Lecture 22
Theory, Depth, Representation, Future

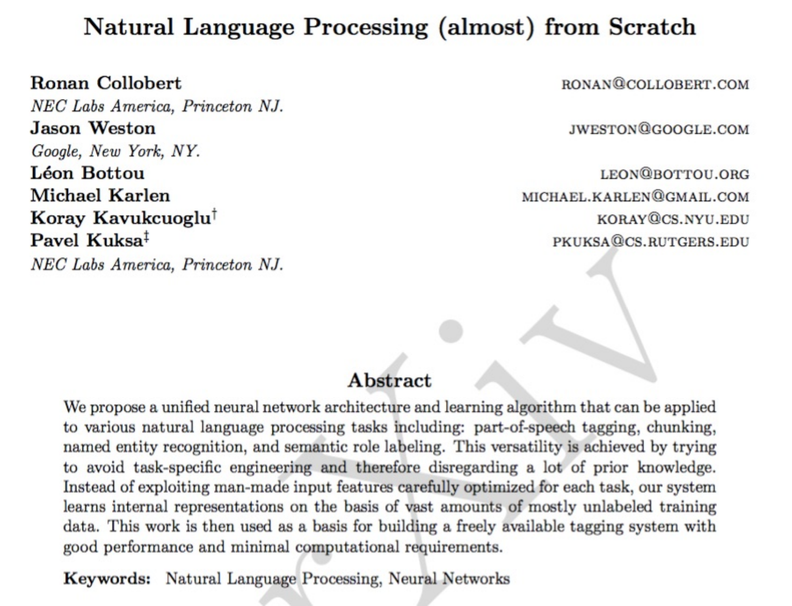
27 April 2016

Taylor B. Arnold
Yale Statistics
STAT 365/665

Yale

Notes

- Problem sets 7 and 8 will be returned by Thursday (maybe early Friday)
- Problem set 9 is due next Monday
- Blog posts coming soon!



Natural Language Processing (almost) from Scratch

Ronan Collobert RONAN@COLLOBERT.COM
NEC Labs America, Princeton NJ.
Jason Weston JWESTON@GOOGLE.COM
Google, New York, NY.
Léon Bottou LEON@BOTTOU.ORG
Michael Karlen MICHAEL.KARLEN@GMAIL.COM
Koray Kavukcuoglu[†] KORAY@CS.NYU.EDU
Pavel Kuksa[‡] PKUKSA@CS.RUTGERS.EDU
NEC Labs America, Princeton NJ.

Abstract

We propose a unified neural network architecture and learning algorithm that can be applied to various natural language processing tasks including: part-of-speech tagging, chunking, named entity recognition, and semantic role labeling. This versatility is achieved by trying to avoid task-specific engineering and therefore disregarding a lot of prior knowledge. Instead of exploiting man-made input features carefully optimized for each task, our system learns internal representations on the basis of vast amounts of mostly unlabeled training data. This work is then used as a basis for building a freely available tagging system with good performance and minimal computational requirements.

Keywords: Natural Language Processing, Neural Networks

The full citation:

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. "Natural language processing (almost) from scratch." The Journal of Machine Learning Research 12 (2011): 2493-2537.

Theory

One of the earliest theoretical results relating to neural networks:

> *Barron, Andrew. "Universal Approximation Bounds for Superpositions of a Sigmoidal Function." IEEE Transactions on Information Theory, Vol. 39, No.3, May 1993.*

Consider functions of the form:

$$f_n(x) = \sum_{k=1}^{n} c_k \cdot \sigma(a_k x + b_k) + c_0$$

Which map $\mathbb{R}^d$ into $\mathbb{R}$.

This is a neural network with one hidden layer and a single output. The parameters a_k are the hidden weights, the b_k are the biases, c_k are the output weights, and c_0 is the output bias.

In the paper it is shown that for a large class of functions f, we can find a neural network such that:

$$\int_{B_r} (f(x) - f_n(x))^2 \, dx \leq C \times \frac{r^2}{n}$$

For some constant $C > 0$.

This is a formal proof that even shallow neural networks are universal approximators.

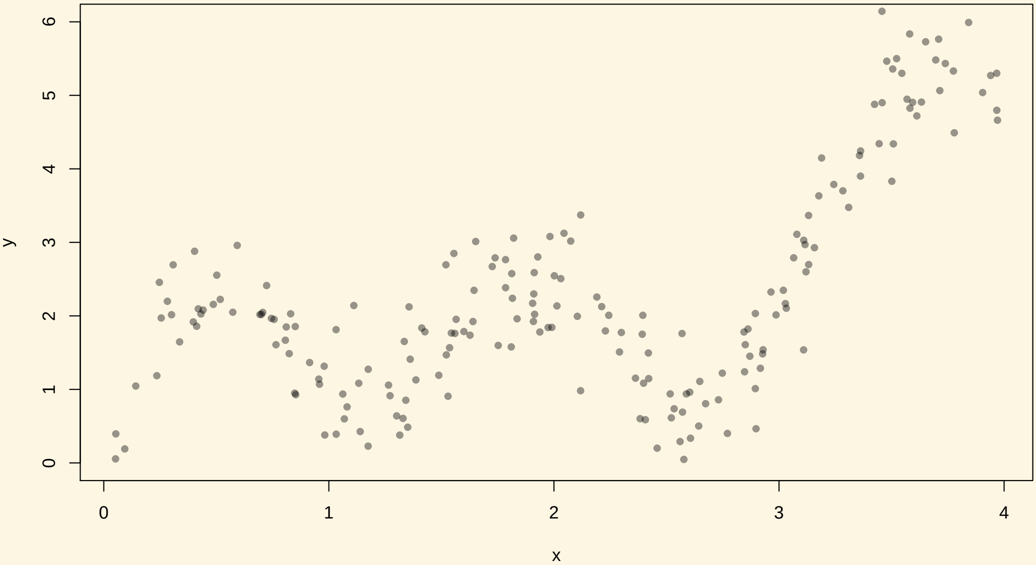
Only recently have we seen theory addressing how well neural networks can reconstruct generative models under noisy observations. Two of the most well-known include:

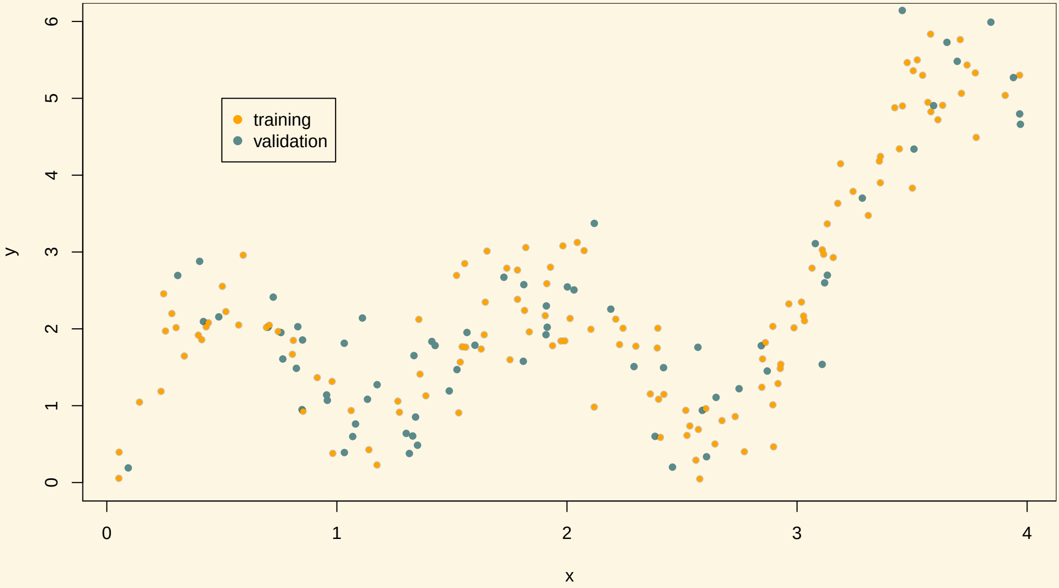
> *Bengio, Yoshua, et al. "Generalized denoising auto-encoders as generative models." Advances in Neural Information Processing Systems. 2013.*

And:

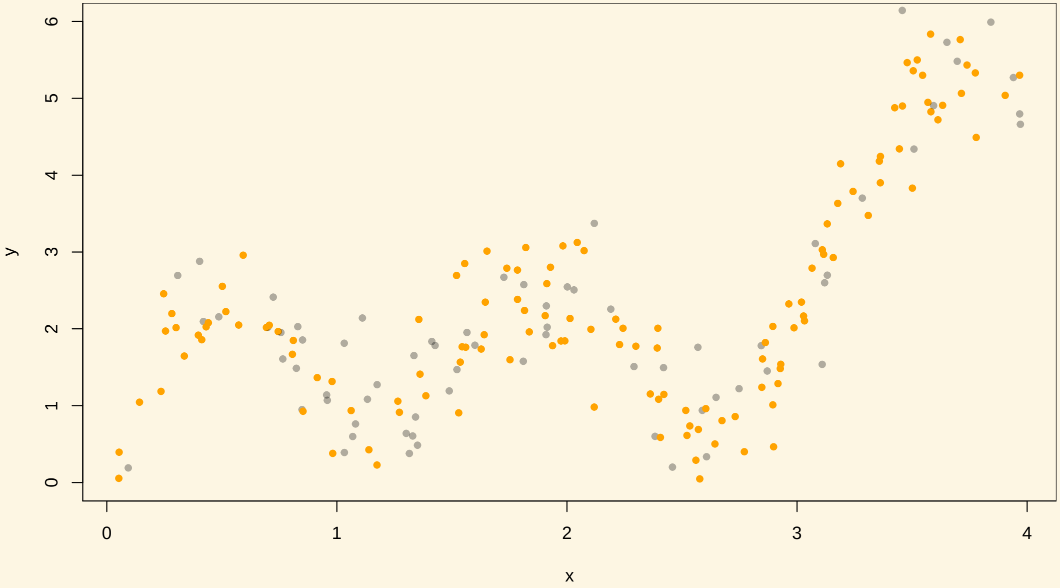
> *Alain, Guillaume, and Yoshua Bengio. "What regularized auto-encoders learn from the data-generating distribution."The Journal of Machine Learning Research 15.1 (2014): 3563-3593.*

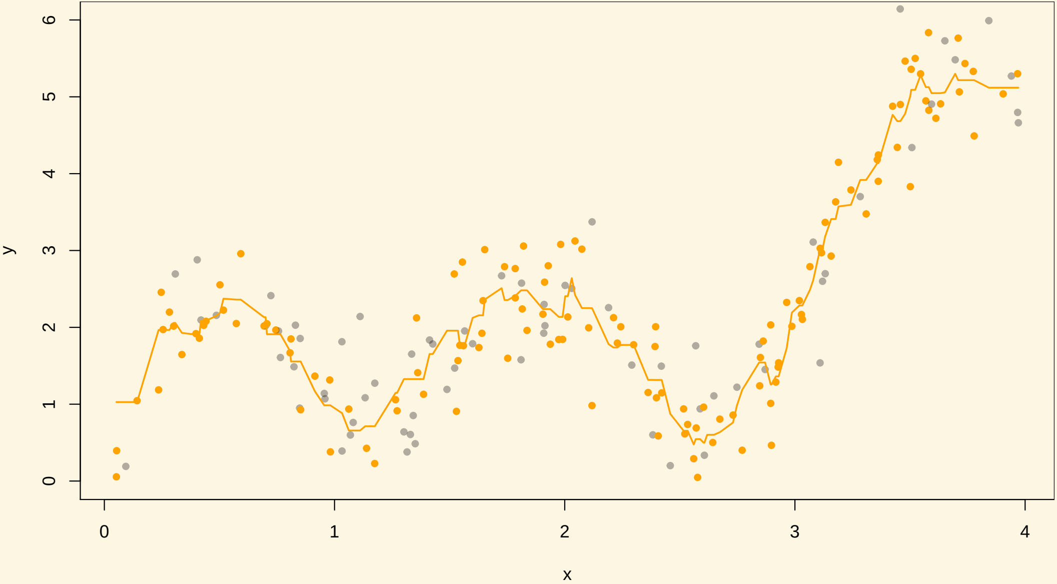
If you are interested in this line of work, I suggest setting up an arXiv alert for people such as Yoshia Bengio, Guillaume Alain, Razvan Pascanu, and Guido Montúfar.

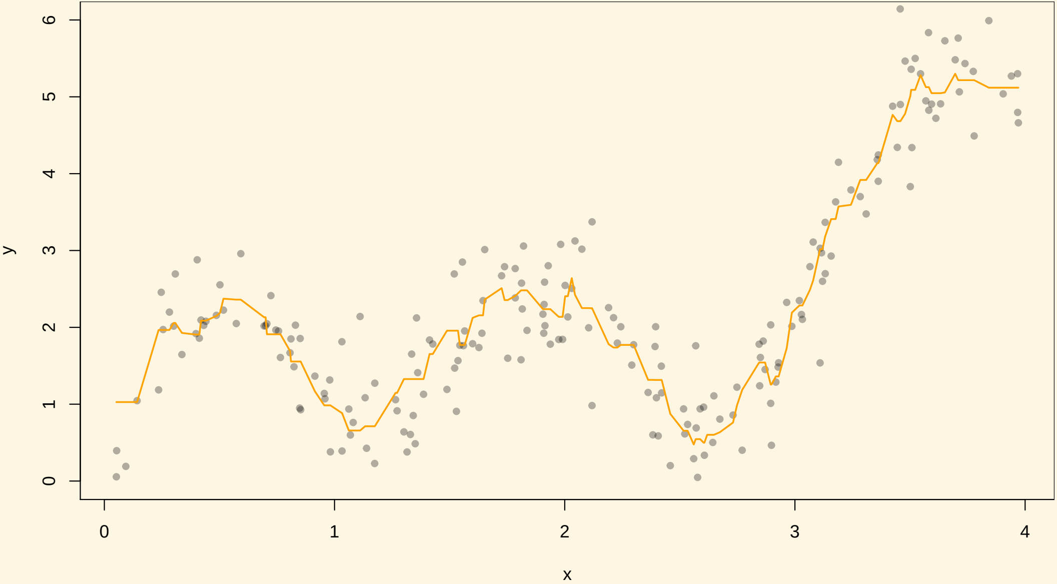

y
0
1
2
3
4
5
6
x
0
1
2
3
4

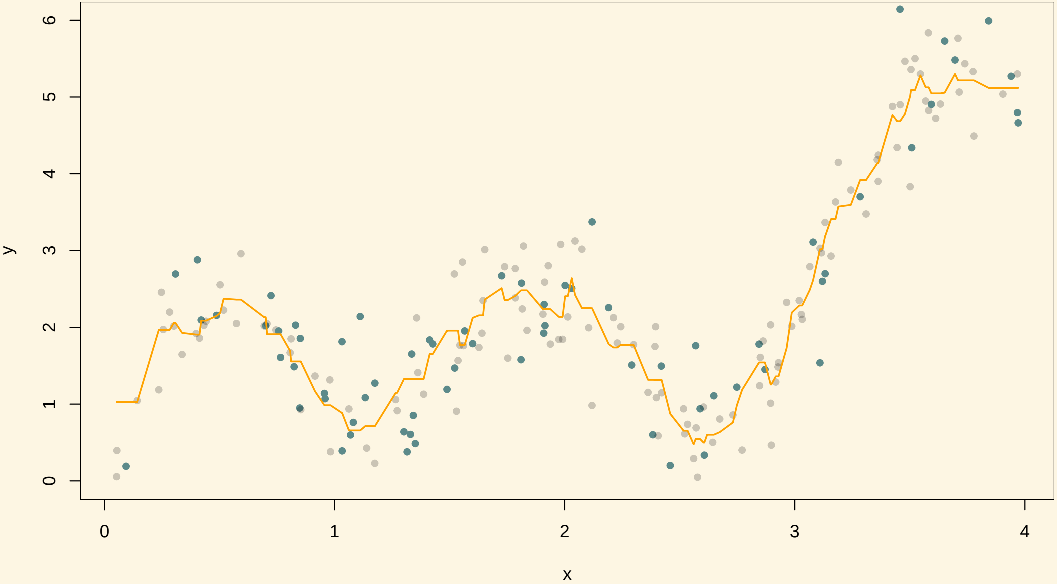


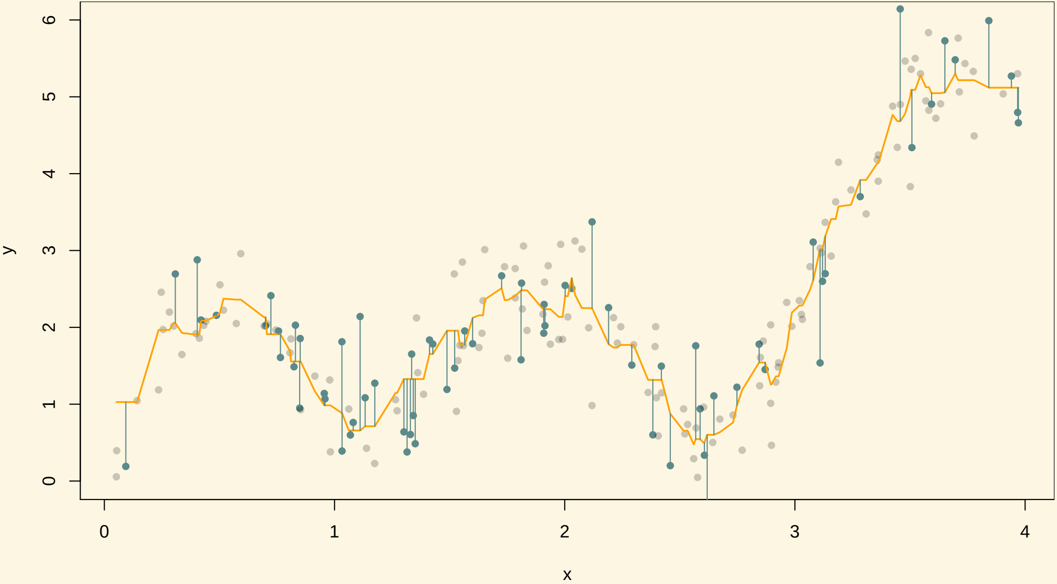
training
validation
0
1
2
3
4
5
6
y
0
1
2
3
4
x


y
0
1
2
3
4
5
6
0
1
2
3
4
x


y
0
1
2
3
4
5
6
x
0
1
2
3
4


y
0
1
2
3
4
5
6
x
0
1
2
3
4


y
0
1
2
3
4
5
6
0
1
2
3
4
x


y
0
1
2
3
4
5
6
x
0
1
2
3
4

Depth

So if shallow neural networks can represent arbitrary functions, why have we been creating deeper and deeper networks? A recent theoretical paper tries to explain why deeper networks perform significantly better:

> *Montufar, Guido F., et al. "On the number of linear regions of deep neural networks." Advances in neural information processing systems. 2014.*

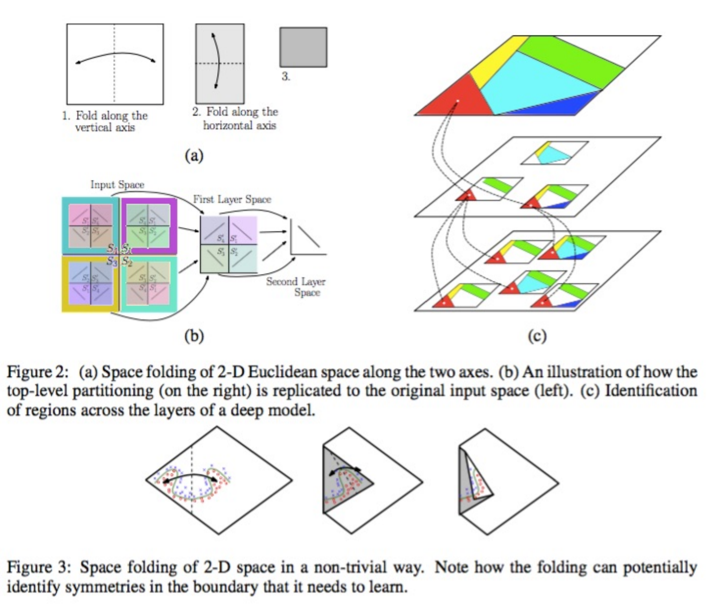


Figure 2: (a) Space folding of 2-D Euclidean space along the two axes. (b) An illustration of how the top-level partitioning (on the right) is replicated to the original input space (left). (c) Identification of regions across the layers of a deep model.

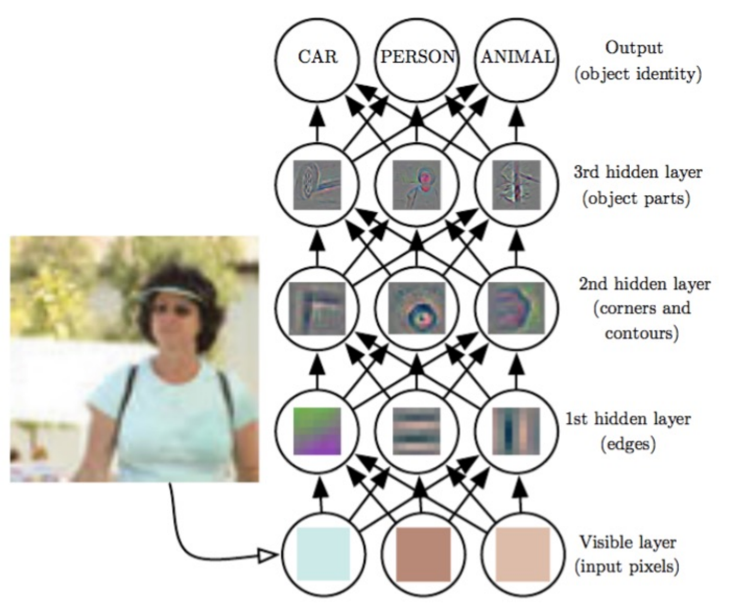
Figure 3: Space folding of 2-D space in a non-trivial way. Note how the folding can potentially identify symmetries in the boundary that it needs to learn.

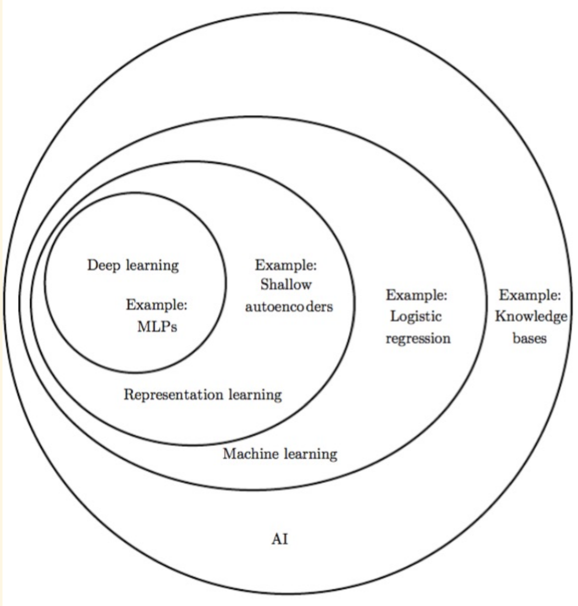
A summary of one result from the paper: A dense neural network with rectified linear activations having n_0 input units and L hidden layers of width $n \geq n_0$ can compute function that have:

$$\Omega\left(\left(\frac{n}{n_0}\right)^{(L-1)\cdot n_0} n^{n_0}\right)$$

Number of linear regions. This shows that the expressibility of the network grows exponentially with L but only polynomially with n.

So, deeper models approximate a larger class of functions with fewer parameters.


CAR
PERSON
ANIMAL
Output
(object identity)
3rd hidden layer
(object parts)
2nd hidden layer
(corners and
contours)
1st hidden layer
(edges)
Visible layer
(input pixels)


Deep learning
Example:
MLPs
Example:
Shallow
autoencoders
Representation learning
Example:
Logistic
regression
Machine learning
Example:
Knowledge
bases
AI

Representation

I. Problem Set 8 part 1
II. Problem Set 8, Part 2
III. Transfer Learning: IMDB Sentiment analysis

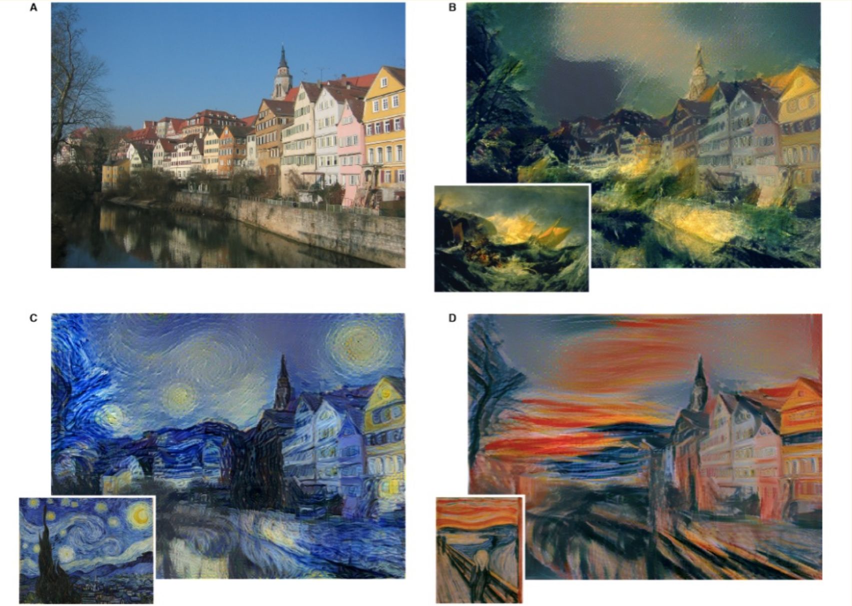
Neural networks have had amazingly successful results learning things such as basic mathematical operations:

> *Franco, Leonardo, and Sergio A. Cannas. "Solving arithmetic problems using feed-forward neural networks." Neurocomputing 18.1 (1998): 61-79.*

IV. Learning addition

There as also been work on using neural networks to capture subjective features such as painting style:

> *Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. ”A neural algorithm of artistic style.” arXiv preprint arXiv:1508.06576 (2015).*


A
B
C
D

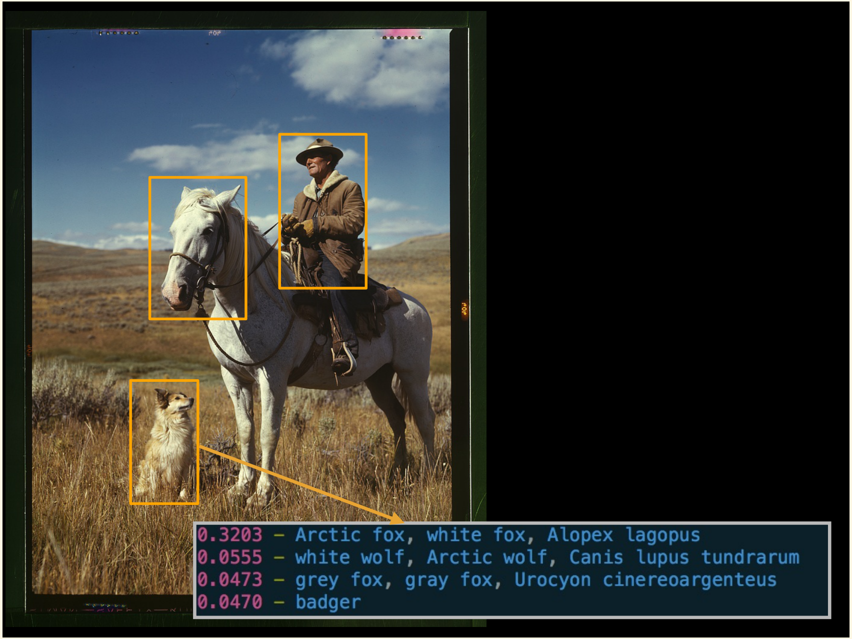
Future

Near-term popular areas of study in deep learning:

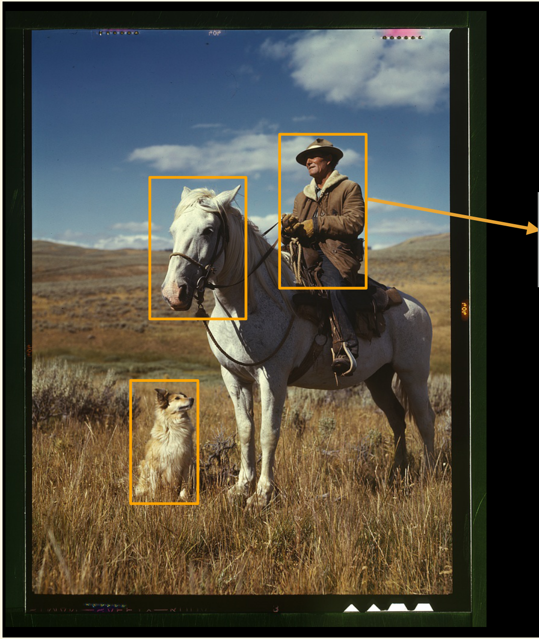
- compression of neural networks
- consolidating the CNN tricks and tips; when will this ever slow down or end?
- deep residual neural networks

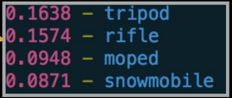
A robot at work:

http://www.youtube.com/watch?v=2yRRNGr_4yY&t=1m0s



```
0.3203 – Arctic fox, white fox, Alopex lagopus
0.0555 – white wolf, Arctic wolf, Canis lupus tundrarum
0.0473 – grey fox, gray fox, Urocyon cinereoargenteus
0.0470 – badger
```





```
0.1638 – tripod
0.1574 – rifle
0.0948 – moped
0.0871 – snowmobile
```

To me, one of the more exciting papers on deep learning produced in the past year:

> *Gregor, K., Danihelka, I., Graves, A., & Wierstra, D. (2015). DRAW: A recurrent neural network for image generation. arXiv preprint arXiv:1502.04623.*

It makes meaningful departures from prior methods and gets more directly at the generative model. It is also closer to our understanding of how visual processing happens in the brain.

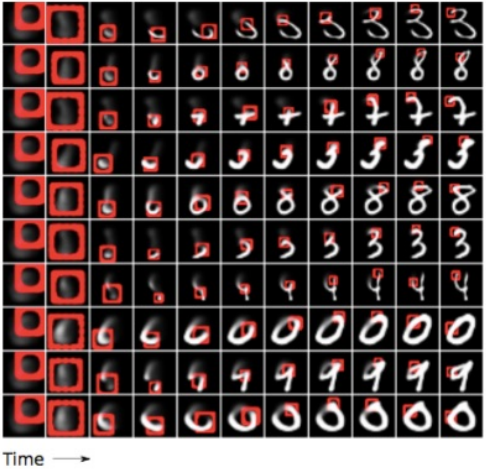


Figure 1. **A trained DRAW network generating MNIST digits.** Each row shows successive stages in the generation of a single digit. Note how the lines composing the digits appear to be "drawn" by the network. The red rectangle delimits the area attended to by the network at each time-step, with the focal precision indicated by the width of the rectangle border.

Longer-term areas in deep learning:

- deep reinforcement learning
- better architectures or training algorithms
- (real) unsupervised learning

Thanks!