Introduction to Deep Learning through PyTorch

Nemanja Mićović


Универзитет у Београду
Математички факултет
МАТФ

MATF - moduli

whoami

- Teaching assistant and PhD student at Faculty of Mathematics
- ML/AI Research scientist at Nordeus
- I enjoy:
  - Artificial intelligence
  - Machine learning
  - Education and teaching
  - GNU/Linux and open source community
  - Python
  - Epic and science fantasy
  - Video games!
  - Krav Maga

Organization RISK (Računarstvo i informatika studentski klub)

R/SK

- Organization for computer science that meets every 2-3 weeks
- Working on hot topics like:
  - video games development
  - artificial intelligence
  - android development
  - web programming
  - blockchain...
- Lectures come from both industry and academia

R/SK> (org. team)

Nemanja Mićović

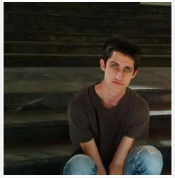
Organizacija, osnivač

nemanja_micovic@matf.bg.ac.rs

Pedja Trifunov

Organizacija

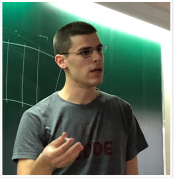
trifunov.pedja@gmail.com



Stevan Nestorović

Organizacija

stevannestorovic@hotmail.com



Vuk Novaković

Organizacija

vnovakovic96@gmail.com

Aleksandar Stefanović

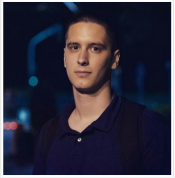
Organizacija

stefanoviczaleksandar@gmail.com

Olivera Popović

Društvene mreže

o.popovic@outlook.com



Filip Filipović

Video montaža

ffilipovicc98@gmail.com

Konstantin Klima

Organizacija

konstantin.klima@live.com

Ana Bolović

Organizacija

anabolovic98@gmail.com

Marko Šerer

Dizajner

sherer.marko@gmail.com

R/SK>

How to reach us

- instagram: @riskmatf - follow us
- web: risk.matf.bg.ac.rs
- github: @riskmatf
- youtube - Video of meetings are here

Machine learning

Machine learning

- Field of artificial intelligence
- Amazing results in the last 10 years
- Very attractive, dynamic and active field
- Devoted to systems who can improve their performance through time
  - Systems that **learn**
- Field consists of:
  - Supervised learning
  - Unsupervised learning
  - Semi-supervised learning
  - Reinforcement learning

Where is ML applied today?

- Autonomous driving
- Bioinformatics
- Social networks
- Algorithm portfolio
- Playing video games
- Image classification
- Recognizing harndwritting
- Natural language processing
- Generating optimization algorithms
- Generating images
- Computer vision
- Detecting frauds
- Data mining
- Medical use
- Marketing and targeted marketing
- Robotics
- Economy
- Speech recognition
- Speech synthesis
- Recommender systems

Machine learning and deep learning

Artificial Intelligence

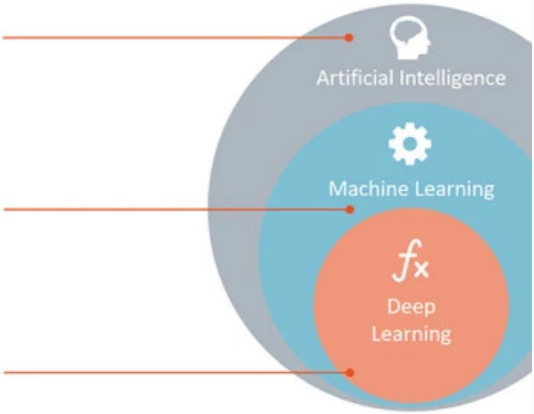
Any technique which enables computers to mimic human behavior.

Machine Learning

Subset of AI techniques which use statistical methods to enable machines to improve with experiences.

Deep Learning

Subset of ML which make the computation of multi-layer neural networks feasible.



Deep learning

- Field of ML
- Focused on neural networks
- Some of most amazing results come from here
- Deep learning is a **subset** of machine learning
- Often distinction isn't made due to marketing purposes
- Even worse, **deep learning** is used as a synonym for **AI**

AlphaGo

- *Lee Sedol*, world champion in game of Go losses from system AlphaGo (Google) 2015.
- Google used 1920 CPUs and 280 GPUs (per some reports)
- Game of Go is very complex
- It wasn't expected that AI would be able to conquer it for some time

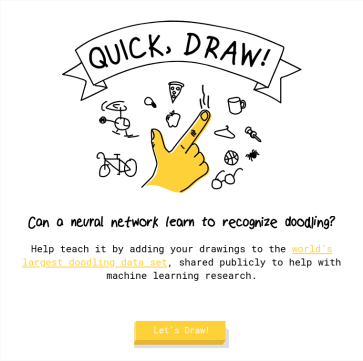
AlphaZero

- Continuation of algorithm AlphaGo 2015.
- Trained to play against itself
  - Defeated AlphaGo in Go with 60:40
- It can also play games of chess and shogi
- Currently is considered the best AI systems for go, chess and shogi

Autonomous driving

- Tesla: video
- Waymo: video
- Waymo taxi: video

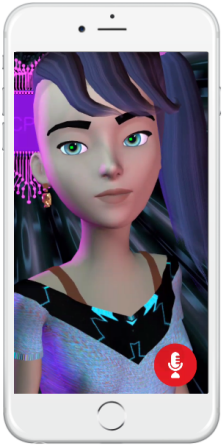
Google QuickDraw



- Sistem recognizing user drawn objects
- live demo

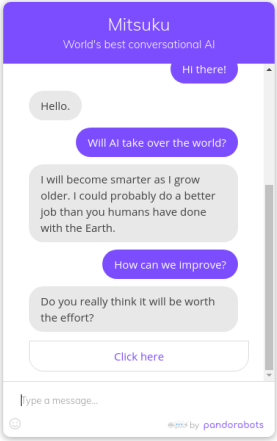
DeepFake (terminator video)

Mitsuku (online demo)

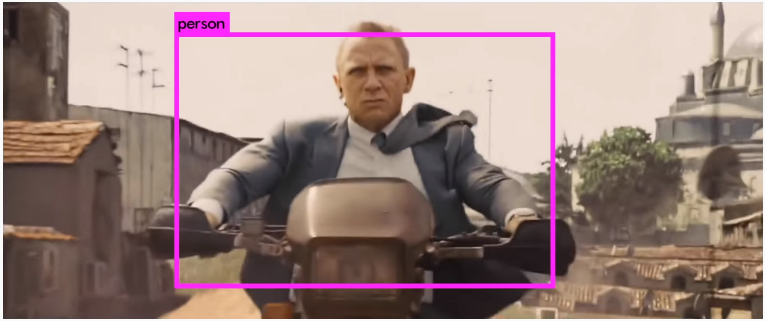


Meet Mitsuku

Mitsuku, a record breaking five-time winner of the Loebner Prize Turing Test, is the world's best conversational chatbot. As featured in the New York Times, Wall Street Journal, BBC, Guardian, Wired, and more.

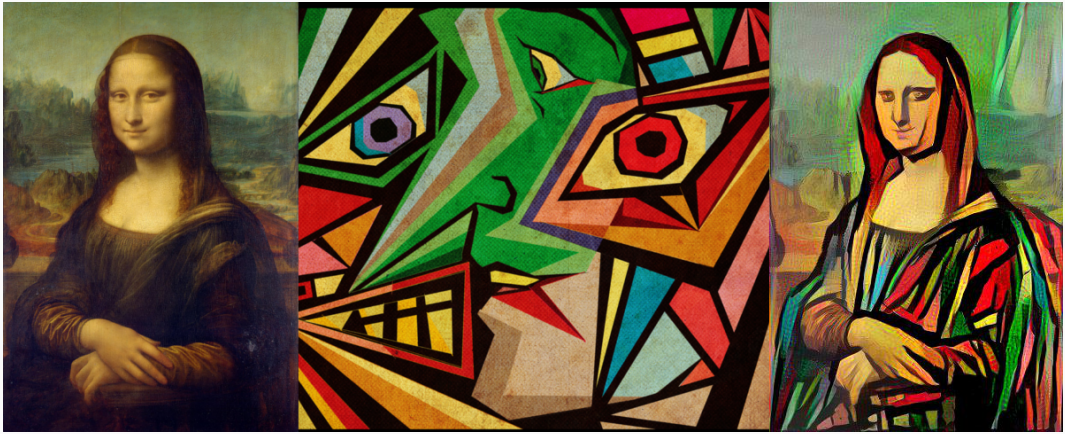


Detecting objects



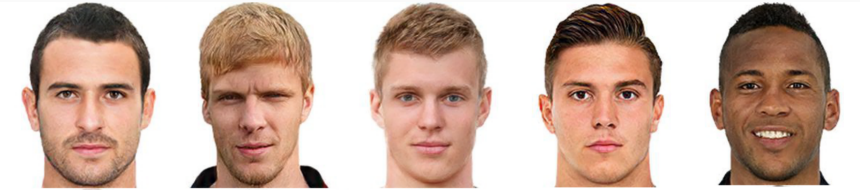
- Detecting objects - video
- Detecting objects - video
- James Bond - video

Style transfer (video)



Generating images (faces)

- Example shown by Nordeus at Machine learning seminar at MATF
- Which face is generated?



Writing poetry - real VS generated

Sveti Jovan od zemlje na noge,
Sve pod njima konja privatiše,
Pod Stjepana grada bijeloga,
Pa podiže sirotinja rodila,
Pa pogubi pod svoje postajemo,
ne bi li me provizur-Mijkom.
Kad su bili na noge lagane.

Sveti Jovan otisnu jabuku,
Ona pade moru u dubine,
Tople su ga suze propanule,
No mu care riječ progovara:
"A ne plači, dragi pobratime!
"Ne moj mene ugrabit' korunu,
"Ja ću tebe izvadit' jabuku."

Neural networks

Neural networks

- Universal function approximators
- Basic building blocks of many ML algorithms
- Not so new, some first versions shown in '60s and '70s of previous century
- Inspired with the way our brain works

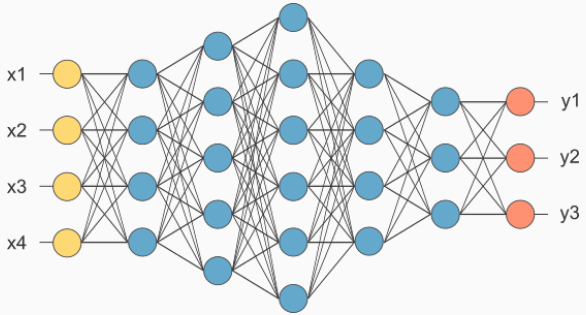


Figure 1: Arhitektura neuronske mreže.

Neuron of a neural network

Terminology:

- Activation: a_i
- Neuron weight: w_i
- Bias: b
- Non linear function: g
- Output is calculated as:

$$a_{out} = g(b + \sum_{i=1}^{N} a_i w_i)$$

Neuron of a Neural network

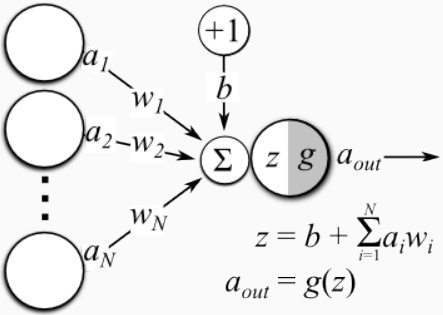


Figure 2: Neuron illustration

Activation function of a neural nework

- Very important to apply non linear transformation
  - Otherwise, function will stay linear
- Some popular activation functions:
  - ReLU: $g(x) = max(0, x)$
  - Sigmoid function: $g(x) = \frac{1}{1+e^{-x}}$
  - Tangens hyperbolic $g(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$
  - Lots of other ones, LeakyReLU, SinReLU, SeLU... z ## Training neural networks

Training neural networks

- Training is done with variants of (stochastic) gradient descent
- Weights are updated so that overall loss L of model on data is **reduced**
- Stochasticity is introduced to improve training speed
  - Reason is that gradient isn't calculated on **all** of data, but on a **subset**
  - This subset is called *batch*
- Term *epoch* is used to denote 1 run through the training data

Gradient descent

$$w_{i+1} = w_i - \alpha \cdot \nabla L$$

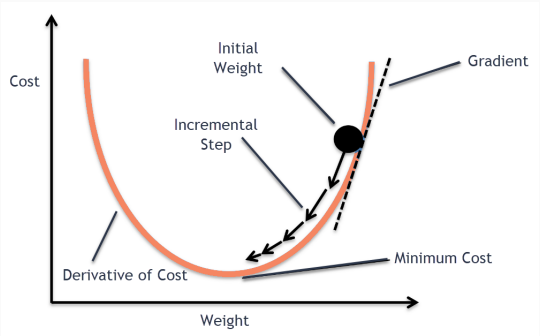


Figure 3: Visualizing gradient descent

Training neural networks

Backpropagation

- Calculate the function gradient relative to the weights of neurons
- Is a basic building block for training neural networks
- GPUs allow for extreme parallelization of calculations
  - They are one of the main reasons for explosion of the field
  - Also, a great excuse to buy a powerful GPU that can run modern video games :)

Convolutional neural networks (CNN)

- Type of neural networks
- Leading approach for image classifications
- Devoted to signal processing in where there is space locality (images, audio, video)
- CNNs are constructing new attributes from the input
- Very extensively used in computer vision
- Complexity of constructed attributes increases with the depth of network
- Contain partial interpretability for their work

Convolutional neural networks (CNN)

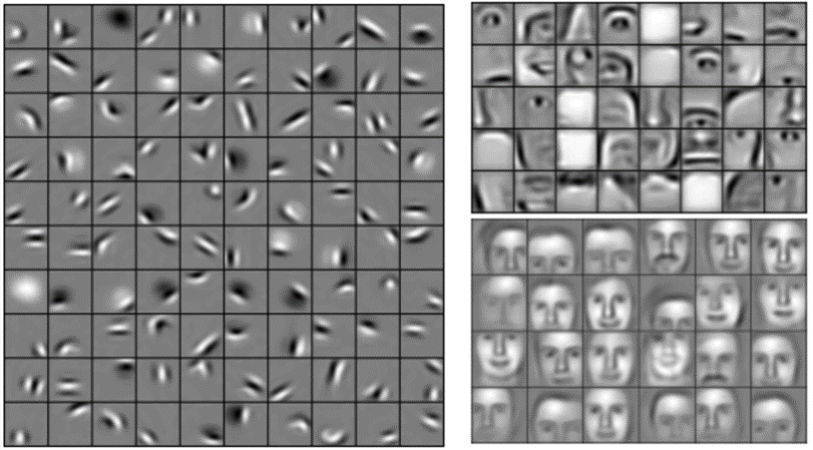


Figure 4: Visualizing learned filters.

Convolutional neural networks (CNN)

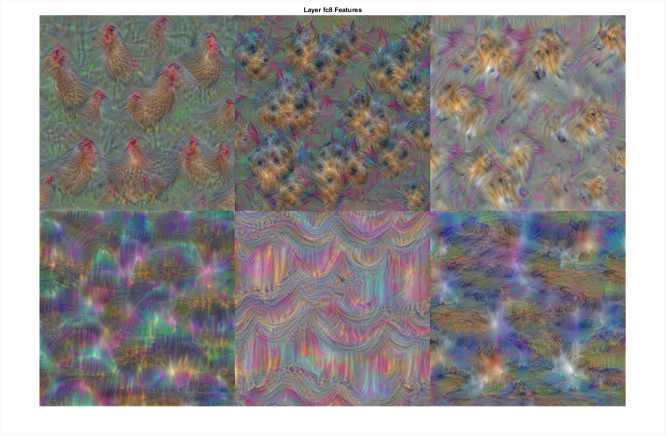


Figure 5: Visualizing learned filters.

Convolutional neural networks (CNN) - architecture

Architecture is mostly a combination of following elements:

- Convolution layer
- Aggregation layer
- Fully connected layer

In the last few years:

- Skip connections (He et al. 2015)
- Inception module (Szegedy et al. 2014)

Convolutional neural networks (CNN) - architecture

Convolution layer:

- Detects a certain template in data
- For example, detect horizontal or vertical lines (lower layers)
- Or detect eyes and ears (higher layers)

Aggregate layers (eng. pooling):

- Aggregate information from previous layers (mostly convolutional layers)
- As aggregating function mostly *max* or *average* is used

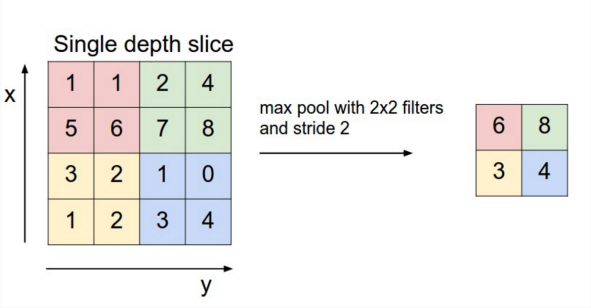


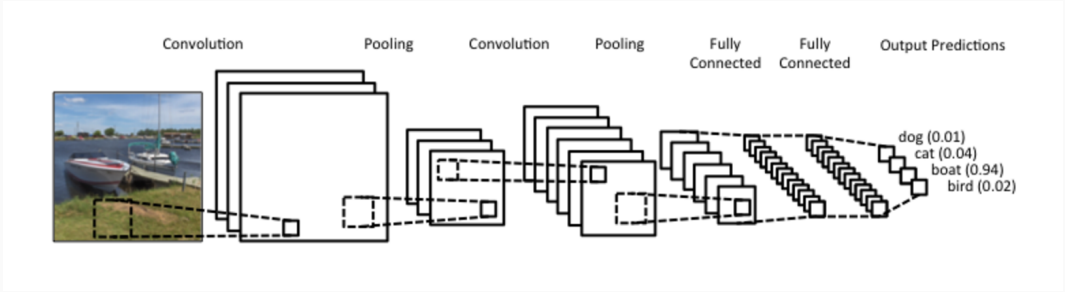
Figure 6: Aggregation with maximum function.

Convolutional neural networks (CNN) - architecture

Fully connected layer:

- Mostly used in the last few layers in a CNN to build a reggresor or classifier

Convolutional neural networks (CNN) - example



Convolutional neural networks (CNN) - example

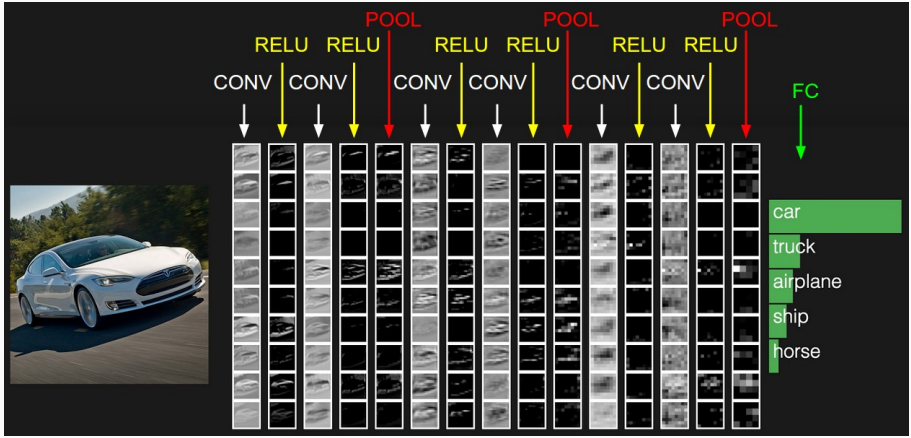
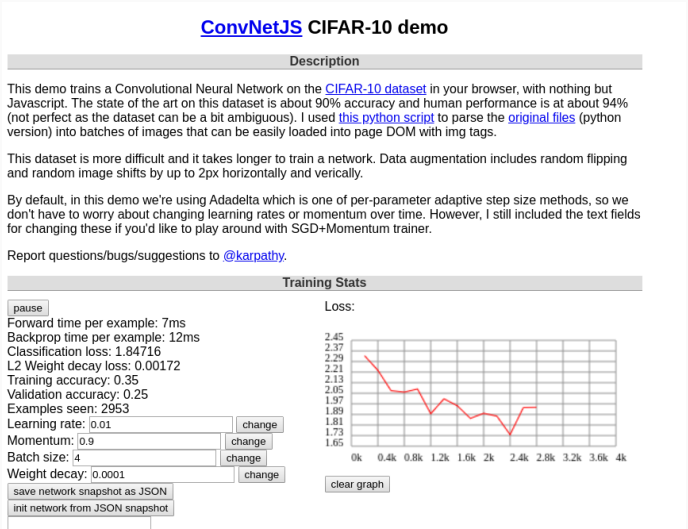


Figure 7: CNN exmaple.

Convolutional neural networks (CNN)

- Nice interactive example: ovde



ConvNetJS CIFAR-10 demo

Description

This demo trains a Convolutional Neural Network on the CIFAR-10 dataset in your browser, with nothing but Javascript. The state of the art on this dataset is about 90% accuracy and human performance is at about 94% (not perfect as the dataset can be a bit ambiguous). I used this python script to parse the original files (python version) into batches of images that can be easily loaded into page DOM with img tags.

This dataset is more difficult and it takes longer to train a network. Data augmentation includes random flipping and random image shifts by up to 2px horizontally and verically.

By default, in this demo we're using Adadelta which is one of per-parameter adaptive step size methods, so we don't have to worry about changing learning rates or momentum over time. However, I still included the text fields for changing these if you'd like to play around with SGD+Momentum trainer.

Report questions/bugs/suggestions to @karpathy.

Training Stats

pause

Forward time per example: 7ms
Backprop time per example: 12ms
Classification loss: 1.84716
L2 Weight decay loss: 0.00172
Training accuracy: 0.35
Validation accuracy: 0.25
Examples seen: 2953

Learning rate: 0.01 change
Momentum: 0.9 change
Batch size: 4 change
Weight decay: 0.0001 change

save network snapshot as JSON
init network from JSON snapshot

Loss:

2.45 2.37 2.29 2.21 2.13 2.05 1.97 1.89 1.81 1.73 1.65

0k 0.4k 0.8k 1.2k 1.6k 2k 2.4k 2.8k 3.2k 3.6k 4k

clear graph


PyTorch

PyTorch

- Library and framework for deep learning
- Continuation of the original *Torch* library for language *Lua*
- Primarily developed by AI team from *Facebook*
- PyTorch is *free* and *open source*
- Primary used from language `Python`, but there is also an **experimental** C++ support
  - If you really need C++, `TensorFlow` is recommended

PyTorch: Computation graph

Computation graph

- Neural networks represents an approximation of some function
  - Neural network is also a *complex* function
  - For example, function gets an image as input, and gives a probability of a person smiling or not
- To represent this function easier, we used a computation graph instead of a formula
- Computation graph shows **how** is *input* is transformed onto *output*
- Shows how is the data transformed **during** computations

Computation graph

$$f(x, y, z) = (x + y) \cdot z$$

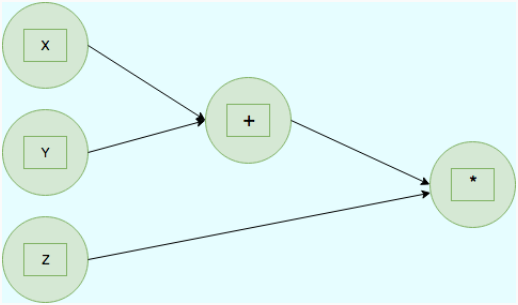


Figure 8: Computation graph of function f, source

Computation graph

- Using library as `PyTorch` we can define a computation graph
- Libraries are allowing us to perform these computations on GPUs
- This is very useful as this can reduce our work and reduce calculation time
- Lots of operations in NNs are matrix operations

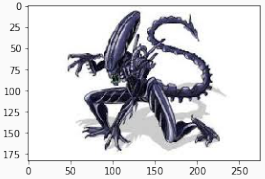
PyTorch: computation graph (code!)

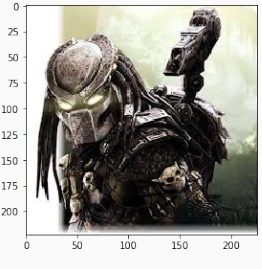
PyTorch: CNN (code!)

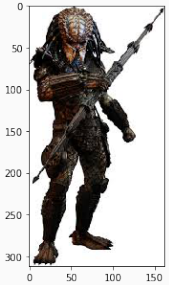
PyTorch: transfer learning

Predator vs Alien

Predator vs Alien







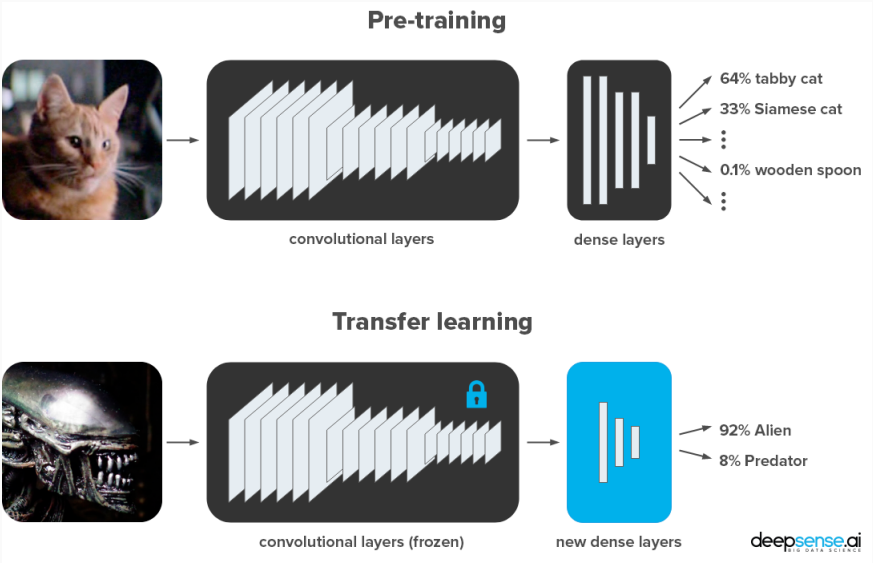
Predator vs Alien

- We wish to perform binary classification
- But, we only have 447 images per class
- CNNs often require big amount of data to train properly

Transfer learning

- Includes using and training an already *trained* model on some other data
- Idea is that learned filters in one problem are useful in some other *similar* problem
- There is not a *formal* algorithm for transfer learning, but some variants are:
  - Freeze convolutional layers, remove all fully connected, and put new fully connected layers
  - TODO
  - Koristiti konvolutivne slojeve neke mreže za dobijanje nove reprezentacije podataka, dalje raditi sa raznim drugim modelima
  - Zamrznuti deo konvolutivnih slojeva, obučiti ostatak
  - I slično...

Transfer učenja



Transfer učenja (code!)

Questions?

Literatura

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. “Deep Residual Learning for Image Recognition.” *CoRR* abs/1512.03385. http://arxiv.org/abs/1512.03385.

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. “Going Deeper with Convolutions.” http://arxiv.org/abs/1409.4842.