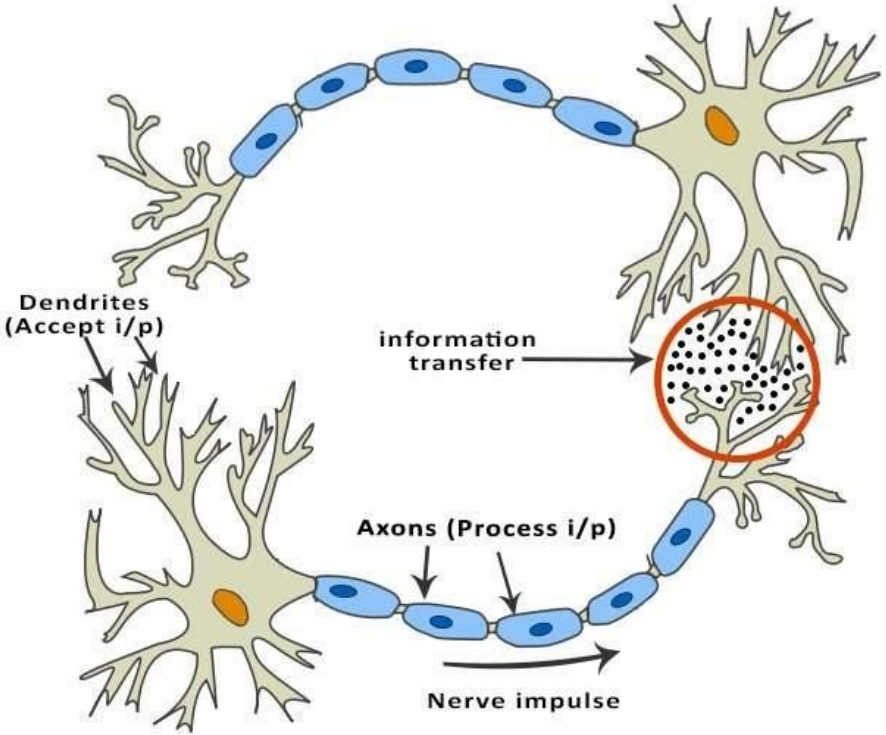
Искусственный интеллект и Software 2.0

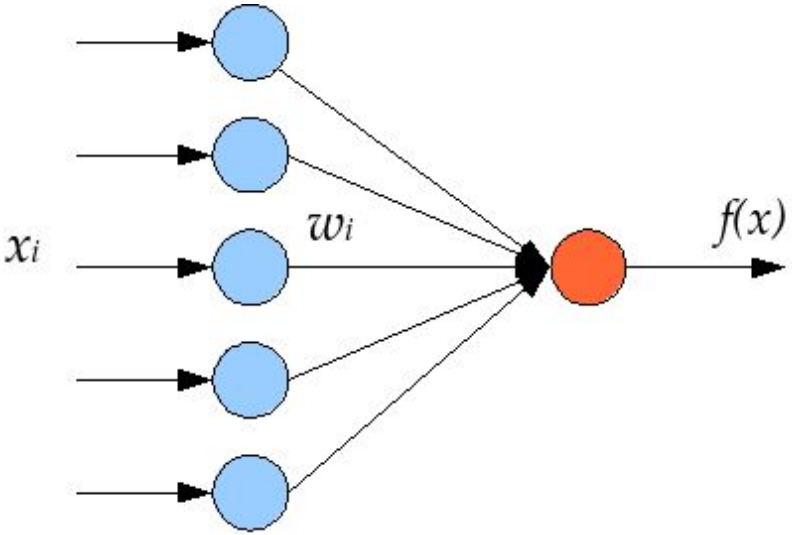
Рассадин Александр

Cherry Labs | ЛАТАС | ПОМИ РАН

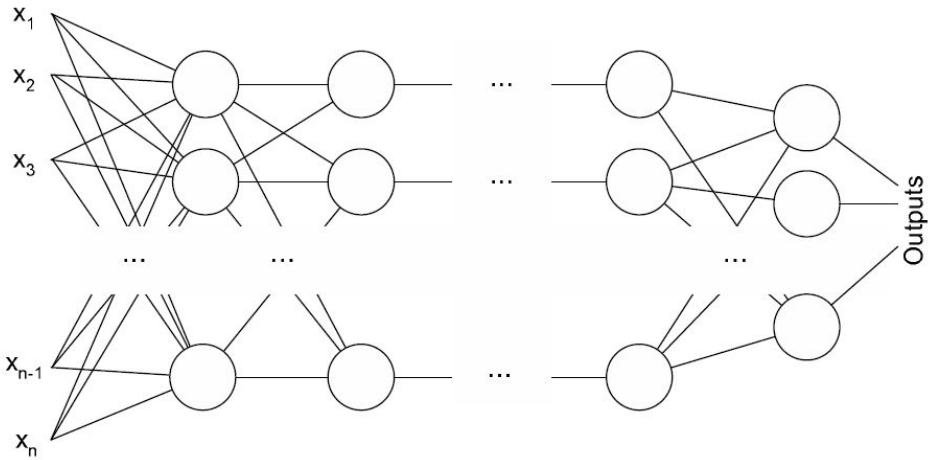


Концепция нейронных сетей была «придумана» еще в первой половине прошлого века и являлась попыткой частично смоделировать поведение мозга.

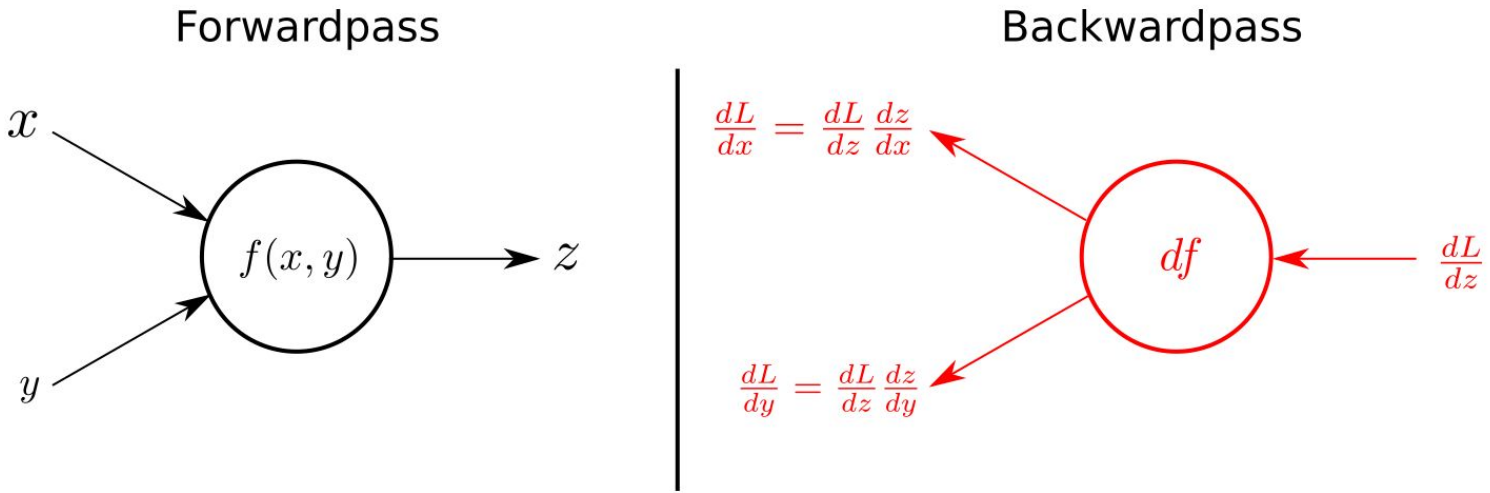
Мозг — это «естественная» нейронная сеть. В ней сигнал распространяется от узла к узлу - от *дендрита* к *дендриту* через *аксоны*.



McCulloch, Pitts. *"A logical calculus of the ideas immanent in nervous activity"*, 1943



Rosenblatt / Rumelhart. *Концепция многослойного перцептрона, 1960-е*



Rumelhart, Hinton, Williams. *"Learning representations by back-propagating errors", 1986*

1974 / 1982: сеть Хопфилда

1986: Rumelhart, Hinton, Williams. "Learning representations by back-propagating errors"

1943: McCulloch, Pitts. “A logical calculus of the ideas immanent in nervous activity”

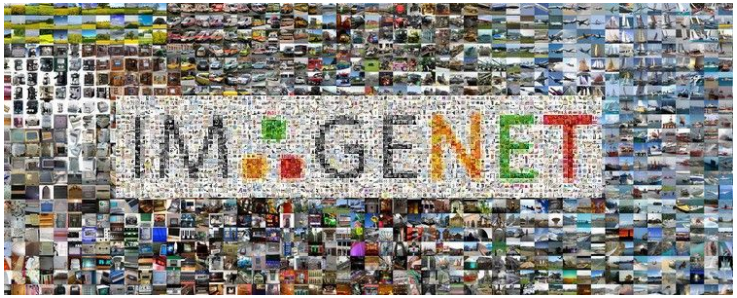
1989: LeCun, Boser, Denker, Henderson, Howard, Hubbard, Jackel. Backpropagation Applied to Handwritten Zip Code Recognition

Rosenblatt / Rumelhart. Концепция многослойного перцептрона

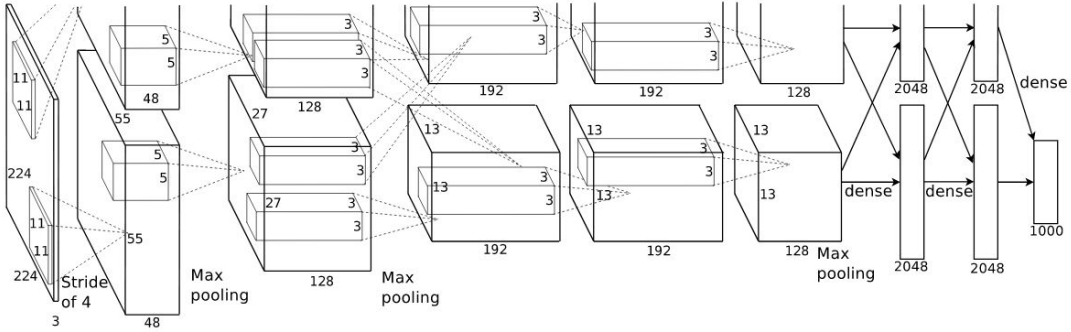
1993: Schmidhuber. Habilitation thesis: System modeling and optimization

1997: Hochreiter, Schmidhuber. “Long Short-Term Memory”

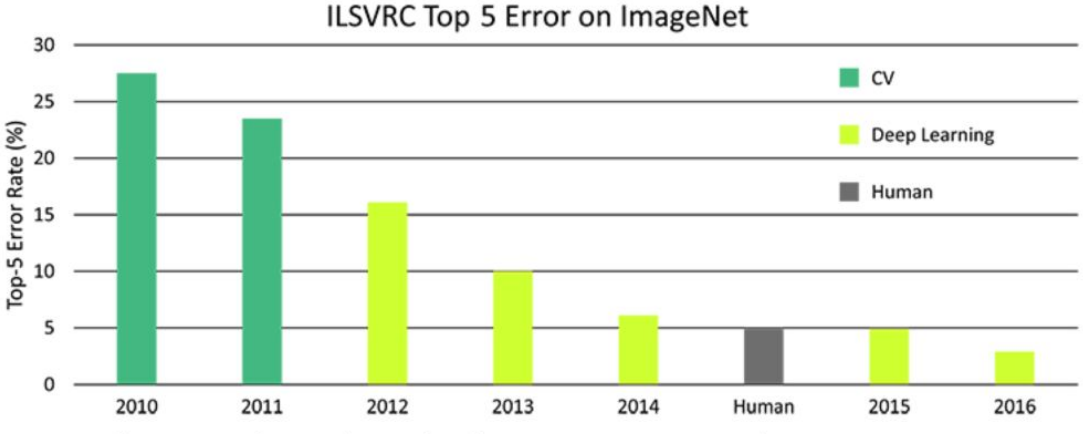
1998: LeCun, Bottou, Bengio, Haffner. Gradient-based learning applied to document recognition



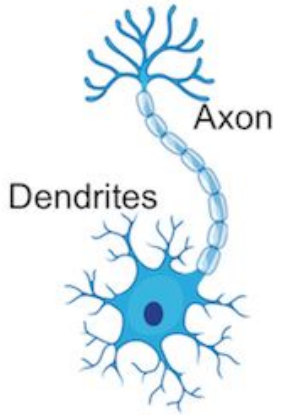
***ImageNet**: 14,197,122 изображения / 21,841 категории (конец 2017)*

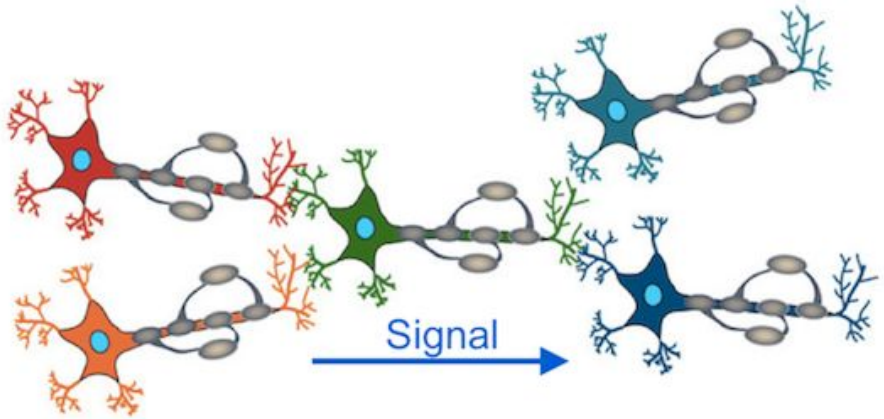


***Krizhevsky, Slutskever, Hinton.** ImageNet Classification with Deep Convolutional Neural Networks, 2012*

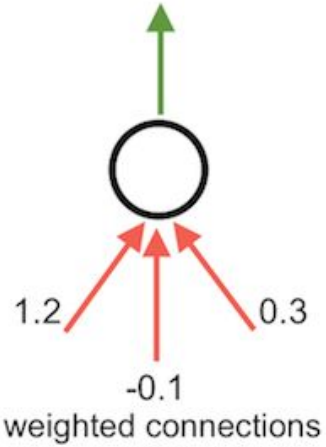


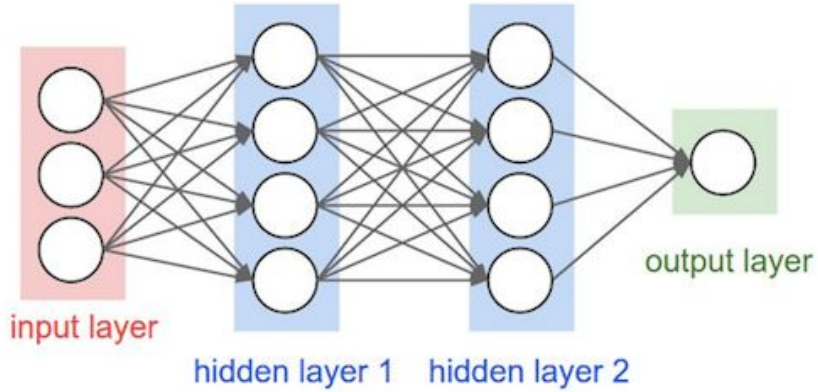
Biology



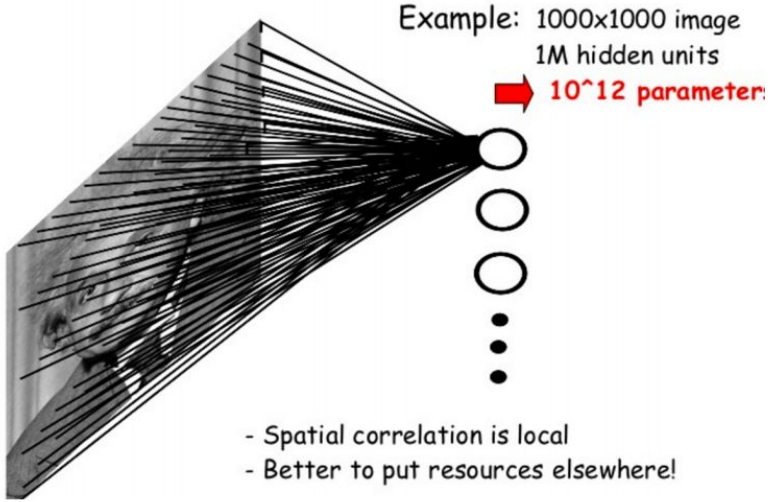


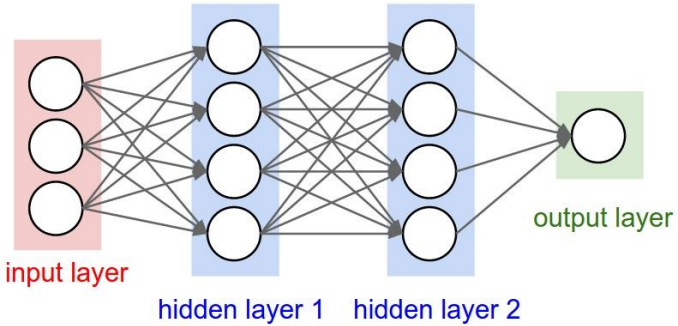
Deep learning



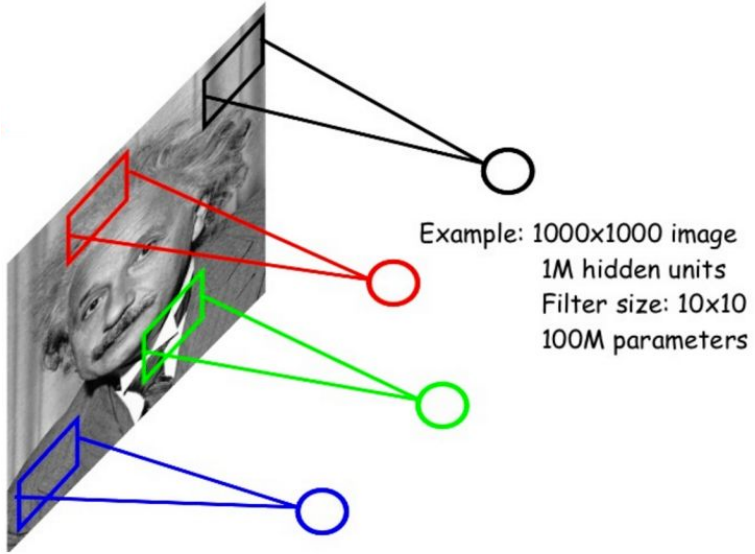


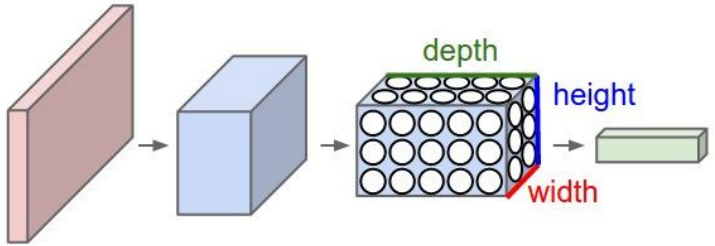
FULLY CONNECTED NEURAL NET

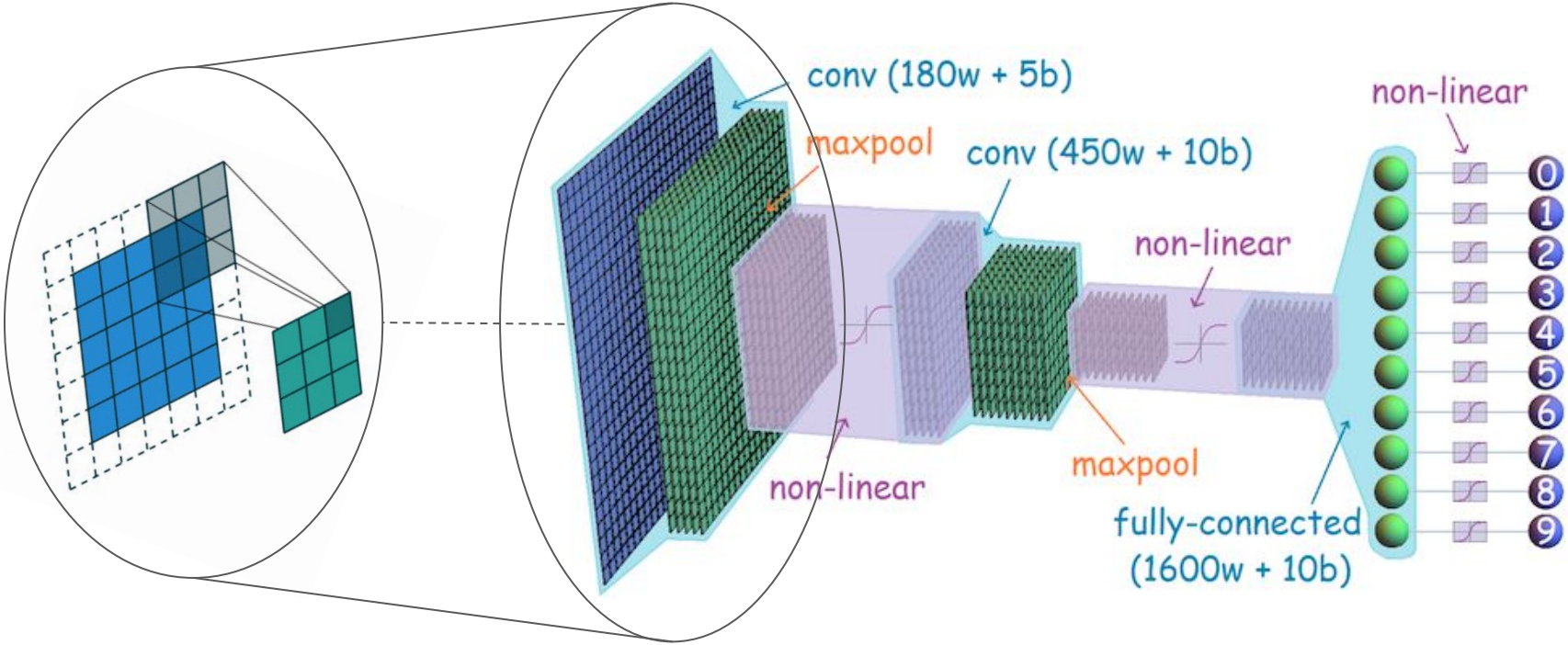


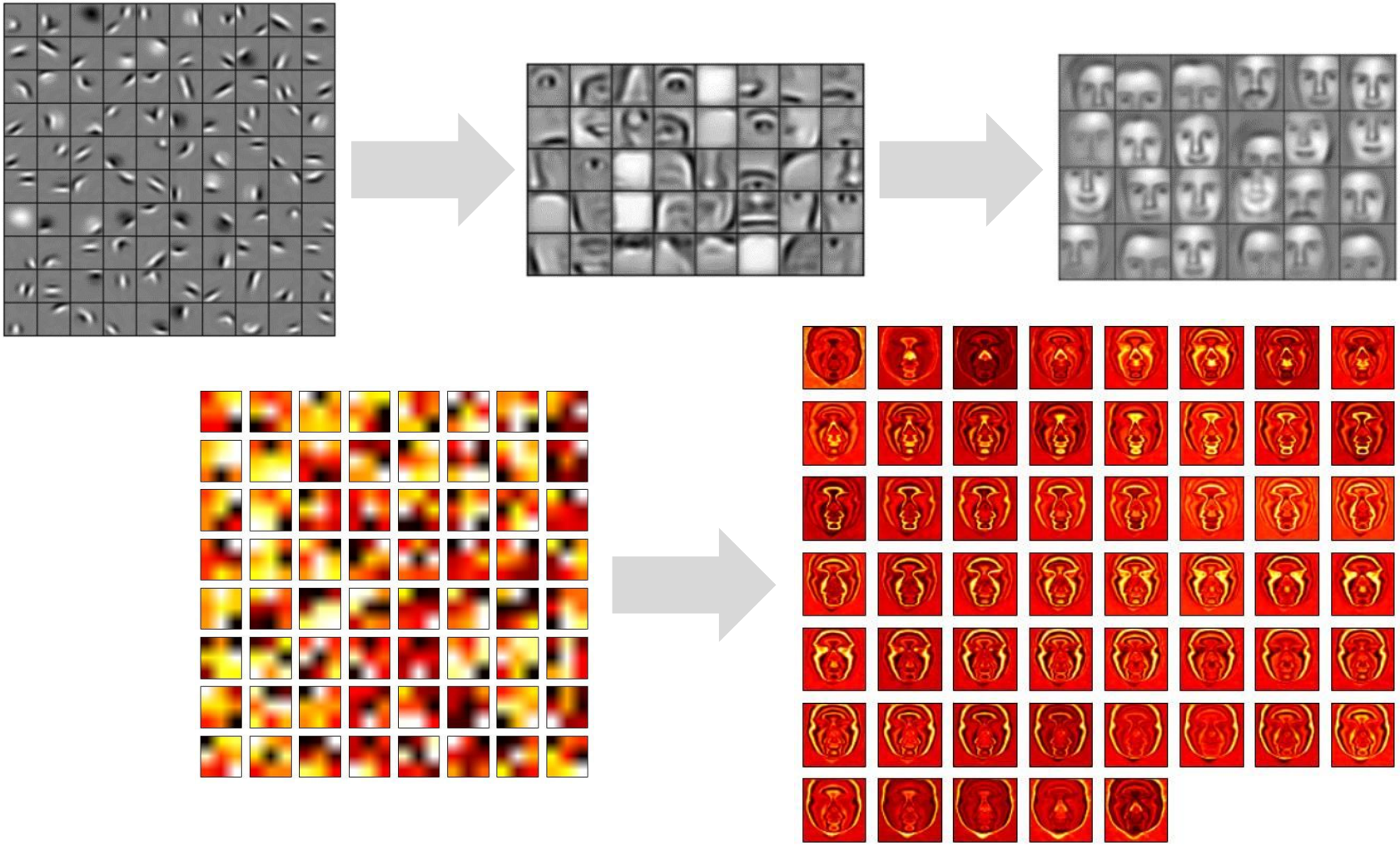


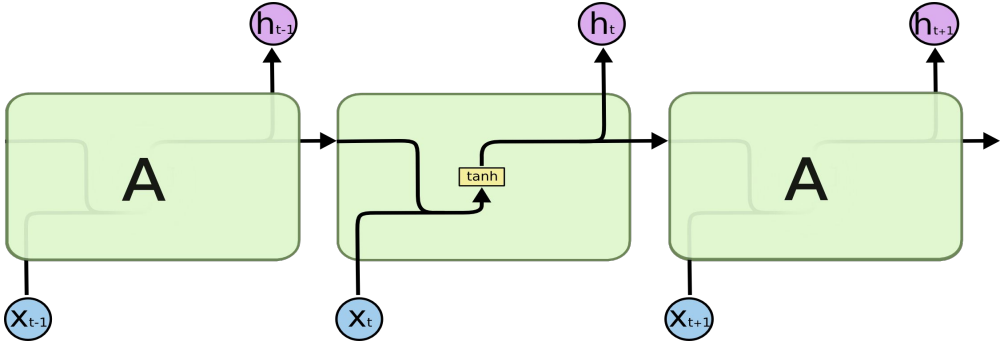
LOCALLY CONNECTED NEURAL NET



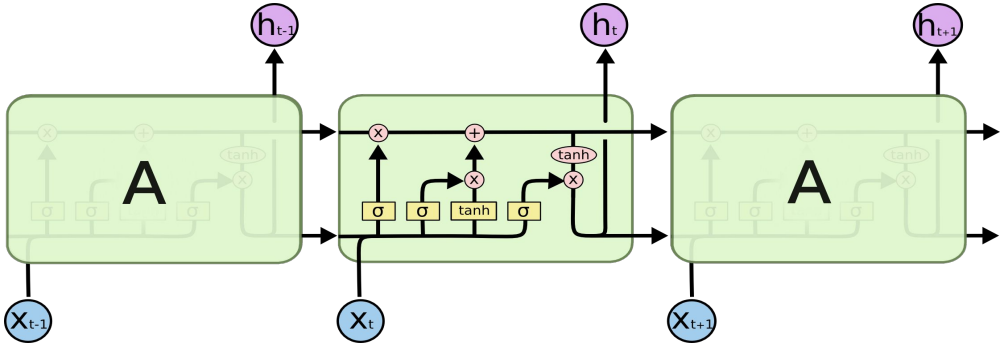



conv (180w + 5b)
maxpool
conv (450w + 10b)
non-linear
non-linear
maxpool
fully-connected
(1600w + 10b)
non-linear
0
1
2
3
4
5
6
7
8
9

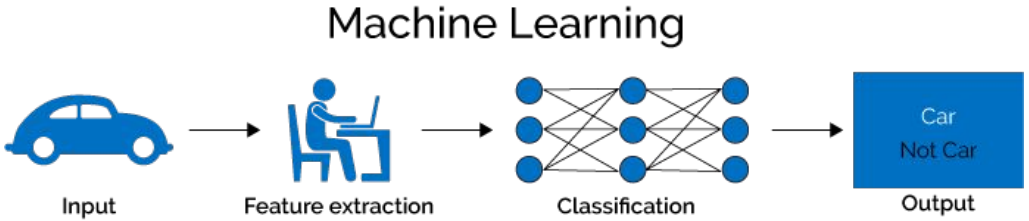


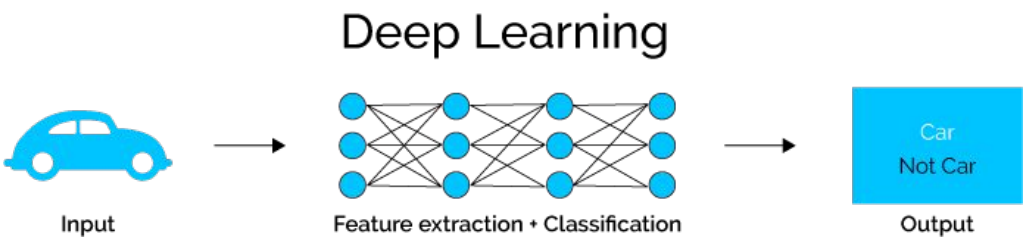


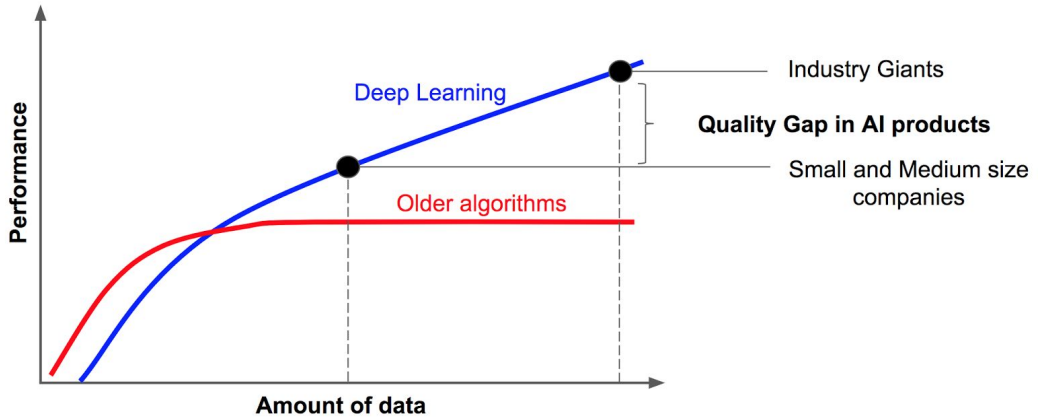
“обычная” рекуррентная ИНС

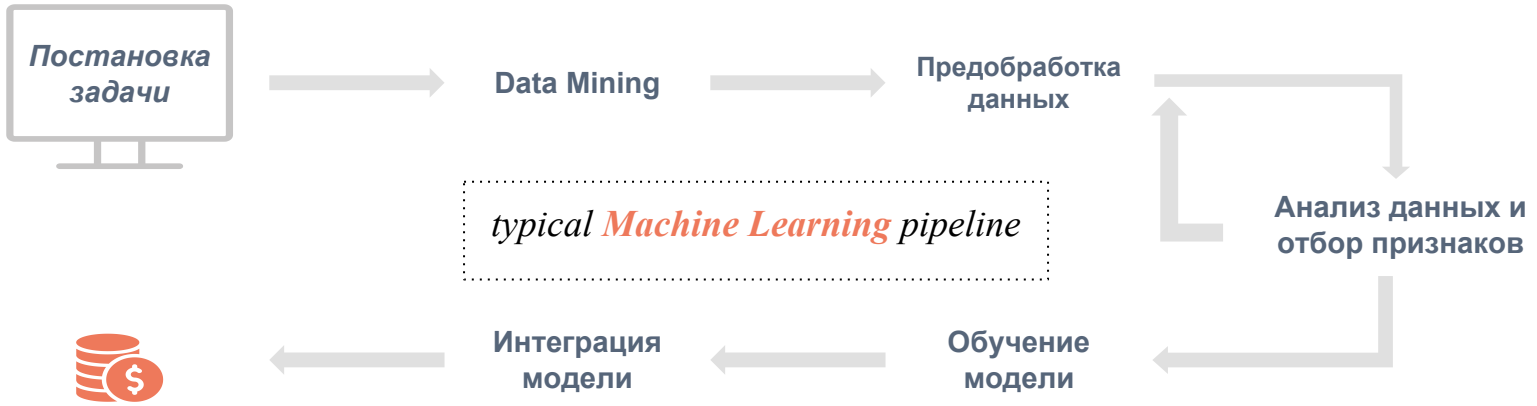


Long Short-Term Memory


Machine Learning
Input
Feature extraction
Classification
Car
Not Car
Output


Deep Learning
Input
Feature extraction + Classification
Car
Not Car
Output


Performance
Amount of data
Deep Learning
Older algorithms
Industry Giants
Quality Gap in AI products
Small and Medium size companies


Постановка задачи
Data Mining
Предобработка данных
Анализ данных и отбор признаков
typical Machine Learning pipeline
Обучение модели
Интеграция модели


Постановка задачи
Разметка данных
Выбор архитектуры
Обучение модели
typical Deep Learning pipeline
Оптимизация модели
Интеграция модели

Andrej Karpathy: Software 2.0

- *Однородные вычисления*
- *Проще “запечь” в кремнии*
- *Постоянное время работы*
- *Постоянное использование памяти*
- *Хорошая переносимость*
- *Хорошая гибкость*
- *Модули могут оптимально объединяться в одно целое*
- *Его легко изучать*
- *Она лучше тебя*

Open Source в эру Software 2.0

Software	Software 2.0
Open	
• code	• knowledge (papers) • model and weights • training code == reproducibility • runtime code
Proprietary	
• binaries	• model and weights • runtime code

WHAT IS A.I.?

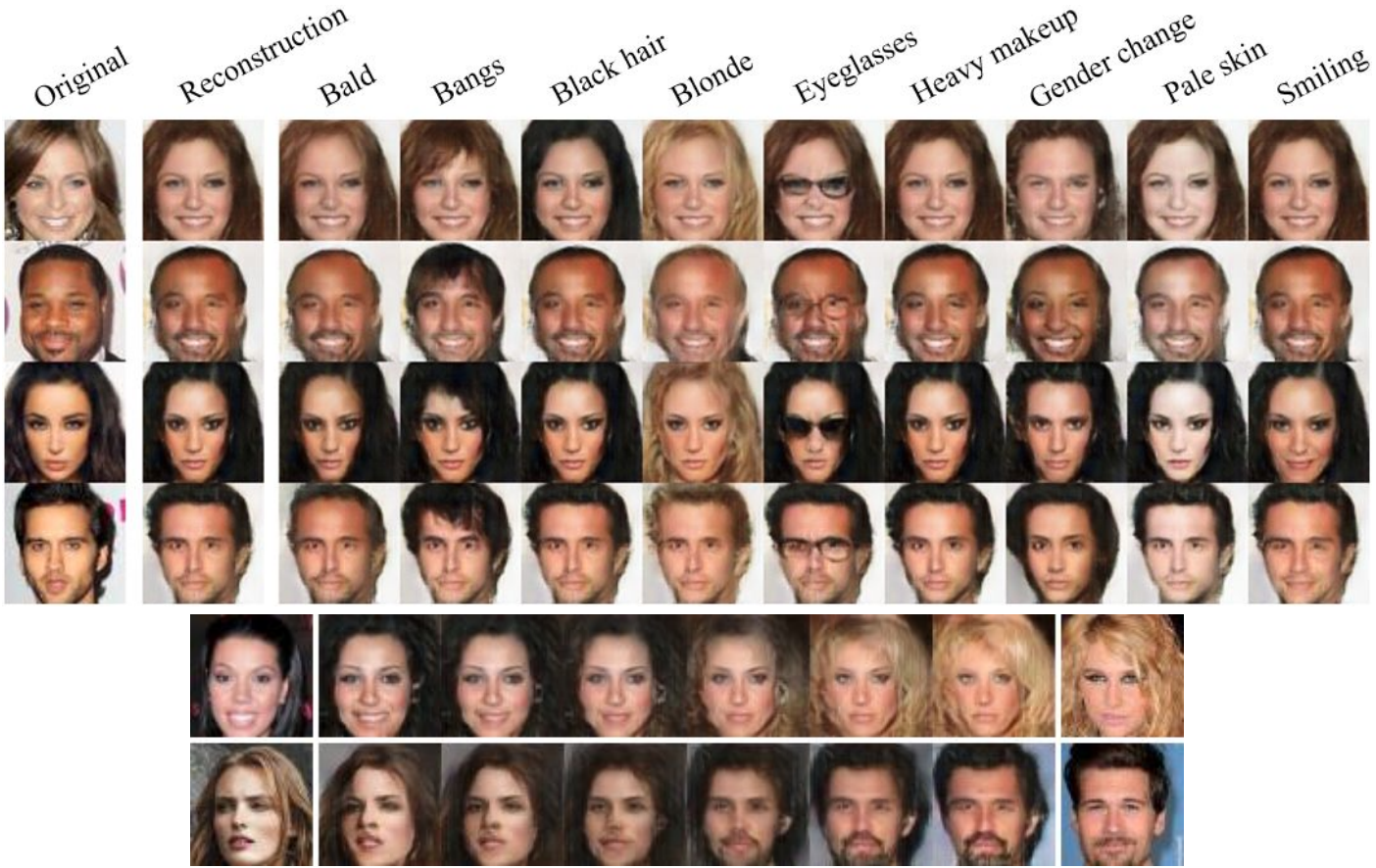
- умеет решать широкий круг задач
- решает их хорошо (не хуже человека)
- способен развиваться

(e) Bicubic (f) SRCNN (g) A+ (h) RAISR

Romano, Isidoro, Milanfar. *RAISR: Rapid and Accurate Image Super Resolution, 2016*

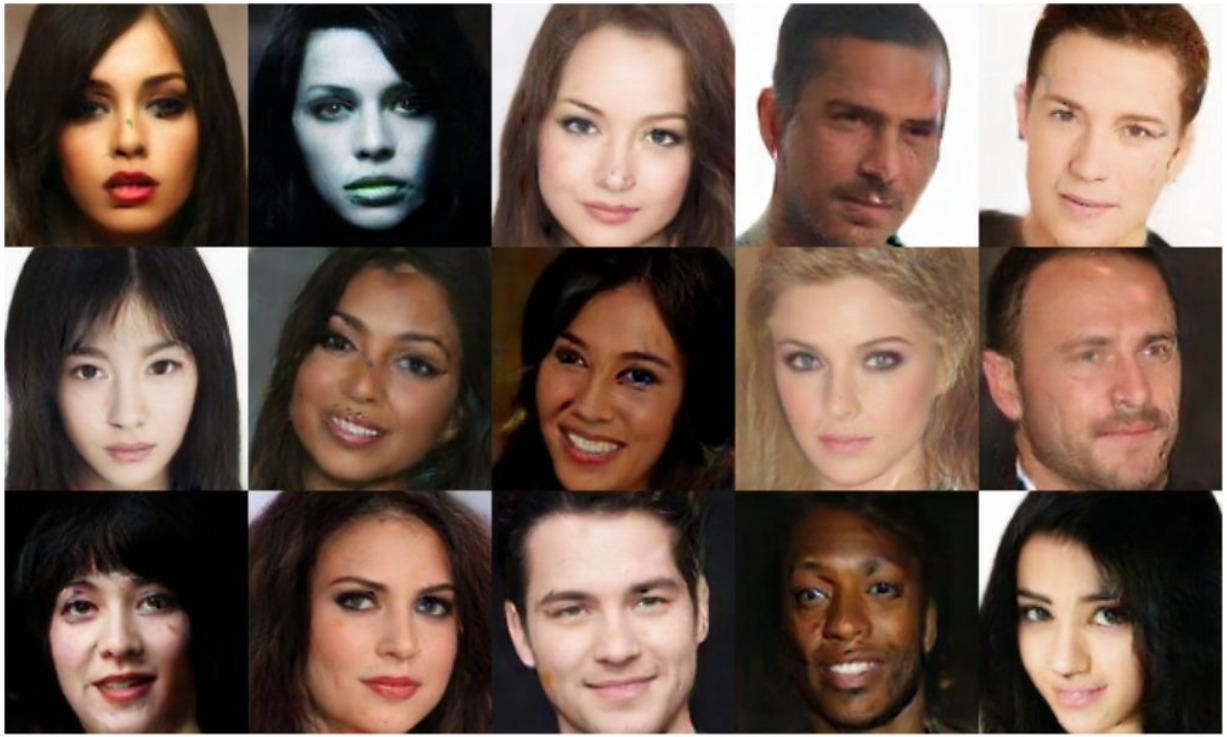
Gharbi, Chen, Barron, Hasinoff, Durand. *Deep Bilateral Learning for Real-Time Image Enhancement, 2017*

- ***Google.*** *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, 2016*
- ***Assael, Shillingford, Whiteson, Freitas.*** *LipNet: End-to-End Sentence-level Lipreading, 2016*
- ***DeepMind.*** *WaveNet: A Generative Model for Raw Audio, 2016*
- ***Micro$oft.*** *Achieving Human Parity in Conversational Speech Recognition, 2016*
- ***Liu, Wen, Yu, Li, Raj, Song.*** *SphereFace : Deep Hypersphere Embedding for Face Recognition, 2017*
- ***Deng, Guo, Zafeiriou.*** *ArcFace: Additive Angular Margin Loss for Deep Face Recognition, 2018*



Perarnau, van de Weijer, Raducanu, Álvarez. *Invertible Conditional GANs for image editing, 2016*

Berthelot, Schumm, Metz. *BEGAN: Boundary Equilibrium Generative Adversarial Networks, 2017*

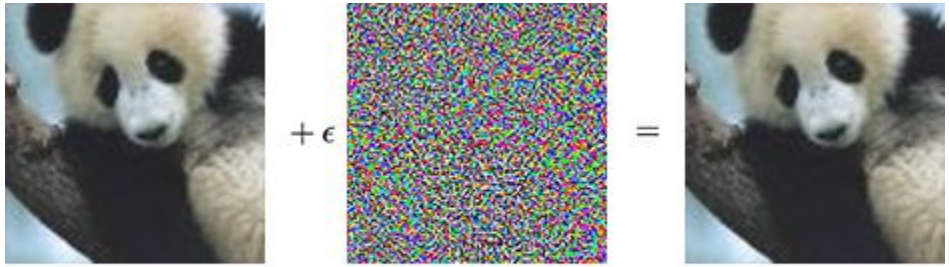


Berthelot, Schumm, Metz. *BEGAN: Boundary Equilibrium Generative Adversarial Networks, 2017*

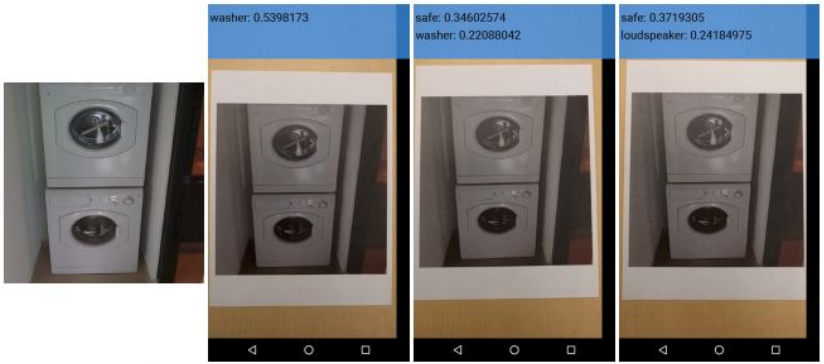
Antipov, Baccouche, Dugelay. *Face Aging With Conditional Generative Adversarial Networks, 2017*

Zhu, Park, Isola, Efros. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks, 2017*

Boston Dynamics



Goodfellow, Shlens, Szegedy. *Explaining and Harnessing Adversarial Examples, 2014*



Goodfellow, Shlens, Szegedy. *Explaining and Harnessing Adversarial Examples, 2014*

Эра Software 2.0: How-To

- **Python** ecosistem: **numpy**, **scipy**, **sklearn**
- своя **GPU** или в облаке: Vectordash, FloydHub, M$ Azure, GPUEATER
- Курсы и книги: **Coursera** (Andrew Ng), **"The Deep Learning Book"**, csXXX (cs231, cs224), Deep Learning by Google, Deep Learning for Coders
- Фреймворк: **Keras**, **Caffe**, **pyTorch**, **Tensorflow**
- Community: **Open Data Science**, **Kaggle**, **Medium**
- **Kaggle** и хакатоны

- Keras Model Zoo
- Tensorflow Model Zoo
- Caffe Model Zoo
- GitHub
- Age and Gender recognition example

```
from keras.applications.resnet50 import ResNet50
from keras.preprocessing import image
from keras.applications.resnet50 import preprocess_input, decode_predictions
import numpy as np

model = ResNet50(weights='imagenet')

img_path = 'elephant.jpg'
img = image.load_img(img_path, target_size=(224, 224))
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)

preds = model.predict(x)
# decode the results into a list of tuples (class, description, probability)
# (one such list for each sample in the batch)
print('Predicted:', decode_predictions(preds, top=3)[0])
# Predicted: [(u'n02504013', u'Indian_elephant', 0.82658225), (u'n01871265', u'tusker', 0.1122357),
```

```
from keras.applications.inception_v3 import InceptionV3
from keras.preprocessing import image
from keras.models import Model
from keras.layers import Dense, GlobalAveragePooling2D
from keras import backend as K

base_model = InceptionV3(weights='imagenet', include_top=False)

x = base_model.output
x = GlobalAveragePooling2D()(x)
# let's add a fully-connected layer
x = Dense(1024, activation='relu')(x)

predictions = Dense(200, activation='softmax')(x)

model = Model(inputs=base_model.input, outputs=predictions)

for layer in base_model.layers:
    layer.trainable = False

model.compile(optimizer='rmsprop', loss='categorical_crossentropy')

model.fit_generator(...)

for layer in model.layers[:249]:
   layer.trainable = False
for layer in model.layers[249:]:
   layer.trainable = True

from keras.optimizers import SGD
model.compile(optimizer=SGD(lr=0.0001, momentum=0.9), loss='categorical_crossentropy')

model.fit_generator(...)
```

Спасибо за внимание!

Мои контакты

Email

work@arassadin.ru

LinkedIn

linkedin.com/u/AlexandrRassadin