Convolutional Neural Networks

——Trends in My view

Shenglin Zhao
Department of Computer Science and Engineering
The Chinese University of Hong Kong
slzhao@cse.cuhk.edu.hk

Outline

- Structure
  - From deep to deeper
  - CNN Variants
- Application in other areas
  - Using the structure
  - Using the generated features
- Related papers
  - Classic paper
  - Latest studies

Structure

From deep to deeper

- Milestone: Residual Network
  - He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
  - Citation: 3854

From deep to deeper

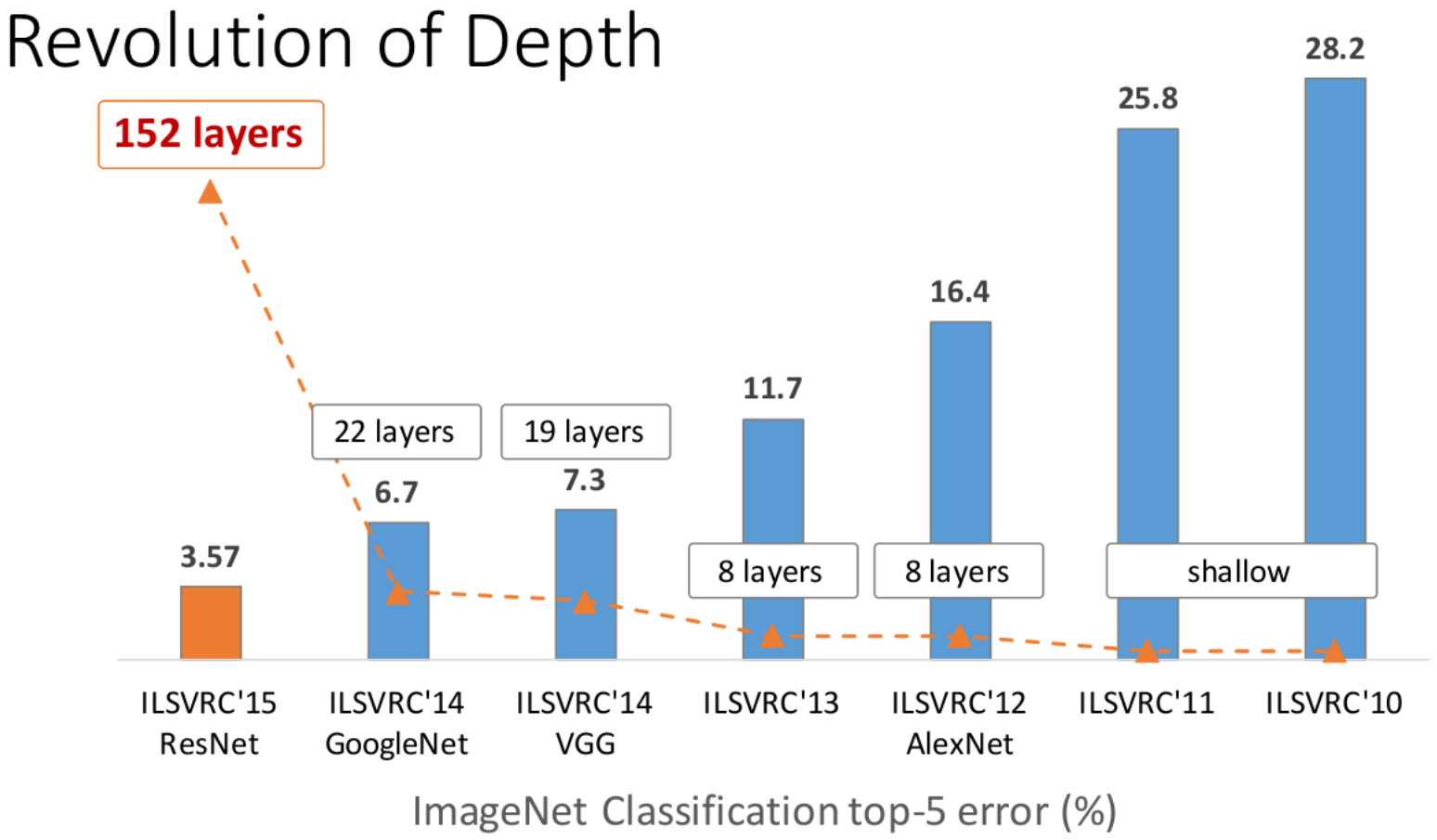
- Milestone: Residual Network

ResNets @ ILSVRC & COCO 2015 Competitions

- **1st places in all five main tracks**
  - ImageNet Classification: "*Ultra-deep*" 152-layer nets
  - ImageNet Detection: 16% better than 2nd
  - ImageNet Localization: 27% better than 2nd
  - COCO Detection: 11% better than 2nd
  - COCO Segmentation: 12% better than 2nd

Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

From deep to deeper

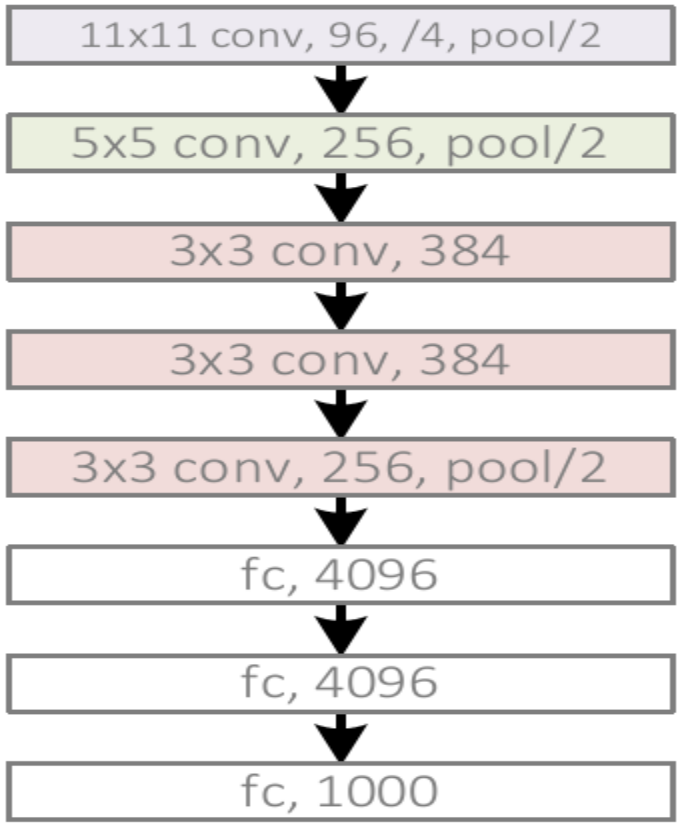


Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

From deep to deeper

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

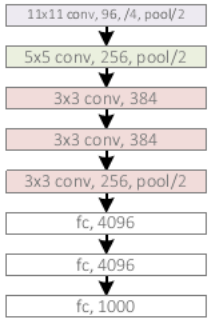


Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

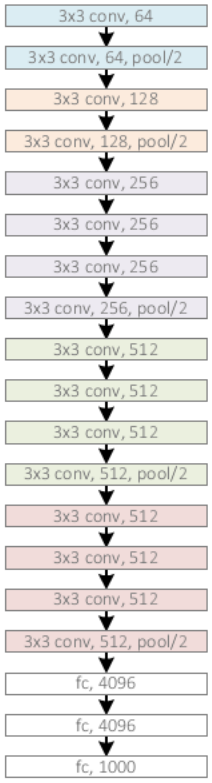
From deep to deeper

Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



GoogleNet, 22 layers
(ILSVRC 2014)

Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

From deep to deeper

Revolution of Depth

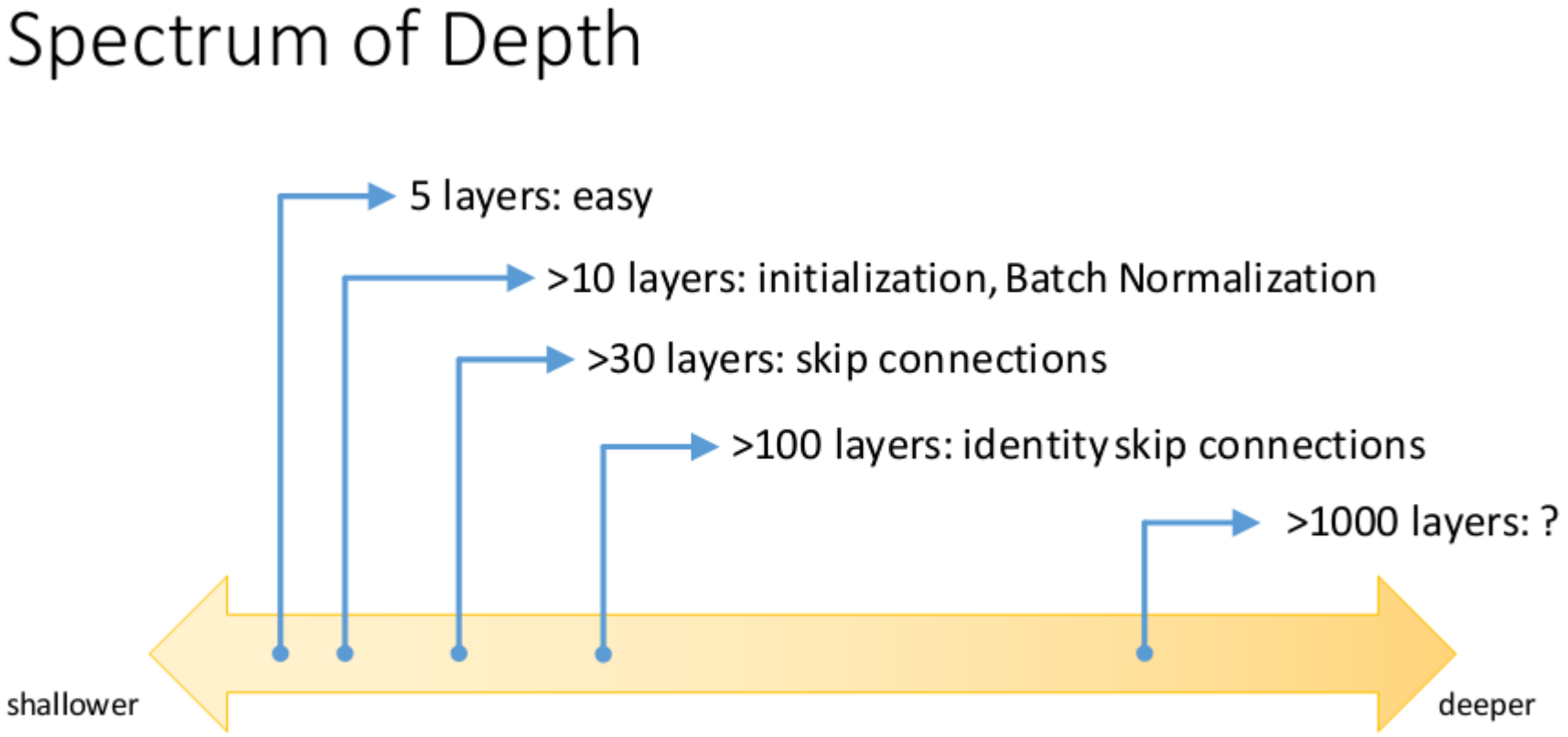
AlexNet, 8 layers
(ILSVRC 2012)

VGG, 19 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

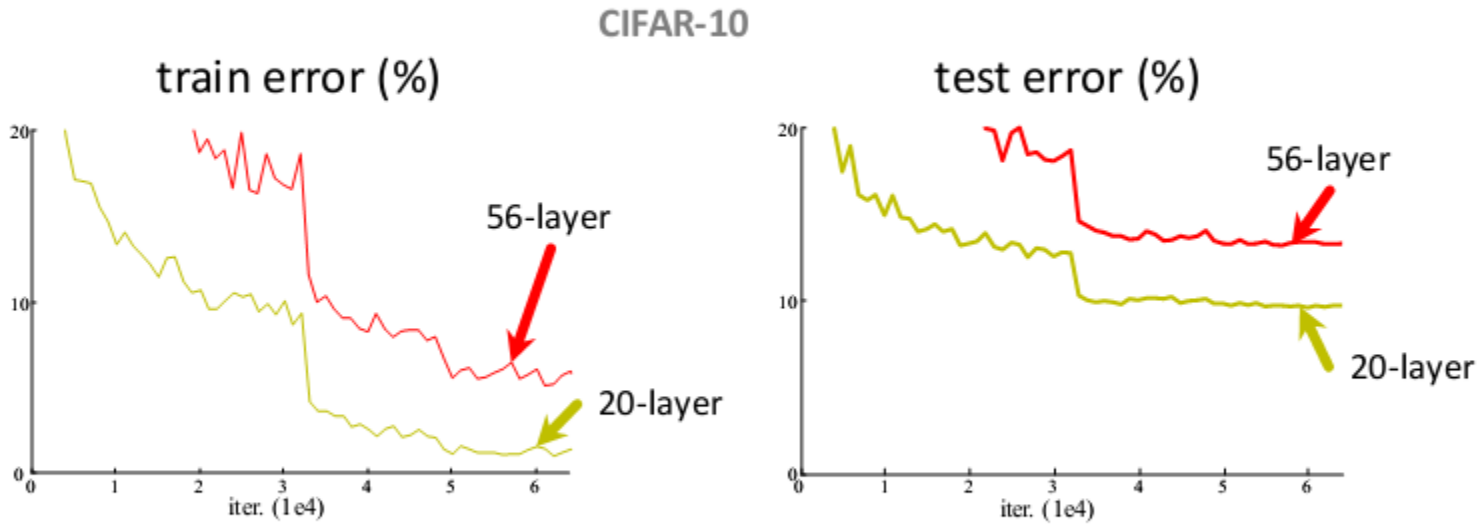
From deep to deeper



Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

From deep to deeper

Simply stacking layers?

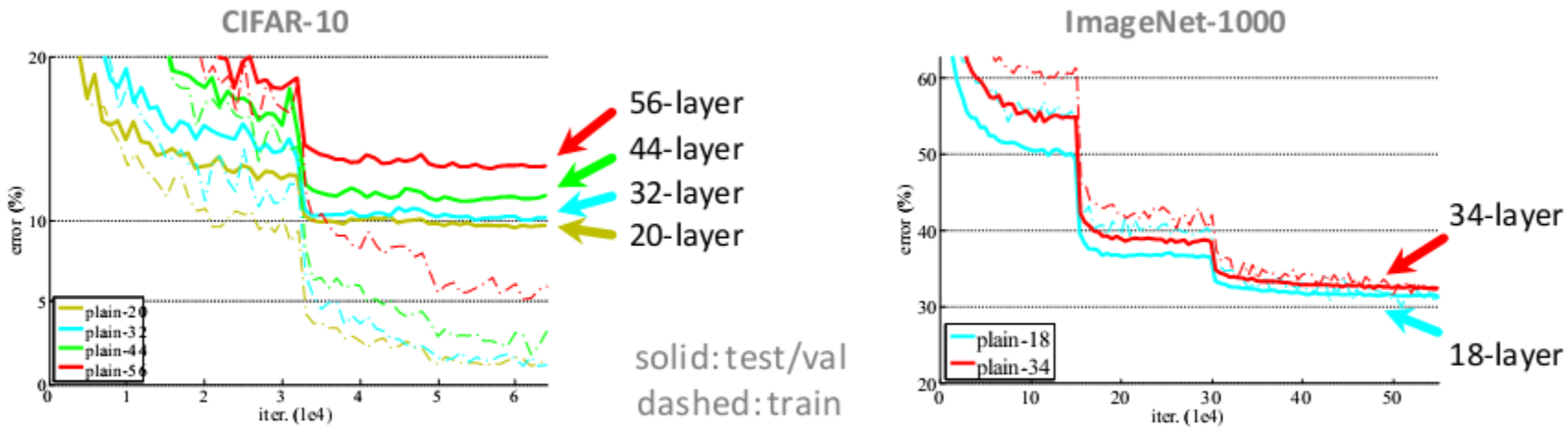


- *Plain* nets: stacking 3x3 conv layers…
- 56-layer net has **higher training error** and test error than 20-layer net

Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

From deep to deeper

Simply stacking layers?



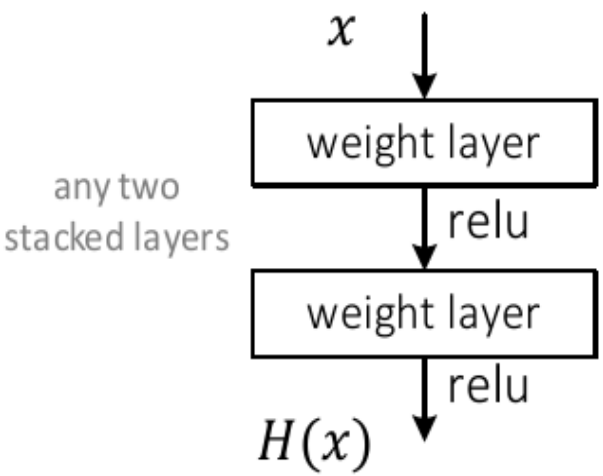
- "Overly deep" plain nets have **higher training error**
- A general phenomenon, observed in many datasets

Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

From deep to deeper

Deep Residual Learning

- Plaint net



$H(x)$ is any desired mapping,

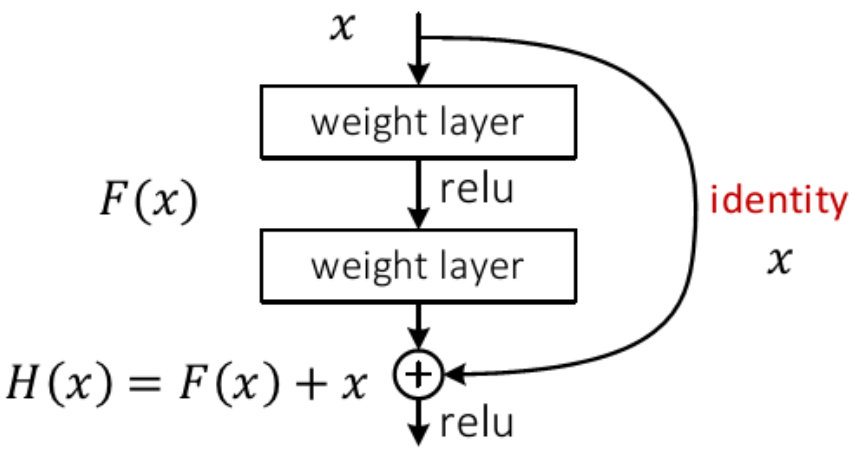
hope the 2 weight layers fit $H(x)$

Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

From deep to deeper

Deep Residual Learning

- Residual net



$H(x)$ is any desired mapping,

~~hope the 2 weight layers fit $H(x)$~~

hope the 2 weight layers fit $F(x)$

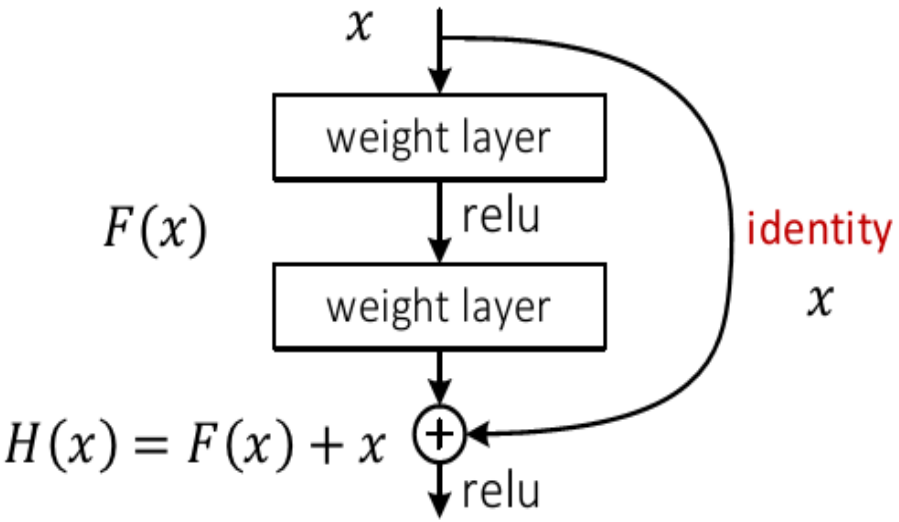
let $H(x) = F(x) + x$

Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

From deep to deeper

Deep Residual Learning

- $F(x)$ is a residual mapping w.r.t. identity



- If identity were optimal, easy to set weights as 0
- If optimal mapping is closer to identity, easier to find small fluctuations

Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

From deep to deeper

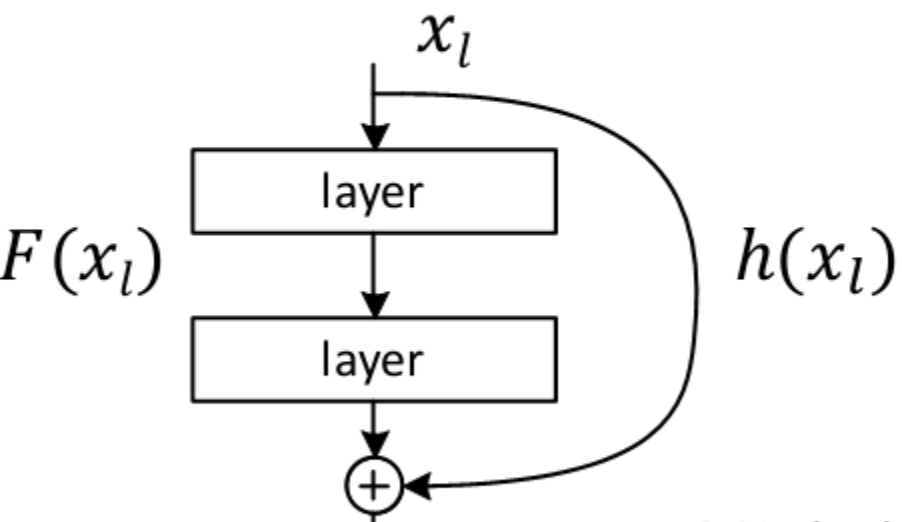
On the Importance of Identity Mapping

From 100 layers to 1000 layers

Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

From deep to deeper

On identity mappings for **optimization**



- shortcut mapping: h = identity
- ~~after-add mapping: f = ReLU~~
- What if f = identity?

Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

From deep to deeper

Very smooth forward propagation

$$x_{l+1} = x_l + F(x_l)$$

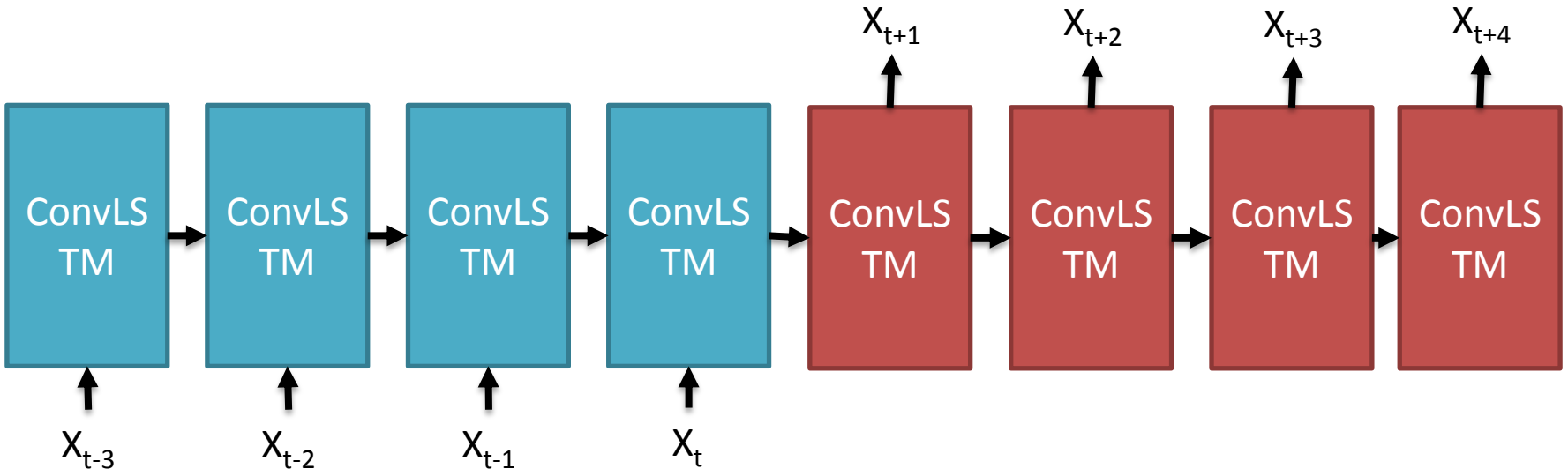
$$x_{l+2} = x_{l+1} + F(x_{l+1})$$

Kaiming He. Deep Residual Networks: Deep Learning Gets Way Deeper. ICML 2016 tutorial.

CNN Variants

- Recurrent CNN
  - Lai S, Xu L, Liu K, et al. Recurrent Convolutional Neural Networks for Text Classification[C]//AAAI. 2015, 333: 2267-2273.
  - Pinheiro P, Collobert R. Recurrent convolutional neural networks for scene labeling[C]//International Conference on Machine Learning. 2014: 82-90.
- Convolutional LSTM
  - Xingjian S H I, Chen Z, Wang H, et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting[C]//Advances in neural information processing systems. 2015: 802-810.
- PixelCNN
  - Salimans T, Karpathy A, Chen X, et al. PixelCNN++: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications[J]. arXiv preprint arXiv:1701.05517, 2017.
  - van den Oord A, Kalchbrenner N, Espeholt L, et al. Conditional image generation with pixelcnn decoders[C]//Advances in Neural Information Processing Systems. 2016: 4790-4798.

Convolutional LSTM



Comparison between FC-LSTM & ConvLSTM

FC-LSTM

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o) \\
h_t &= o_t \circ \tanh(c_t)
\end{aligned}
$$

Input & state at a timestamp are 1D vectors. Dimensions of the state can be permuted without affecting the overall structure.

ConvLSTM

$$
\begin{aligned}
i_t &= \sigma(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ \mathcal{C}_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ \mathcal{C}_{t-1} + b_f) \\
\mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ \tanh(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ \mathcal{C}_t + b_o) \\
\mathcal{H}_t &= o_t \circ \tanh(\mathcal{C}_t)
\end{aligned}
$$

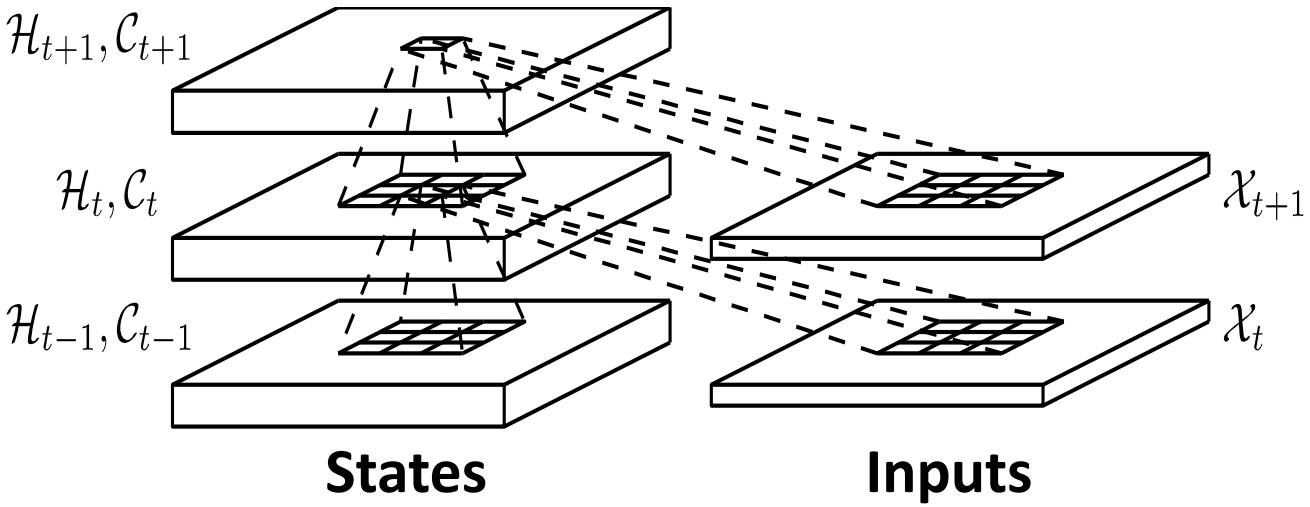
Input & state at a timestamp are 3D tensors. Convolution is used for both input-to-state and state-to-state connection.

Use Hadamard product to keep the constant error carousel (CEC) property of cells

Convolutional LSTM

Using 'state of the outside world' for boundary grids. Zero padding is used to indicate 'total ignorance' of the outside.

In fact, other padding strategies (learn the padding) can be used, we just choose the simplest one.



FC-LSTM can be viewed as a special case of ConvLSTM with all features standing on a single cell.

For convolutional recurrence, 1X1 kernel and larger kernels are totally different!

Later states → Larger receptive field

Application

Using the structure

- CNN in NLP
  - Representative:
    - Kim Y. Convolutional neural networks for sentence classification[J]. arXiv preprint arXiv:1408.5882, 2014. EMNLP short paper.
    - Citation: 1206
  - Latest:
    - Gehring J, Auli M, Grangier D, et al. Convolutional Sequence to Sequence Learning[J]. arXiv preprint arXiv:1705.03122, 2017. ICML.
    - Citation: 25

CNN for sentence classificaiton

$$\mathbf{x}_{1:n} = \mathbf{x}_1 \oplus \mathbf{x}_2 \oplus \ldots \oplus \mathbf{x}_n \qquad c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b) \qquad \mathbf{c} = [c_1, c_2, \ldots, c_{n-h+1}]$$

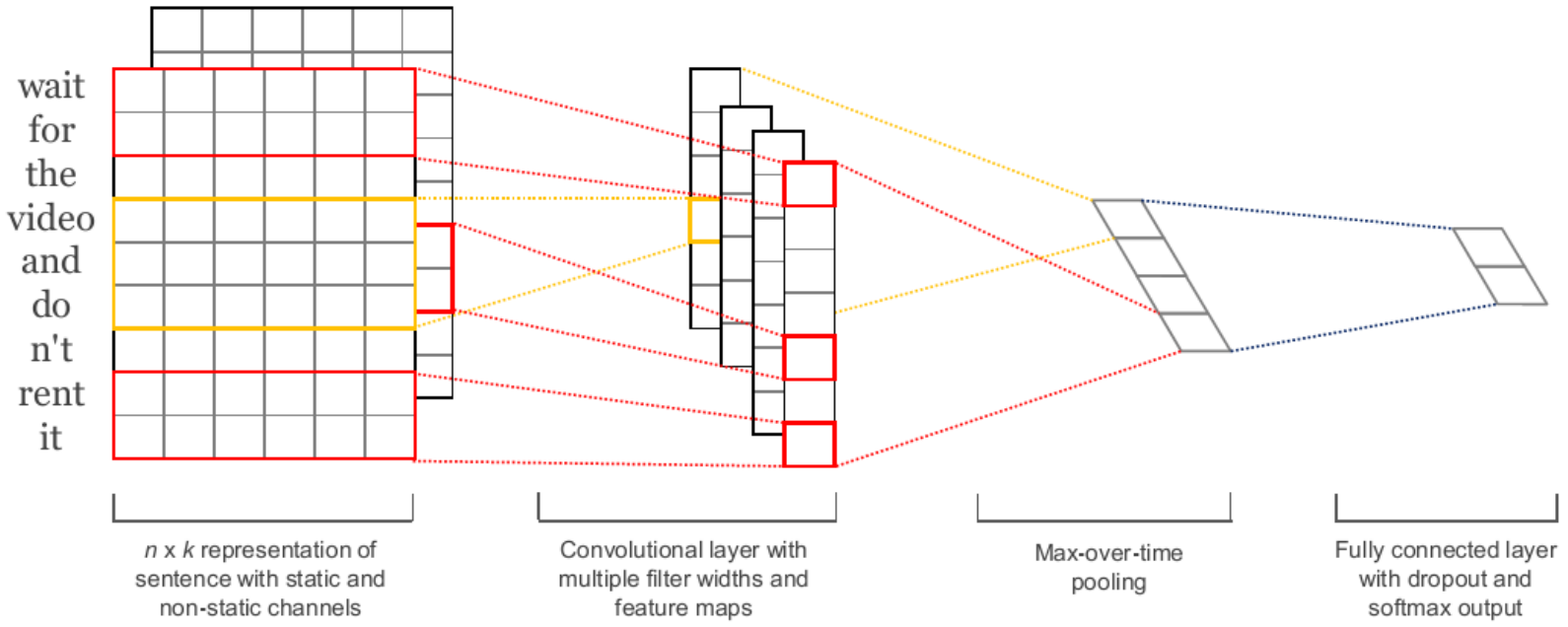
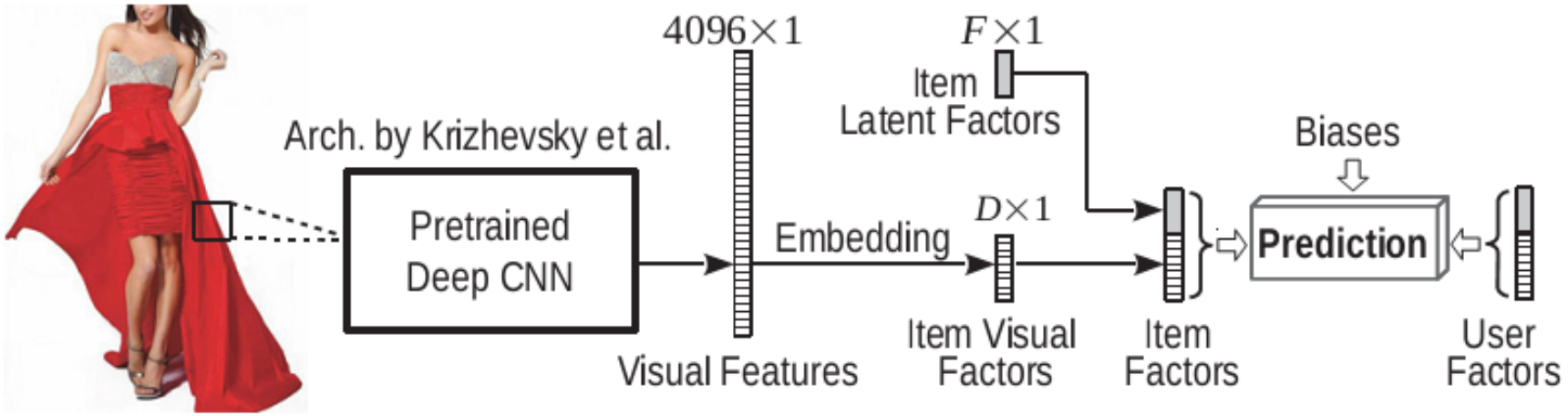


Figure 1: Model architecture with two channels for an example sentence.

Using the generated feature

- CNN in RS:
  - Representative
    - He R, McAuley J. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback[C]//AAAI. 2016: 144-150.
    - Citation: 42
  - Latest
    - Wang S, Wang Y, Tang J, et al. What your images reveal: Exploiting visual contents for point-of-interest recommendation[C]//Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017: 391-400.
    - Citation: 11

VBPR

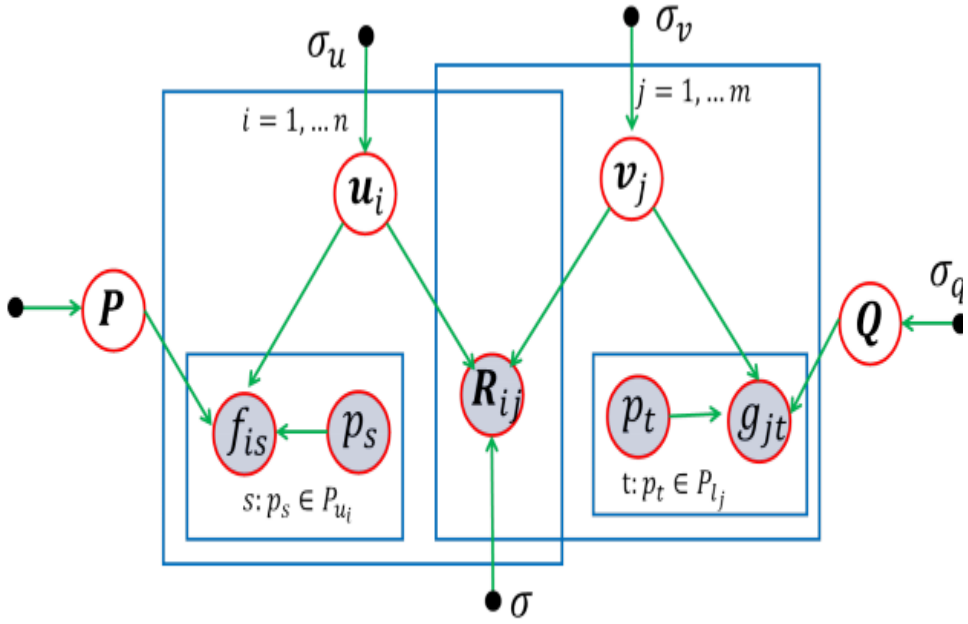


$$\widehat{x}_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u^T \gamma_i$$

$$\widehat{x}_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u^T \gamma_i + \theta_u^T \theta_i$$

$$\widehat{x}_{u,i} = \alpha + \beta_u + \beta_i + \gamma_u^T \gamma_i + \theta_u^T (\mathbf{E} f_i) + \beta'^T f_i$$

VPOI



$$\max_{\mathbf{U},\mathbf{V},\mathbf{P},\mathbf{Q},CNN} -\|\mathbf{Y}\odot(\mathbf{R}-\mathbf{U}^T\mathbf{V})\|_F^2-\lambda_1(\|\mathbf{U}\|_F^2+\|\mathbf{V}\|_F^2)$$

$$+\alpha\sum_{i=1}^{n}\sum_{p_k\in\mathcal{P}_{u_i}}\log P(f_{ik}=1|u_i,p_k)-\lambda_2\|\mathbf{P}\|_F^2$$

$$+\alpha\sum_{j=1}^{m}\sum_{p_k\in\mathcal{P}_{v_j}}\log P(g_{jk}=1|v_j,p_k)-\lambda_2\|\mathbf{Q}\|_F^2$$

Figure 3: A Graphical Representation of the Model

.

$$P(f_{is}=1|u_i,p_s)=\frac{\exp(\mathbf{u}_i^T\cdot\mathbf{P}\cdot CNN(p_s))}{\sum_{p_k\in\mathcal{P}}\exp(\mathbf{u}_i^T\cdot\mathbf{P}\cdot CNN(p_k))}\qquad P(g_{jt}=1|l_j,p_t)=\frac{\exp(\mathbf{v}_j^T\cdot\mathbf{Q}\cdot CNN(p_t))}{\sum_{p_k\in\mathcal{P}}\exp(\mathbf{v}_i^T\cdot\mathbf{Q}\cdot CNN(p_k))}$$

$$P(\mathcal{F},\mathcal{G}|\mathcal{P},\mathbf{U},\mathbf{V},\mathbf{P},\mathbf{Q})$$

$$=[\prod_{i=1}^{n}\prod_{p_s\in\mathcal{P}_{u_i}}P(f_{is}=1|u_i,p_s)]\cdot[\prod_{j=1}^{m}\prod_{p_t\in\mathcal{P}_{l_j}}P(g_{jt}=1|l_j,p_t)]$$

Best References

Classic Deep CNN Papers

- **Rethinking the inception architecture for computer vision** (2016), C. Szegedy et al.
- **Inception-v4, inception-resnet and the impact of residual connections on learning** (2016), C. Szegedy et al.
- **Identity Mappings in Deep Residual Networks** (2016), K. He et al.
- **Deep residual learning for image recognition** (2016), K. He et al.
- **Going deeper with convolutions** (2015), C. Szegedy et al.
- **Very deep convolutional networks for large-scale image recognition** (2014), K. Simonyan and A. Zisserman

https://github.com/terryum/awesome-deep-learning-papers

Classic Deep CNN Papers

- **Spatial pyramid pooling in deep convolutional networks for visual recognition** (2014), K. He et al.
- **Return of the devil in the details: delving deep into convolutional nets** (2014), K. Chatfield et al.
- **OverFeat: Integrated recognition, localization and detection using convolutional networks** (2013), P. Sermanet et al.
- **Maxout networks** (2013), I. Goodfellow et al.
- **Network in network** (2013), M. Lin et al.
- **ImageNet classification with deep convolutional neural networks** (2012), A. Krizhevsky et al.

https://github.com/terryum/awesome-deep-learning-papers

Latest Studies

- ICLR 2017
  - Incremental Network Quantization: Towards Lossless CNNs with Low-precision Weights
  - Incorporating long-range consistency in CNN-based texture generation
  - PixelCNN++: A PixelCNN Implementation with Discretized Logistic Mixture Likelihood and Other Modifications
  - Steerable CNNs
  - Trusting SVM for Piecewise Linear CNNs
  - Regularizing CNNs with Locally Constrained Decorrelations
  - Faster CNNs with Direct Sparse Convolutions and Guided Pruning

Latest Studies

- ICLR 2017
  - Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer
  - Pruning Filters for Efficient ConvNets
  - Do Deep Convolutional Nets Really Need to be Deep and Convolutional?
  - Pruning Convolutional Neural Networks for Resource Efficient Inference
  - FILTER SHAPING FOR CONVOLUTIONAL NEURAL NETWORKS
  - Batch Policy Gradient Methods for Improving Neural Conversation Models
  - Inductive Bias of Deep Convolutional Networks through Pooling Geometry
  - Semi-Supervised Classification with Graph Convolutional Networks

Latest Studies

- ICML 2017
  - ***Warped Convolutions: Efficient Invariance to Spatial Transformations***
  - ***Convexified Convolutional Neural Networks***
  - ***Warped Convolutions: Efficient Invariance to Spatial Transformations***
  - ***Deep Tensor Convolution on Multicores***
  - ***MEC: Memory-efficient Convolution for Deep Neural Network***
  - ***Dance Dance Convolution***
  - ***Language Modeling with Gated Convolutional Networks***
  - ***Convolutional Sequence to Sequence Learning***
  - ***Improved Variational Autoencoders for Text Modeling using Dilated Convolutions***
  - ***Accelerating Eulerian Fluid Simulation With Convolutional Networks***
  - ***PixelCNN Models with Auxiliary Variables for Natural Image Modeling***

Latest Studies

- NIPS 2017
  - **Gated Recurrent Convolution Neural Network for OCR**
  - **Towards Accurate Binary Convolutional Neural Network**
  - **Flat2Sphere: Learning Spherical Convolution for Fast Features from 360° Imagery**
  - **Introspective Classification with Convolutional Nets**
  - **MolecuLeNet: A continuous-filter convolutional neural network for modeling quantum interactions**
  - **Learning the Morphology of Brain Signals Using Alpha-Stable Convolutional Sparse Coding**
  - **Convolutional Gaussian Processes**
  - **Spherical convolutions and their application in molecular modelling**
  - **Sparse convolutional coding for neuronal assembly detection**
  - **Incorporating Side Information by Adaptive Convolution**
  - **Convolutional Phase Retrieval**
  - **Invariance and Stability of Deep Convolutional Representations**
  - **Protein Interface Prediction using Graph Convolutional Networks**

Q & A