

DeepDream

Сањање и визуелизација конволутивних неуронских мрежа

Семинарски рад у оквиру курса

Истраживање података

Математички факултет

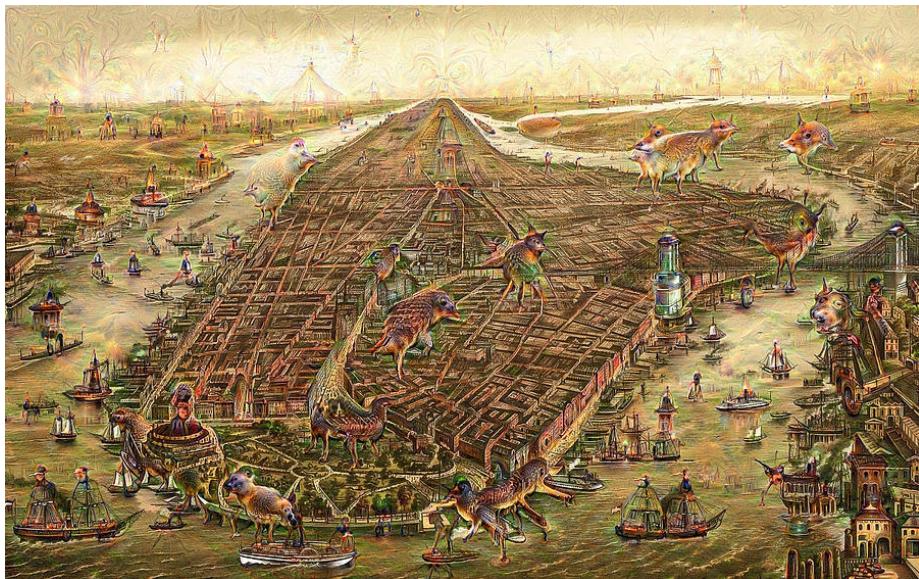
Немања Мићовић

nmicovic@outlook.com

Абстракт

Неуронске мреже у последњих неколико година добијају на значају услед ефикасне паралелизације њиховог тренинга што омогућава тренирање модела над великим скупом инстанци као и конструкцију архитектура које поседују велики број слојева. Иако постижу значајне резултате, интерпретабилност и разумевање добијеног модела остаје отворен и нерешен проблем.

Пројекат DeepDream за циљ има да визуализује објекте и шаблоне које је конволутивна мрежа научила. Добијена слика се потом користи у генерирању нових слика у којима се амплификују појмови које је мрежа научила, како би се не крају добиле слике које многе подсећају на снове.



Садржај

1 Увод	3
2 Неуронске мреже	3
2.1 Перцептрон	3
2.2 Неуронске мреже са пропагацијом унапред	3
2.3 Конволутивне неуронске мреже	4
2.3.1 Архитектура конволутивне мреже	5
2.3.2 Визуелизација конволутивне мреже	7
2.3.3 Такмичење ILSVRC	8
3 Deep dream	8
3.1 Генерисање слика	9
3.2 Програмски код за генерисање слика	11
4 Закључак	17
Литература	18

1 Увод

Неуронске мреже постижу значајне резултате на многим пољима вештачке интелигенције. Упркос томе, наше познавање њиховог интерног рада и интерпретабилности добијених модела, остају изненађujuће на ниском нивоу.

У делу 2 дат је кратак преглед неуронских мрежа, но услед сложености тематике, изложени материјал је више информативан него технички. За технички део се препоручују књиге [1, 2, 3].

Део 3 описује пројекат DeepDream. Секција 3.1 детаљније описује пројекат, а 3.2 приказује програмски код којим се неуронска мрежа може натерати да сања, односно омогућава генерирање слика које поседују објекте које је мрежа научила да препознаје.

2 Неуронске мреже

2.1 Перцептрон

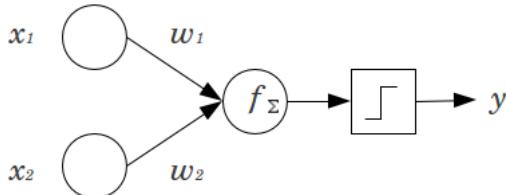
Перцептрон је први пример неурона који се појавио у области [4]. Оригинални перцептрон је представљао бинарни класификатор који је могао да изврши класификацију уколико су инстанце линеарно сепарабилне, а иначе тренинг не би успевао.

Класификација се врши формулом:

$$f(x) = \begin{cases} 1, & \text{ако } w \cdot x + b > 0 \\ 0, & \text{иначе} \end{cases}$$

Вектор реалних вредности означавамо са w и називамо га *тежине* (eng. *weights*), вредност b називамо пристрасност (eng. *bias*), а оператор \cdot означава скаларни производ, дефинисан као $\sum_{i=1}^m w_i x_i$. Вредност $f(x) = 1$ означава да је инстанца x класификована као *позитивна*, а вредност $f(x) = 0$ да је класификована као *негативна*.

Слика 1 приказује перцептрон за $m = 2$.



Слика 1: Пеџентрон

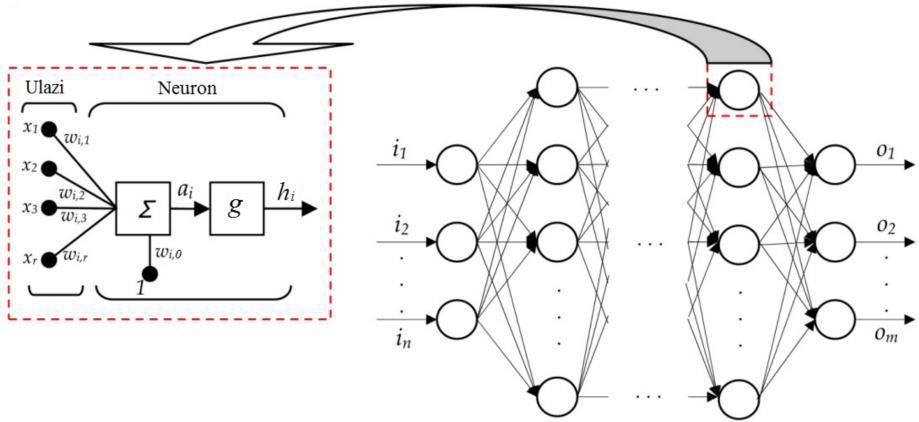
2.2 Неуронске мреже са пропагацијом унапред

Неуронска мрежа са пропагацијом унапред састоји се из више слојева који садрже *неуроне*. Слојеви између улаза и излаза се називају *скрипсни слојеви* (eng. *hidden layers*). Неурон представља рачунску јединицу сличну перцептрону. Оно што је кључно у неуронским мрежама јесте нелинеарност која се уводи код неурона, чиме се омогућава

да неуронска мрежа изврши класификацију и нелинеарно сепарабилних инстанци. Постоји и теорема која гарантује да неуронска мрежа са једним скривеним слојем може апроксимирати произвољну функцију.

Неуронске мреже могу имати више неурона на улазу и излазу и њихов број варира од архитектуре, скупа података и домена примене. Често су улази у неуронску мрежу атрибути и представљају делове улазних података (на пример број соба за одређивање цене куће). На слици 2 приказана је неуронска мрежа са n улаза и m излаза.

Тренинг неуронске мреже се врши алгоритмом пропагације уназад (eng. *backpropagation*). Више о алгоритму и математичкој формулатури о неуронским мрежама може се пронаћи у [1, 2, 3].



Слика 2: Неуронска мрежа са пропагацијом унапред

2.3 Конволутивне неуронске мреже

Идеја о конволутивним неуронским мрежама јавила се још раних десетдесетих [5], но праву популарност конволутивне мреже достиже у последњих неколико година услед могућности да се процес тренинга мрежа ефикасно паралелизује на графичким картицама. Успешна паралелизација је омогућила да се врши тренинг над милијардама података као и да се користе дубоке неуронске мреже са по неколико хиљада слојева. Популарност конволутивних мрежа нагло пораста, а сама тема постаје изузетно истраживачки популарна.

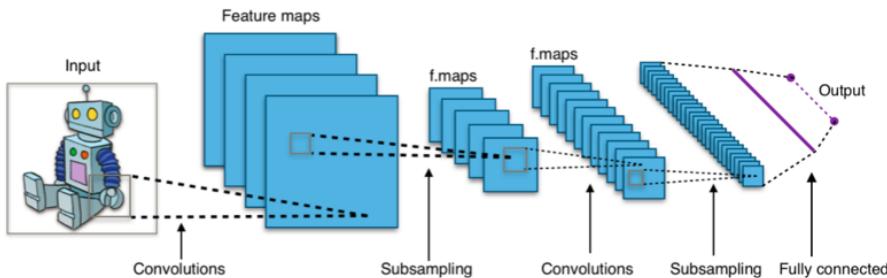
Конволутивне неуронске мреже постигле су изузетне резултате у проблемима попут класификације [6, 7, 8], а генерално се могу користити када је у питању нека врста улазног сигнала (слика, звук, видео). Коришћене су и за класификацију текста [9] а занимљива је и примена у раду [10].

Конволутивне мреже се могу применити на било какве податке који поседују структуралну информацију. Другачије речено, уколико се редослед података/сигнала једне инстанце перmutује, губи се суштина информације. На пример, уколико се перmutују пиксели слике, губи се информација као и сам објекат који је на слици.

Неурони конволутивне мреже уче да препознају неку врсту шаблона на основу тренинг скупа. Што је неурон дубље у мрежи, ша-

блон који препознаје постаје комплекснији. На примеру слике при препознавању лица, један од неурона у почетним слојевима би препознавао хоризонталну линију, а неурон при крају, очи или нос. Комбинација неурона омогућава да се дође до одговора шта је на слици. Често се на крају користи софтмакс техника (eng. *softmax*) како би се неурони на последњем слоју користили за оцену вероватноће класе којој инстанца припада [1].

Слика 5 приказује пример конволутивне мреже. У делу 2.3.1 ће детаљније бити образложени делови типичне архитектуре.



Слика 3: Пример архитектуре конволутивне неуронске мреже

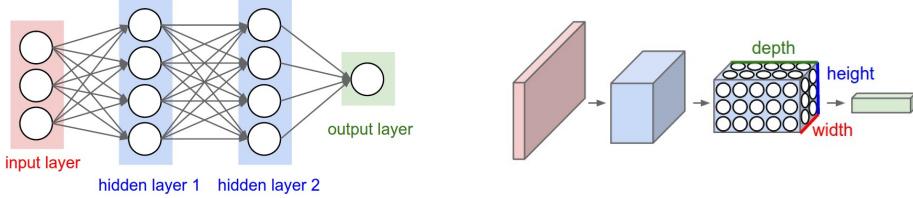
2.3.1 Архитектура конволутивне мреже

Архитектура конволутивне мреже се састоји из више слојева. Слојеви су најчешће:

- Потпуно повезани слој (eng. *fully connected*)
- Конволутивни слој (eng. *convolutional*)
- Слој агрегације (eng. *pooling*)

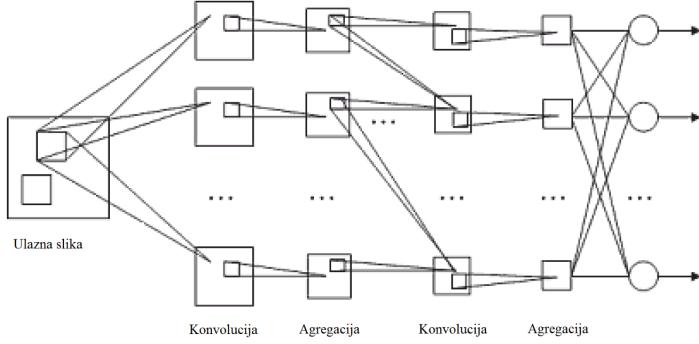
Потпуно повезани слојеви Потпуно повезани слој је сличан стандардним неуронским мрежама и често се користи на крају мреже заједно са софтмакс функцијом како би се оцениле вероватноће класе. Слике 4 и 5 приказују разлику између обичне неуронске мреже и конволутивне неуронске мреже, слика 6 приказује пример архитектуре једне конволутивне мреже, а слика 7 приказује мрежу VGG16 (eng. Visual Geometry Group) која је победила на такмичењу ILSVRC 2014 (видети део 2.3.3).

Конволутивни слојеви Конволутивни слој врши конструкцију нових атрибути који се потом прослеђују даље у мрежу [1, 11]. Често се на истом слоју дефинише више паралелних конволутивних слојева. Паралелни конволутивни слојеви могу на пример препознавати где су хоризонталне и верикалне линије, док ће се у наредним слојевима та информација користити да се препознају комплекснији облици. Конволутивни слој превлачи филтер преко (на пример) слике и тиме детектује присуство одређеног атрибута на слици. Овај процес се назива *конволуција*. Процесом тренинга атрибути које конволутивни слој тражи се уче. Величине филтера су типично мале, на пример 2×2 или 3×3 пиксела на примеру слика.

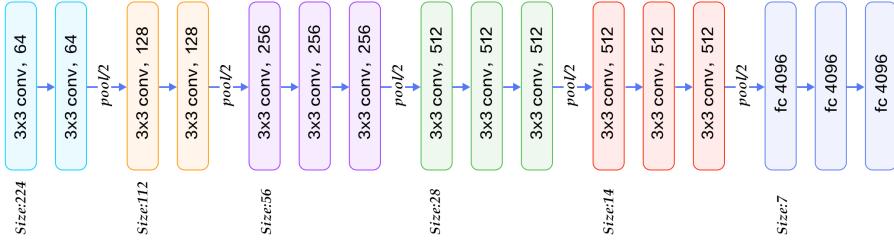


Слика 4: Неуронска мрежа

Слика 5: Конволутивна неуронска мрежа



Слика 6: Пример архитектуре конволутивне мреже

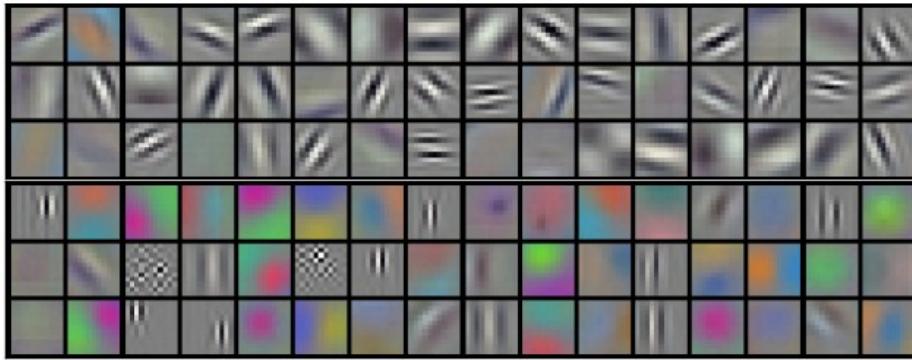


Слика 7: Мрежа VGG16

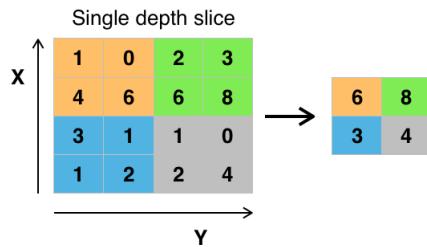
Слика 8 приказује филтере које је научила мрежа коришћена у раду [6].

Слој агрегације Агрегациони слој израчунава функцију укрупњавања над улазом, најчешће просек или максимум, и на тај начин укрупњава информације које добија из претходног слоја [11]. Често слоју агрегације претходи конволутивни слој. Уколико се улаз агрегира са 2×2 филтером, онда је излаз агрегационог слоја 2 пута мањи од свог улаза. На овај начин се смањује садржај али задржава суштина.

Колико се заиста губи на информацији у агрегационом слоју? Уколико се користи агрегациона функција за максимум (слика 9), онда се при укрупњивању губи информација о томе где тачно налази атрибут који је детектован, али не и информација о томе да ли је он детектован, што је најчешће сасвим доволно за проблем класификације.



Слика 8: Пример филтера које је научила конволутивна мрежа



Слика 9: Агрегација која користи функцију максимума

Губитак на прецизности о позицији атрибута такође чини мрежу робуснијом и омогућава већи степен генерализације [1]. Тиме се допушта да позиција атрибута на слици буде флексибилнија. На пример уколико се препознаје слово A, врх слова (који личи на шпиц) ће се очекивати у средини слике и то на горњој половини, али ће бити допуштено да позиција буде слободнија. Атрибут (врх слова A) неће бити изгубљен при агрегацији који користи функцију за максимум.

2.3.2 Визуелизација конволутивне мреже

Велике конволутивне мреже су показале изузетне резултате у класификацији [6], али до скоро није постојало дубоко разумевање зашто дају тако добре резултате или како се могу побољшати. Рад [12] се бави њиховом визуелизацијом и један од резултата јесте нова архитектура конволутивне мреже које је постигла боље резултате од тадашњих (година 2014.) најбољих. Рад [13] приказује како се конволутивне неуронске мреже могу заварати користећи технику реактивације неурона са градијентним успоном. Издавање атрибута и њихово разумевање и интерпретација остају тежак проблем, чак и када су над визуелним подацима. Неколико метода предложени су у радовима [14, 15, 16].

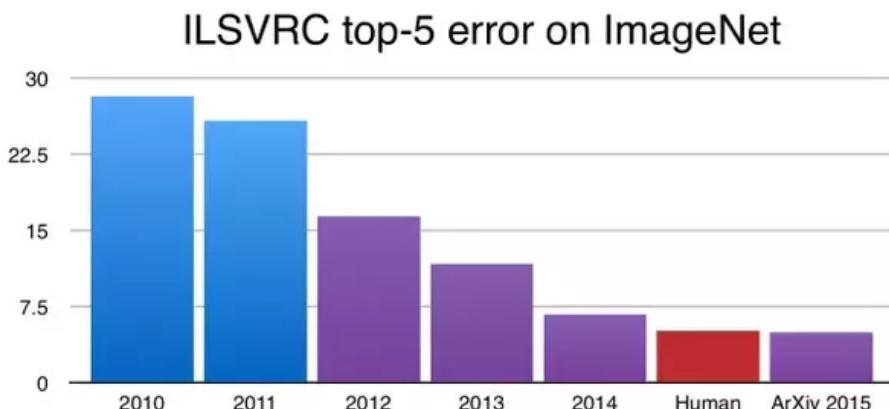
Пројекат DeepDream приказан у делу 3 је делом мотивисан потребом за дубљим разумевањем функционисања конволутивних мрежа.

2.3.3 Такмичење ILSVRC

Најпознатији скуп података у области класификације слика је *ImageNet* који се користи на такмичењу *ILSVRC*. *ImageNet* за циљ има да по-нуди барем 1000 слика за сваки скуп синонима у бази *WordNet* којих има преко 100 000 инстанци.

Google је 2010. године организовао такмичење *The ImageNet Large Scale Visual Recognition Challenge (ILSVRC)* чија је тема била класификација и препознавање објекта на сликама. Такмичење је за циљ имало да мотивише истраживаче да даље развијају алгоритме за класификацију објекта на сликама као и да им омогући да своје резултате могу да упореде са другима у области. Такмичење се оджава сваке године почевши од 2010. и представља један од најбитнијих догађаја у овом делу области машинског учења.

Око 2011. године, грешка класификације на такмичењу је била 25%, 2012. године је дубока неуронска мрежа постигла 16%, а у следећих неколико година је грешка је драстично падала. Слика 10 приказује грешку класификације на неколико узастопних такмичења. Занимљиво је поменути да су 2015. године и Microsoft и Google успели да постигну мању грешку од човека (5.1%), 4.94% и 4.9% респективно.



Слика 10: Грешка класификације на скупу *ImageNet*

3 Deep dream

DeepDream представља пројекат који је конструисао Google. Главна мотивација за пројекат било је интересовање истраживача да разумеју и визуализују како се понашају неурони утrenirane конволутивне мреже.

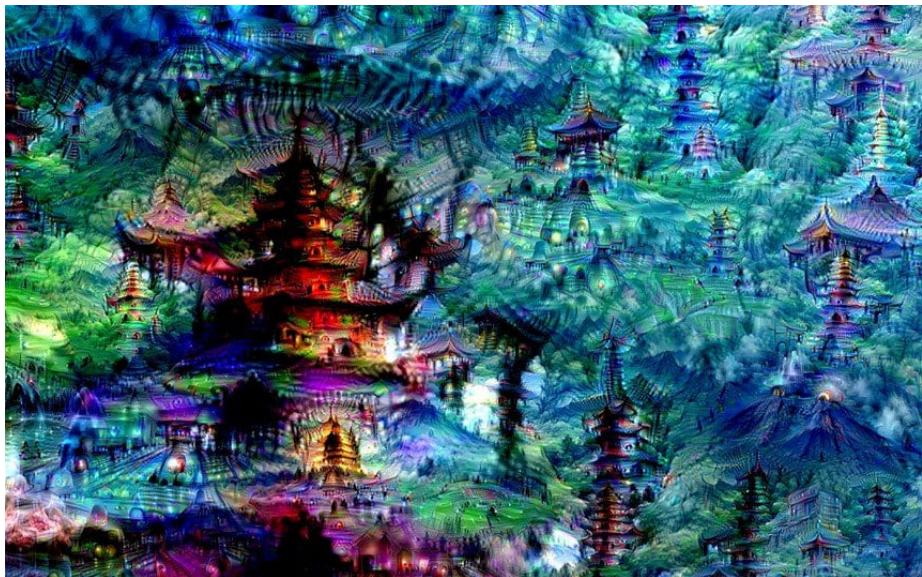
Реч сан (eng. *dream*) се налази у називу јер је истраживаче ово подсећало на визуелизацију снова. Google је на неки реч брендирао реч *сњетак* (eng. *dreaming*) у овом контексту и њена употреба је честа када се мисли на процес генерисања слика које проузрокују тражене активације у тренираној мрежи.

Приступ који се користи је да се утreniranoј конволутивној неуронској мрежи која препознаје слике проследи нека произвољна слика,

а онда се прослеђена слика мења тако да што више активира циљани слој неуронске мреже. Део 3.1 ће прецизније изложити процес генерисања слике.



Слика 11: Пример слике пројекта DeepDream



Слика 12: Пример слике пројекта DeepDream

3.1 Генерисање слика

Процес генерисања слике истраживачки тим је називао *санђање* (eng. *dreaming*), и једна од инспирација су им били снови. Доби-



Слика 13: Визуелизација мреже која препознаје банане

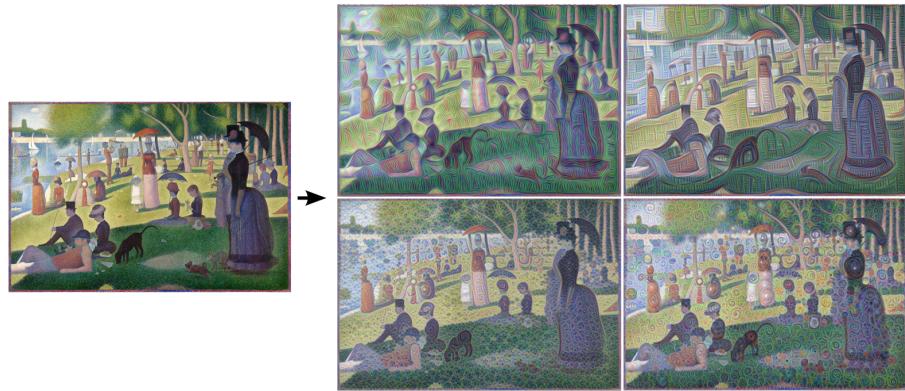


Слика 14: Визуелизација мреже које препознаје тегове

јене слике поседују халуциногени ефекат који многе подсећа на снове. Слике 11 и 12 приказују неке од добијених слика.

Једна од највећих замерки које постоје за неуронске мреже јесте неинтерпретабилност модела, те је од велика важности да постоји барем неки начин да се објасни добијени модел. Визуелизација мреже игра врлобитну улогу јер омогућава да се провери шта је то мрежа научила. Уколико мрежа учи да препознаје да ли је нешто банана или не, било би добро од мреже добити одређене слике о томе шта је за мрежу банана, а шта није (слика 13). Слика 14 приказује шта за неуронску мрежу значи тег. Може се приметити да је услед слика која су садржали људску руку која држи тег, мрежа научила да је рука саставни део тега.

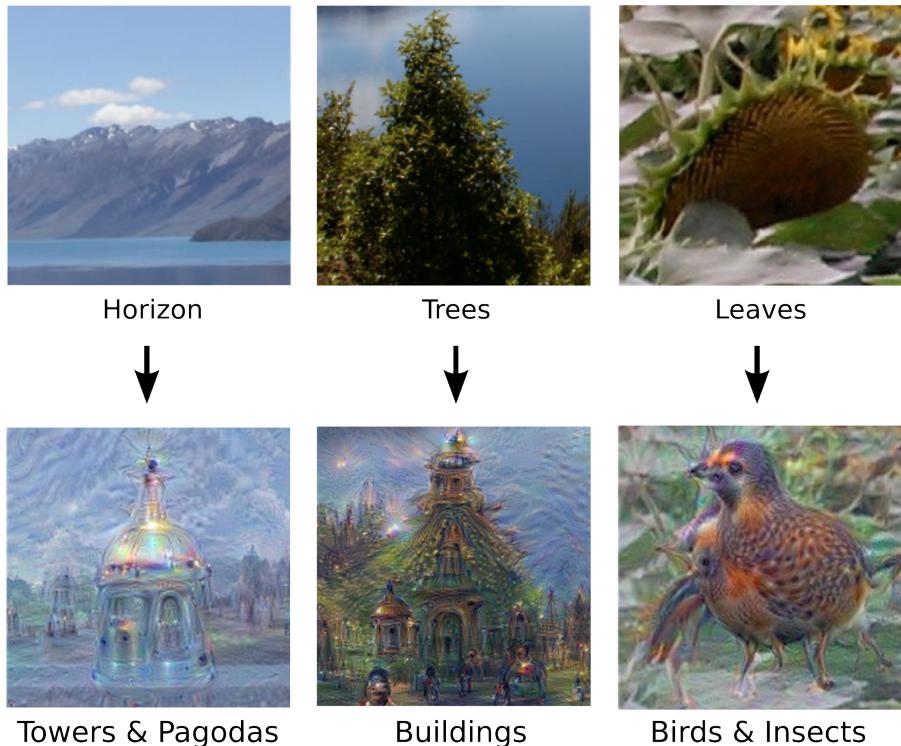
Техника коју Google користи је да се мрежи проследи одређена слика, мрежа се пусти да изврши детекцију, а потом се бира слој и врши се амплификација дела слике који је слој детектовао. Као што је раније поменуто, слојеви конволутивне мреже варирају по комплексности атрибута које описују те ће и резултати визуелизације варирати. Слике 15 и 16 приказују различите ефекте визуелизације у зависности од изабраног слоја.



Слика 15: Пример визуелизације, Google

Уколико се бирају виши слојеви, на сликама ће се појављивати комплексније фигуре попут грађевина и животиња. Главна идеја јесте да неуронска мрежа појача објекте које детектује како би се добиле јасније слике. Кроз итеративни процес неуронска мрежа добија слику из претходне итерације (у првој итерацији се даје произвољна слика) и природно амплификује атрибут који слој детектује. Уколико се

дозволи велики број итерација, на сликама ће се појављивати изузетно изражени објекти које су неурони у изабраном слоју мреже научили да препознају. Од оригиналне слике зависи шта и где ће начелно мрежа пронаћи и касније амплификовати кроз итеративни процес.



Слика 16: Пример визуелизације, Google

Генерирање слика се врши тако што се покреће процес грађијентног успона који покушава да максимизује l_2 норму активација изабраног слоја мреже. Google предлаже неколико трикова како би се добили бољи резултати:

- нормализовати магнитуду корака грађијентног успона;
- додати мало шума;
- применити успон на неколико октава.

3.2 Програмски код за генерирање слика

У овом делу биће приказани делови кода који је коришћен у пројекту DeepDream уз кратка објашњења. Оригинални код доступан је на адреси:

<https://github.com/google/deepdream>.

```
# Bir se utrenirani model
model_path = 'putanja do modela'
net_fn    = model_path + 'deploy.prototxt'
```

```

param_fn = model_path + 'bvlc_googlenet.caffemodel'

# Model se modifikuje da izracunava gradijente unazad
model = caffe.io.caffe_pb2.NetParameter()
text_format.Merge(open(net_fn).read(), model)
model.force_backward = True
open('tmp.prototxt', 'w').write(str(model))

# Konstrise se mreza od utreniranog modela
net = caffe.Classifier('tmp.prototxt', param_fn,
                        mean = np.float32([104.0, 116.0, 122.0]),
                        channel_swap = (2,1,0))

```

```

def objective_L2(dst):
    dst.diff[:] = dst.data

def make_step(net, step_size=1.5, end='inception_4c/output',
             jitter=32, clip=True, objective=objective_L2):
    '''Funkcija koja vrši korak gradijentnog uspona.'''
    # Izvuci se referencia na ulaznu sliku za mrezu
    src = net.blobs['data']
    dst = net.blobs[end]

    # Generise se i primenjuje sum
    ox, oy = np.random.randint(-jitter, jitter+1, 2)
    src.data[0] = np.roll(np.roll(src.data[0], ox, -1), oy, -2)

    net.forward(end=end)
    # Postavlja se funkcija cilja
    objective(dst)
    net.backward(start=end)
    g = src.diff[0]
    # Primjenjuje se normalizovani korak uspona na sliku
    src.data[:] += step_size/np.abs(g).mean() * g
    src.data[0] = np.roll(np.roll(src.data[0], -ox, -1), -oy, -2)

    if clip:
        bias = net.transformer.mean['data']
        src.data[:] = np.clip(src.data, -bias, 255-bias)

```

Функција врши градијентни успон кроз различите скале, такозване *oktave* (eng. *octave*).

```

def deepdream(net, base_img, iter_n=10, octave_n=4,
              octave_scale=1.4, end='inception_4c/output',
              clip=True, **step_params):
    # Priprema se slika za sve oktave
    octaves = [preprocess(net, base_img)]
    for i in xrange(octave_n-1):
        octaves.append(nd.zoom(octaves[-1],
                               (1, 1.0/octave_scale,1.0/octave_scale),

```

```

        order=1))

src = net.blobs['data']
# Alocira se nova slika
detail = np.zeros_like(octaves[-1])
for octave, octave_base in enumerate(octaves[::-1]):
    h, w = octave_base.shape[-2:]
    if octave > 0:
        # Zumiraju se detalji iz prethodne oktave
        h1, w1 = detail.shape[-2:]
        detail = nd.zoom(detail, (1, 1.0*h/h1, 1.0*w/w1),
                          order=1)

    src.reshape(1,3,h,w)
    src.data[0] = octave_base+detail
    for i in xrange(iter_n):
        make_step(net, end=end, clip=clip, **step_params)

        # Vizuelizacija
        vis = deprocess(net, src.data[0])
        if not clip:
            vis = vis*(255.0/np.percentile(vis, 99.98))
        showarray(vis)
        print octave, i, end, vis.shape
        clear_output(wait=True)

    # Vrsi se ekstrakcija detalja koje je pronasla
    # tekuca oktava
    detail = src.data[0]-octave_base
    # Vraca se dobijena slika
return deprocess(net, src.data[0])

```

Бирамо слику која ће бити база за *сањање*. Слика је приказана на слици 17.

```

img = np.float32(PIL.Image.open('matf2.jpg'))
showarray(img)

```

Генерише се нова слика и појављују се нови облици. Добијени резултат је приказан на слици 18.

```

img = np.float32(PIL.Image.open('matf2.jpg'))
showarray(img)

```

Уколико изаберемо нешто нижи слој, на слици ће се појављивати једноставити облици, резултат се може видети на слици 19.

```

img = np.float32(PIL.Image.open('matf2.jpg'))
showarray(img)

```



Слика 17: Слика која је коришћена за основу сањања

Наредба приказује све слојеве који постоје у мрежи.

```
net.blobs.keys()
# ['data',
#  'conv1/7x7_s2',
#  'pool1/3x3_s2',
#  'pool1/norm1',
#  'inception_5b/1x1',
#  ...
#  'inception_5b/3x3_reduce',
#  'inception_5b/3x3',
#  'inception_5b/5x5_reduce',
#  'inception_5b/5x5',
#  'inception_5b/pool',
#  'inception_5b/pool_proj',
#  'inception_5b/output',
#  'pool5/7x7_s1',
#  'loss3/classifier',
#  'prob']
```

```
# Kreira se direktorijum 'frames' ukoliko ne postoji
!mkdir frames
frame = img
frame_i = 0
```



Слика 18: Слика која је добијена након првог грађиентног успона

```
# Generise se 100 slika kroz iterativni proces
h, w = frame.shape[:2]
# koeficijent skaliranja
s = 0.05
for i in xrange(100):
    frame = deepdream(net, frame)
    PIL.Image.fromarray(np.uint8(frame))
    .save("frames/%04d.jpg"%frame_i)
    frame = nd.affine_transform(frame, [1-s,1-s,1],
                                [h*s/2,w*s/2,0],
                                order=1)
    frame_i += 1
```

Слика 20 приказује неколико слика од генерисаних 100.

Може се користити и слика која наводи генератор. Слика 21 приказује слику која се користи за навођење, а слика 22 слику која је добијена.

```
# Ucitava se slika koji navodi generisanje
guide = np.float32(PIL.Image.open('flowers.jpg'))
showarray(guide)

end = 'inception_3b/output'
h, w = guide.shape[:2]
```



Слика 19: Слика која је добијена након коришћења нижег слоја

```
src, dst = net.blobs['data'], net.blobs[end]
src.reshape(1,3,h,w)
src.data[0] = preprocess(net, guide)
net.forward(end=end)
guide_features = dst.data[0].copy()

def objective_guide(dst):
    x = dst.data[0].copy()
    y = guide_features
    ch = x.shape[0]
    x = x.reshape(ch,-1)
    y = y.reshape(ch,-1)
    # Izracunava se matrica skalarnog
    # proizvoda sa slikom koja navodi generisanje
    A = x.T.dot(y)
    # Bira se vrednost koja je najveca
    dst.diff[0].reshape(ch,-1)[:] = y[:,A.argmax(1)]

# Generise se slika koristeci sliku koja navodi generisanje
_=deepdream(net, img, end=end, objective=objective_guide)
```



Слика 20: Генерисане слике 0, 3, 8, 12, 17, 38, 99 од 100 - крећући се кроз врсте почевши одозго

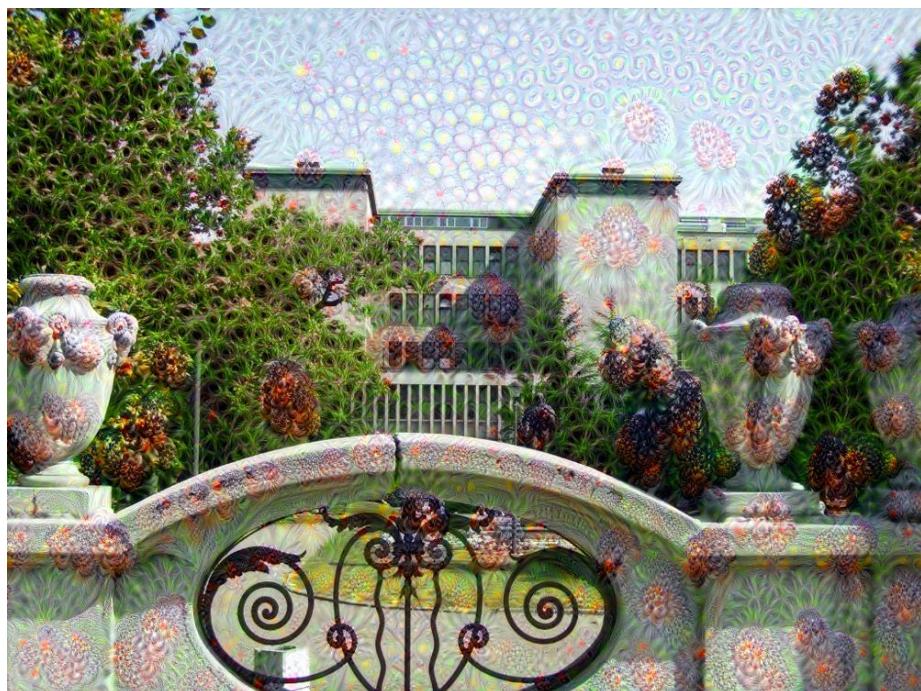
4 Закључак

Пројекат DeepDream који се ослања на раније радове [12, 13, 16, 15] указује да је постигнуто одређено разумевање дубинског рада конвултивних неуронских мрежа. Доприносио је даљој популаризацији неуронских мрежа и инспирисао истраживаче да дубље истраже понашање конвултивних мрежа.

Разумевање рада неуронских мрежа и интерпретабилност њихових модела и даље остају отворена значајна тема у области машинског учења, но свакако су постигнути значајни успеси у визуелизацији рада мрежа у последњих неколико година.



Слика 21: Слика која наводи генератор



Слика 22: Добијена слике користећи слику навођења

Литература

- [1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [2] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.
- [3] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

- [4] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.
- [5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, Dec 1989.
- [6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [7] D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, June 2012.
- [8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [9] X. Zhang, J. Zhao, and Y. LeCun. Character-level Convolutional Networks for Text Classification. *ArXiv e-prints*, September 2015.
- [10] N. Mohammadian Rad, A. Bizzego, S. M. Kia, G. Jurman, P. Venuti, and C. Furlanello. Convolutional Neural Network for Stereotypical Motor Movement Detection in Autism. *ArXiv e-prints*, November 2015.
- [11] Анђелка Зечевић Младен Николић. *Научно израчунавање*. 2017.
- [12] M. D Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. *ArXiv e-prints*, November 2013.
- [13] A. Nguyen, J. Yosinski, and J. Clune. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. *ArXiv e-prints*, December 2014.
- [14] A. Mahendran and A. Vedaldi. Understanding Deep Image Representations by Inverting Them. *ArXiv e-prints*, November 2014.
- [15] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *ArXiv e-prints*, December 2013.
- [16] A. Dosovitskiy and T. Brox. Inverting Visual Representations with Convolutional Networks. *ArXiv e-prints*, June 2015.