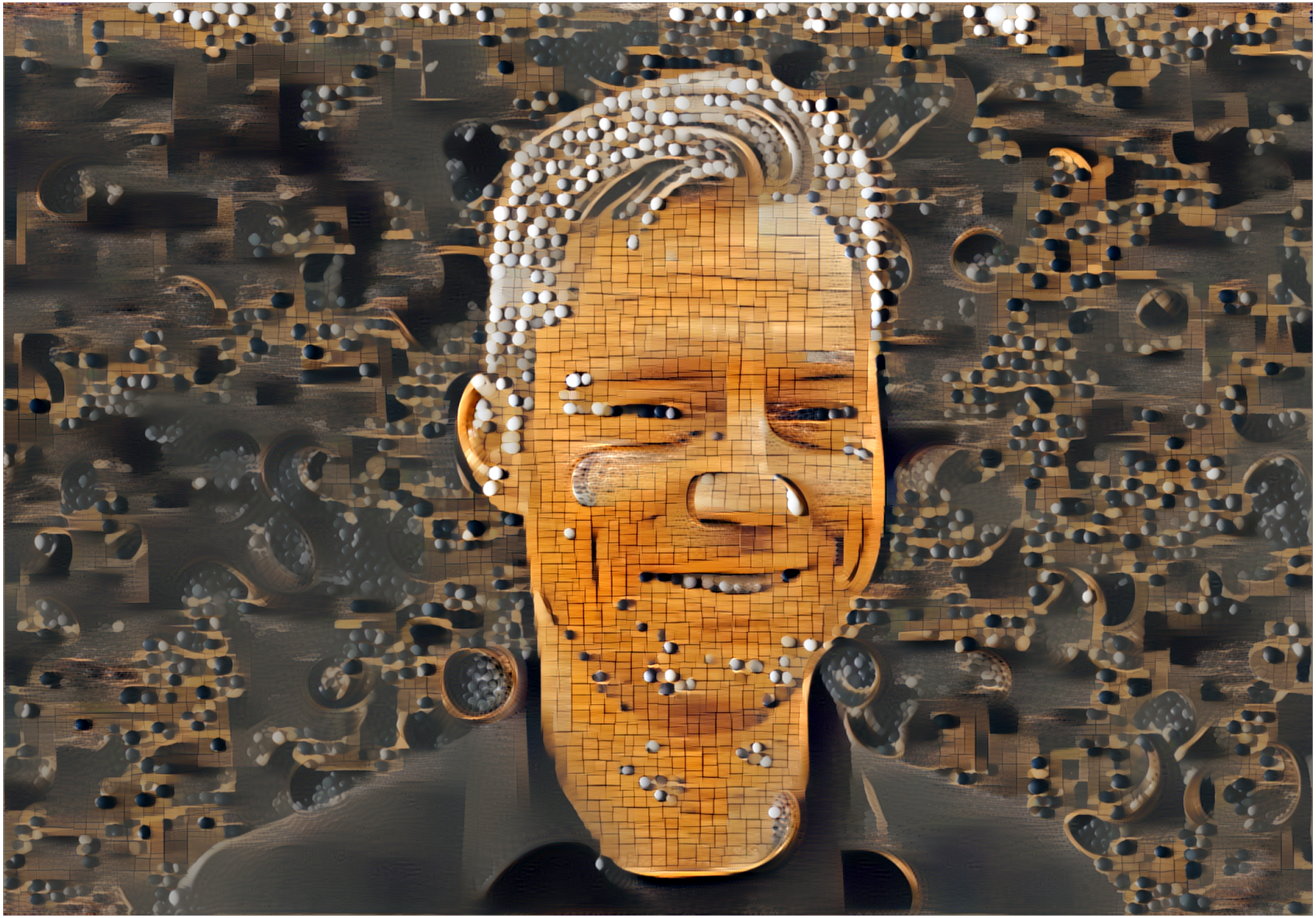
Deep learning summer camp
1 Introduction

Ole Winther

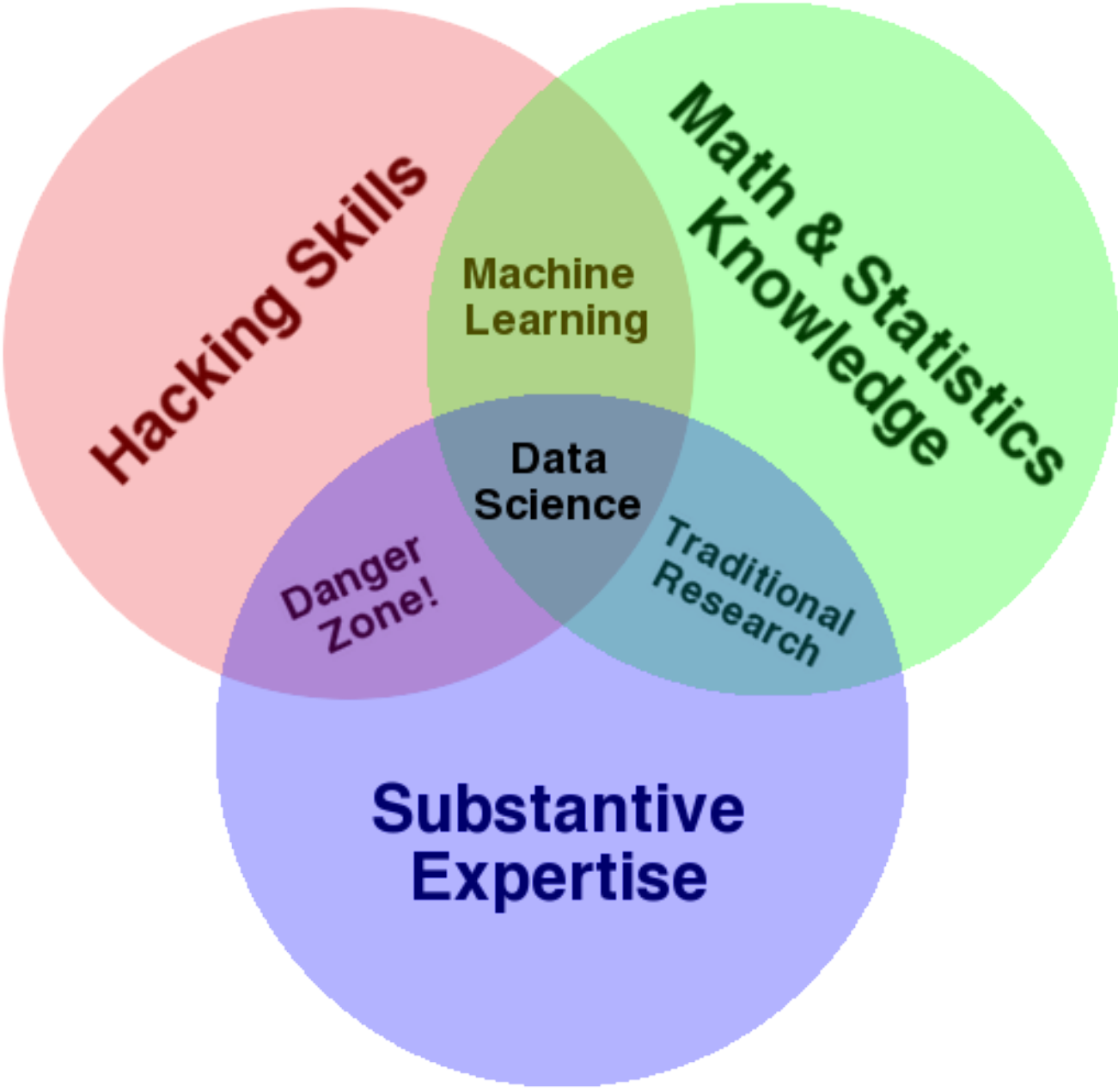
Dept for Applied Mathematics and Computer Science
Technical University of Denmark (DTU)

July 4, 2016

Ole Winther - a bit about myself



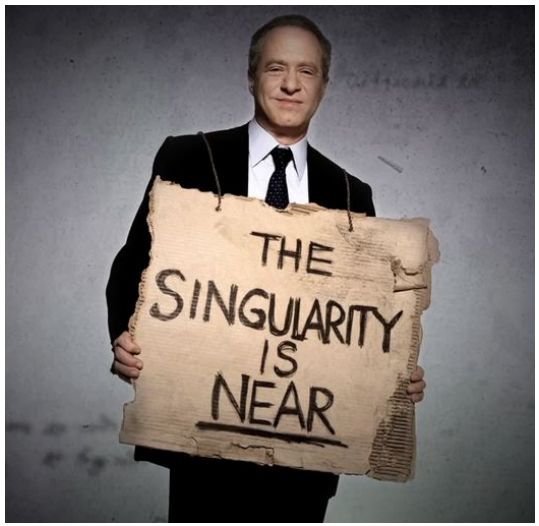
Data science - adding domain knowledge



Objectives of talk

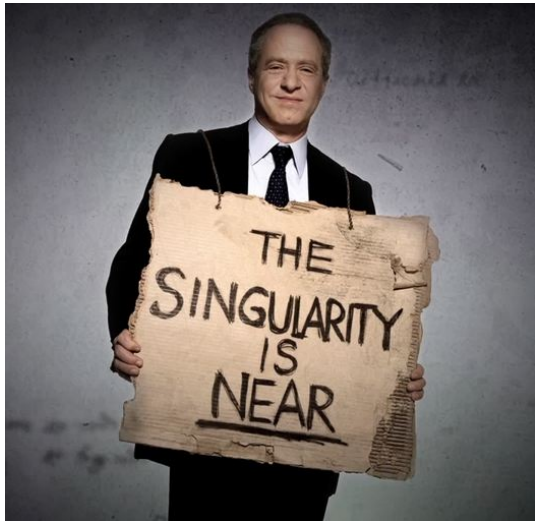
- What is deep learning and statistical artificial intelligence?
- How does it work?
- The feed-forward neural network (FFNN)

Are we heading towards the singularity?



kurzweilai.net

Are we heading towards the singularity?

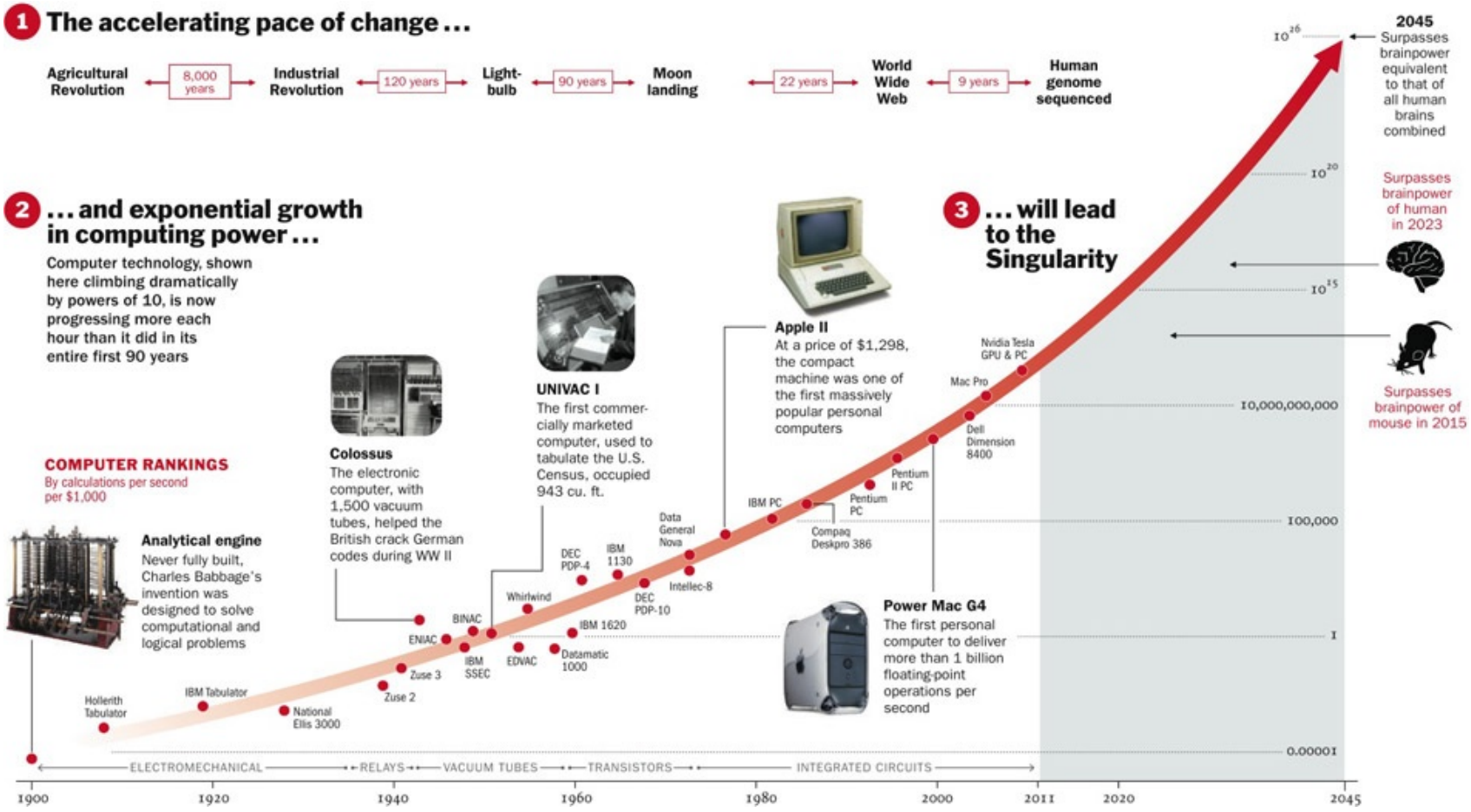


kurzweilai.net

- Elon Musk at MIT AeroAstro Symp:
- If I were to guess at what our biggest existential threat is, it's probably that...
- With artificial intelligence, we are summoning the demon..
- Inofficial quotes (email to friend):
- The risk of something seriously dangerous happening is in the five year timeframe. 10 years at most,
- Unless you have direct exposure to groups like Deepmind, you have no idea how fast — it is growing at a pace close to exponential.

Reinforcement learning

Growth in computer power



Major areas in AI

- Speech recognition
- Image classification
- Machine translation
- Question-answering
- Self-driving vehicles
- Dialogue systems
- General unsupervised learning

Major areas in AI

- Speech recognition
- Image classification
- Machine translation
- Question-answering
- Self-driving vehicles
- Dialogue systems
- General unsupervised learning

Part 1:

The deep learning revolution

Achilles' heel of traditional AI: Perception in natural environment

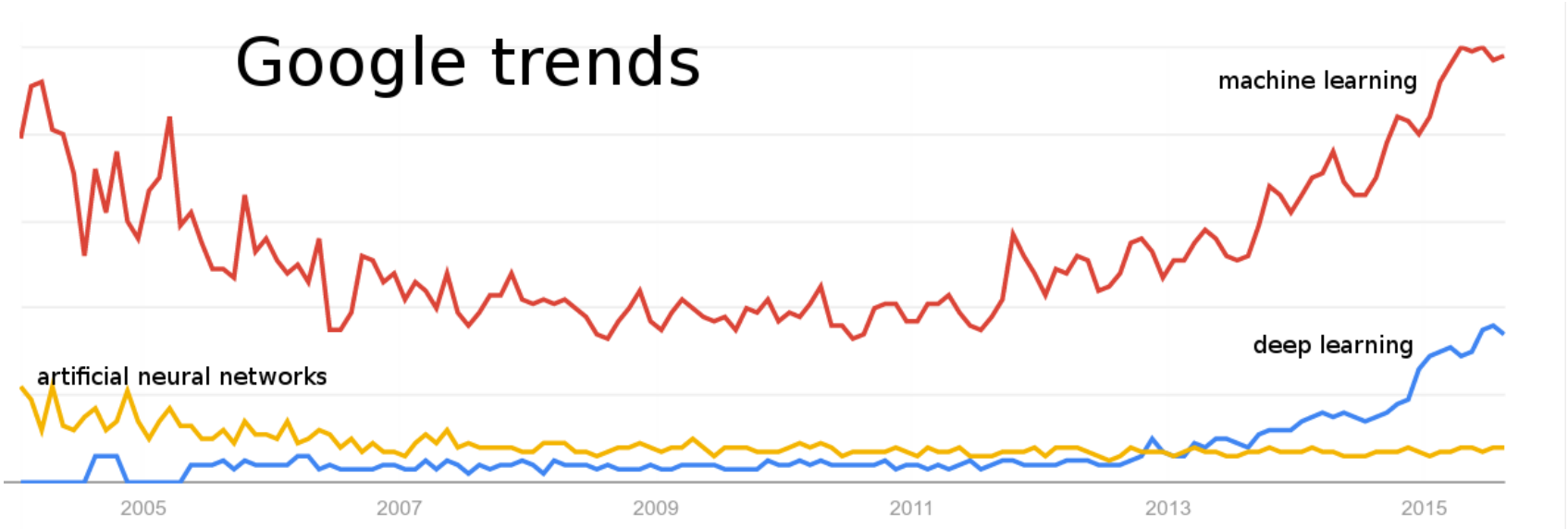
xkcd.com/1425

Many thanks to Tapani Raiko for sharing slides!

Deep learning is hot in academia

- ”Deep learning . . . dramatically improved the state-of-the-art in speech recognition, visual object recognition. . . ” (LeCun et al., Nature, 2015)
- ”. . . bridges the divide between high-dimensional sensory inputs and actions, resulting in the first artificial agent. . . ” (Mnih et al., Nature, 2015)
- ”Knowing the sequence specificities of DNA- and RNA-binding proteins is essential . . . deep learning outperforms other state-of-the-art methods” (Alipanahi et al., Nature Biotechnology, 2015)
- ”This is the first time that a computer program has defeated a human professional payer in the full-sized game of Go” (Silver et al, Nature 2016)

Deep learning is hot in industry



Google acquired startup DeepMind for $500M in 2014.
Also racing: Facebook, Baidu, IBM, Amazon, Samsung, Nvidia, Apple, Nokia, . . .

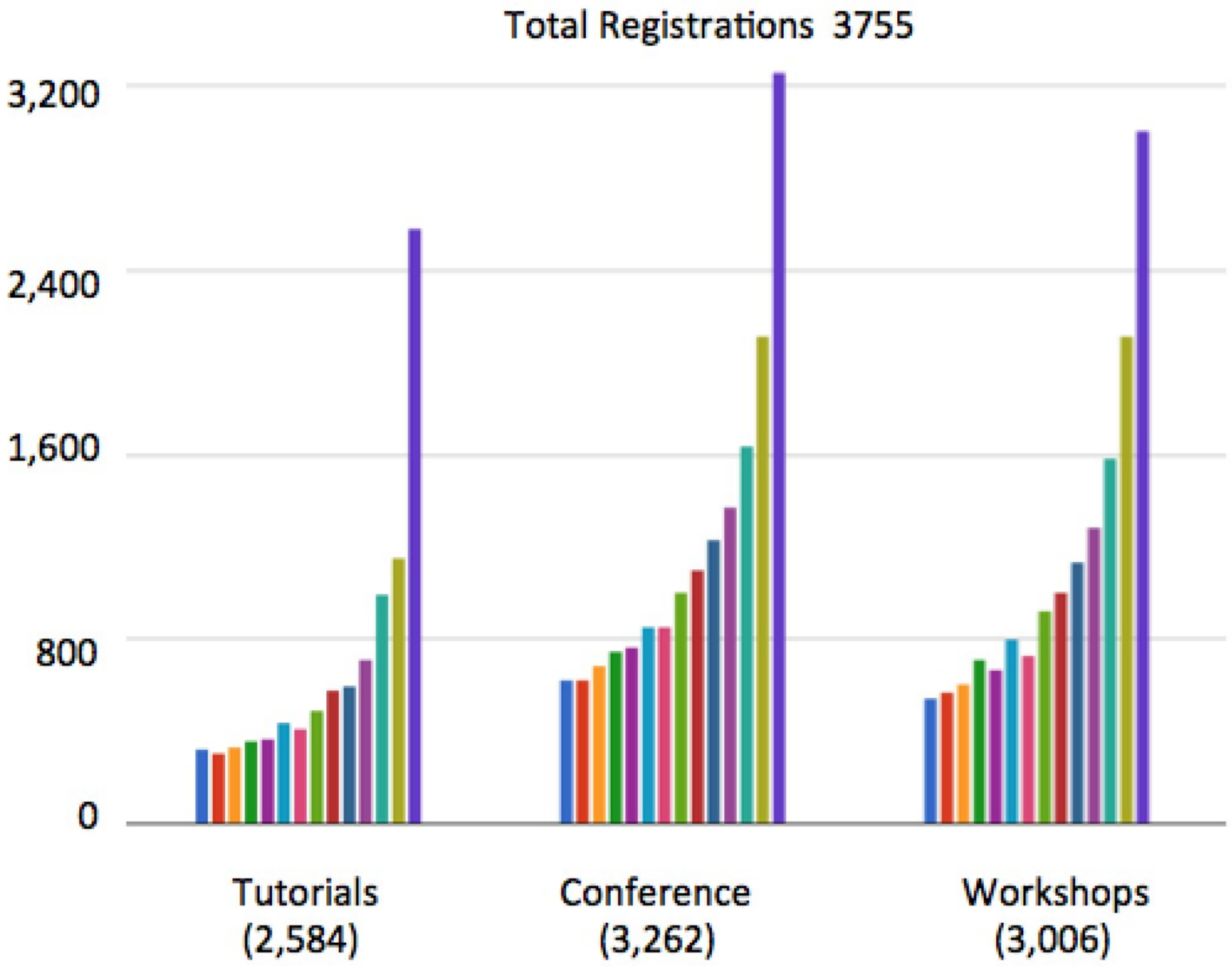
On the cover of The Economist

Economist 2nd July 2016

On the cover of The Economist

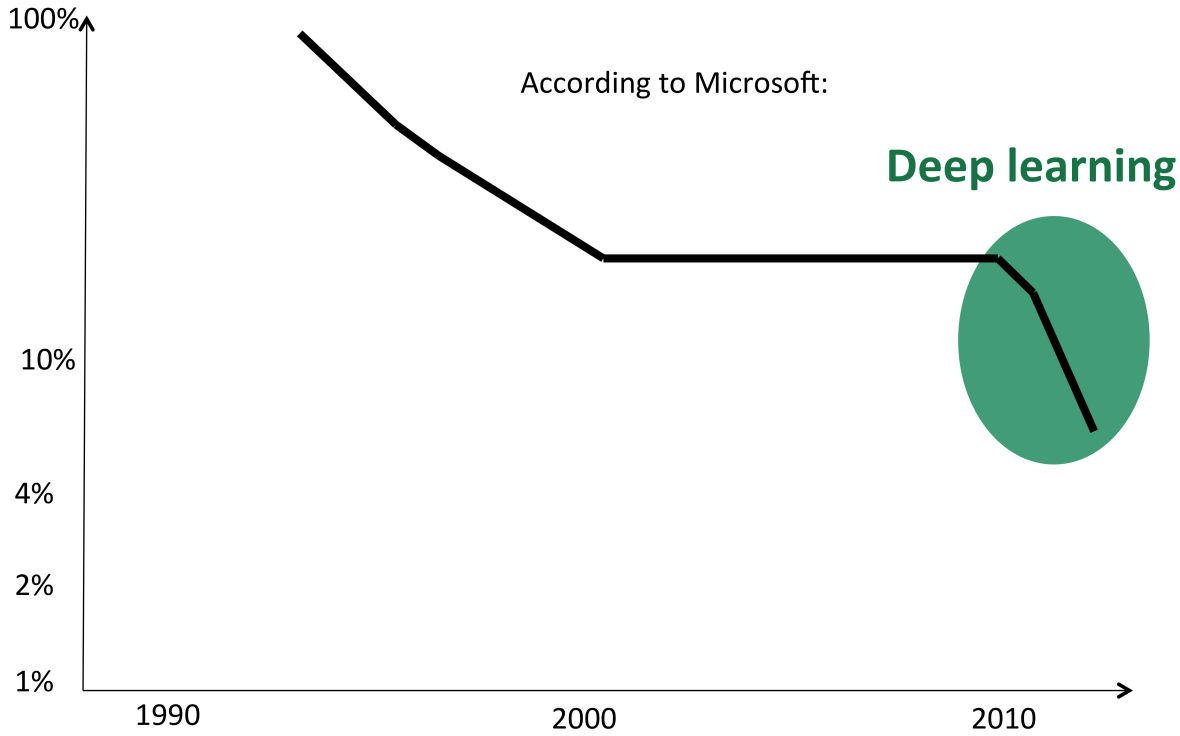
Economist 25th June 2016

NIPS Growth


Total Registrations 3755
3,200
2,400
1,600
800
0
Tutorials
(2,584)
Conference
(3,262)
Workshops
(3,006)

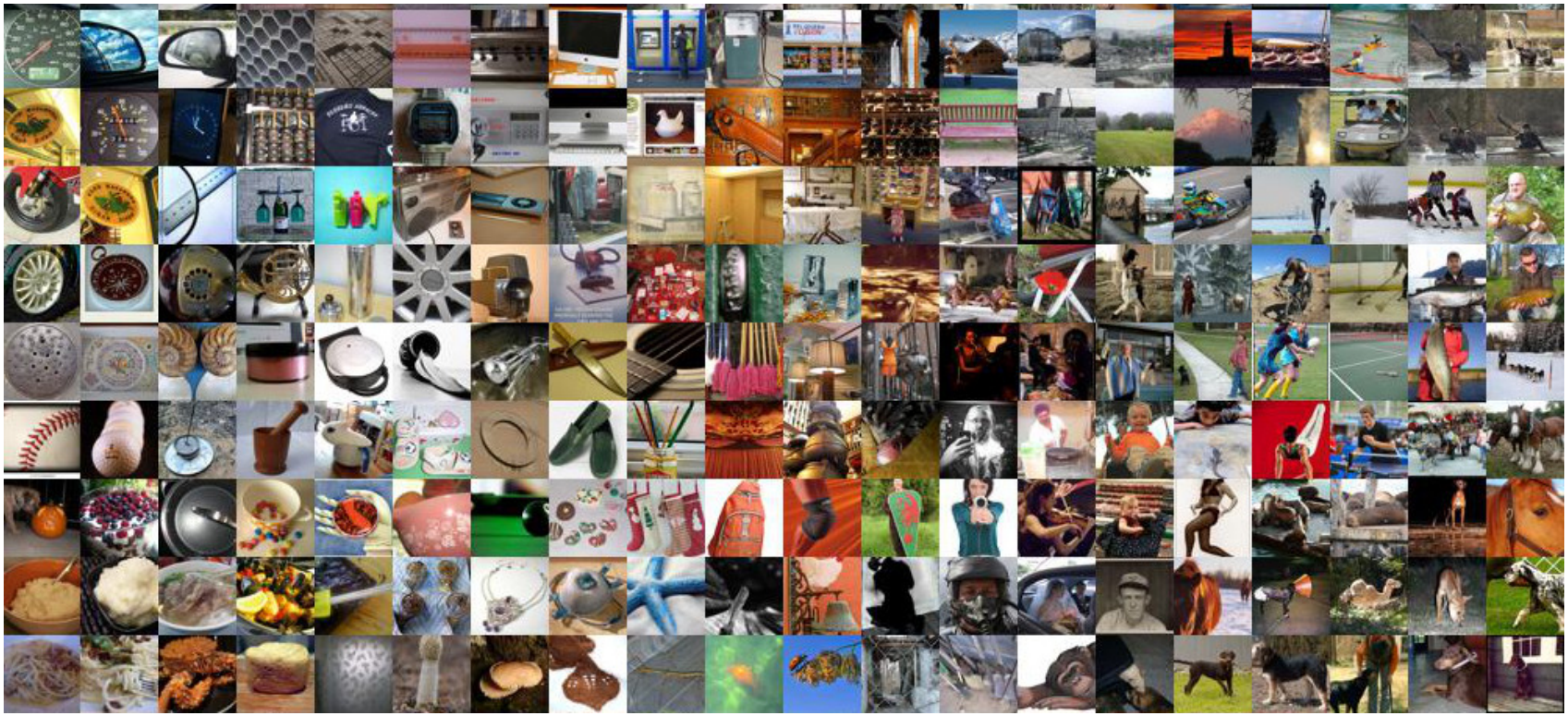
Deep learning is changing the world

Speech recognition breakthrough



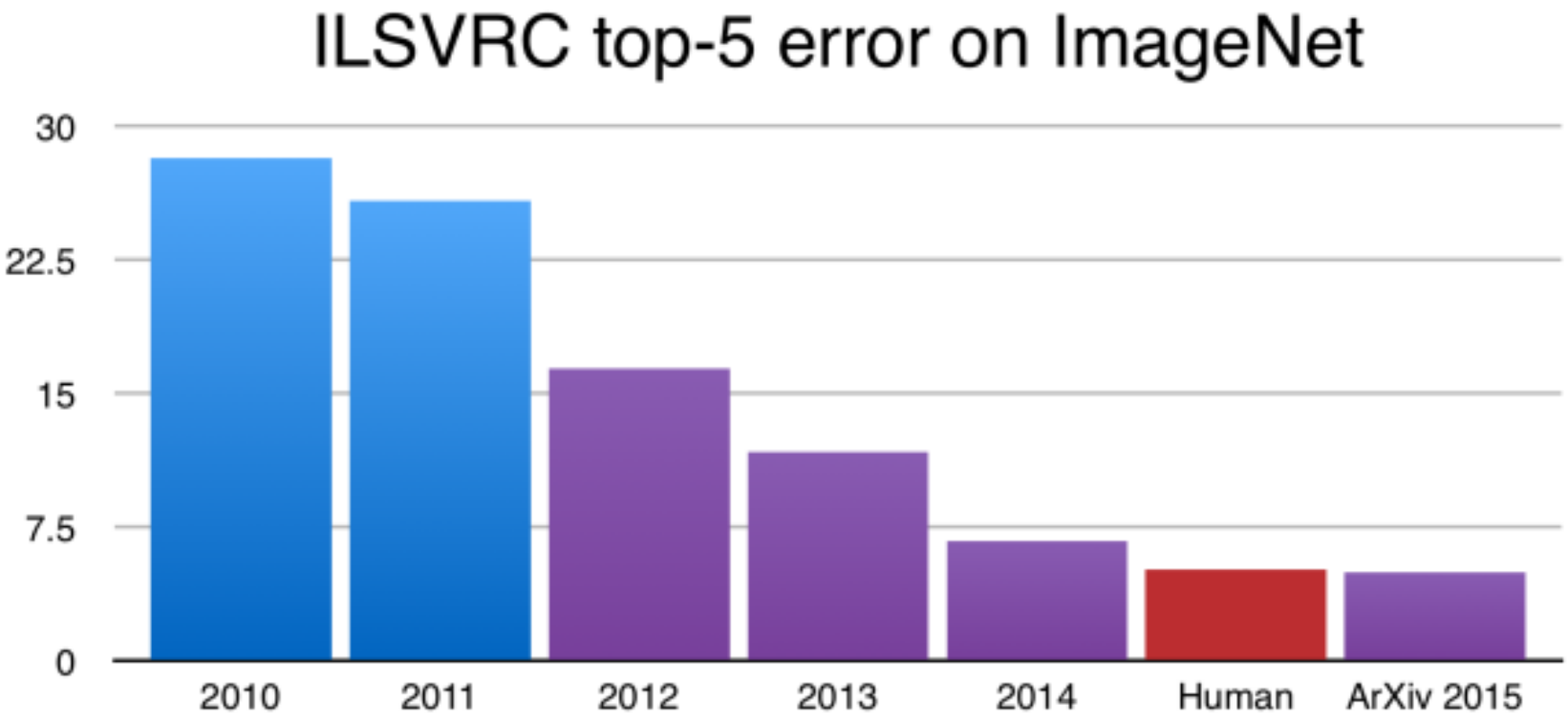
Plot from Yoshua Bengio

Imagenet classification challenge



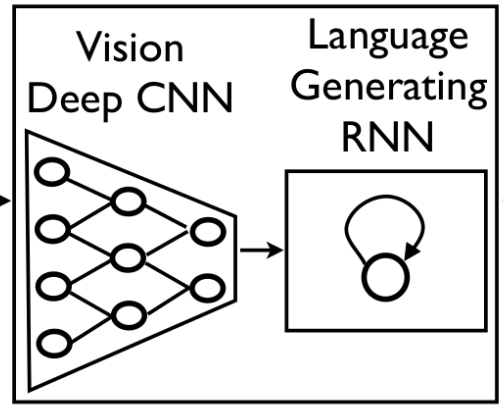
Annual competition in computer vision.

Imagenet classification challenge



Krizhevsky et al. (2012) won with huge margin
(16.4% error compared to 26.2%) by deep learning.
Soon everyone started using deep learning and GPUs.

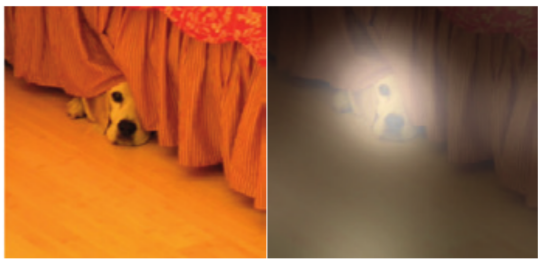
Caption generation (Xu et al., 2015)



A group of people shopping at an outdoor market.

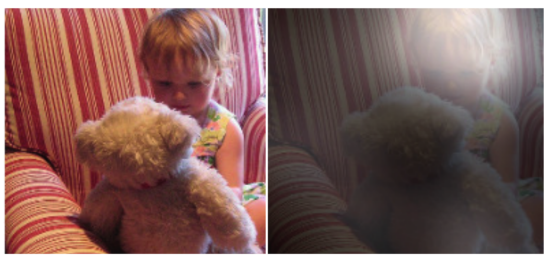
There are many vegetables at the fruit stand.

A woman is throwing a **frisbee** in a park.



A **dog** is standing on a hardwood floor.

A **stop** sign is on a road with a mountain in the background

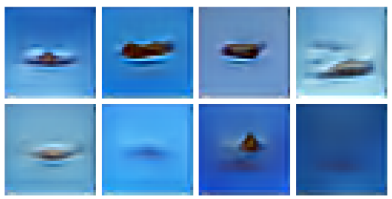


A little **girl** sitting on a bed with a teddy bear.

A group of **people** sitting on a boat in the water.

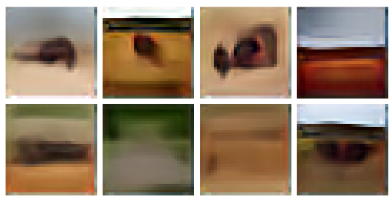
A giraffe standing in a forest with **trees** in the background.

Text to images (Mansimov et al., 2015)

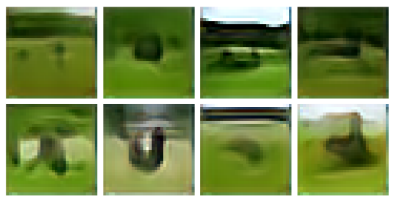


A very large commercial plane flying in blue skies.

A very large commercial plane flying in rainy skies.



A herd of elephants walking across a dry grass field.



A herd of elephants walking across a green grass field.

Modifying visual features (Larsen et al., 2015)

Representation learning

Traditional way:
Data $\rightarrow$ Feature engineering $\rightarrow$ Machine learning

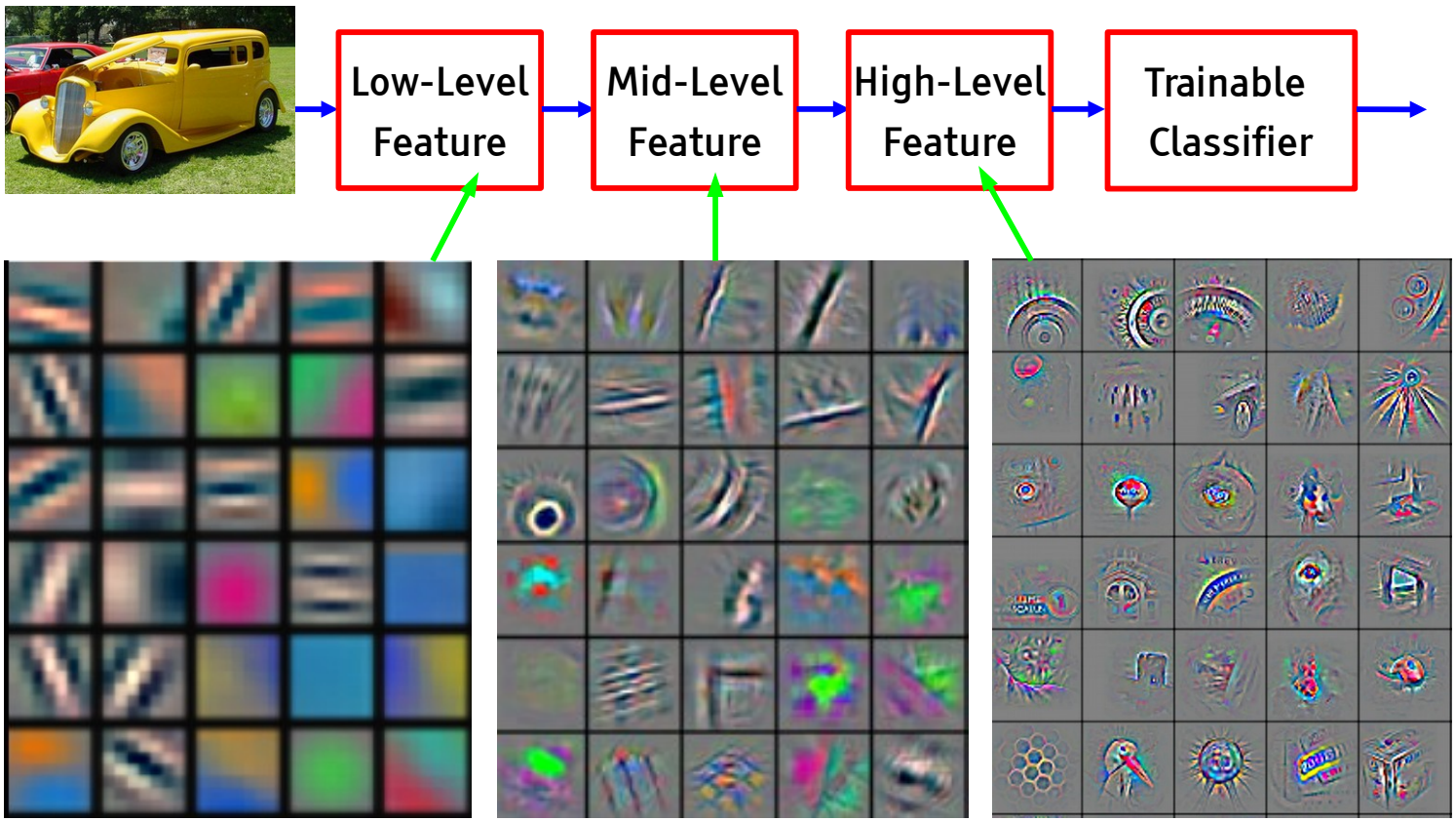
- Feature selection
- Feature extraction (e.g. PCA)
- Feature construction (e.g. SIFT)

Deep learning way:
Data $\rightarrow$ End-to-end learning

Deep Learning = Learning Hierarchical Representations

Y LeCun

- It's deep if it has more than one stage of non-linear feature transformation



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

- Review article, May 2015:

Deep learning

Yann LeCun[1,2], Yoshua Bengio[3] & Geoffrey Hinton[4,5]

nature

Deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. These methods have dramatically improved the state-of-the-art in speech recognition, visual object recognition, object detection and many other domains such as drug discovery and genomics. Deep learning discovers intricate structure in large data sets by using the backpropagation algorithm to indicate how a machine should change its internal parameters that are used to compute the representation in each layer from the representation in the previous layer. Deep convolutional nets have brought about breakthroughs in processing images, video, speech and audio, whereas recurrent nets have shone light on sequential data such as text and speech.

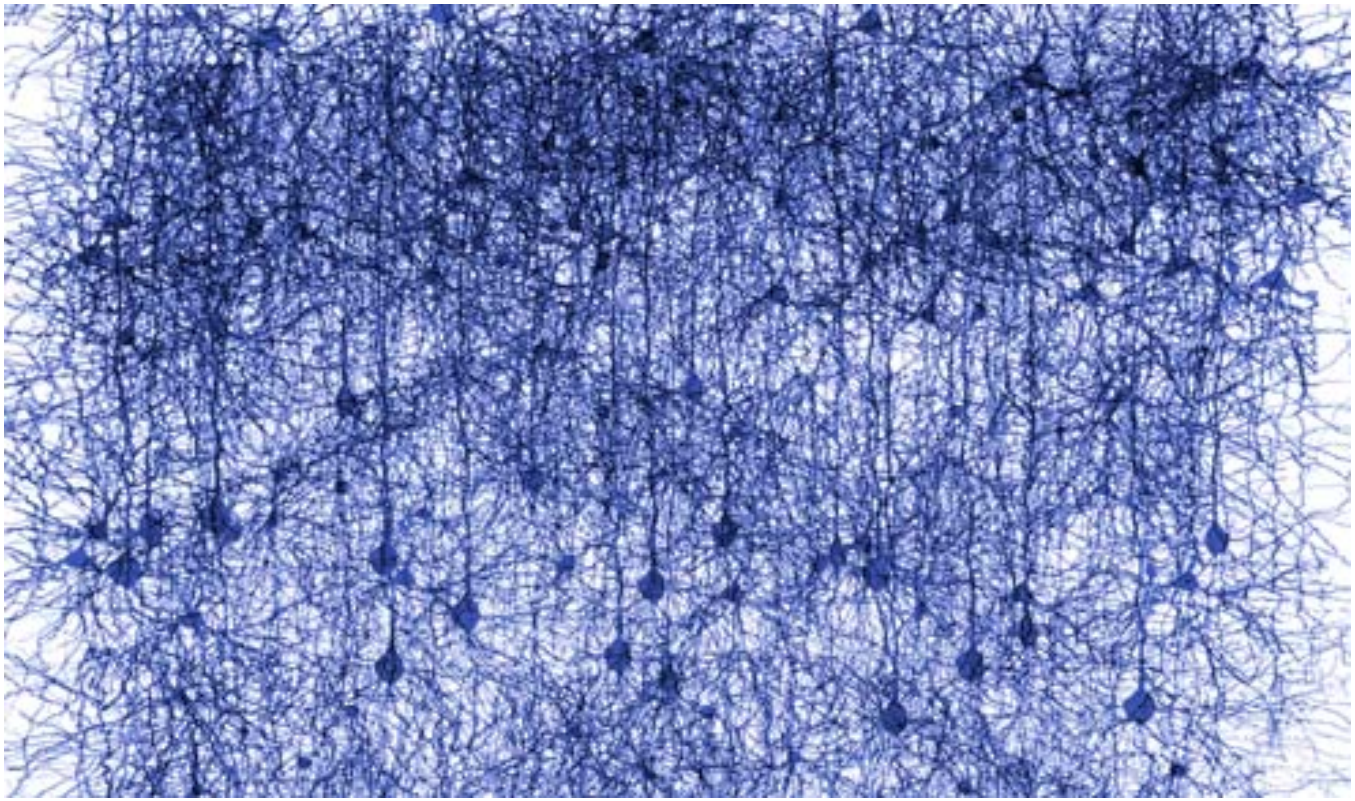
- Book: M.Nielsen, Neural networks and deep learning
- Book, draft available online:

- Portal: `deeplearning.net`

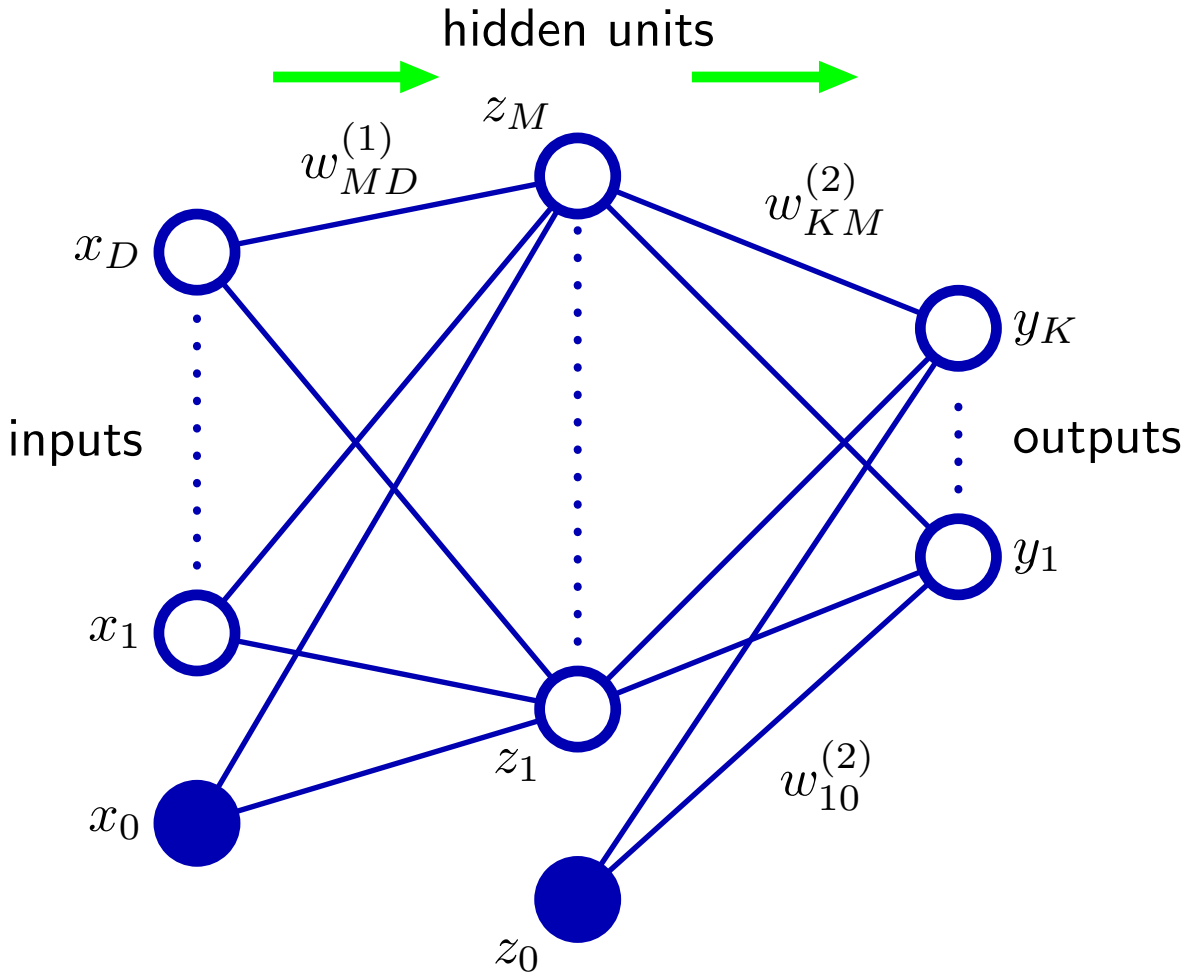
Part 2:
Neural networks

Neural networks (NNs)

- Feedforward neural networks (FFNNs)
- Convolutional neural networks (CNNs)
- Recurrent Neural Networks (RNNs)
- Auto-encoders (AE)



Feed forward neural networks



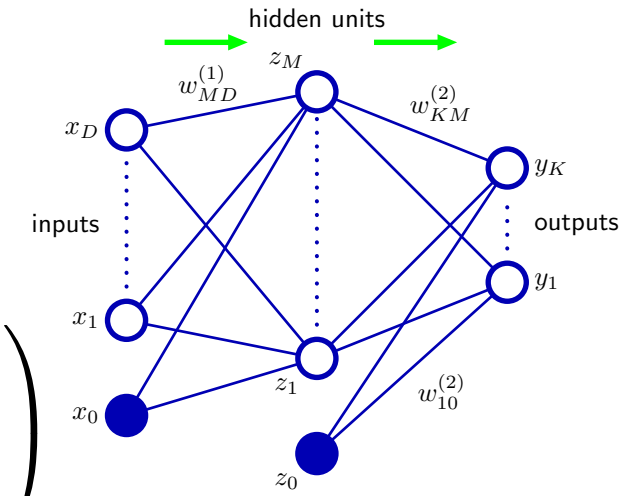
Neural network mapping

- Compute weighted sum of inputs:

$$\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} = \sum_{i=0}^{D} w_{ji}^{(1)} x_i$$

- Output k two-layer network:

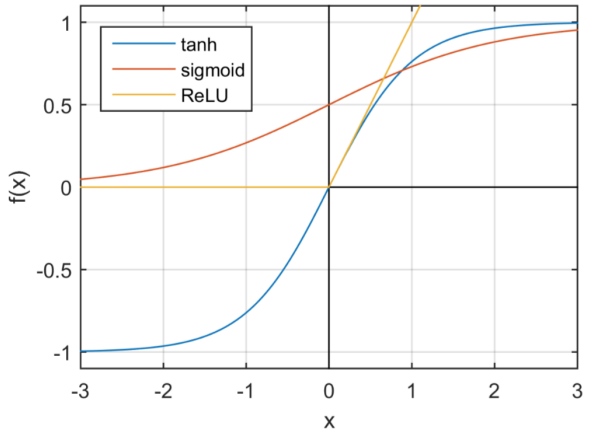
$$h_k^{(2)}(\mathbf{x}, \mathbf{w}) = f_2 \left(\sum_{j=0}^{M} w_{kj}^{(2)} f_1 \left(\sum_{i=0}^{D} w_{ji}^{(1)} x_i \right) \right)$$



- f_1 and f_2 hidden unit activation functions

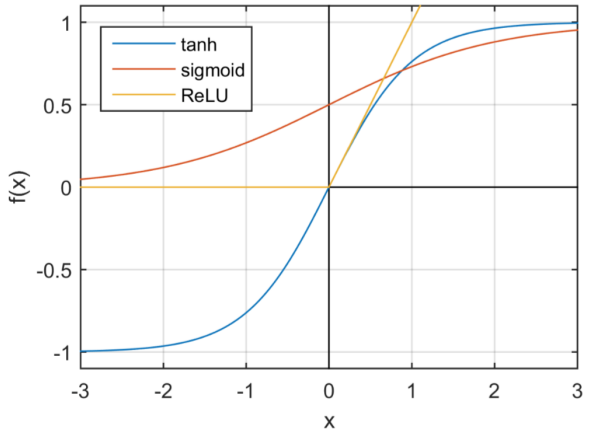
Non-linearity and training

- Linear activation functions will give a linear network.
- Logistic function $\sigma(a) = \frac{1}{1+e^{-a}}$
- Hyperbolic tangent $\tanh(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$
- Rectified linear $\mathrm{relu}(a) = \max(0, a)$



Non-linearity and training

- Linear activation functions will give a linear network.
- Logistic function $\sigma(a) = \frac{1}{1+e^{-a}}$
- Hyperbolic tangent $\tanh(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$
- Rectified linear $\mathrm{relu}(a) = \max(0, a)$



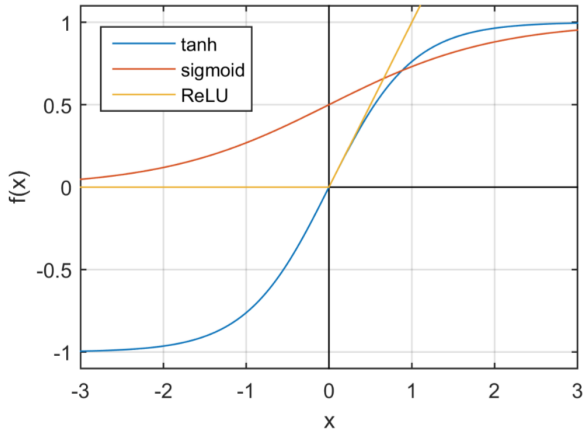
- Supervised learning
- Labeled training set

$$\mathcal{D} = \{(x_i, y_i) | i = 1, \ldots, n\} \ .$$

- Input x_i and output y_i.

Non-linearity and training

- Linear activation functions will give a linear network.
- Logistic function $\sigma(a) = \frac{1}{1+e^{-a}}$
- Hyperbolic tangent $\tanh(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$
- Rectified linear $\mathrm{relu}(a) = \max(0, a)$



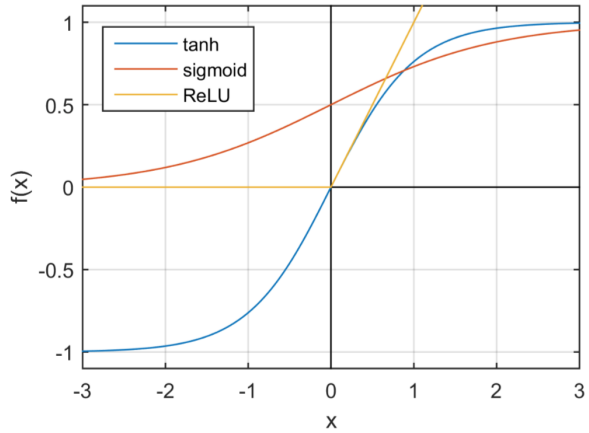
- Supervised learning
- Labeled training set

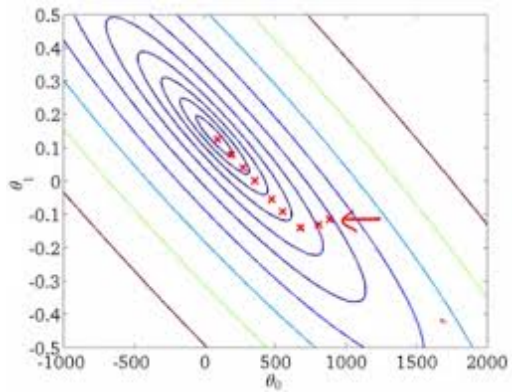
$$\mathcal{D} = \{(x_i, y_i) | i = 1, \ldots, n\} .$$

- Input x_i and output y_i.
- Minimize training error by (stochastic) gradient descent

Non-linearity and training

- Linear activation functions will give a linear network.
- Logistic function $\sigma(a) = \frac{1}{1+e^{-a}}$
- Hyperbolic tangent $\tanh(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$
- Rectified linear $\mathrm{relu}(a) = \max(0, a)$





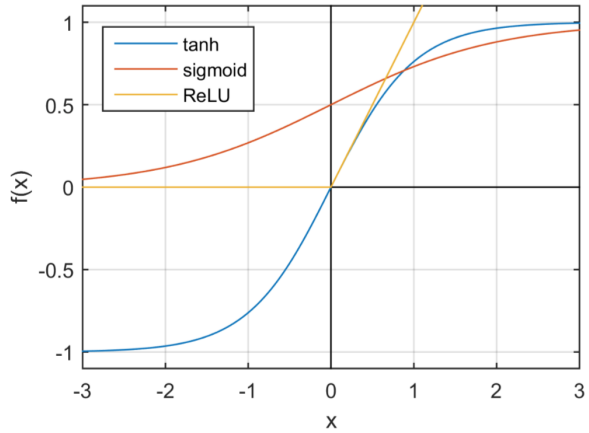
- Supervised learning
- Labeled training set

$$\mathcal{D} = \{(x_i, y_i) | i = 1, \ldots, n\} .$$

- Input x_i and output y_i.
- Minimize training error by (stochastic) gradient descent

Non-linearity and training

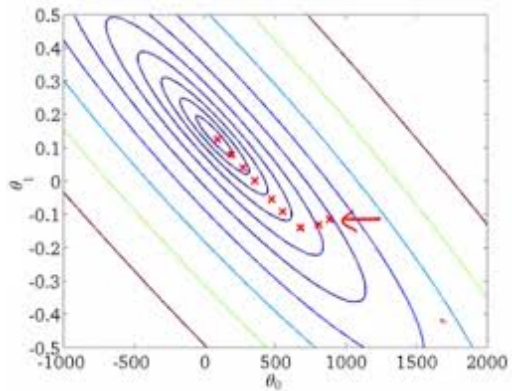
- Linear activation functions will give a linear network.
- Logistic function $\sigma(a) = \frac{1}{1+e^{-a}}$
- Hyperbolic tangent $\tanh(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$
- Rectified linear $\mathrm{relu}(a) = \max(0, a)$

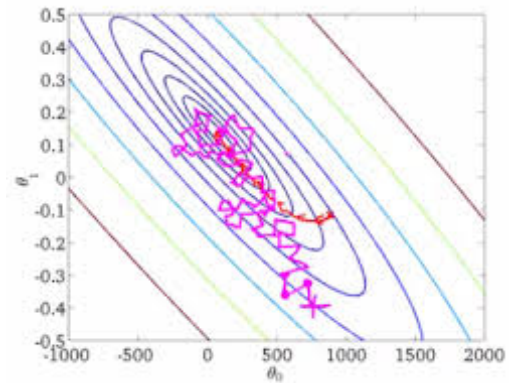


- Supervised learning
- Labeled training set

$$\mathcal{D} = \{(x_i, y_i) | i = 1, \ldots, n\} .$$

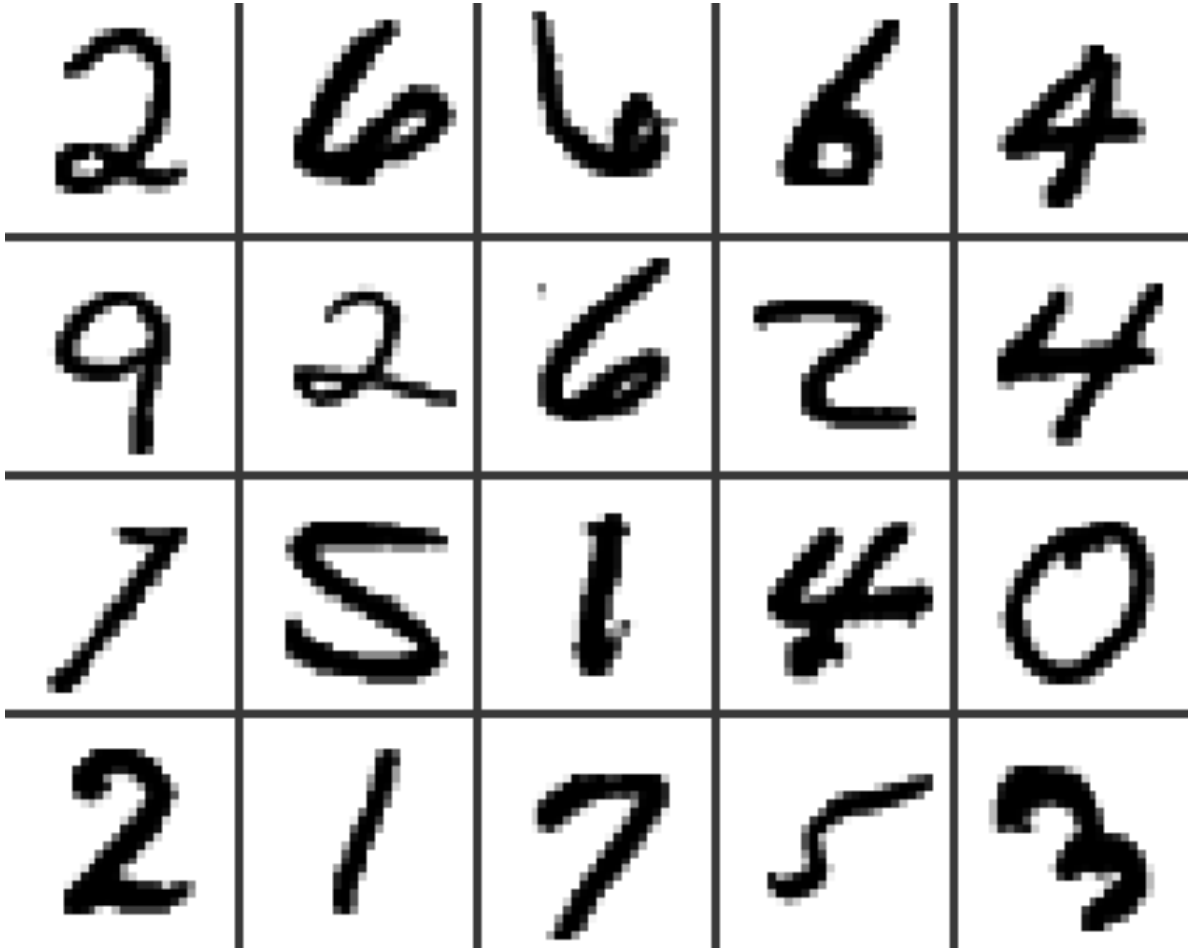
- Input x_i and output y_i.
- Minimize training error by (stochastic) gradient descent





Overfitting!

Example: MNIST handwritten digits

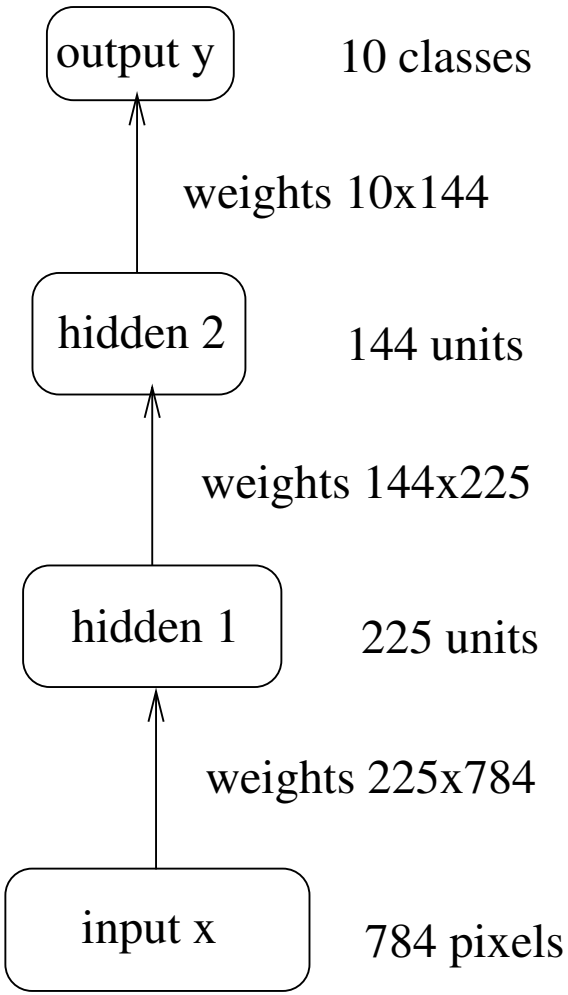


Train a network to classify 28×28 images.

Data: 60000 input images $\mathbf{x}(n)$ and labels $y(n)$.

Example model gives around 1.2% test error.

Example Network



$$\mathbf{h}^{(3)} = \text{softmax}(\mathbf{W}^{(3)}\mathbf{h}^{(2)} + \mathbf{b}^{(3)})$$

$$\mathbf{h}^{(2)} = \text{relu}(\mathbf{W}^{(2)}\mathbf{h}^{(1)} + \mathbf{b}^{(2)})$$

$$\mathbf{h}^{(1)} = \text{relu}(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)})$$

$$\text{softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

$$\text{relu}(z) = max(0, z)$$

Softmax

- Softmax function

$$\text{softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

has two nice properties:

- $\text{softmax}(\mathbf{z})_i \geq 0$
- $\sum_i \text{softmax}(\mathbf{z})_i = 1$

Softmax

- Softmax function

$$\text{softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

  has two nice properties:

- $\text{softmax}(\mathbf{z})_i \geq 0$
- $\sum_i \text{softmax}(\mathbf{z})_i = 1$
- MNIST, output labels: $0, 1, \ldots, 9$.
- Output of network

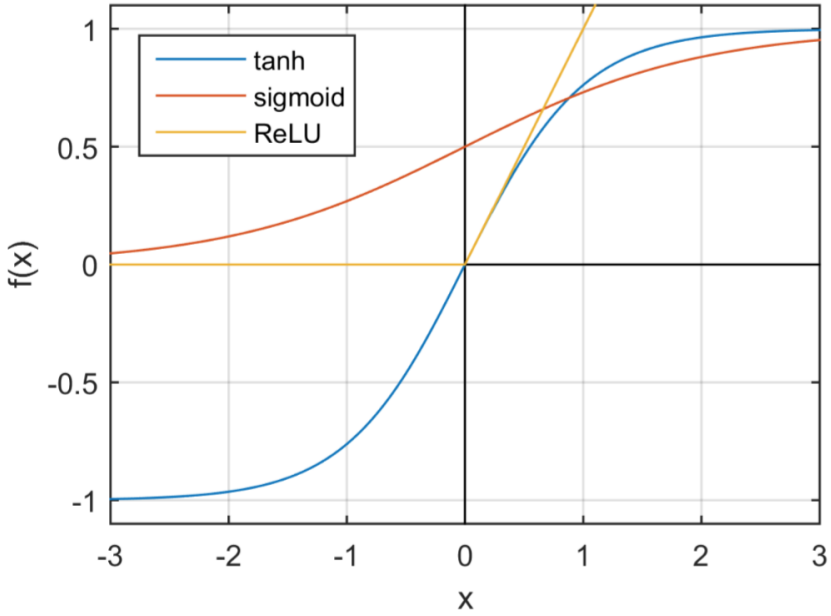
$$\mathbf{h}^{(3)} = \text{softmax}(\mathbf{W}^{(3)}\mathbf{h}^{(2)} + \mathbf{b}^{(3)})$$

  interpreted as class(-conditional) probabilities:

- For example:

$$p(5|\mathbf{x}) = \mathbf{h}_5^{(3)}$$

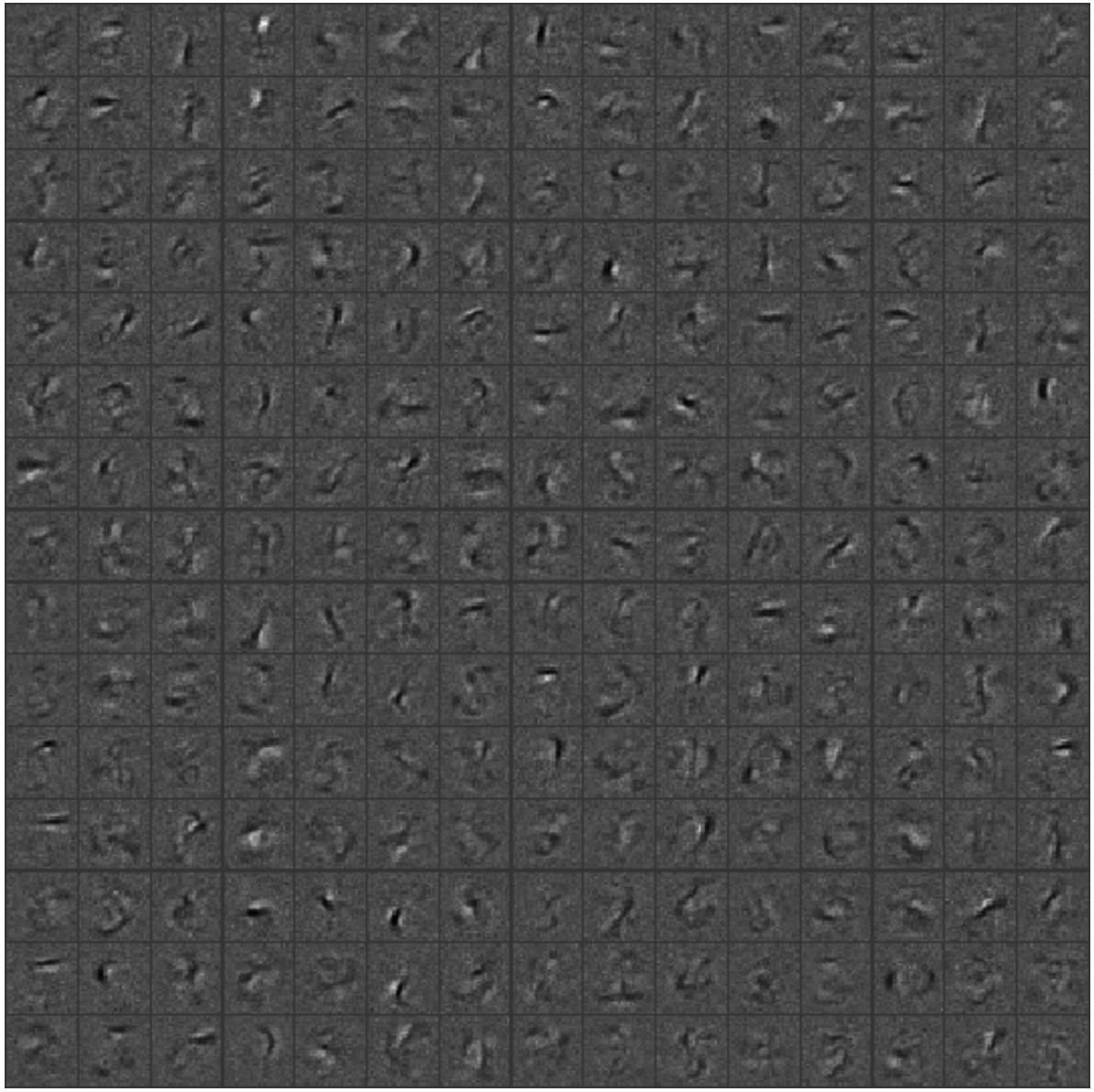
On activation functions



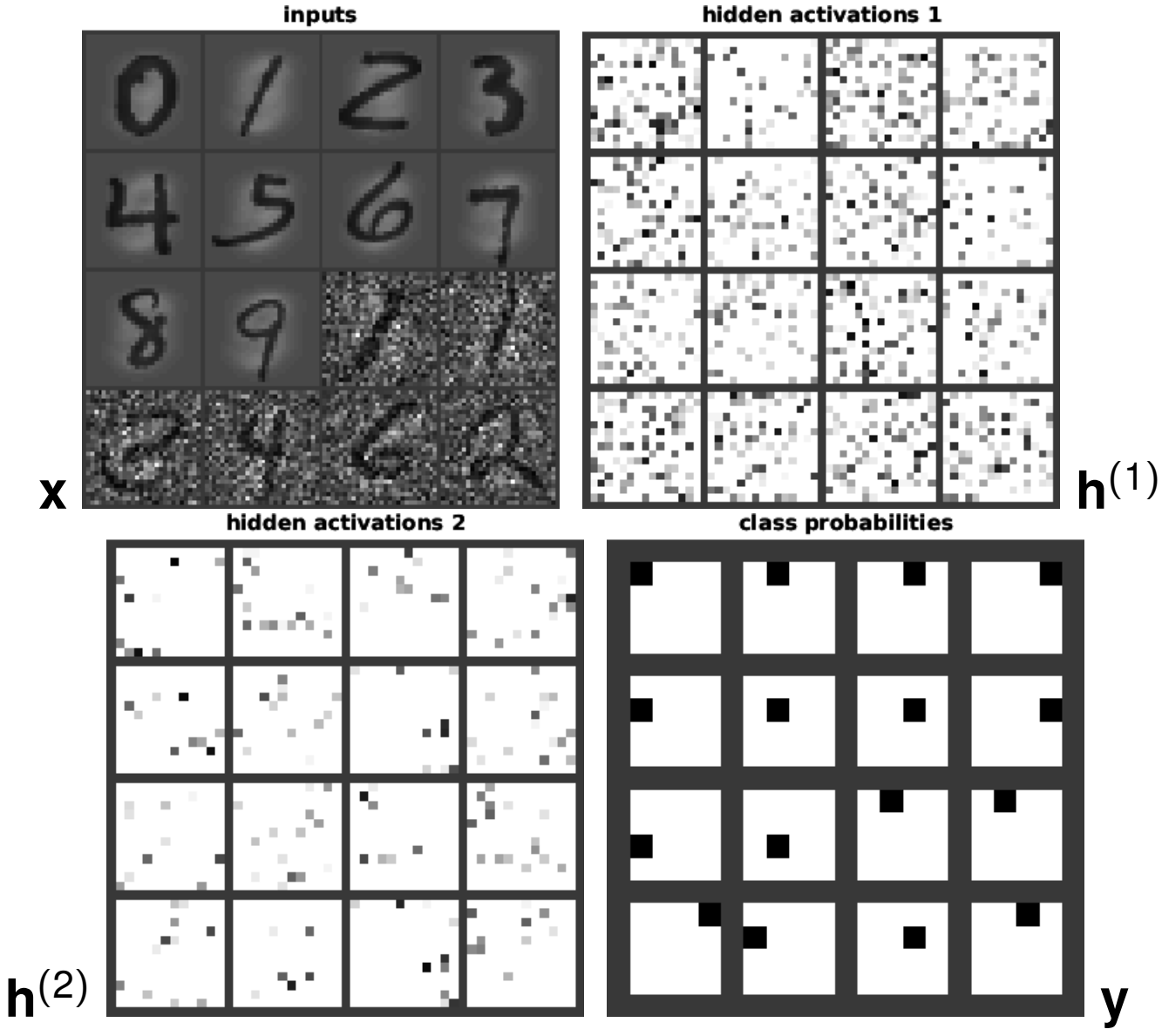
- $\text{relu}(z) = \max(0, z)$ is replacing old sigmoid and tanh.
- Note that identity function would lead into:

$$\begin{aligned}
\mathbf{h}^{(2)} &= \mathbf{W}^{(2)}\mathbf{h}^{(1)} + \mathbf{b}^{(2)} \\
&= \mathbf{W}^{(2)}(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) + \mathbf{b}^{(2)} \\
&= (\mathbf{W}^{(2)}\mathbf{W}^{(1)})\mathbf{x} + (\mathbf{W}^{(2)}\mathbf{b}^{(1)} + \mathbf{b}^{(2)}) \\
&= \mathbf{W}'\mathbf{x} + \mathbf{b}'
\end{aligned}$$

Weight matrix $\mathbf{W}^{(1)}$ size 225×784

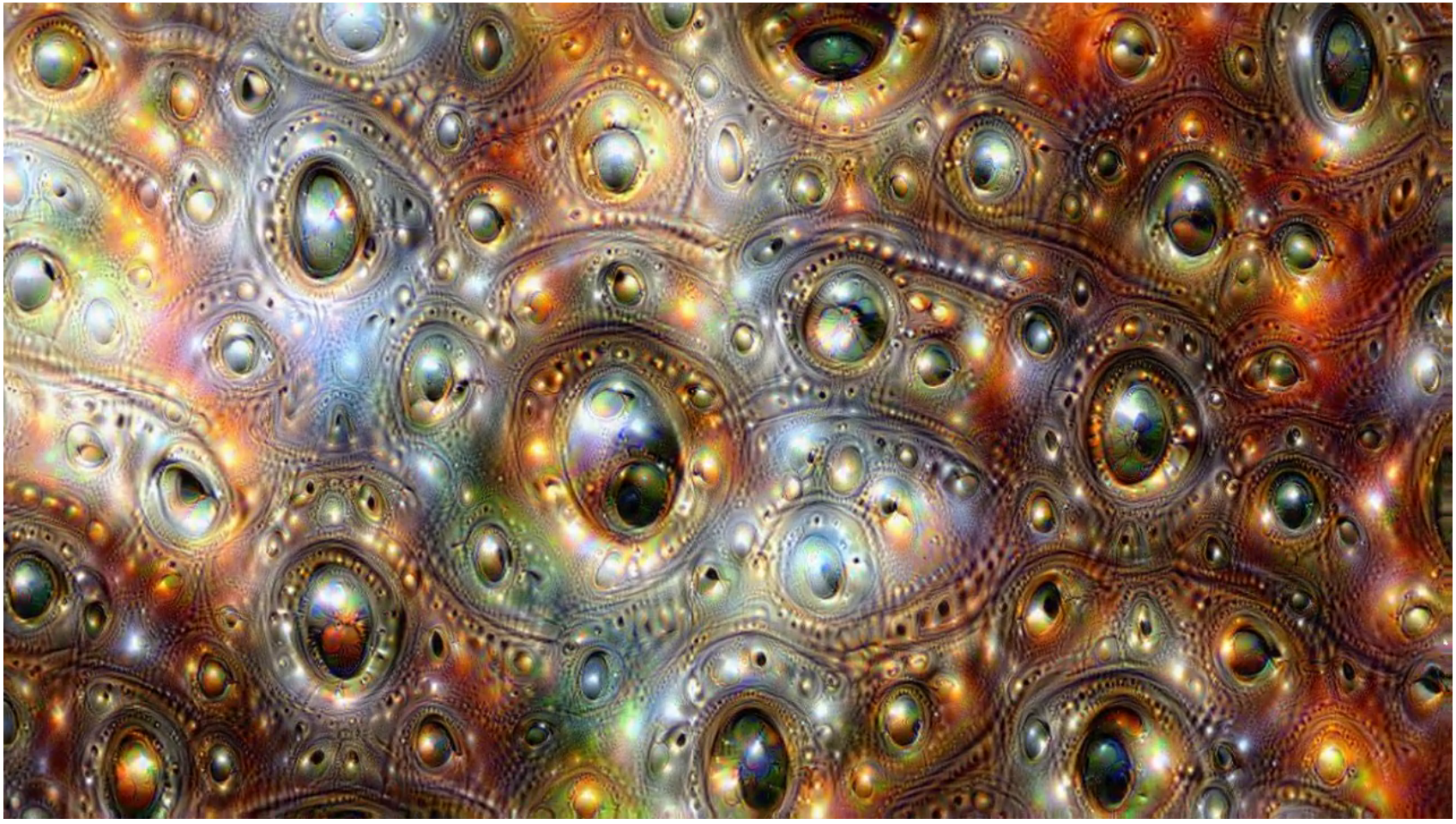


Signals $\mathbf{x} \rightarrow \mathbf{h}^{(1)} \rightarrow \mathbf{h}^{(2)} \rightarrow \mathbf{h}^{(3)}$



References

- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton, Deep learning, Nature 521.7553 (2015): 436-444.
- Mnih, Volodymyr, et al., Human-level control through deep reinforcement learning, Nature 518.7540 (2015): 529-533.
- Alipanahi, Babak, et al., Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning, Nature biotechnology (2015).
- Silver, David, et al., Mastering the game of Go with deep neural networks and tree search, Nature 529.7587 (2016): 484-489.
- Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhutdinov, R., Zemel, R., & Bengio, Y. (2015), Show, attend and tell: Neural image caption generation with visual attention. arXiv preprint arXiv:1502.03044.
- Mansimov, Elman, et al., Generating Images from Captions with Attention. arXiv preprint arXiv:1511.02793 (2015).
- Larsen, Anders Boesen Lindbo, Soren Kaae Sonderby, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. arXiv preprint arXiv:1512.09300 (2015).
- Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. Book in preparation for MIT Press, `http://goodfeli.github.io/dlbook/`, 2016.
- Michael Nielsen, Neural Networks and Deep Learning, `http://neuralnetworksanddeeplearning.com/`
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, ImageNet Classification with Deep Convolutional, NIPS, 2012. Neural Networks,



Thanks!
Ole Winther