Deep learning with Keras

Joachim Mathiesen

Niels Bohr Institute

Feature Engineering – Alleviating the Curse of Dimensionality

- Feature Extraction – the art of keeping only the relevant information and discard everything else. In other words, the task is to perform a sensible dimensionality reduction of data through a linear or non-linear transformation of data.

- Often the feature extraction is a significant part of the overall process

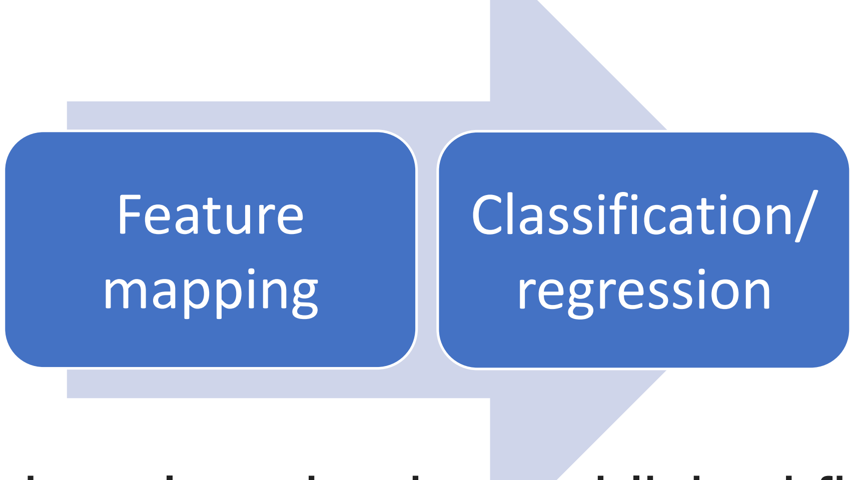
Several general methods are frequently used, for example

- Eyeballing, ad-hoc reduction, “qualified” guessing
- t-Distributed Stochastic Neighbor Embedding
- Factor Analysis
- Principal Component Analysis
- Non-linear reduction methods

Deep Learning

Deep learning loosely refers to a broad range of machine learning methods, though we typically mean deep neural networks

Deep learning can be the jack of all trades - handling in one go both the feature engineering and the classification/regression.

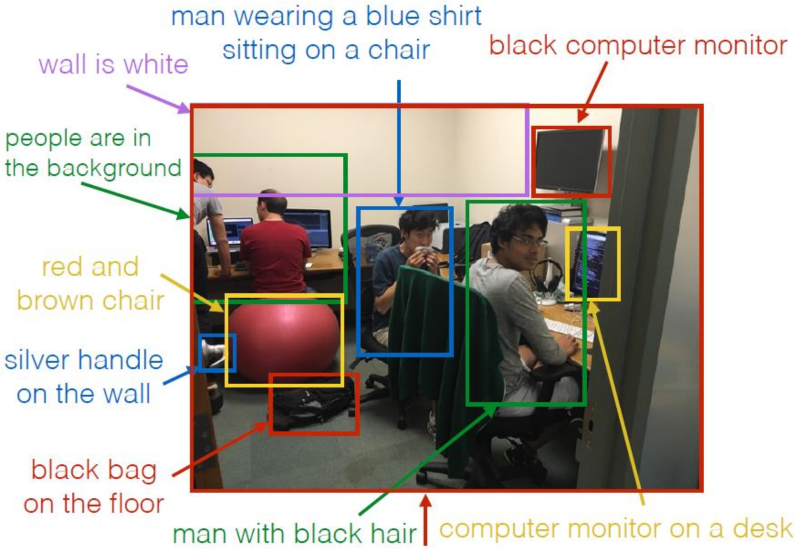


The success of deep learning is established first and foremost in the fields of computer vision, speech recognition, language processing – but new fields are opened continuously.

Deep Learning

State of the art examples include face recognition, robotic vision and dense image captioning – useful e.g. in self-driving cars

a group of people in an office



https://cs.stanford.edu/people/karpathy/

ConvNets

Convolutional networks were among the first examples of machine learning methods to perform well. Convnets have been adopted since the early 90s. AT&T used it in automatized cheque reading, Microsoft used it early on for handwriting recognition, etc.

The commercial breakthrough recently is mostly due to convnets outperforming other methods in object classification.

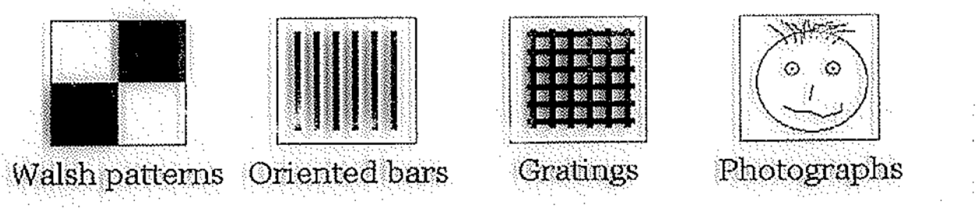
Everybody can take part in the success: a number of high performing software libraries including implementations of convnets are freely availably and open source.

ConvNets

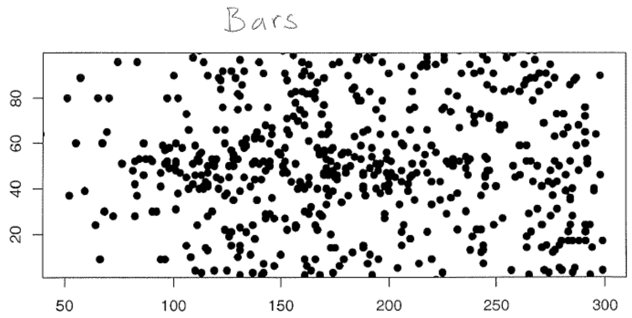
Convolutional networks (convnets) are one of the corner stones in most deep learning methods. Convnets are highly inspired by the way visual information is processed in the primary visual cortex of mammals.

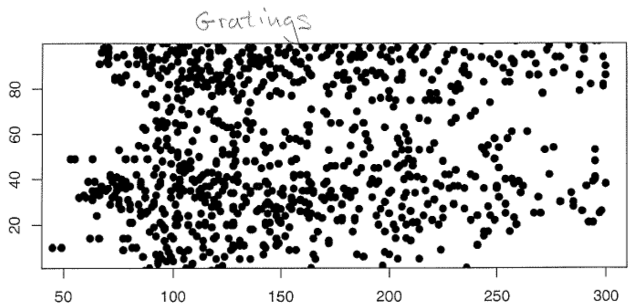
Research example on reading neural spike trains

- Encoding of visual stimuli by complex cells in the primary visual cortex of awake monkeys.



Heller J, Hertz JA, Kjær TW, Richmond BJ. Information flow and temporal coding in primate pattern vision. Journal of computational neuroscience. 1995 Sep 1;2(3):175-93.





TensorFlow

- TensorFlow™ is a software library of core methods used in machine learning and is an excellent starting point for advanced neural network models.

- The library is open source and highly flexible. It can run across a broad range of platforms and on CPUs as well as GPUs.

- The library was originally developed by people at the Google brain team but was later opened to the public.

Tensors

Tensor dimension	Description	Example data format (sample number, [data])
0	A tensor containing only a single number – a so-called scalar. For example temperature T.	(n,[T])
1	vectors containing a list of variables, for example velocity, which is a 3D vector, is a 1D tensor	(n, [(v_x,v_y,v_z)])
2	a matrix containing both rows and columns	(n, [(a,b,);(c,d)])
3	matrices packed in an array, for example an image which contains (height, width, color depth)	(n, [(width);(height);(depth)])
4	time series of images (videos)	(n, [(width);(height);(depth);(time)])

Keras

Here we will use the Keras application programming interface (API), which is a high-level interface for implementing neural network models. Keras has a growing user base.

Keras is highly user friendly and allows for quick experimentation and implementation of convolutional and recurrent networks. The syntax in both R and Python is intuitive and offers great insight into neural networks.

Supports both CPU and GPU (the latter based on CUDA, meaning NVIDIA GPU) and can run on top of Cognitive Toolkit, TensorFlow, and Theano. We shall here use Tensorflow.

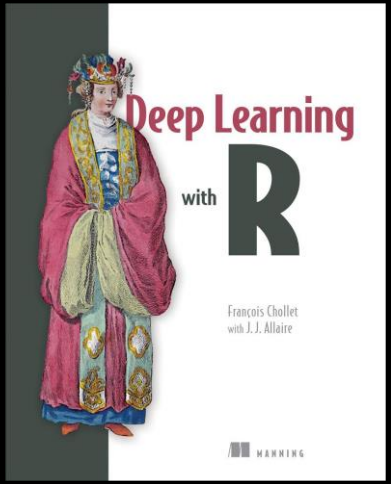
https://keras.io/

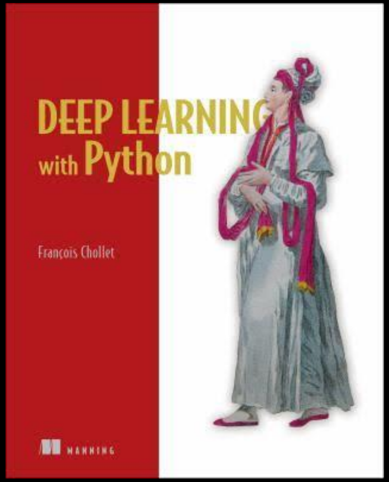
Keras

Many good tutorials and examples can be found at keras.io.

Keras comes with a number of datasets you can use directly for learning the interface, for quick model testing and for building new models.

A basic and easy to read introduction by the creator of Keras is the book:





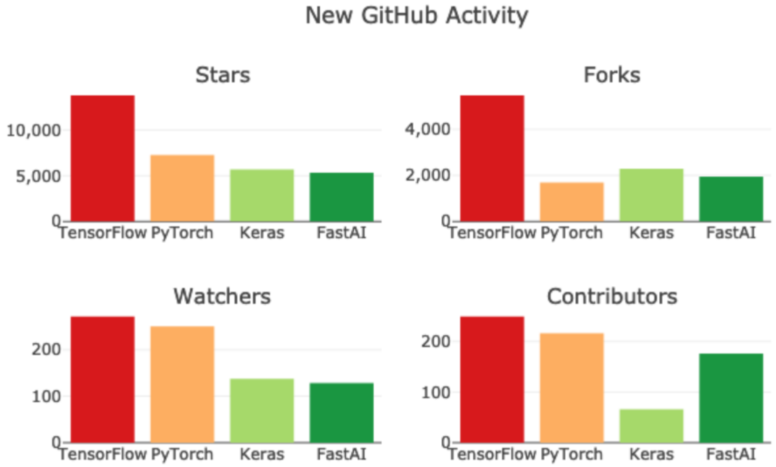
https://keras.io/

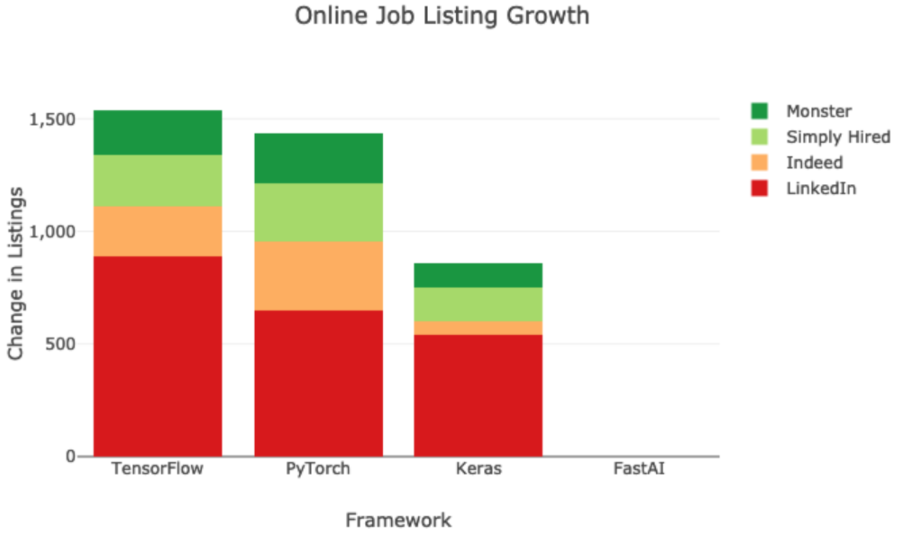
Alternatives to TensorFlow

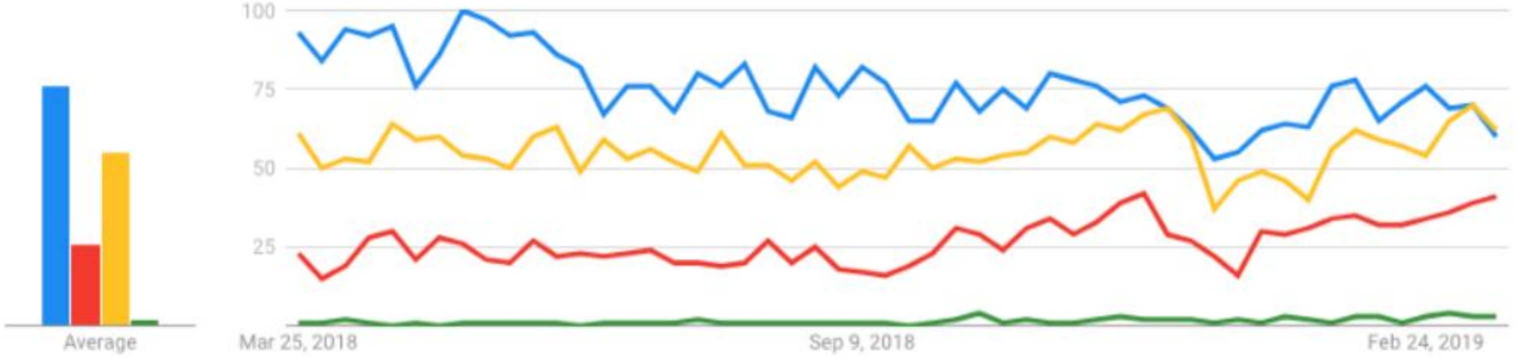
PyTorch (and Torch) is a tensor library for deep learning on GPUs and CPUs running in Python. Supposedly the learning curve on PyTorch is less steep than that of Tensorflow. Tensorflow has the largest community behind it.

https://github.com/microsoft/cntk

Which framework to choose?







https://towardsdatascience.com/which-deep-learning-framework-is-growing-fastest-3f77f14aa318

TensorFlow in blue; Keras in yellow, PyTorch in red, fastai in green

Convolutional Network: the ultra short version

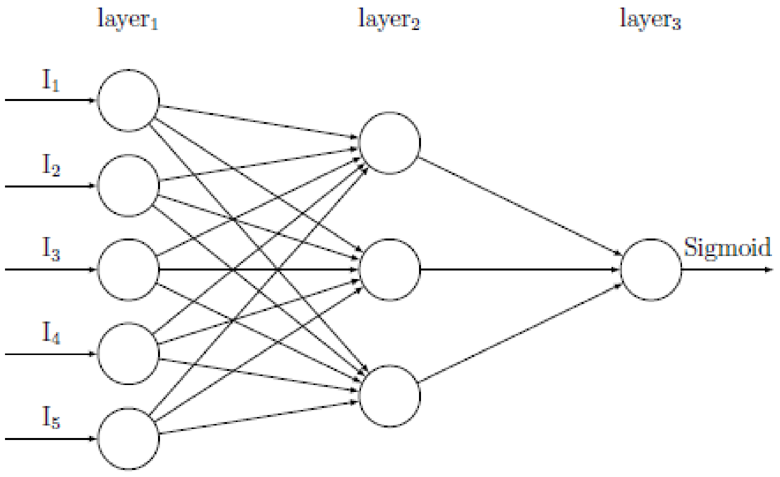
Sequential neural network:

1) Data formatted and fed to an input layer. Data comes with corresponding targets.

2) Layers are organized sequentially, such that information passes from layer(i) to layer(i+1). The connection between layers are defined by a set of weights.

3) Loss function to measure performance

4) Training procedure (optimizer)



Convolutional network - filters

An image is a three dimensional tensor [(width); (height); (color depth)].

A filter is applied to an image “I” by folding in each channel the image with a filter kernel “K”.

The result is a new image “S”, where

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n)$$

Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur 3 × 3 (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

https://en.wikipedia.org/wiki/Kernel_(image_processing)

ConvNets applied to the MNIST data

THE MNIST DATABASE

of handwritten digits

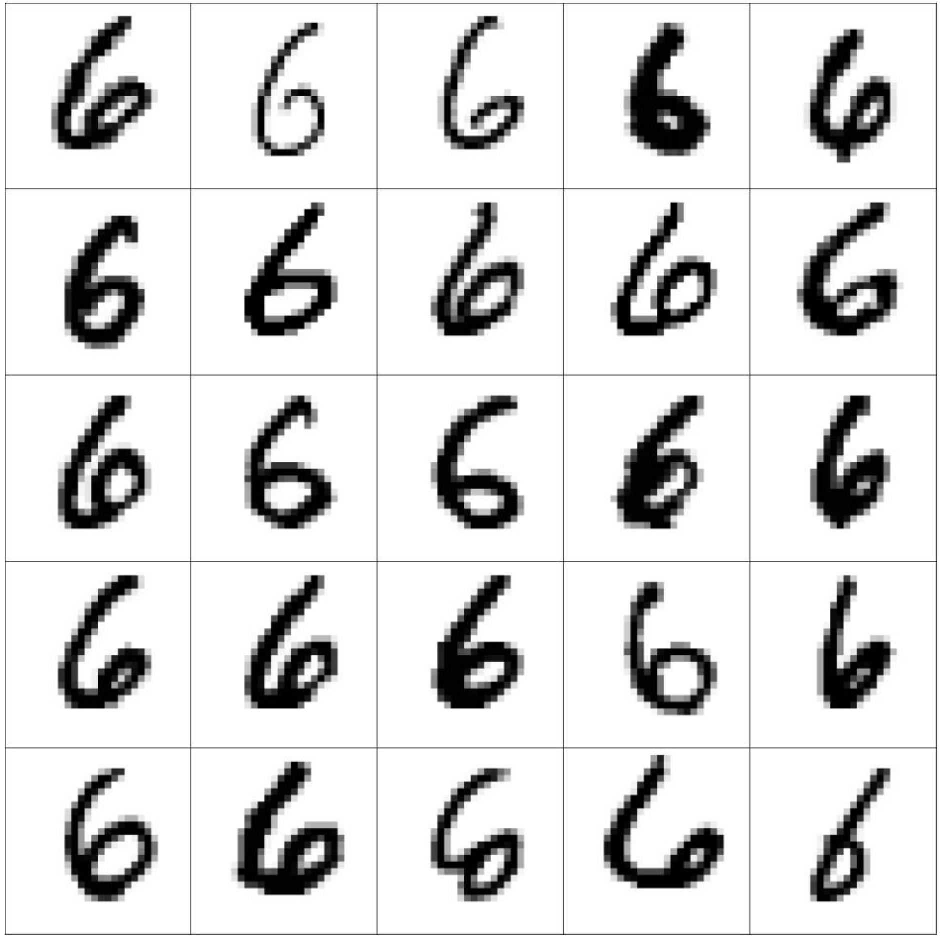
Yann LeCun, Courant Institute, NYU
Corinna Cortes, Google Labs, New York
Christopher J.C. Burges, Microsoft Research, Redmond

The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image.

It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting.

Four files are available on this site:

```
train-images-idx3-ubyte.gz:  training set images (9912422 bytes)
train-labels-idx1-ubyte.gz:  training set labels (28881 bytes)
t10k-images-idx3-ubyte.gz:   test set images (1648877 bytes)
t10k-labels-idx1-ubyte.gz:   test set labels (4542 bytes)
```

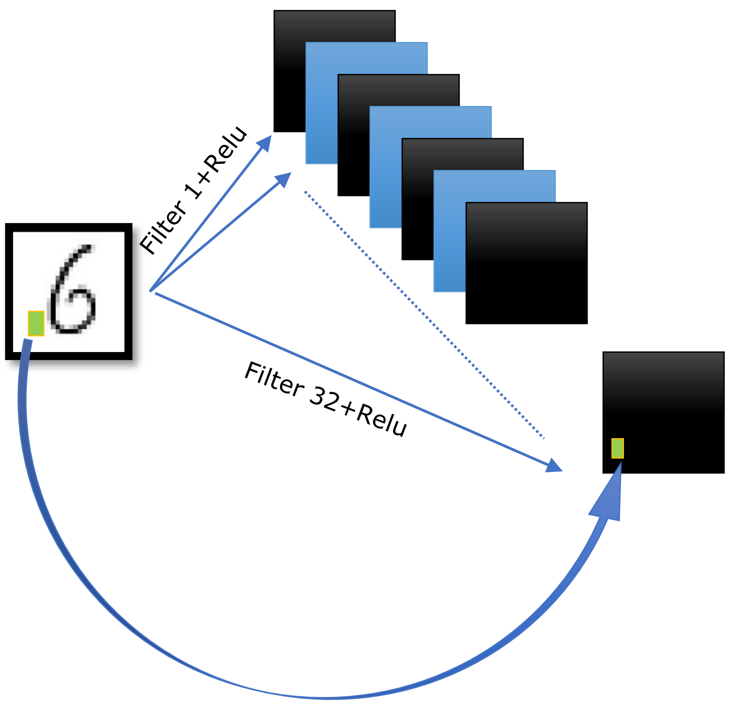


All digits come as 28x28 grey scale values in the range 0-255.

http://yann.lecun.com/exdb/mnist/

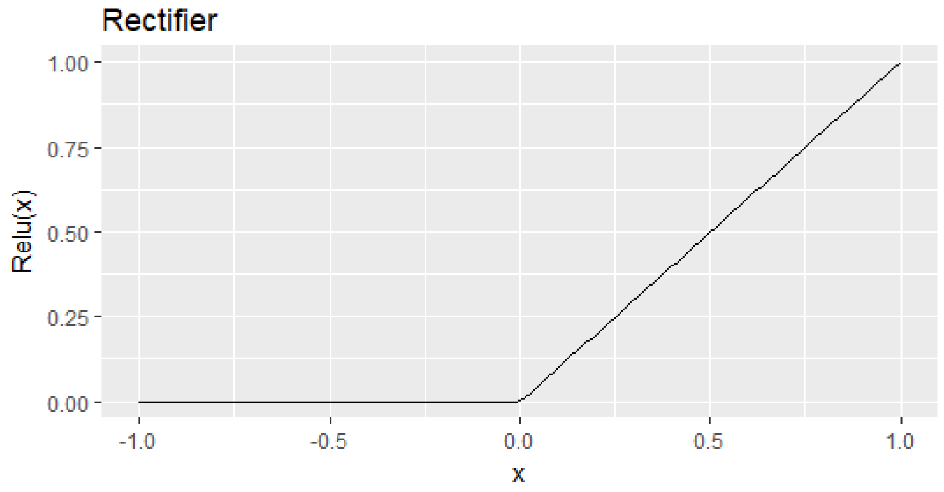
Convolutional network

Output size after applying filters (26x26)



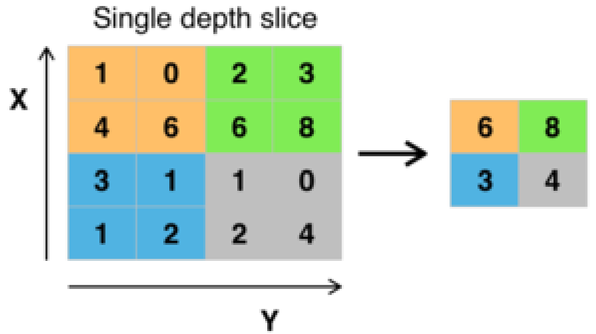
Outputs from a layer are rectified with Relu(x)=max(0,x)

NOTE!!! This is how we introduce non-linearity in the model.

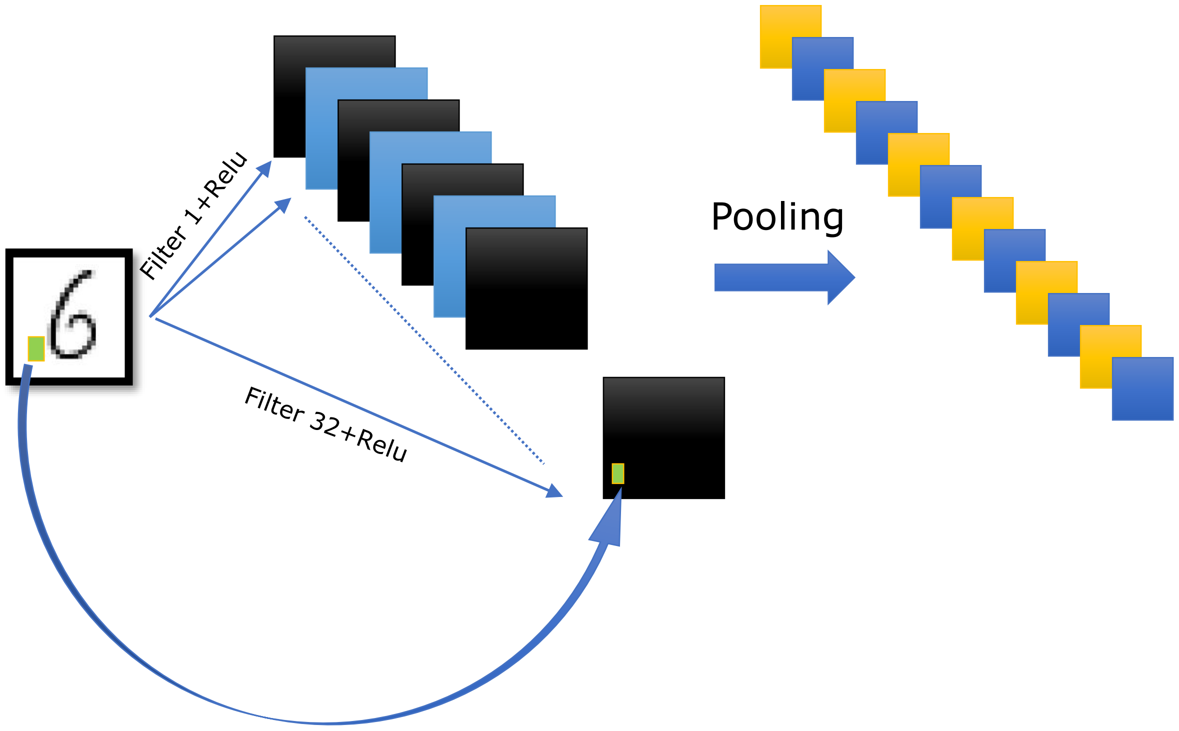


Convolutional network - pooling

In smart ways, we seek to lower the dimensionality, and still let most information pass from one layer to the next. A common approach is to pool pixels. Pixels can be pooled in many ways, for example taking averages over groups of pixels or simply just keep the pixel with the maximum value. Here a basic example of 2x2 pooling with a stride value of 2.



https://en.wikipedia.org/wiki/Convolutional_neural_network#Pooling



Convolutional network – basic example using Keras

Layer_conv_2d()
32 of 3x3 filters
Rectified linear activation

→

Layer_max_pooling_2d()
Pools over 2x2 blocks, default stride value equals pool size

→

Layer_conv_2d()
64 of 3x3 filters
Rectified linear activation

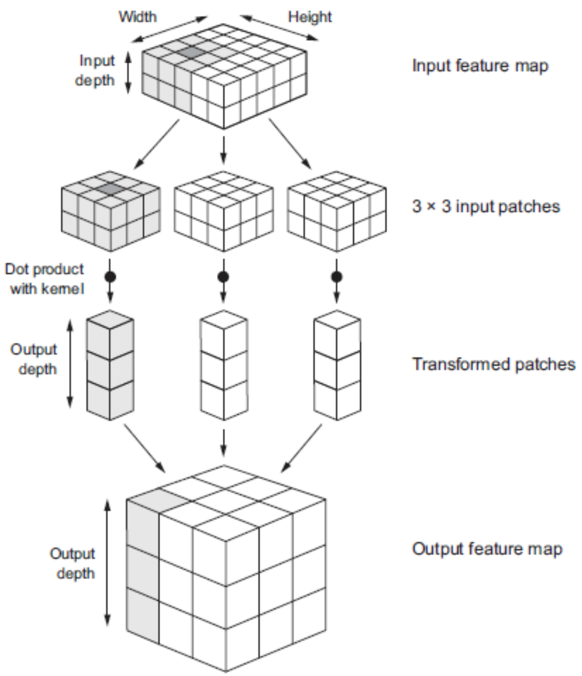
→

layer_max_pooling_2d()
Pools over 2x2 blocks, default stride value equals pool size

→

layer_conv_2d()
64 of 3x3 filters
Rectified linear activation

Deep Learning with Python, F. Chollet




```
model <- keras_model_sequential() %>%
  layer_conv_2d(filters = 32, kernel_size = c(3, 3), activation = "relu",
  input_shape = c(28, 28, 1)) %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_conv_2d(filters = 64, kernel_size = c(3, 3), activation = "relu") %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_conv_2d(filters = 64, kernel_size = c(3, 3), activation = "relu")
summary(model)

## ________________________________________________________________________________
## Layer (type)                     Output Shape                  Param #
## ================================================================================
## conv2d_1 (Conv2D)                (None, 26, 26, 32)            320
## ________________________________________________________________________________
## max_pooling2d_1 (MaxPooling2D)   (None, 13, 13, 32)            0
## ________________________________________________________________________________
## conv2d_2 (Conv2D)                (None, 11, 11, 64)            18496
## ________________________________________________________________________________
## max_pooling2d_2 (MaxPooling2D)   (None, 5, 5, 64)              0
## ________________________________________________________________________________
## conv2d_3 (Conv2D)                (None, 3, 3, 64)              36928
## ================================================================================
## Total params: 55,744
## Trainable params: 55,744
## Non-trainable params: 0
## ________________________________________________________________________________
```

Convolutional network – basic example in R using Keras

Layer_conv_2d()
32 of 3x3 filters
Rectified linear activation

→

Layer_max_pooling_2d()
Pools over 2x2 blocks, default stride value equals pool size

→

Layer_conv_2d()
64 of 3x3 filters
Rectified linear activation

→

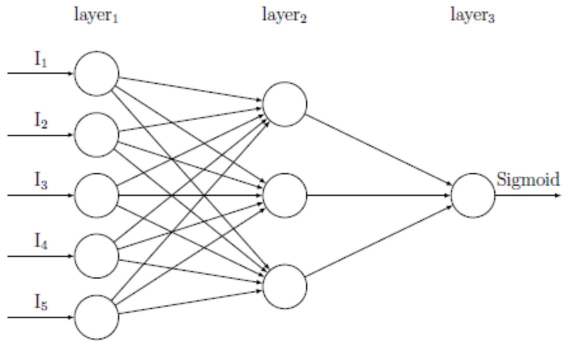
layer_max_pooling_2d()
Pools over 2x2 blocks, default stride value equals pool size

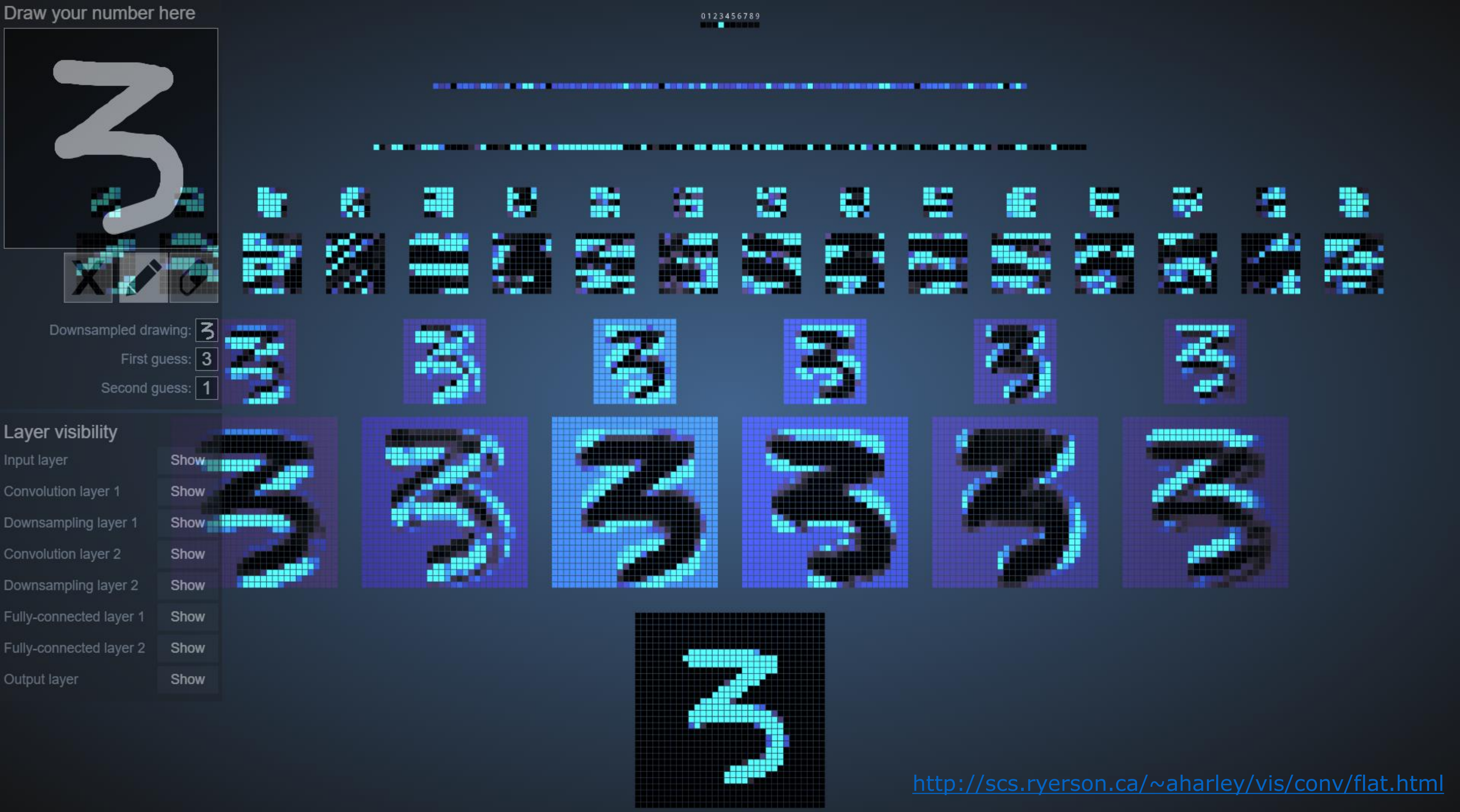
→

layer_conv_2d()
64 of 3x3 filters
Rectified linear activation

This was the part, we tried to do with PCA

We now want to do the 10-way classification and determine which digit was presented to our model. We do this by adding a classification step after. For that purpose we use a dense neural network, why not?





http://scs.ryerson.ca/~aharley/vis/conv/flat.html

Very Deep Convolutional Networks for Large-Scale Image Recognition

The amount of data and the computational resources needed to train deep convolutional networks quickly becomes excessive. At least you would need GPUs for efficient training. Do not bother with CPUs.

Instead of training your models, you can download pre-trained models and save your time and money.

Several winning and pre-trained models from the IMAGENET Large Scale Visual Recognition Challenge (ILSVRC) can be downloaded. For example the 2014 competition winner, were 150,000 labelled photographs used in the training.

The competition photographs were collected from flickr and various search engines. The competition was to classify an image among 1000 predefined categories. The winning model was VGG19.

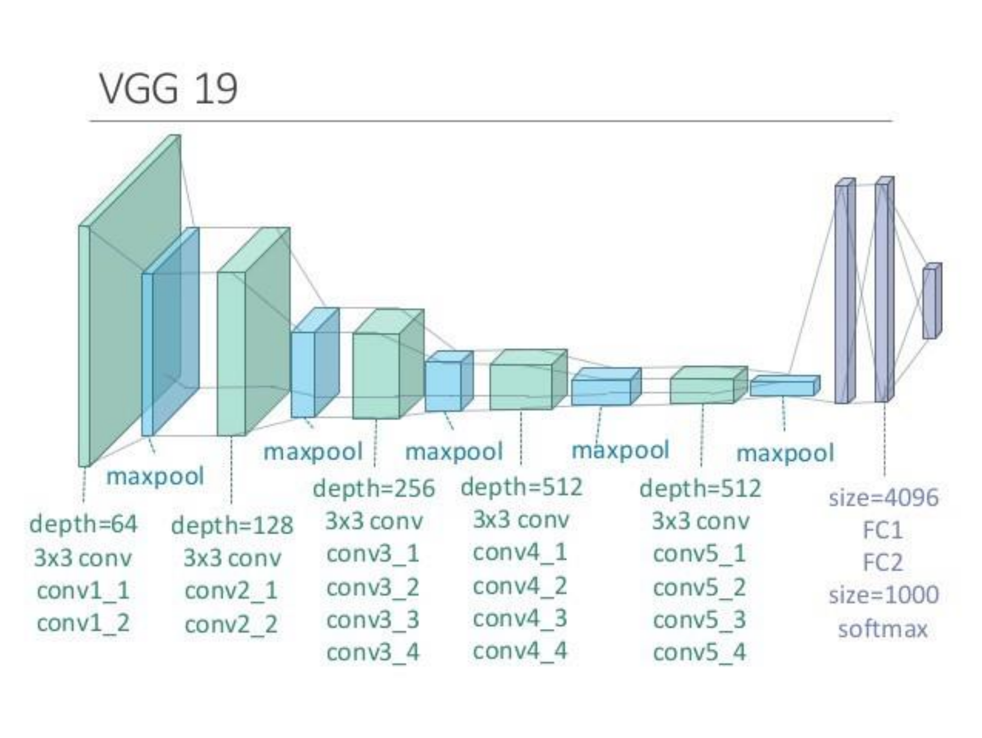
http://www.image-net.org/challenges/LSVRC/

kit fox, Vulpes macrotis, English setter, Australian terrier, grey whale, gray whale, devilfish, Eschrichtius gibbosus, Eschrichtius robustus, lesser panda, red panda, panda, bear cat, cat bear, Ailurus fulgens, Egyptian cat, ibex, Capra ibex, Persian cat, cougar, puma, catamount, mountain lion, painter, panther, Felis concolor, gazelle, porcupine, hedgehog, sea lion, badger, Great Dane, Scottish deerhound, deerhound, killer whale, killer, orca, grampus, sea wolf, Orcinus orca, mink, African elephant, Loxodonta africana, red wolf, maned wolf, Canis rufus, Canis niger, jaguar, panther, Panthera onca, Felis onca, hyena, hyaena, titi, titi monkey, three-toed sloth, ai, Bradypus tridactylus, sorrel, black-footed ferret, ferret, Mustela nigripes, dalmatian, coach dog, carriage dog, Staffordshire bullterrier, Staffordshire bull terrier, Bouvier des Flandres, Bouviers des Flandres, weasel, miniature poodle, bighorn, bighorn sheep, cimarron, Rocky Mountain bighorn, Rocky Mountain sheep, Ovis canadensis, fox squirrel, eastern fox squirrel, Sciurus niger, colobus, colobus monkey, tiger cat, impala, Aepyceros melampus, coyote, prairie wolf, brush wolf, Canis latrans, Yorkshire terrier, Newfoundland, Newfoundland dog, red fox, Vulpes vulpes, hartebeest, grey fox, gray fox, Urocyon cinereoargenteus, Pekinese, Pekingese, Peke, guenon, guenon monkey, mongoose, indri, indris, Indri indri, Indri brevicaudatus, tiger, Panthera tigris, wild boar, boar, Sus scrofa, zebra, ram, tup, orangutan, orang, orangutang, Pongo pygmaeus, basenji, leopard, Panthera pardus, vizsla, Hungarian pointer, squirrel monkey, Saimiri sciureus, Siamese cat, Siamese, chimpanzee, chimp, Pan troglodytes, komondor, proboscis monkey, Nasalis larvatus, guinea pig, Cavia cobaya, white wolf, Arctic wolf, Canis lupus tundrarum, ice bear, polar bear, Ursus Maritimus, Thalarctos maritimus, gorilla, Gorilla gorilla, ox, Tibetan mastiff, spider monkey, Ateles geoffroyi, Doberman, Doberman pinscher, warthog, Arabian camel, dromedary, Camelus dromedarius, siamang, Hylobates syndactylus, Symphalangus syndactylus, golden retriever, Border collie, hare, boxer, patas, hussar monkey, Erythrocebus patas, baboon, macaque, capuchin, ringtail, Cebus capucinus, flat-coated retriever, hog, pig, grunter, squealer, Sus scrofa, Eskimo dog, husky, Brittany spaniel, dial telephone, dial phone, maze, labyrinth, Gordon setter, dingo, warrigal, warragal, Canis dingo, hamster, Arctic fox, white fox, Alopex lagopus, water buffalo, water ox, Asiatic buffalo, Bubalus bubalis, American black bear, black bear, Ursus americanus, Euarctos americanus, Angora, Angora rabbit, bison, howler monkey, howler, hippopotamus, hippo, river horse, Hippopotamus amphibius, giant panda, panda, panda bear, coon bear, Ailuropoda melanoleuca, tabby, tabby cat, marmoset, Saint Bernard, St Bernard, armadillo, redbone, polecat, fitch, foulmart, foumart, Mustela putorius, marmot, gibbon, Hylobates lar, llama, wood rabbit, cottontail, cottontail rabbit, lion, king of beasts, Panthera leo, Irish setter, red setter, dugong, Dugong dugon, Indian elephant, Elephas maximus, beaver, Madagascar cat, ring-tailed lemur, Lemur catta, Rhodesian ridgeback, lynx, catamount, African hunting dog, hyena dog, Cape hunting dog, Lycaon pictus, langur, timber wolf, grey wolf, gray wolf, Canis lupus, cheetah, chetah, Acinonyx jubatus, sloth bear, Melursus ursinus, Ursus ursinus, German shepherd, German shepherd dog, German police dog, alsatian, otter, koala, koala bear, kangaroo bear, native bear, Phascolarctos cinereus, tusker, echidna, spiny anteater, anteater, wallaby, brush kangaroo, platypus, duckbill, duckbilled platypus, duck-billed platypus, Ornithorhynchus anatinus, wombat, revolver, six-gun, six-shooter, umbrella, schooner, soccer ball, accordion, piano accordion, squeeze box, ant, emmet, pismire, starfish, sea star, chambered nautilus, pearly nautilus, nautilus, grand piano, grand, laptop, laptop computer, strawberry, airliner, warplane, military plane, airship, dirigible, balloon, space shuttle, fireboat, gondola, speedboat, lifeboat, canoe, yawl, catamaran, trimaran, container ship, containership, container vessel, liner, ocean liner, pirate, pirate ship, aircraft carrier, carrier, flattop, attack aircraft carrier, submarine, pigboat, sub, U-boat, wreck, half track, tank, army tank, armored combat vehicle, armoured combat vehicle, missile, bobsled, bobsleigh, bob, dogsled, dog sled, dog sleigh, bicycle-built-for-two, tandem bicycle, tandem, mountain bike, all-terrain bike, off-roader, freight car, passenger car, coach, carriage, barrow, garden cart, lawn cart, wheelbarrow, shopping cart, motor scooter, scooter, forklift, electric locomotive, steam locomotive, amphibian, amphibious vehicle, ambulance, beach wagon, station wagon, wagon, estate car, beach waggon, station waggon, waggon, cab, hack, taxi, taxicab, convertible, jeep, landrover, limousine, limo, minivan, Model T, racer, race car, racing car, sports car, sport car, go-kart, golfcart, golf cart, moped, snowplow, snowplough, fire engine, fire truck, garbage truck, dustcart, pickup, pickup truck, tow truck, tow car, wrecker, trailer truck, tractor trailer, trucking rig, rig, articulated lorry, semi, moving van, police van, police wagon, paddy wagon, patrol wagon, wagon, black Maria, recreational vehicle, RV, R.V., streetcar, tram, tramcar, trolley, trolley car, snowmobile, tractor, mobile home, manufactured home, tricycle, trike, velocipede, unicycle, monocycle, horse cart, horse-cart, mosquito net, oxcart, bassinet, cradle, crib, cot, four-poster, bookcase, china cabinet, china closet, medicine chest, medicine cabinet, chiffonier, commode, table lamp, file, file cabinet, filing cabinet, pay-phone, pay-station, park bench, barber chair, throne, folding chair, rocking chair, rocker, studio couch, day bed, toilet seat, desk, pool table, billiard table, snooker table, dining table, board, entertainment center, wardrobe, closet, press, Granny Smith, orange, lemon, fig, pineapple, ananas, banana, jackfruit, jak, jack, custard apple, pomegranate, acorn, hip, rose hip, rosehip, ear, spike, capitulum, rapeseed, corn, buckeye, horse chestnut, conker, organ, pipe organ, upright, upright piano, chime, bell, gong, drum, membranophone, tympan, gong, tam-tam, maraca, marimba, xylophone, steel drum, banjo, cello, violoncello, lampshade, lamp shade, harp, acoustic guitar, electric guitar, cornet, horn, trumpet, trump, French horn, horn, trombone, harmonica, mouth organ, harp, mouth harp, ocarina, sweet potato, panpipe, pandean pipe, syrinx, bassoon, sax, saxophone, flute, transverse flute, daisy, yellow lady's slipper, yellow lady-slipper, Cypripedium calceolus, Cypripedium parviflorum, cliff, drop, drop-off, valley, vale, alp, volcano, promontory, headland, head, foreland, sandbar, sand bar, coral reef, lakeside, lakeshore, seashore, coast, seacoast, sea-coast, geyser, hatchet, cleaver, meat cleaver, chopper, letter opener, paper knife, paperknife, plane, carpenter's plane, woodworking plane, power drill, lawn mower, mower, hammer, corkscrew, bottle screw, can opener, tin opener, plunger, plumber's helper, screwdriver, shovel, plow, plough, chain saw, chainsaw, cock, hen, ostrich, Struthio camelus, brambling, Fringilla montifringilla, goldfinch, Carduelis carduelis, house finch, linnet, Carpodacus mexicanus, junco, snowbird, indigo bunting, indigo finch, indigo bird, Passerina cyanea, robin, American robin, Turdus migratorius, bulbul, jay, magpie, chickadee, water ouzel, dipper, kite, bald eagle, American eagle, Haliaeetus leucocephalus, vulture, great grey owl, great gray owl, Strix nebulosa, black grouse, ptarmigan, ruffed grouse, partridge, Bonasa umbellus, prairie chicken, prairie grouse, prairie fowl, peacock, quail, partridge, African grey, African gray, Psittacus erithacus, macaw, sulphur-crested cockatoo, Kakatoe galerita, Cacatua galerita, lorikeet, coucal, bee eater, hornbill, hummingbird, jacamar, toucan, drake, red-breasted merganser, Mergus serrator, goose, black swan, Cygnus atratus, white stork, Ciconia ciconia, black stork, Ciconia nigra, spoonbill, flamingo, American egret, great white heron, Egretta albus, little blue heron, Egretta caerulea, bittern, crane, limpkin, Aramus pictus, American coot, marsh hen, mud hen, water hen, Fulica americana, bustard, ruddy turnstone, Arenaria interpres, red-backed sandpiper, dunlin, Erolia alpina, redshank, Tringa totanus, dowitcher, oystercatcher, oyster catcher, European gallinule, Porphyrio porphyrio, pelican, king penguin, Aptenodytes patagonica, albatross, mollymawk, great white shark, white shark, man-eater, man-eating shark, Carcharodon carcharias, tiger shark, Galeocerdo cuvieri, hammerhead, hammerhead shark, electric ray, crampfish, numbfish, torpedo, stingray, barracouta, snoek, coho, cohoe, coho salmon, blue jack, silver salmon, Oncorhynchus kisutch, tench, Tinca tinca, goldfish, Carassius auratus, eel, rock beauty, Holocanthus tricolor, anemone fish, lionfish, puffer, pufferfish, blowfish, globefish, sturgeon, gar, garfish, garpike, billfish, Lepisosteus osseus, loggerhead, loggerhead turtle, Caretta caretta, mud turtle, terrapin, box turtle, box tortoise, banded gecko, common iguana, iguana, Iguana iguana, American chameleon, anole, Anolis carolinensis, whiptail, whiptail lizard, agama, frilled lizard, Chlamydosaurus kingi, alligator lizard, Gila monster, Heloderma suspectum, green lizard, Lacerta viridis, African chameleon, Chamaeleo chamaeleon, Komodo dragon, Komodo lizard, dragon lizard, giant lizard, Varanus komodoensis, triceratops, African crocodile, Nile crocodile, Crocodylus niloticus, American alligator, Alligator mississipiensis, thunder snake, worm snake, Carphophis amoenus, ringneck snake, ring-necked snake, ring snake, hognose snake, puff adder, sand viper, green snake, grass snake, king snake, kingsnake, garter snake, grass snake, water snake, vine snake, night snake, Hypsiglena torquata, boa constrictor, Constrictor constrictor, rock python, rock snake, Python sebae, Indian cobra, Naja naja, green mamba, sea snake, horned viper, cerastes, sand viper, horned asp, Cerastes cornutus, stone wall, sidewinder, horned rattlesnake, Crotalus cerastes, European fire salamander, Salamandra salamandra, common newt, Triturus vulgaris, eft, spotted salamander, Ambystoma maculatum, axolotl, mud puppy, Ambystoma mexicanum, bullfrog, Rana catesbeiana, tree frog, tree-frog, tailed frog, bell toad, ribbed toad, tailed toad, Ascaphus trui, whistle, wing, paintbrush, oxygen mask, snorkel, loudspeaker, speaker, speaker unit, loudspeaker system, speaker system, microphone, mike, screen, CRT screen, mouse, computer mouse, electric fan, blower, oil filter, strainer, space heater, stove, guillotine, barometer, rule, ruler, odometer, hodometer, mileometer, milometer, scale, weighing machine, analog clock, digital clock, wall clock, hourglass, sundial, parking meter, stopwatch, stop watch, digital watch, stethoscope, syringe, magnetic compass, binoculars, field glasses, opera glasses, projector, sunglasses, dark glasses, shades, loupe, jeweler's loupe, radio telescope, radio reflector, bow, cannon, assault rifle, assault gun, rifle, projectile, missile, computer keyboard, keypad, typewriter keyboard, crane, lighter, light, igniter, ignitor, abacus, cash machine, cash dispenser, automated teller machine, automatic teller machine, automated teller, automatic teller, ATM, slide rule, slipstick, desktop computer, hand-held computer, hand-held microcomputer, notebook, notebook computer, padlock, harvester, reaper, thresher, thrasher, threshing machine, printer, slot, one-armed bandit, vending machine, sewing machine, joystick, switch, electric switch, electrical switch, hook, claw, car wheel, paddlewheel, paddle wheel, pinwheel, potter's wheel, gas pump, gasoline pump, petrol pump, island dispenser, carousel, carrousel, merry-go-round, roundabout, whirligig, swing, reel, radiator, puck, hockey puck, hard disc, hard disk, fixed disk, sunglass, pick, plectrum, plectron, car mirror, solar dish, solar collector, solar furnace, remote control, remote, disk brake, disc brake, buckle, hair slide, knot, combination lock, web site, website, internet site, site, nail, safety pin, screw, muzzle, seat belt, seatbelt, ski, candle, taper, wax light, jack-o'-lantern, spotlight, spot, torch, neck brace, pier, tripod, maypole, hand blower, blow dryer, blow drier, hair dryer, hair drier, mousetrap, spider web, spider's web, trilobite, harvestman, daddy longlegs, Phalangium opilio, scorpion, black and gold garden spider, Argiope aurantia, barn spider, Araneus cavaticus, garden spider, Aranea diademata, black widow, Latrodectus mactans, tarantula, wolf spider, hunting spider, tick, centipede, isopod, Dungeness crab, Cancer magister, rock crab, Cancer irroratus, fiddler crab, king crab, Alaska crab, Alaskan king crab, Alaska king crab, Paralithodes camtschatica, American lobster, Northern lobster, Maine lobster, Homarus americanus, spiny lobster, langouste, rock lobster, crawfish, crayfish, sea crawfish, crayfish, crawfish, crawdad, crawdaddy, hermit crab, tiger beetle, ladybug, ladybeetle, lady beetle, ladybird, ladybird beetle, ground beetle, carabid beetle, long-horned beetle, longicorn, longicorn beetle, leaf beetle, chrysomelid, dung beetle, rhinoceros beetle, weevil, fly, bee, grasshopper, hopper, cricket, walking stick, walkingstick, stick insect, cockroach, roach, mantis, mantid, cicada, cicala, leafhopper, lacewing, lacewing fly, dragonfly, darning needle, devil's darning needle, sewing needle, snake feeder, snake doctor, mosquito hawk, skeeter hawk, damselfly, admiral, ringlet, ringlet butterfly, monarch, monarch butterfly, milkweed butterfly, Danaus plexippus, cabbage butterfly, sulphur butterfly, sulfur butterfly, lycaenid, lycaenid butterfly, jellyfish, sea anemone, anemone, brain coral, flatworm, platyhelminth, nematode, nematode worm, roundworm, conch, snail, slug, sea slug, nudibranch, chiton, coat-of-mail shell, sea cradle, polyplacophore, sea urchin, sea cucumber, holothurian, iron, smoothing iron, espresso maker, microwave, microwave oven, Dutch oven, rotisserie, toaster, waffle iron, vacuum, vacuum cleaner, dishwasher, dish washer, dishwashing machine, refrigerator, icebox, washer, automatic washer, washing machine, Crock Pot, frying pan, frypan, skillet, wok, caldron, cauldron, coffeepot, teapot, spatula, altar, triumphal arch, patio, terrace, steel arch bridge, suspension bridge, viaduct, barn, greenhouse, nursery, glasshouse, palace, monastery, library, apiary, bee house, boathouse, church, church building, mosque, stupa, tope, planetarium, restaurant, eating house, eating place, eatery, cinema, movie theater, movie theatre, movie house, picture palace, home theater, home theatre, lumbermill, sawmill, coil, spiral, volute, whorl, helix, obelisk, totem pole, castle, prison, prison house, grocery store, grocery, food market, market, bakery, bakeshop, bakehouse, barbershop, bookshop, bookstore, bookstall, butcher shop, meat market, confectionery, confectionary, candy store, shoe shop, shoe-shop, shoe store, tobacco shop, tobacconist shop, tobacconist, toyshop, fountain, cliff dwelling, yurt, dock, dockage, docking facility, brass, memorial tablet, plaque, megalith, megalithic structure, bannister, banister, balustrade, balusters, handrail, breakwater, groin, groyne, mole, bulwark, seawall, jetty, dam, dike, dyke, chainlink fence, picket fence, paling, worm fence, snake fence, snake-rail fence, Virginia fence, diamondback, diamondback rattlesnake, Crotalus adamanteus, grille, radiator grille, sliding door, turnstile, mountain tent, scoreboard, honeycomb, plate rack, pedestal, plinth, footstall, beacon, lighthouse, beacon light, pharos, leatherback turtle, leatherback, leathery turtle, Dermochelys coriacea, mashed potato, bell pepper, head cabbage, broccoli, cauliflower, zucchini, courgette, spaghetti squash, acorn squash, butternut squash, cucumber, cuke, artichoke, globe artichoke, cardoon, mushroom, shower curtain, jean, blue jean, denim, carton, handkerchief, hankie, hanky, hankey, sandal, ashcan, trash can, garbage can, wastebin, ash bin, ash-bin, ashbin, dustbin, trash barrel, trash bin, safe, plate, necklace, croquet ball, fur coat, thimble, pajama, pyjama, pj's, jammies, running shoe, oboe, hautboy, hautbois, cocktail shaker, chest, manhole cover, modem, tub, vat, tray, balance beam, beam, bagel, beigel, violin, fiddle, prayer rug, prayer mat, kimono, hot pot, hotpot, whiskey jug, knee pad, book jacket, dust cover, dust jacket, dust wrapper, spindle, ski mask, beer bottle, crash helmet, bottlecap, tile roof, mask, maillot, Petri dish, football helmet, bathing cap, swimming cap, teddy, teddy bear, holster, pop bottle, soda bottle, photocopier, vestment, crossword puzzle, crossword, golf ball, trifle, suit, suit of clothes, water tower, feather boa, boa, cloak, red wine, drumstick, shield, buckler, Christmas stocking, hoopskirt, crinoline, menu, stage, bonnet, poke bonnet, meat loaf, meatloaf, baseball, face powder, scabbard, sunscreen, sunblock, sun blocker, beer glass, hen-of-the-woods, hen of the woods, Polyporus frondosus, Grifola frondosa, guacamole, wool, woolen, woollen, hay, bow tie, bow-tie, bowtie, mailbag, postbag, water jug, bucket, pail, dishrag, dishcloth, soup bowl, eggnog, mortar, trench coat, paddle, boat paddle, chain, swab, swob, mop, mixing bowl, potpie, wine bottle, shoji, bulletproof vest, drilling platform, offshore rig, binder, ring-binder, cardigan, sweatshirt, pot, flowerpot, birdhouse, jinrikisha, ricksha, rickshaw, hamper, ping-pong ball, pencil box, pencil case, consomme, apron, punching bag, punch bag, punching ball, punchball, backpack, back pack, knapsack, packsack, rucksack, haversack, groom, bridegroom, bearskin, busby, shako, pencil sharpener, broom, abaya, mortarboard, poncho, crutch, Polaroid camera, Polaroid Land camera, space bar, cup, racket, racquet, traffic light, traffic signal, stoplight, quill, quill pen, radio, wireless, snow leopard, ounce, Panthera uncia, dough, cuirass, military uniform, lipstick, lip rouge, shower cap, monitor, oscilloscope, scope, cathode-ray oscilloscope, CRO, mitten, brassiere, bra, bandeau, French loaf, vase, milk can, rugby ball, paper towel, earthstar, envelope, miniskirt, mini, cowboy hat, ten-gallon hat, trolleybus, trolley coach, trackless trolley, perfume, essence, bathtub, bathing tub, bath, tub, hotdog, hot dog, red hot, coral fungus, bullet train, bullet, pillow, toilet tissue, toilet paper, bathroom tissue, cassette, carpenter's kit, tool kit, ladle, stinkhorn, carrion fungus, lotion, hair spray, academic gown, academic robe, judge's robe, dome, crate, wig, burrito, pill bottle, chain mail, ring mail, mail, chain armor, chain armour, ring armor, ring armour, theater curtain, theatre curtain, window shade, barrel, cask, washbasin, handbasin, washbowl, lavabo, wash-hand basin, ballpoint, ballpoint pen, ballpen, Biro, basketball, bath towel, cowboy boot, gown, window screen, agaric, standard poodle, cellular telephone, cellular phone, cellphone, cell, mobile phone, nipple, barbell, mailbox, letter box, lab coat, laboratory coat, fire screen, fireguard, minibus, packet, brown bear, bruin, Ursus arctos, pole, horizontal bar, high bar, sombrero, pickelhaube, rain barrel, wallet, billfold, notecase, pocketbook, cassette player, comic book, piggy bank, penny bank, street sign, bell cote, bell cot, fountain pen, Windsor tie, volleyball, overskirt, sarong, purse, bolo tie, bolo, bola tie, bola, bib, parachute, chute, sleeping bag, television, television system, swimming trunks, bathing trunks, measuring cup, espresso, pizza, pizza pie, breastplate, aegis, egis, shopping basket, wooden spoon, saltshaker, salt shaker, chocolate sauce, chocolate syrup, ballplayer, baseball player, goblet, gyromitra, stretcher, water bottle, skunk, polecat, wood pussy, soap dispenser, jersey, T-shirt, tee shirt, school bus, jigsaw puzzle, plastic bag, reflex camera, diaper, nappy, napkin, Band Aid, ice lolly, lolly, lollipop, popsicle, velvet, tennis ball, gasmask, respirator, gas helmet, doormat, welcome mat, Loafer, ice cream, icecream, pretzel, quilt, comforter, comfort, puff, maillot, tank suit, tape player, clog, geta, patten, sabot, iPod, bolete, meerkat, mierkat, scuba diver, pitcher, ewer, matchstick, bikini, two-piece, sock, CD player, lens cap, lens cover, thatch, thatched roof, vault, beaker, bubble, cheeseburger, parallel bars, bars, flagpole, flagstaff, coffee mug, rubber eraser, rubber, pencil eraser, stole, carbonara, dumbbell, , Synsets new in ILSVRC2012, Siberian husky, English springer, English springer spaniel, malamute, malemute, Alaskan malamute, Walker hound, Walker foxhound, Welsh springer spaniel, whippet, Weimaraner, soft-coated wheaten terrier, Dandie Dinmont, Dandie Dinmont terrier, Old English sheepdog, bobtail, otterhound, otter hound, bloodhound, sleuthhound, Airedale, Airedale terrier, giant schnauzer, black-and-tan coonhound, papillon, Mexican hairless, Cardigan, Cardigan Welsh corgi, malinois, Lhasa, Lhasa apso, Norwegian elkhound, elkhound, Rottweiler, Saluki, gazelle hound, schipperke, Brabancon griffon, West Highland white terrier, Sealyham terrier, Sealyham, Irish wolfhound, EntleBucher, French bulldog, Bernese mountain dog, Maltese dog, Maltese terrier, Maltese, Norfolk terrier, toy terrier, cairn, cairn terrier, groenendael, clumber, clumber spaniel, Afghan hound, Afghan, Japanese spaniel, borzoi, Russian wolfhound, toy poodle, Kerry blue terrier, Scotch terrier, Scottish terrier, Scottie, Boston bull, Boston terrier, Greater Swiss Mountain dog, Appenzeller, Shih-Tzu, Irish water spaniel, Pomeranian, Bedlington terrier, miniature schnauzer, collie, Irish terrier, affenpinscher, monkey pinscher, monkey dog, silky terrier, Sydney silky, beagle, Leonberg, German short-haired pointer, dhole, Cuon alpinus, Chesapeake Bay retriever, bull mastiff, kuvasz, pug, pug-dog, curly-coated retriever, Norwich terrier, keeshond, Lakeland terrier, standard schnauzer, Tibetan terrier, chrysanthemum dog, wire-haired fox terrier, basset, basset hound, chow, chow chow, American Staffordshire terrier, Staffordshire terrier, American pit bull terrier, pit bull terrier, Shetland sheepdog, Shetland sheep dog, Shetland, Great Pyrenees, Chihuahua, Labrador retriever, Samoyed, Samoyede, bluetick, kelpie, miniature pinscher, Italian greyhound, cocker spaniel, English cocker spaniel, cocker, Sussex spaniel, Pembroke, Pembroke Welsh corgi, Blenheim spaniel, Ibizan hound, Ibizan Podenco, English foxhound, briard, Border terrier

VGG19 architecture

VGG19 consists of 19 sequential layers (which in 2014 was considered deep). Current networks may go beyond 1000 layers.
Even with 19 layers, training is extremely slow, and alternative training methods are used, e.g. alternate training of sub-layers. The file with the weights for VGG19 is more than 500mb, i.e. more than 100m parameters!

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					



https://www.kaggle.com/keras/vgg19/home

Deep Dreams

- Based on the Inception V3 model.

- Animals are “overrepresented” in the pool of training images and therefore animal features dominate the final dream picture.

- Algorithmically, the dreams are produced by running convolutional nets in reverse. You keep the network weights fixed but alter the input image to match a pre-defined output

Style transfer

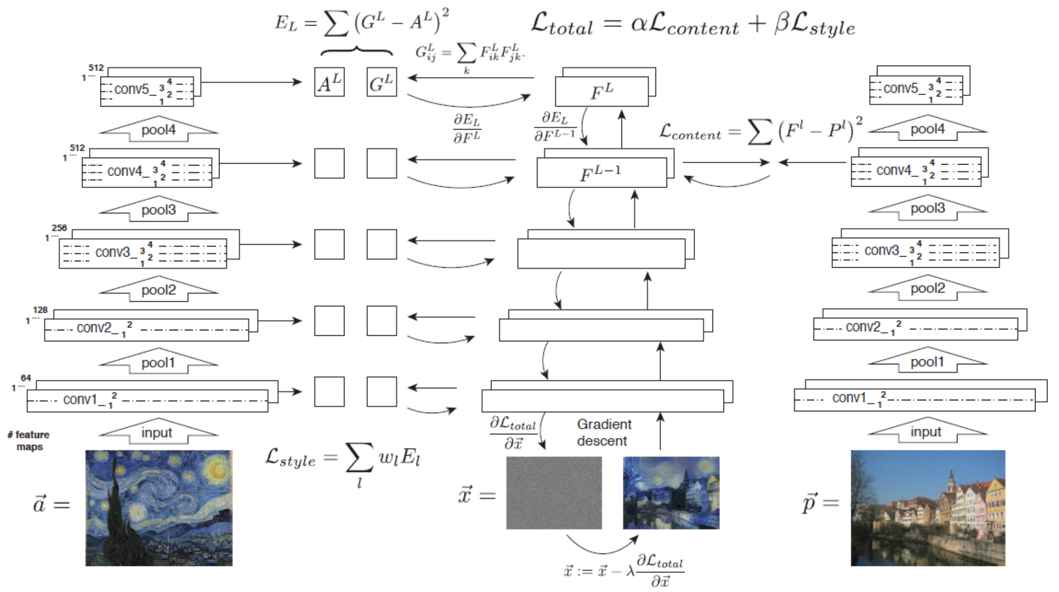
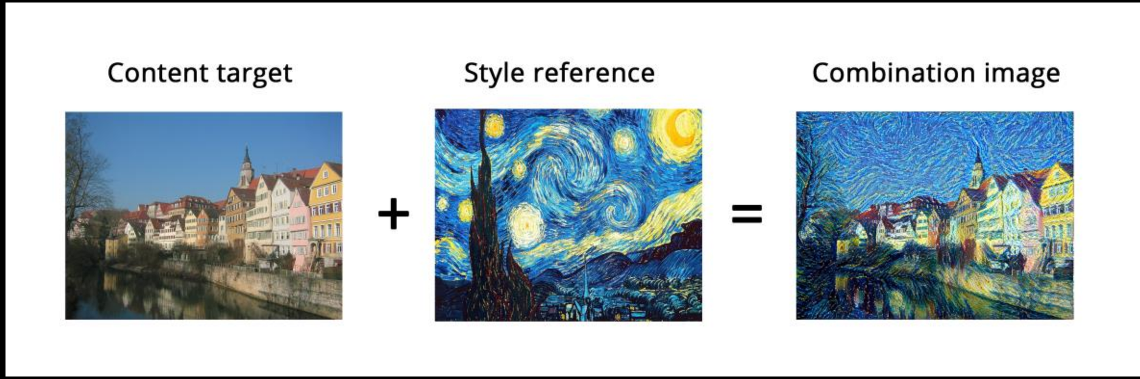


Figure 2. Style transfer algorithm. First content and style features are extracted and stored. The style image $\vec{a}$ is passed through the network and its style representation A^l on all layers included are computed and stored (left). The content image $\vec{p}$ is passed through the network and the content representation P^l in one layer is stored (right). Then a random white noise image $\vec{x}$ is passed through the network and its style features G^l and content features F^l are computed. On each layer included in the style representation, the element-wise mean squared difference between G^l and A^l is computed to give the style loss $\mathcal{L}_{style}$ (left). Also the mean squared difference between F^l and P^l is computed to give the content loss $\mathcal{L}_{content}$ (right). The total loss $\mathcal{L}_{total}$ is then a linear combination between the content and the style loss. Its derivative with respect to the pixel values can be computed using error back-propagation (middle). This gradient is used to iteratively update the image $\vec{x}$ until it simultaneously matches the style features of the style image $\vec{a}$ and the content features of the content image $\vec{p}$ (middle, bottom).

loss <- distance(style(reference_image) - style(generated_image)) + distance(content(original_image) - content(generated_image))



Taken from Gatys, Ecker and Bethge, IEEX Xplore 2016

Style transfer

style

+

image

=

Exercise 1: Gender prediction using the Keras library

We now return to gender prediction data set, but instead of focusing on the Big Five Inventory, we consider an enlarged data set with 272 questions and 942 participants. Based on these questions, we then ask how well we now can predict gender!

```
# Load the data file and convert it to a matrix
dat1=read.table("RGender_all_filt.dat",stringsAsFactors = F)
dat1=(as.matrix(dat1))

# basic info about data file
str(dat1)

##  num [1:942, 1:272] 1 1 1 1 0 1 0 1 1 1 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:942] "u1" "u2" "u3" "u4" ...
##   ..$ : chr [1:272] "gender" "function_duties.answer" "narcissism_c
```

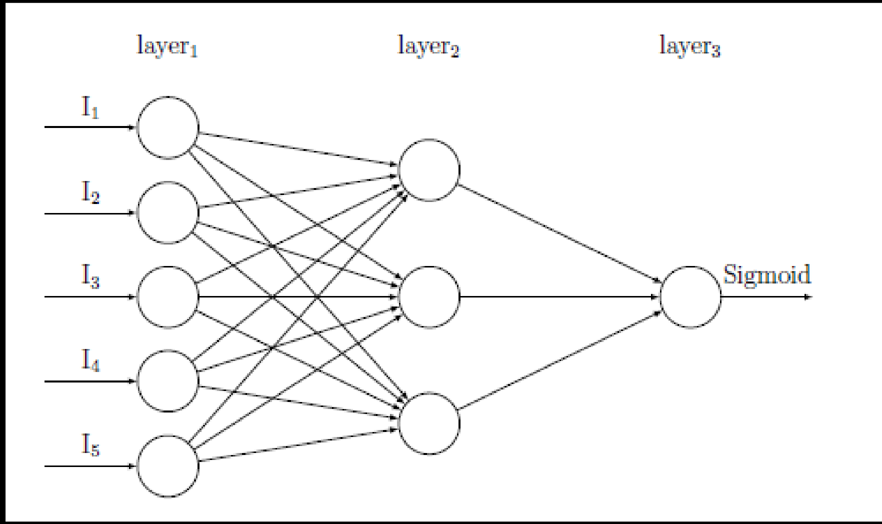
The data is a 1D tensor, with the format

(n, [answers])

Where [answers] is a 272 dimensional vector.

Gender prediction using the Keras library

We shall use a dense neural network for the classification. Convolutional networks are of less use for this data.



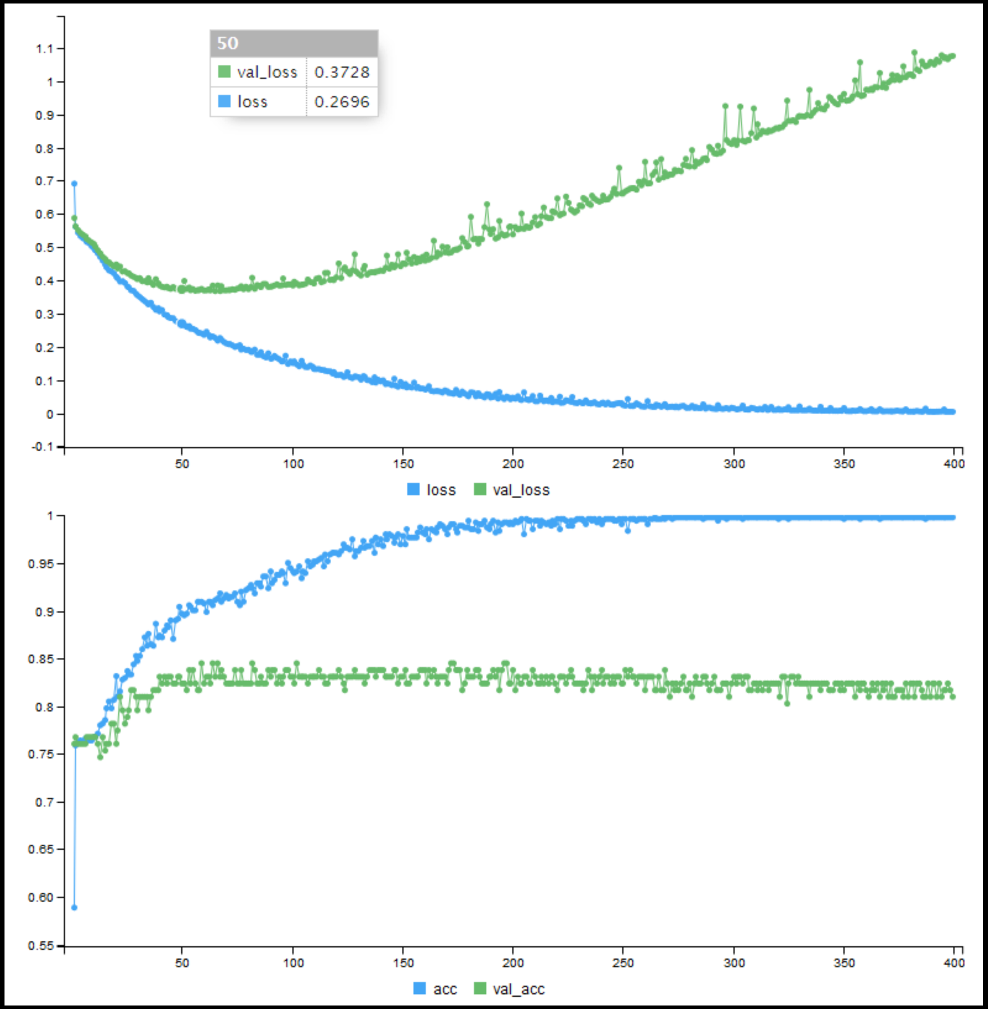

```
model %>%
  layer_dense(units = 5, activation = 'relu', input_shape = c(nc)) %>%
  layer_dense(units = 3, activation = 'relu') %>%
  layer_dense(units = 1, activation = 'sigmoid')
```

```
summary(model)


## ___________________________________________________________________________
## Layer (type)                     Output Shape                  Param #
## ===========================================================================
## dense_1 (Dense)                  (None, 5)                     1360
## ___________________________________________________________________________
## dense_2 (Dense)                  (None, 3)                     18
## ___________________________________________________________________________
## dense_3 (Dense)                  (None, 1)                     4
## ===========================================================================
## Total params: 1,382
## Trainable params: 1,382
## Non-trainable params: 0
## ___________________________________________________________________________
```

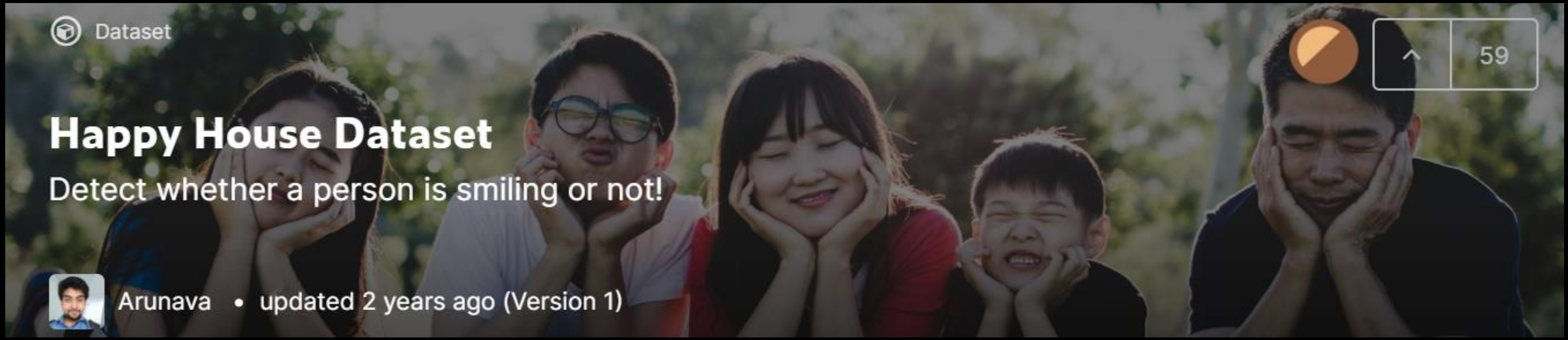
Playing around with different numbers of layers, how well can we predict gender? Here we use a 5-3-1 network, with relu activated units and a sigmoid output unit, representing the probability for female/male.

```
history <- model %>% fit(
  x_learn, y_learn,
  epochs =400, batch_size = 100,
  validation_split = .2
  #validation_data = list(x_valid,y_valid)
)
```

```
model %>% evaluate(x_test, y_test)

## $loss
## [1] 0.3609048
##
## $acc
## [1] 0.8347458
```



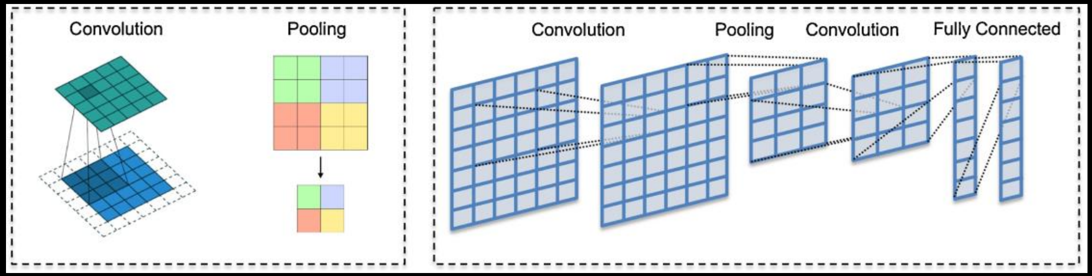
Exercise 2: Predict happy faces in the HappyHouse data



The data consists of 600 training images of 64x64x3 pixels which are either happy or not happy. An additional 150 images are retained for model testing.

Build a model combing Conv layers, pooling and finally a densely connected output module.

Choose an appropriate optimizer.




```
model = Sequential()
model.add(Conv2D(filters = 32, kernel_size = (3,3),padding = 'Same', input_shape = (64,64,3),activation ='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Conv2D(filters = 32, kernel_size = (3,3),padding = 'Same', activation ='relu'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Conv2D(filters = 32, kernel_size = (3,3),padding = 'Same', activation ='relu',name='lastconv'))
model.add(MaxPool2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(units=16,activation='relu',kernel_initializer="uniform"))
#Output Layer
model.add(Dense(1, kernel_initializer="uniform", activation = 'sigmoid',name='prediction_layer'))
opt = Nadam(beta_1=0.9, beta_2=0.999, epsilon=1e-07)
#opt = RMSprop()
# Compiling Neural Network
model.compile(optimizer = opt, loss = 'binary_crossentropy', metrics = ['accuracy'])
```

https://www.kaggle.com/iarunava/happy-house-dataset

Improve convergence

Optimizers

Stochastic gradient descent

Stochastic gradient descent (SGD) in contrast performs a parameter update for *each* training example $x^{(i)}$ and label $y^{(i)}$:

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^{(i)}; y^{(i)}).$$

Batch gradient descent performs redundant computations for large datasets, as it recomputes gradients for similar examples before each parameter update. SGD does away with this redundancy by performing one update at a time. It is therefore usually much faster and can also be used to learn online. SGD performs frequent updates with a high variance that cause the objective function to fluctuate heavily as in Image 1.

RMSprop

RMSprop is an unpublished, adaptive learning rate method proposed by Geoff Hinton in Lecture 6e of his Coursera Class.

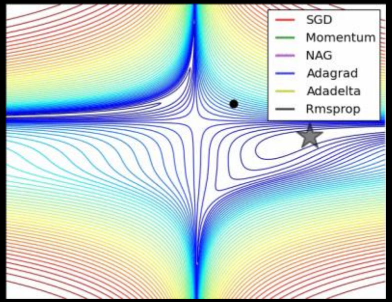
RMSprop and Adadelta have both been developed independently around the same time stemming from the need to resolve Adagrad's radically diminishing learning rates. RMSprop in fact is identical to the first update vector of Adadelta that we derived above:

$$E[g^2]_t = 0.9E[g^2]_{t-1} + 0.1g_t^2$$
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

Adam

Adaptive Moment Estimation (Adam) [14] is another method that computes adaptive learning rates for each parameter. In addition to storing an exponentially decaying average of past squared gradients v_t like Adadelta and RMSprop, Adam also keeps an exponentially decaying average of past gradients m_t, similar to momentum. Whereas momentum can be seen as a ball running down a slope, Adam behaves like a heavy ball with friction, which thus prefers flat minima in the error surface [15]. We compute the decaying averages of past and past squared gradients m_t and v_t respectively as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$



https://ruder.io/optimizing-gradient-descent/index.html

Dropout

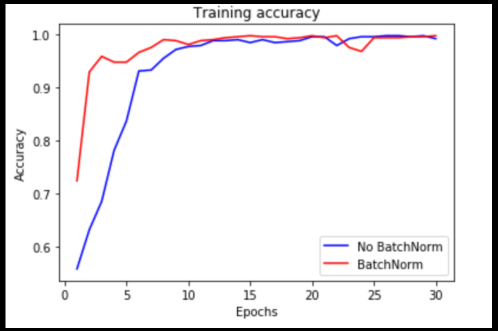
The dropout rate determines a proportion of neurons to be randomly ignored during each training round. You remove temporally weight updates and information transmission in parts of the network. Effectively reduces overfitting.

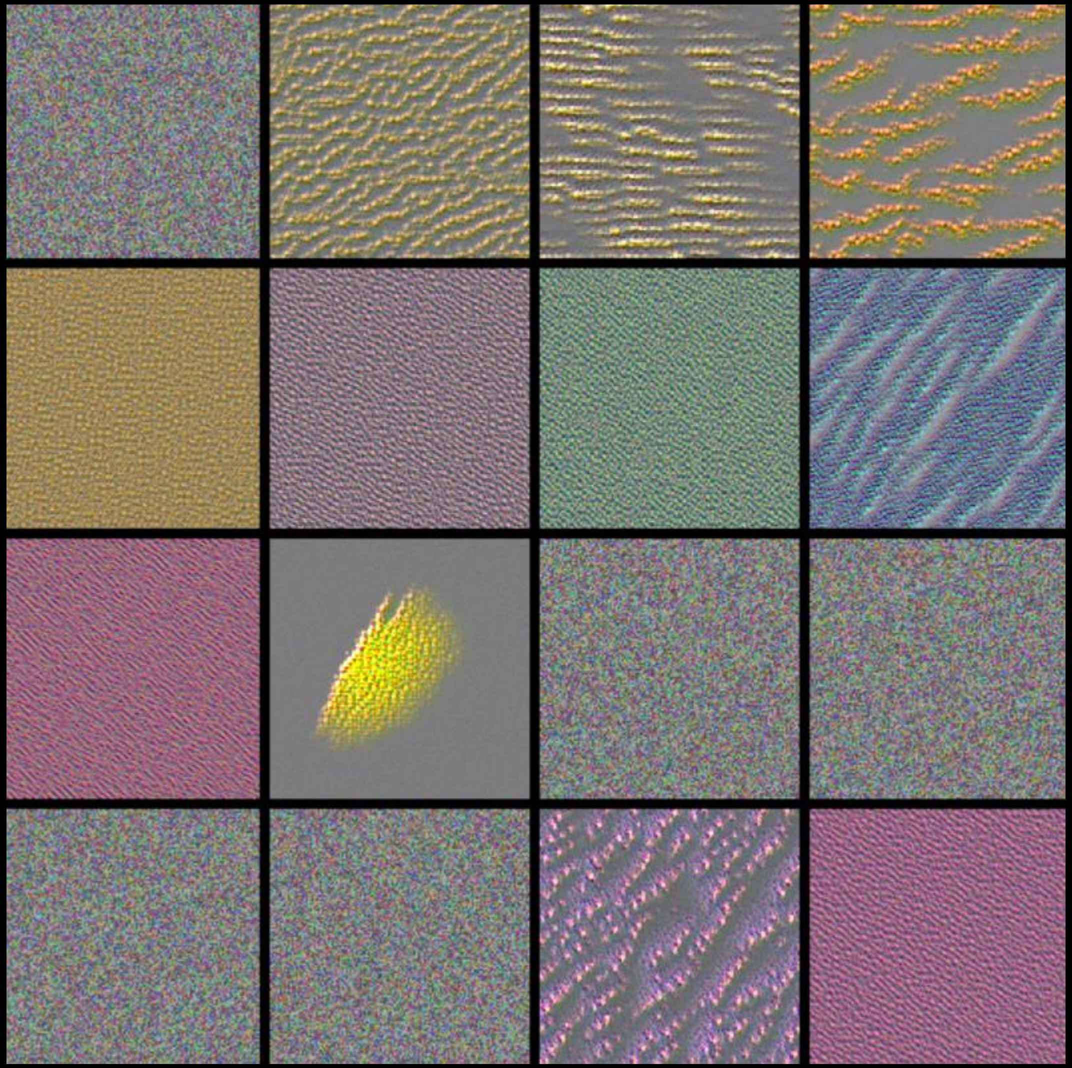
```
from keras.layers import Dropout
model = keras.models.Sequential()
n = x_learn.shape[1]
model.add(keras.layers.Dense(8, activation='relu', input_shape=(n,)))
model.add(Dropout(0.4))
model.add(keras.layers.Dense(8, activation='relu'))
model.add(Dropout(0.25))
```

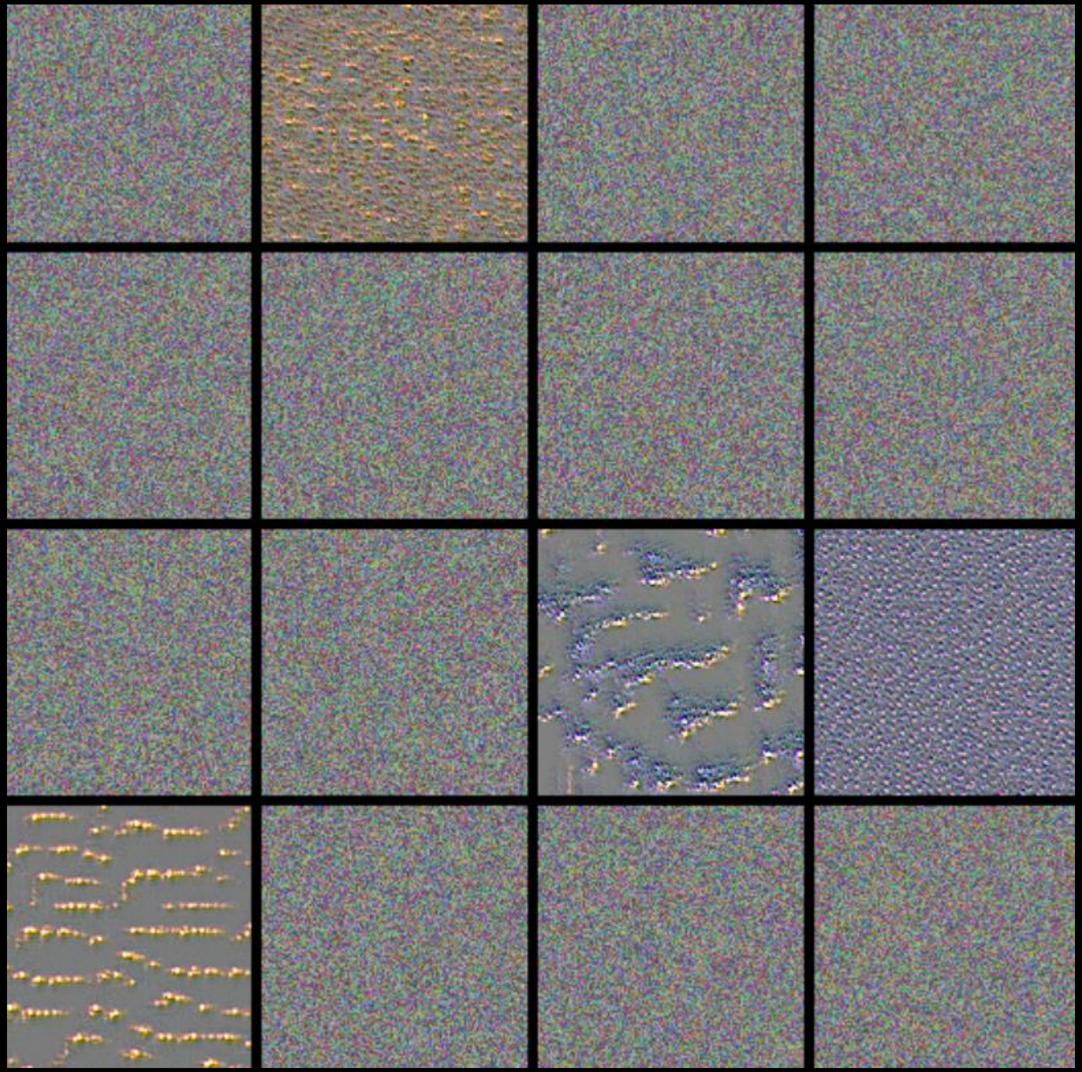
Batch Normalization

"Batch normalization has been arguably one of the most successful architectural innovations in deep learning. But even though its effectiveness is indisputable, we do not have a firm understanding of why this is the case. Broadly speaking, BatchNorm is a mechanism that aims to stabilize the distribution (over a minibatch) of inputs to a given network layer during training. This is achieved by augmenting the network with additional layers that set the first two moments (mean and variance) of the distribution of each activation to be zero and one respectively. Then, the batch normalized inputs are also typically scaled and shifted based on trainable parameters to preserve model expressivity."

https://arxiv.org/pdf/1805.11604.pdf







HappyHouse GPU VS CPU

```
physical_device_desc: "device: 0, name: GeForce RTX 2080 Ti, pci bus
id: 0000:41:00.0, compute capability: 7.5" ]
```

```
2020-08-10 07:00:33.673154: I tensorflow/core/platform/cpu_feature_guard.cc:142]
This TensorFlow binary is optimized with Intel(R) MKL-DNN to use the following C
PU instructions in performance-critical operations:  SSE3 SSE4.1 SSE4.2 AVX AVX2
AVX512F FMA
```

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_90 (Conv2D)           (None, 64, 64, 64)        1792
_________________________________________________________________
max_pooling2d_41 (MaxPooling (None, 32, 32, 64)        0
_________________________________________________________________
conv2d_91 (Conv2D)           (None, 32, 32, 64)        36928
_________________________________________________________________
max_pooling2d_42 (MaxPooling (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_92 (Conv2D)           (None, 16, 16, 64)        36928
_________________________________________________________________
max_pooling2d_43 (MaxPooling (None, 8, 8, 64)          0
_________________________________________________________________
flatten_15 (Flatten)         (None, 4096)              0
_________________________________________________________________
dense_16 (Dense)             (None, 1024)              4195328
_________________________________________________________________
prediction_layer (Dense)     (None, 1)                 1025
=================================================================
Total params: 4,272,001
Trainable params: 4,272,001
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/100
6/6 [==============================] - 0s 29ms/step - loss: 0.7646 - accuracy: 0.5185 - val_loss: 0.6871 - val_accuracy: 0.4833
Epoch 2/100
6/6 [==============================] - 0s 15ms/step - loss: 0.6926 - accuracy: 0.5463 - val_loss: 0.6674 - val_accuracy: 0.8000
Epoch 3/100
6/6 [==============================] - 0s 16ms/step - loss: 0.6710 - accuracy: 0.5926 - val_loss: 0.6540 - val_accuracy: 0.5167
Epoch 4/100
6/6 [==============================] - 0s 14ms/step - loss: 0.6398 - accuracy: 0.6130 - val_loss: 0.6301 - val_accuracy: 0.5167
```

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 64, 64, 64)        1792
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 32, 32, 64)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 32, 32, 64)        36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 16, 16, 64)        36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 4096)              0
_________________________________________________________________
dense (Dense)                (None, 1024)              4195328
_________________________________________________________________
prediction_layer (Dense)     (None, 1)                 1025
=================================================================
Total params: 4,272,001
Trainable params: 4,272,001
Non-trainable params: 0
_________________________________________________________________
None
Epoch 1/100
6/6 [==============================] - 1s 133ms/step - loss: 0.7755 - accuracy: 0.5074 - val_loss: 0.6692 - val_accuracy: 0.8333
Epoch 2/100
6/6 [==============================] - 1s 121ms/step - loss: 0.6987 - accuracy: 0.6241 - val_loss: 0.6736 - val_accuracy: 0.7000
Epoch 3/100
6/6 [==============================] - 1s 121ms/step - loss: 0.6791 - accuracy: 0.6704 - val_loss: 0.6490 - val_accuracy: 0.6167
Epoch 4/100
6/6 [==============================] - 1s 120ms/step - loss: 0.6305 - accuracy: 0.6370 - val_loss: 0.7825 - val_accuracy: 0.5167
```

Gaming graphics card 1k€

80 cores, 160 threads, ~25k€