ResNets, GANs, RNNs and LSTM

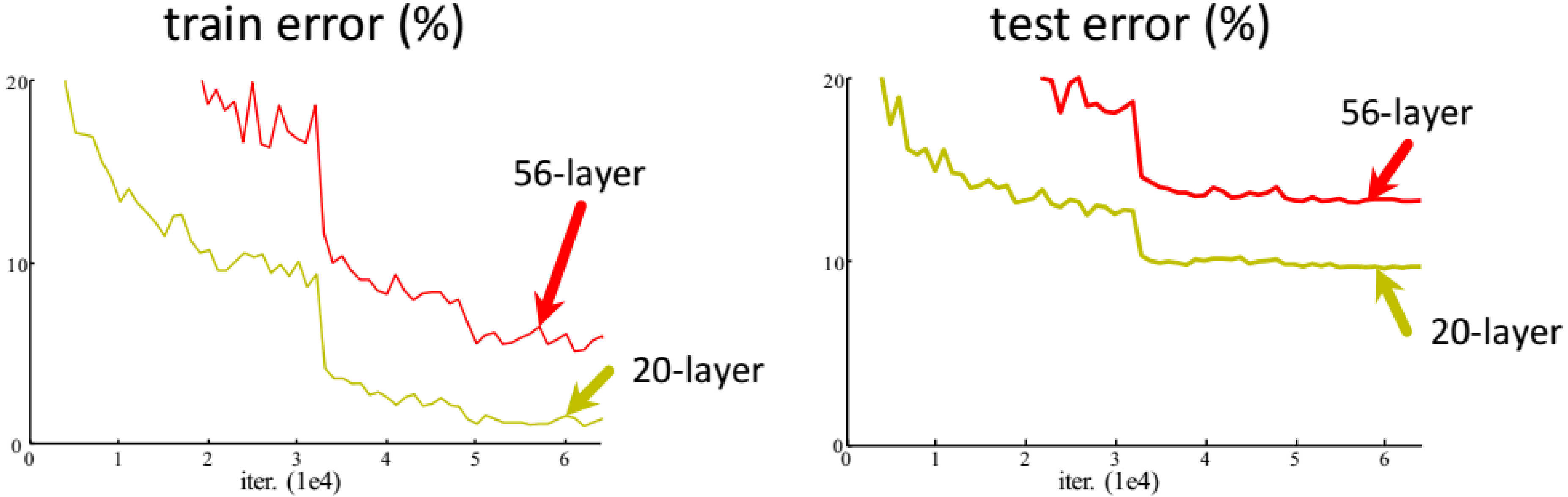
Dr. Fayyaz ul Amir Afsar Minhas
PIEAS Biomedical Informatics Research Lab
Department of Computer and Information Sciences
Pakistan Institute of Engineering & Applied Sciences
PO Nilore, Islamabad, Pakistan
http://faculty.pieas.edu.pk/fayyaz/

What’s wrong with Convolutional Neural Networks?

- Watch the talk by Geoff Hinton on the subject:
  - https://www.youtube.com/watch?v=rTawFwUvnLE
- Also watch: The failures of deep learning
  - https://www.youtube.com/watch?v=jWVZnkTfB3c
- Beyond DCNN
  - Gabor Convolutional Networks
    - https://arxiv.org/abs/1705.01450v2
  - Convolutional Sequence to Sequence Learning
    - https://arxiv.org/abs/1705.03122v2
  - Do Deep Convolutional Nets Really Need to be Deep and Convolutional
    - https://arxiv.org/abs/1603.05691v4
  - Picasso: A Neural Network Visualizer
    - https://arxiv.org/abs/1705.05627v1

Increasing Depth (10-100 Layers)

- What if we keep on stacking layers?
  - 56-layer net has **higher training error** and test error than 20-layer net

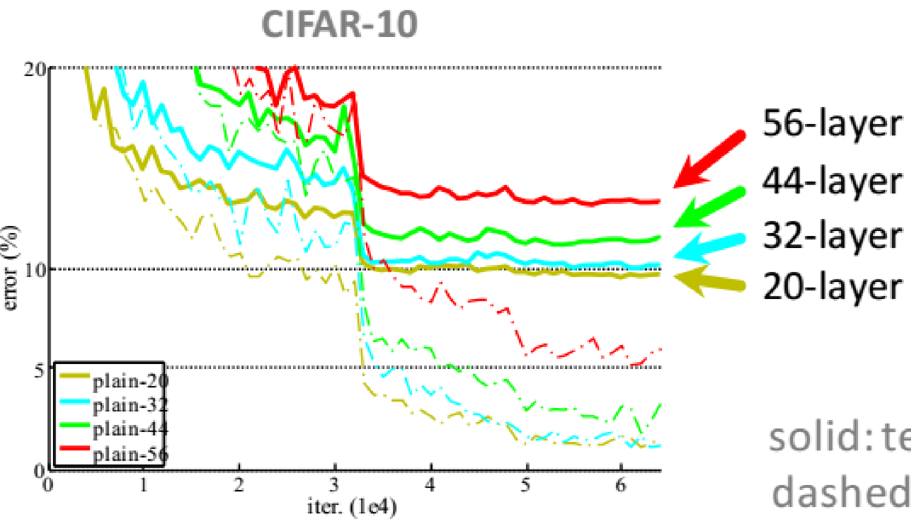


CIFAR-10

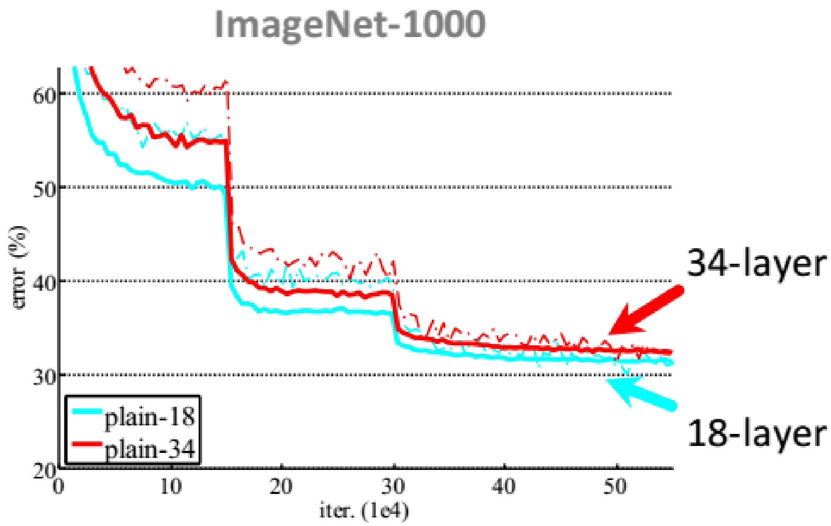
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016

Simply Stacking Layers?

- “Overly deep” plain nets have **higher training error**
- A general phenomenon, observed in many datasets
- Reasons
  - Optimization failure

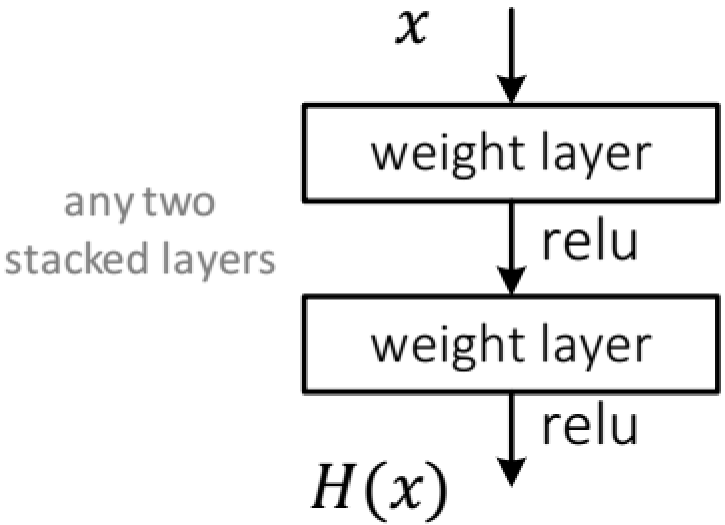

CIFAR-10

solid: test/val
dashed: train


ImageNet-1000

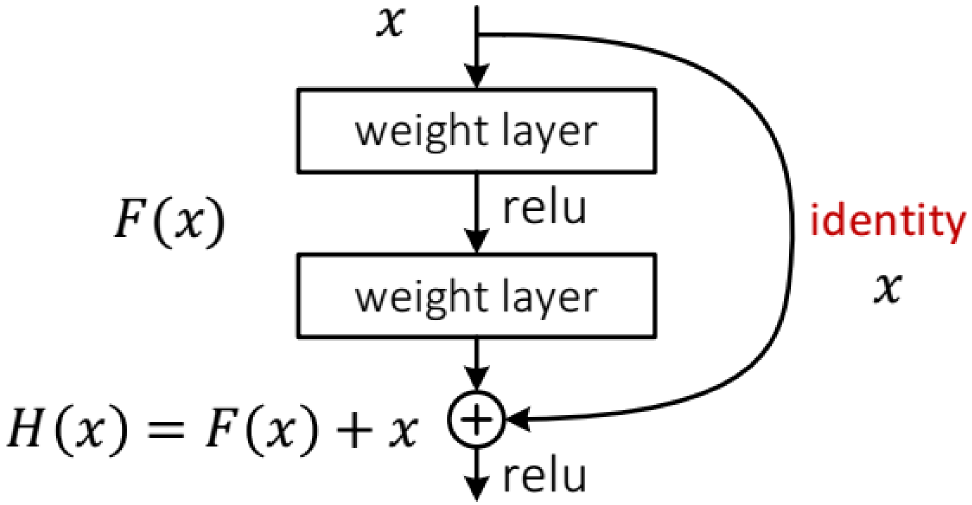
Residual Learning

Plain Network



H(x) is any desired mapping
Hope the 2 weight layers fit *H(x)*

Residual Network



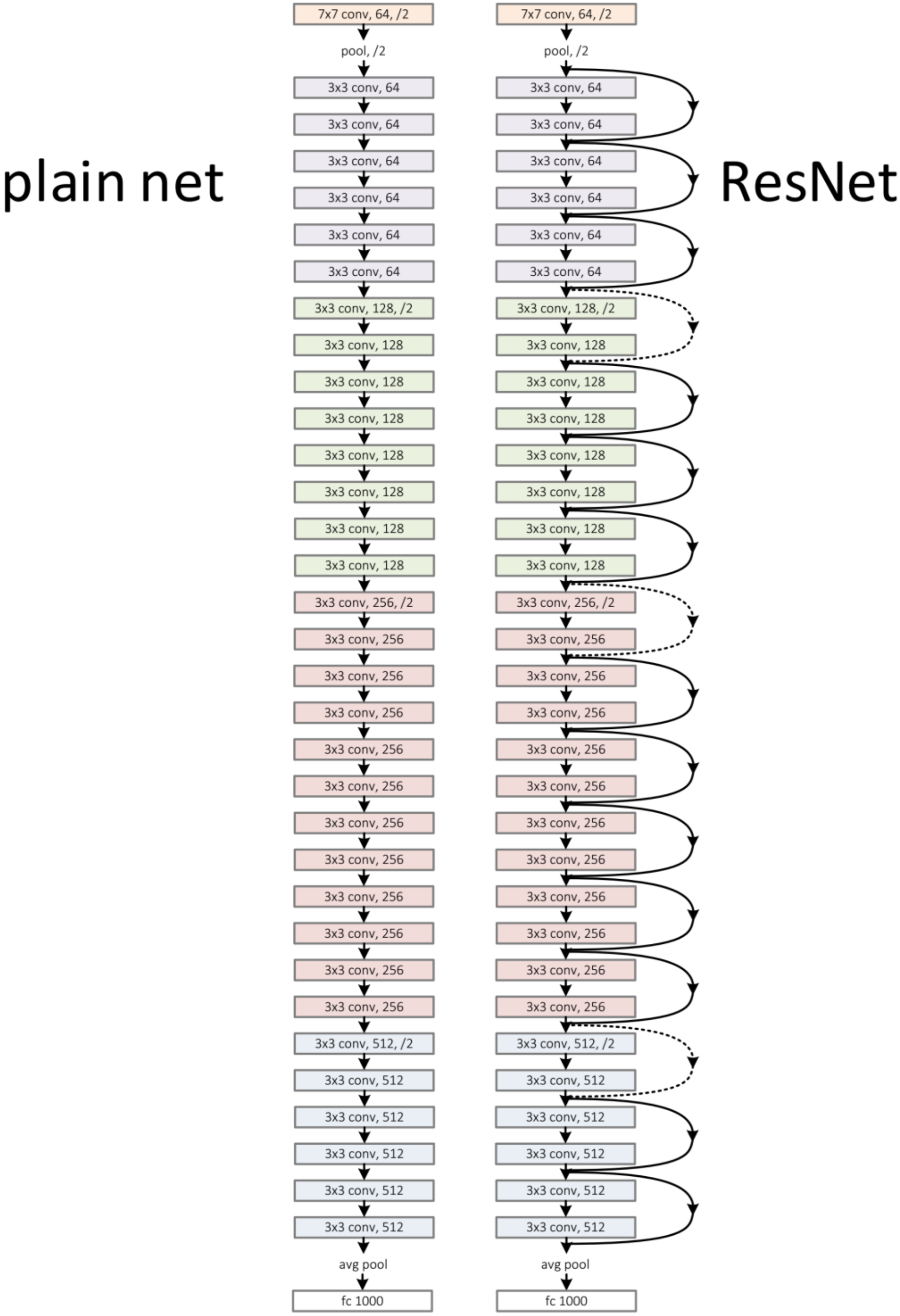
H(x) is any desired mapping
Hope the 2 weight layers fit *F(x)*

The network learns fluctuations *F(x)=H(x)-x*
Easier!

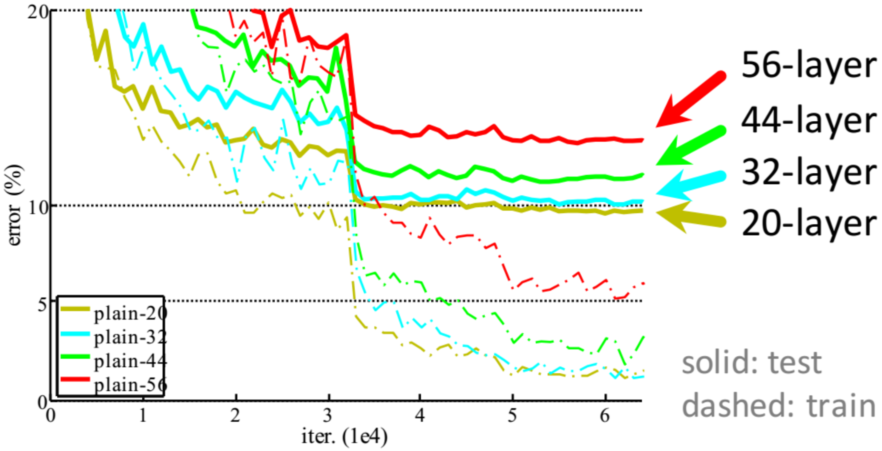
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

ResNet Models

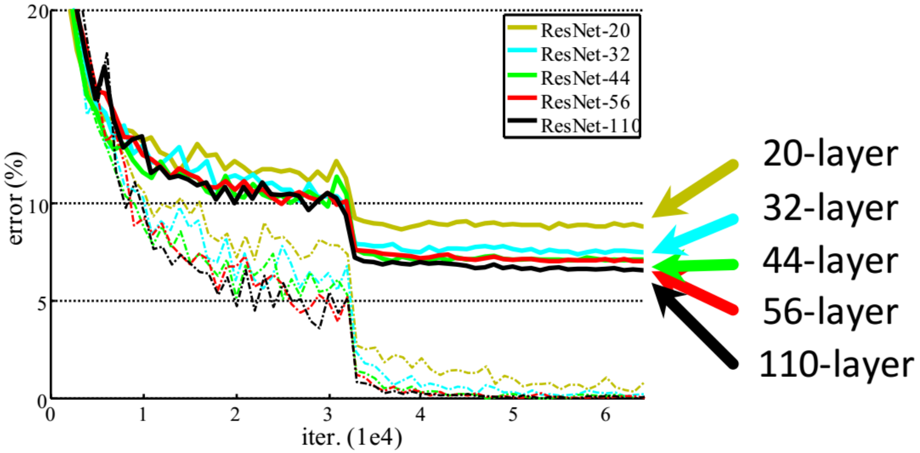
- No Dropout
- With Batch Normalization
- Use Data Augmentation



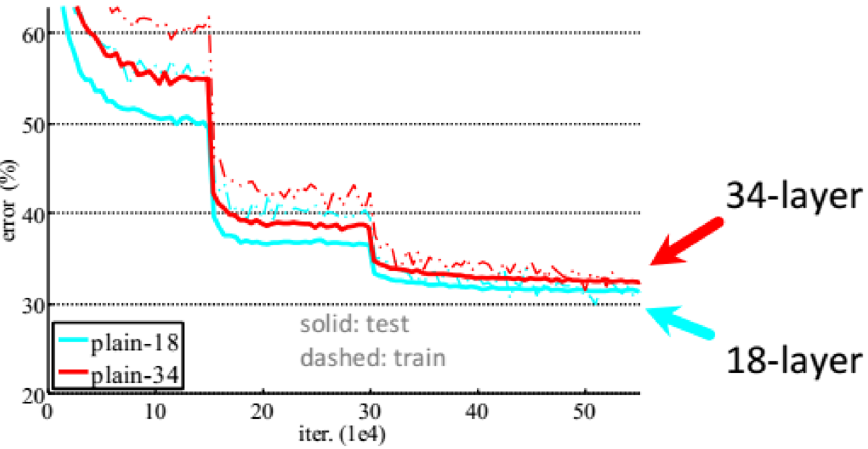
CIFAR-10 plain nets



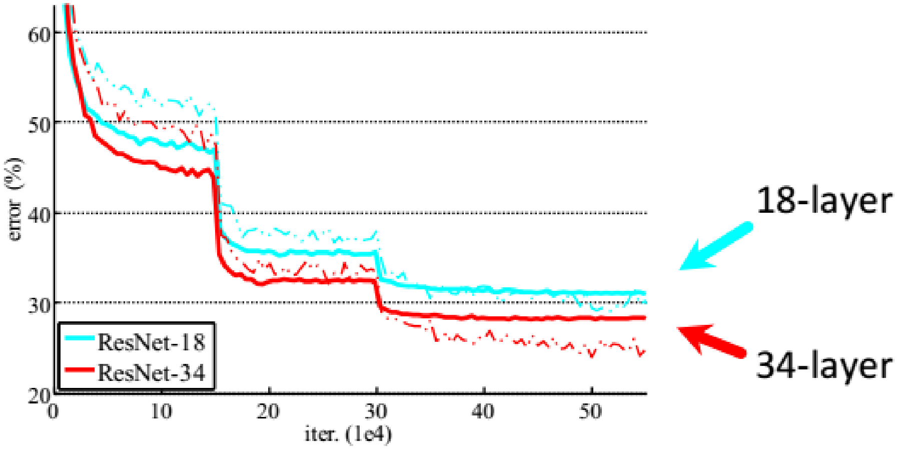
CIFAR-10 ResNets



ImageNet plain nets

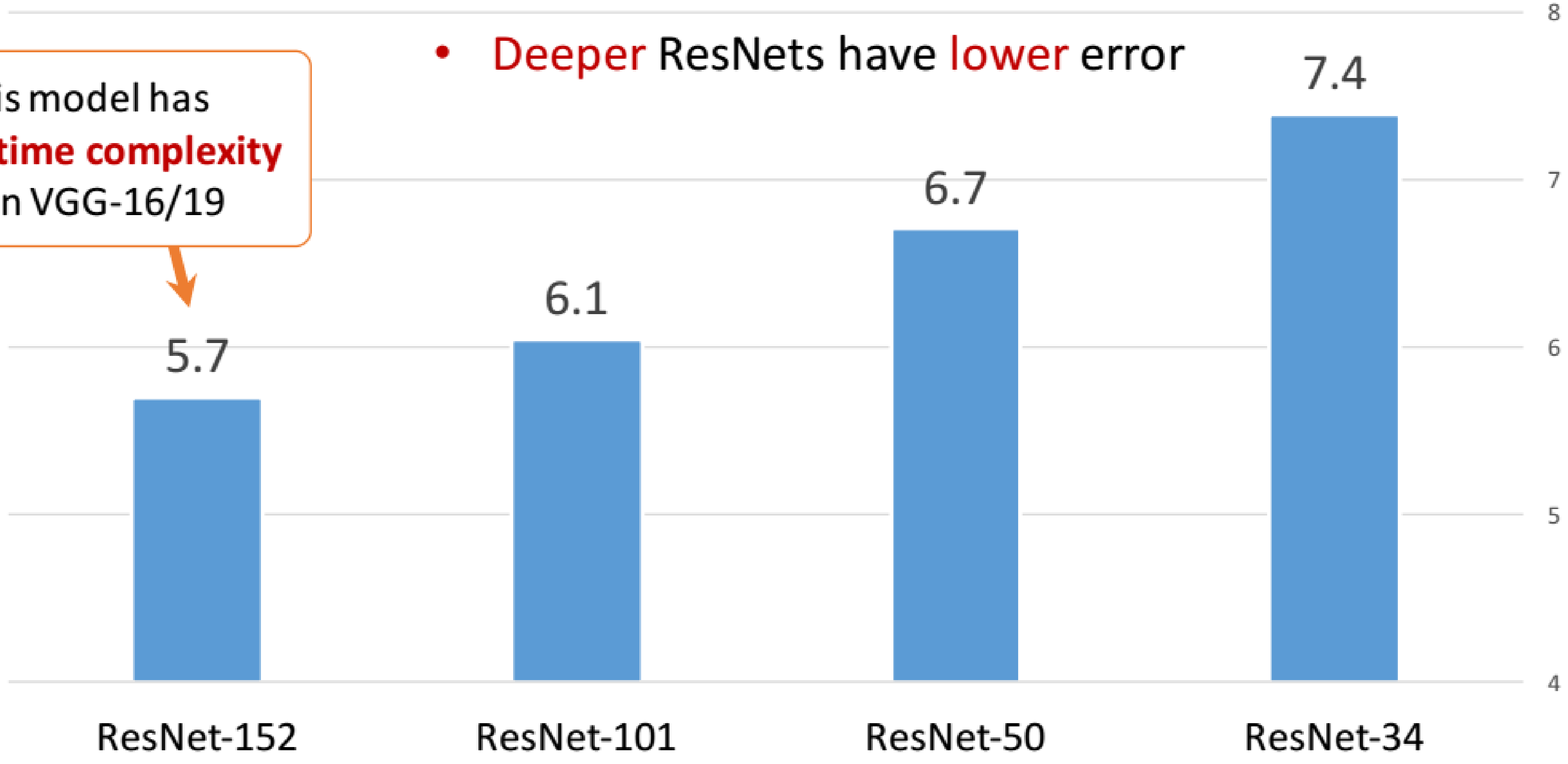


ImageNet ResNets



- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

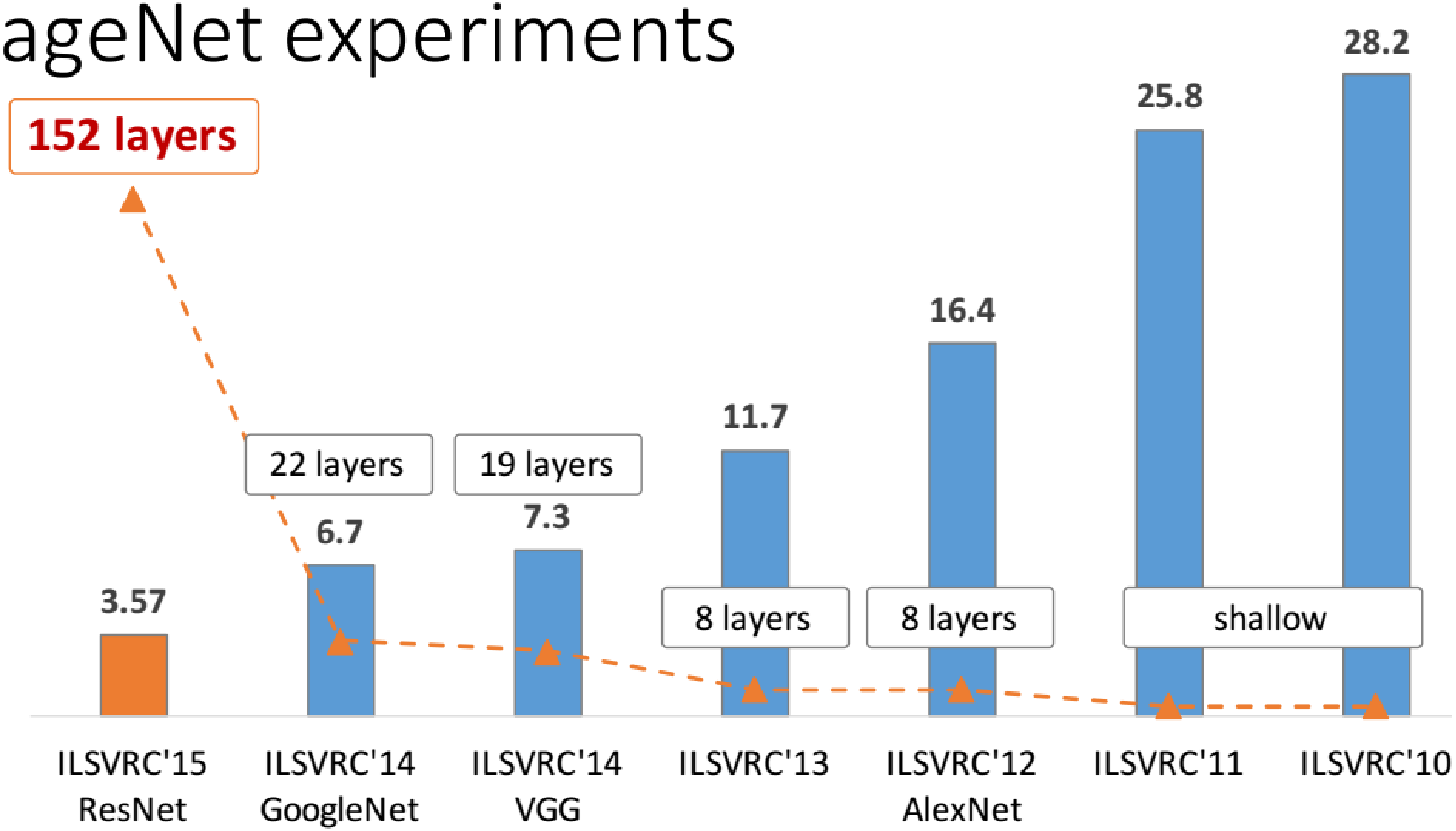
- Deeper ResNets have lower error



this model has **lower time complexity** than VGG-16/19

10-crop testing, top-5 val error (%)

ImageNet experiments



ImageNet Classification top-5 error (%)

ResNet Results

- **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

Residual Networks

- Required Reading
  - Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.
- Many third-party implementations
  - list in https://github.com/KaimingHe/deep-residual-networks
  - Torch ResNet: https://github.com/facebook/fb.resnet.torch

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.

Beyond ResNets

- Residual Networks Behave Like Ensembles of Relatively Shallow Networks
  - https://arxiv.org/abs/1605.06431v2
- Fractal Networks
  - https://arxiv.org/abs/1605.07648
- Deep Stochastic Networks
  - https://arxiv.org/abs/1603.09382

Moving towards Generative Models

- Uptil now our models have been discriminatory
  - Discriminate between classes
- Generative Models
  - Models that can be used to generate examples!

Generative Models

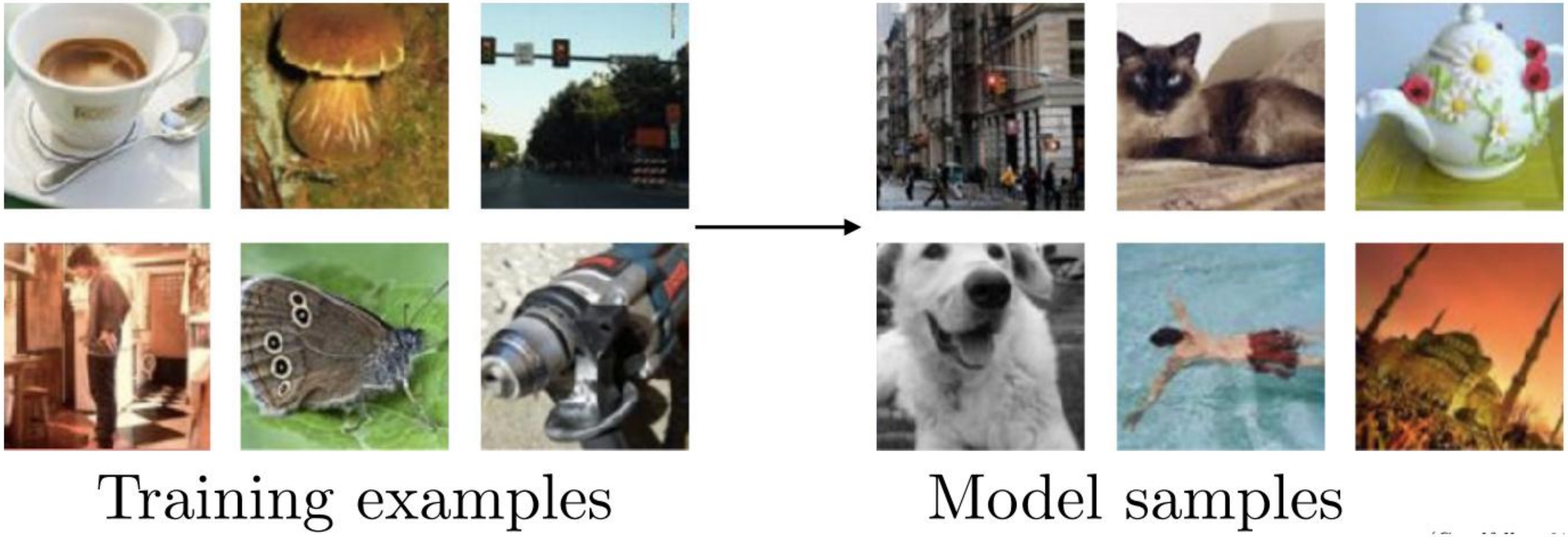
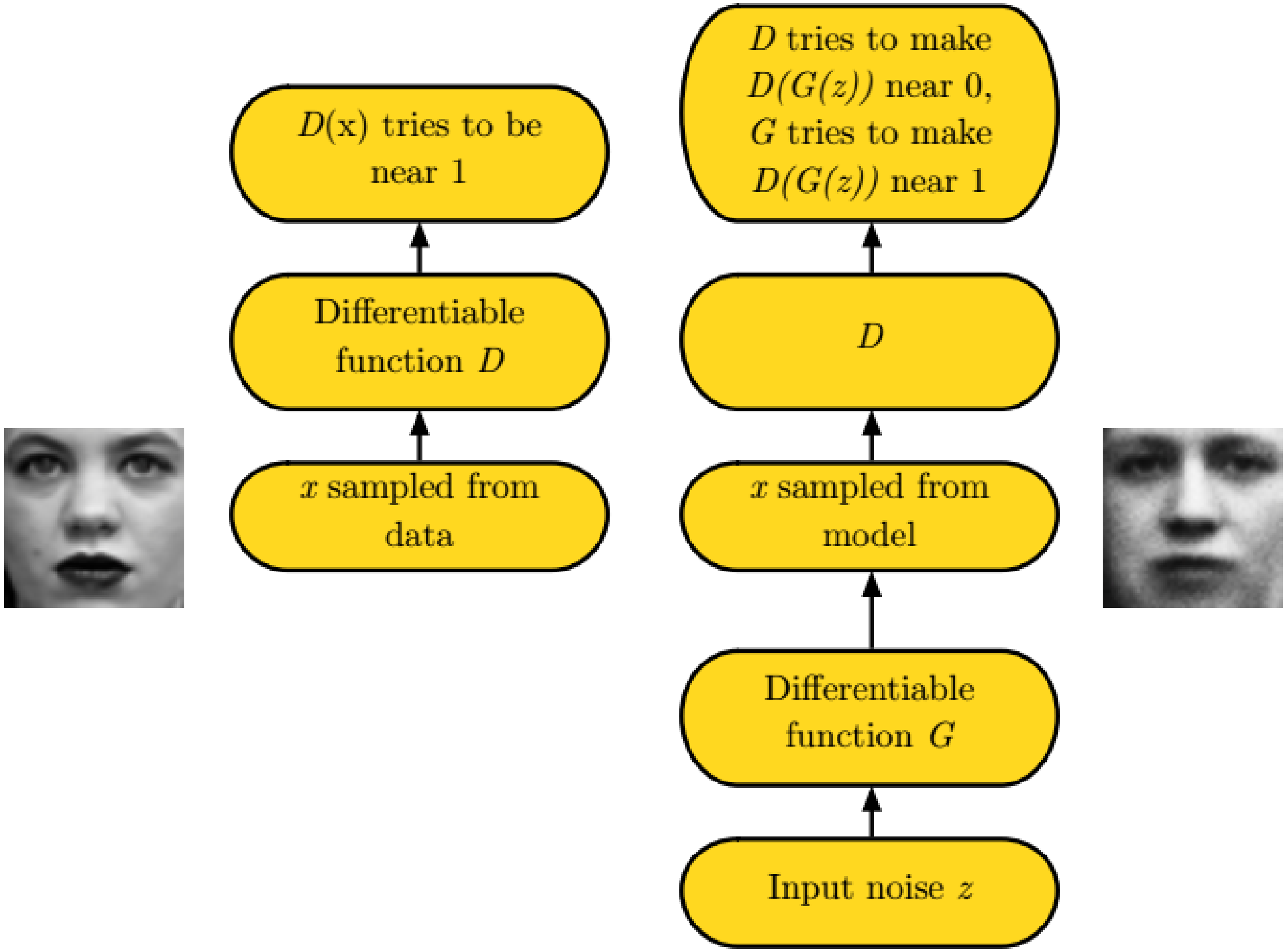


Figure 2: Some generative models are able to generate samples from the model distribution. In this illustration of the process, we show samples from the ImageNet (Deng et al., 2009, 2010; Russakovsky et al., 2014) dataset. An ideal generative model would be able to train on examples as shown on the left and then create more examples from the same distribution as shown on the right. At present, generative models are not yet advanced enough to do this correctly for ImageNet, so for demonstration purposes this figure uses actual ImageNet data to illustrate what an ideal generative model would produce.

NIPS 2016 Tutorial: Generative Adversarial Networks, Ian Goodfellow , https://arxiv.org/abs/1701.00160

Generative Adversarial Networks

- Also known as Turing Learning
- Unsupervisec realistic exam
- Consists of tv
  - Discrimina
    - Given a
    - If an ex dataset value o
    - If the e
  - Generator
    - Input: F
    - Output exampl
    - The dif of 0.0 f networ
    - The obj which D
    - Perforn



$D(\mathrm{x})$ tries to be near 1
Differentiable function D
x sampled from data
D tries to make $D(G(z))$ near 0, G tries to make $D(G(z))$ near 1
D
x sampled from model
Differentiable function G
Input noise z

https://arxiv.org/abs/1406.2661

GANs Applications: Super-resolution Imaging

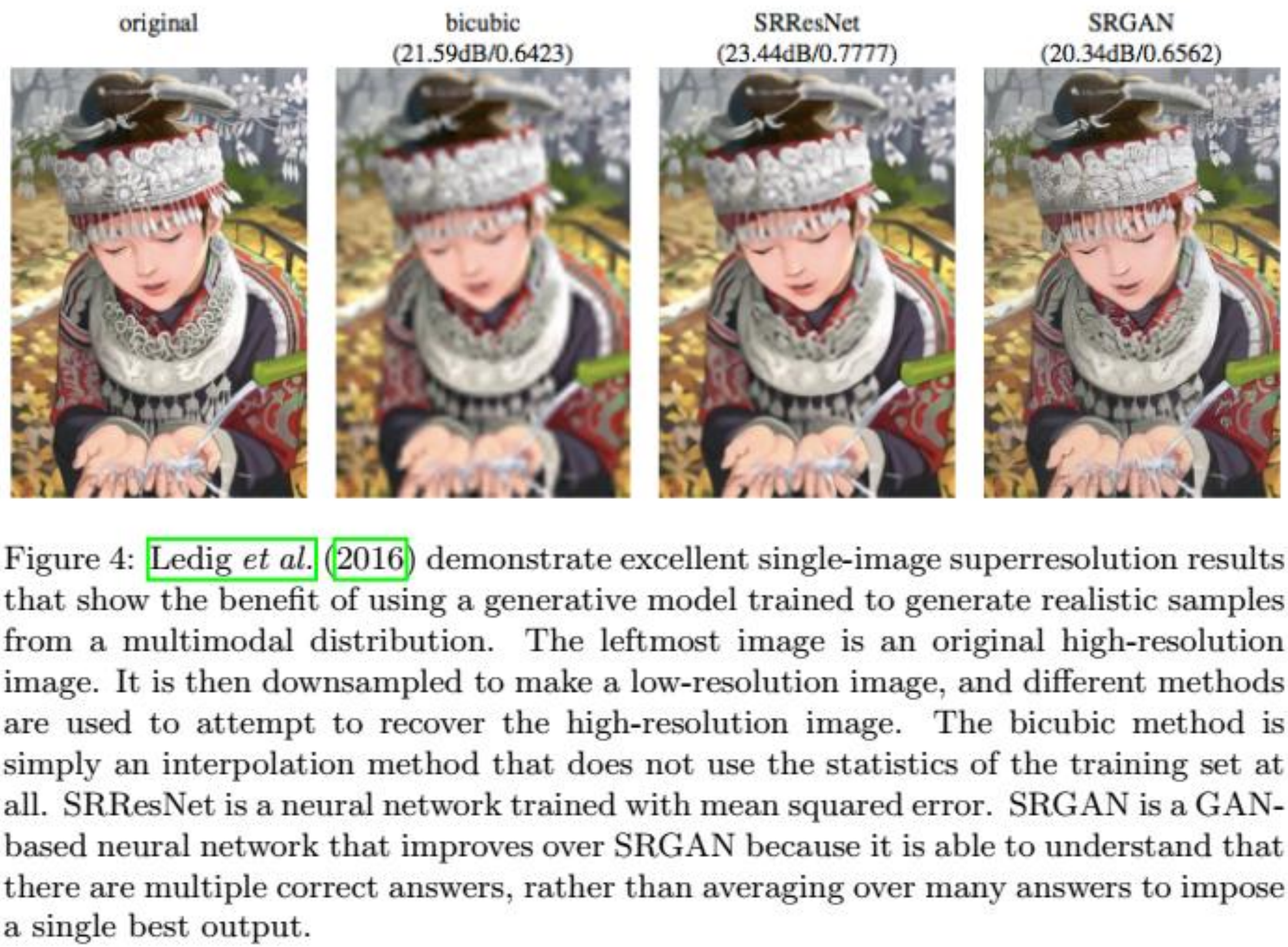


Figure 4: Ledig *et al.* (2016) demonstrate excellent single-image superresolution results that show the benefit of using a generative model trained to generate realistic samples from a multimodal distribution. The leftmost image is an original high-resolution image. It is then downsampled to make a low-resolution image, and different methods are used to attempt to recover the high-resolution image. The bicubic method is simply an interpolation method that does not use the statistics of the training set at all. SRResNet is a neural network trained with mean squared error. SRGAN is a GAN-based neural network that improves over SRGAN because it is able to understand that there are multiple correct answers, rather than averaging over many answers to impose a single best output.

GAN: Manipulation of images

- Interactive Image Generation, Modification and Warping
  - https://youtu.be/9c4z6YsBGQ0

Image to Image Translation

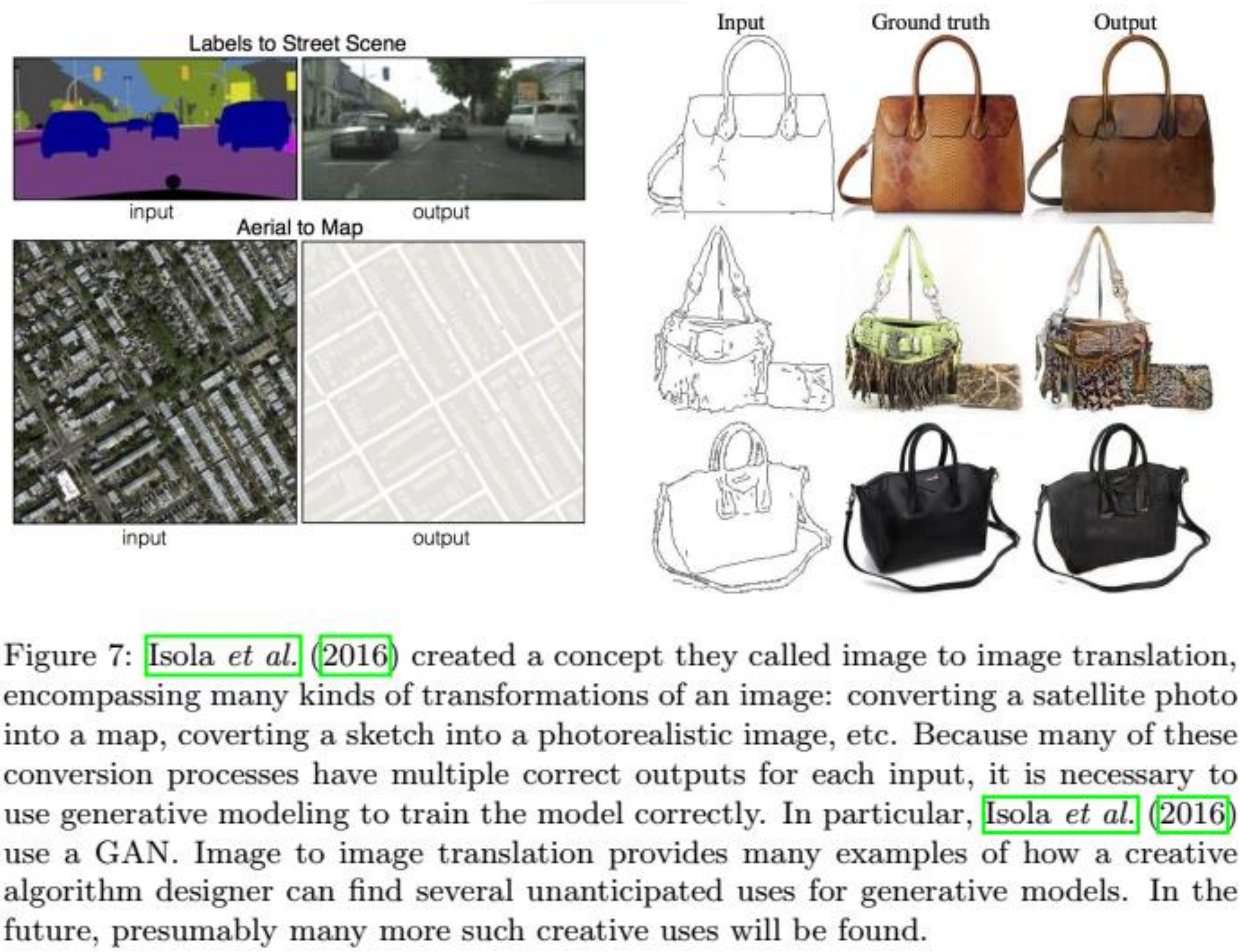
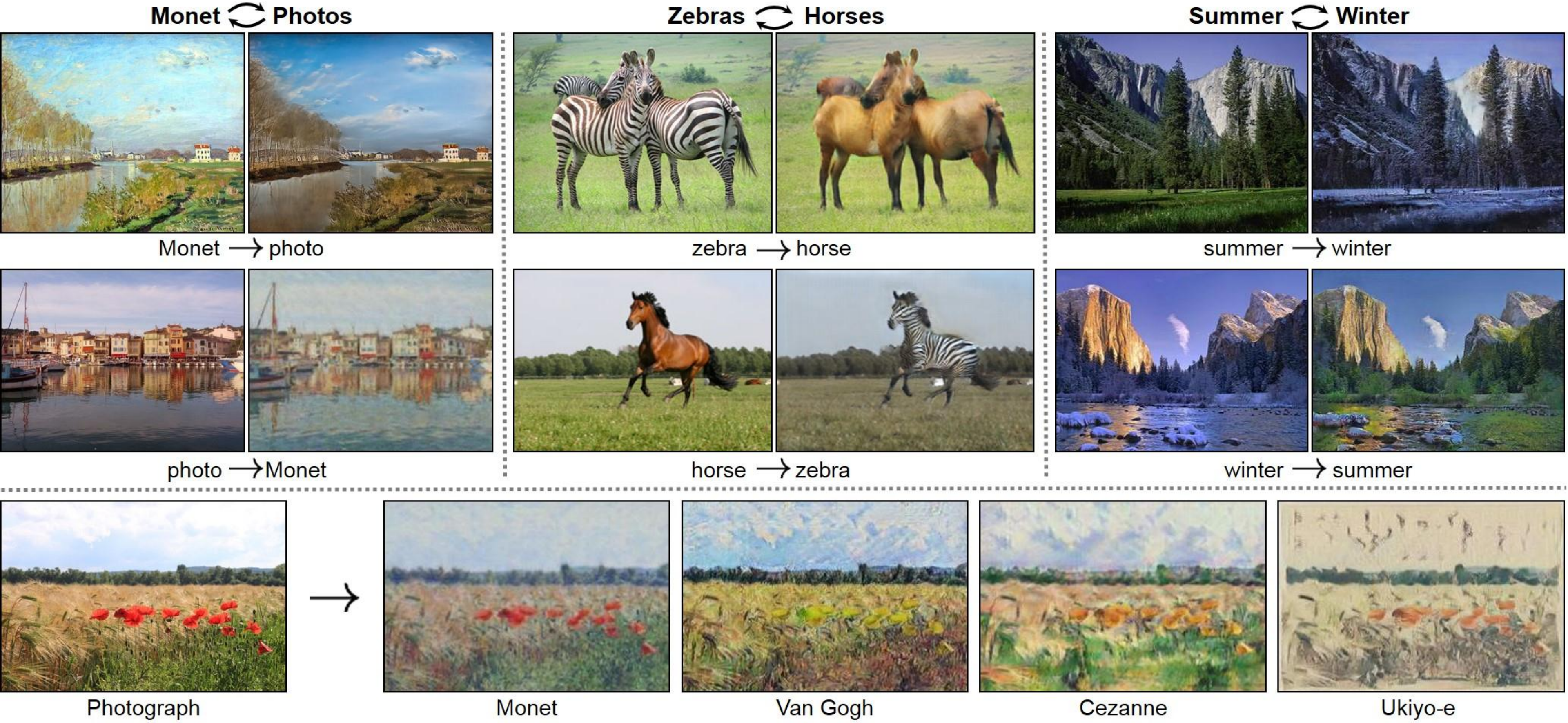


Figure 7: Isola *et al.* (2016) created a concept they called image to image translation, encompassing many kinds of transformations of an image: converting a satellite photo into a map, coverting a sketch into a photorealistic image, etc. Because many of these conversion processes have multiple correct outputs for each input, it is necessary to use generative modeling to train the model correctly. In particular, Isola *et al.* (2016) use a GAN. Image to image translation provides many examples of how a creative algorithm designer can find several unanticipated uses for generative models. In the future, presumably many more such creative uses will be found.

Neural Style Transfer

- Using CycleGAN
  - https://github.com/junyanz/CycleGAN

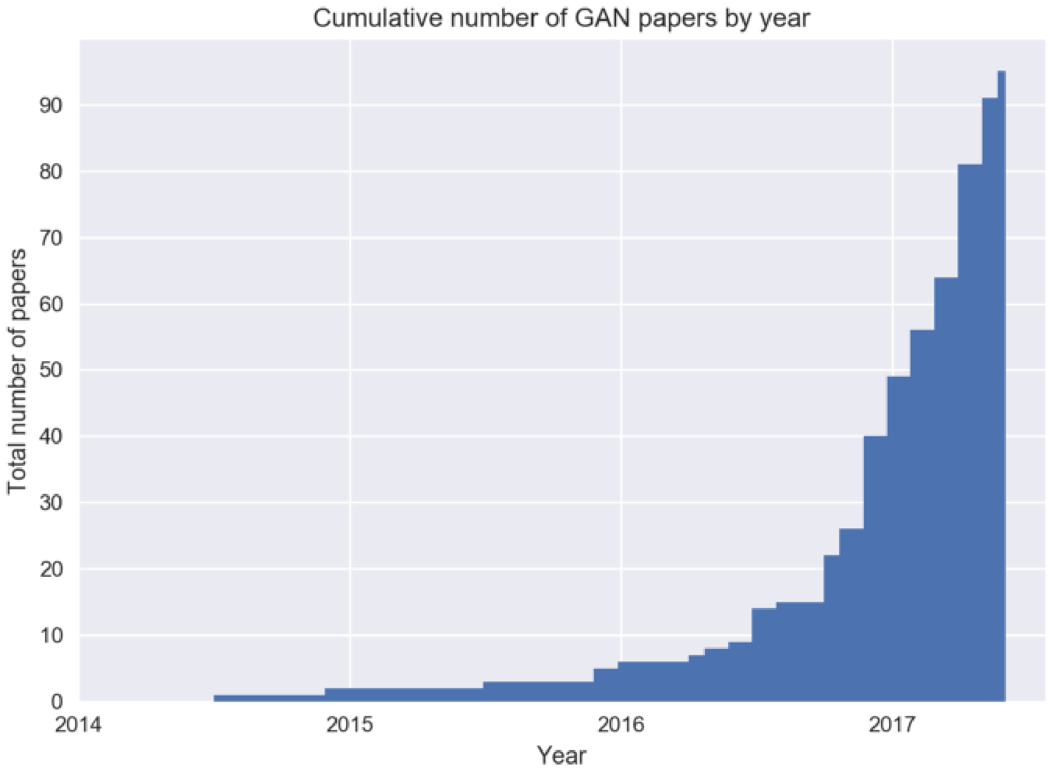


GAN Applications

- https://lyrebird.ai/demo
- Copy anyone’s voice!

GAN Zoo

- https://deephunt.in/the-gan-zoo-79597dc8c347

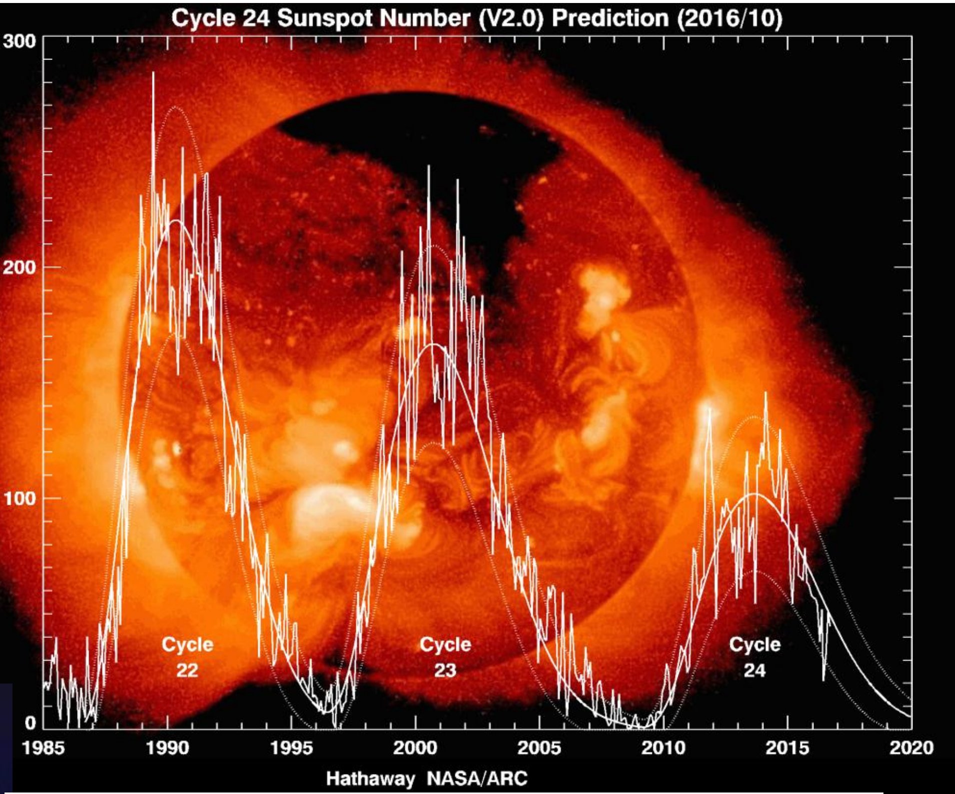


Applications

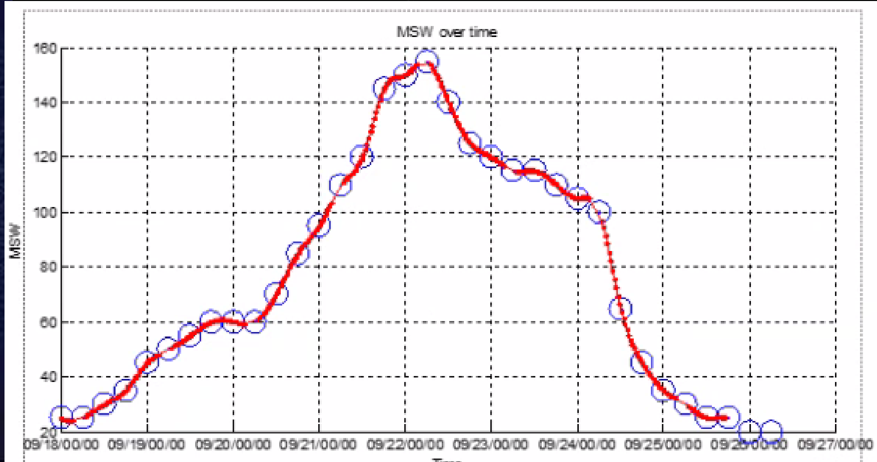
- 9 Cool Deep Learning Applications | Two Minute Papers
- https://youtu.be/Bui3DWs02h4?list=PLujxSBD-JXgnqDD1n-V30pKtp6Q886x7e
- https://www.youtube.com/watch?v=aKSILzbAqJs&index=65&list=PLujxSBD-JXgnqDD1n-V30pKtp6Q886x7e
- You said that? https://arxiv.org/abs/1705.02966v1

Predicting Temporal Data

- Predicting time series data
  - Number of sunspots
  - Hurricane intensity
- Mathematically,
- $h_t = f(\boldsymbol{x_1}, \boldsymbol{x_2}, \dots \boldsymbol{x_t}; h_1, h_2, \dots h_{t-1})$
  - f_t should approximate true values y_t for all times in the future

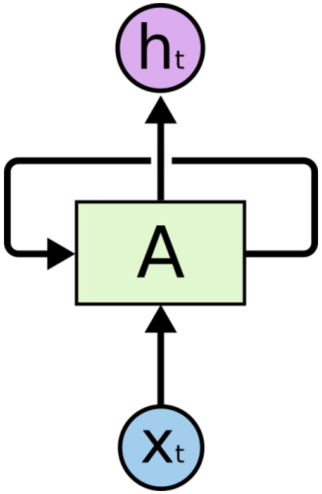


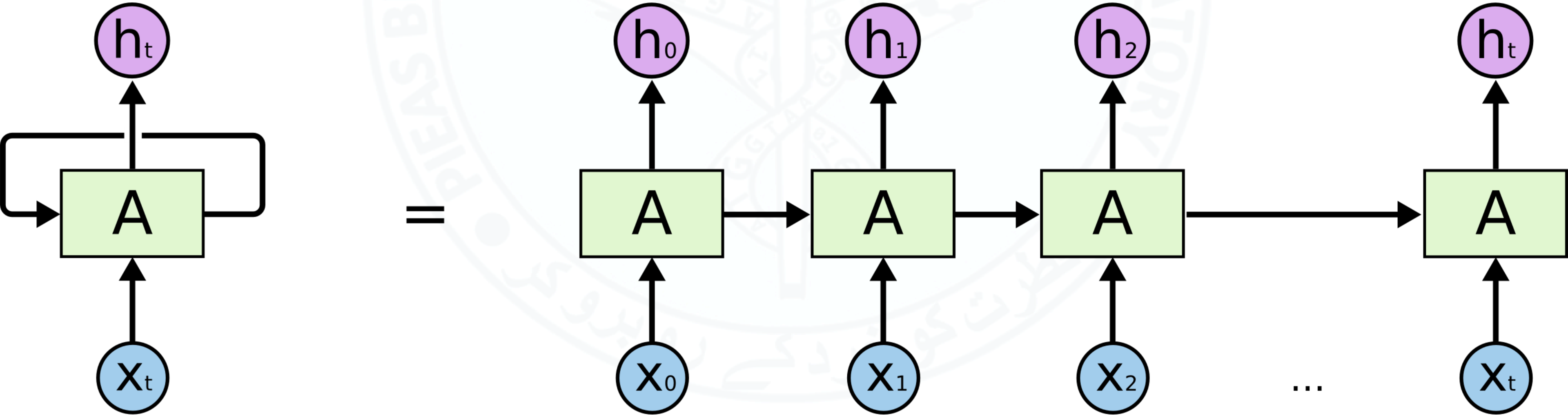
HURRICANE KATRINA
AUG 23-30, 2005
1,836 KILLED, 705 MISSING
DAMAGE: $90.1 BILLION (2010 USD)
WINDS TO 175 MPH
INTENSITY: 902 mb
HURRICANE
TROPICAL STORM
TROPICAL DEPRESSION
29th CAT 3 2nd Landfall
28th CAT 5
25th CAT 1 1st Landfall



Using Recurrent Networks

- Given the input $\mathbf{x}_t$ at time t and the previous outputs, predict the current output using a neural network A
- We can unroll the network and do "backpropagation through time"





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

"The Unreasonable Effectiveness of Recurrent Neural Networks." http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

Applications

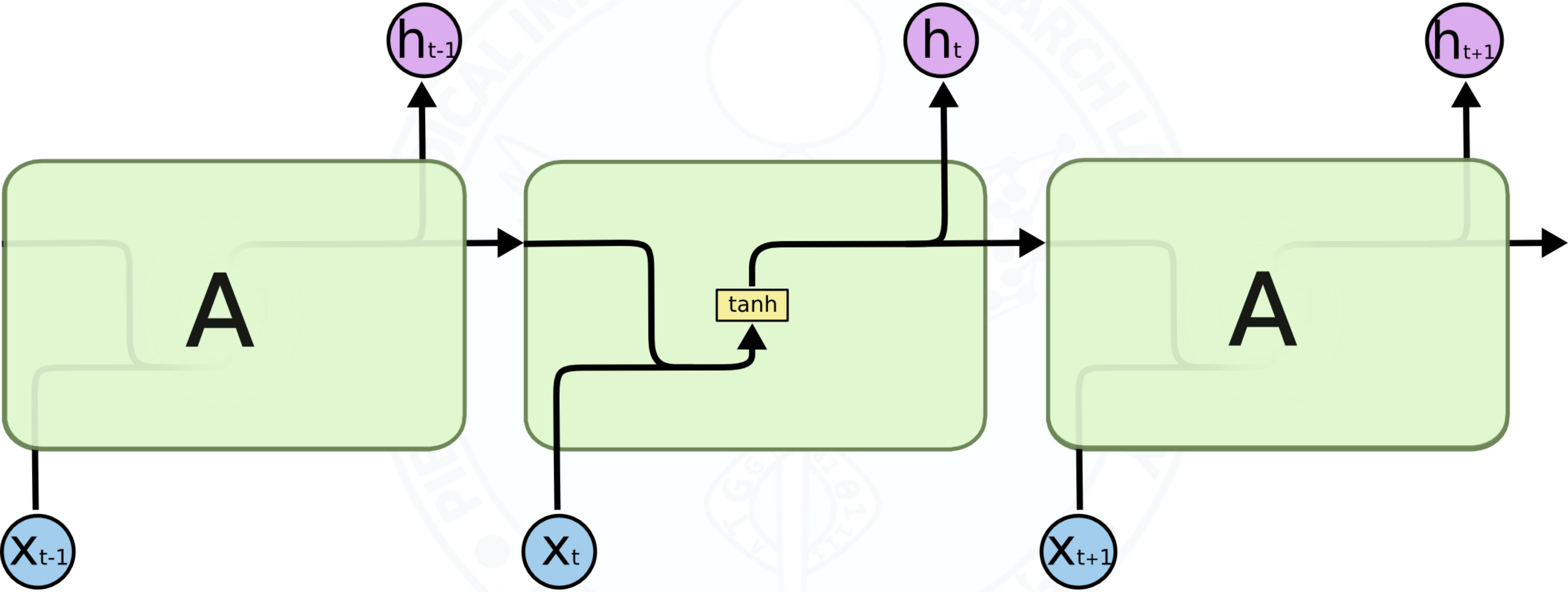
- Spellings
- Grammar
- Learning to write text
- Poetry
- Write code
- Predicting time series

Issues with RNNs

- Fill in the blanks:
  - The car is on the ______.
  - The clouds are in the _____.
    - The gap between relevant information and the place where it is needed is small – easy for RNN to learn
  - Bismillah and Adiba are doing their projects with Dr. Fayyaz Minhas. The are classmates. They sit together in the lab. The project reports are due tomorrow and must be submitted to the supervisor for review prior to final submission. Bismillah and Adiba will submit their reports to ________.
    - Irrelevant information
    - Gap between relevant information and the place where it is used is larger
    - RNNs will have difficulty here.
- Long term dependencies are an issue
  - Solution: Long-Short Term Memories (LSTM)

LSTM (Hochreiter & Schmidhuber 1997)

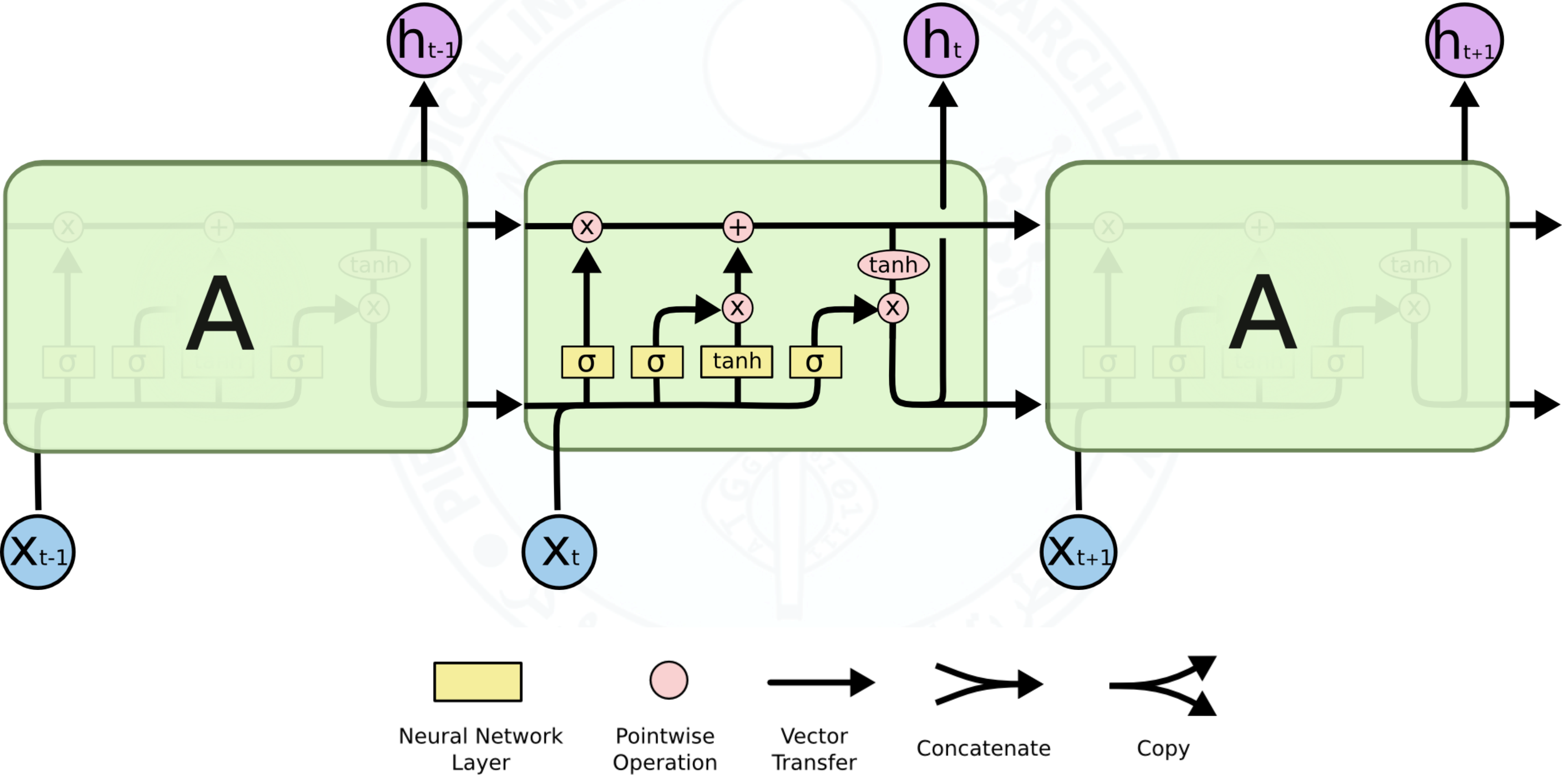
- RNNs can be represented as



The repeating module in a standard RNN contains a single layer.

$$h_t = tanh(\langle W, [h_{t-1}, x_t] \rangle)$$

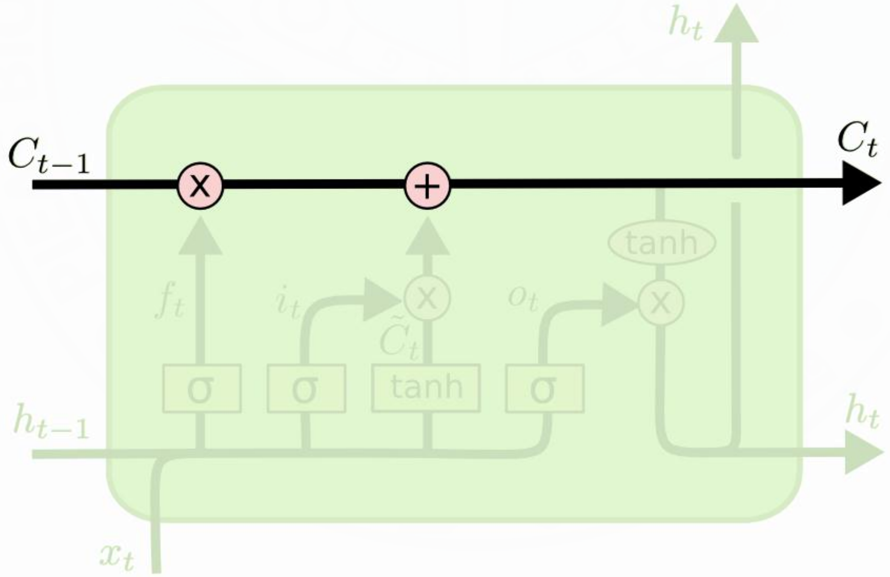
LSTM



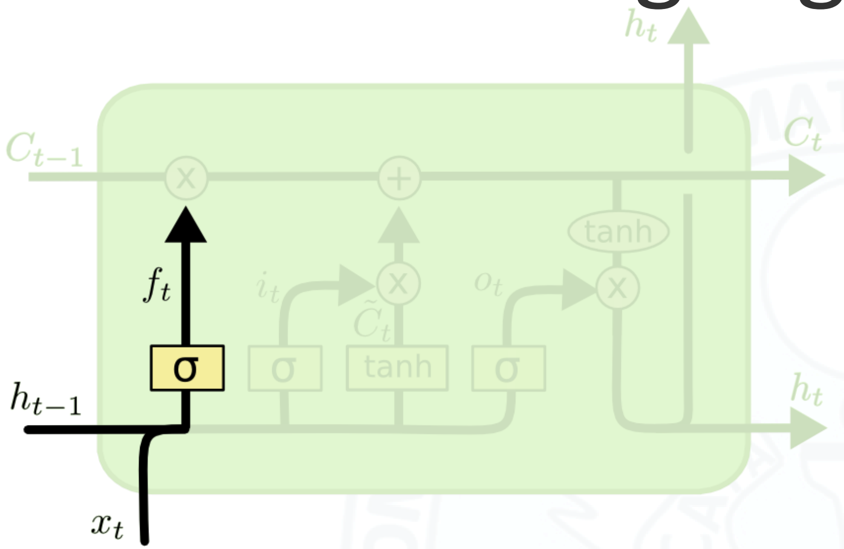
h_{t-1}
h_t
h_{t+1}
A
A
x
+
tanh
x
x
σ
σ
tanh
σ
X_{t-1}
X_t
X_{t+1}
Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy

LSTM: Cell State

- Each cell's output is dependent on its cell state which is "gated", i.e., information can be added or removed from the state



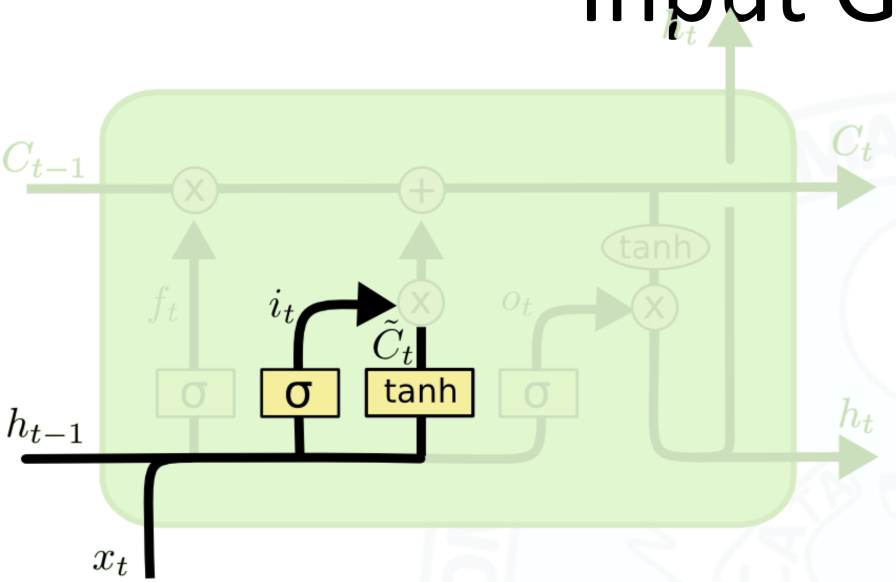
Forget gate layer



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

- It looks at ht-1 and xt and outputs a number between 0 and 1 which is multiplied with the cell state Ct-1 in an element-wise manner
  - In a language model trying to predict the next word based on all the previous ones. In such a problem, the cell state might include the gender of the present subject, so that the correct pronouns can be used. When we see a new subject, we want to forget the gender of the old subject.

Input Gate Layer

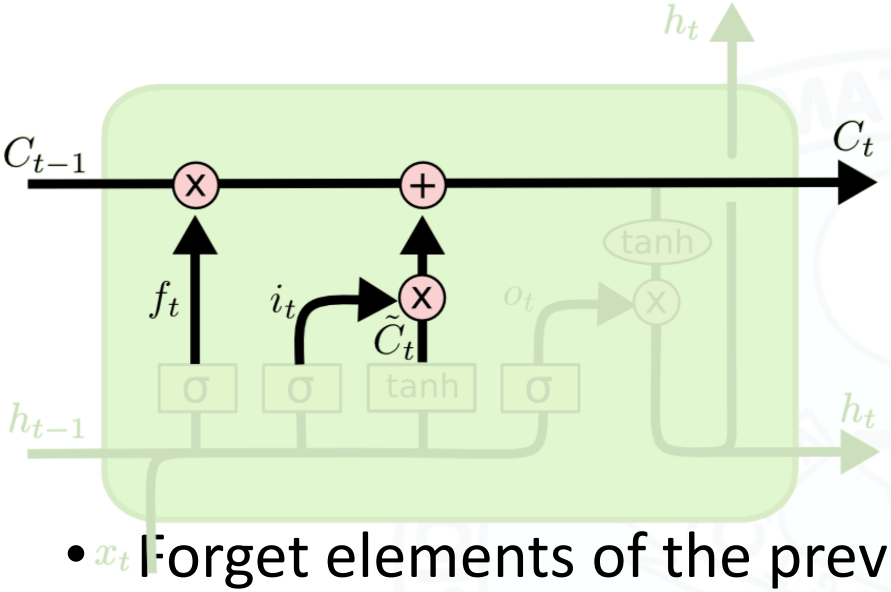


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

- The next step is to decide what new information we're going to store in the cell state. This has two parts.
  - First, a sigmoid layer called the "input gate layer" decides which values will be updated.
  - Next, a tanh layer creates a vector of new candidate values, $\widetilde{C}_t$, that could be added to the state. In the next step, we'll combine these two to create an update to the state.
- In the example of our language model, we'd want to add the gender of the new subject to the cell state, to replace the old one we're forgetting.

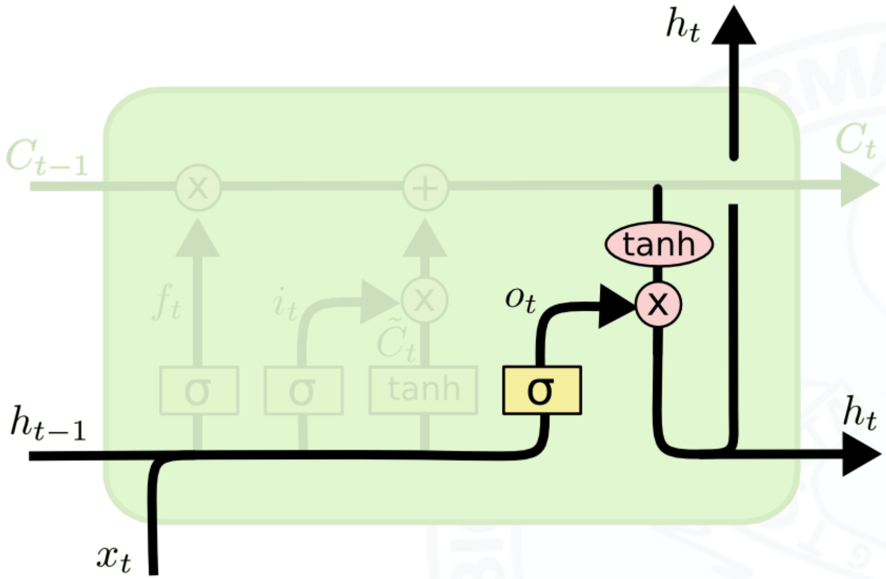
Cell State Update



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

- Forget elements of the previous cell state
- Create a tentative new cell state based on the current time cell, scale it by how much each element is to be updated and then add it to the gated previous cell state
- In the case of the language model, this is where we'd actually drop the information about the old subject's gender and add the new information, as we decided in the previous steps.

Generate Predictions



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

- Output is filtered version of the cell state

Applications

- Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras
  - http://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/
- Using MLP
  - http://machinelearningmastery.com/time-series-prediction-with-deep-learning-in-python-with-keras/
- Time Series Forecasting with the Long Short-Term Memory Network in Python
  - http://machinelearningmastery.com/time-series-forecasting-long-short-term-memory-network-python/

Applications

- **https://www.quora.com/What-are-the-various-applications-where-LSTM-networks-have-been-successfully-used**
- Language modeling (The tensorflow tutorial on PTB is a good place to start Recurrent Neural Networks) character and word level LSTM's are used
- Machine Translation also known as sequence to sequence learning (https://arxiv.org/pdf/1409.3215.pdf)
- Image captioning (with and without attention, https://arxiv.org/pdf/1411.4555v...)
- Hand writing generation (http://arxiv.org/pdf/1308.0850v5...)
- Image generation using attention models - my favorite (https://arxiv.org/pdf/1502.04623...)
- Question answering (http://www.aclweb.org/anthology/...)
- Video to text (https://arxiv.org/pdf/1505.00487...)

Transfer Learning

- http://sebastianruder.com/transfer-learning/
- Transfer Learning - Machine Learning's Next Frontier

Issues

- Deep Neural Networks are Easily Fooled
  - https://arxiv.org/abs/1412.1897v4
- Failures of deep learning
  - https://arxiv.org/abs/1703.07950
- Requires rethinking generalization
- Steps toward deep kernel methods from infinite neural networks
  - https://arxiv.org/abs/1508.05133
- Do Deep Neural Networks Really Need to be Deep?

The Future

- AutoML
  - DeepArchitect: Automatically Designing and Training Deep Architectures by Renato Negrinho, Geoff Gordon
    - https://github.com/negrinho/deep_architect
- Unsupervised Learning
  - GANs and GAN inspired models
  - Stopping GAN Violence with GUNs
    - https://arxiv.org/abs/1703.02528v1
  - Deep Stubborn Networks
    - http://www.kdnuggets.com/2017/04/deep-stubborn-networks-gan-refinement.html
  - Generative Ladder Networks
    - https://medium.com/towards-data-science/a-new-kind-of-deep-neural-networks-749bcde19108
- Applications of Deep Learning

End of Lecture-1

We want to make a machine that will be proud of us.

- Danny Hillis