Why Deep Neural Networks

Danilo Pau

Advanced System Technology

Agrate Brianza

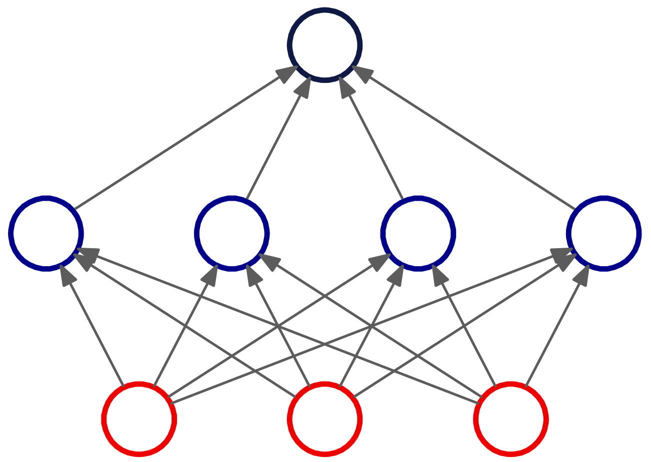
life.augmented

Shallow vs. Deep Feed-Forward Neural Networks

- **Increasing network depth**

$$\tilde{y} = \boldsymbol{w} \cdot g(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{c}^{(1)}) + c$$

  - A feed-forward neural network with one hidden layer



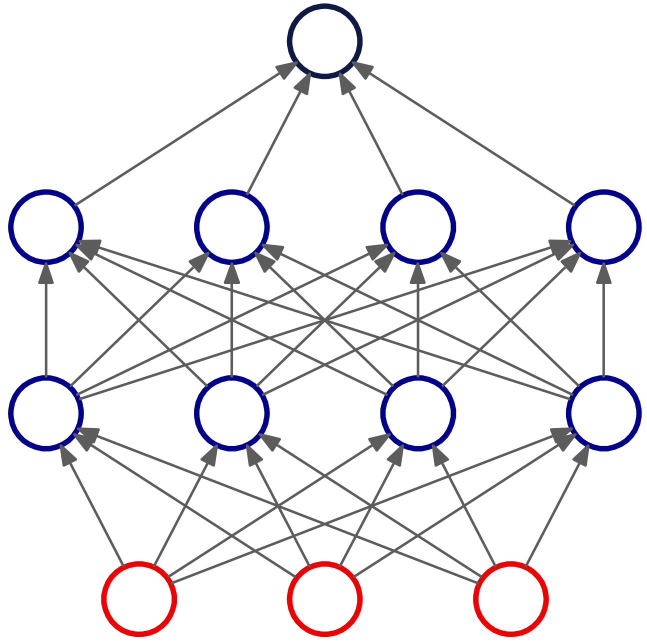
Credits https://vision.unipv.it/AI/AIRG.html

Shallow vs. Deep Feed-Forward Neural Networks

- **Increasing network depth**

$$\tilde{y} = \boldsymbol{w} \cdot g(\boldsymbol{W}^{(1)} g(\boldsymbol{W}^{(2)} \boldsymbol{x} + \boldsymbol{c}^{(2)}) + \boldsymbol{c}^{(1)}) + c$$

  - A feed-forward neural network with two hidden layers



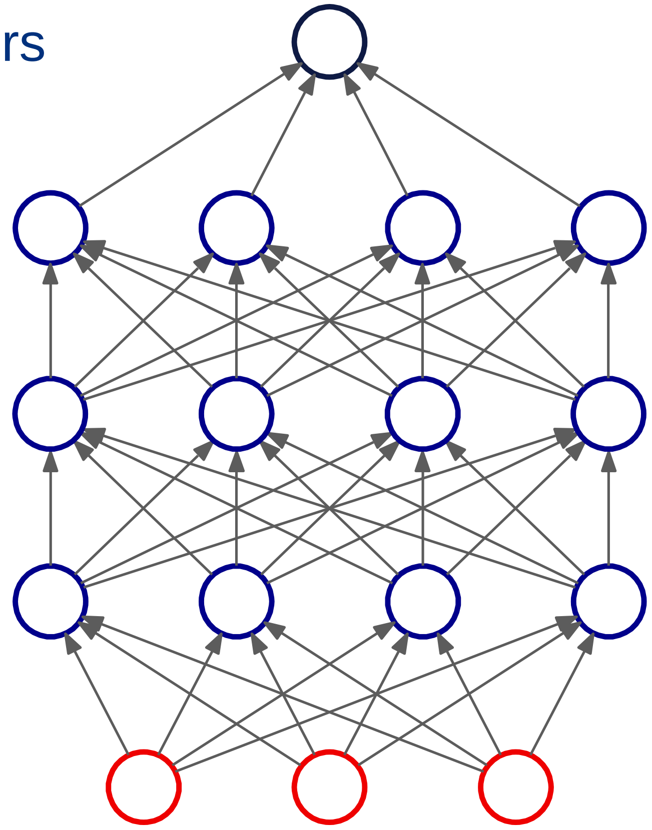
Credits https://vision.unipv.it/AI/AIRG.html

Shallow vs. Deep Feed-Forward Neural Networks

- **Increasing network depth**

$$\tilde{y} = \boldsymbol{w} \cdot g(\boldsymbol{W}^{(1)} g(\boldsymbol{W}^{(2)} g(\boldsymbol{W}^{(3)} \boldsymbol{x} + \boldsymbol{c}^{(3)}) + \boldsymbol{c}^{(2)}) + \boldsymbol{c}^{(1)}) + c$$

  - A feed-forward neural network with three hidden layers



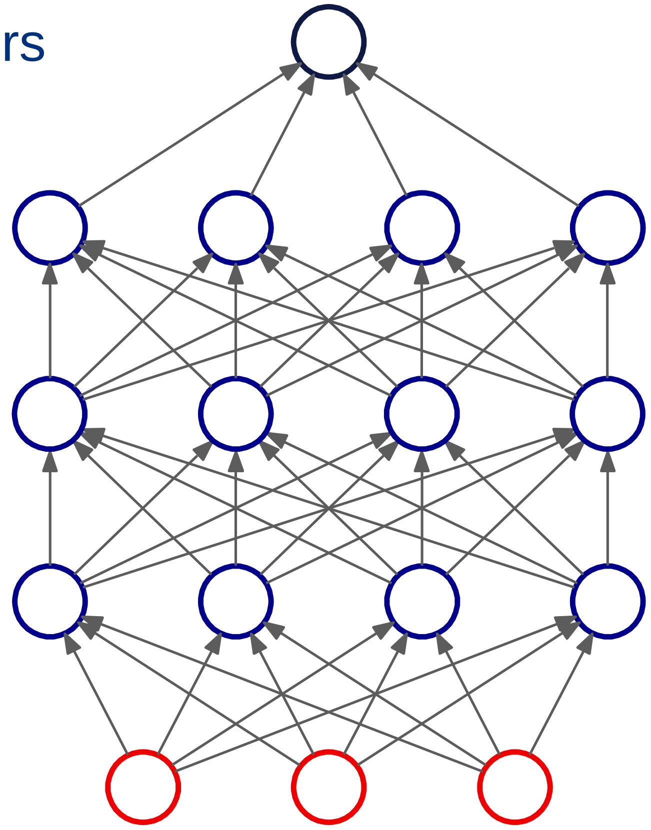
Credits https://vision.unipv.it/AI/AIRG.html

Shallow vs. Deep Feed-Forward Neural Networks

- **Increasing network depth**

$$\tilde{y} = \boldsymbol{w} \cdot g(\boldsymbol{W}^{(1)} g(\boldsymbol{W}^{(2)} g(\boldsymbol{W}^{(3)} \boldsymbol{x} + \boldsymbol{c}^{(3)}) + \boldsymbol{c}^{(2)}) + \boldsymbol{c}^{(1)}) + c$$

  - A feed-forward neural network with three hidden layers

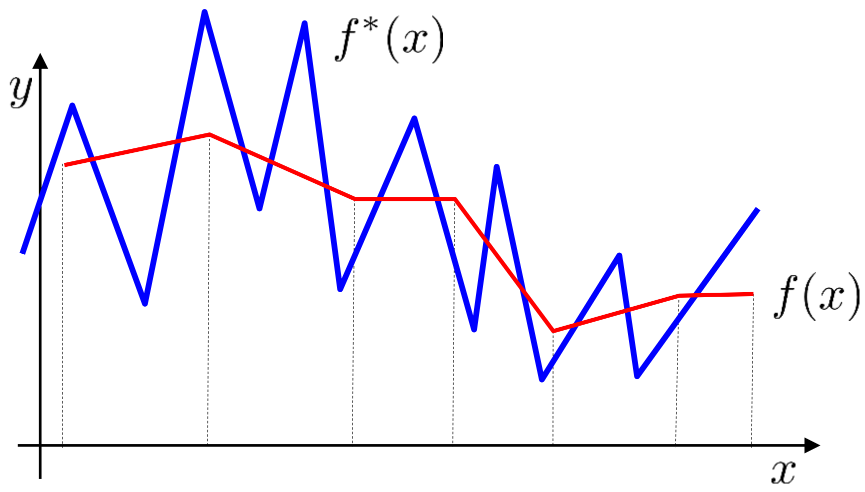


  - What is the need for such increase in depth?
  - **Universal Approximation Theorem** states one layer is enough…
  - …and each layer brings in some extra computational complexity and further parameters.

Credits https://vision.unipv.it/AI/AIRG.html

Piecewise linear functions

- How to approximate a zig-zag function:



- Intuitively, the accuracy of the approximation depends on x input space partitioning
- Without enough regions in the partition, approximation will be inaccurate
- Assume we want to use a deep neural network with ReLU

$$\tilde{y} = \boldsymbol{w} \cdot max(0, \boldsymbol{W}^{(1)} \cdots max(0, \boldsymbol{W}^{(k)} x + \boldsymbol{c}^{(k)}) \cdots + \boldsymbol{c}^{(1)}) + c$$

Credits https://vision.unipv.it/AI/AIRG.html

life.augmented

Piecewise linear functions

- Using a deep neural network with ReLU as approximator

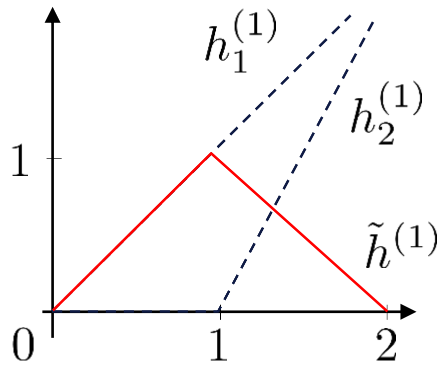
$$\tilde{y} = \boldsymbol{w} \cdot max(0, \boldsymbol{W}^{(1)} \cdots max(0, \boldsymbol{W}^{(k)} x + \boldsymbol{c}^{(k)}) \cdots + \boldsymbol{c}^{(1)}) + c$$

$$\boldsymbol{h}^{(1)} := [h_1^{(1)}, h_2^{(1)}]$$

$$h_1^{(1)} := \max(0, x)$$

$$h_2^{(1)} := \max(0, 2(x-1))$$

$$\tilde{h}^{(1)} := \max(0, x) - \max(0, 2(x-1))$$

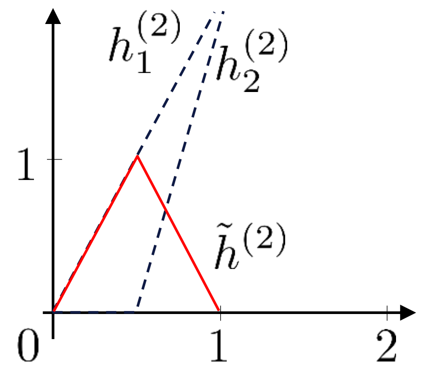


$$\boldsymbol{h}^{(2)} := [h_1^{(2)}, h_2^{(2)}]$$

$$h_1^{(2)} := \max(0, 2x)$$

$$h_2^{(2)} := \max(0, 4(x-1/2))$$

$$\tilde{h}^{(2)} := \max(0, 2x) - \max(0, 4(x-1/2))$$



$$x \in [0, 2]$$

Credits https://vision.unipv.it/AI/AIRG.html

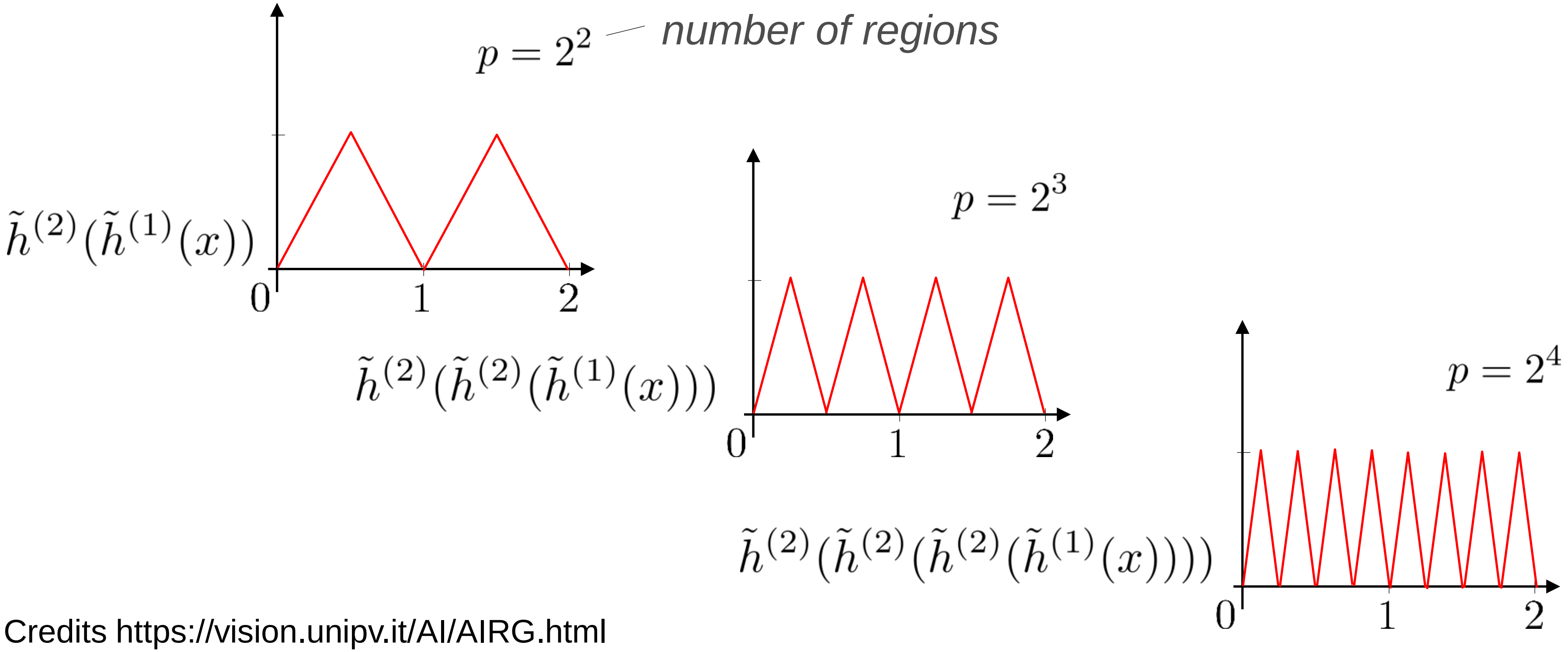
life.augmented

Piecewise linear functions

- Using a deep neural network with ReLU as approximator

$$\tilde{y} = \boldsymbol{w} \cdot max(0, \boldsymbol{W}^{(1)} \cdots max(0, \boldsymbol{W}^{(k)} x + \boldsymbol{c}^{(k)}) \cdots + \boldsymbol{c}^{(1)}) + c$$

- Assume that all hidden layers k > 2 are identical to $h^{(2)}$



Credits https://vision.unipv.it/AI/AIRG.html

Effect of Depth

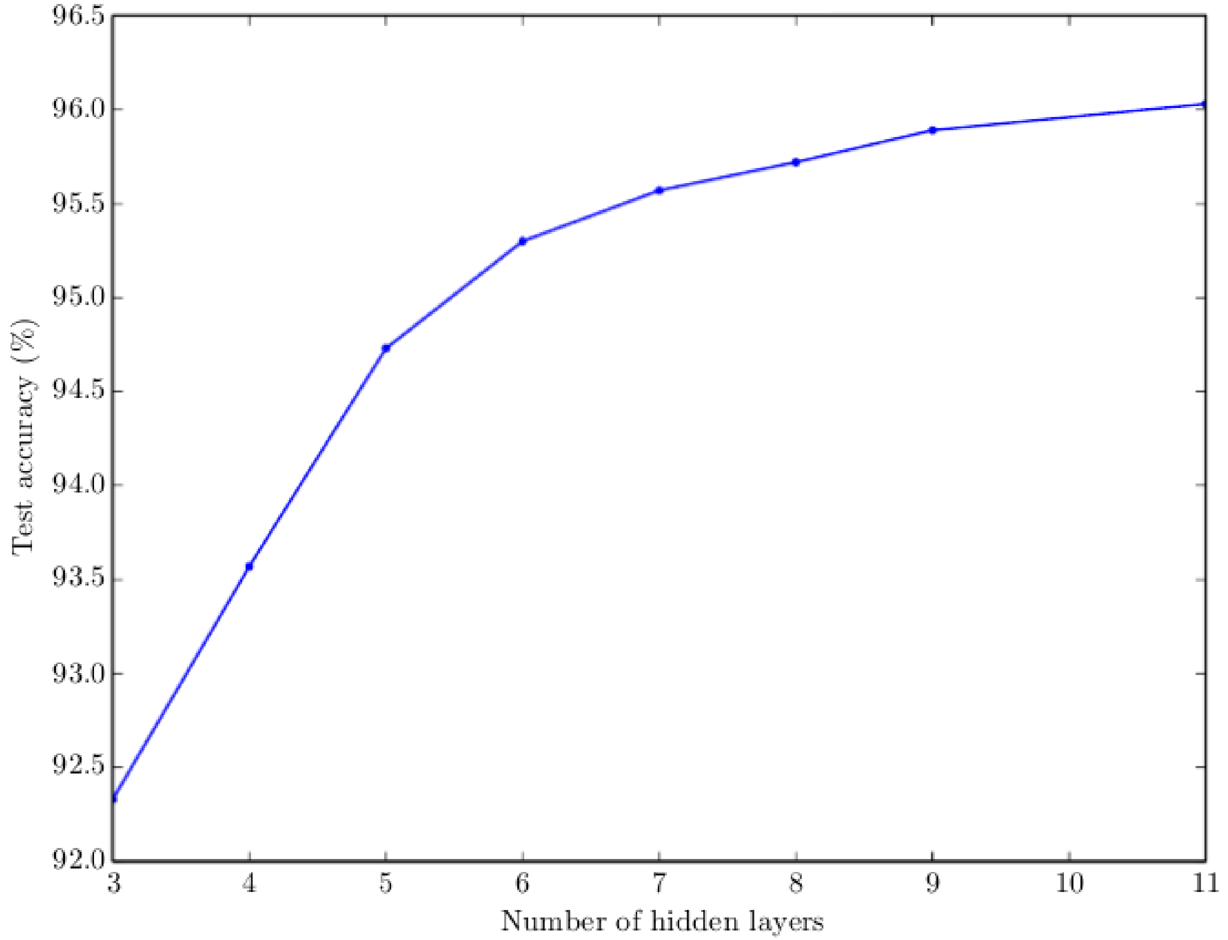
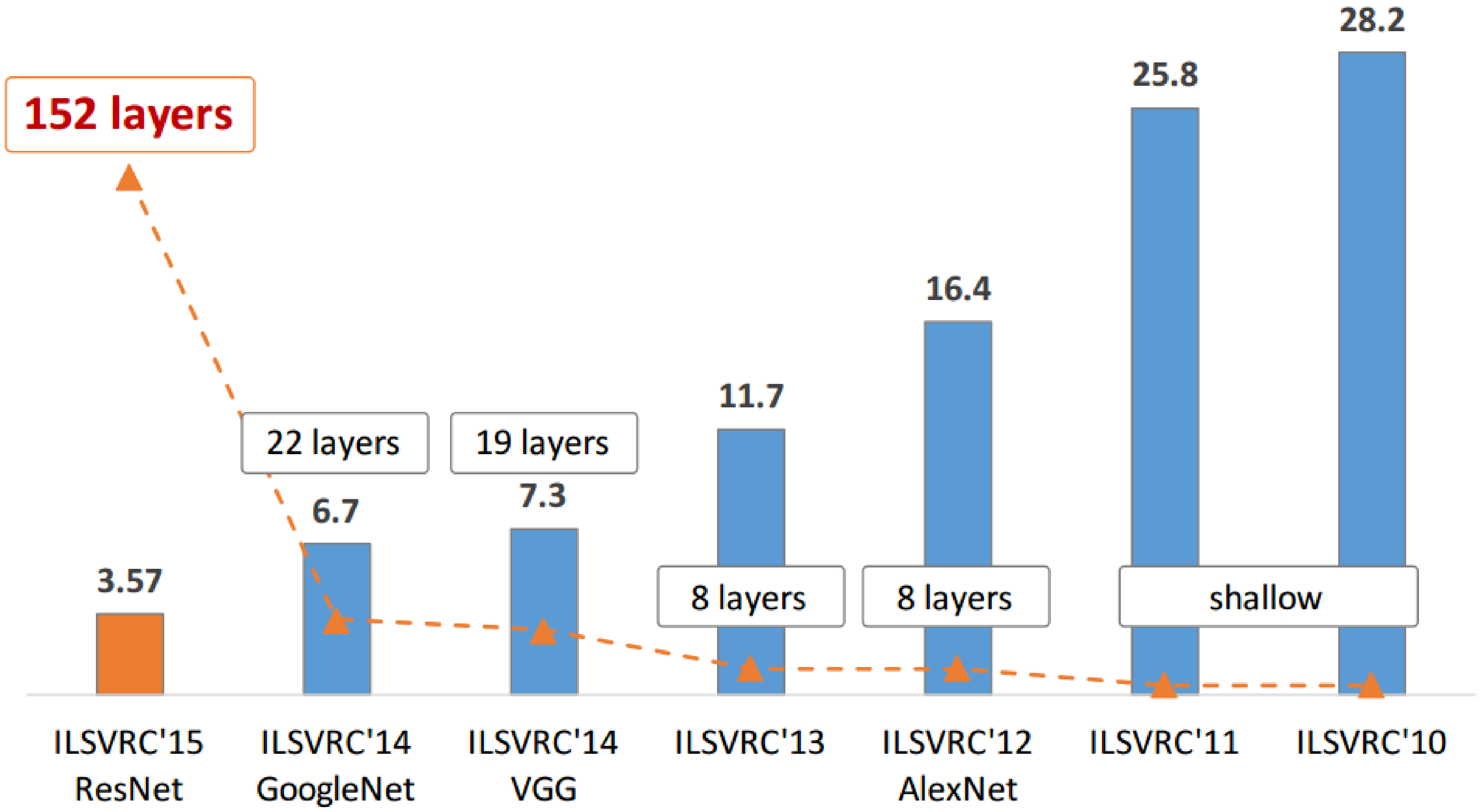


Figure 6.6: Empirical results showing that deeper networks generalize better when used to transcribe multi-digit numbers from photographs of addresses. Data from Goodfellow *et al.* (2014d). The test set accuracy consistently increases with increasing depth. See Fig. 6.7 for a control experiment demonstrating that other increases to the model size do not yield the same effect.

life.augmented

Effect of Depth



ImageNet Classification top-5 error (%)