Introduction to Deep Learning

Milan Straka

March 04, 2019

EUROPEAN UNION
European Structural and Investment Fund
Operational Programme Research, Development and Education

Charles University in Prague
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Deep Learning Highlights

- Image recognition
- Object detection
- Image segmentation,
- Human pose estimation
- Image labeling
- Visual question answering
- Speech recognition and generation
- Lip reading
- Machine translation
- Machine translation without parallel data
- Chess, Go and Shogi
- Multiplayer Capture the flag

Notation

- a, $\boldsymbol{a}$, $\boldsymbol{A}$, A: scalar (integer or real), vector, matrix, tensor
- a, $\mathbf{a}$, $\mathbf{A}$: scalar, vector, matrix random variable
- $\frac{df}{dx}$: derivative of f with respect to x
- $\frac{\partial f}{\partial x}$: partial derivative of f with respect to x
- $\nabla_{\boldsymbol{x}} f$: gradient of f with respect to $\boldsymbol{x}$, i.e., $\left(\frac{\partial f(\boldsymbol{x})}{\partial x_1}, \frac{\partial f(\boldsymbol{x})}{\partial x_2}, \ldots, \frac{\partial f(\boldsymbol{x})}{\partial x_n}\right)$

A random variable x is a result of a random process. It can be discrete or continuous.

Probability Distribution

A probability distribution describes how likely are individual values a random variable can take.

The notation $\mathrm{x} \sim P$ stands for a random variable x having a distribution P.

For discrete variables, the probability that x takes a value x is denoted as $P(x)$ or explicitly as $P(\mathrm{x} = x)$.

For continuous variables, the probability that the value of x lies in the interval $[a, b]$ is given by $\int_a^b p(x)\,\mathrm{d}x$.

Expectation

The expectation of a function $f(x)$ with respect to discrete probability distribution $P(x)$ is defined as:

$$\mathbb{E}_{\mathrm{x}\sim P}[f(x)] \stackrel{\text{def}}{=} \sum_x P(x)f(x)$$

For continuous variables it is computed as:

$$\mathbb{E}_{\mathrm{x}\sim p}[f(x)] \stackrel{\text{def}}{=} \int_x p(x)f(x)\,\mathrm{d}x$$

If the random variable is obvious from context, we can write only $\mathbb{E}_P[x]$ of even $\mathbb{E}[x]$.

Expectation is linear, i.e.,

$$\mathbb{E}_{\mathrm{x}}[\alpha f(x) + \beta g(x)] = \alpha\mathbb{E}_{\mathrm{x}}[f(x)] + \beta\mathbb{E}_{\mathrm{x}}[g(x)]$$

Variance

Variance measures how much the values of a random variable differ from its mean $\mu = \mathbb{E}[x]$.

$$\mathrm{Var}(x) \stackrel{\text{def}}{=} \mathbb{E}\left[\big(x - \mathbb{E}[x]\big)^2\right], \text{ or more generally}$$
$$\mathrm{Var}(f(x)) \stackrel{\text{def}}{=} \mathbb{E}\left[\big(f(x) - \mathbb{E}[f(x)]\big)^2\right]$$

It is easy to see that

$$\mathrm{Var}(x) = \mathbb{E}\left[x^2 - 2x\mathbb{E}[x] + \big(\mathbb{E}[x]\big)^2\right] = \mathbb{E}\left[x^2\right] - \big(\mathbb{E}[x]\big)^2.$$

Variance is connected to $E[x^2]$, a *second moment* of a random variable – it is in fact a *centered* second moment.

Bernoulli Distribution

The Bernoulli distribution is a distribution over a binary random variable. It has a single parameter $\varphi \in [0, 1]$, which specifies the probability of the random variable being equal to 1.

$$
\begin{aligned}
P(x) &= \varphi^x (1-\varphi)^{1-x} \\
\mathbb{E}[x] &= \varphi \\
\mathrm{Var}(x) &= \varphi(1-\varphi)
\end{aligned}
$$

Categorical Distribution

Extension of the Bernoulli distribution to random variables taking one of k different discrete outcomes. It is parametrized by $\boldsymbol{p} \in [0, 1]^k$ such that $\sum_{i=1}^{k} p_i = 1$.

$$
\begin{aligned}
P(\boldsymbol{x}) &= \prod\nolimits_i^k p_i^{x_i} \\
\mathbb{E}[x_i] &= p_i, \mathrm{Var}(x_i) = p_i(1-p_i)
\end{aligned}
$$

Self Information

Amount of *surprise* when a random variable is sampled.

- Should be zero for events with probability 1.
- Less likely events are more surprising.
- Independent events should have *additive* information.

$$I(x) \stackrel{\text{def}}{=} -\log P(x) = \log \frac{1}{P(x)}$$

Entropy

Amount of *surprise* in the whole distribution.

$$H(P) \stackrel{\text{def}}{=} \mathbb{E}_{\mathrm{x}\sim P}[I(x)] = -\mathbb{E}_{\mathrm{x}\sim P}[\log P(x)]$$

- for discrete P: $H(P) = -\sum_x P(x)\log P(x)$
- for continuous P: $H(P) = -\int P(x)\log P(x)\,\mathrm{d}x$

Cross-Entropy

$$H(P, Q) \stackrel{\text{def}}{=} -\mathbb{E}_{\mathrm{x} \sim P}[\log Q(x)]$$

- Gibbs inequality
  - $H(P, Q) \geq H(P)$
  - $H(P) = H(P, Q) \Leftrightarrow P = Q$
  - Proof: Using Jensen's inequality, we get

$$\sum_x P(x) \log \frac{Q(x)}{P(x)} \leq \log \sum_x P(x) \frac{Q(x)}{P(x)} = \log \sum_x Q(x) = 0.$$

  - Corollary: For a categorical distribution with n outcomes, $H(P) \leq \log n$, because for $Q(x) = 1/n$ we get $H(P) \leq H(P, Q) = -\sum_x P(x) \log Q(x) = \log n.$
- generally $H(P, Q) \neq H(Q, P)$

Kullback-Leibler Divergence (KL Divergence)

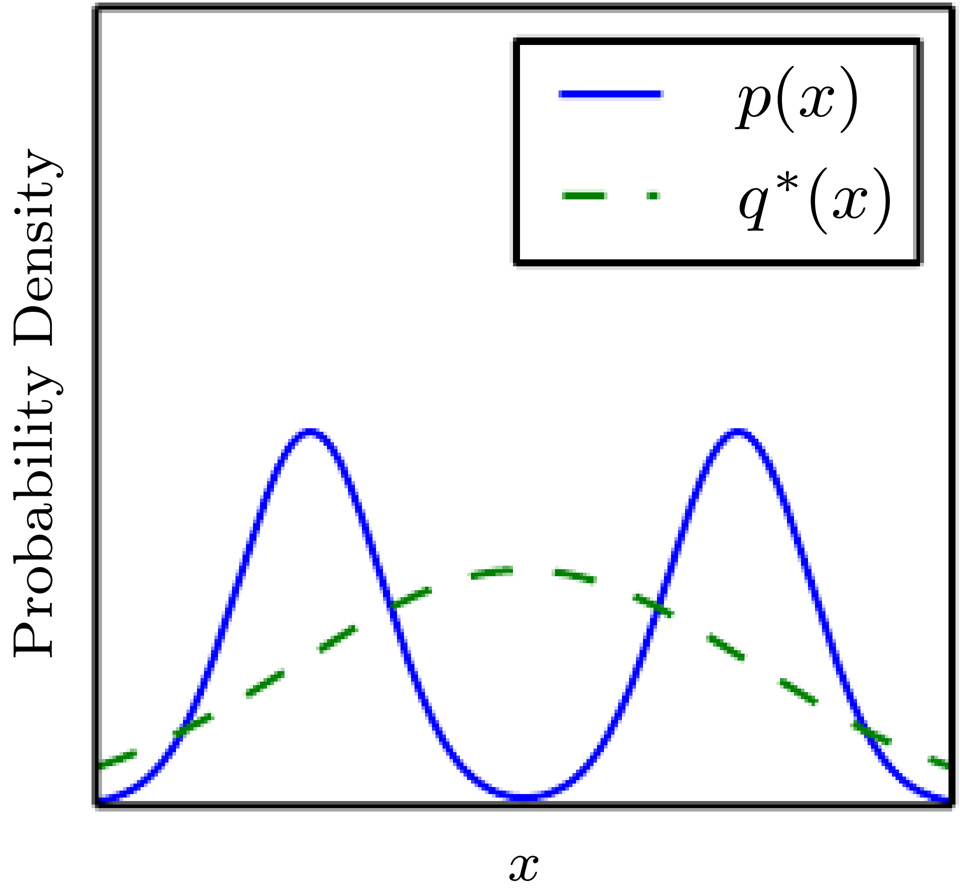
Sometimes also called *relative entropy*.

$$D_{\mathrm{KL}}(P||Q) \stackrel{\text{def}}{=} H(P,Q) - H(P) = \mathbb{E}_{\mathrm{x}\sim P}[\log P(x) - \log Q(x)]$$

- consequence of Gibbs inequality: $D_{\mathrm{KL}}(P||Q) \geq 0$
- generally $D_{\mathrm{KL}}(P||Q) \neq D_{\mathrm{KL}}(Q||P)$

Nonsymmetry of KL Divergence

$$q^* = \operatorname{argmin}_q D_{\mathrm{KL}}(p\|q)$$



$$q^* = \operatorname{argmin}_q D_{\mathrm{KL}}(q\|p)$$

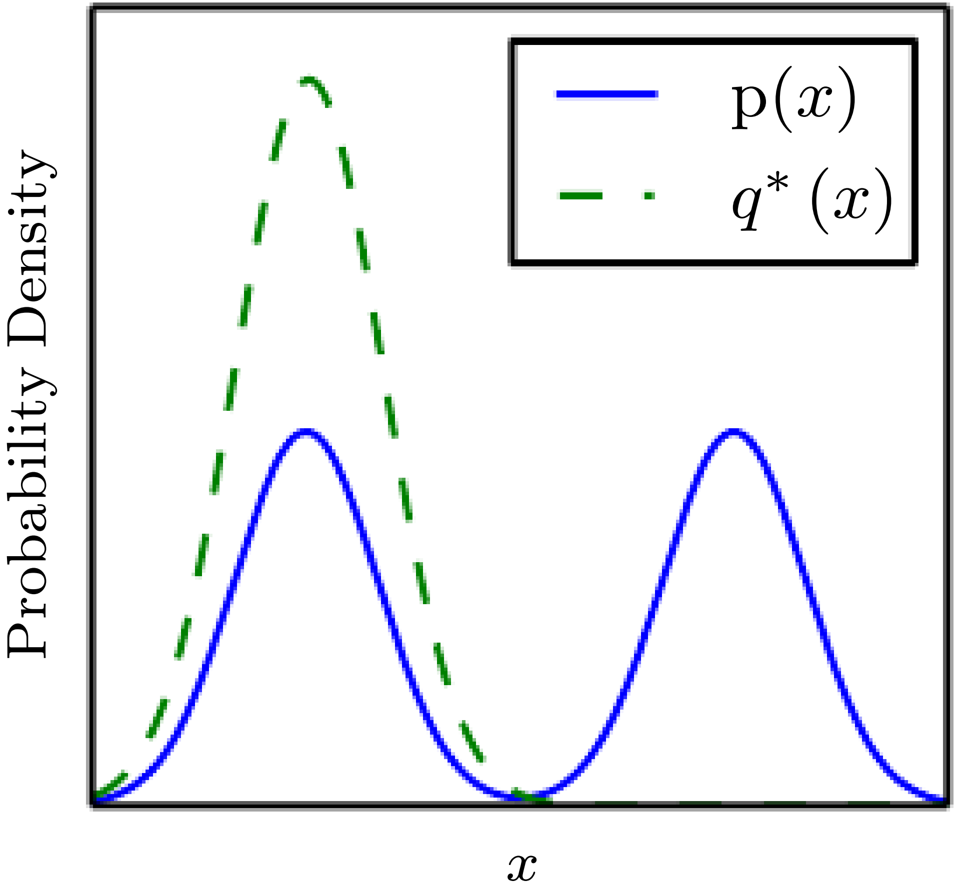


Figure 3.6, page 76 of Deep Learning Book, http://deeplearningbook.org

Normal (or Gaussian) Distribution

Distribution over real numbers, parametrized by a mean μ and variance σ^2:

$$\mathcal{N}(x; \mu, \sigma^2) = \sqrt{\frac{1}{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

For standard values $\mu = 0$ and $\sigma^2 = 1$ we get $\mathcal{N}(x; 0, 1) = \sqrt{\frac{1}{2\pi}} e^{-\frac{x^2}{2}}$.

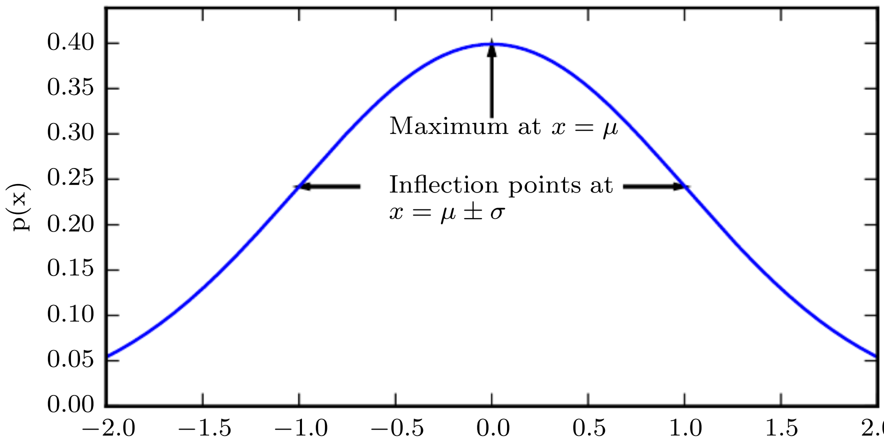


Figure 3.1, page 64 of Deep Learning Book, http://deeplearningbook.org.

Why Normal Distribution

Central Limit Theorem

The sum of independent identically distributed random variables with finite variance converges to normal distribution.

Principle of Maximum Entropy

Given a set of constraints, a distribution with maximal entropy fulfilling the constraints can be considered the most general one, containing as little additional assumptions as possible.

Considering distributions with a given mean and variance, it can be proven (using variational inference) that such a distribution with *maximal entropy* is exactly the normal distribution.

Machine Learning

A possible definition of learning from Mitchell (1997):

> A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

- Task T
  - *classification*: assigning one of k categories to a given input
  - *regression*: producing a number $x \in \mathbb{R}$ for a given input
  - *structured prediction*, *denoising*, *density estimation*, …
- Experience E
  - *supervised*: usually a dataset with desired outcomes (*labels* or *targets*)
  - *unsupervised*: usually data without any annotation (raw text, raw images, …)
  - *reinforcement learning*, *semi-supervised learning*, …
- Measure P
  - *accuracy*, *error rate*, *F-score*, …

Well-known Datasets

Name	Description	Instances
MNIST	Images (28x28, grayscale) of handwritten digits.	60k
CIFAR-10	Images (32x32, color) of 10 classes of objects.	50k
CIFAR-100	Images (32x32, color) of 100 classes of objects (with 20 defined superclasses).	50k
ImageNet	Labeled object image database (labeled objects, some with bounding boxes).	14.2M
ImageNet-ILSVRC	Subset of ImageNet for Large Scale Visual Recognition Challenge, annotated with 1000 object classes and their bounding boxes.	1.2M
COCO	*Common Objects in Context*: Complex everyday scenes with descriptions (5) and highlighting of objects (91 types).	2.5M

ImageNet-ILSVRC

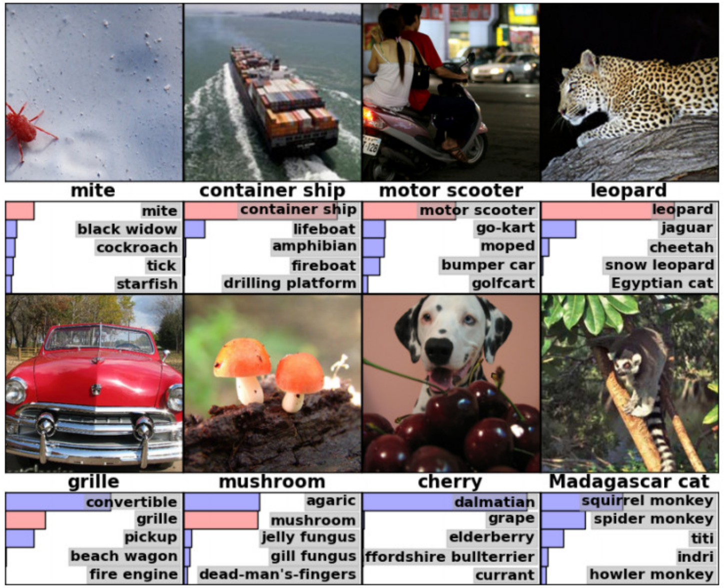


Image from "ImageNet Classification with Deep Convolutional Neural Networks" paper by Alex Krizhevsky et al.

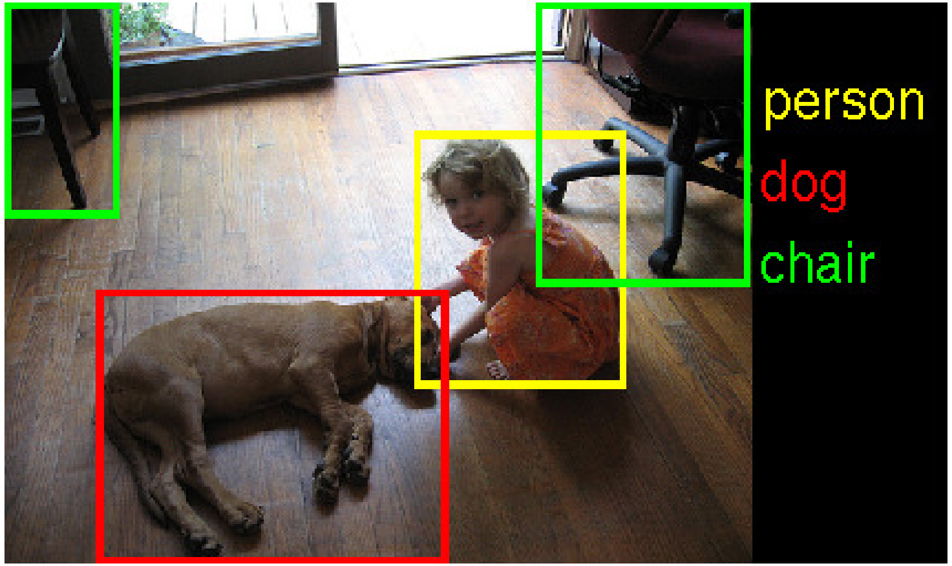


Image from http://image-net.org/challenges/LSVRC/2014/.

Well-known Datasets

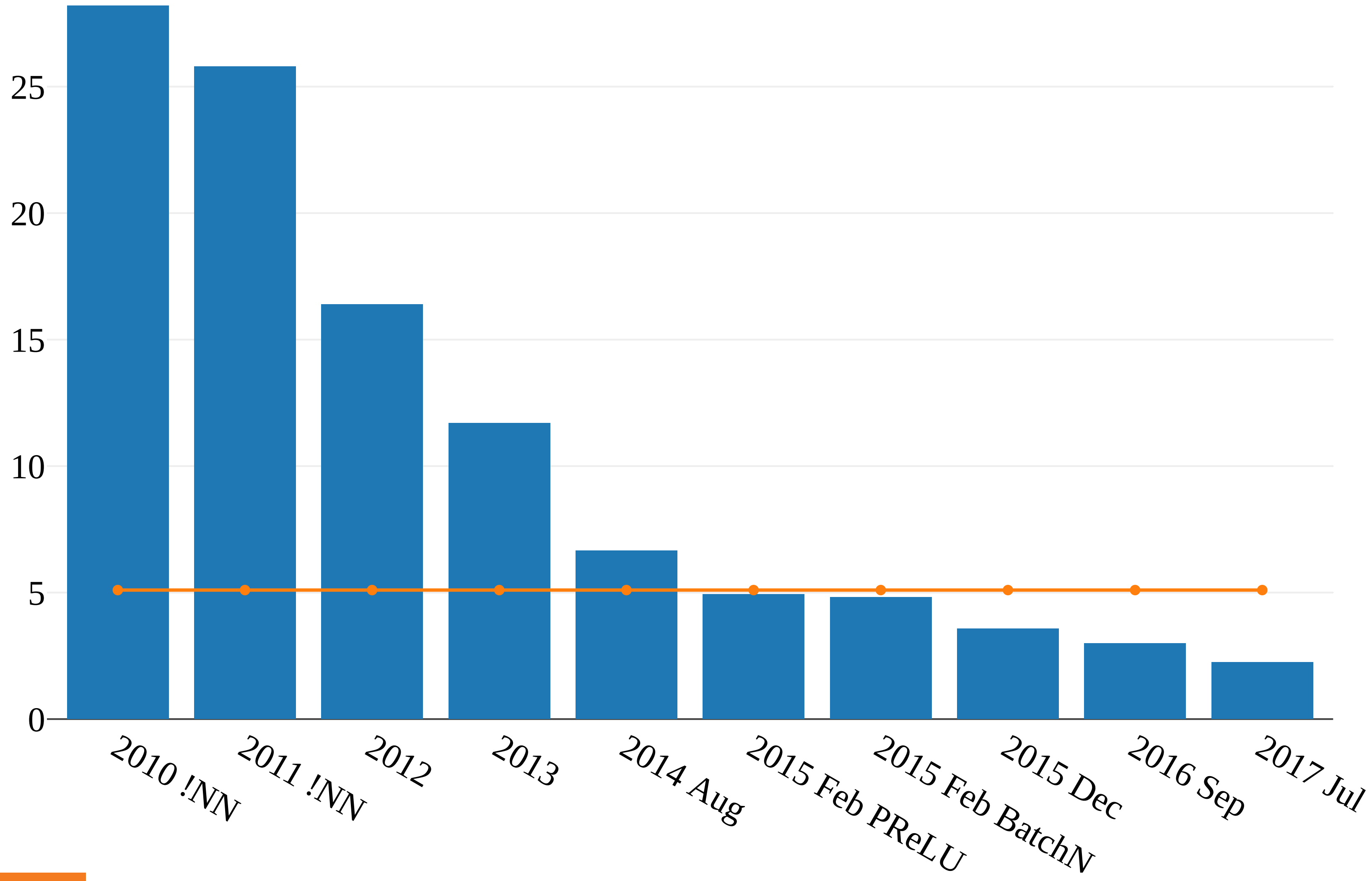
COCO

Image from http://mscoco.org/dataset/\#detections-challenge2016.

Well-known Datasets

Name	Description	Instances
IAM-OnDB	Pen tip movements of handwritten English from 221 writers.	86k words
TIMIT	Recordings of 630 speakers of 8 dialects of American English.	6.3k sents
CommonVoice	400k recordings from 20k people, around 500 hours of speech.	400k
PTB	*Penn Treebank*: 2500 stories from Wall Street Journal, with POS tags and parsed into trees.	1M words
PDT	*Prague Dependency Treebank*: Czech sentences annotated on 4 layers (word, morphological, analytical, tectogrammatical).	1.9M words
UD	*Universal Dependencies*: Treebanks of 76 languages with consistent annotation of lemmas, POS tags, morphology and syntax.	129 treebanks
WMT	Aligned parallel sentences for machine translation.	gigawords

ILSVRC Image Recognition Error Rates



In summer 2017, a paper came out describing automatic generation of neural architectures using reinforcement learning.

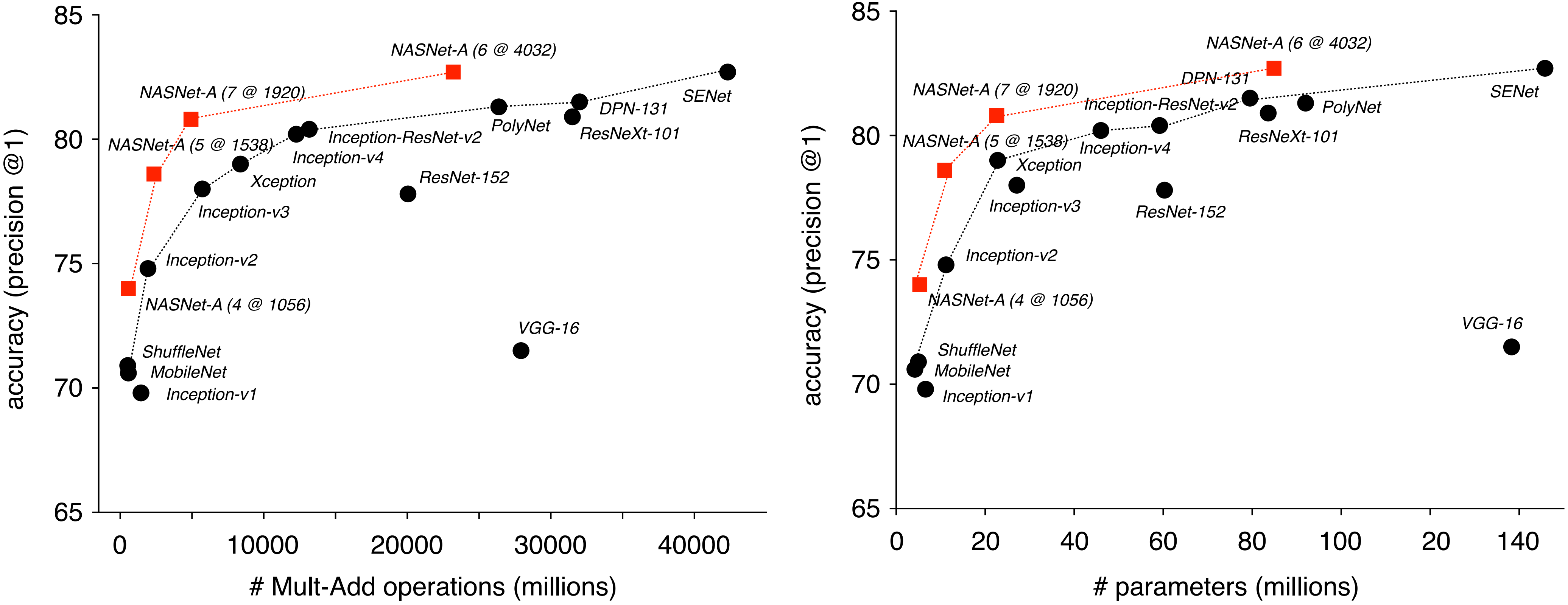
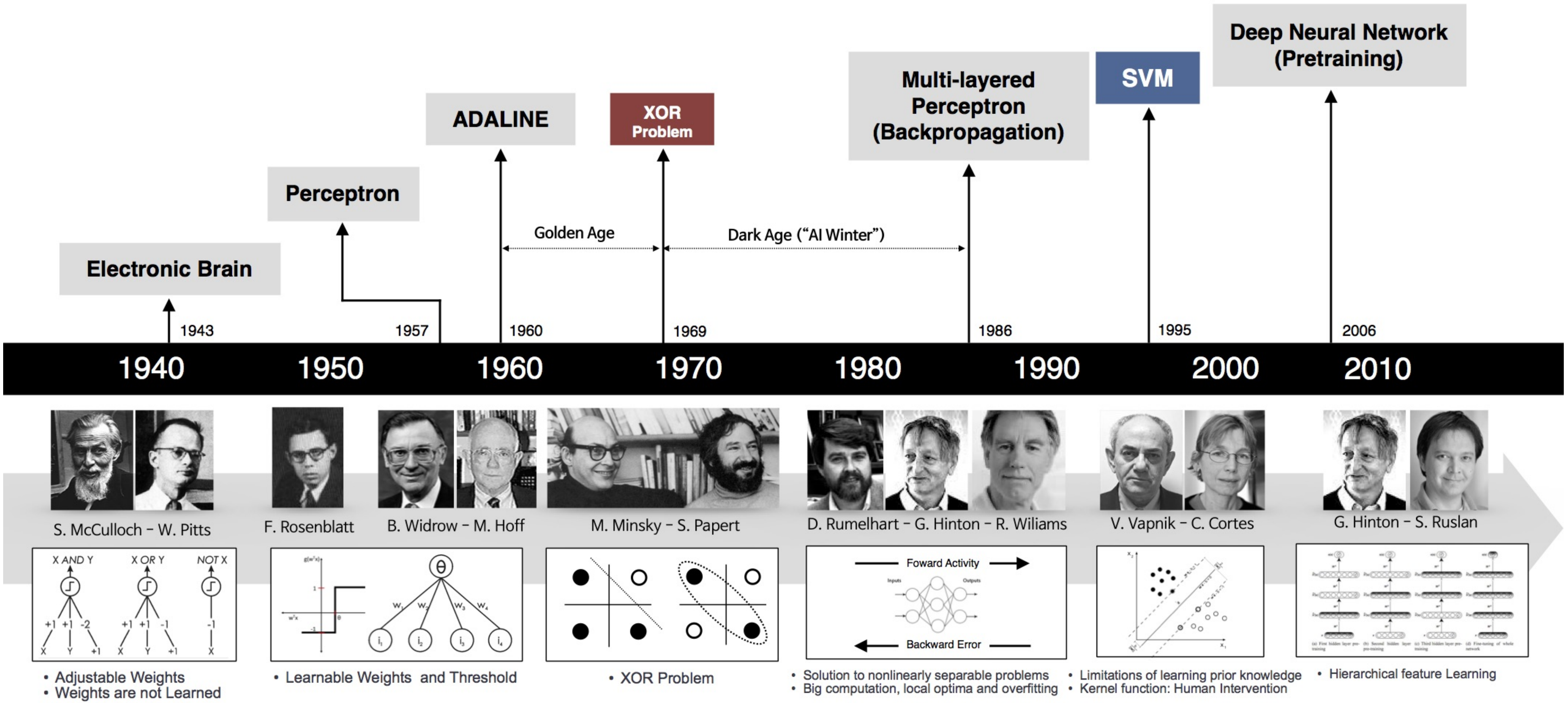


Figure 5 of paper "Learning Transferable Architectures for Scalable Image Recognition", https://arxiv.org/abs/1707.07012.

Introduction to Machine Learning History



https://www.slideshare.net/deview/251-implementing-deep-learning-using-cu-dnn/4

Curse of Dimensionality

Figure 5.9, page 156 of Deep Learning Book, http://deeplearningbook.org.

Machine and Representation Learning

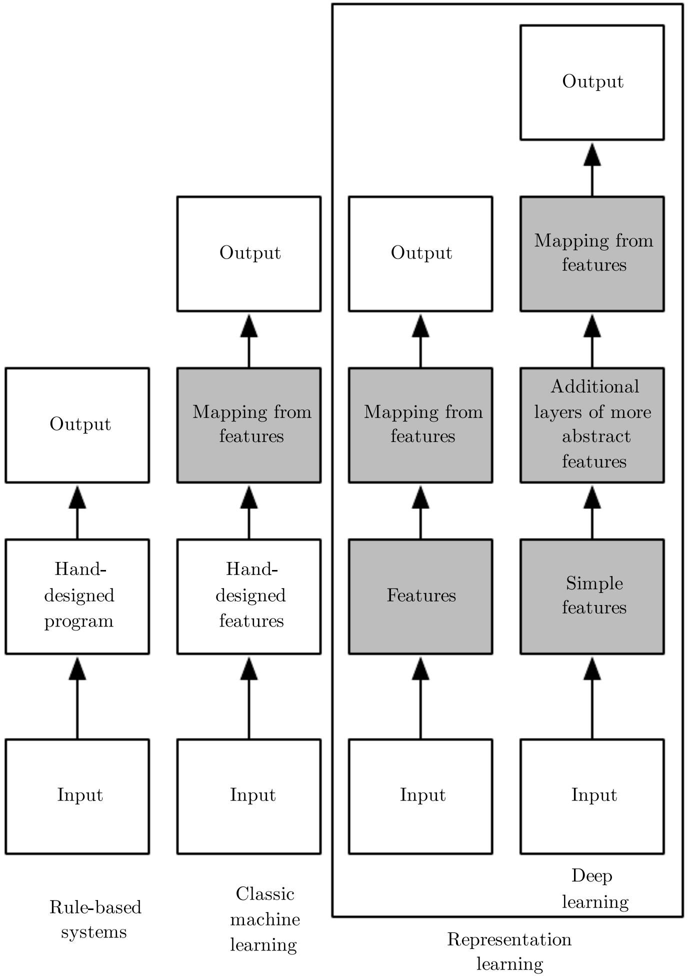
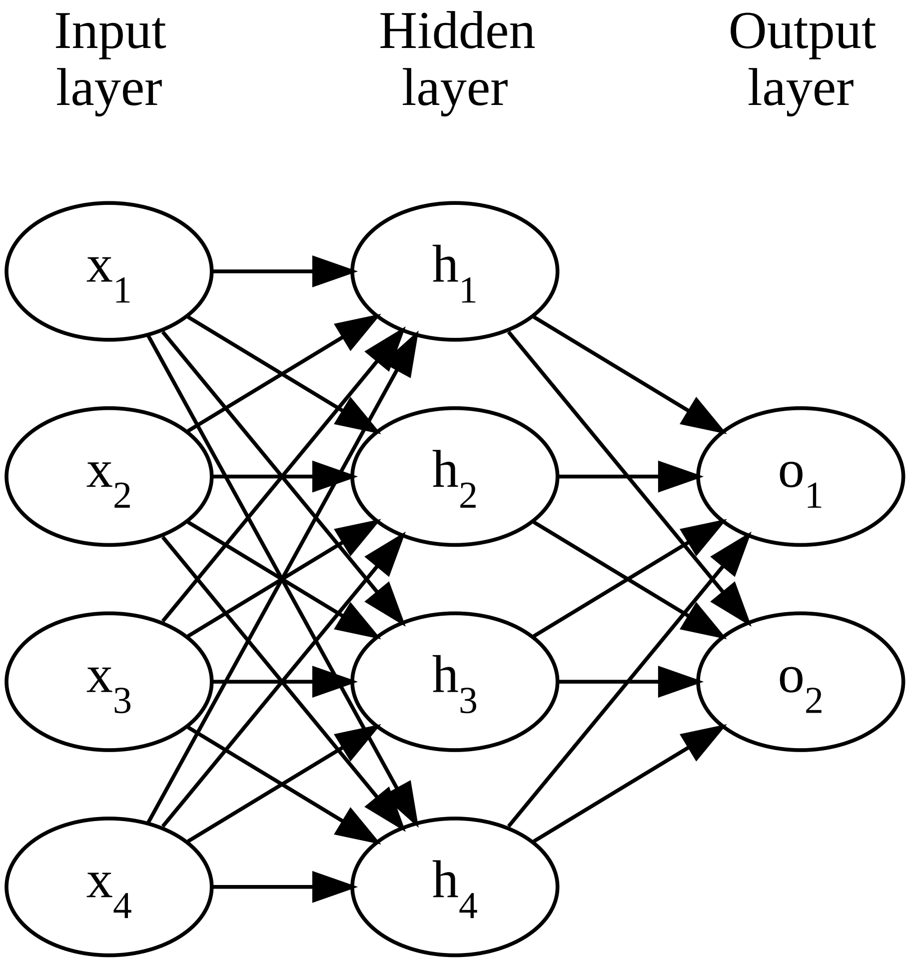


Figure 1.5, page 10 of Deep Learning Book, http://deeplearningbook.org.

Neural Network Architecture à la '80s



Neural Network Architecture

There is a weight on each edge, and an activation function f is performed on the hidden layers, and optionally also on the output layer.

$$h_i = f\left(\sum_j w_{i,j} x_j\right)$$

If the network is composed of layers, we can use matrix notation and write:

$$\boldsymbol{h} = f\left(\boldsymbol{W}\boldsymbol{x}\right)$$

Output Layers

- none (linear regression if there are no hidden layers)
- σ (sigmoid; logistic regression if there are no hidden layers)

$$\sigma(x) \stackrel{\text{def}}{=} \frac{1}{1 + e^{-x}}$$

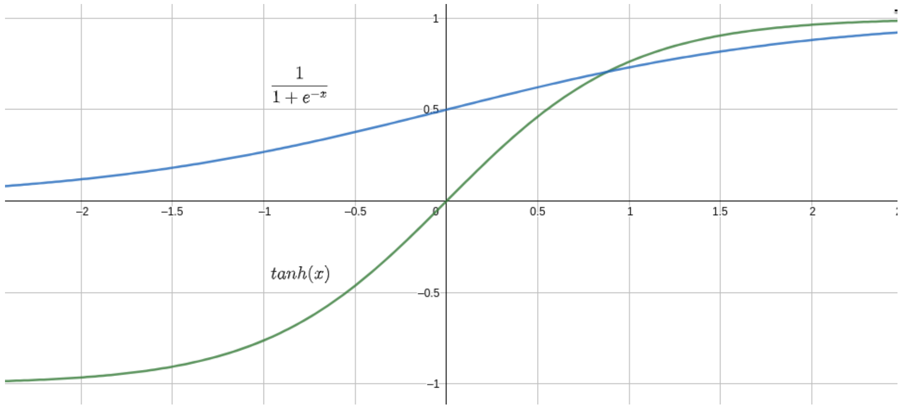
- $\operatorname{softmax}$ (maximum entropy model if there are no hidden layers)

$$\operatorname{softmax}(\boldsymbol{x}) \propto e^{\boldsymbol{x}}$$

$$\operatorname{softmax}(\boldsymbol{x})_i \stackrel{\text{def}}{=} \frac{e^{x_i}}{\sum_j e^{x_j}}$$

Hidden Layers

- none (does not help, composition of linear mapping is a linear mapping)
- σ (but works badly – nonsymmetrical, $\frac{d\sigma}{dx}(0) = 1/4$)
- $\tanh$
  - result of making σ symmetrical and making derivation in zero 1
  - $\tanh(x) = 2\sigma(2x) - 1$



- ReLU
  - $\max(0, x)$

Universal Approximation Theorem '89

Let $\varphi(x)$ be a nonconstant, bounded and monotonically-increasing continuous function.

Then for any $\varepsilon > 0$ and any continuous function f on $[0,1]^m$ there exists an $N \in \mathbb{N}, v_i \in \mathbb{R}, b_i \in \mathbb{R}$ and $\boldsymbol{w_i} \in \mathbb{R}^m$, such that if we denote

$$F(\boldsymbol{x}) = \sum_{i=1}^N v_i \varphi(\boldsymbol{w_i} \cdot \boldsymbol{x} + b_i)$$

then for all $x \in [0,1]^m$

$$|F(\boldsymbol{x}) - f(\boldsymbol{x})| < \varepsilon.$$

Evolving ReLU Approximation