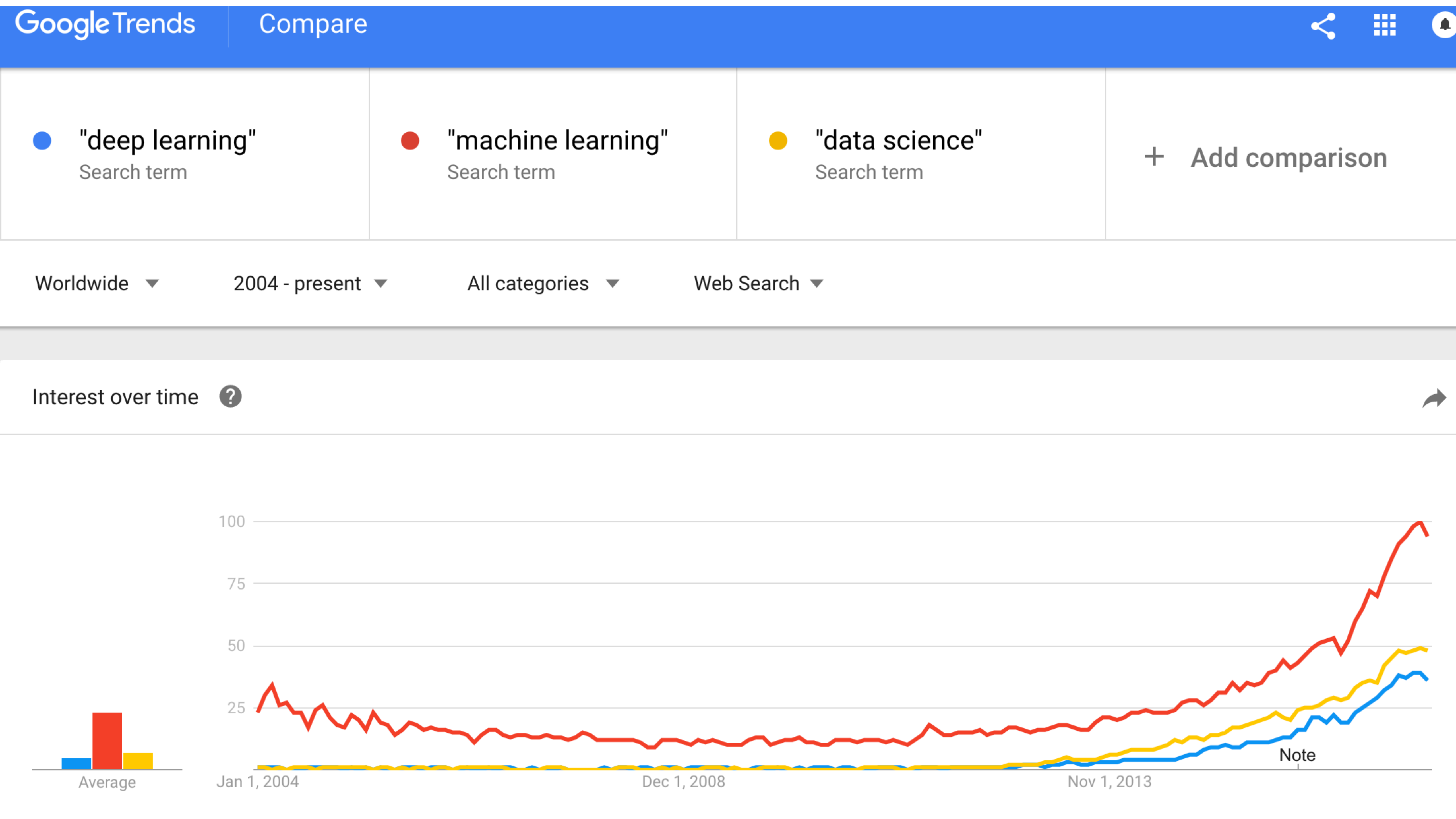
Introduction to Deep Learning

Andreas Damianou

Amazon.com

DSA Arusha, Tanzania, 19 July 2017


Google Trends
Compare
"deep learning"
Search term
"machine learning"
Search term
"data science"
Search term
+ Add comparison
Worldwide
2004 - present
All categories
Web Search
Interest over time
100
75
50
25
Average
Jan 1, 2004
Dec 1, 2008
Nov 1, 2013
Note

- Motivation
- Some cool stuff. Categorize: deep learning is … models include... Demonstrate with google classification api (take photos from room)
- 1 Unit = linear regression
- Losses. Connection with prob. models.
- Deeper . http://playground.tensorflow.org/
- Back-propagation & training. Learning rates & batch-size
- Implement my first Neural Network.
- Issues: overfitting (demo)
- Regularization: Dropout…
- Some architectures: FF, CNNs (perhaps show only spatial pyramids), RNNs, …
- NN weights as features. => Represetation learning
- Bayesian NN & generative modeling. Modeling p(X) vs only p(Y|X)
- Discussion: Strengths and drawbacks.

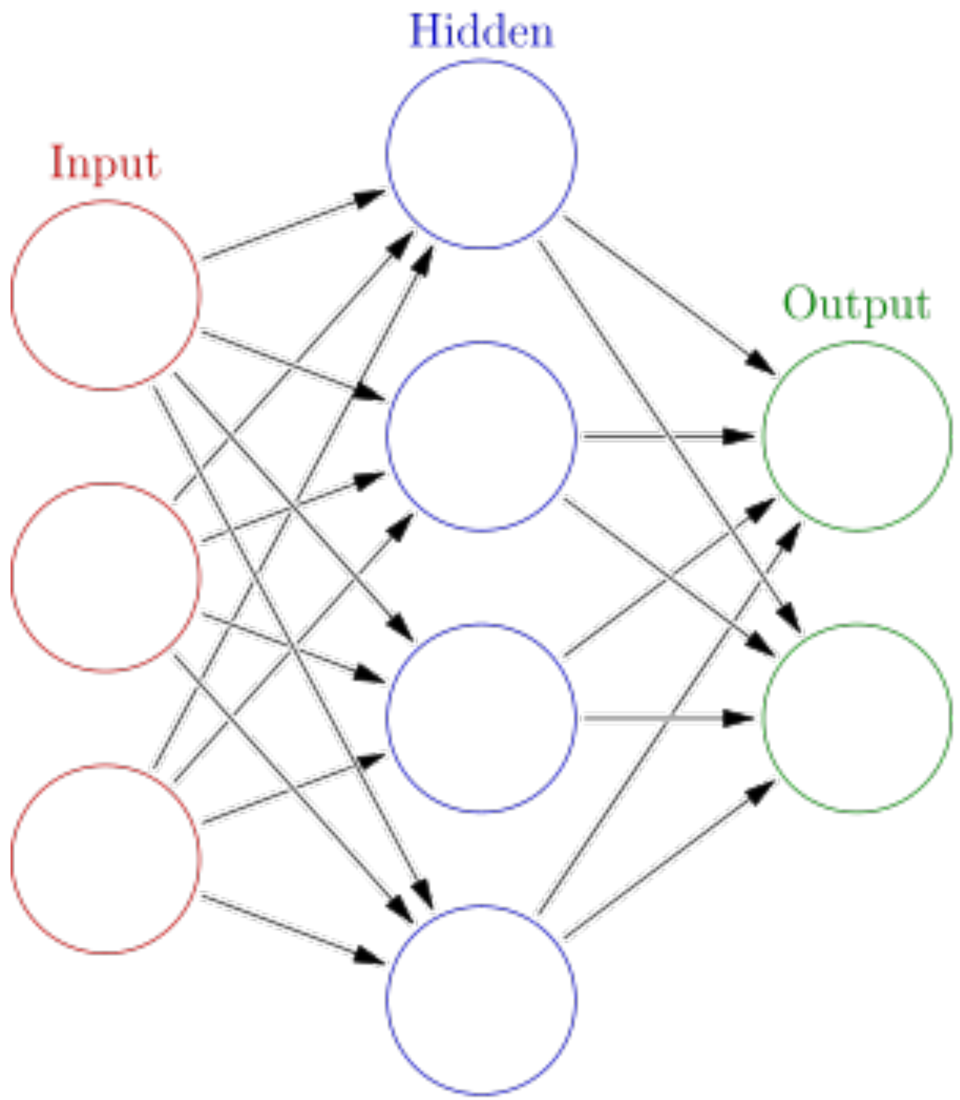
67 24

5 8 8

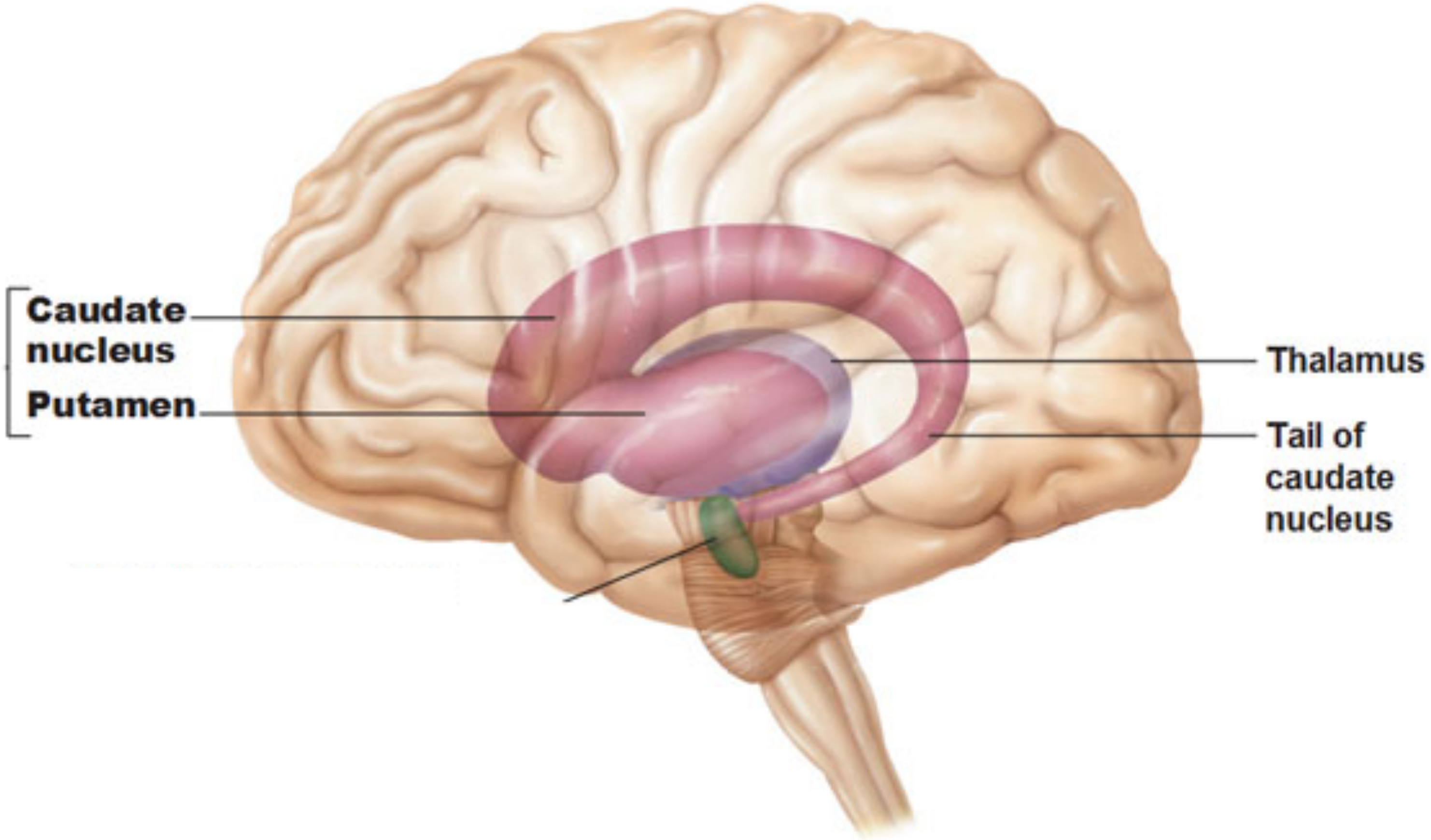
Motivation for Learning deeply

- Decompose learning task
- Learn simple concepts. Use this to build up knowledge of more complex concepts.

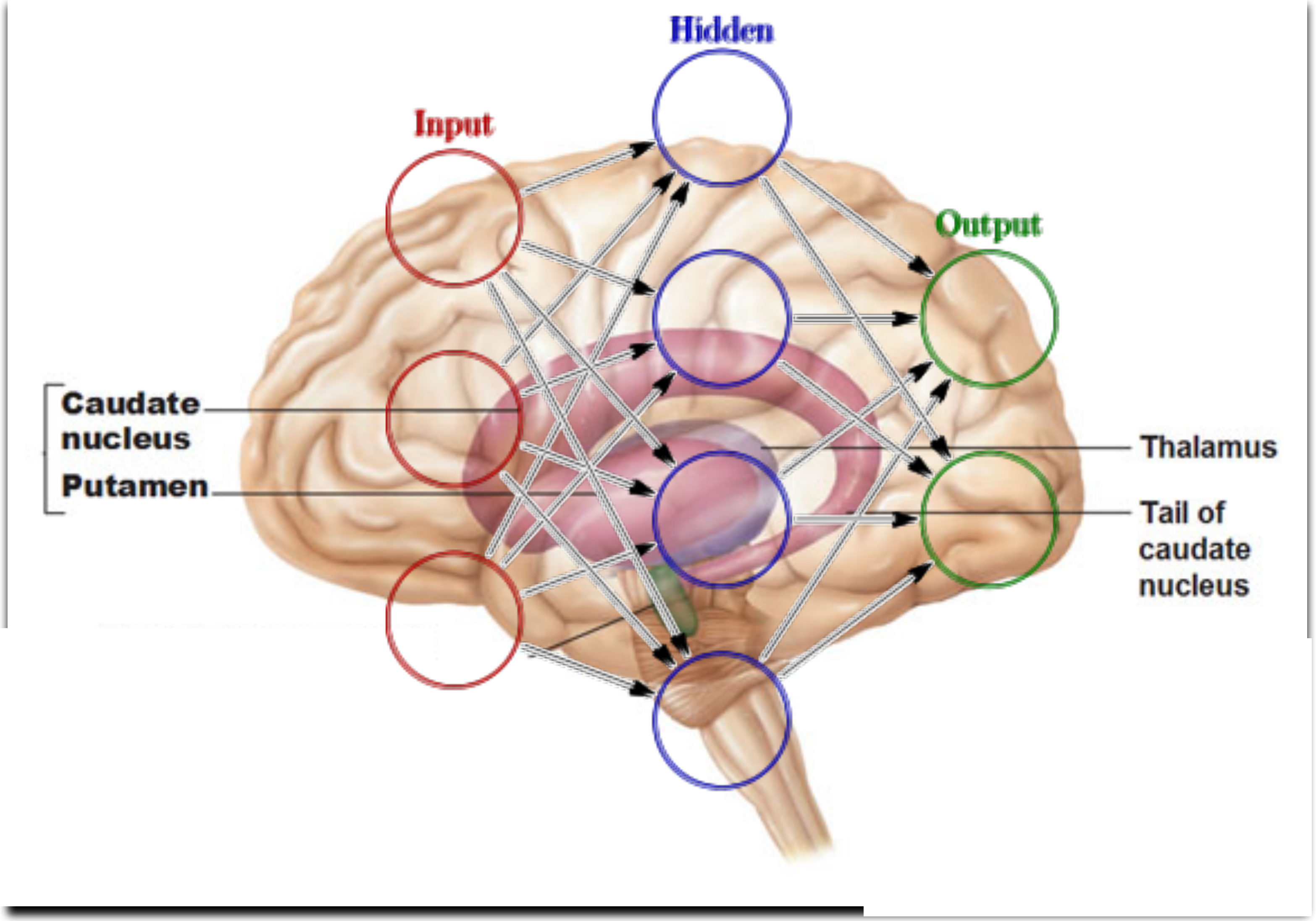
A neural network



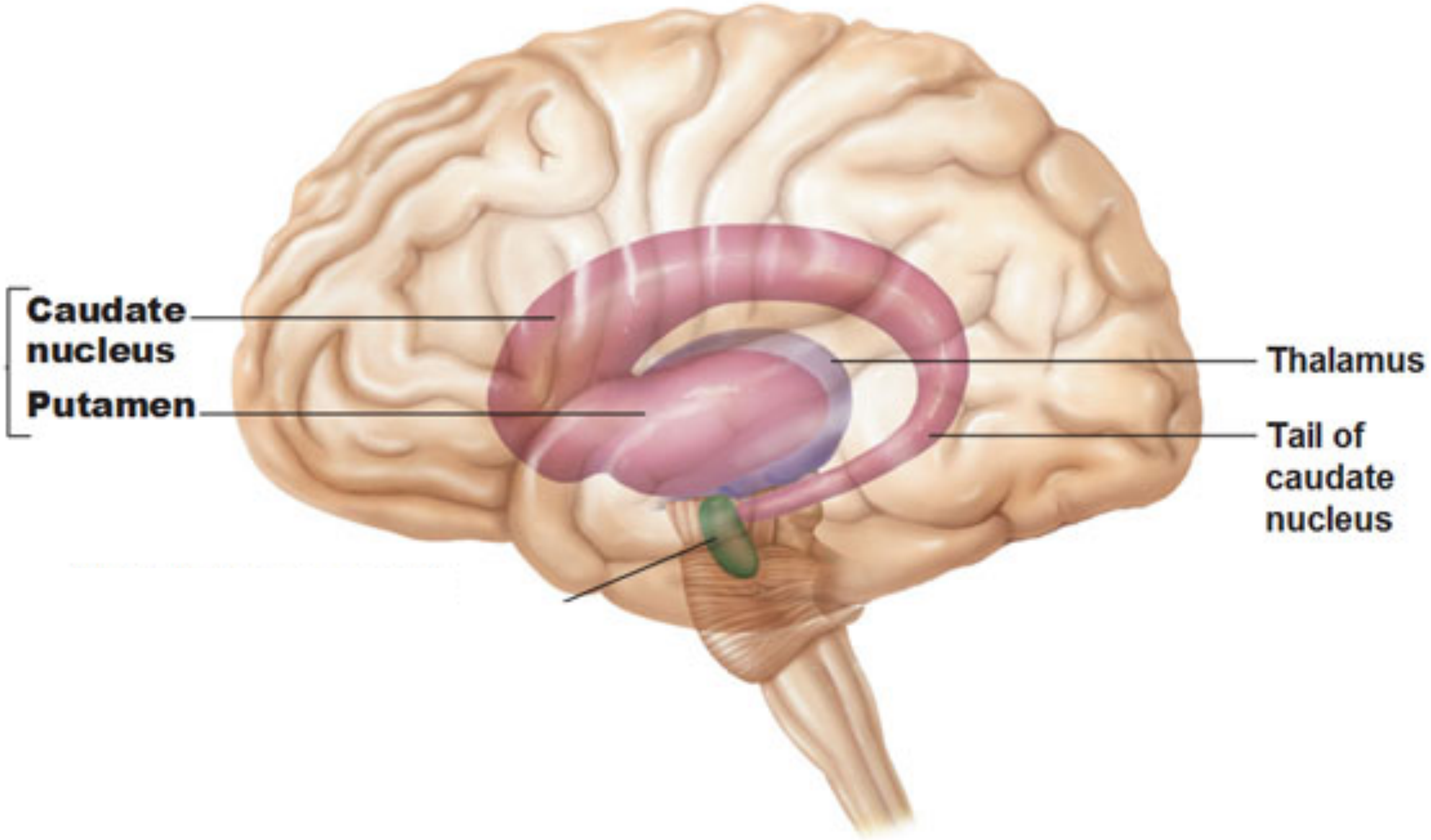
Connectionism



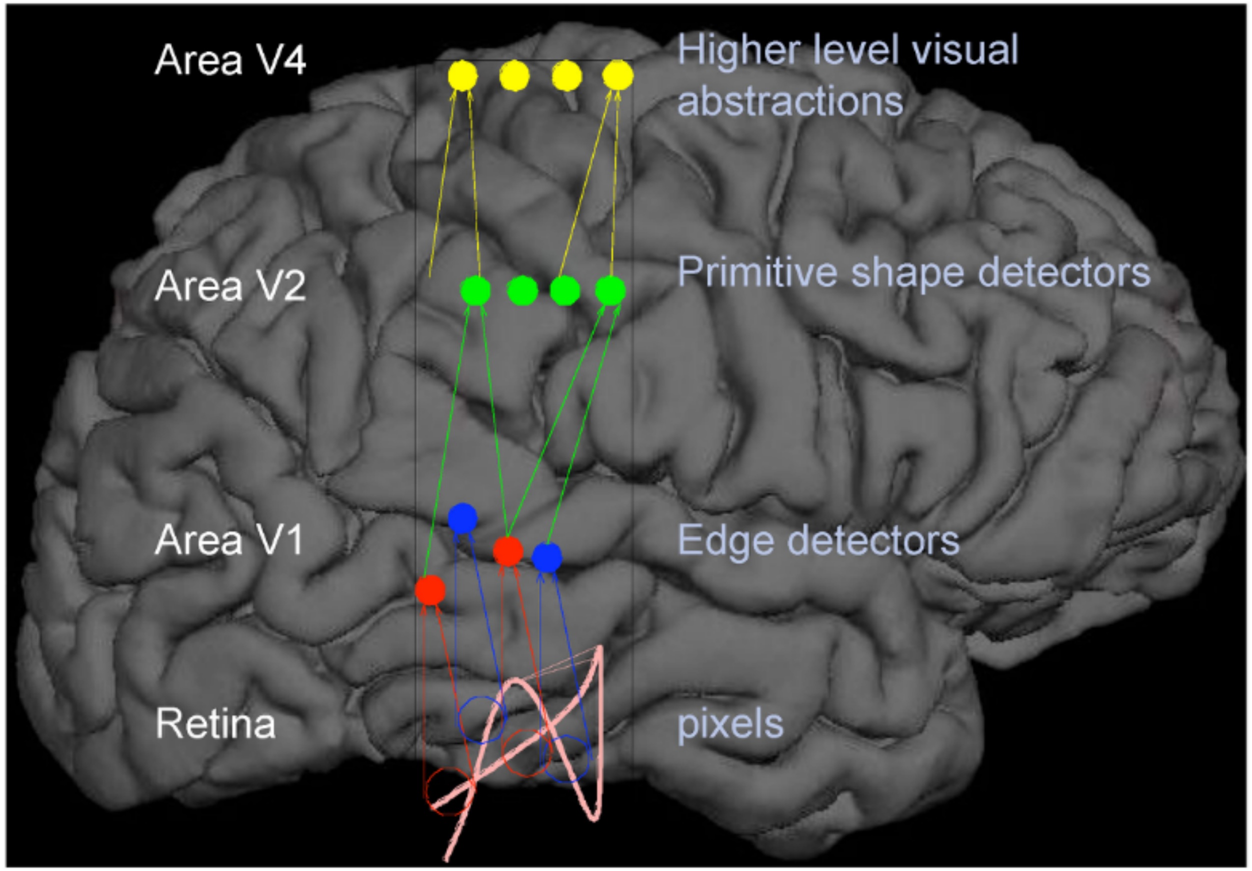
~~Connectionism~~ Neural Network



~~Connectionism~~ ~~Neural Network~~



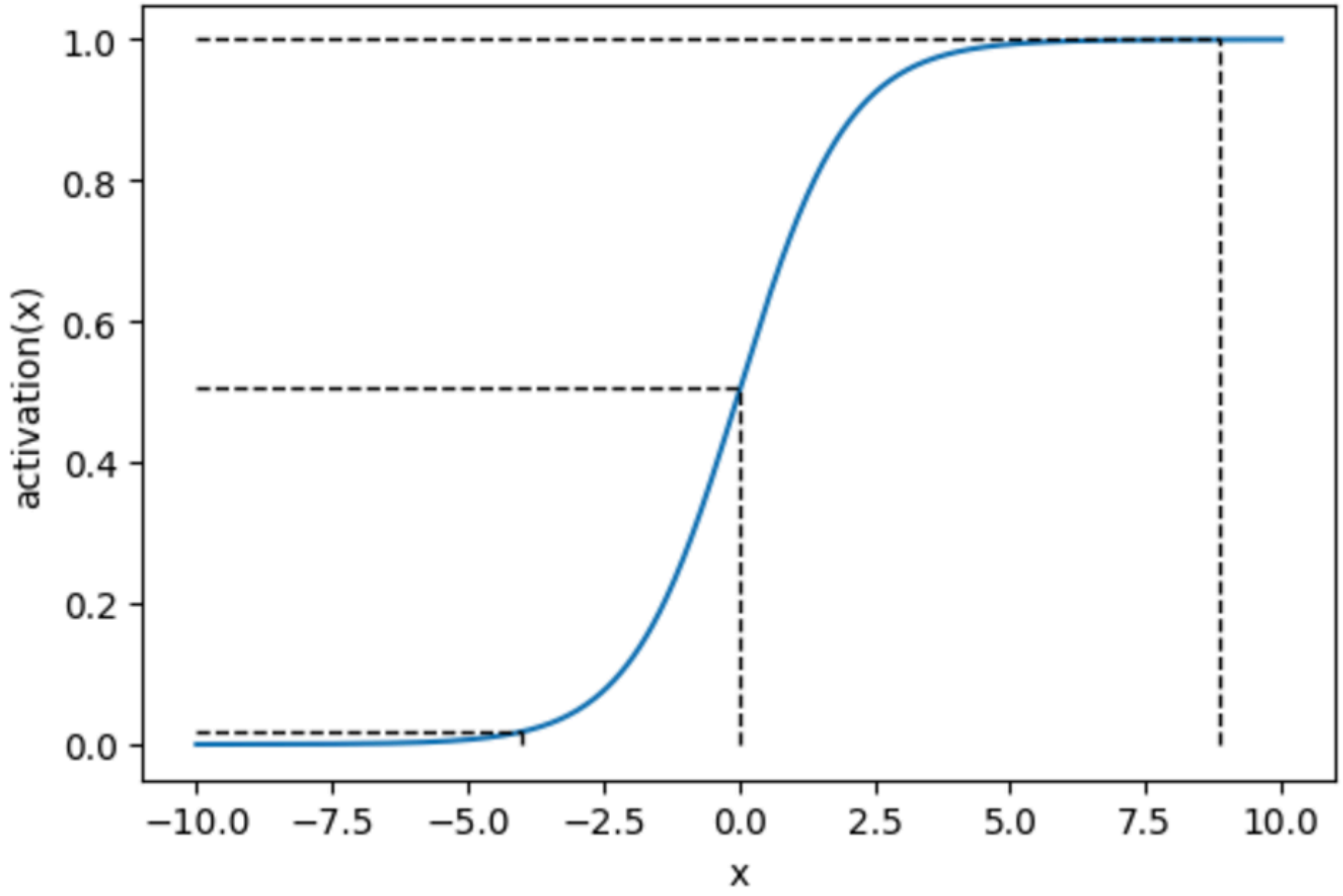
Deep Architecture in the Brain



Ref: antranik.org

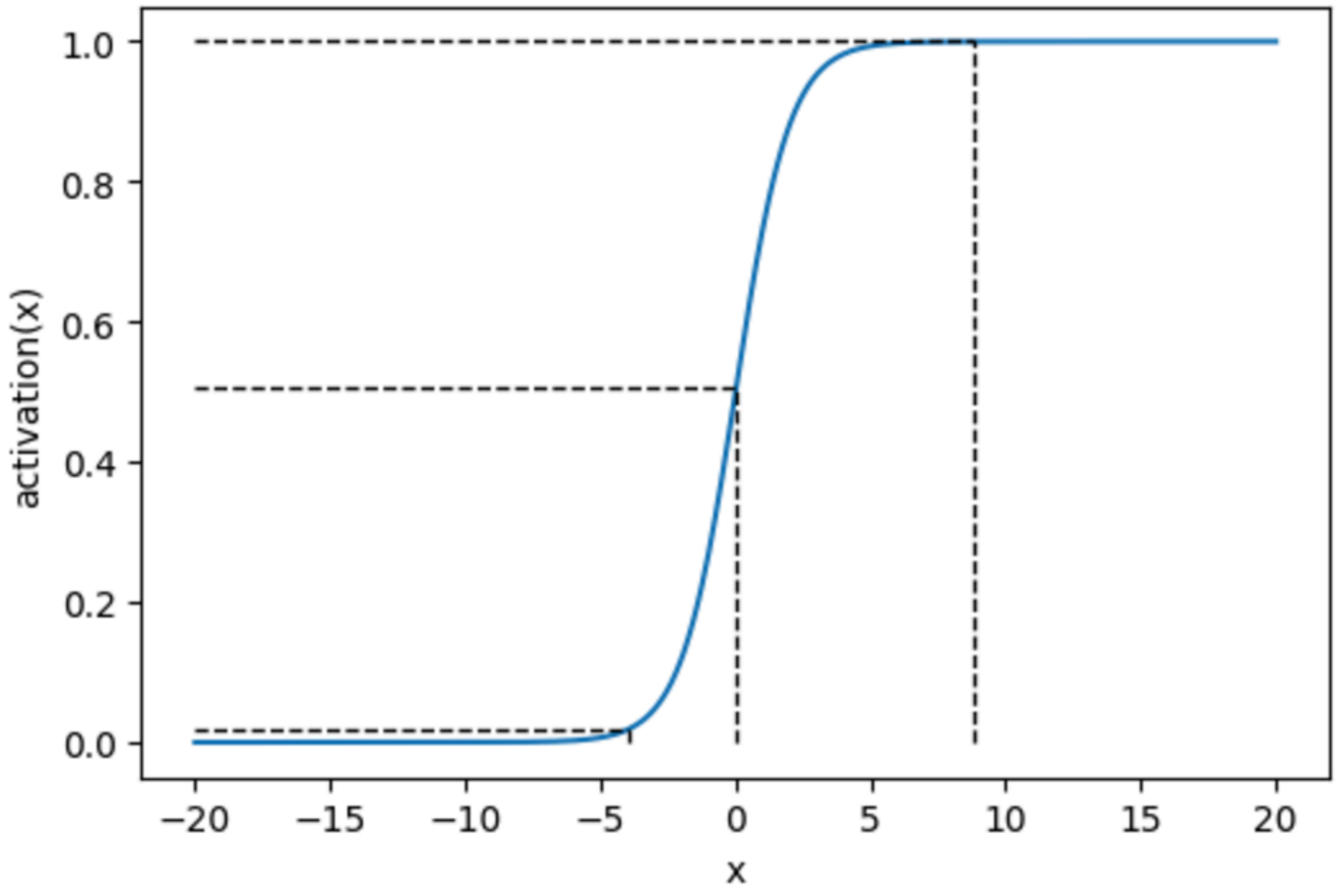
Some applications…

Starting simple: Logistic Regression



$$x = wx_0$$
$$y = activation(x)$$

Starting simple: Logistic Regression



$$x = wx_0$$
$$y = activation(x)$$

Starting simple: Logistic Regression

X (inputs)			y(outputs)
0	0	1	0
0	1	1	0
1	0	1	1
1	1	1	1

This is a linear problem!

Starting simple: Logistic Regression

$$Loss = \frac{1}{2}(f - y)^2$$

$$f = \phi(XW)$$

$$\frac{\vartheta Loss}{\vartheta W} = \underbrace{(y - f)}_{\epsilon} \frac{\vartheta \phi(XW)}{\vartheta W}$$

Starting simple: Logistic Regression

$$Loss = \frac{1}{2}(f - y)^2$$

$$f = \phi(XW)$$

$$\frac{\vartheta Loss}{\vartheta W} = \underbrace{(y - f)}_{\epsilon} \frac{\vartheta \phi(XW)}{\vartheta W}$$

...Go to notebook...

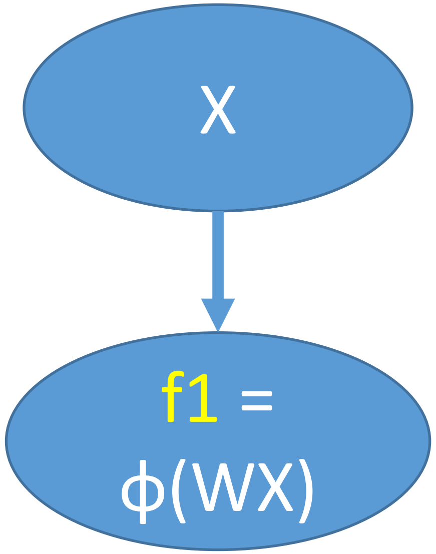
Starting simple: Logistic Regression

$$Loss = \frac{1}{2}(f - y)^2$$

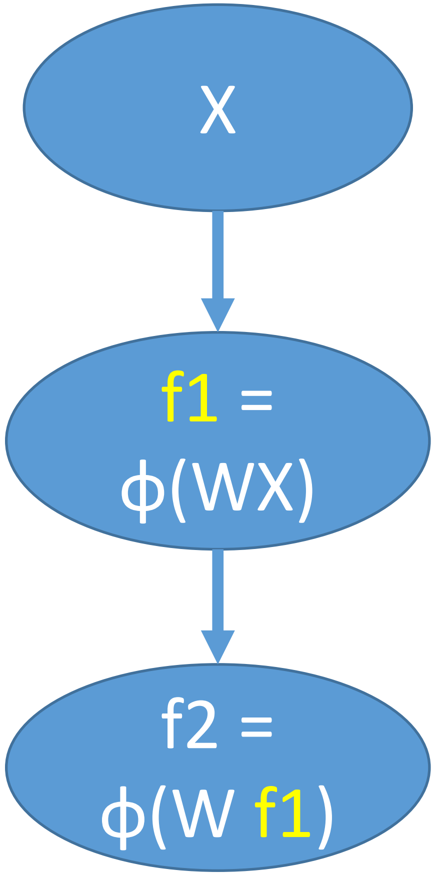
$$f = \phi(XW)$$

$$\frac{\vartheta Loss}{\vartheta W} = \underbrace{(y - f)}_{\epsilon} \frac{\vartheta \phi(XW)}{\vartheta W}$$

Logistic Regression



Deep Neural Network



Logistic Regression

$$Loss = \frac{1}{2}(y - f)^2$$

$$f = \phi(XW)$$

$$\frac{\vartheta Loss}{\vartheta W} = \underbrace{(y - f)}_{\epsilon} \frac{\vartheta \phi(XW)}{\vartheta W}$$

Deep neural network

$$Loss = \frac{1}{2}(y - f_2)^2$$

$$f_2 = \phi\left[\underbrace{\phi(XW_0)}_{f_1} W_1\right]$$

$$\frac{\vartheta Loss}{\vartheta W_0} = \cdots$$

$$\frac{\vartheta Loss}{\vartheta W_1} = \cdots$$

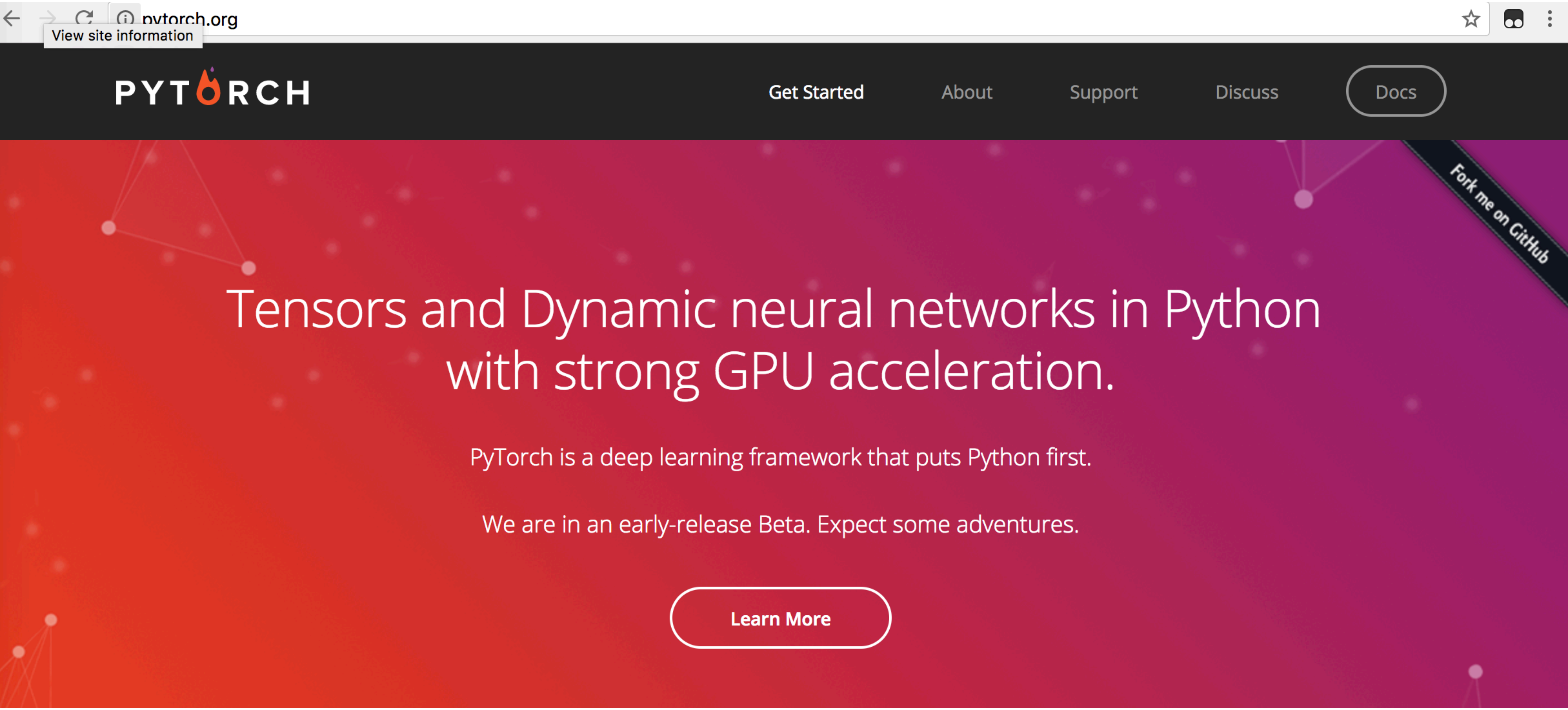
$$\frac{\vartheta (f_2 - y)^2 \frac{1}{2}}{\vartheta W_1} = -2\frac{1}{2}(f_2 - y)\frac{\vartheta f_2}{\vartheta W1} =$$

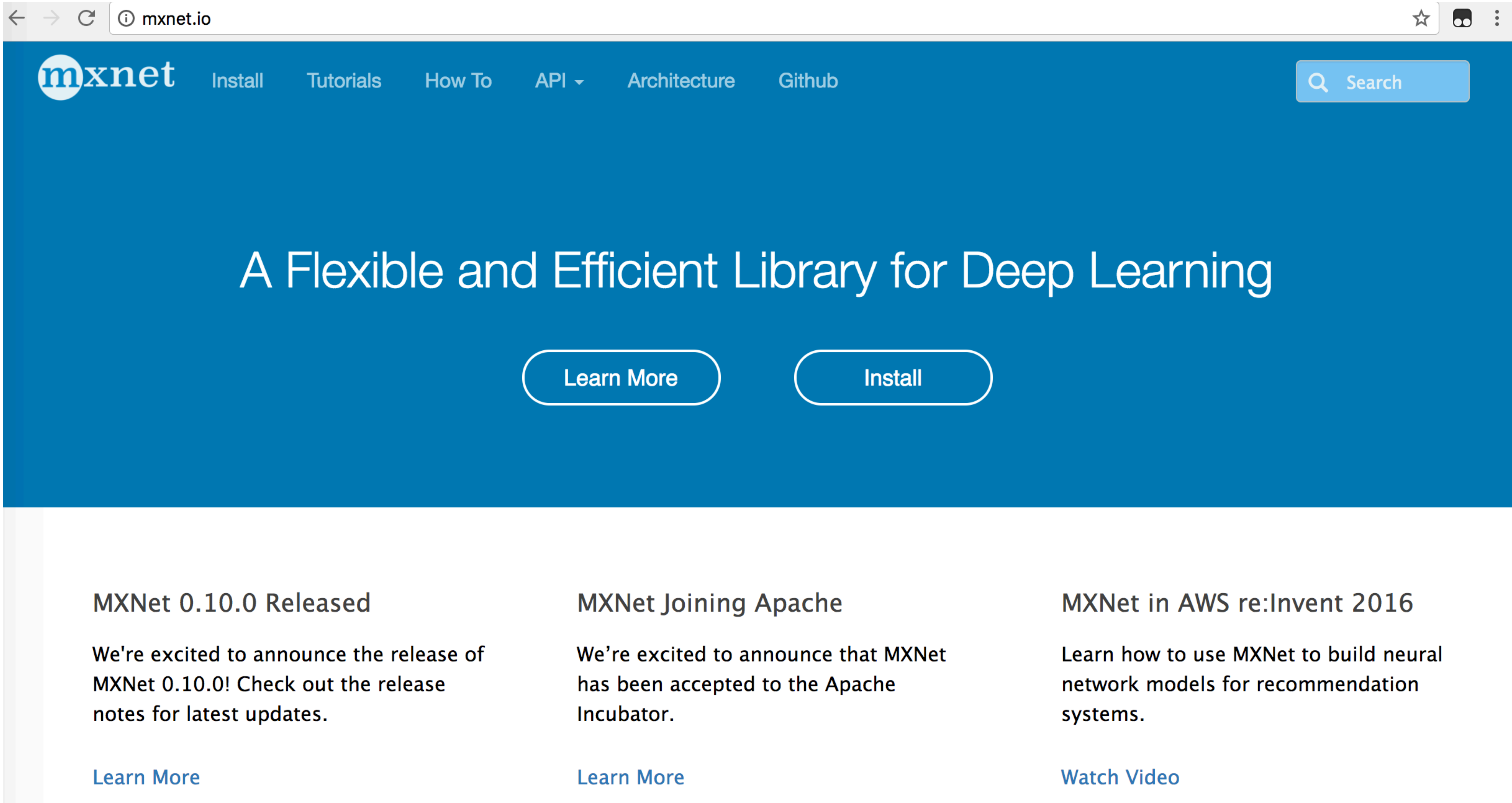
$$= (y - f_2)\frac{\vartheta \phi(f_1 W_1)}{\vartheta f_1 W_1}\frac{\vartheta f_1 W_1}{\vartheta W_1} =$$

$$= (y - f_2)\frac{\vartheta \phi f_1 W_1}{\vartheta W_1} =$$

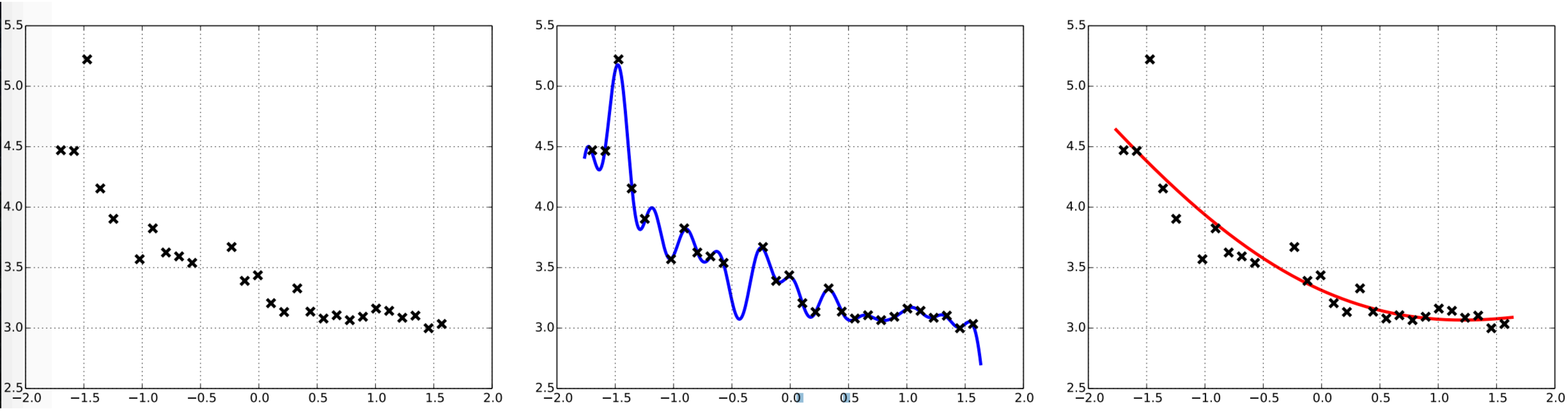
$$= \underbrace{(y - f_2)}_{\epsilon_2}\underbrace{\frac{\vartheta \phi f_1 W_1}{\vartheta W_1}}_{g_2} f_1^T$$

$$
\begin{aligned}
\frac{\vartheta (f_2 - y)^2}{\vartheta W_0} &= -2\frac{1}{2}(f_2 - y)\frac{\vartheta f_2}{\vartheta W_0} = \\
&= (y - f_2)\frac{\vartheta \phi(f_1 W_1)}{\vartheta f_1 W_1}\frac{\vartheta f_1 W_1}{\vartheta f_1}\frac{\vartheta f_1}{\vartheta W_0} = \\
&= \epsilon_2 \; g_2 \; W_1^T \; \frac{\vartheta \phi(X W_0)}{X W_0}\frac{\vartheta X W_0}{\vartheta W_0} = \\
&= \epsilon_2 \; g_2 \; W1^T \frac{\vartheta \phi(X W_0)}{X W_0} \; X^T
\end{aligned}
$$


pytorch.org
View site information
PYTORCH
Get Started
About
Support
Discuss
Docs
Fork me on GitHub
Tensors and Dynamic neural networks in Python with strong GPU acceleration.
PyTorch is a deep learning framework that puts Python first.
We are in an early-release Beta. Expect some adventures.
Learn More


mxnet.io
mxnet
Install
Tutorials
How To
API
Architecture
Github
Search
A Flexible and Efficient Library for Deep Learning
Learn More
Install
MXNet 0.10.0 Released
We're excited to announce the release of MXNet 0.10.0! Check out the release notes for latest updates.
Learn More
MXNet Joining Apache
We're excited to announce that MXNet has been accepted to the Apache Incubator.
Learn More
MXNet in AWS re:Invent 2016
Learn how to use MXNet to build neural network models for recommendation systems.
Watch Video

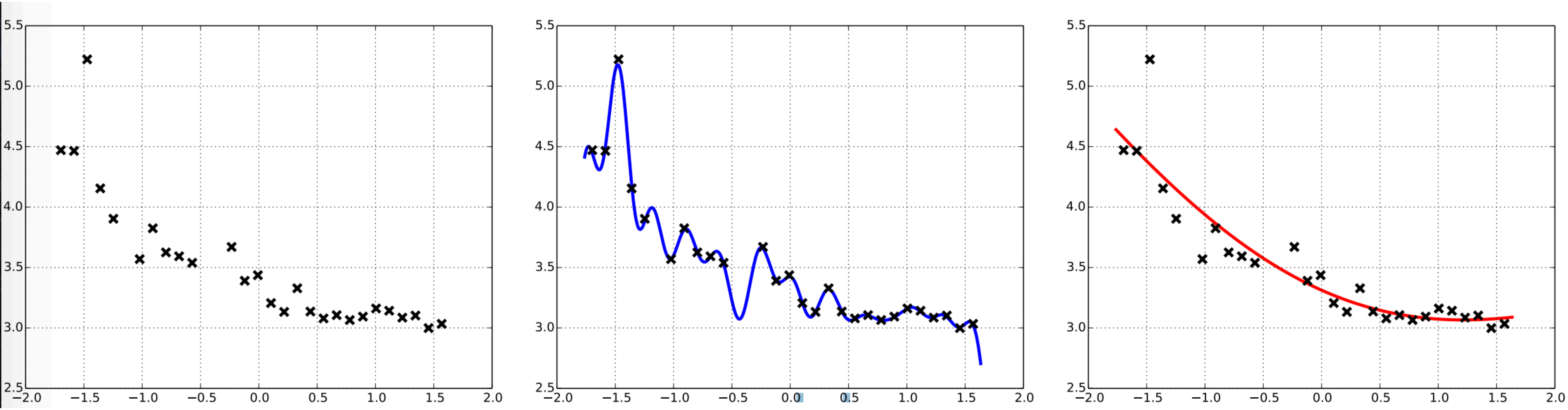
Overfitting



(a) (b) (c)

Which of the two curves (b) or (c) are better models for training data shown in (a)?

Overfitting



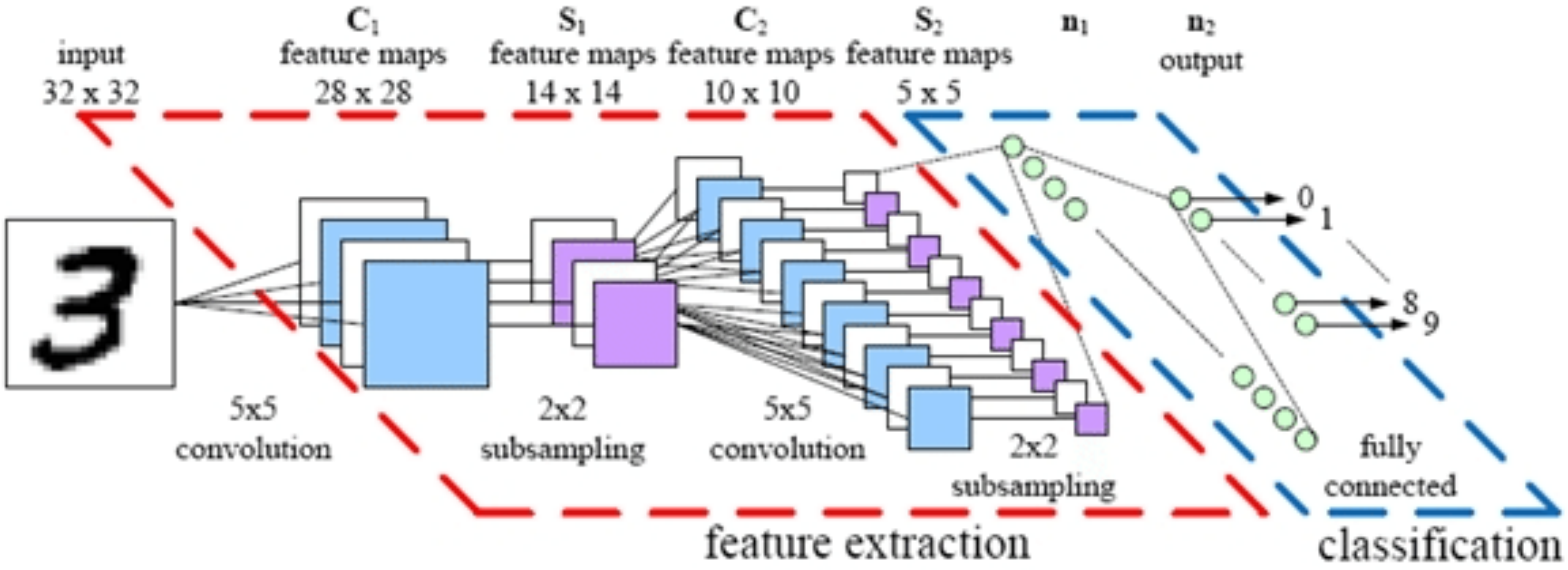
(a) (b) (c)

The middle picture (b) interpolates EVERY training point.
Does that make it the best model?

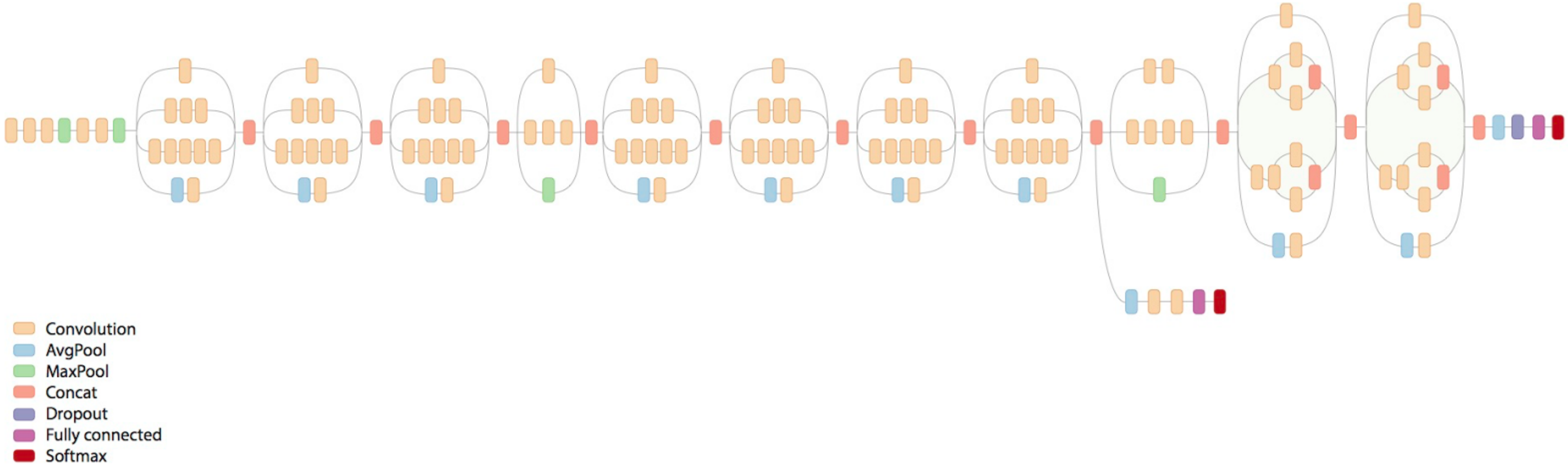
- We can consider also different activations and “wiring” of the network. We can combine networks. And more. As long as everything remains differentiable.

- http://playground.tensorflow.org/

- There are various techniques to improve optimization:
    - Early stopping (prevent overfitting)
    - Dropout
    - Adaptive learning rates
    - ….



input
32 x 32
C1
feature maps
28 x 28
S1
feature maps
14 x 14
C2
feature maps
10 x 10
S2
feature maps
5 x 5
n1
n2
output
5x5
convolution
2x2
subsampling
5x5
convolution
2x2
subsampling
fully
connected
0
1
8
9
feature extraction
classification



Inception-v3

Image classification

Labradoodle or fried chicken

Image from: Yangyan Li

Puppy or bagel

Image from: Yangyan Li

Sheepdog or mop

Image from: Yangyan Li

Chihuahua or muffin

Image from: Yangyan Li

Barn owl or apple

Image from: Yangyan Li

Parrot or guacamole

Image from: Yangyan Li

Raw chicken or Donald Trump

Image from: Yangyan Li

Image Recognition: beyond binary classification

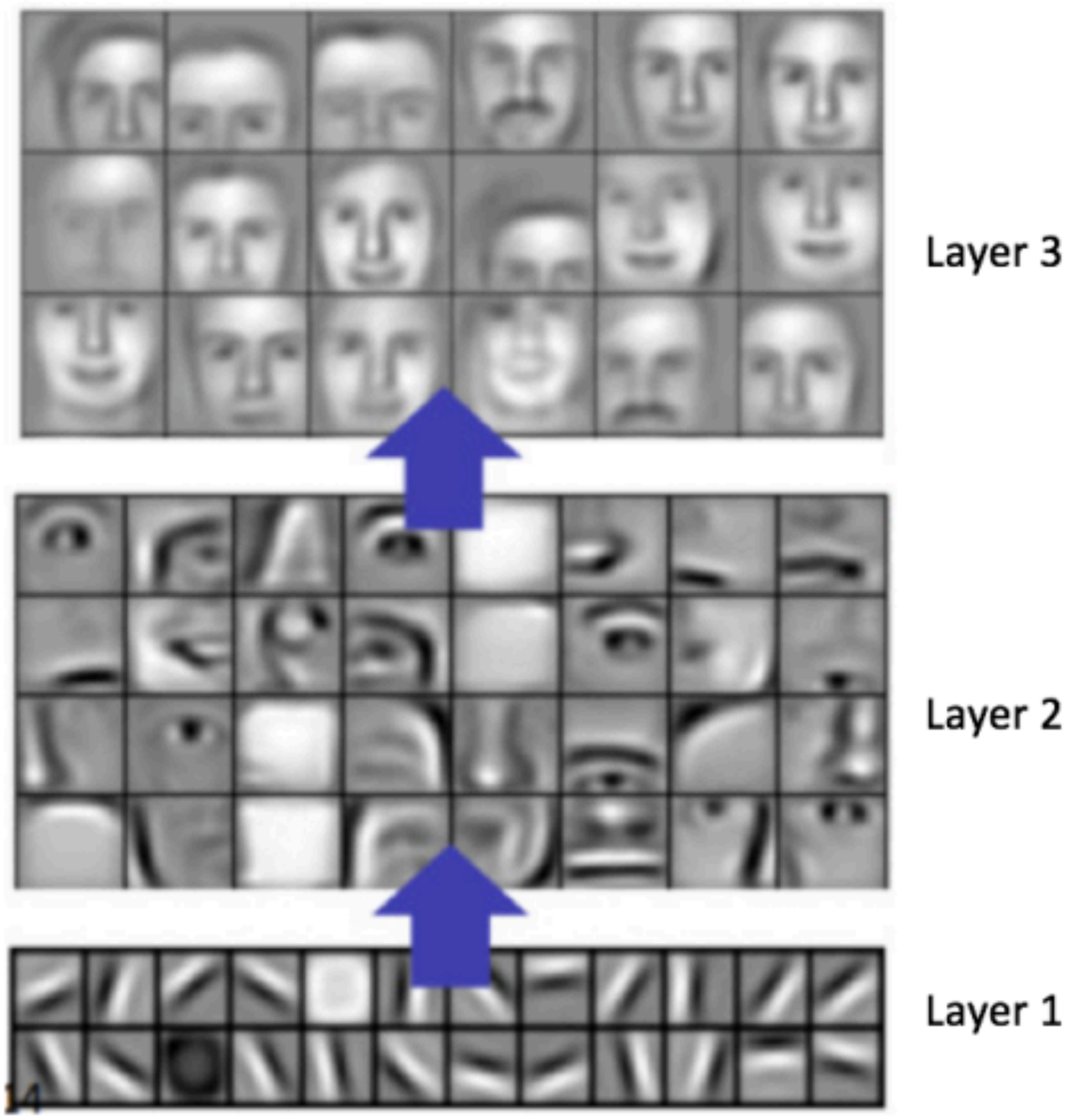
Menu | amazon web services | Products | Solutions | Pricing | Software | More | English | My Account | Sign In to the Console

Object and Scene Detection

Rekognition identifies thousands of objects such as vehicles, pets, or furniture, and provides a confidence score. Rekognition also detects scenes within an image, such as a sunset or beach. This makes it easy for you to add features that search, filter, and curate large image libraries.

Convolutional Neural Network

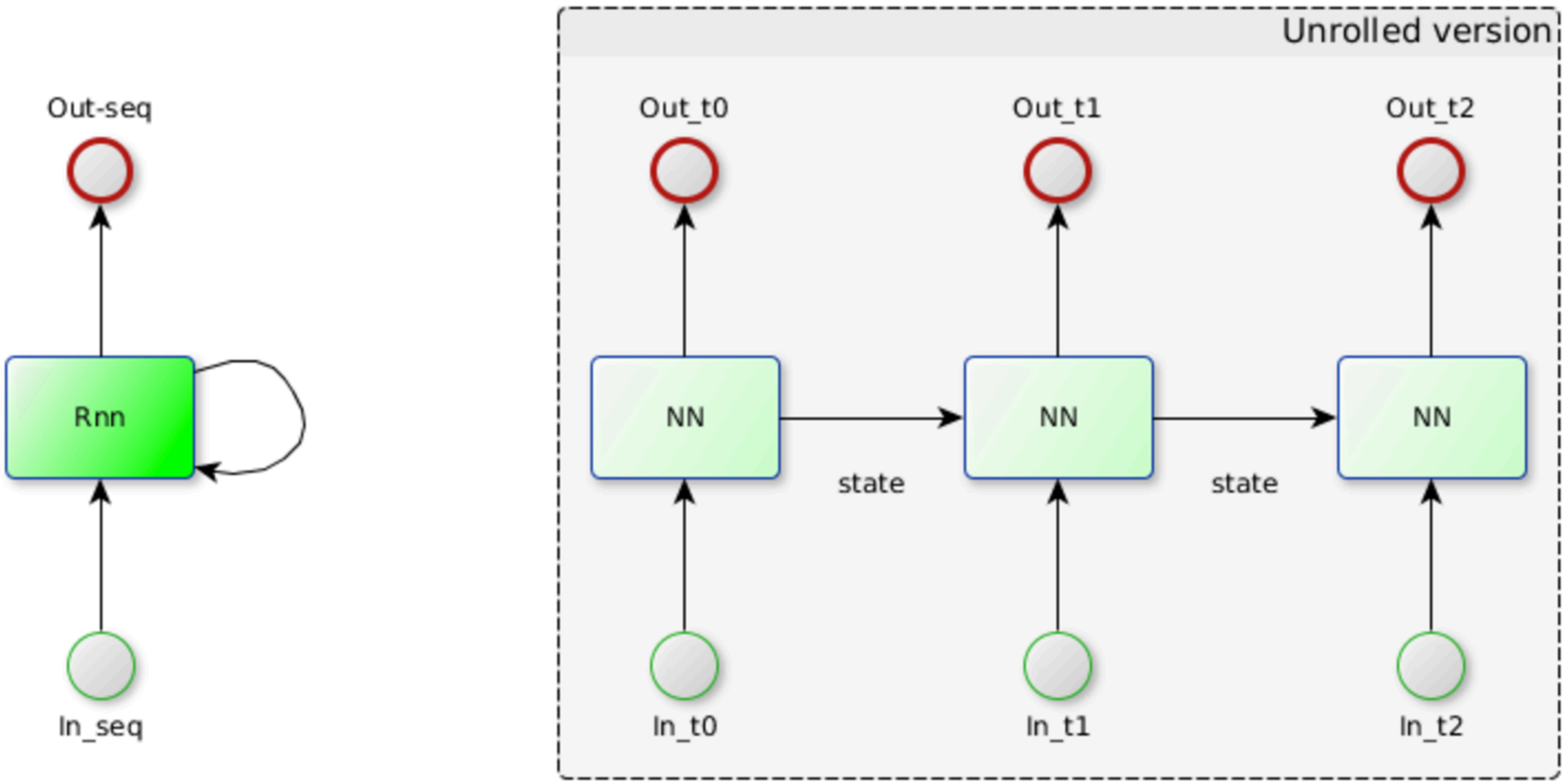

Input



ujjwalkarn.me

Measuring pollution in Kampala (Mike Smith)

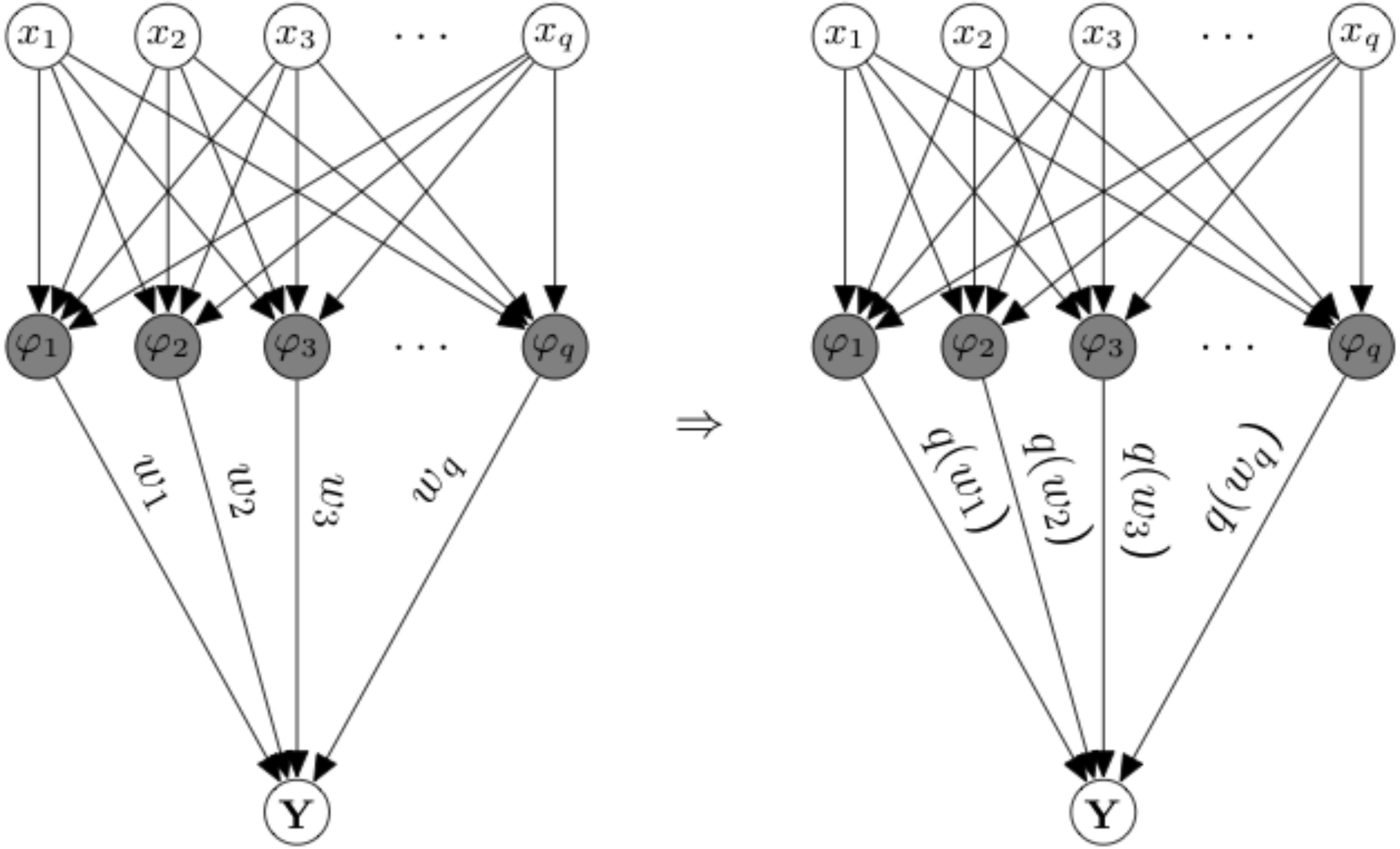
Sequential modeling: Recurrent neural networks


Out-seq
Rnn
In_seq
Unrolled version
Out_t0
Out_t1
Out_t2
NN
NN
NN
state
state
In_t0
In_t1
In_t2

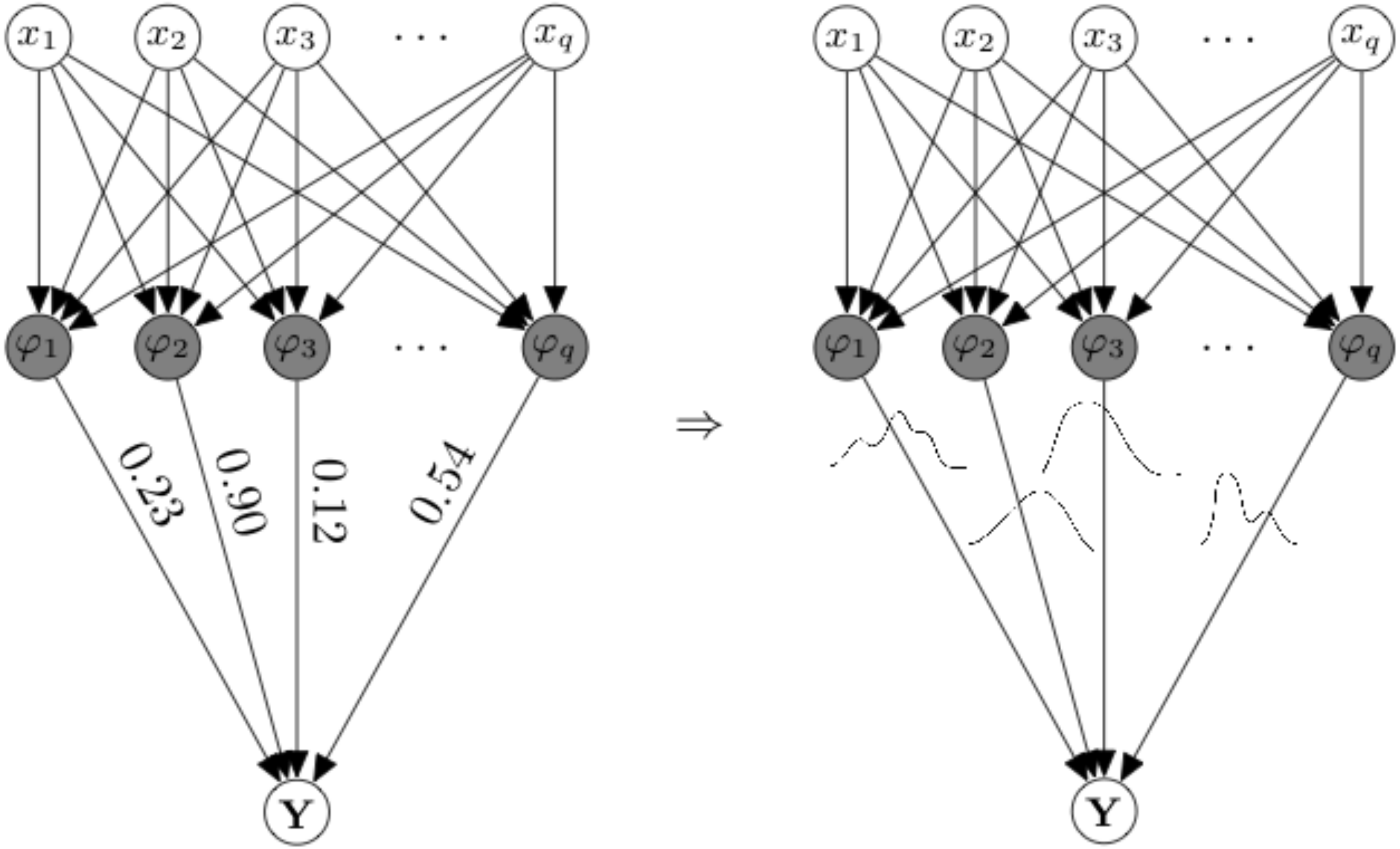
Need for uncertainty

- Reinforcement learning
- Critical predictive systems
- Active learning
- Semi-automatic systems
- Scarce data scenarios
- ...

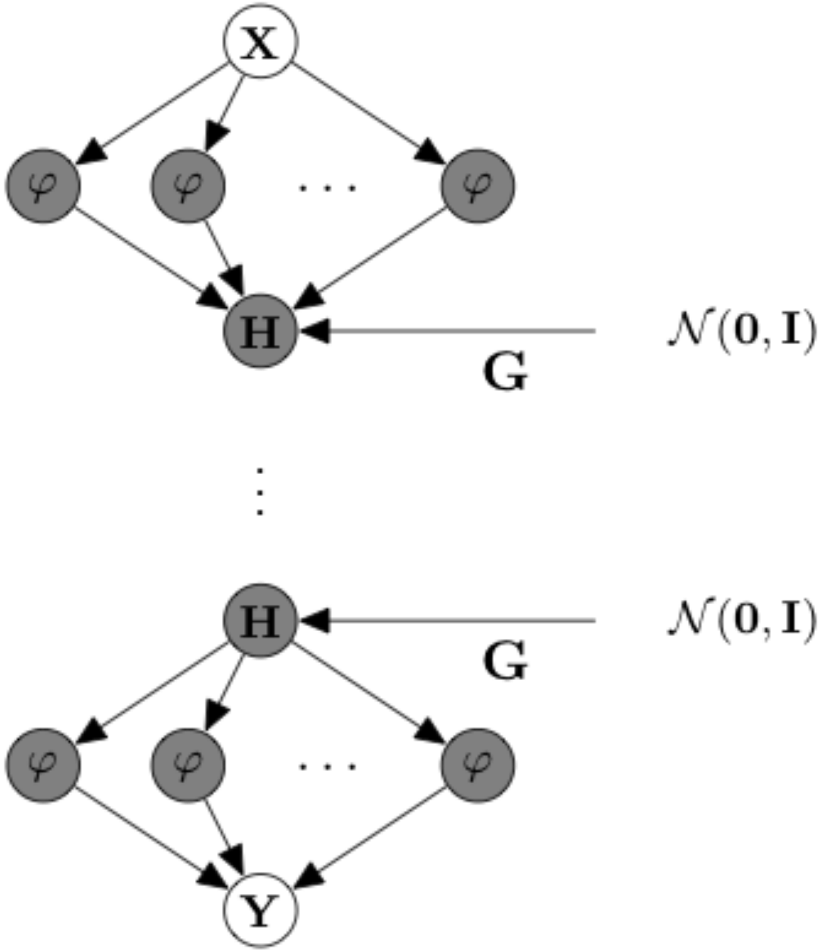
BNN with priors on its weights



BNN with priors on its weights



Stochastic warping



Inference:

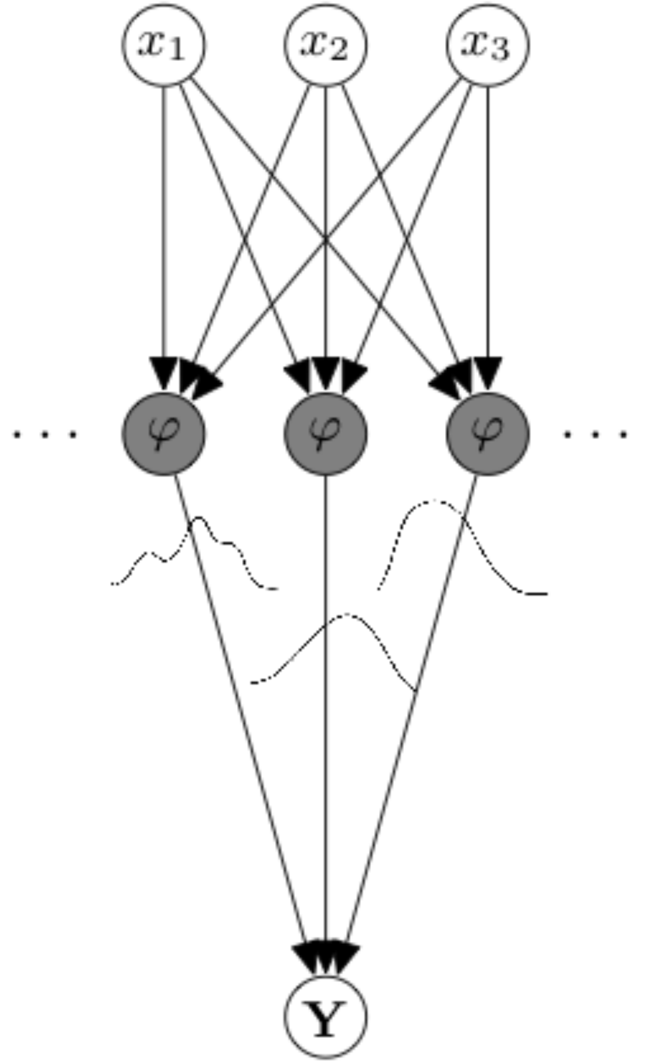
- Need to infer posteriors on $\mathbf{H}$
- Define $q(\mathbf{H})$ and proceed as before with VI/MC.

From NN to GP

- In the limit of infinite weights with a prior, we obtain a GP*.
- Think of a function as an infinite dimensional vector.

$$y = f(x) + \epsilon$$

$f \sim \mathcal{GP}(0, k(x, x'))$. f is *stochastic*!



Conclusions…

Deep learning is cool and gives you great power…

…but is not a solution to everything…

… and please don't conflate Deep Learning with Machine Learning

This makes machine learners sad ☹

Deep Training yourself

- For now: continue working on the Jupyter notebook we saw today.
  - Extend for arbitrary depth
  - Add a bias in the activation
  - Play with different parameterizations
  - Put it on the side and re-implement it!
- Be aware of caveats
- Watch online videos (this included)
- Also plenty of blogs
- Learn mxnet, pytorch, tensorflow, ….