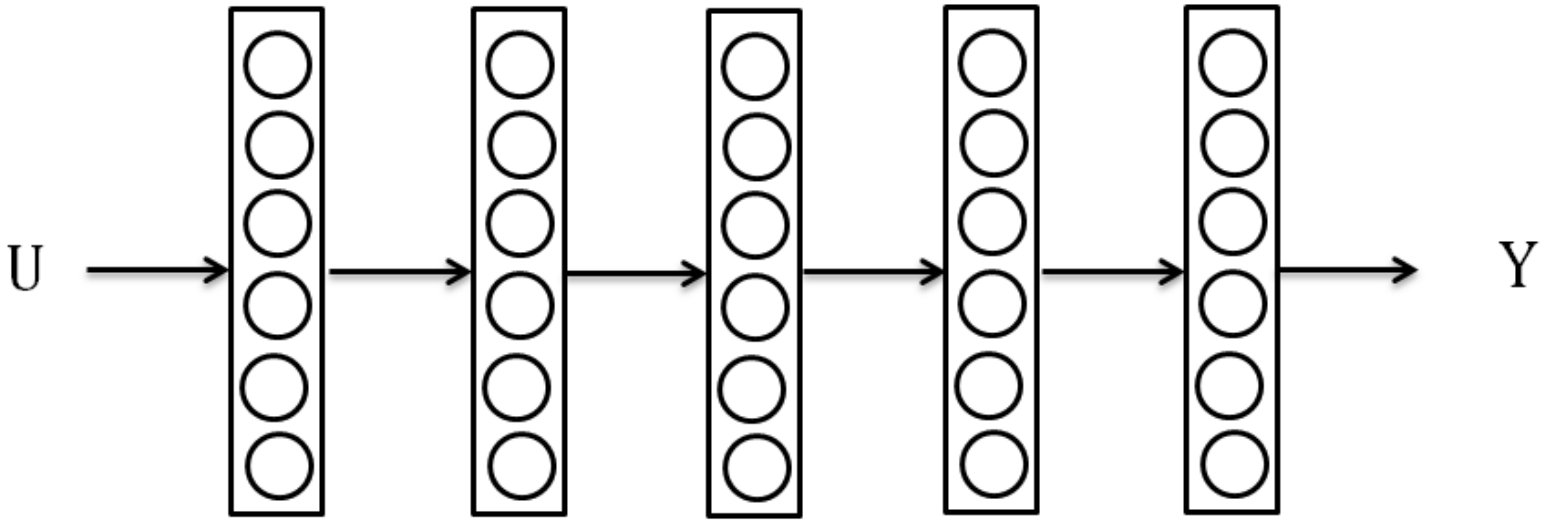


Classic Feedforward Neural Networks (before 2006).



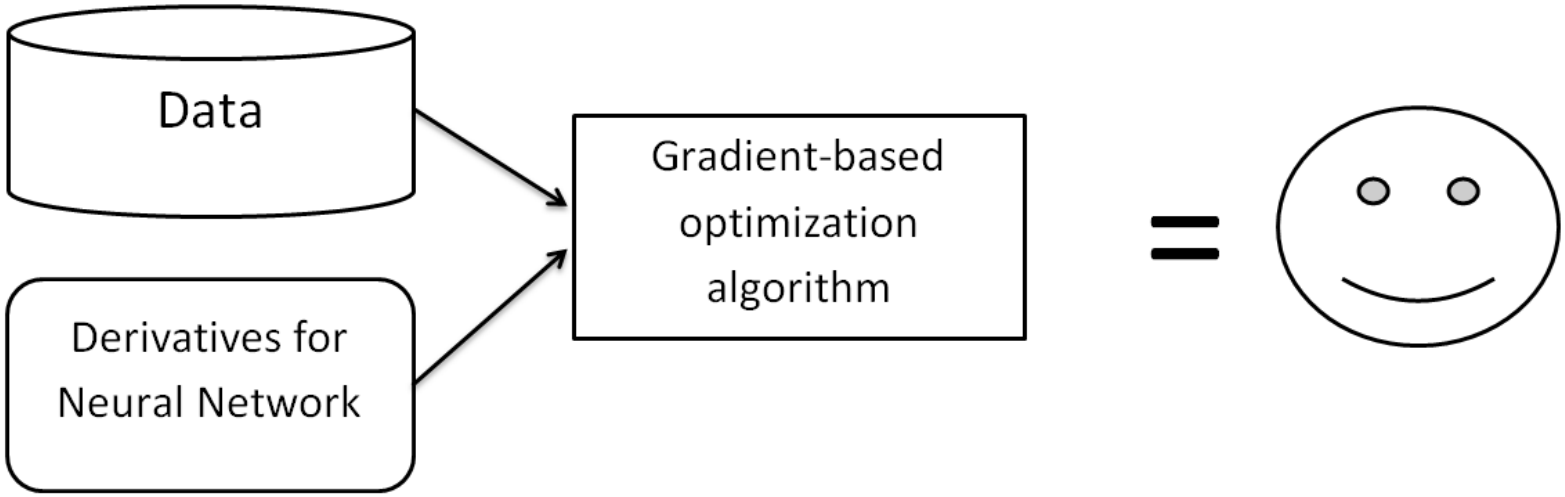
- Single hidden layer (Kolmogorov-Cybenko Universal Approximation Theorem as the main hope).
- Vanishing gradients effect prevents using more layers.
- Less than 10K free parameters.
- Feature preprocessing stage is often critical.

Deep Feedforward Neural Networks

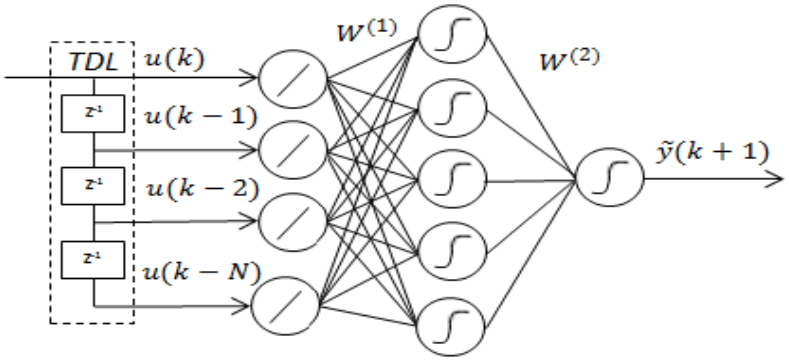


- Number of hidden layers > 1 (usually 6-9).
- 100K – 100M free parameters.
- No (or less) feature preprocessing stage.
- 2-stage training process: i) unsupervised pre-training; ii) fine tuning (vanishing gradients problem is beaten!).

Training the Neural Network: derivative + optimization



DMLP: 1) forward propagation pass



$$z_j = f(\sum_i w_{ji}^{(1)} x_i),$$

$$\tilde{y}(k+1) = g(\sum_j w_j^{(2)} z_j),$$

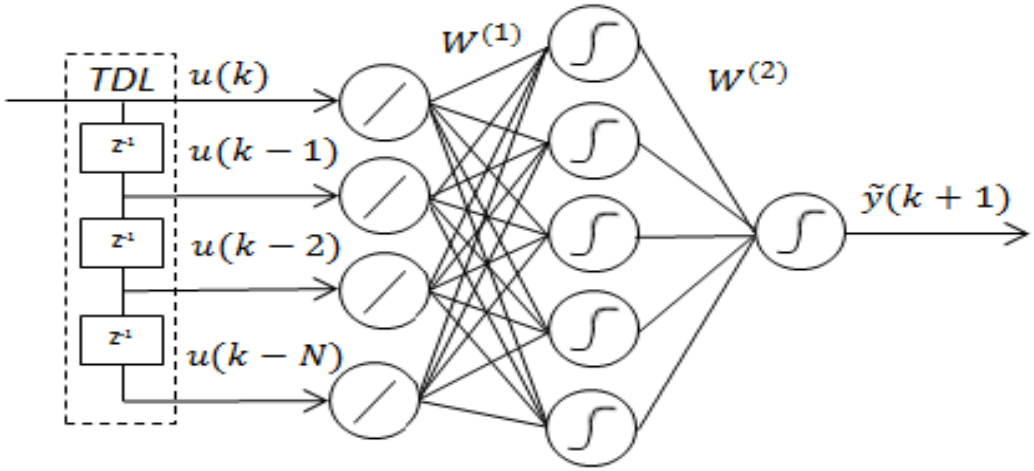
where z_j is the postsynaptic value for the j-th hidden neuron, $w^{(1)}$ are the hidden layer's weights, $f()$ are the hidden layer's activation functions, $w^{(2)}$ are the output layer's weights, and $g()$ are the output layer's activation functions.

DMLP: 2) backpropagation pass

Local gradients calculation:

$$\delta^{OUT} = t(k+1) - \tilde{y}(k+1),$$

$$\delta_j^{HID} = f'(z_j) w_j^{(2)} \delta^{OUT}.$$



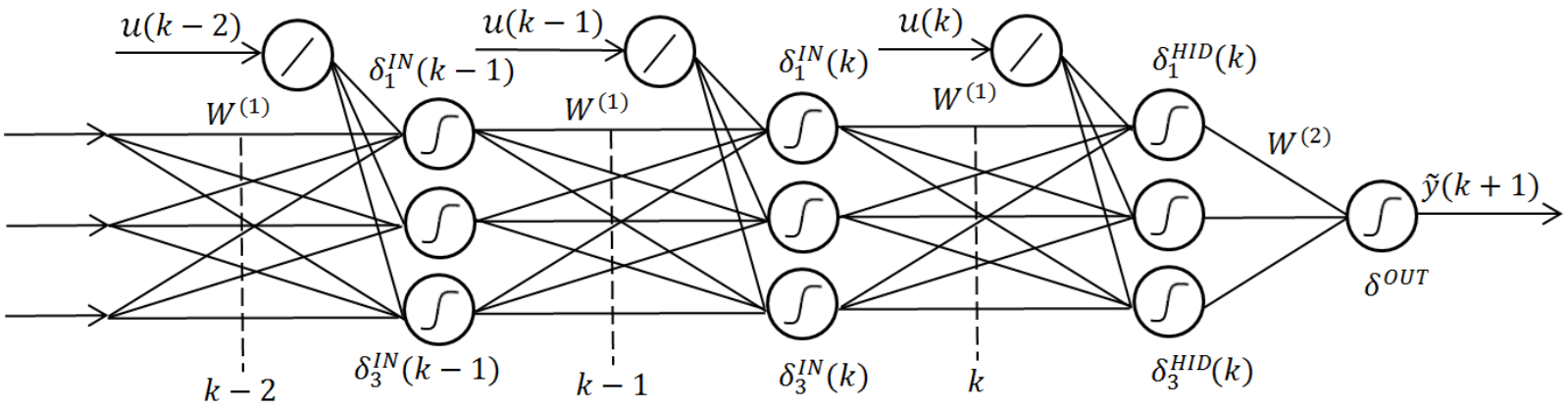
Derivatives calculation:

$$\frac{\partial E(k)}{\partial w_j^{(2)}} = \delta^{OUT} z_j,$$

$$\frac{\partial E(k)}{\partial w_{ji}^{(1)}} = \delta_j^{IN} x_i.$$

Backpropagation Through Time (BPTT)

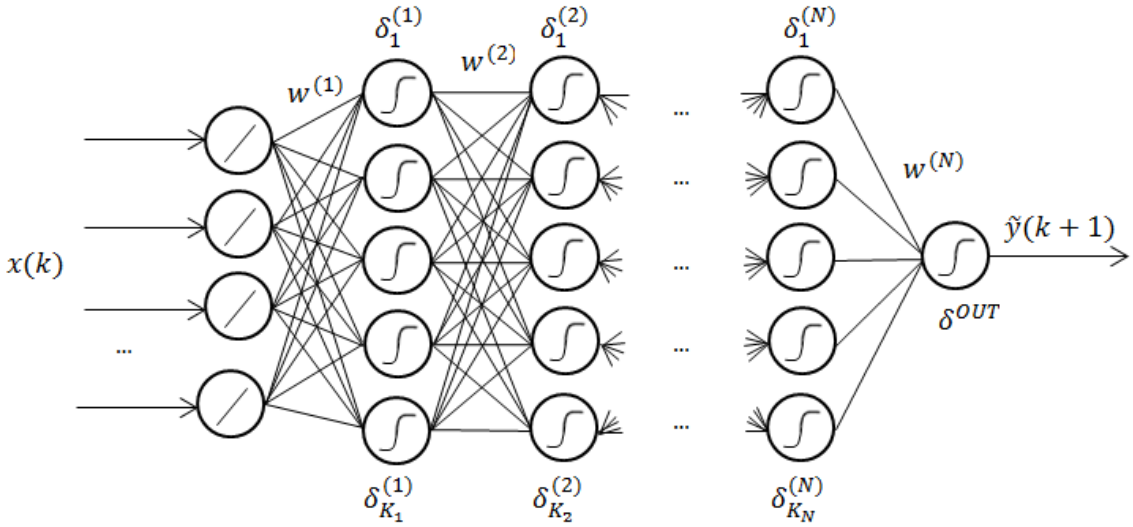
BPTT = unrolling RNN to Deep Feedforward Neural Network



$$\frac{\partial E_{BPTT}(k)}{\partial w_{ji}^{(1)}} = \frac{1}{h}\sum_{n=0}^{h}\frac{\partial E(k-n)}{\partial w_{ji}^{(1)}}.$$

where h = truncation depth.

Bad effect of vanishing (exploding) gradients: a problem



$$\delta_j^{(m)} = f_j^{(m-1)\prime} \sum_i w_{ij}^{(m)} \delta_i^{(m+1)},$$

$$\frac{\partial E(k)}{\partial w_{ji}^{(m)}} = \delta_j^{(m)} z_i^{(m-1)},$$

$$\Rightarrow \qquad \frac{\partial E(k)}{\partial w_{ji}^{(m)}} \to 0 \ \ for\, m \to 1$$

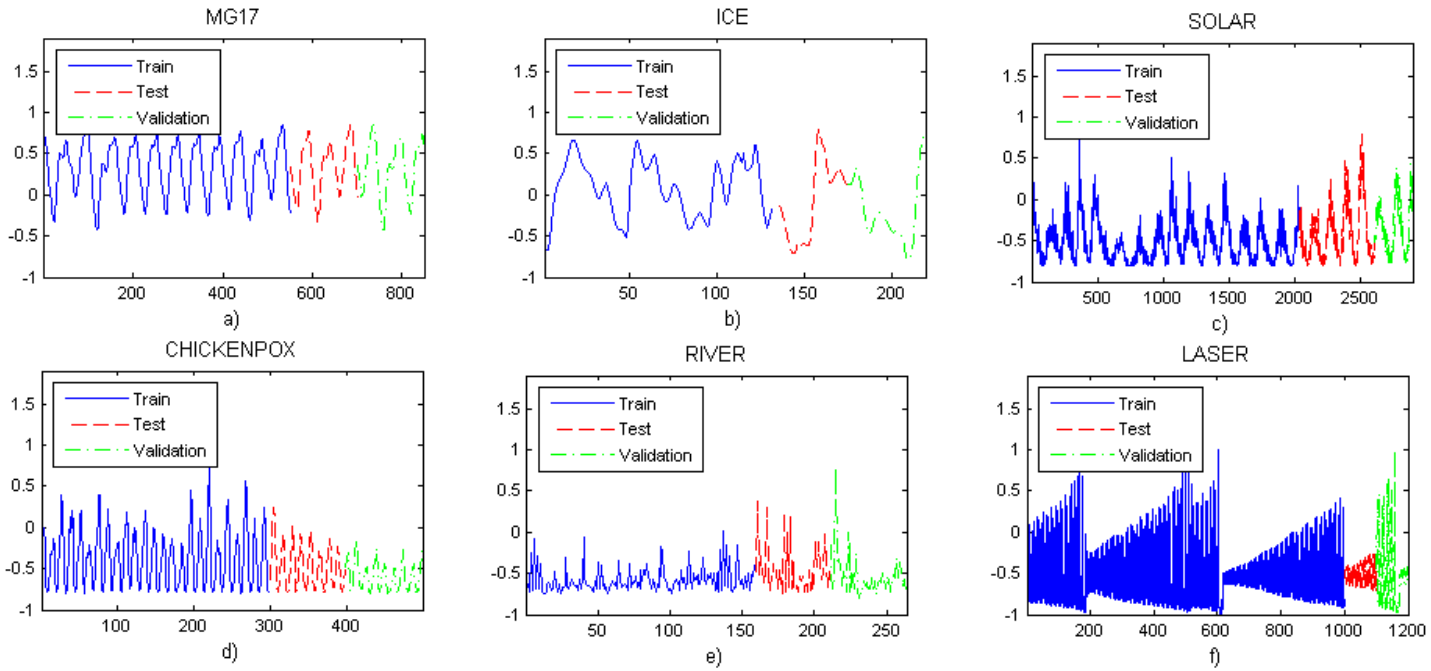
Vanishing gradients effect: solutions

- *Accurate initialization of weights:*
  - if the weights are small, the gradients shrink exponentially;
  - if the weights are big the gradients grow exponentially.
- *Hessian Free Optimization: use new optimizer by Martens.*

J. Martens. Deep Learning via Hessian-Free Optimization // Proc. International Conference of Machine Learning, pp. 735-742, 2010.

J. Martens and I. Sutskever. Learning Recurrent Neural Networks with Hessian-Free Optimization // Proc. ICML, 2011.

Time-series Single-Step-Ahead prediction problems



A. Chernodub. Training Dynamic Neural Networks Using the Extended Kalman Filter for Multi-Step-Ahead Predictions // Artificial Neural Networks Methods and Applications in Bio-/Neuroinformatics. Petia Koprinkova-Hristova, Valeri Mladenov, Nikola K. Kasabov (Eds.), Springer, 2014, pp. 221-244.

Time-series Single-Step-Ahead predictions: the results

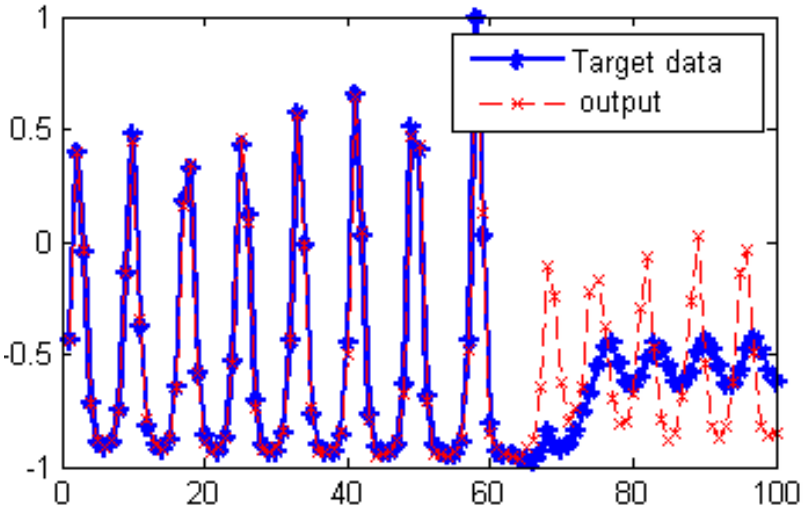
	MG17	ICE	SOLAR	CHICKENPOX	RIVER	LASER
DLNN (FF)	0.0042	0.0244	0.1316	0.8670	4.3638	0.7711
DMLP (FF)	0.0006	0.0376	0.1313	0.4694	0.9979	0.0576
RMLP (RNN)	0.0050	0.0273	0.1493	0.7608	6.4790	0.2415
NARX (RNN)	0.0010	0.1122	0.1921	1.4736	2.0685	0.1332

Mean NMSE Single-Step-Ahead predictions for different neural network architectures

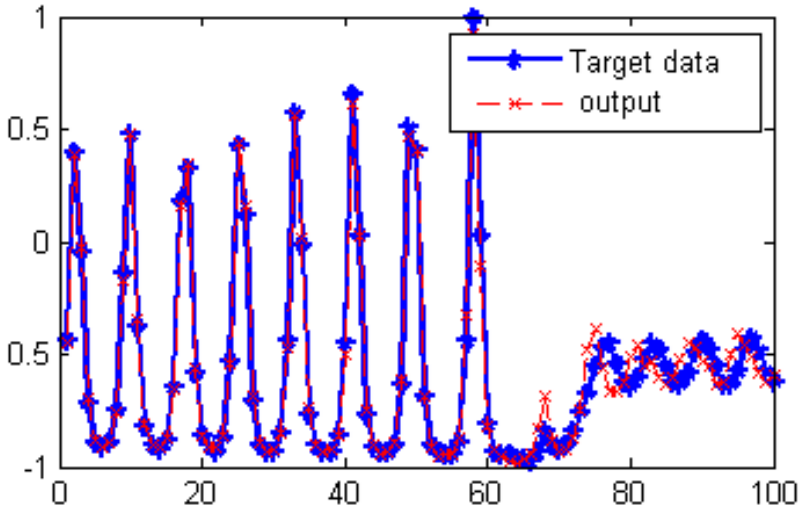
A. Chernodub. Training Dynamic Neural Networks Using the Extended Kalman Filter for Multi-Step-Ahead Predictions // Artificial Neural Networks Methods and Applications in Bio-/Neuroinformatics. Petia Koprinkova-Hristova, Valeri Mladenov, Nikola K. Kasabov (Eds.), Springer, 2014, pp. 221-244.

?!!

Multi-Step-Ahead prediction, Laser Dataset



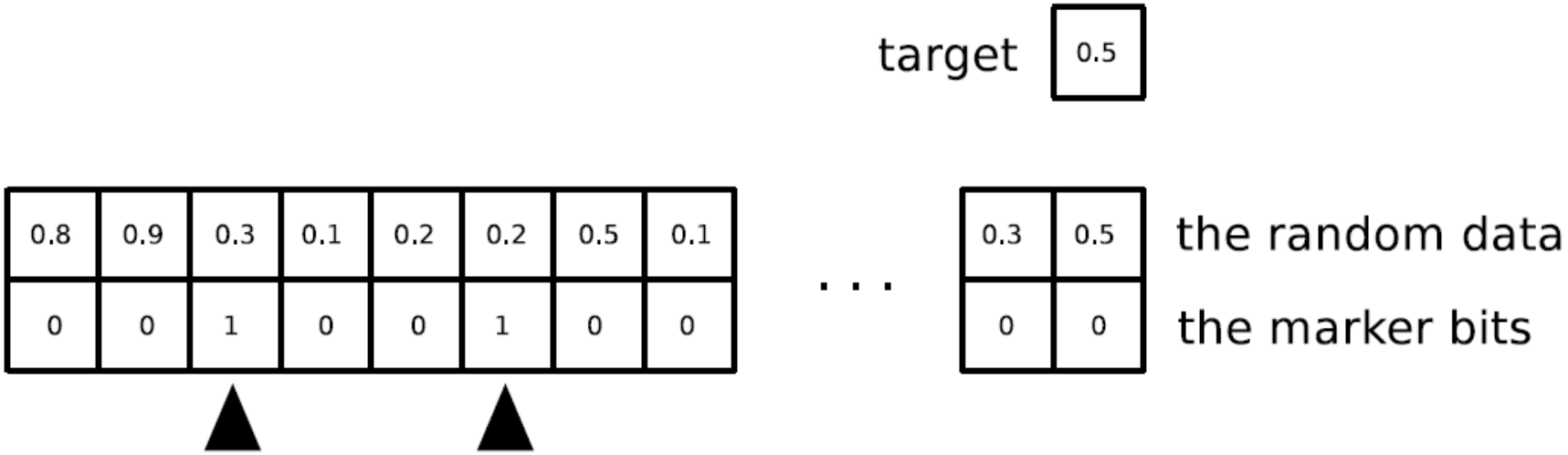
feedforward data flow



recurrent data flow

A. Chernodub. Training Dynamic Neural Networks Using the Extended Kalman Filter for Multi-Step-Ahead Predictions // Artificial Neural Networks Methods and Applications in Bio-/Neuroinformatics. Petia Koprinkova-Hristova, Valeri Mladenov, Nikola K. Kasabov (Eds.), Springer, 2014, pp. 221-244.

Long-term Dependencies catching: T = 50, 100, 200 …

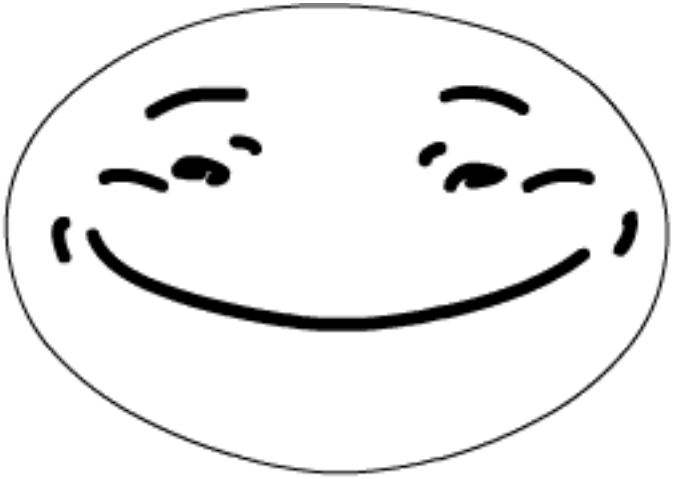


R. Pascanu, Y. Bengio. On the Difficulty of Training Recurrent Neural Networks // Tech. Rep. arXiv:1211.5063, Universite de Montreal (2012).

RNNvs FFNN

	Deep Feed-Forward NNs	*Recurrent NNs*
Pre-training stage	Yes	No (rarely)
Derivatives calculation	Backpropagation (BP)	Backpropagation Through Time (BPTT)
Multi-Step-Ahead Prediction Quality	Bad (-)	Good (+)
Sequence Recognition Quality	Bad (-)	Good (+)
Vanishing gradient effect	Present (-)	Present (-)

Thanks!



e-mail: a.chernodub@gmail.com
web: http://zzphoto.me