Outline for today

- Recurrent neural nets (RNNs): example through DeepSpeech and LSTM
- Capsule networks
- Residual networks
- Very brief summary of the course

Recurrent Neural Nets: Deep Speech (Hannun et al. 14'[1])

Consider a data set $\mathcal{X} = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots\}$ with each $x_t^{(i)}$ for $t = 1, \ldots, T^{(i)}$ a time series (ordering matters). Let the input $h_t^{(0)}$ to the net be a local portion of the time series: $x_{t-C:t+C}^{(i)}$ with C chosen as, say 7. The Deep Speech net is then designed with three normal fully connected layers

$$h_t^{\ell} = \sigma(W^{(\ell)} h_t^{(\ell-1)} + b^{(\ell)})$$

with say $\sigma(z) = \max(0, z)$, followed by forward and backward recurrence layers

$$\begin{aligned} h_t^{(f)} &= \sigma(W^{(4)} h_t^{(3)} + W_r^{(f)} h_{t-1}^{(f)} + b^{(4)}) \\ h_t^{(b)} &= \sigma(W^{(4)} h_t^{(3)} + W_r^{(b)} h_{t+1}^{(f)} + b^{(4)}) \end{aligned}$$

Then a final fifth layer

$$h_t^{(5)} = \sigma(W^{(5)} h_t^{(4)} + b^{(5)})$$

where $h_t^{(4)} = h_t^{(f)} + h_t^{(b)}$ and a softmax applied to $h^{(5)}$.

[1] https://arxiv.org/pdf/1412.5567.pdf

Recurrent Neural Nets: Deep Speech (Hannun et al. 14'[2])

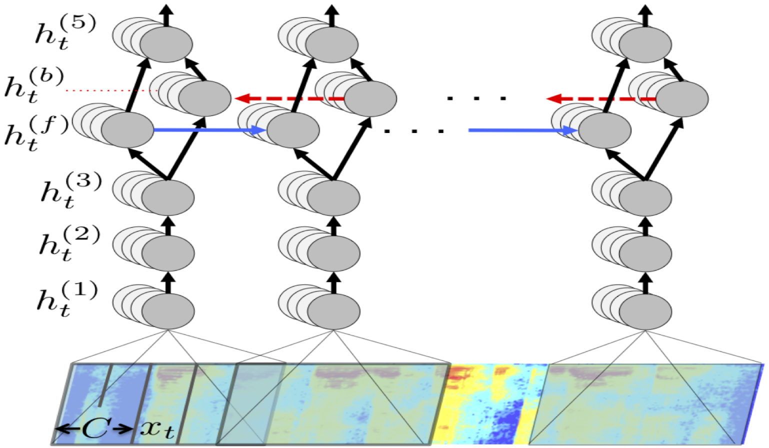
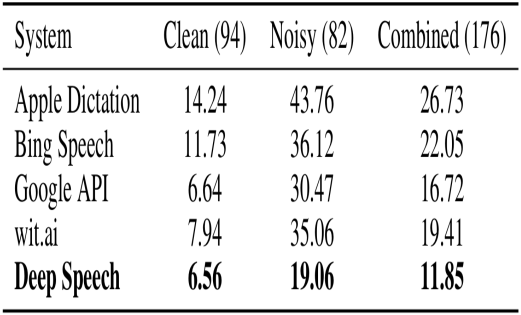


Figure 1: Structure of our RNN model and notation.

[2] https://arxiv.org/pdf/1412.5567.pdf

Recurrent Neural Nets: Deep Speech (Hannun et al. 14'[3])



System	Clean (94)	Noisy (82)	Combined (176)
Apple Dictation	14.24	43.76	26.73
Bing Speech	11.73	36.12	22.05
Google API	6.64	30.47	16.72
wit.ai	7.94	35.06	19.41
Deep Speech	**6.56**	**19.06**	**11.85**

Table 4: Results (%WER) for 5 systems evaluated on the original audio. Scores are reported *only* for utterances with predictions given by all systems. The number in parentheses next to each dataset, e.g. Clean (94), is the number of utterances scored.

[3] `https://arxiv.org/pdf/1412.5567.pdf`

LSTM with forget gates (Gers et al. 99'[4])

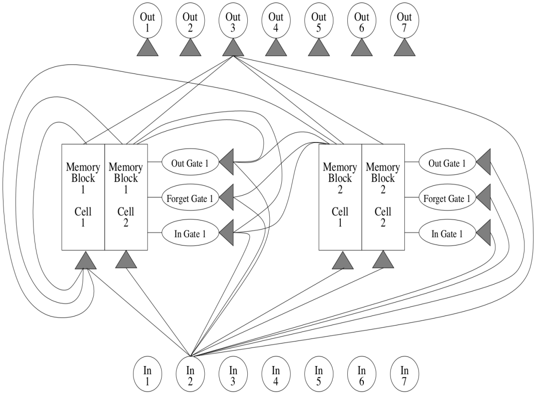


Figure 3: Three layer LSTM topology with recurrence limited to the hidden layer consisting of four extended LSTM memory blocks (only two shown) with two cells each. Only a limited subset of connections are shown.

[4] https://pdfs.semanticscholar.org/e10f/98b86797ebf6c8caea6f54cacbc5a50e8b34.pdf

LSTM with forget gates (Gers et al. 99'[5])

In more standard notation:

$$\begin{aligned}
\text{Forget: } f_t &= \sigma_f(W_f x_t + U_f h_{t-1} + b_f) \\
\text{Input: } i_t &= \sigma_i(W_i x_t + U_i h_{t-1} + b_i) \\
\text{Output: } o_t &= \sigma_o(W_o x_t + U_o h_{t-1} + b_o) \\
\text{Gate: } c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}$$

where $\circ$ is entrywise product and each layer uses both the initial input x_t along with a progressing hidden state with a variety of affine transformations.

[5] `https://pdfs.semanticscholar.org/e10f/98b86797ebf6c8caea6f54cacbc5a50e8b34.pdf`

Recurrent Neural Nets: bi-dir. LSTM (Fan et al. 15'[6])

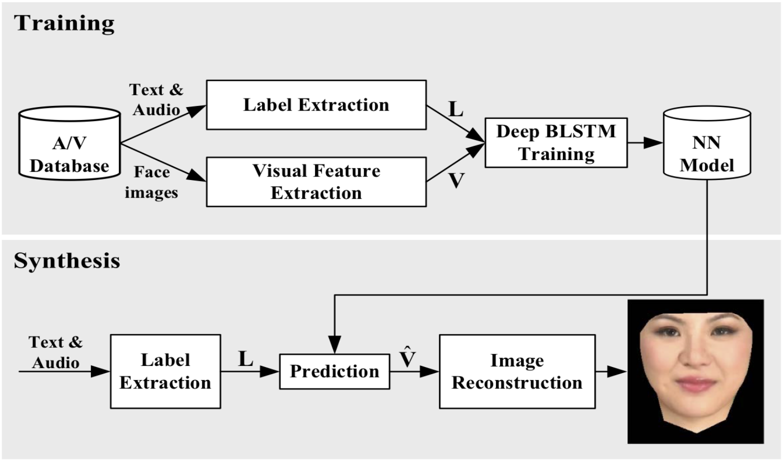


Fig. 1. System overview of the proposed talking head.

[6] https://www.microsoft.com/en-us/research/wp-content/uploads/2015/04/icassp2015_fanbo_1009.pdf

Recurrent Neural Nets: bi-dir. LSTM (Fan et al. 15'[7])

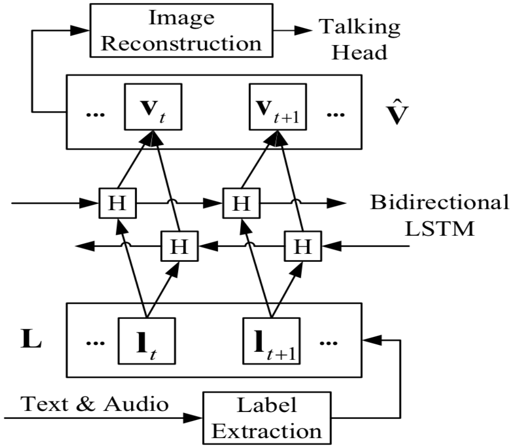
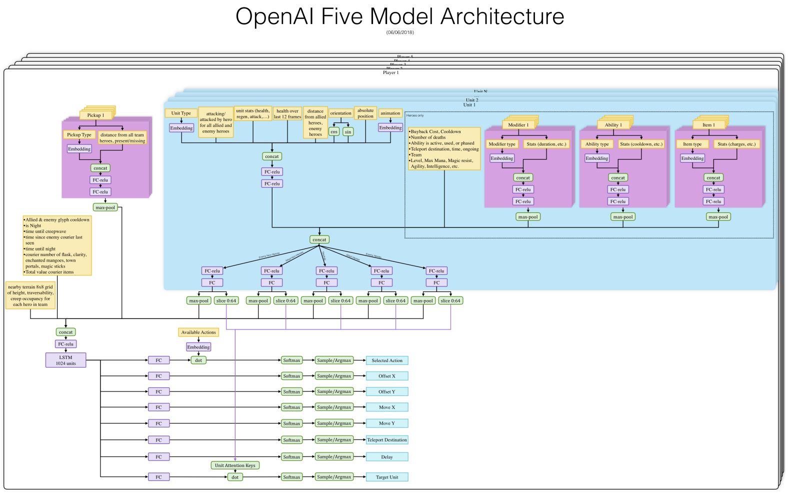


Fig. 3. BLSTM neural network in our talking head system.

[7] https://www.microsoft.com/en-us/research/wp-content/uploads/2015/04/icassp2015_fanbo_1009.pdf

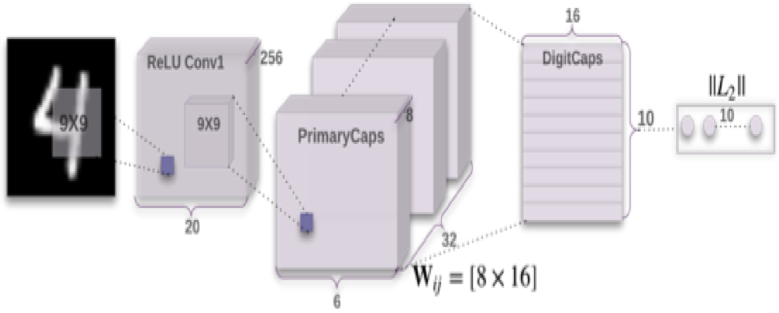
LSTM in Dota2 game arch. (OpenAI 18'[8])



[8] https://blog.openai.com/openai-five/

Capsules (Sabour 17'[9])

Figure 1: A simple CapsNet with 3 layers. This model gives comparable results to deep convolutional networks (such as Chang and Chen [2015]). The length of the activity vector of each capsule in DigitCaps layer indicates presence of an instance of each class and is used to calculate the classification loss. $\mathbf{W}_{ij}$ is a weight matrix between each $\mathbf{u}_i, i \in (1, 32 \times 6 \times 6)$ in PrimaryCapsules and $\mathbf{v}_j, j \in (1, 10)$.



Note, this concept is much older, see "credibility networks" by Hinton, Ghahramani, and Teh.

[9] https://arxiv.org/pdf/1710.09829.pdf

Capsules (Sabour 17'[10])

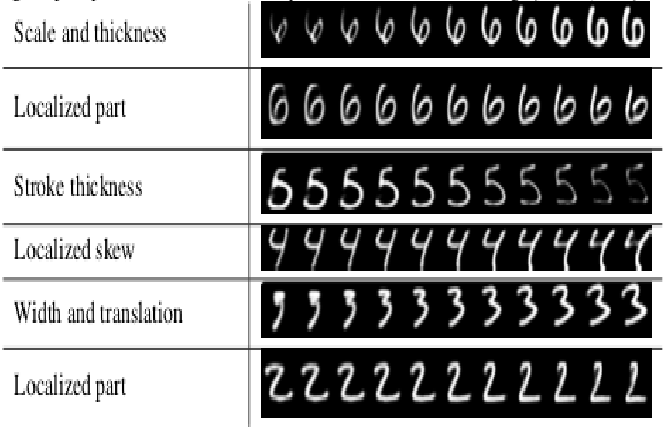
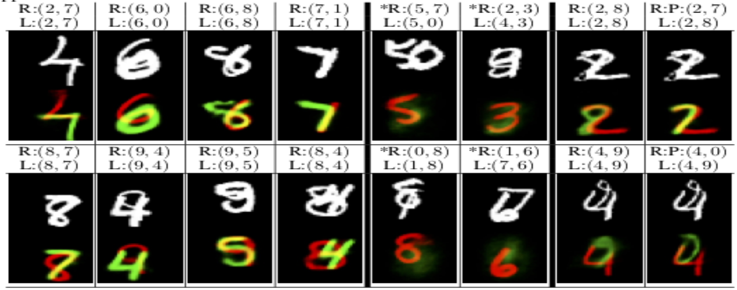


Figure 4: Dimension perturbations. Each row shows the reconstruction when one of the 16 dimensions in the DigitCaps representation is tweaked by intervals of 0.05 in the range $[-0.25, 0.25]$.

[10] https://arxiv.org/pdf/1710.09829.pdf

Capsules (Sabour 17'[11])

Figure 5: Sample reconstructions of a CapsNet with 3 routing iterations on MultiMNIST test dataset. The two reconstructed digits are overlayed in green and red as the lower image. The upper image shows the input image. L:(l_1, l_2) represents the label for the two digits in the image and R:(r_1, r_2) represents the two digits used for reconstruction. The two right most columns show two examples with wrong classification reconstructed from the label and from the prediction (P). In the $(2, 8)$ example the model confuses 8 with a 7 and in $(4, 9)$ it confuses 9 with 0. The other columns have correct classifications and show that the model accounts for all the pixels while being able to assign one pixel to two digits in extremely difficult scenarios (column $1 - 4$). Note that in dataset generation the pixel values are clipped at 1. The two columns with the (*) mark show reconstructions from a digit that is neither the label nor the prediction. These columns suggests that the model is not just finding the best fit for all the digits in the image including the ones that do not exist. Therefore in case of $(5, 0)$ it cannot reconstruct a 7 because it knows that there is a 5 and 0 that fit best and account for all the pixels. Also, in case of $(8, 1)$ the loop of 8 has not triggered 0 because it is already accounted for by 8. Therefore it will not assign one pixel to two digits if one of them does not have any other support.



[11] https://arxiv.org/pdf/1710.09829.pdf

Residual Networks (He 15'[12])

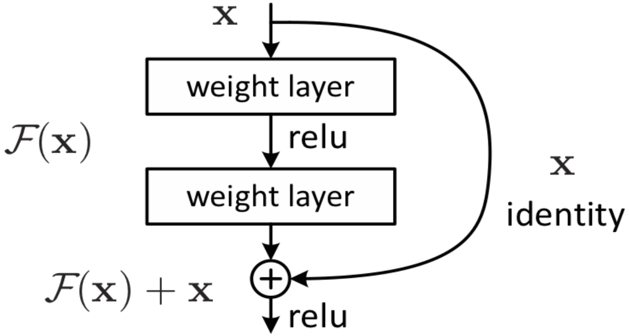
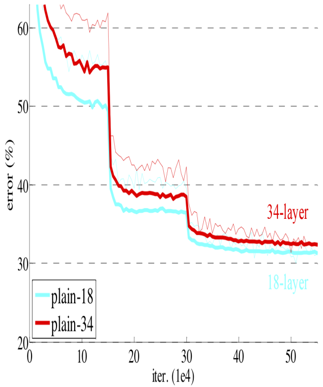


Figure 2. Residual learning: a building block.

If the block is attempting to learn a map $\mathcal{H}(x)$ the ResNet instead attempts to learn $\mathcal{F}(x) := \mathcal{H}(x) - x$ which is the residual. One can speculate this is easier to learn if $H(x)$ is approximately an identity map.

[12] `https://arxiv.org/pdf/1512.03385.pdf`

Residual Networks: training vs plain (He 15'[13])



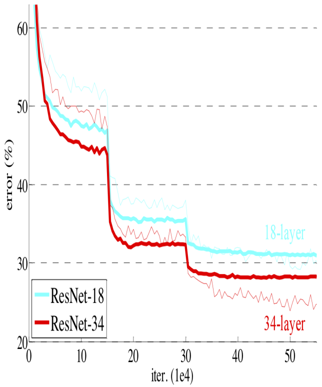
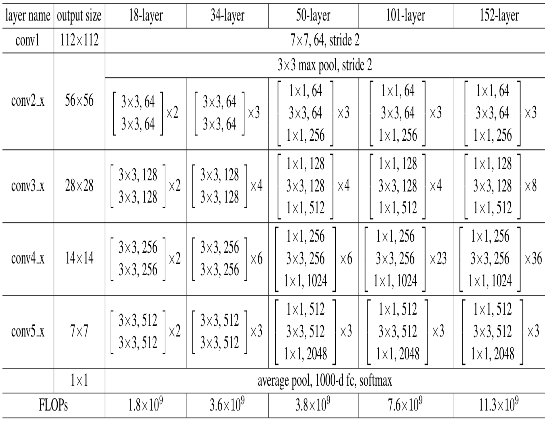


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

[13] `https://arxiv.org/pdf/1512.03385.pdf`

Residual Networks: training vs plain (He 15'[14])



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Table 1. Architectures for ImageNet. Building blocks are shown in brackets (see also Fig. 5), with the numbers of blocks stacked. Down-sampling is performed by conv3_1, conv4_1, and conv5_1 with a stride of 2.

[14] `https://arxiv.org/pdf/1512.03385.pdf`

Residual Networks (He 15'[15])

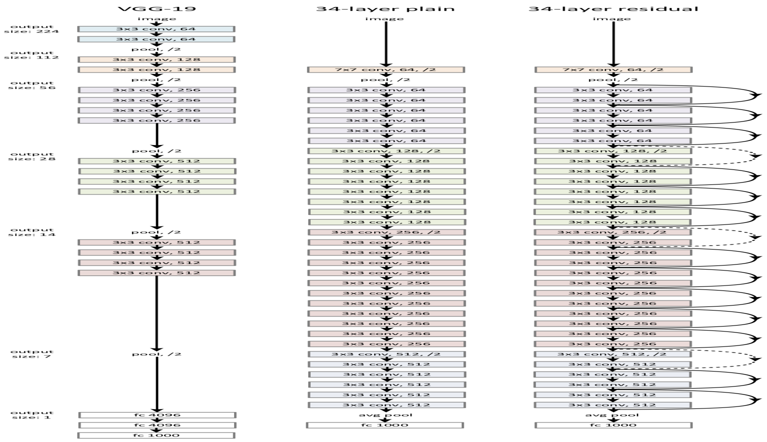
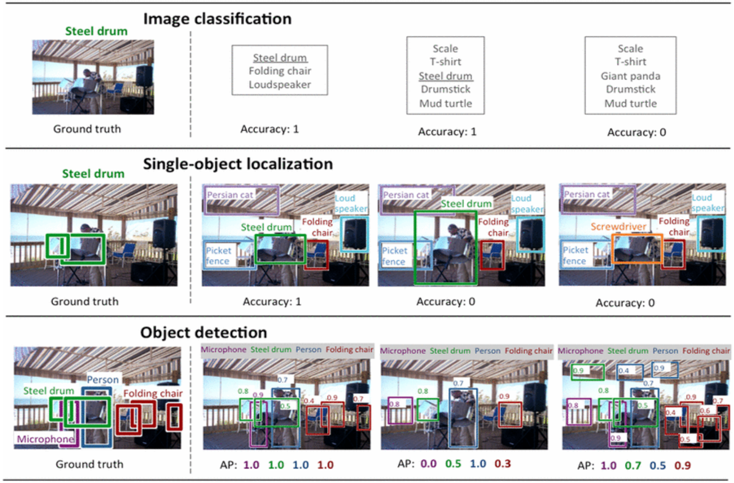


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [41] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

[15]`https://arxiv.org/pdf/1512.03385.pdf`

Example of ILSVRC localisation and object detection

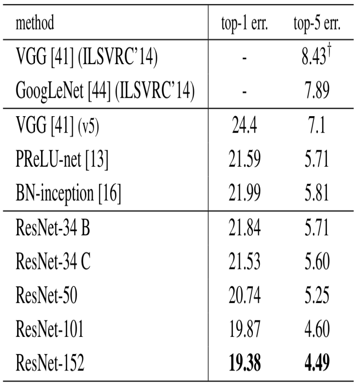


[16]

2013-14: All object classification per scene; 456,000 training set

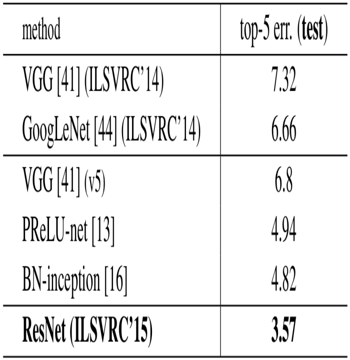
[16] ImageNet Large Scale Visual Recognition Challenge, Russakovsky et al. 2015.

Residual Networks (He 15'[18])



method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except † reported on the test set).



method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Table 5. Error rates (%) of **ensembles**. The top-5 error is on the test set of ImageNet and reported by the test server.

See also Highway nets by Srivastava et al. 15[17] which very the amount of x being passed through a skip connection.

[17]https://arxiv.org/pdf/1505.00387.pdf

[18]https://arxiv.org/pdf/1512.03385.pdf

Very brief summary of the course

Architecture: expressivity benefits of depth, plus more complex notions such as VAEs, GANs, RNNs, LSTM, Capsules, ResNets

Data: High dimensional data of interest living on a much lower dimensional space than the ambient dimension. Deep nets are able to effectively express functions that take on desired values on these lower dimensional manifolds. Models for real world data and deep net approximation rates is an area in need of further work.

Learning (ability to train): scalable and efficient optimization algorithms are essential, but must good initializations and smart architectures are also essential: batch normalization, vanishing/exploding gradients, ResNets, etc...

Applications: while not the focus here, much of the interest in deep learning is due to its remarkable efficacy in a wide range of applications from computer vision, to playing games, to generating realistic synthetic objects/art. Thank you for your attention