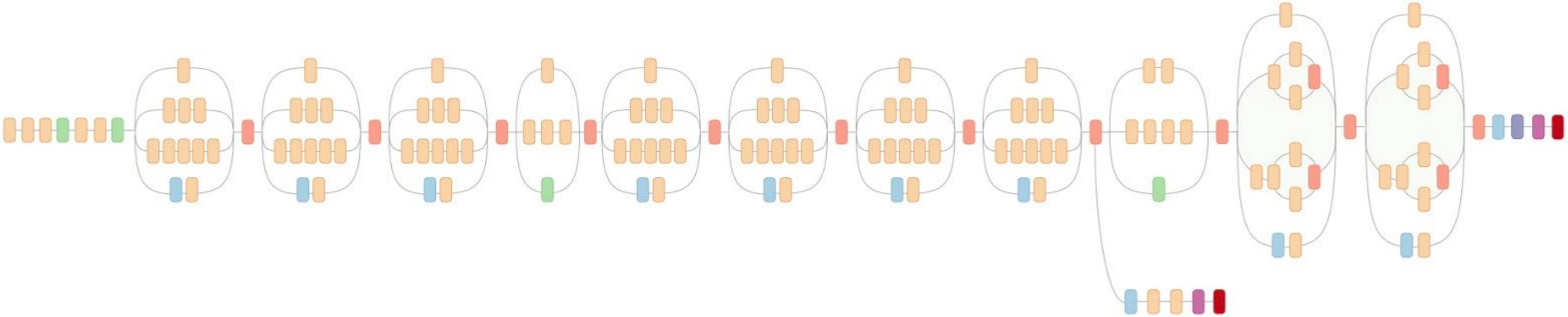
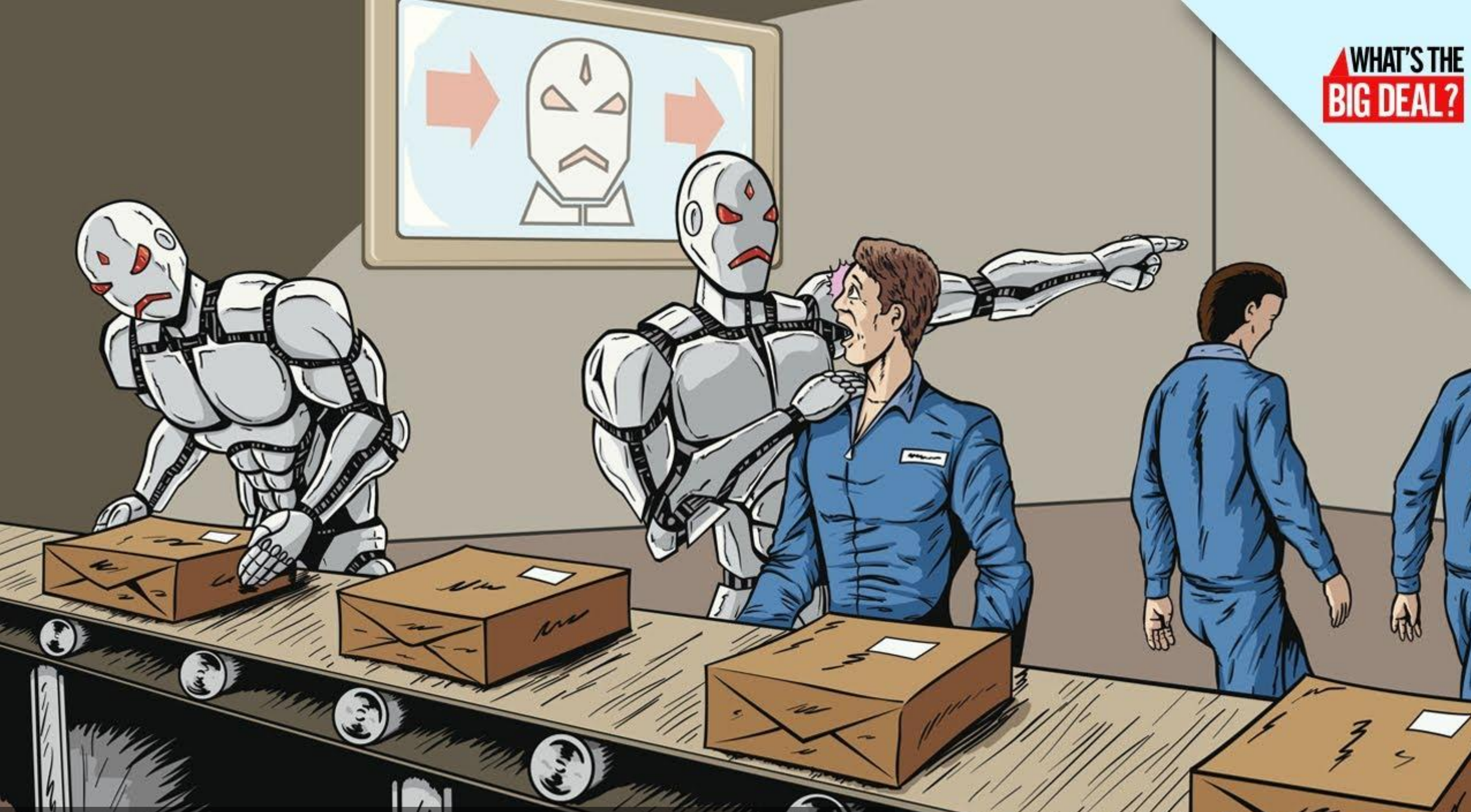


Deep Learning

Sigurd Spieckermann
Siemens Corporate Technology

2016/06/01

SOURCE: http://www.clickode.com/en/2016/02/01/google-rilascia-gratis-lezioni-di-deep-learning/

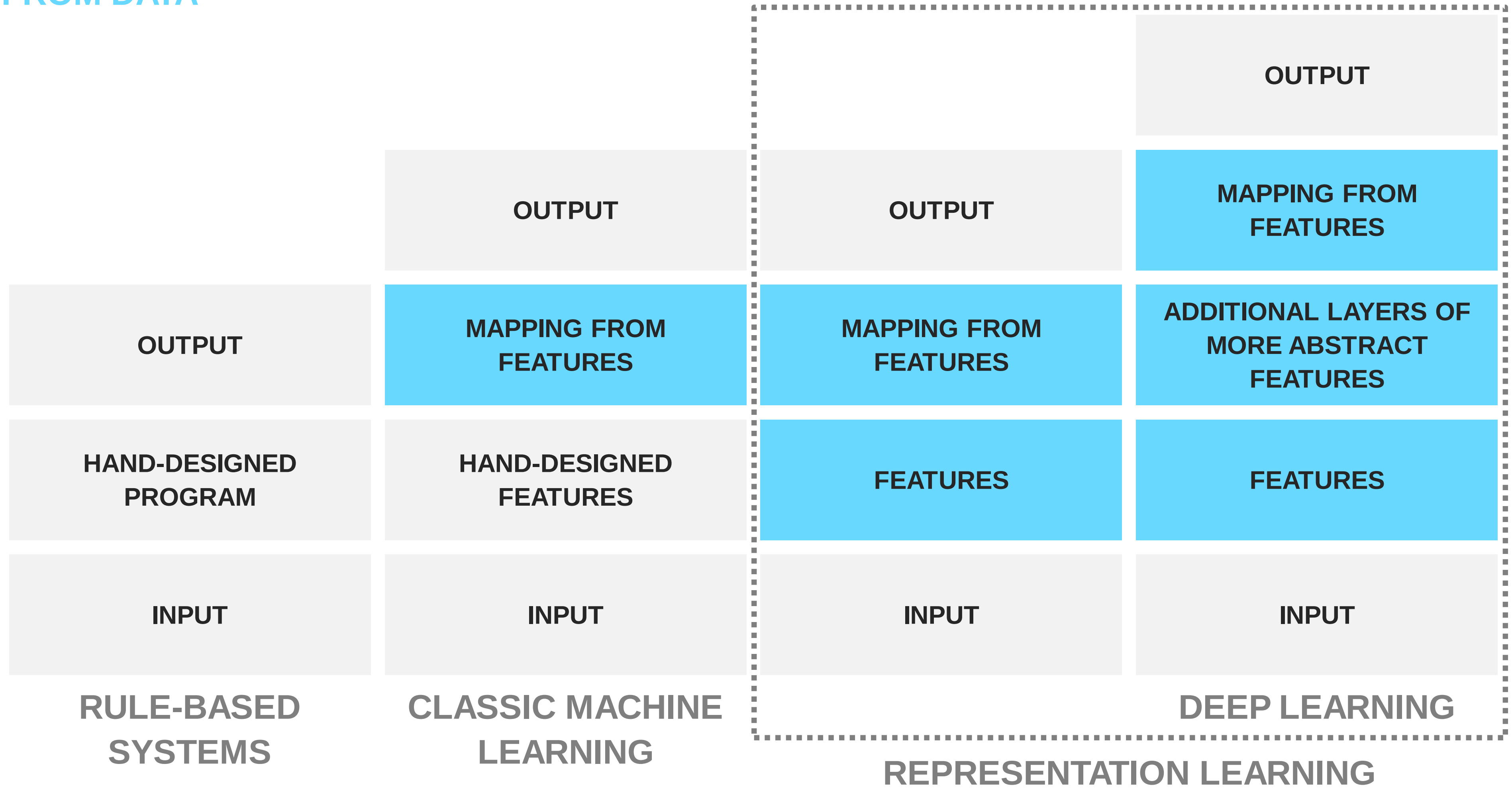


SOURCE: http://community.mis.temple.edu/mis4596sec003sp2016/2016/01/31/will-robots-take-over-your-job/

OK, LET’S GET SERIOUS NOW …

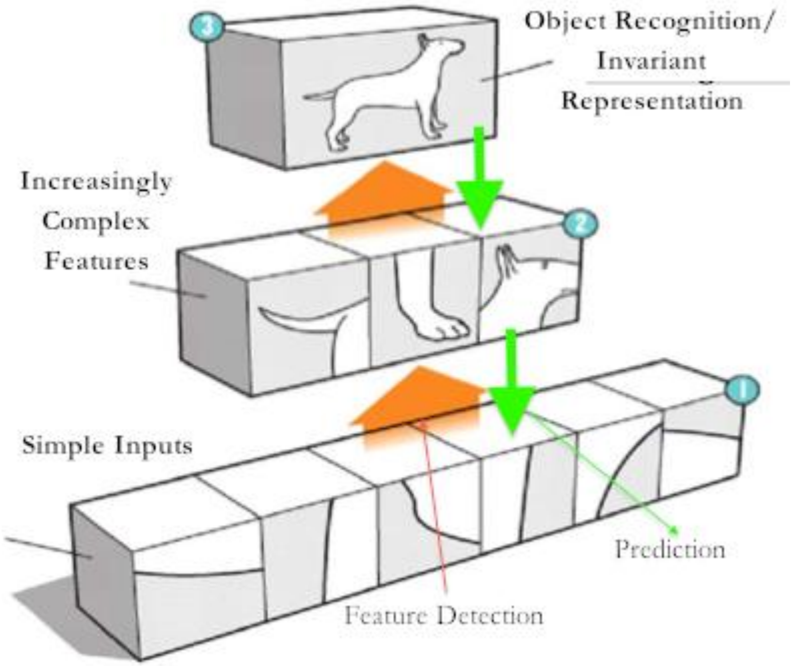
Deep Learning vs. Classic Data Modeling

LEARNED FROM DATA



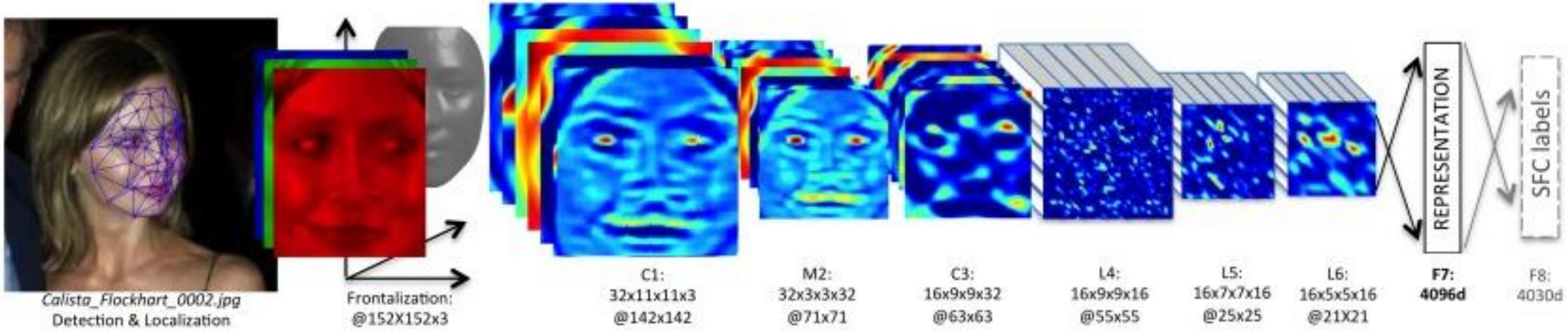
SOURCE: http://www.deeplearningbook.org/contents/intro.html

Deep Learning
Hierarchical Feature Extraction



SOURCE: http://www.eidolonspeak.com/Artificial_Intelligence/SOA_P3_Fig4.png

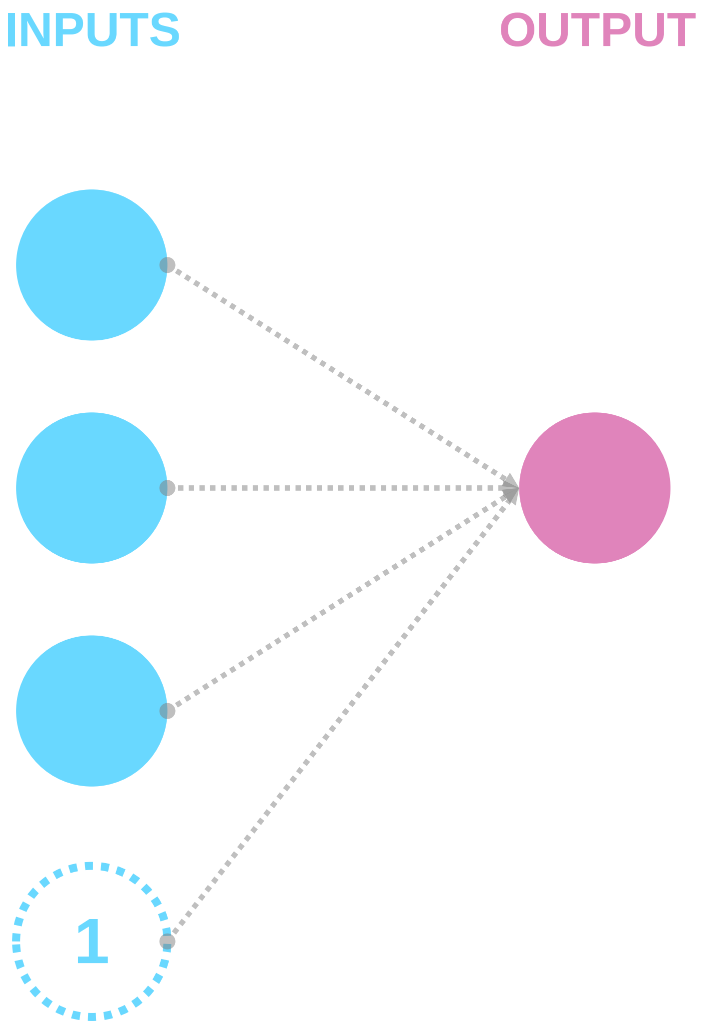
Deep Learning
Hierarchical Feature Extraction



SOURCE:
Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1701-1708).

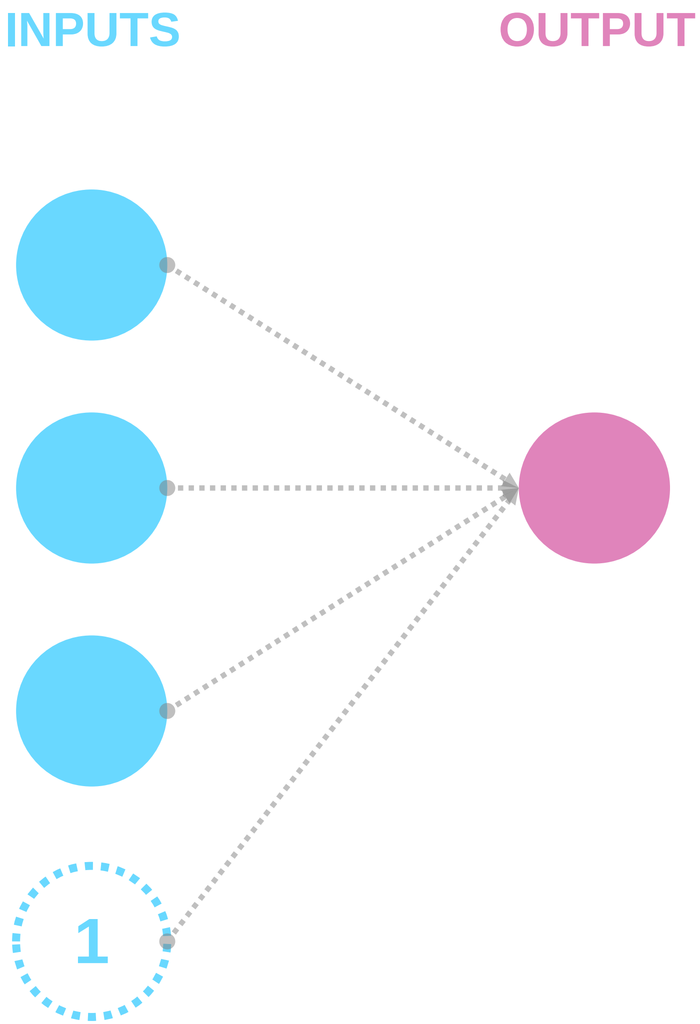
NEURAL NETWORKS

Neural Networks
Linear Regression



$$\hat{y} = w^T x + b$$

Neural Networks
Logistic Regression

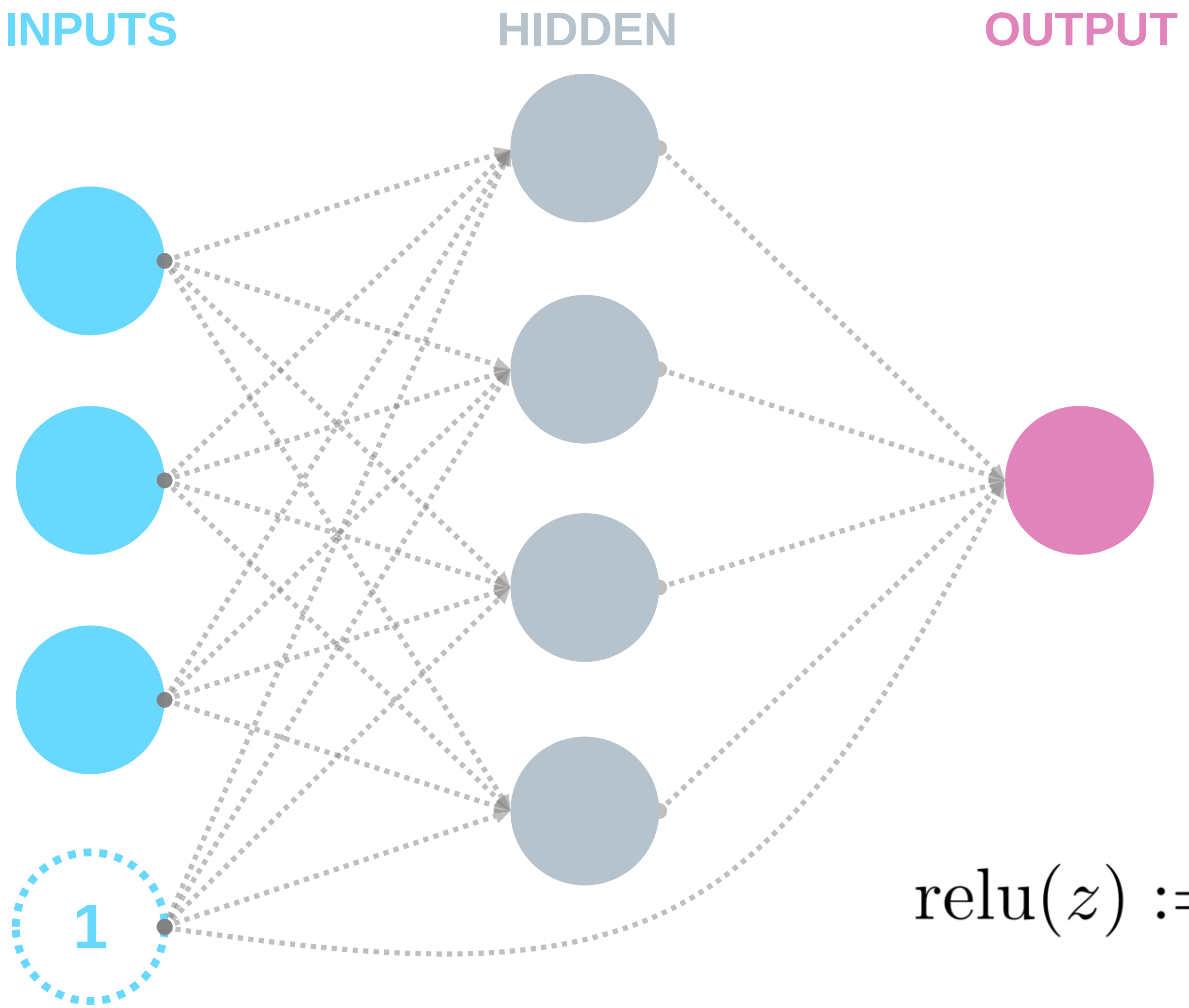


$$\hat{y} = \text{logistic}(w^T x + b)$$

$$\text{logistic}(z) = \frac{1}{1 + e^{-z}}$$

Neural Networks

Fully Connected Feedforward Neural Network



$$h = \phi(W^{(1)}x + b^{(1)})$$

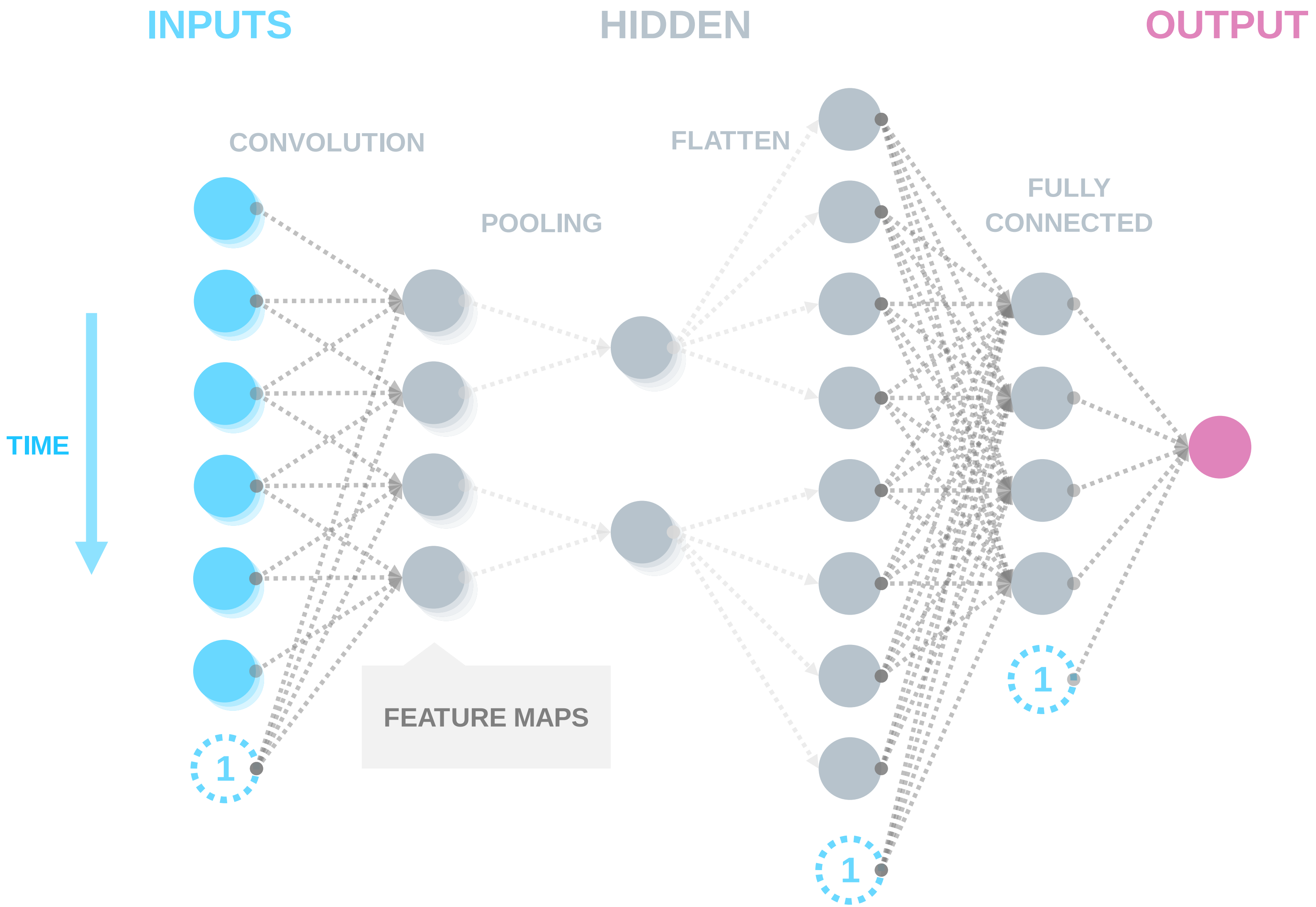
$$\hat{y} = \text{logistic}(W^{(2)}h + b^{(2)})$$

$$\phi(z) = \begin{cases} \tanh(z) \\ \text{relu}(z) \\ \dots \end{cases}$$

$$\text{relu}(z) := \max(0, z) = \begin{cases} z & z > 0 \\ 0 & z \leq 0 \end{cases}$$

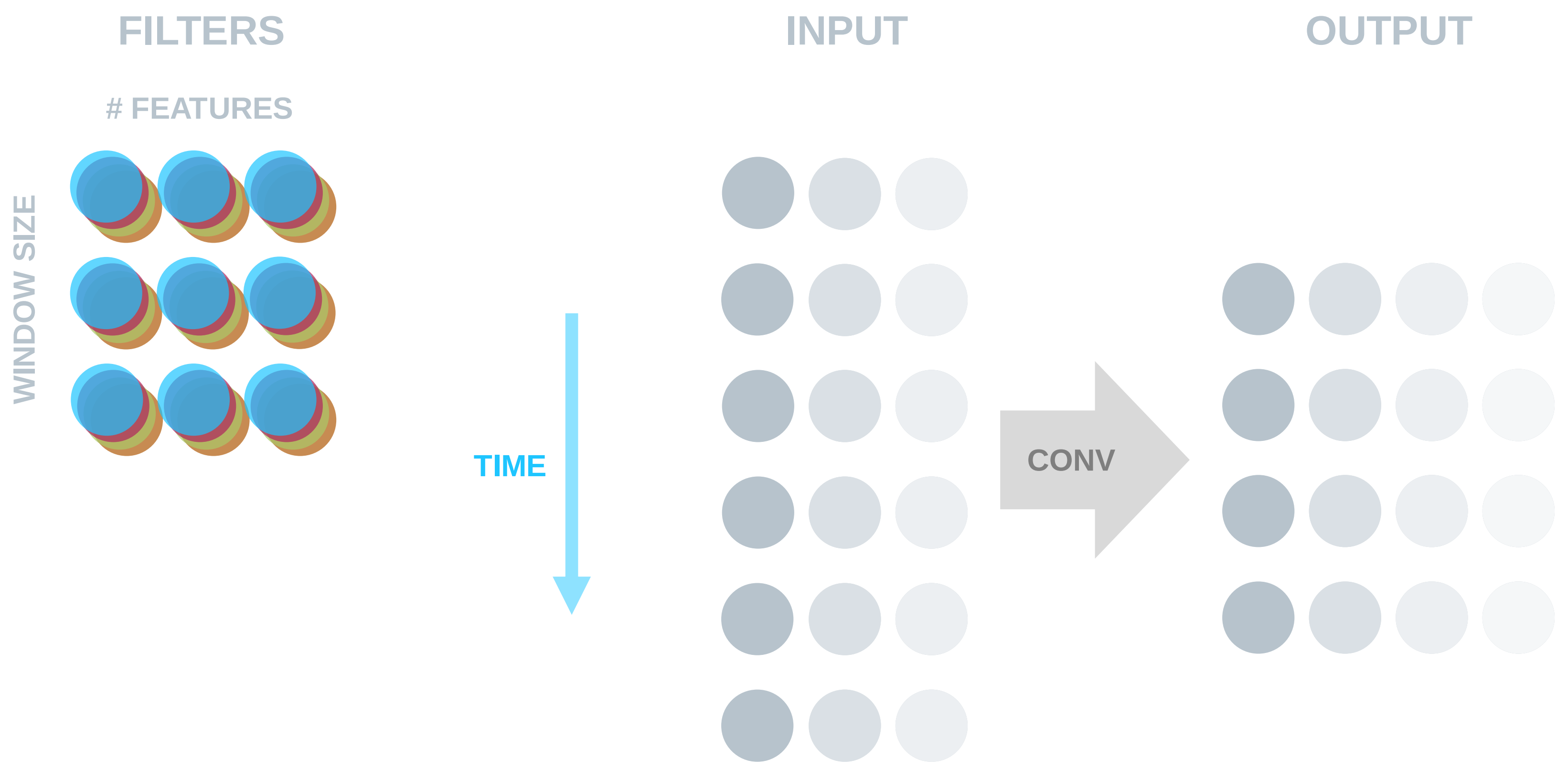
Neural Networks

1D Convolutional Feedforward Neural Network

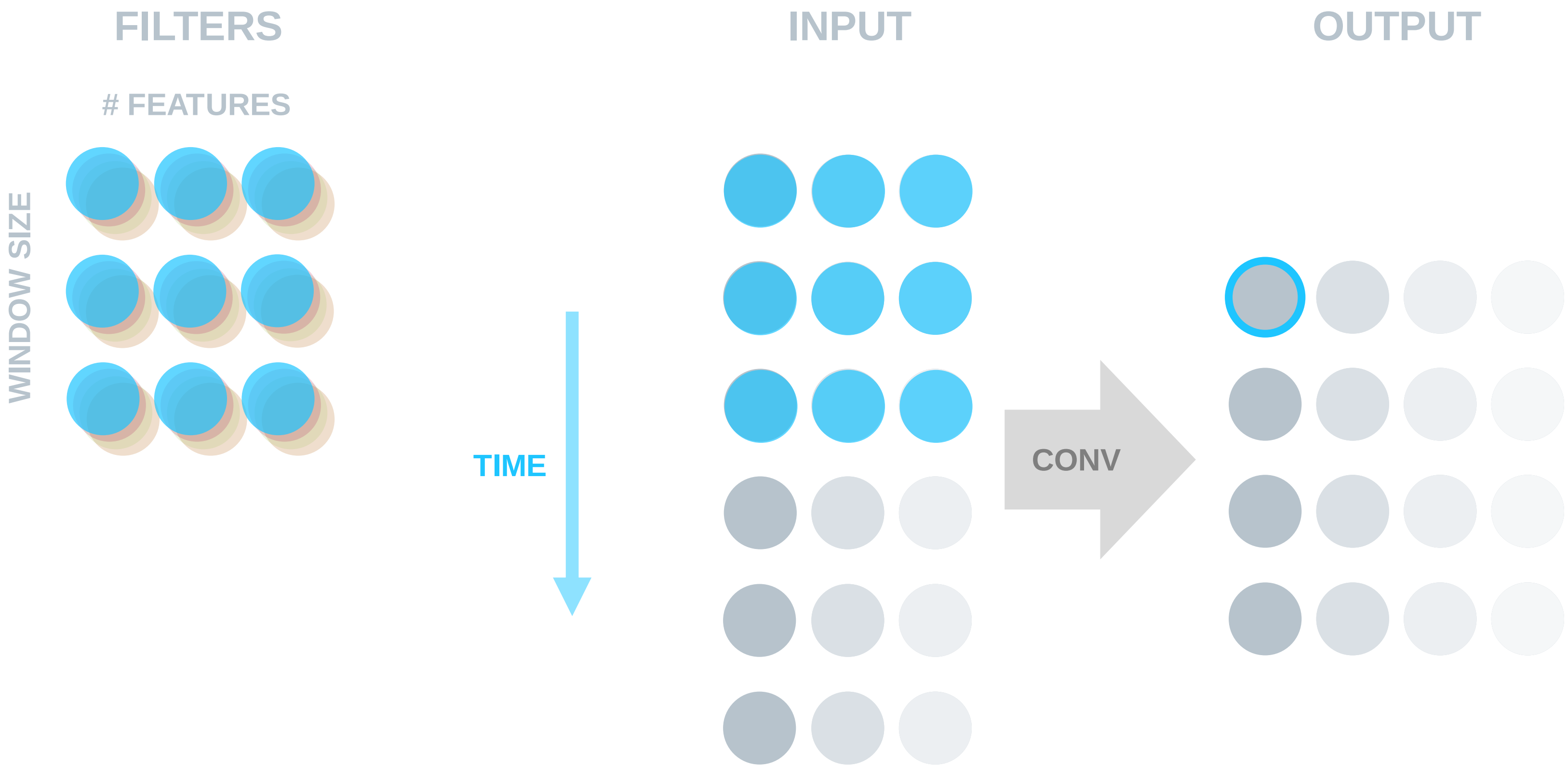


2016/06/01

Neural Networks
1D Convolution

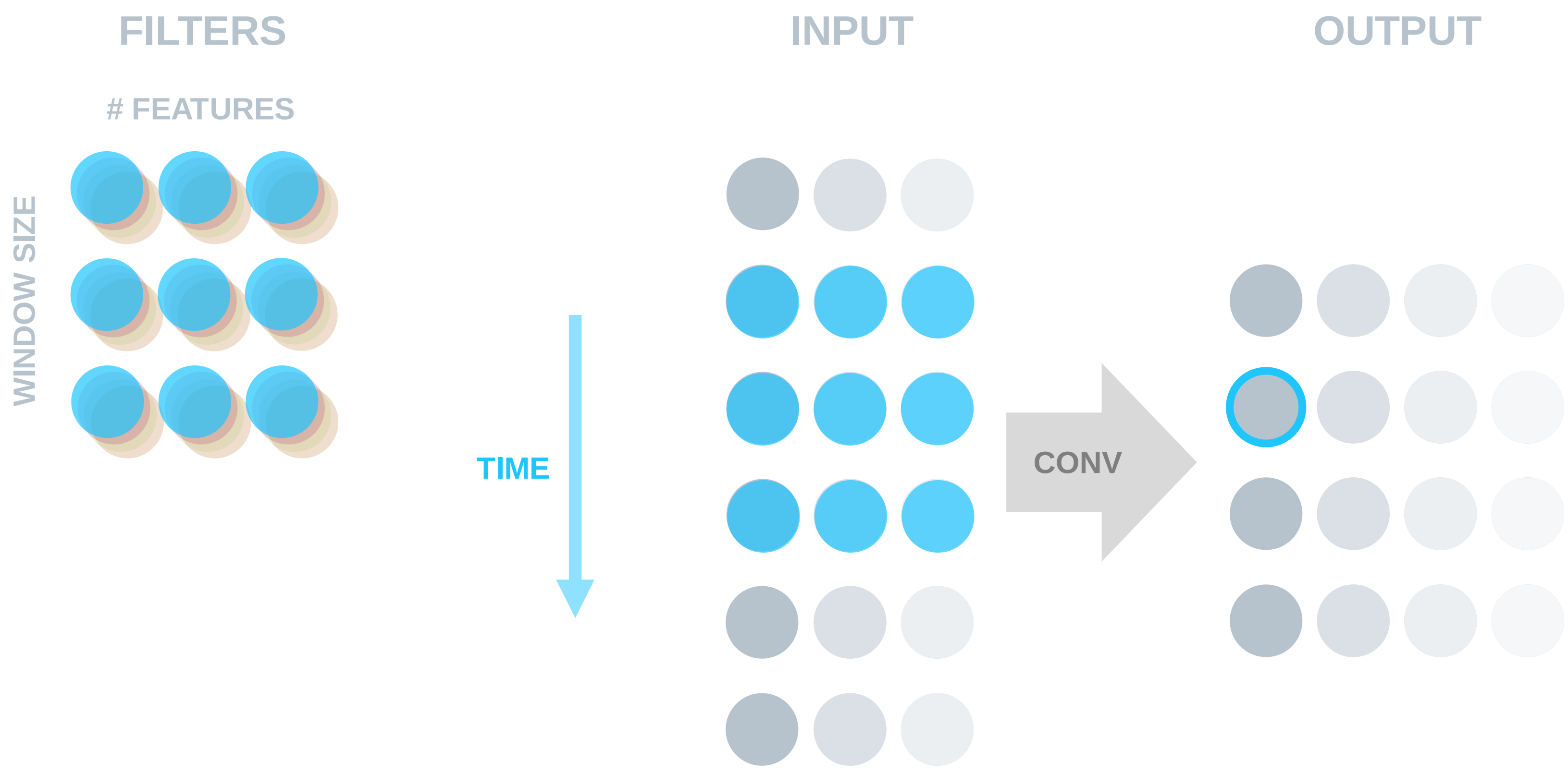


Neural Networks
1D Convolution

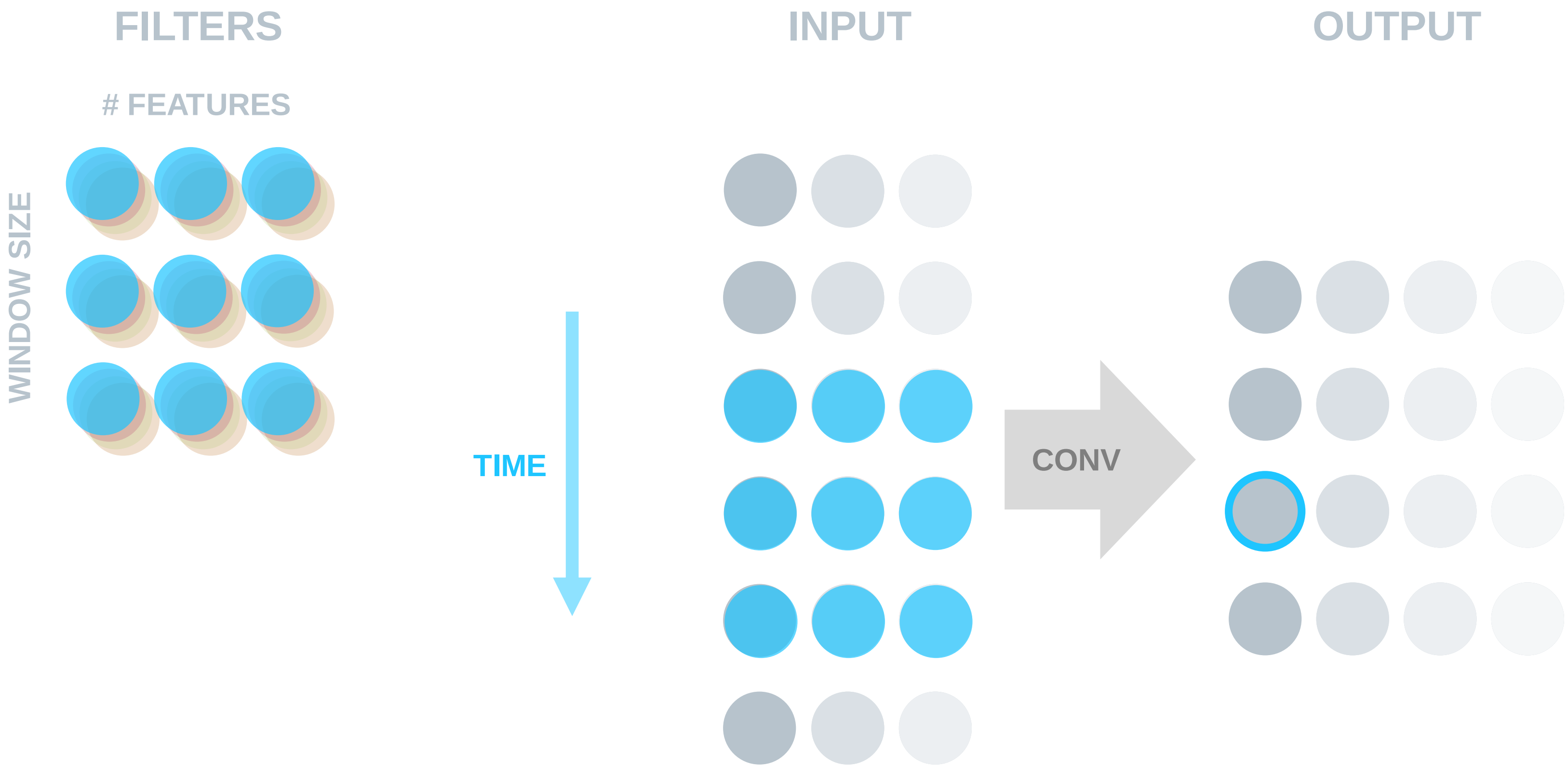


2016/06/01

Neural Networks
1D Convolution

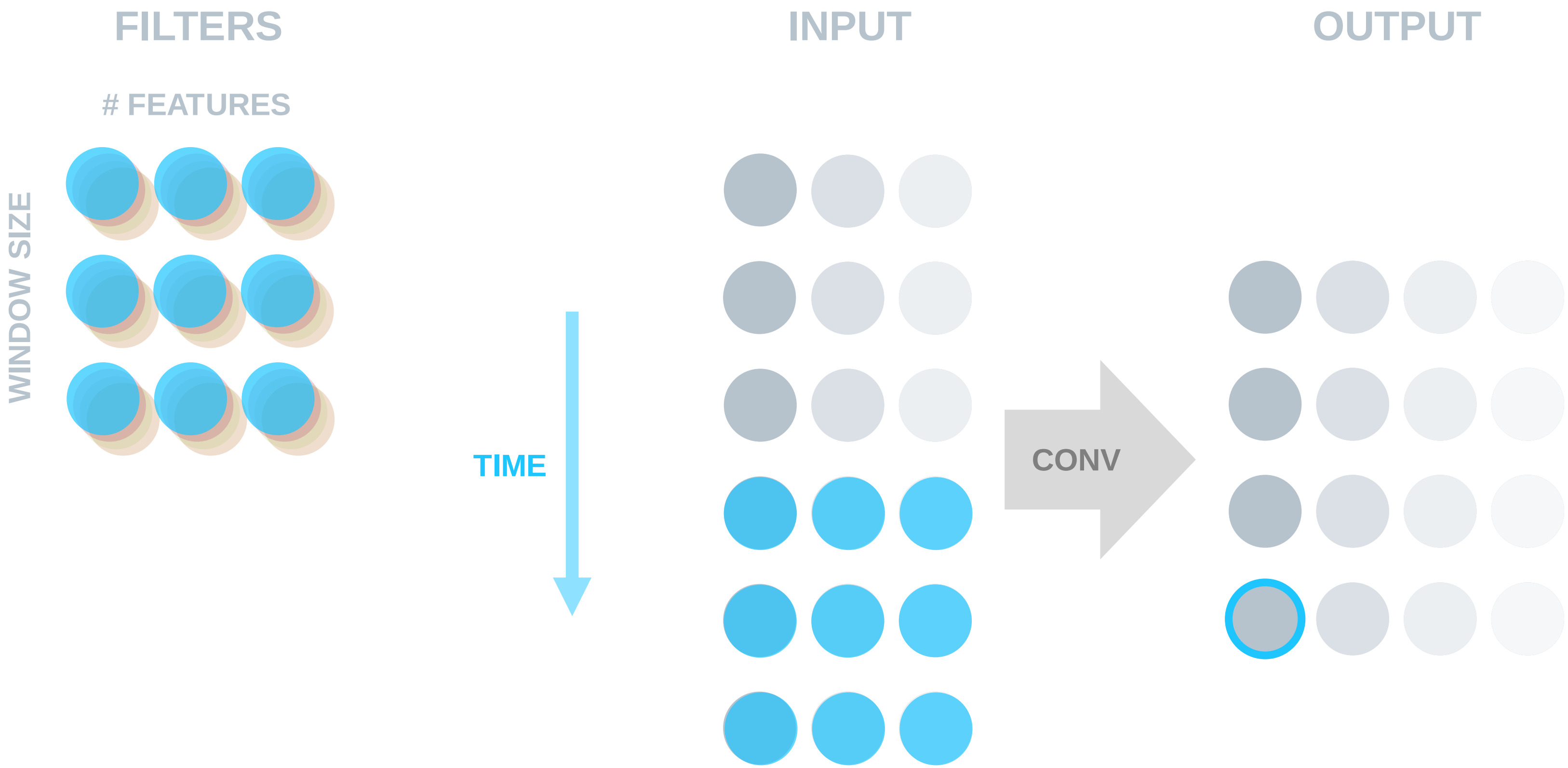


Neural Networks
1D Convolution

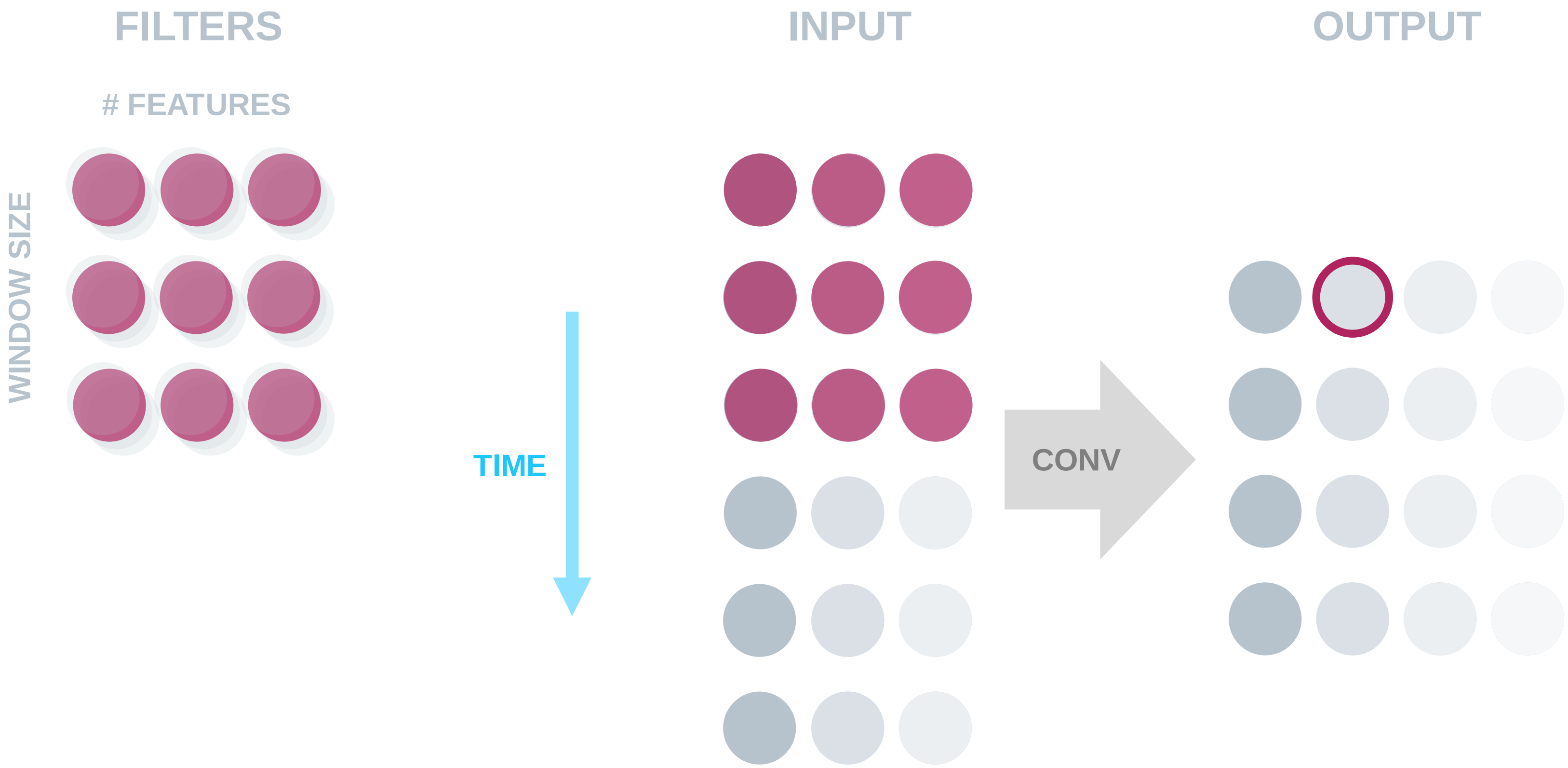


2016/06/01

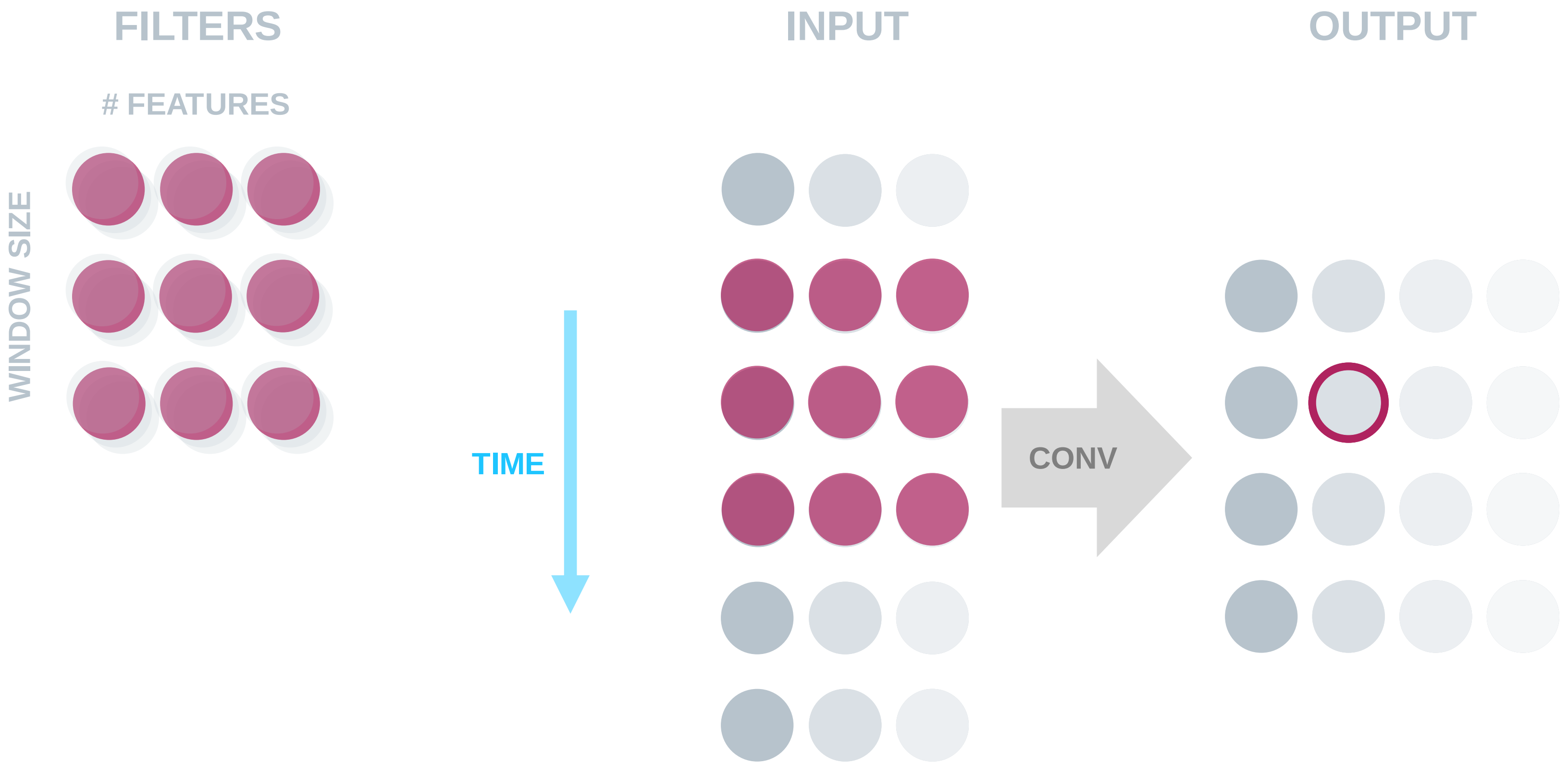
Neural Networks
1D Convolution



Neural Networks
1D Convolution

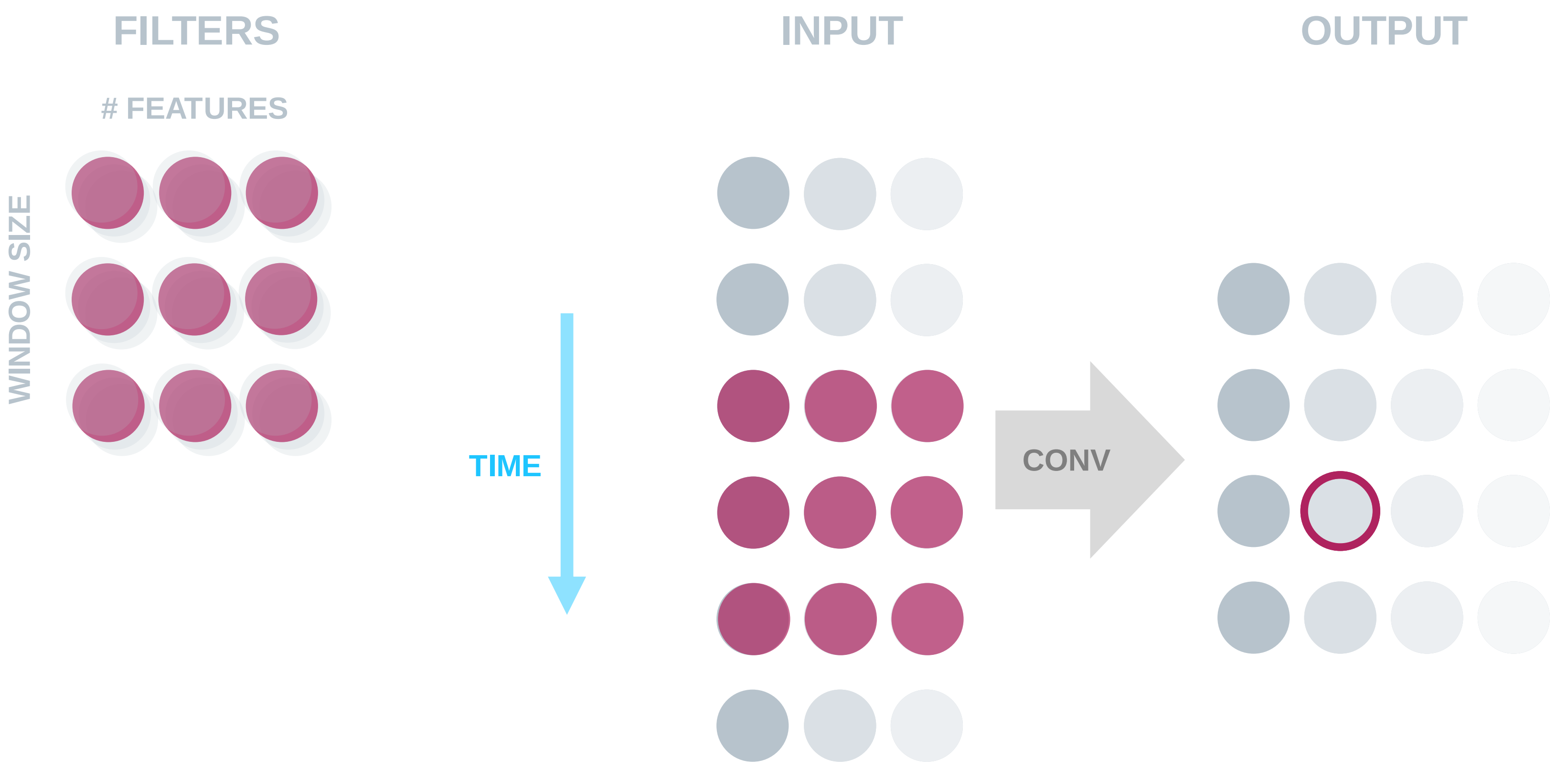


Neural Networks
1D Convolution



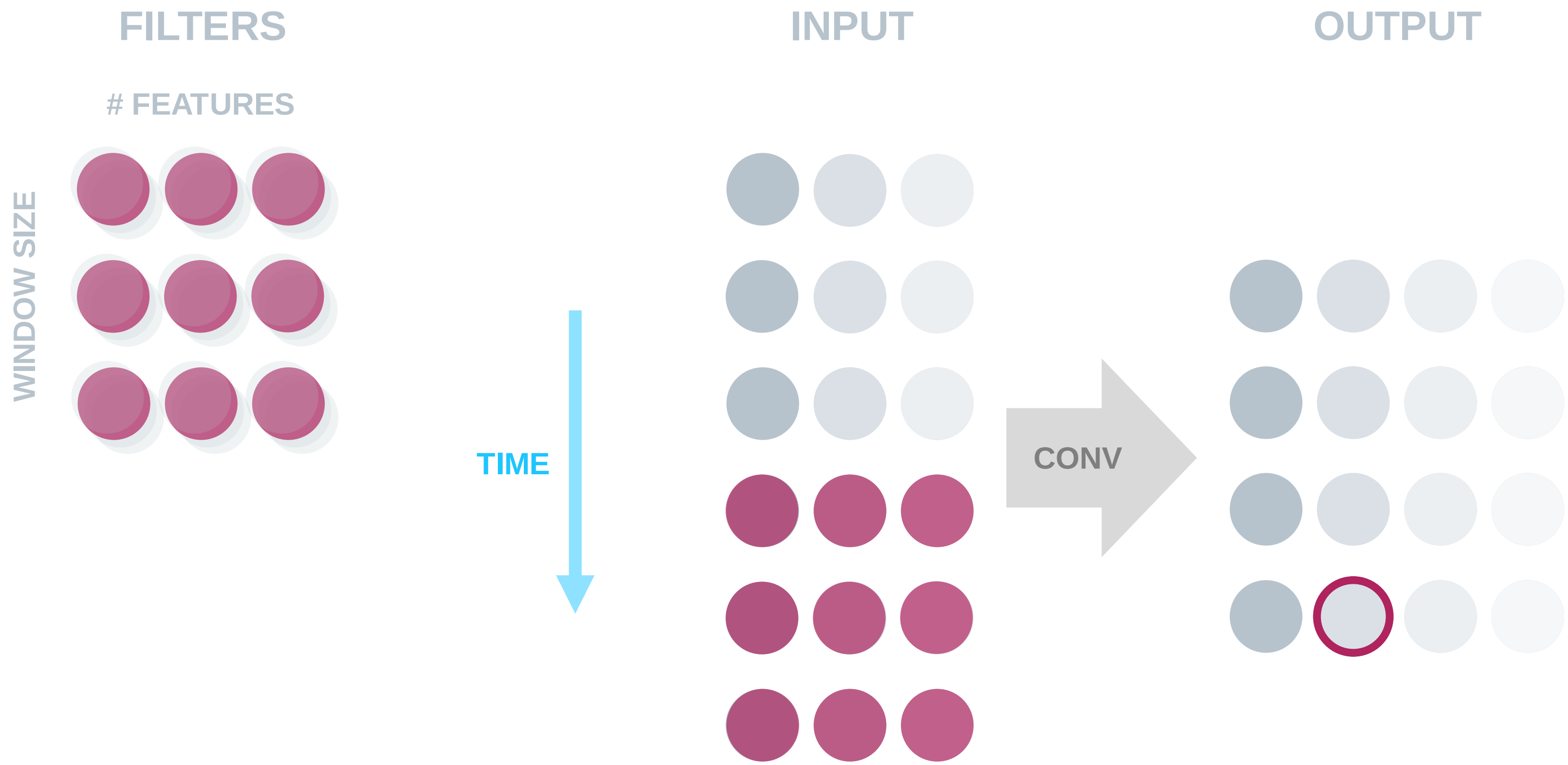
2016/06/01

Neural Networks
1D Convolution

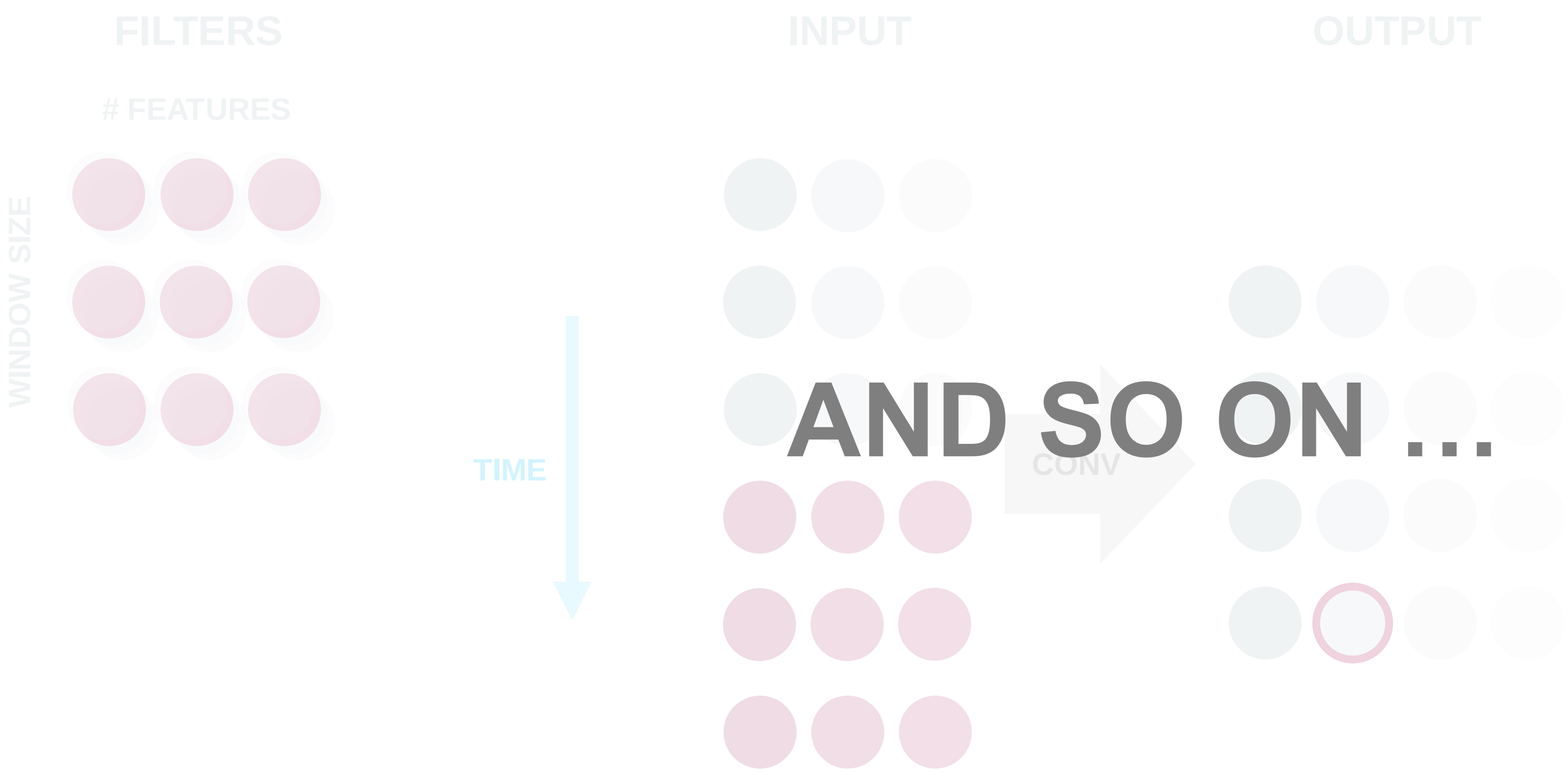


2016/06/01

Neural Networks
1D Convolution

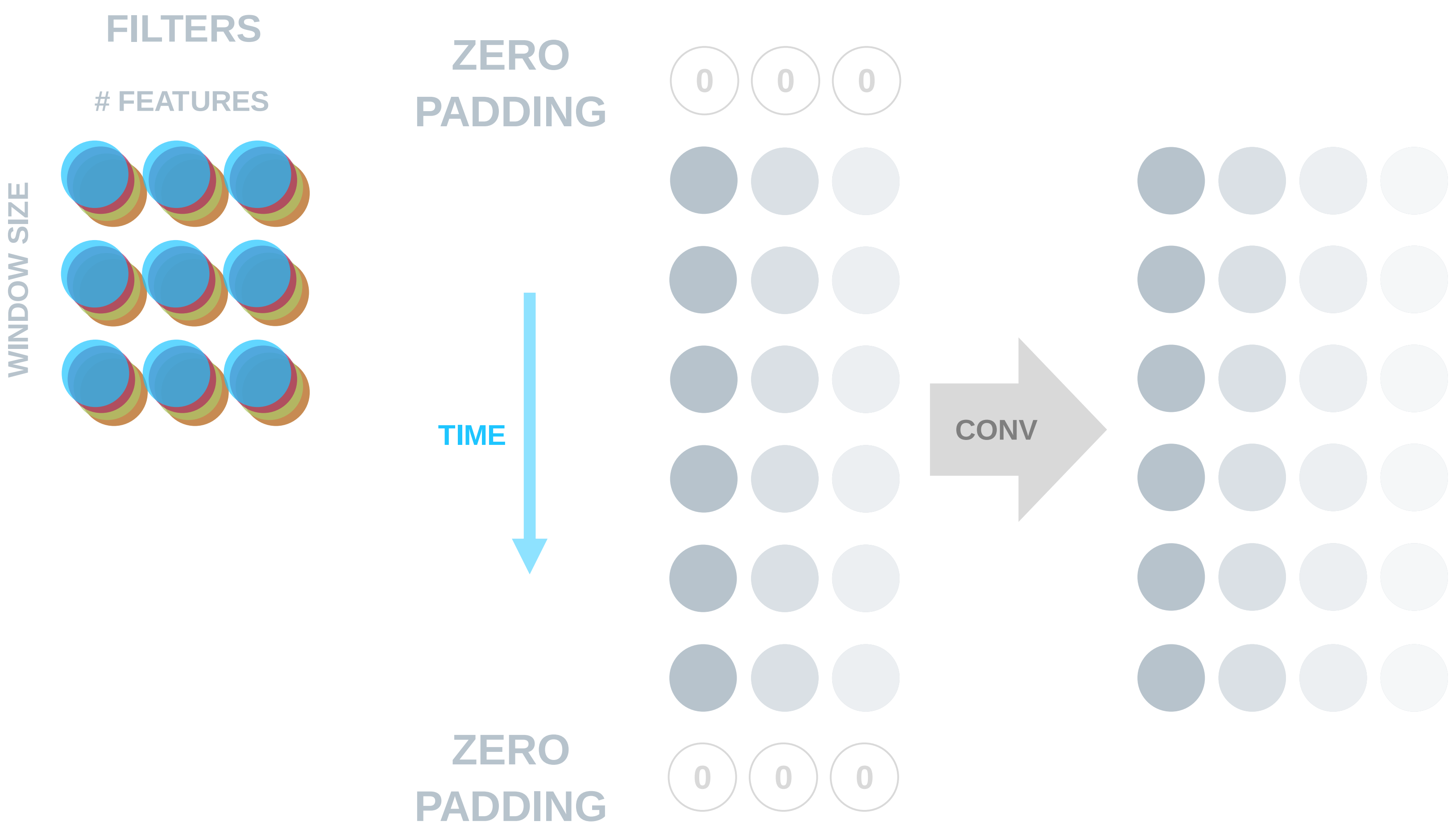


Neural Networks
1D Convolution



AND SO ON …

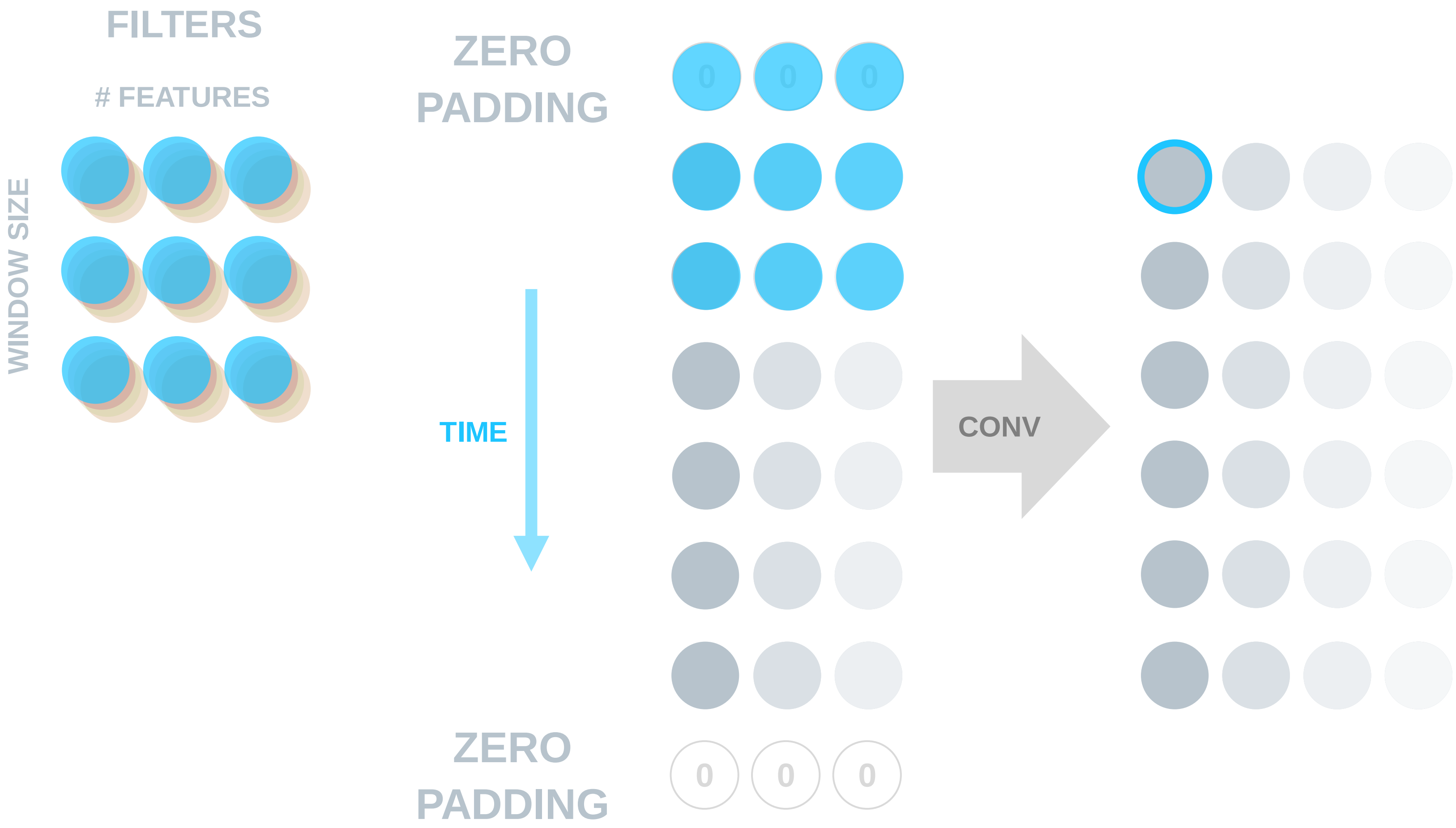
Neural Networks
1D Convolution (mode=HALF)



NOTE THAT THE INPUT AND OUTPUT SEQUENCES HAVE THE SAME LENGTH

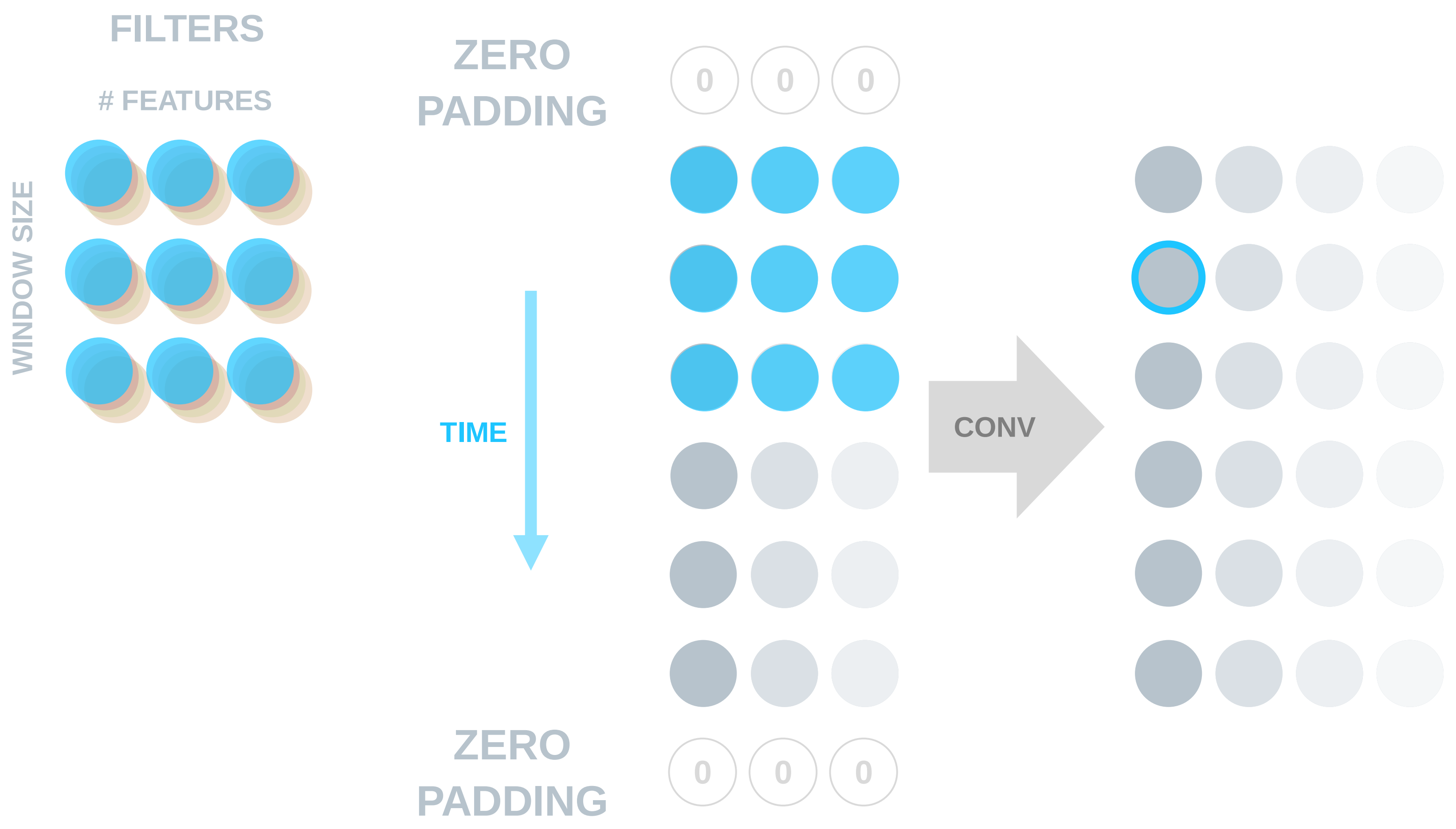
2016/06/01

Neural Networks
1D Convolution (mode=HALF)



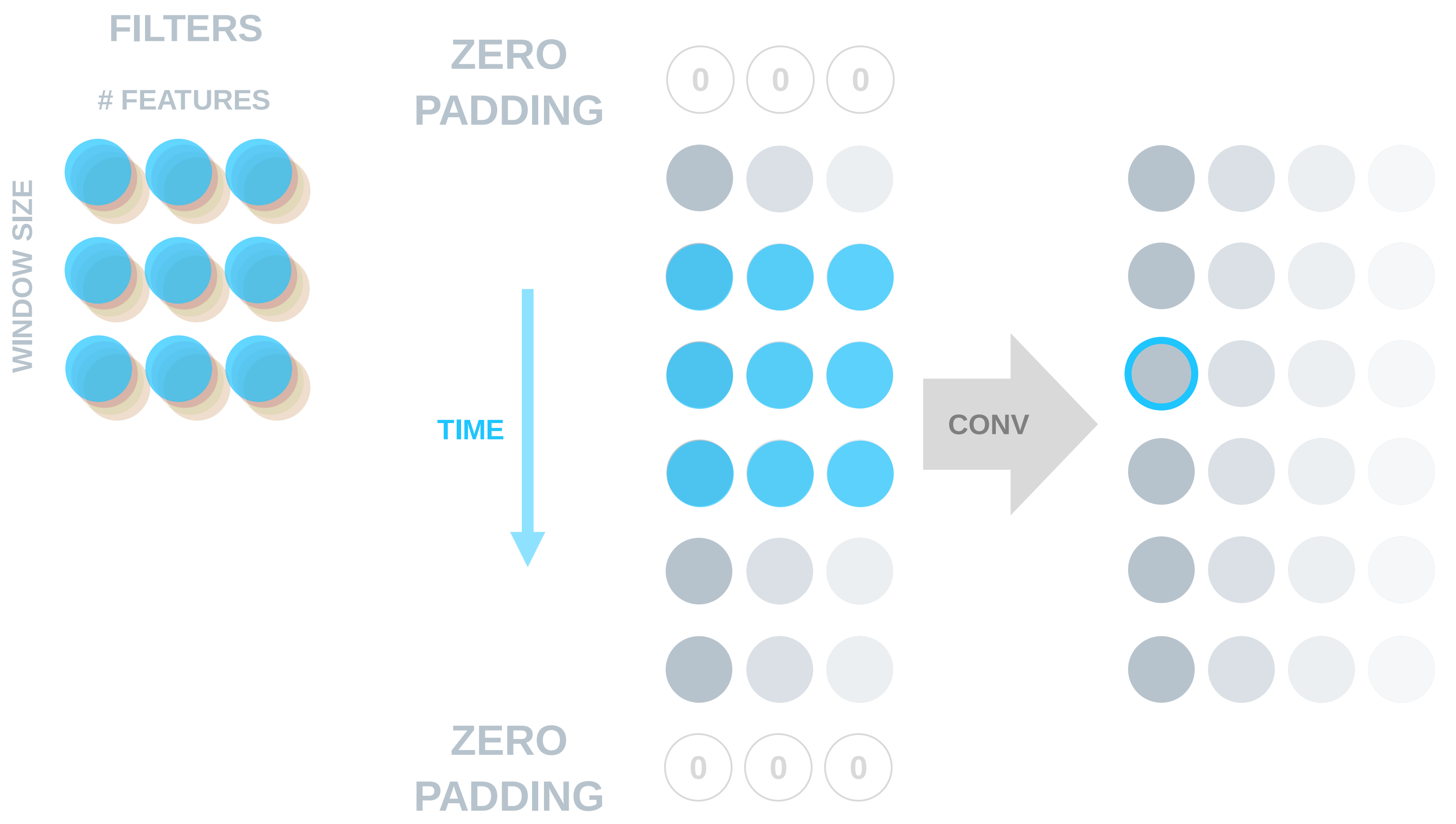
NOTE THAT THE INPUT AND OUTPUT SEQUENCES HAVE THE SAME LENGTH

Neural Networks
1D Convolution (mode=HALF)



NOTE THAT THE INPUT AND OUTPUT SEQUENCES HAVE THE SAME LENGTH

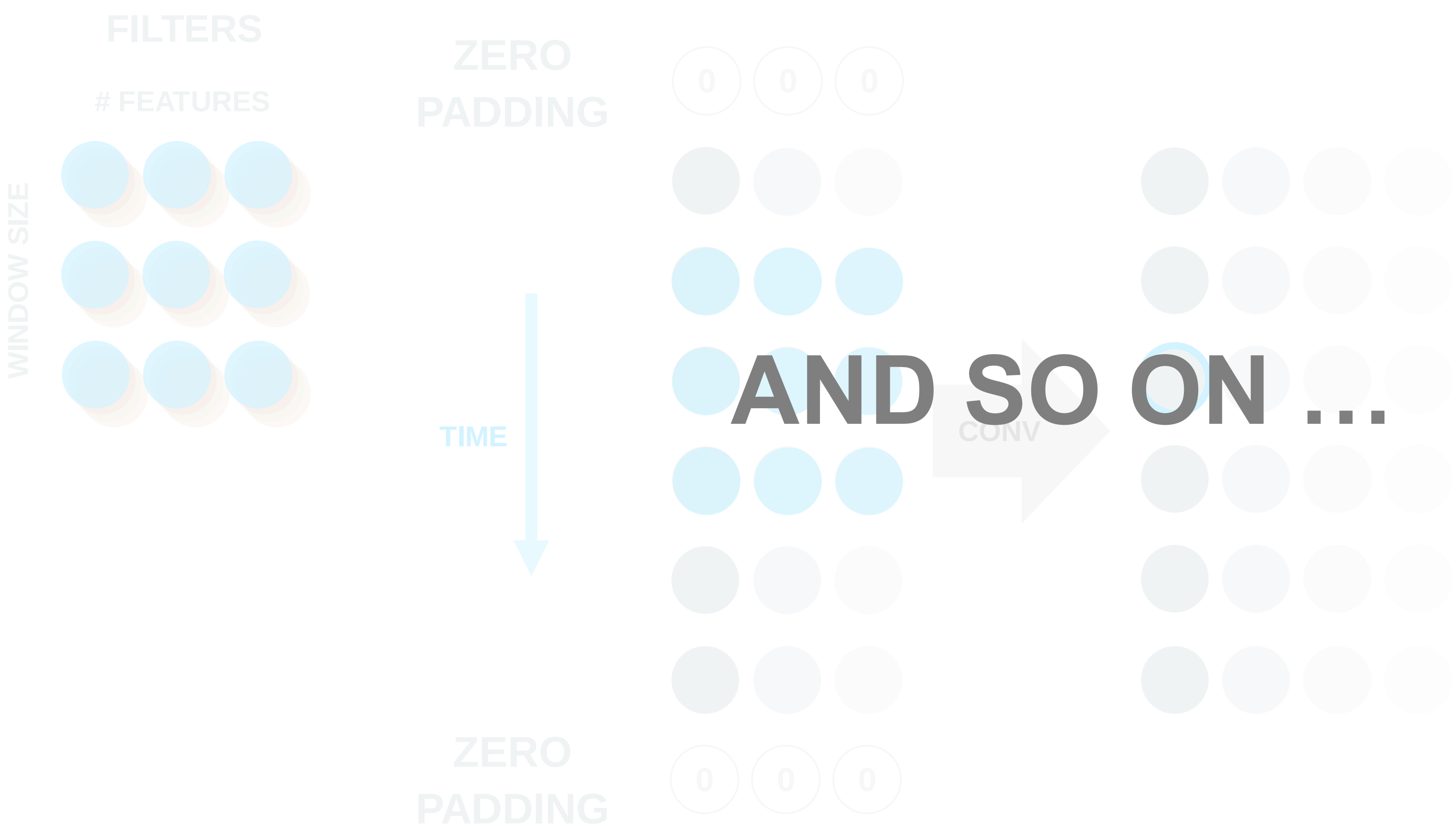
Neural Networks
1D Convolution (mode=HALF)



NOTE THAT THE INPUT AND OUTPUT SEQUENCES HAVE THE SAME LENGTH

2016/06/01

Neural Networks
1D Convolution (mode=HALF)

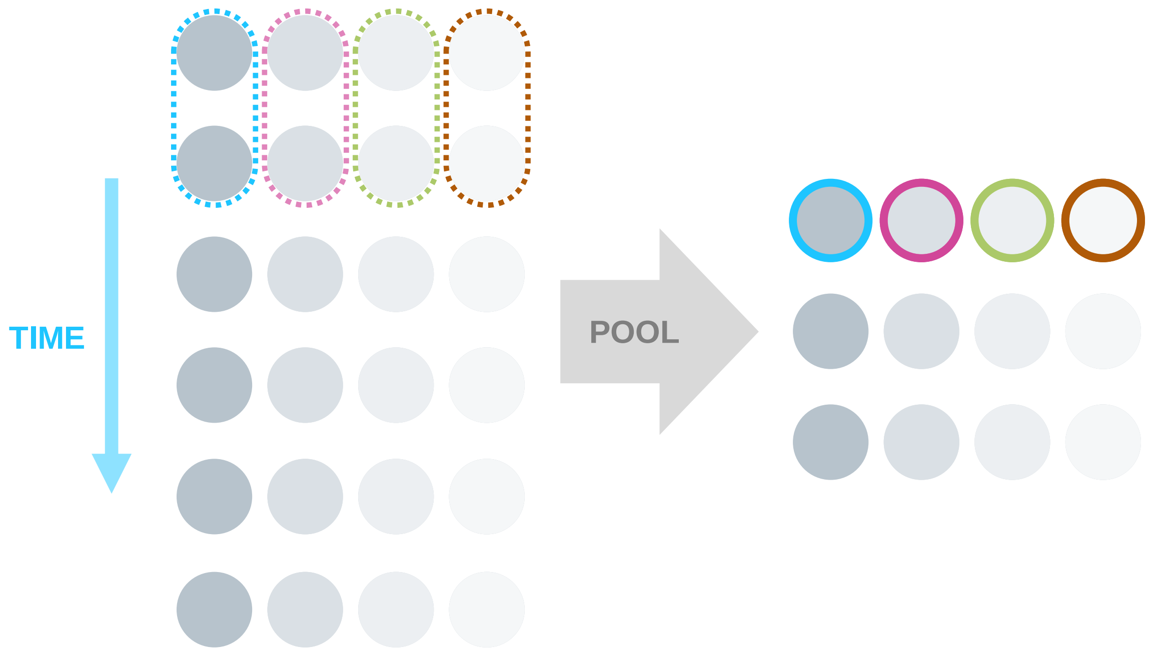


AND SO ON …

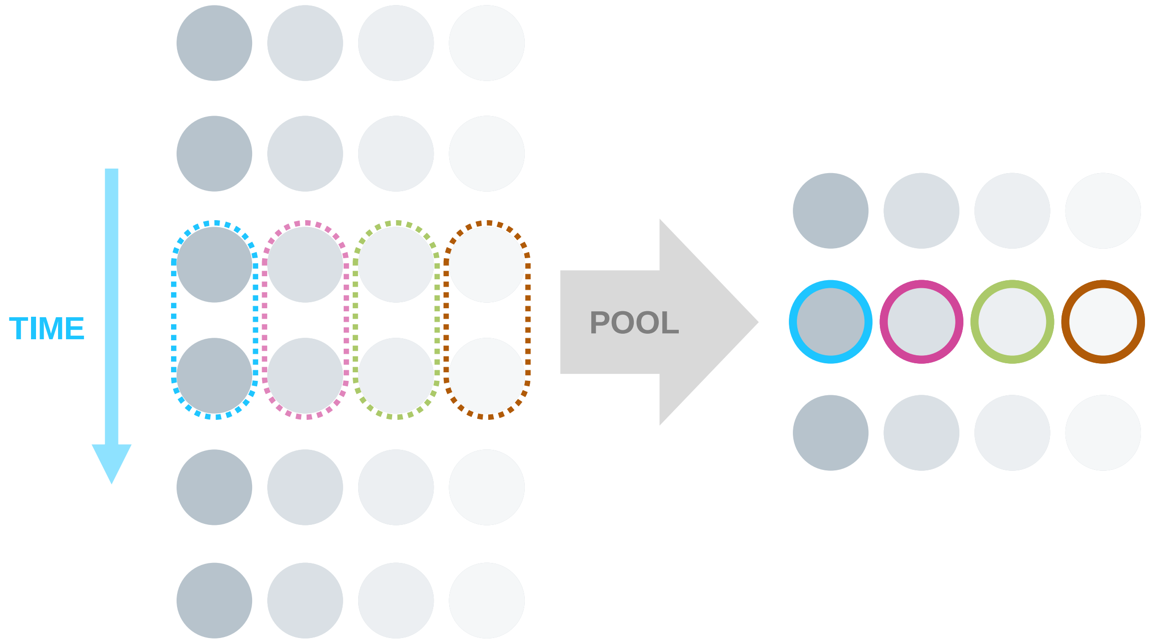
NOTE THAT THE INPUT AND OUTPUT SEQUENCES HAVE THE SAME LENGTH

Neural Networks

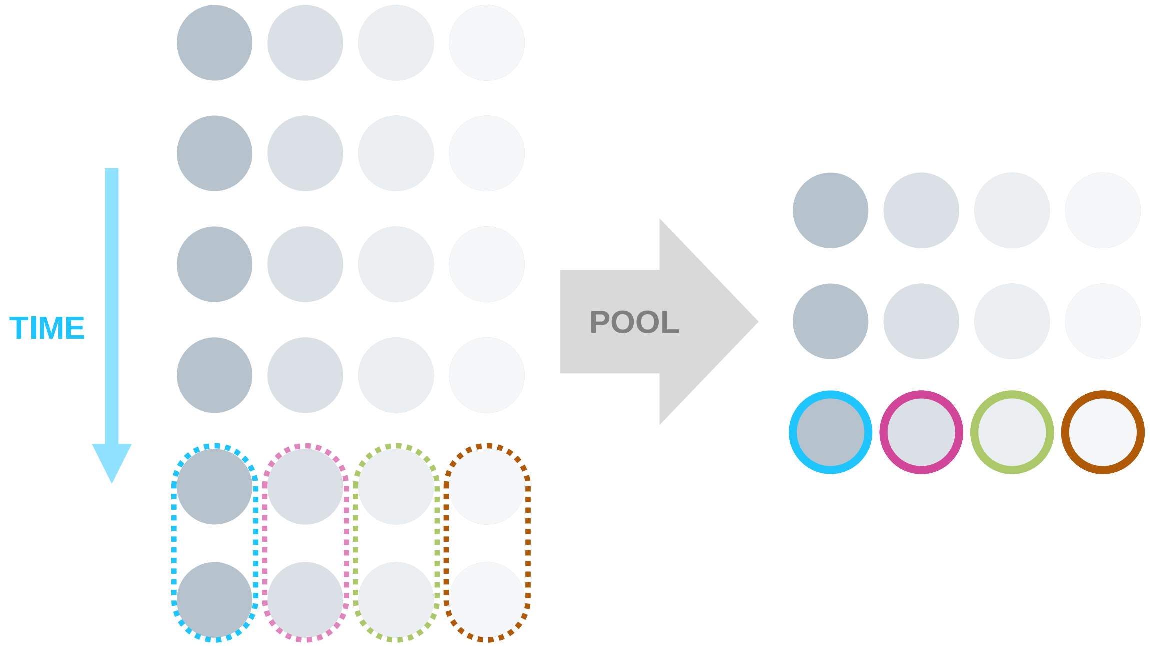
1D Pooling (max, sum, average, …)



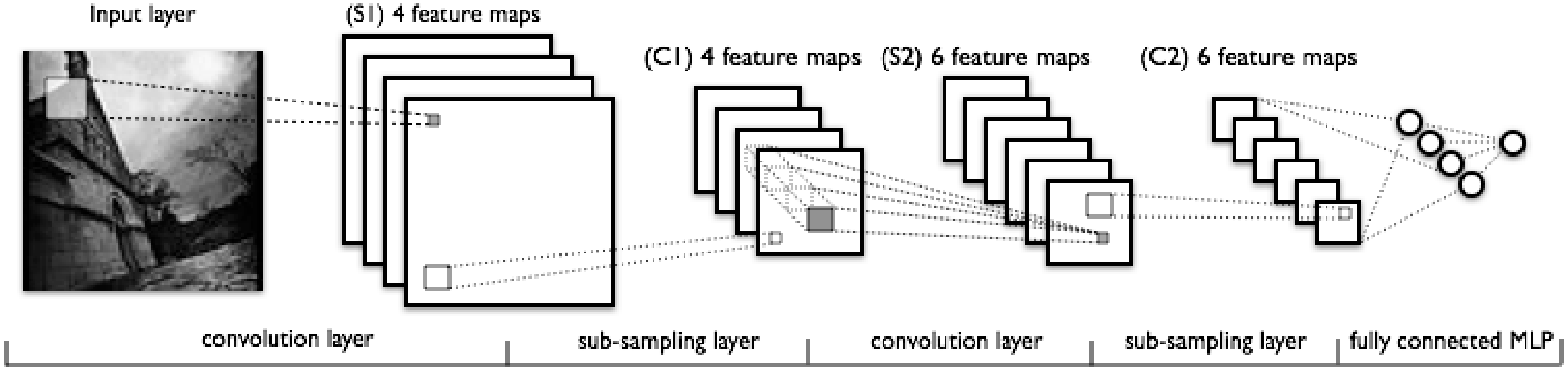
Neural Networks
1D Pooling (max, sum, average, …)



Neural Networks
1D Pooling (max, sum, average, …)



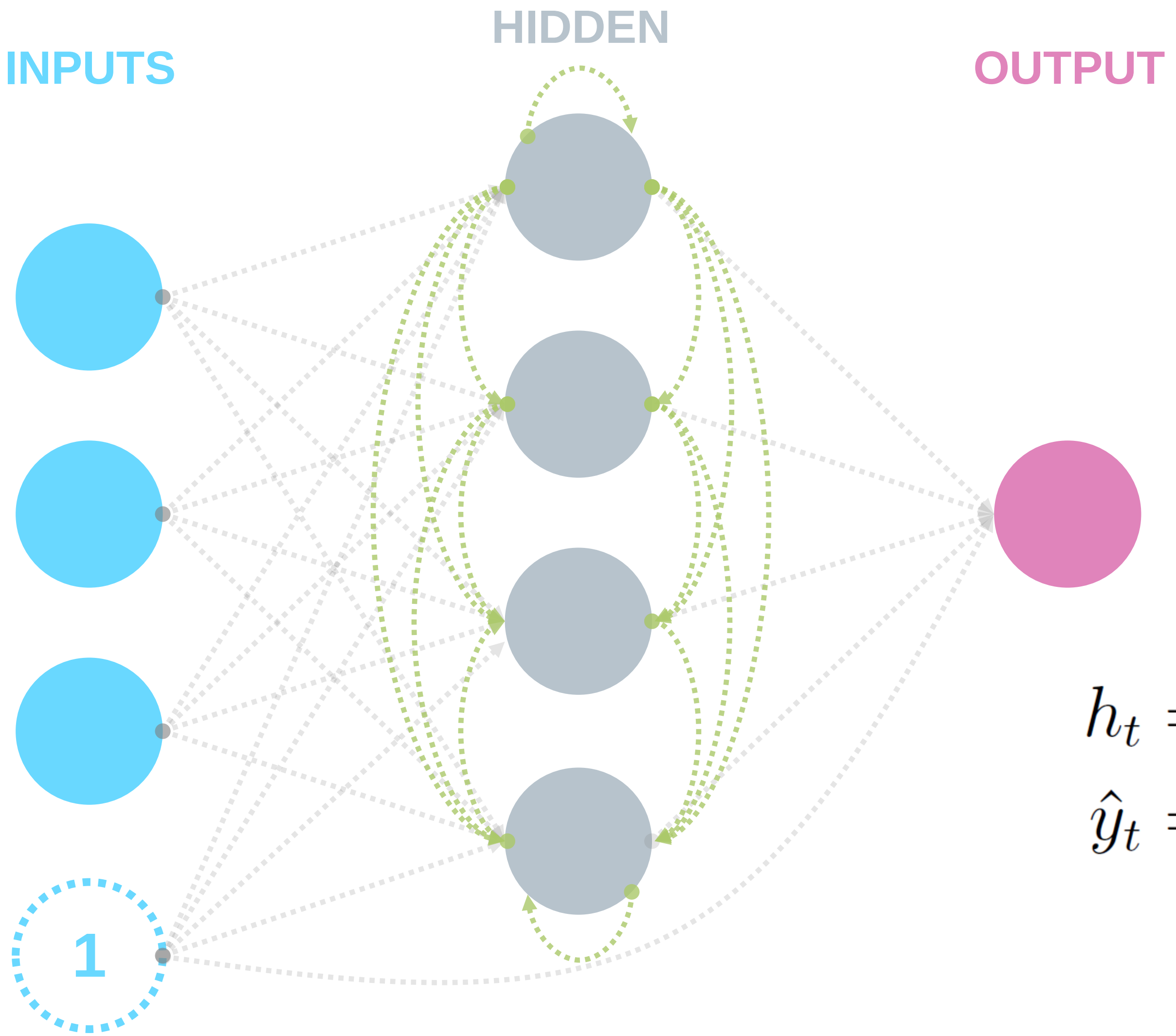
Neural Networks
2D Convolutional Feedforward Neural Network



SOURCE: http://deeplearning.net/tutorial/_images/mylenet.png

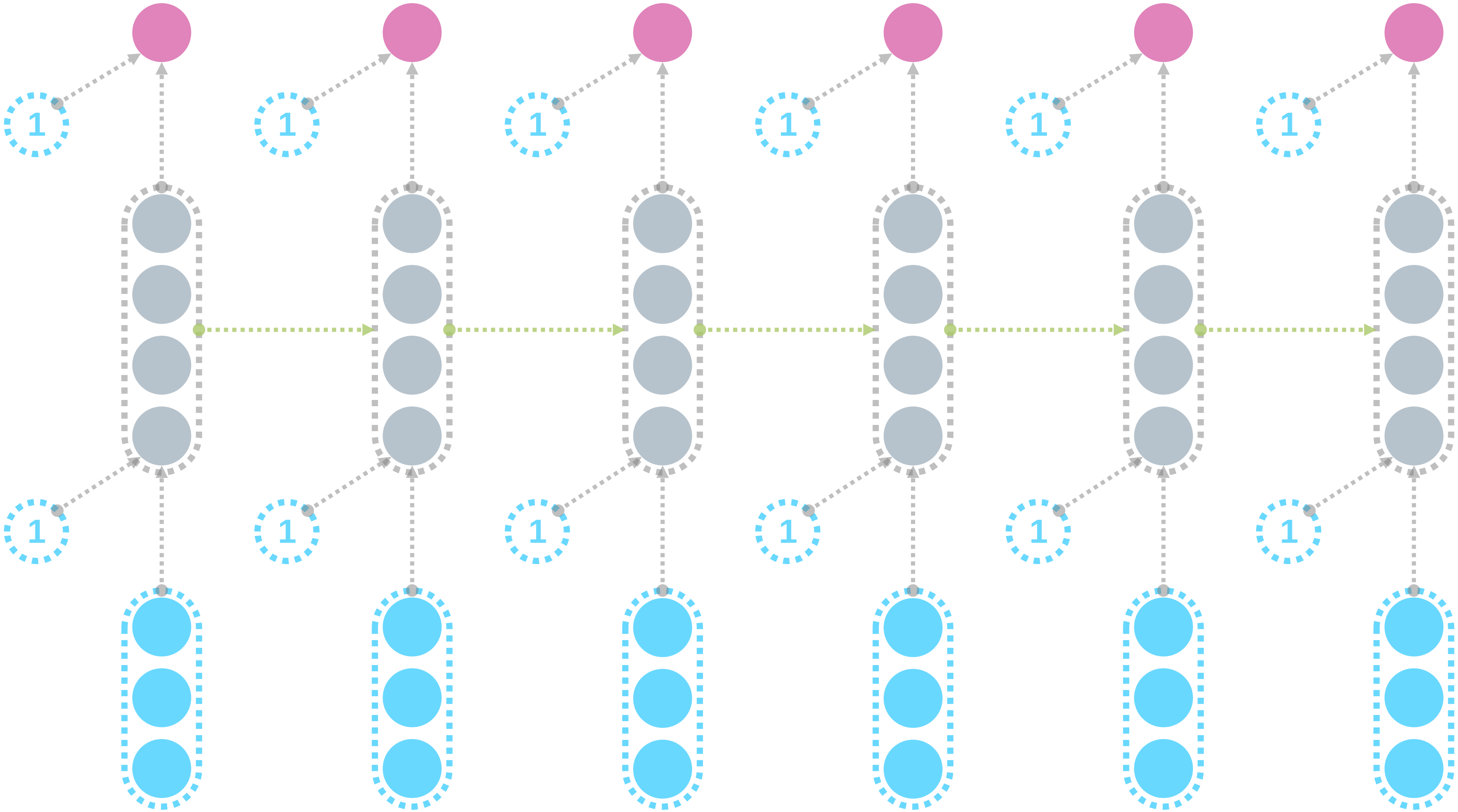
Neural Networks

Recurrent Neural Network (Elman architecture)



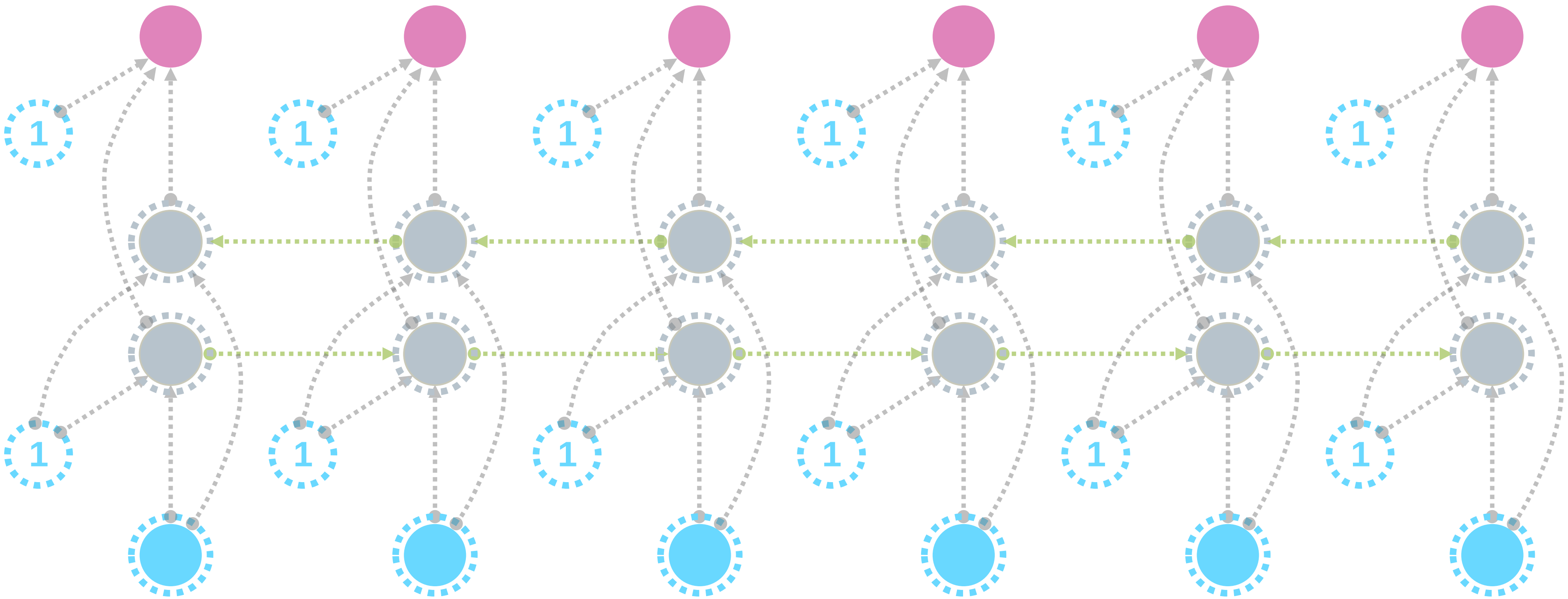
$$h_t = \tanh(Vx_t + Uh_{t-1} + b)$$
$$\hat{y}_t = \phi(Wh_t + d)$$

Neural Networks
Recurrent Neural Network (Elman architecture, unfolded)



Neural Networks

Bi-Directional Recurrent Neural Network (Elman architecture, unfolded)



NEURAL NETWORKS HAVE BEEN AROUND FOR DECADES!

SO WHAT’S NEW?

What's new?

* deprecated

OPTIMIZATION & LEARNING

OPTIMIZATION ALGORITHMS

- AdaGrad
- AdaDelta
- Adam
- RMSProp
- Hessian-Free Optimization
- …

REPARAMETERIZATION

- Batch Normalization
- Weight Normalization
- …

REGULARIZATION

- Dropout
- DropConnect
- …

MODEL ARCHITECTURES

BUILDING BLOCKS

- Spatial/Temporal Pooling
- Attention Mechanism
- Gated Recurrent Units
- Beam-search for sequence generation
- Variable-length sequence modeling
- …

ARCHITECTURES

- Inception (Google)
- VGG (Oxford University)
- Encoder-Decoder Framework
- End-to-end Models
- …

SOFTWARE

- Theano
  - Blocks + Fuel
  - Keras
  - Lasagne
  - PyLearn2*
- TensorFlow
- Torch7
- Caffe…

GENERAL

- GPUs
- Data

What's new?

OPTIMIZATION & LEARNING

OPTIMIZATION ALGORITHMS
- AdaGrad
- AdaDelta
- Adam
- RMSProp
- Hessian-Free Optimization
- …

REPARAMETERIZATION
- Batch Normalization
- Weight Normalization
- …

REGULARIZATION
- Dropout
- DropConnect
- …

MODEL ARCHITECTURES

BUILDING BLOCKS
- Spatial/Temporal Pooling
- Attention Mechanism
- Gated Recurrent Units
- Beam-search for sequence generation
- Variable-length sequence modeling
- …

ARCHITECTURES
- Inception (Google)
- VGG (Oxford University)
- Encoder-Decoder Framework
- End-to-end Models
- …

SOFTWARE
- Theano
  - Blocks + Fuel
  - Keras
  - Lasagne
  - PyLearn2*
- TensorFlow
- Torch7
- Caffe…

GENERAL
- GPUs
- Data

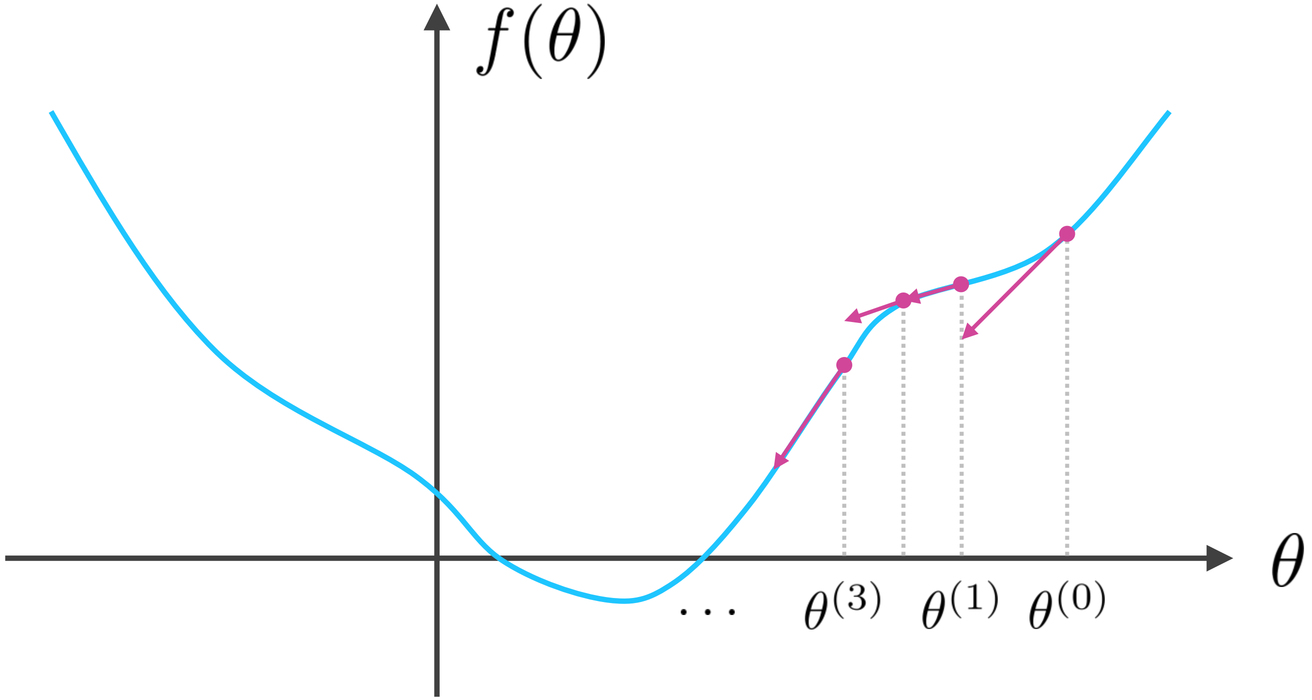
What’s new?
Optimization Algorithms

- Neural networks are composed of differentiable building blocks
- Training a neural network means minimization of some non-convex differentiable cost function using iterative gradient-based optimization methods

What's new?

Optimization Algorithms

- Neural networks are composed of differentiable building blocks
- Training a neural network means minimization of some non-convex differentiable cost function using iterative gradient-based optimization methods



What's new?

Optimization Algorithms

- Neural networks are composed of differentiable building blocks
- Training a neural network means minimization of some non-convex differentiable cost function using iterative gradient-based optimization methods
- Gradients are computed using backpropagation
- The simplest optimization algorithm is "gradient descent"

$$\theta^{(i+1)} \leftarrow \theta^{(i)} - \eta \nabla_{\theta^{(i)}} f\big(\theta^{(i)}\big)$$

What's new?
Optimization Algorithms

- Neural networks are composed of differentiable building blocks
- Training a neural network means minimization of some non-convex differentiable cost function using iterative gradient-based optimization methods
- Gradients are computed using backpropagation
- The simplest optimization algorithm is "gradient descent"
- … but it has limitations

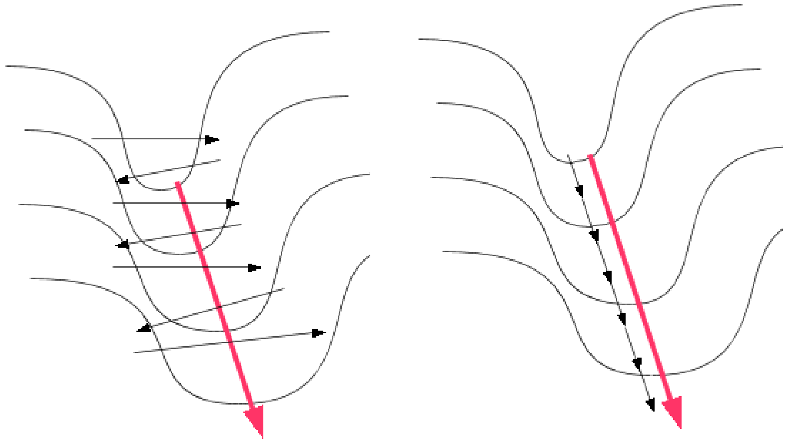


IMAGE SOURCE:
Martens, J. (2010). Deep Learning via Hessian-Free Optimization. In *Proceedings of the 27th International Conference on Machine Learning* (pp. 735-742).

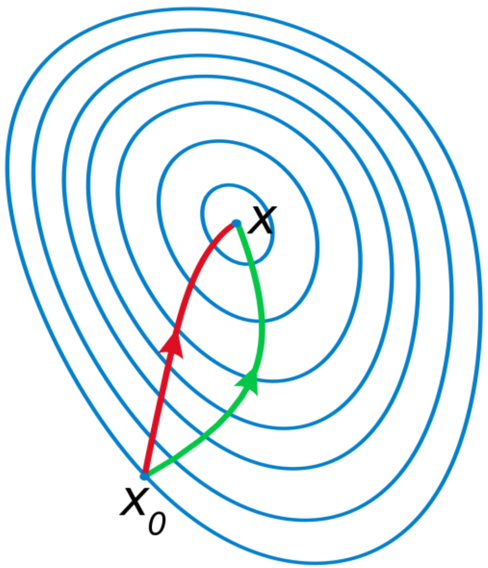
What’s new?

Optimization Algorithms

- Neural networks are composed of differentiable building blocks
- Training a neural network means minimization of some non-convex differentiable cost function using iterative gradient-based optimization methods
- Gradients are computed using backpropagation
- The simplest optimization algorithm is “gradient descent”
- … but it has limitations
- Information about the local curvature of the cost function helps to adjust the direction and magnitude of the gradient for better progress (along the lines of Newton’s method)
- Exact local curvature is infeasible to compute
- Recent optimization algorithms like AdaGrad, RMSProp, AdaDelta etc. try to approximate local curvature information efficiently

What's new?
Reparameterization

- First-order gradient-based optimization methods are not invariant to reparameterization of the optimization objective



What’s new?
Reparameterization

- First-order gradient-based optimization methods are not invariant to reparameterization of the optimization objective
- Instead of using more sophisticated optimization algorithms that are better at dealing with ill-conditioned optimization problems, reparameterize the objective function so that simpler optimization algorithms work better

What’s new?
Reparameterization

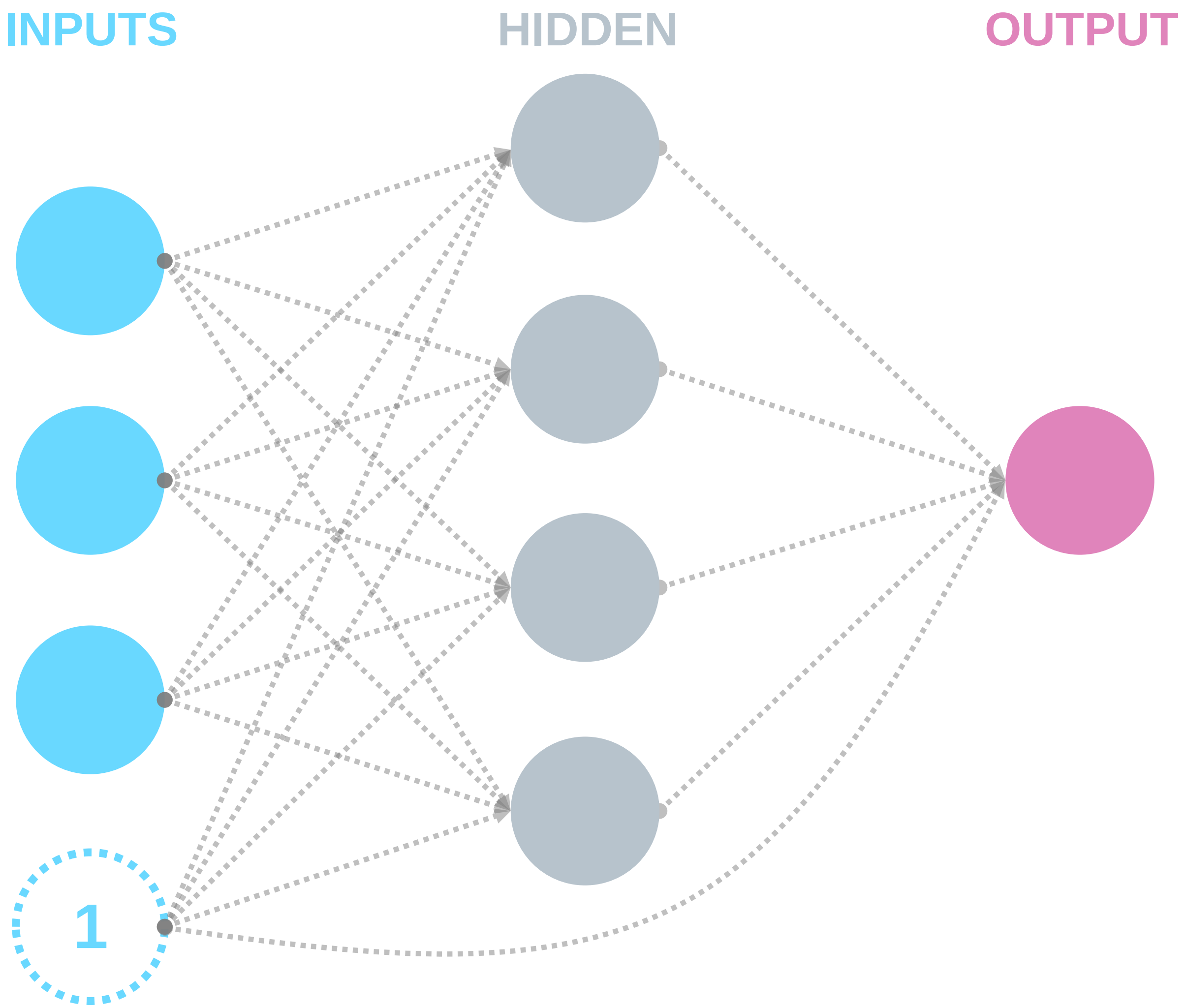
- First-order gradient-based optimization methods are not invariant to reparameterization of the optimization objective
- Instead of using more sophisticated optimization algorithms that are better at dealing with ill-conditioned optimization problems, reparameterize the objective function so that simpler optimization algorithms work better
- We typically standardize (approximately decorrelate) real-valued (Gaussian-like) inputs which makes the optimization problem easier

Reparameterization

- First-order gradient-based optimization methods are not invariant to reparameterization of the optimization objective
- Instead of using more sophisticated optimization algorithms that are better at dealing with ill-conditioned optimization problems, reparameterize the objective function so that simpler optimization algorithms work better
- We typically standardize (approximately decorrelate) real-valued (Gaussian-like) inputs which makes the optimization problem easier
- Why not do this in each (hidden) layer as well?
  $\Rightarrow$ Batch Normalization

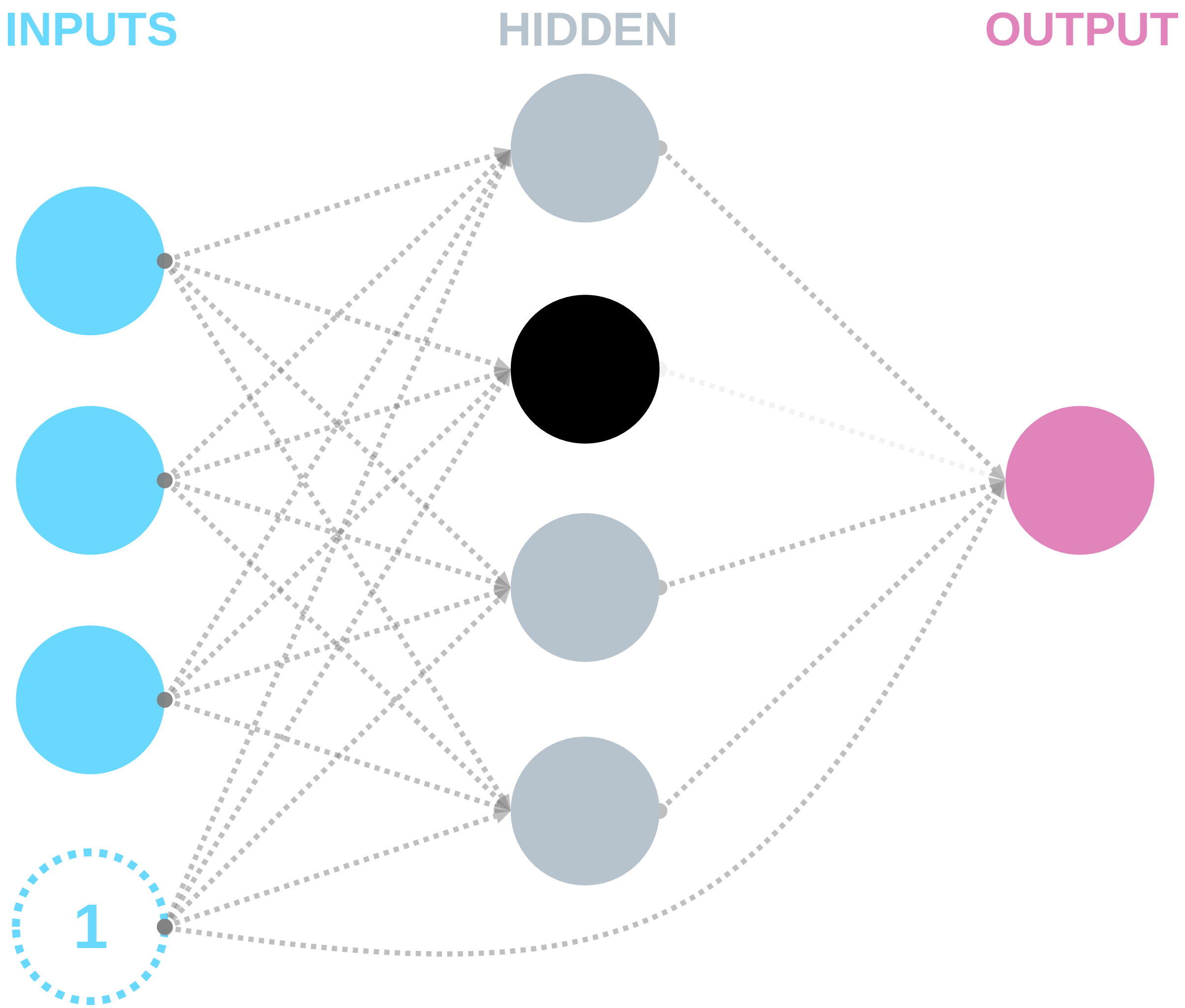
What's new?
Regularization

- Randomly set neurons to zero
- Results in an ensemble with an exponential number of members whose parameters are shared
- Primarily used in fully connected layers because of the large number of parameters
- Rarely used in convolutional layers
- Rarely used in recurrent neural networks (if at all between the hidden state and output)



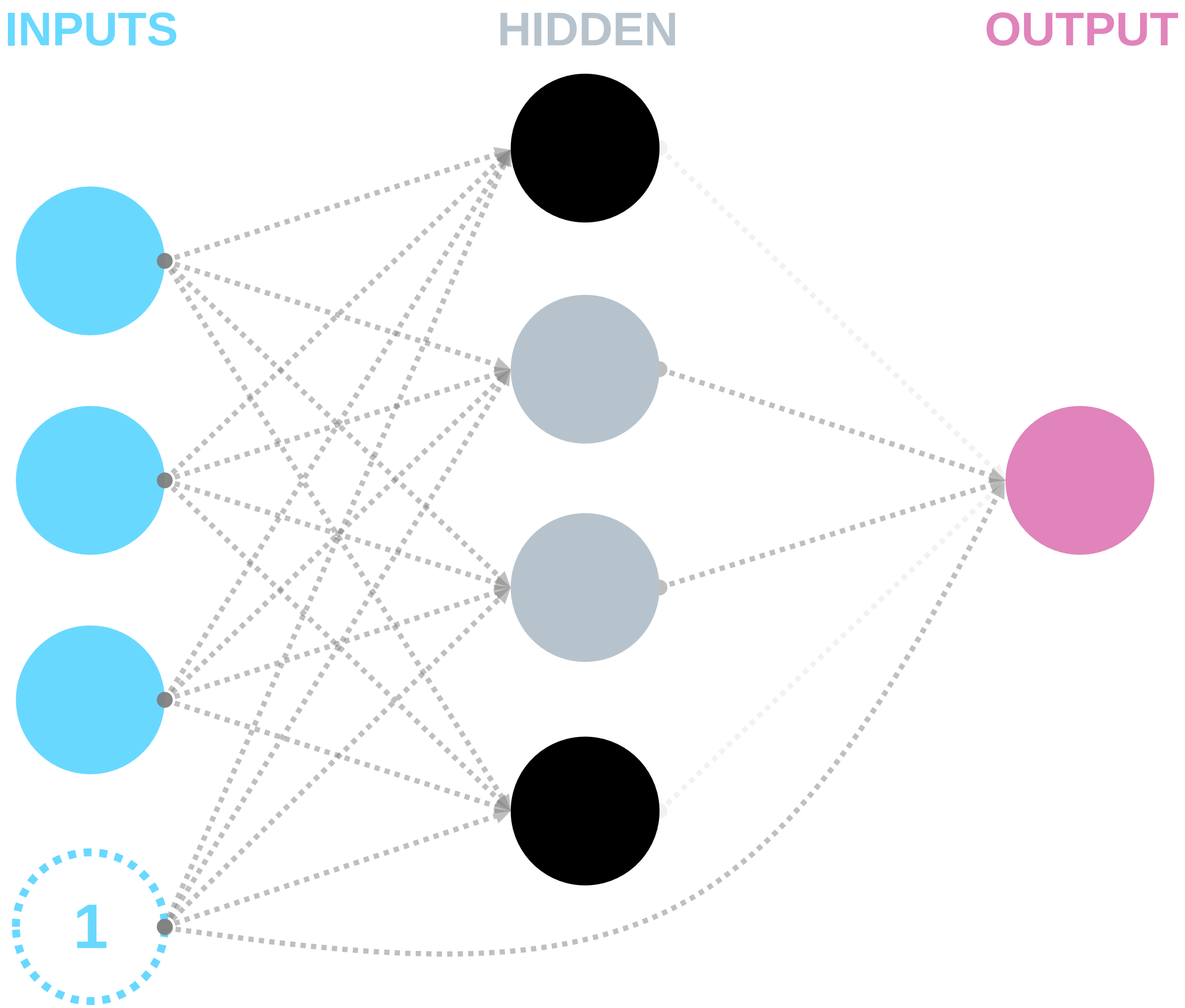
What's new?
Regularization

- Randomly set neurons to zero
- Results in an ensemble with an exponential number of members whose parameters are shared
- Primarily used in fully connected layers because of the large number of parameters
- Rarely used in convolutional layers
- Rarely used in recurrent neural networks (if at all between the hidden state and output)



What’s new?
Regularization

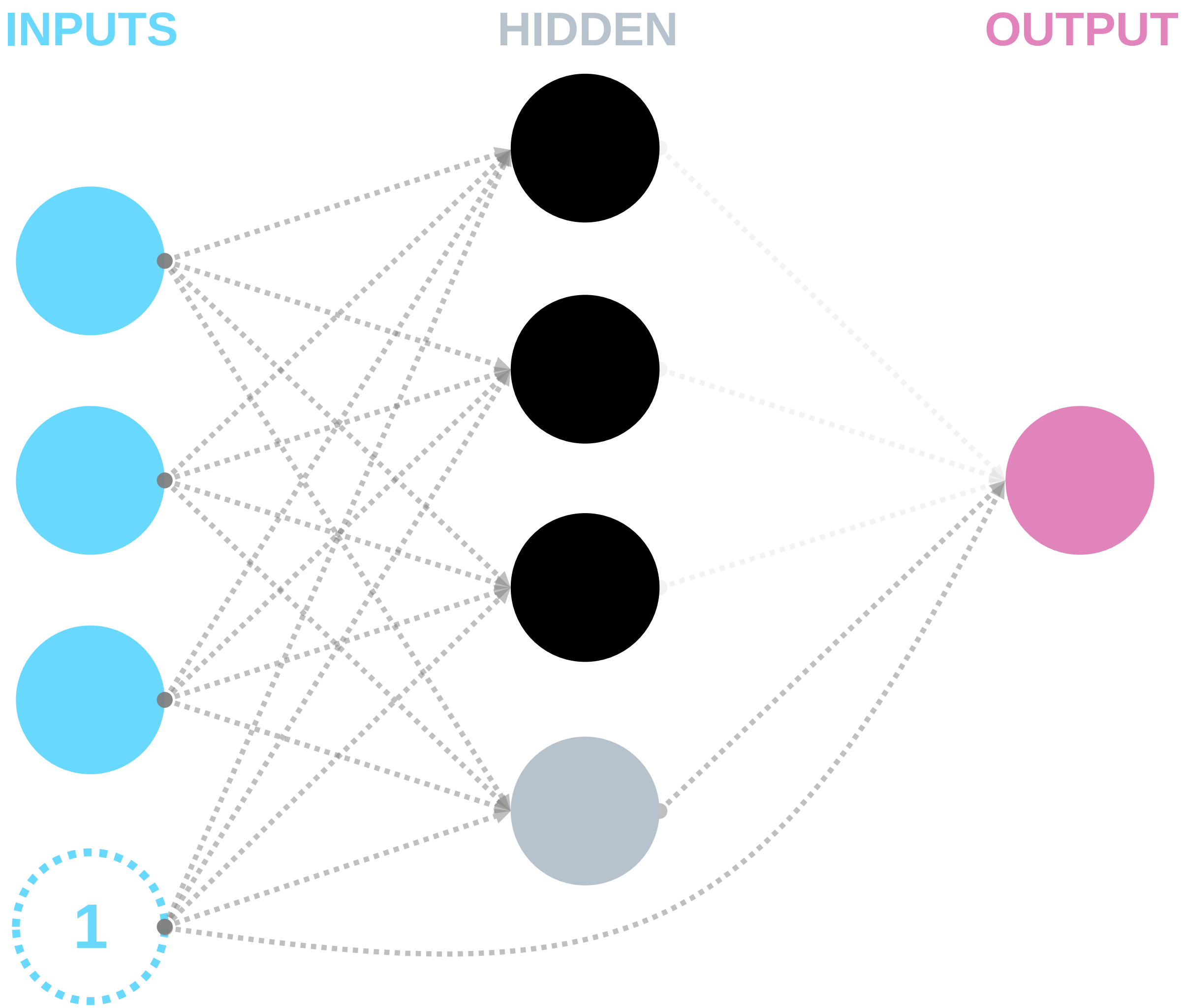
- Randomly set neurons to zero
- Results in an ensemble with an exponential number of members whose parameters are shared
- Primarily used in fully connected layers because of the large number of parameters
- Rarely used in convolutional layers
- Rarely used in recurrent neural networks (if at all between the hidden state and output)



What's new?

Regularization

- Randomly set neurons to zero
- Results in an ensemble with an exponential number of members whose parameters are shared
- Primarily used in fully connected layers because of the large number of parameters
- Rarely used in convolutional layers
- Rarely used in recurrent neural networks (if at all between the hidden state and output)



What's new?

* deprecated

OPTIMIZATION & LEARNING

OPTIMIZATION ALGORITHMS

- AdaGrad
- AdaDelta
- Adam
- RMSProp
- Hessian-Free Optimization
- …

REPARAMETERIZATION

- Batch Normalization
- Weight Normalization
- …

REGULARIZATION

- Dropout
- DropConnect
- …

MODEL ARCHITECTURES

BUILDING BLOCKS

- Spatial/Temporal Pooling
- Attention Mechanism
- Gated Recurrent Units
- Beam-search for sequence generation
- Variable-length sequence modeling
- …

ARCHITECTURES

- Inception
- VGG
- Encoder-Decoder Framework
- End-to-end Models
- …

SOFTWARE

- Theano
  - Blocks + Fuel
  - Keras
  - Lasagne
  - PyLearn2*
- TensorFlow
- Torch7
- Caffe…

GENERAL

- GPUs
- Data

What’s new?
Attention Mechanism in Image Caption Generation

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

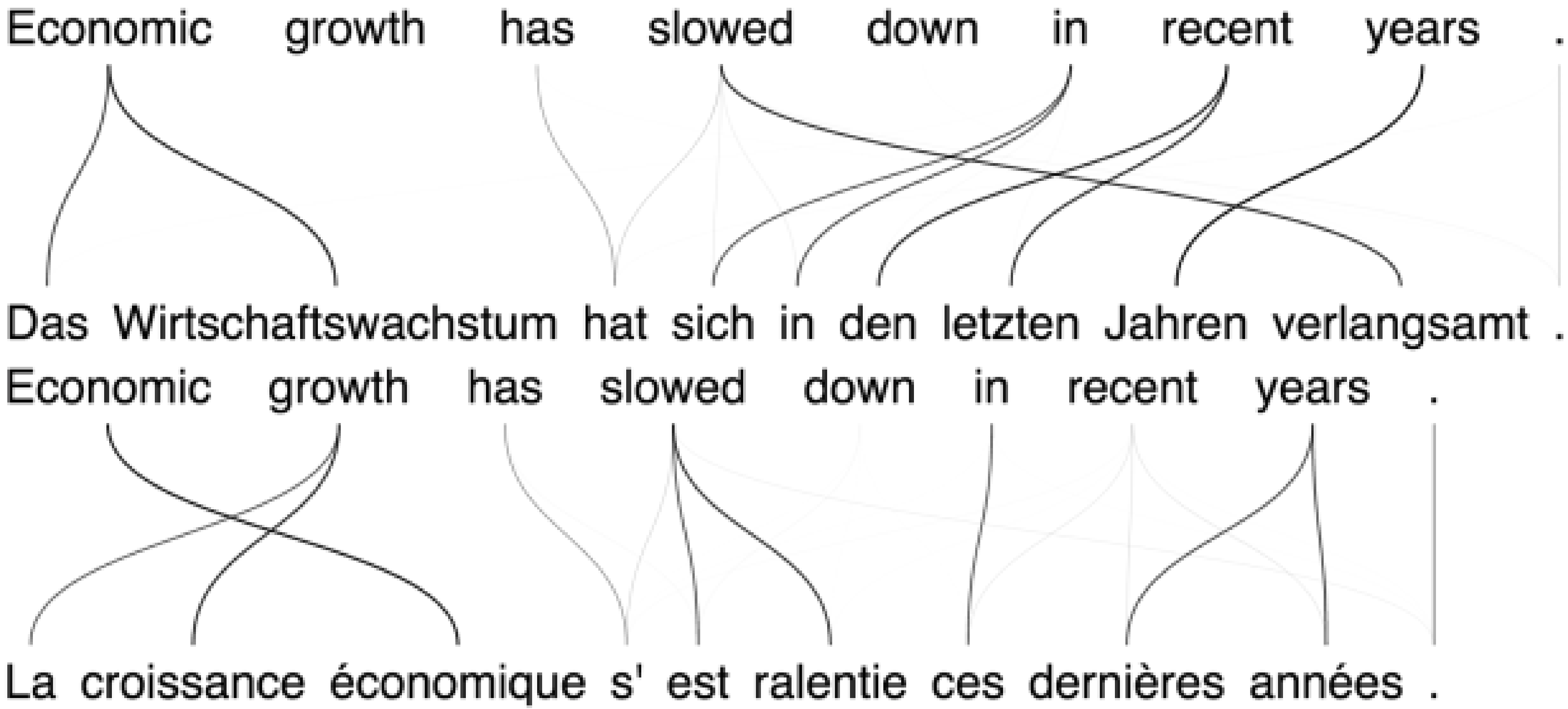
A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

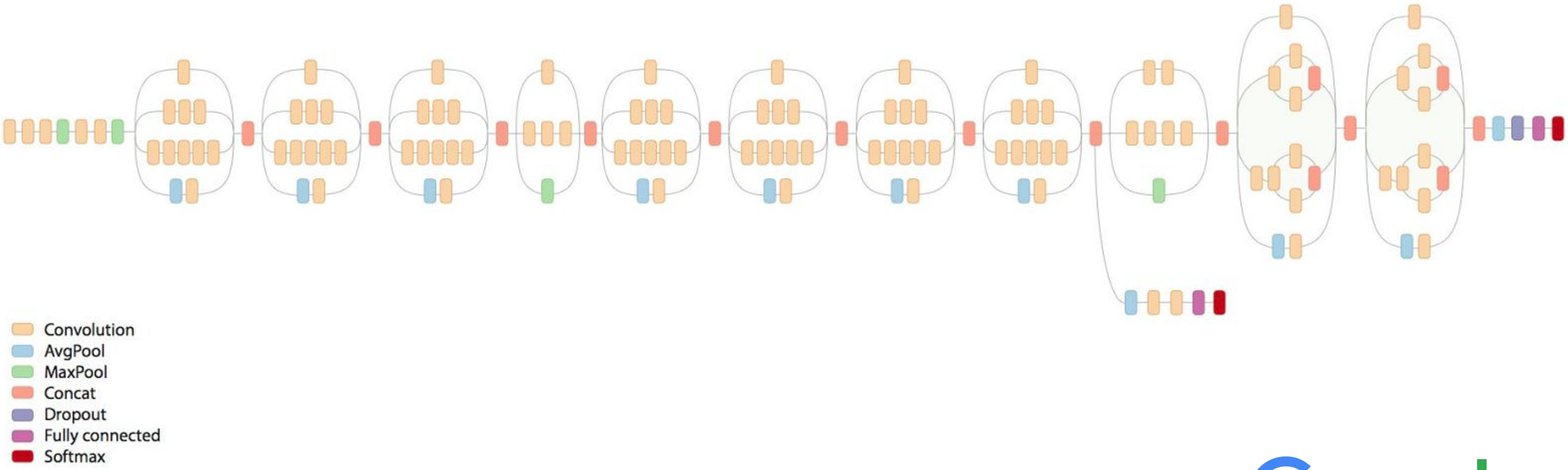
SOURCE:

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ... & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proceedings of The 32nd International Conference on Machine Learning (pp. 2048-2057).

What’s new?
Attention Mechanism in Text Translation

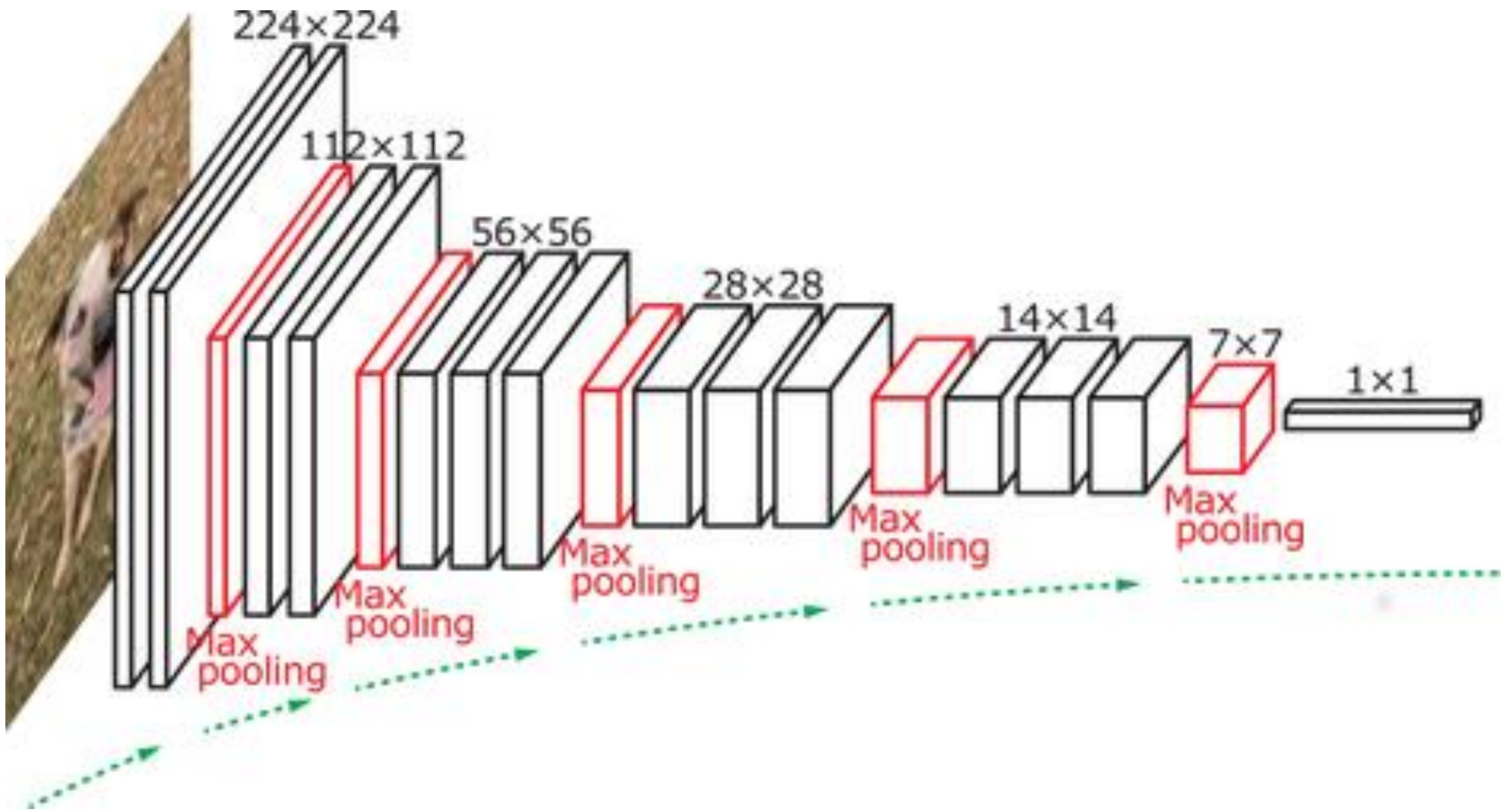


What's new?
Inception Architecture

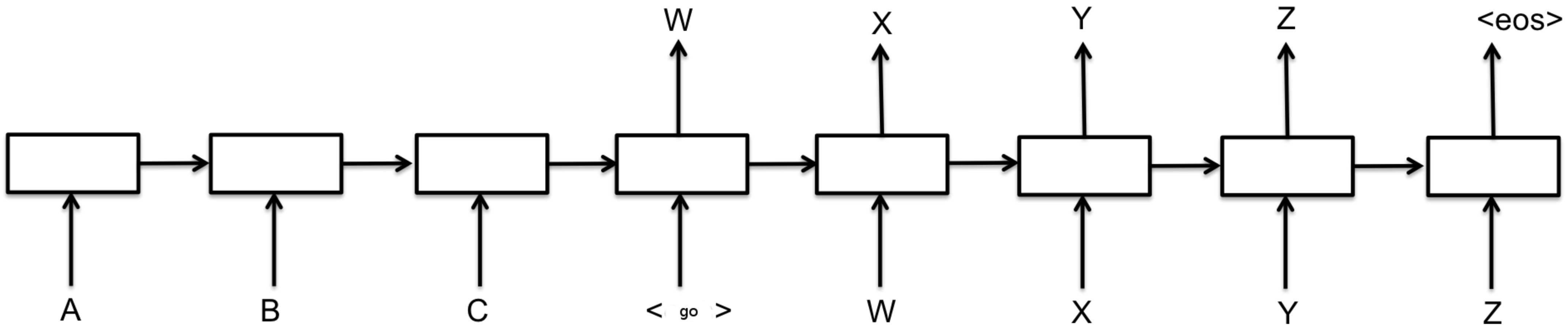


What’s new?
VGG-16 Architecture

- Filter size 3×3
- 2+ successive convolutions with before pooling instead of the common CONV → POOL chain
- Convolution mode “half”
- More layers ⇒ larger capacity
- Parameter-efficient due to small filters



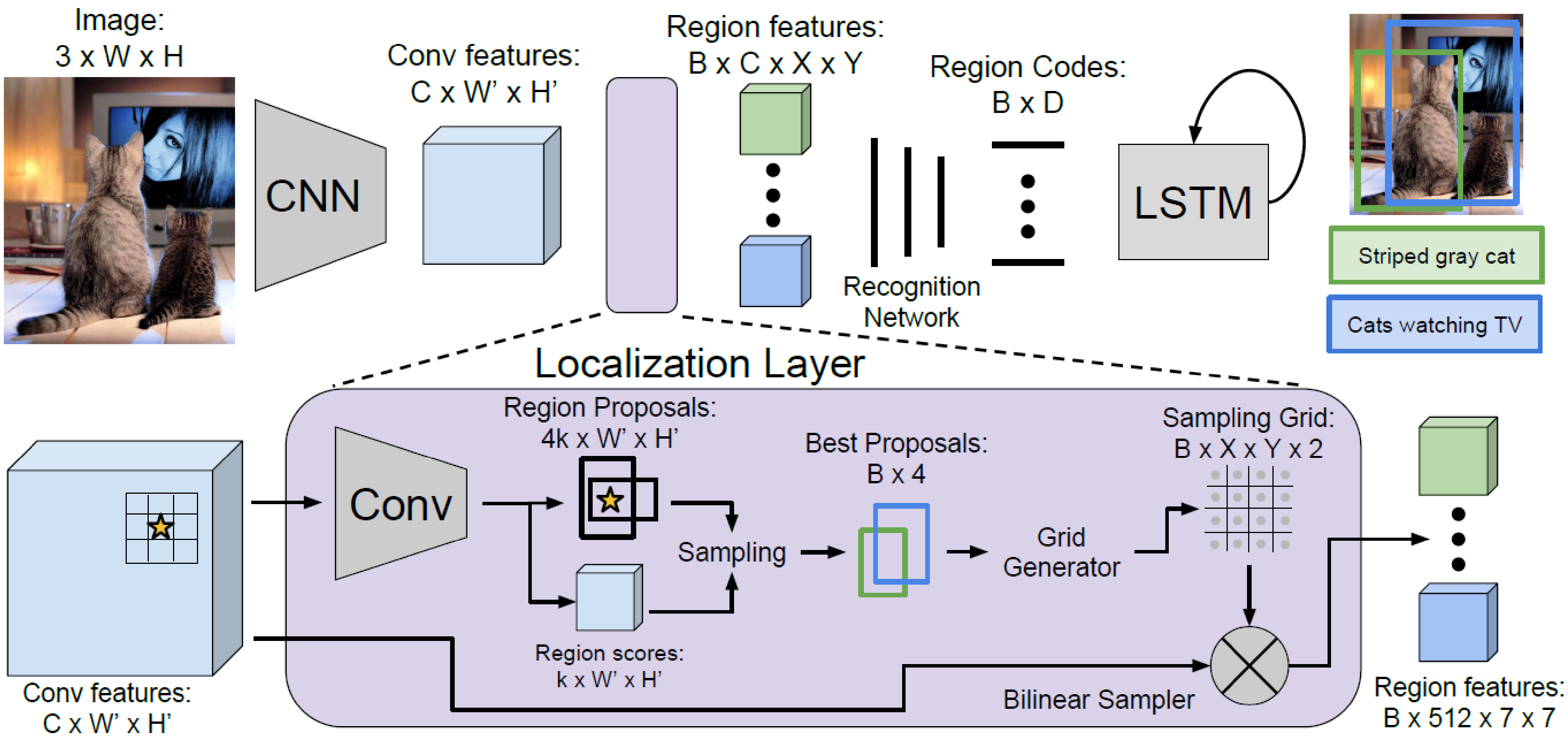
What's new?
Encoder-Decoder Framework



SOURCE: https://www.tensorflow.org/versions/r0.8/tutorials/seq2seq/index.html#sequence-to-sequence-models

What's new?

End-to-end model (object recognition)

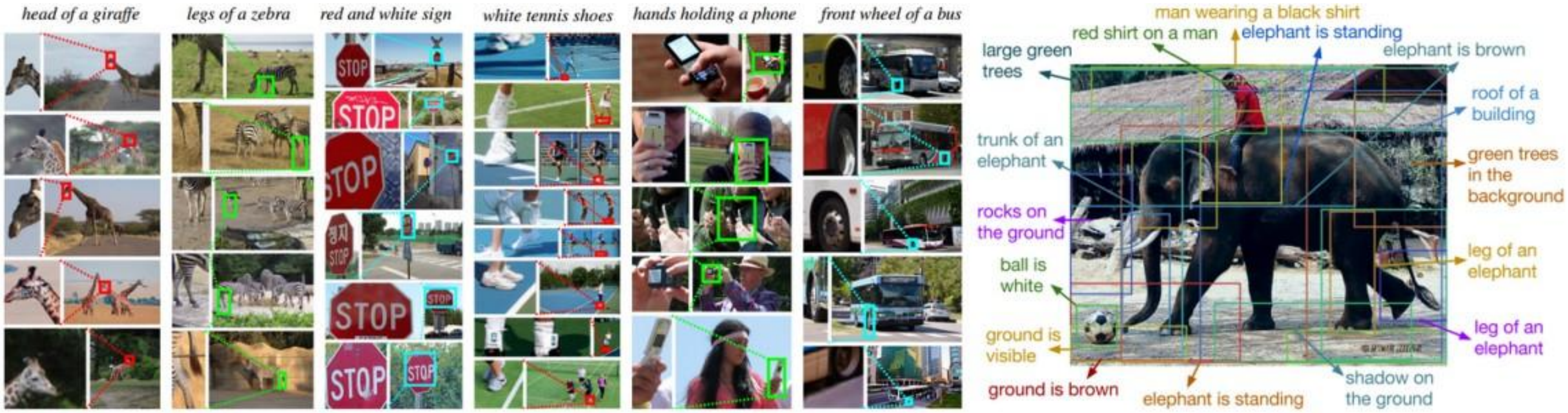


SOURCE:

Johnson, J., Karpathy, A., & Fei-Fei, L. (2015). DenseCap: Fully Convolutional Localization Networks for Dense Captioning. *arXiv preprint arXiv:1511.07571*.

What's new?

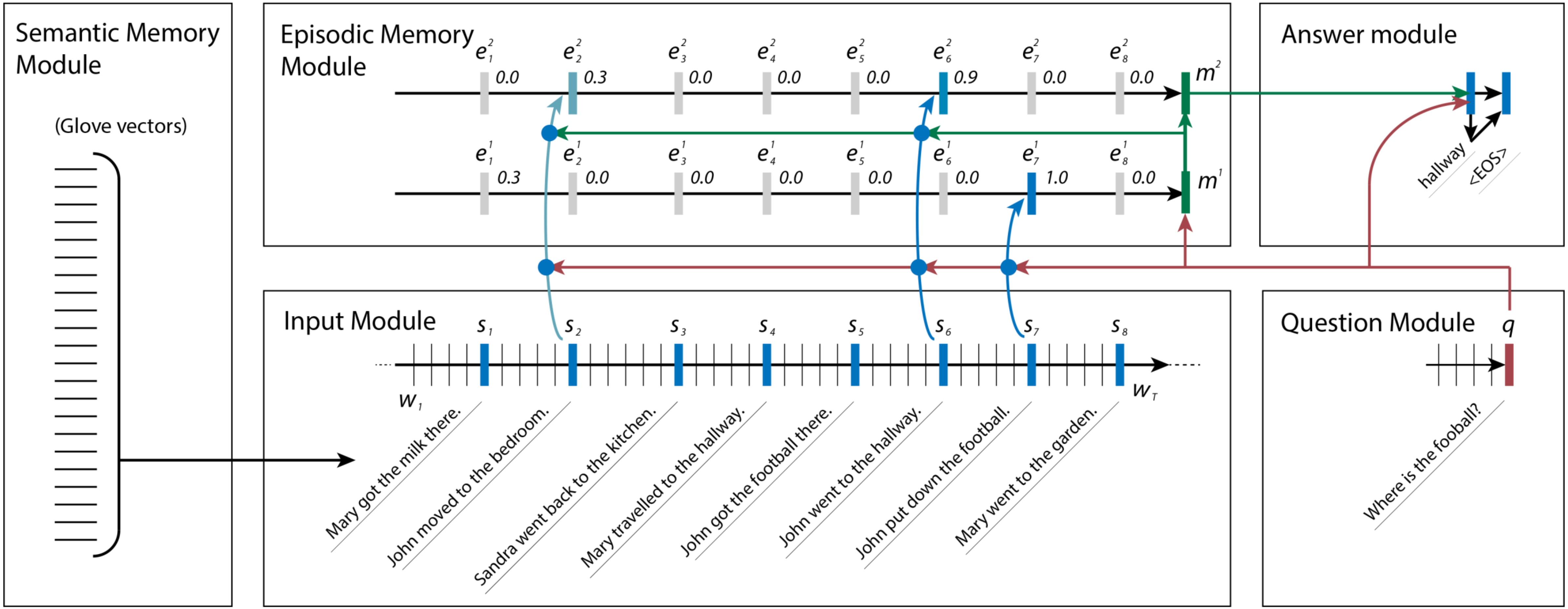
End-to-end model (object recognition)



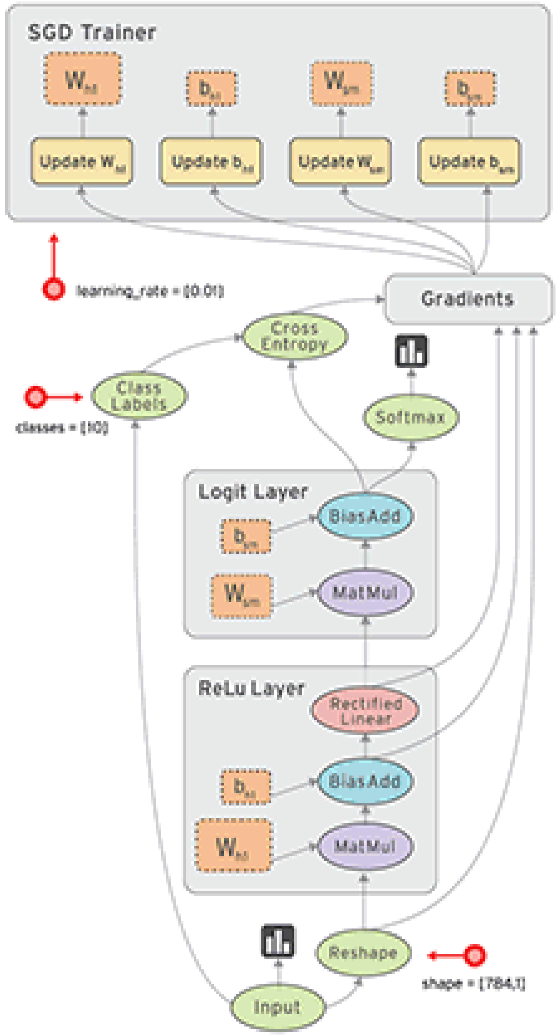
SOURCE:
Johnson, J., Karpathy, A., & Fei-Fei, L. (2015). DenseCap: Fully Convolutional Localization Networks for Dense Captioning. *arXiv preprint arXiv:1511.07571*.

What’s new?

End-to-end model (question answering)



Data Flow Graphs / Computation Graphs



What's new?

* deprecated

OPTIMIZATION & LEARNING

OPTIMIZATION ALGORITHMS

- AdaGrad
- AdaDelta
- Adam
- RMSProp
- Hessian-Free Optimization
- …

REPARAMETERIZATION

- Batch Normalization
- Weight Normalization
- …

REGULARIZATION

- Dropout
- DropConnect
- …

MODEL ARCHITECTURES

BUILDING BLOCKS

- Spatial/Temporal Pooling
- Attention Mechanism
- Gated Recurrent Units
- Beam-search for sequence generation
- Variable-length sequence modeling
- …

ARCHITECTURES

- Inception (Google)
- VGG (Oxford University)
- Encoder-Decoder Framework
- End-to-end Models
- …

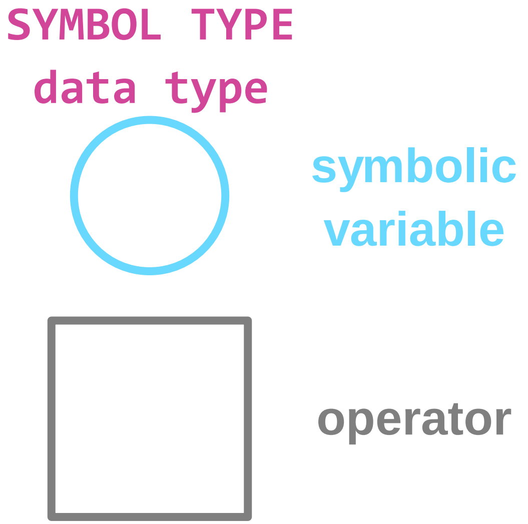
SOFTWARE

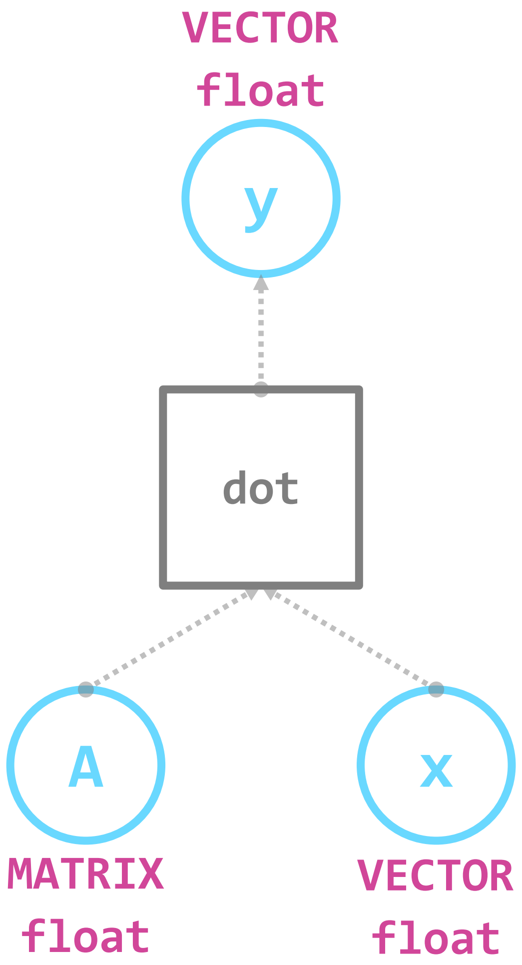
- Theano
  - Blocks + Fuel
  - Keras
  - Lasagne
  - PyLearn2*
- TensorFlow
- Torch7
- Caffe…

GENERAL

- GPUs
- Data

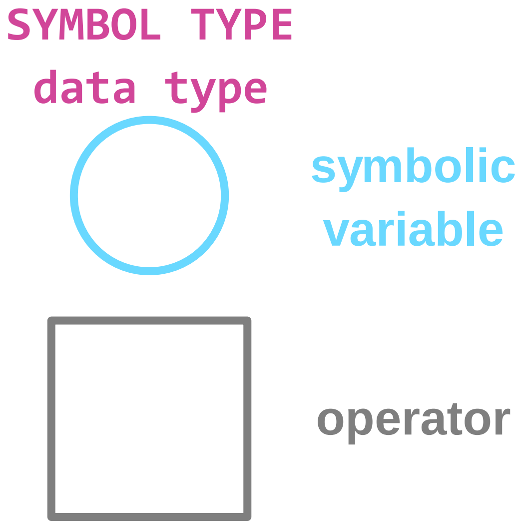
Computation Graphs
Matrix-Vector Multiplication

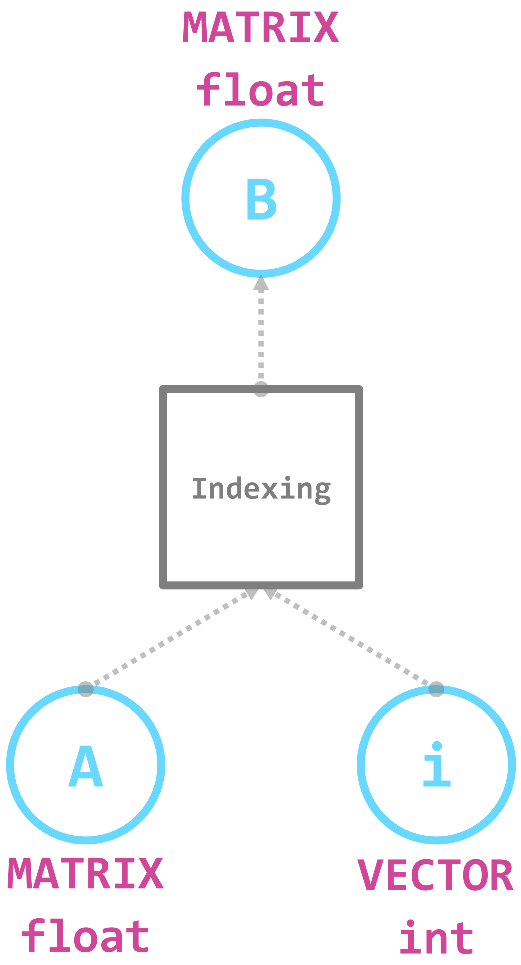


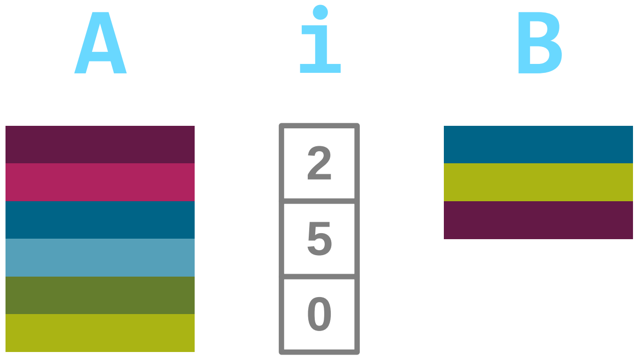


$$y \leftarrow Ax$$
$$A \in \mathbb{R}^{m \times n}$$
$$x \in \mathbb{R}^{n}$$
$$y \in \mathbb{R}^{m}$$

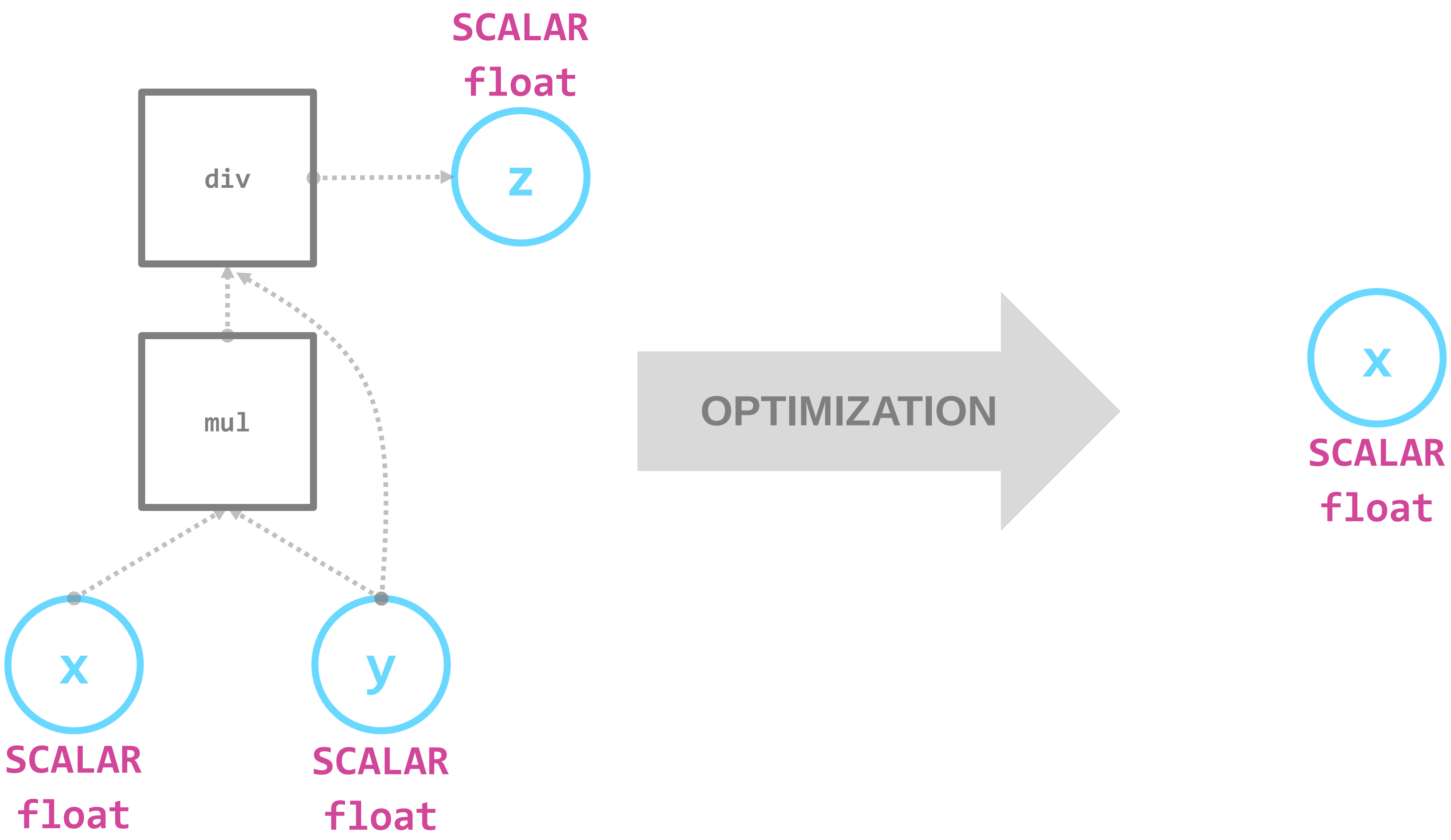
Computation Graphs
Indexing





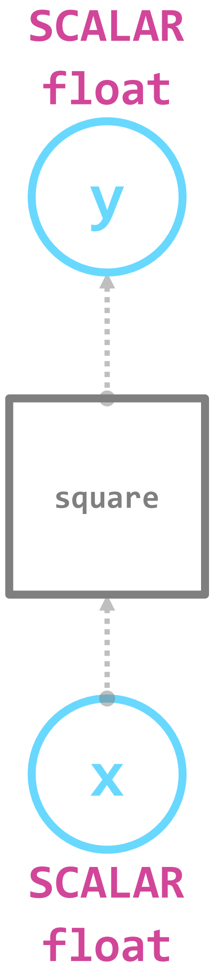


$$z \leftarrow \frac{xy}{y}$$



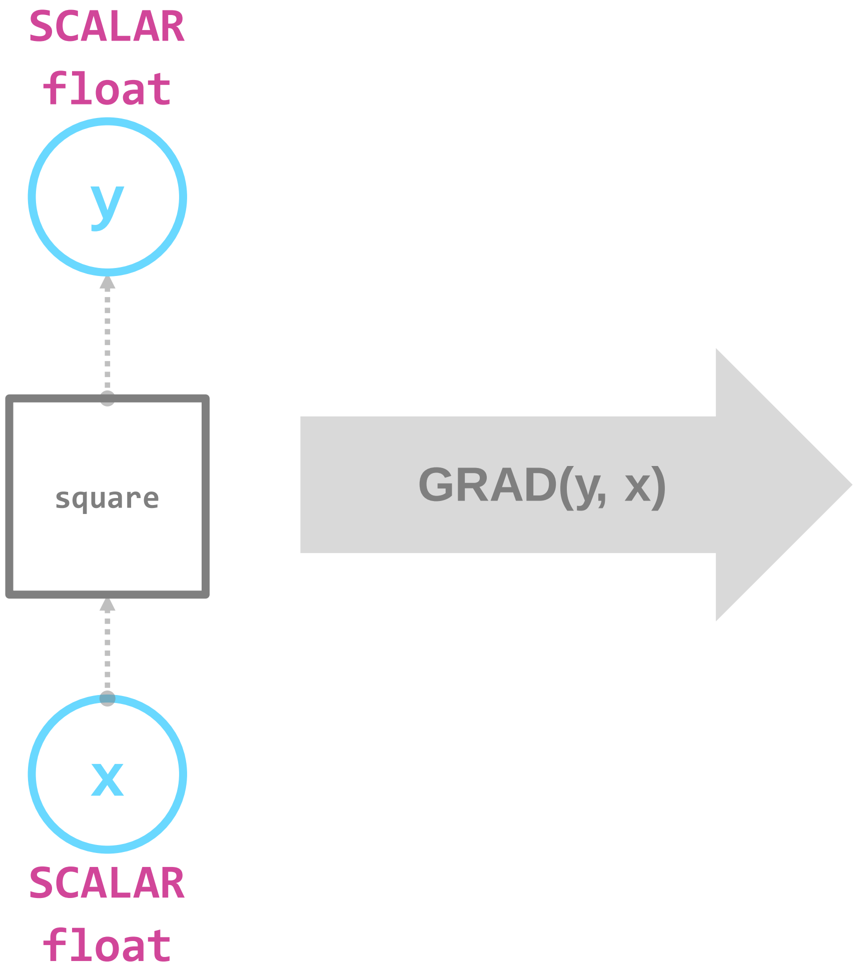
Computation Graphs
Automatic Differentiation

$$y \leftarrow x^2$$



Computation Graphs
Automatic Differentiation

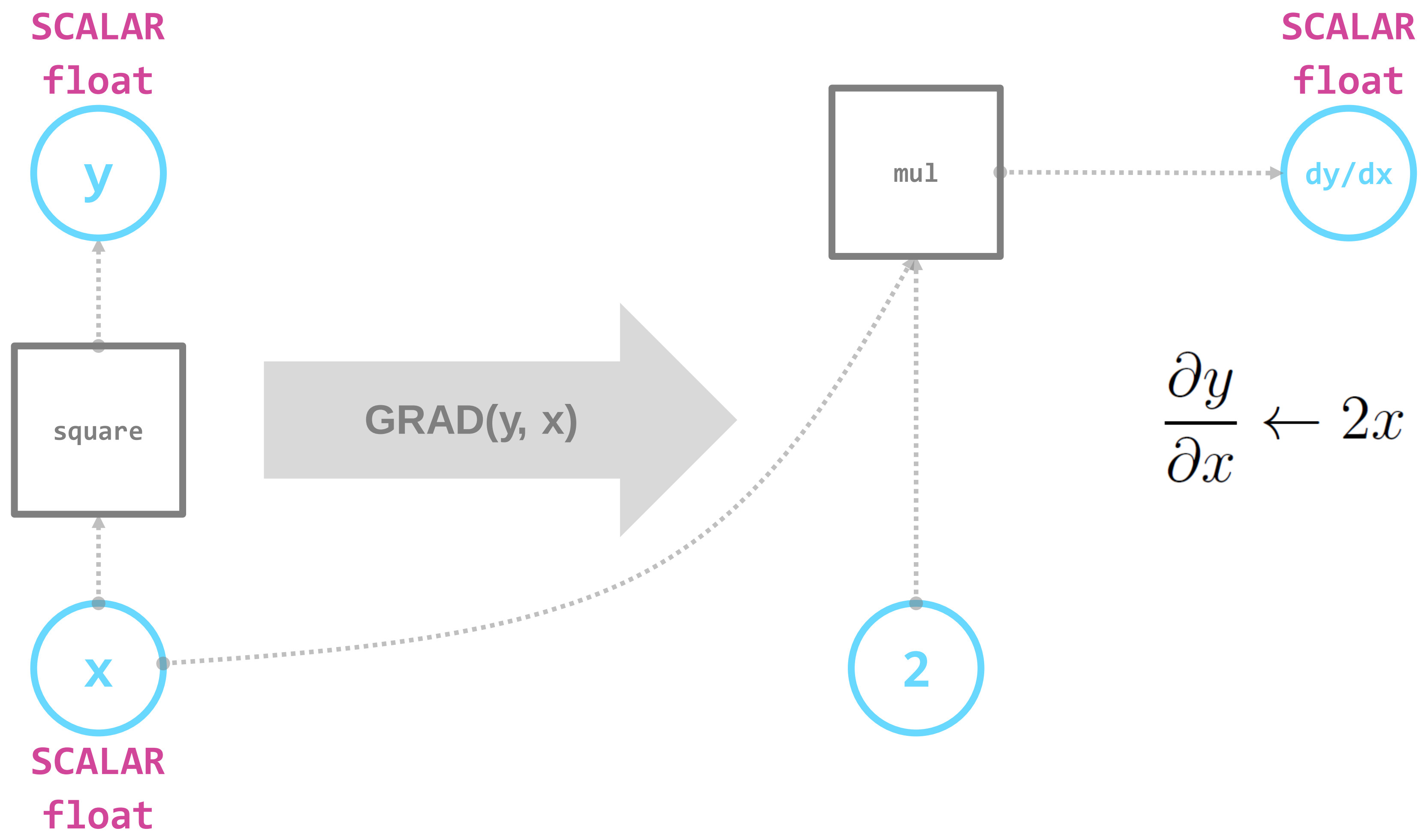
$$y \leftarrow x^2$$



2016/06/01

Computation Graphs
Automatic Differentiation

$$y \leftarrow x^2$$



$$\frac{\partial y}{\partial x} \leftarrow 2x$$

AUTOMATIC DIFFERENTIATION

IS AN

EXTREMELY POWERFUL FEATURE

FOR DEVELOPING MODELS WITH

DIFFERENTIABLE OPTIMIZATION OBJECTIVES

What's new?

* deprecated

OPTIMIZATION & LEARNING

OPTIMIZATION ALGORITHMS

- AdaGrad
- AdaDelta
- Adam
- RMSProp
- Hessian-Free Optimization
- …

REPARAMETERIZATION

- Batch Normalization
- Weight Normalization
- …

REGULARIZATION

- Dropout
- DropConnect
- …

MODEL ARCHITECTURES

BUILDING BLOCKS

- Spatial/Temporal Pooling
- Attention Mechanism
- Gated Recurrent Units
- Beam-search for sequence generation
- Variable-length sequence modeling
- …

ARCHITECTURES

- Inception (Google)
- VGG (Oxford University)
- Encoder-Decoder Framework
- End-to-end Models
- …

SOFTWARE

- Theano
  - Blocks + Fuel
  - Keras
  - Lasagne
  - PyLearn2*
- TensorFlow
- Torch7
- Caffe…

GENERAL

- GPUs
- Data

What’s new?

GPUs

NVIDIA®

What's new?

DEEP LEARNING is NOT only meant literally, but more importantly it is about learning solutions to problems in a **fully automated** way.

Recommended Material

Module 1: Neural Networks

Image Classification: Data-driven Approach, k-Nearest Neighbor, train/val/test splits
L1/L2 distances, hyperparameter search, cross-validation

Linear classification: Support Vector Machine, Softmax
parameteric approach, bias trick, hinge loss, cross-entropy loss, L2 regularization, web demo

Optimization: Stochastic Gradient Descent
optimization landscapes, local search, learning rate, analytic/numerical gradient

Backpropagation, Intuitions
chain rule interpretation, real-valued circuits, patterns in gradient flow

Neural Networks Part 1: Setting up the Architecture
model of a biological neuron, activation functions, neural net architecture, representational power

Neural Networks Part 2: Setting up the Data and the Loss
preprocessing, weight initialization, batch normalization, regularization (L2/dropout), loss functions

Neural Networks Part 3: Learning and Evaluation
gradient checks, sanity checks, babysitting the learning process, momentum (+nesterov), second-order methods, Adagrad/RMSprop, hyperparameter optimization, model ensembles

Module 2: Convolutional Neural Networks

Convolutional Neural Networks: Architectures, Convolution / Pooling Layers
layers, spatial arrangement, layer patterns, layer sizing patterns, AlexNet/ZFNet/VGGNet case studies, computational considerations

Understanding and Visualizing Convolutional Neural Networks
tSNE embeddings, deconvnets, data gradients, fooling ConvNets, human comparisons

Transfer Learning and Fine-tuning Convolutional Neural Networks

Course Instructors


Fei-Fei Li

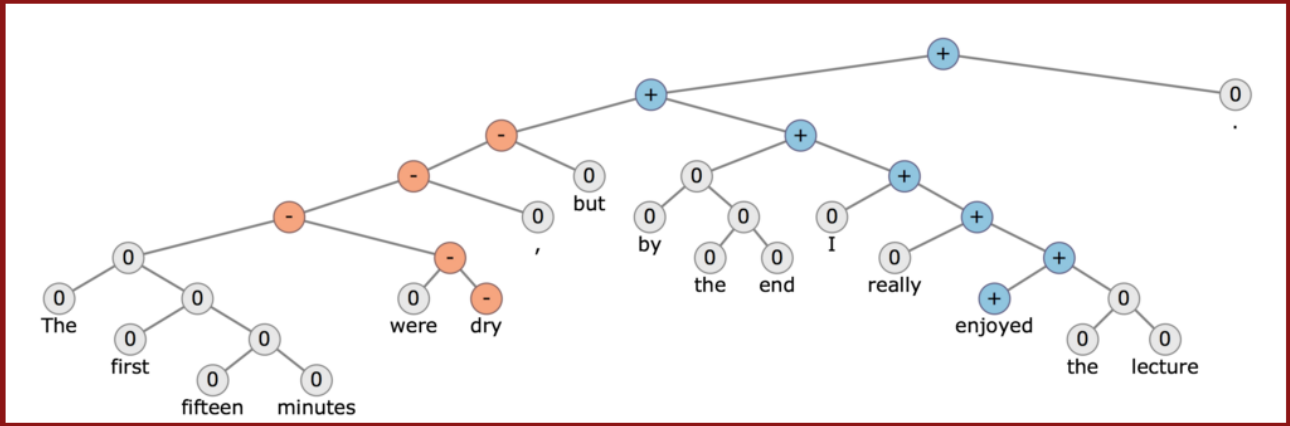

Andrej Karpathy


Justin Johnson

http://cs231n.stanford.edu/
http://cs231n.github.io

Recommended Material

CS224d: Deep Learning for Natural Language Processing



Course Description

Natural language processing (NLP) is one of the most important technologies of the information age. Understanding complex language utterances is also a crucial part of artificial intelligence. Applications of NLP are everywhere because people communicate most everything in language: web search, advertisement, emails, customer service, language translation, radiology reports, etc. There are a large variety of underlying tasks and machine learning models powering NLP applications. Recently, deep learning approaches have obtained very high performance across many different NLP tasks. These models can often be trained with a single end-to-end model and do not require traditional, task-specific feature engineering. In this spring quarter course students will learn to implement, train, debug, visualize and invent their own neural network models. The course provides a deep excursion into cutting-edge research in deep learning applied to NLP. The final project will involve training a complex recurrent neural network and applying it to a large scale NLP problem. On the model side we will cover word vector representations, window-based neural networks, recurrent neural networks, long-short-term-memory models, recursive neural networks, convolutional neural networks as well as some very novel models involving a memory component. Through lectures and programming assignments students will learn the necessary engineering tricks for making neural networks work on practical problems.

Course Instructor

Richard Socher

http://cs224d.stanford.edu/

Recommended Material

INTRODUCTION

- Tutorial on Neural Networks (Deep Learning and Unsupervised Feature Learning): http://deeplearning.stanford.edu/wiki/index.php/UFLDL_Tutorial
- Deep Learning for Computer Vision lecture: http://cs231n.stanford.edu (http://cs231n.github.io)
- Deep Learning for NLP lecture: http://cs224d.stanford.edu (http://cs224d.stanford.edu/syllabus.html)
- Deep Learning for NLP (without magic) tutorial: http://lxmls.it.pt/2014/socher-lxmls.pdf (Videos from NAACL 2013: http://nlp.stanford.edu/courses/NAACL2013)
- Bengio's Deep Learning book: http://www.deeplearningbook.org

Recommended Material

PARAMETER INITIALIZATION

- Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." *International Conference on Artificial Intelligence and Statistics*. 2010.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1026-1034).

BATCH NORMALIZATION

- Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of The 32nd International Conference on Machine Learning* (pp. 448-456).
- Cooijmans, T., Ballas, N., Laurent, C., & Courville, A. (2016). Recurrent Batch Normalization. *arXiv preprint arXiv:1603.09025*.

DROPOUT

- Hinton, Geoffrey E., et al. "Improving neural networks by preventing co-adaptation of feature detectors." *arXiv preprint arXiv:1207.0580* (2012).
- Srivastava, Nitish, et al. "Dropout: A simple way to prevent neural networks from overfitting." *The Journal of Machine Learning Research* 15.1 (2014): 1929-1958.

Recommended Material

OPTIMIZATION & TRAINING

- Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, *12*, 2121-2159.
- Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.
- Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, *4*, 2.
- Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013). On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML)* (pp. 1139-1147).
- Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization.*arXiv preprint arXiv:1412.6980*.
- Martens, J., & Sutskever, I. (2012). Training deep and recurrent networks with hessian-free optimization. In *Neural networks: Tricks of the trade* (pp. 479-535). Springer Berlin Heidelberg.

Recommended Material

COMPUTER VISION

- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1097-1105).
- Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). DeepFace: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1701-1708).
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-9).
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Jaderberg, M., Simonyan, K., & Zisserman, A. (2015). Spatial transformer networks. In *Advances in Neural Information Processing Systems* (pp. 2008-2016).
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems* (pp. 91-99).
- Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ... & Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proceedings of The 32nd International Conference on Machine Learning* (pp. 2048-2057).
- Johnson, J., Karpathy, A., & Fei-Fei, L. (2015). DenseCap: Fully Convolutional Localization Networks for Dense Captioning. *arXiv preprint arXiv:1511.07571*.

Recommended Material

NATURAL LANGUAGE PROCESSING

- Bengio, Y., Schwenk, H., Senécal, J. S., Morin, F., & Gauvain, J. L. (2006). Neural probabilistic language models. In *Innovations in Machine Learning* (pp. 137-186). Springer Berlin Heidelberg.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, *12*, 2493-2537.
- Mikolov, T. (2012). *Statistical language models based on neural networks* (Doctoral dissertation, PhD thesis, Brno University of Technology. 2012.)
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (pp. 3111-3119).
- Mikolov, T., Yih, W. T., & Zweig, G. (2013). Linguistic Regularities in Continuous Space Word Representations. In *HLT-NAACL* (pp. 746-751).
- Socher, R. (2014). *Recursive Deep Learning for Natural Language Processing and Computer Vision* (Doctoral dissertation, Stanford University).
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.