UiT
NORGES ARKTISKE UNIVERSITET

EXAMINATION PAPER

Home Exam in: Fys-3012 Pattern Recognition

Hand-out: Wednesday 18th Oct. 2017 ~ 1500.

Hand-in: Thursday 2nd Nov. 2017 (0900 at latest)

The exam contains 12 pages including this cover page and appendix.

Contact persons:

Robert Jenssen

Email: robert.jenssen@uit.no

Sigurd Løkse

Email sigurd.lokse@uit.no

Introduction

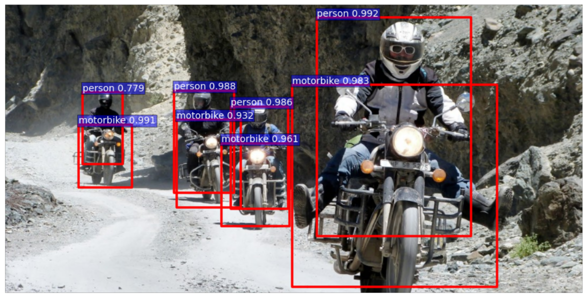

Figure 1: Machine learning power much used technology. Here represented by object detection, localization, and recognition. Figure from `https://www.saagie.com/blog/object-detection-part1`.

Machine learning and pattern recognition are essential for much of today's modern technology. Such technology is found in object recognition and localization in images as shown in Fig. 2, in music recognition (e.g. Shazam app), in health and medicine, as well as in analysis of consumer and customer patterns, to name just a few. Mathematics and statistics are at the core, and new methodology always starts by formulating mathematical optimization problems to be solved. This is the foundation for data analysis research both in academia and in the ICT-companies such as Google, IBM etc.

We will in this home exam develop such technology based on algorithms where the input to the analysis chain is data in the form of various types of images.

Two aspects in particular will be important in addition to be able to program and execute our algorithms, namely

- That we explain mathematically how the methods we use work;
- That we discuss pros and cons.

We have to document our work, and to write a report. More details on how to hand in the report and how to append code in the report will follow in Fronter.

In this home exam, you are free to write your report and to investigate (program, execute and explain) those machine learning and pattern recognition methods you want to, including data transformation and dimensionality reduction methods. The more comprehensive report, the better, of course. When it comes to classification, you should however at least as a minimum at some point investigate the following methods:

- Least (sum of) squares (LS) classifier;
- Two-layer perceptron (2LP);

- Bayes classifier assuming normal data with equal covariance structure;
- Support vector machine (SVM).

Useful SVM Matlab and Python software are provided in Fronter since this method requires optimization knowledge beyond Fys-3012. For Matlab, you have:

1. `SMO2.m`
2. `svcplot_book.m (needs cmap.mat)`
3. `calkKernel.m`

Recall that for the hyperplane $\mathbf{w}^\top \mathbf{x} + b$, SMO2 outputs $-b$.

Data sets are available in Fronter in the folder "DATA" stored in Matlab-format. See Appendix A for a comment on the data sets.

<u>Note</u>: This home exam is NOT about feeding some data to some software library that someone else made (e.g. *Scikit-learn*). There is no way to know if people have understood something by pressing a button and hoping for the best. You will therefore program all algorithms from scratch, except the SVM optimization procedure, as mentioned earlier.

Students normally program in Matlab or Python. You are however free to use whatever programming language you want to. The code should be commented in such a way that any person with programming knowledge should be able to understand how the program works. Further instructions on the hand-in format will follow in Fronter. For those of you using Python, you may use *SciPy* to load the Matlab data files. See

`http://docs.scipy.org/doc/scipy/reference/tutorial/io.html`

for more info.

Good luck!

Object Recognition by Deep Convolutional Neural Networks

Deep convolutional neural networks are specifically designed for various types of image recognition. Imagine e.g. a self-driving car. The car need to scan its surroundings and take decisions on the type of objects in its vicinity. Deep convolutional neural networks excel at this, and have had profound impact. The convolutional aspect ensures a certain robustness to image object translations and rotations. Convolutional networks by-pass the hand-crafted feature engineering stage which is often a pre-cursor to classification algorithms in other domains of machine learning. It has been shown that hidden layers in the network pick up properties of images in an increasing order of complexity. See Fig. 2 for an illustration.

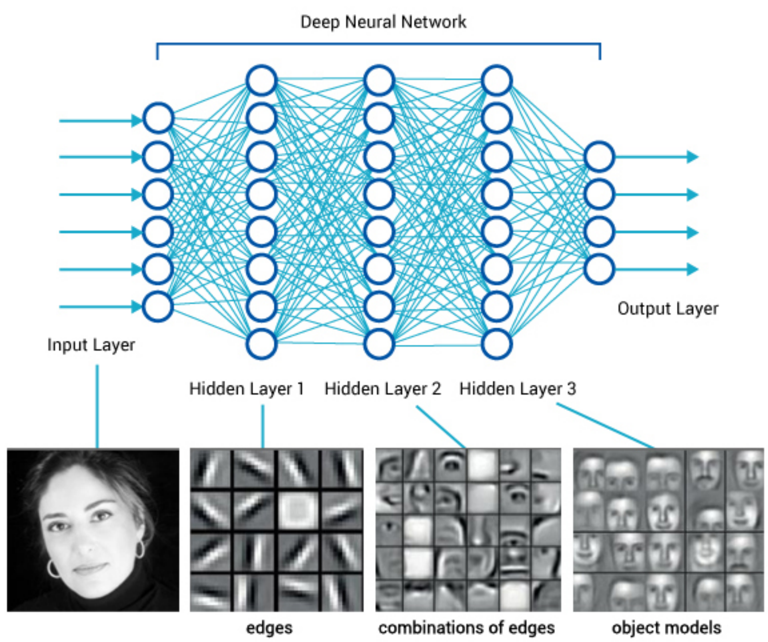

Figure 2: Illustration of a deeop neural networks. Figure from `https://www.saagie.com/blog/object-detection-part1`.

There are two main ways to utilize deep convolutional neural networks. If you have enough labeled images, the network can be trained from scratch. However, this is often not the case. The other approach is to use a so-called pretrained network (publicly available online). In that case, the network is optimized based on the *ImageNet* data set, which is a 1000-class data set of images

[1]. Then, the pre-trained network is *fine-tuned* on whatever image data set you have (insert a new output layer). Finally, the features of the last fully connected layer is taken to represent your particular images. The dimensionality of the last fully connected layer is often 4096. Hence, each of your images will be represented by a 4096-dimensional vector.

There are several options for publicly available pre-trained convolutional neural networks. One is the AlexNet[2]. For the AlexNet, features would be extracted from FC7, as seen in Fig. 3.

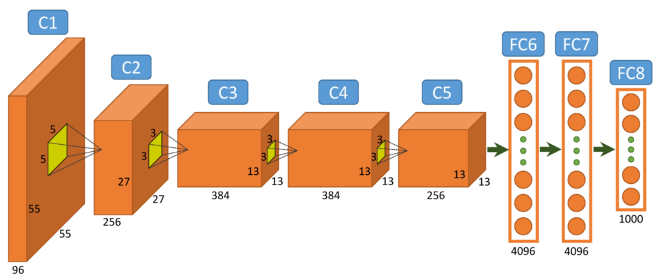

Figure 3: Figure from `https://www.saagie.com/blog/object-detection-part1`.

Problem 1: Seals recognition

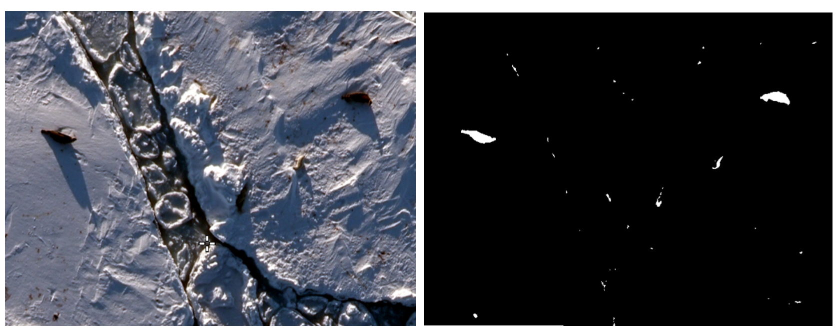

Figure 4: Aerial images (left) and potential locations of seals (right).

The following is a current problem of great importance to the Institute of Marine Research (IMR - Havforskningsinstituttet). It is very important to be able to estimate the population of seals, and in particular seal pups. The most effective way to do this is from aerial images. However, this is a very challenging problem. Fig. 4 (left) shows an example image. One particular challenge is to distinguish

[1]http://www.image-net.org

[2]https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/

between harp seal pups and hooded seal pups. We have access to a dataset consisting of several thousand aerial RGB images acquired during surveys in the West Ice east of Greenland in 2007 and 2012 and east of New Foundland, Canada, in 2012. The images are acquired from approximately 300m altitude, and the pixel spacing is about 3cm (depending on the exact flight altitude). A typical image size is 11500×7500 pixels. From these images potential locations of seals are identified, Fig. 4 (right), and then crops are extracted. A crop is shown in Fig. 5. The Institute of Marine Research, Norway, and the Northwest Atlantic Fisheries Centre, Canada, have provided the images and the ground truth information about the type and location of the seal pups in the images.

Figure 5: An image crop. A classifier needs to determine if this a harp seal pup or a hooded seal pup!

Our challenge is to classify images into the two categories *harp seal pups* or *hooded seal pups*.

You have a very nice colleague (let's call him S. Løkse) who has downloaded the weights for a pre-trained network from the internet, and who has *fine-tuned* the final layers using a subset of the seal images[3]. This subset of images was then discarded and the remaining images were processed by the network to produce the data provided to you, corresponding to the last fully-connected layer, i.e. as 4096-dimensional vectors. The dataset has been balanced, such that the two classes are represented equally (same number of samples from each class).

(*) This is what you will work on. Note that the dataset is not divided into a training set, a validation set, and a test set, so you will need to do this yourself. A common approach is to choose a size for the training set and split the remaining data equally into a validation set and a test set. The training set should be used to train the classifier. The validation set should

[3]The images were upscaled to 227×227 px to fit the AlexNet architecture.

be used to set any hyper-parameters. The accuracy of the test set should be considered the final accuracy of the classifier.

The results might depend on the size of these datasets, so some experimentation is needed to find the best size. You will experience that the classification accuracy might be different if you resample the data sets (even with the same size). Thus, to provide an accurate assessment for the best classification system for this data, you will need to provide statistics[4] for the different classifiers and different sizes of the training set.

You will have access to the original seal images that correspond to each 4096-dimensional vector, that you can use for plotting and analysis and to shed light in the results.

Chlorophyll - An Indicator of the Health of Plants

The amount of chlorophyll in plants is the most important indicator for the health of the plants. Climate changes, for example, may alter the chlorophyll content of the vegetation. The best pattern recognition method and people will provide the best automated monitoring of chlorophyll and hence for monitoring the health of the planet's vegetation.

The Norwegian government has access to recent satellite chlorophyll fluorescence measurements (62 bands/wavelengths) of vegetation corresponding to a region on the Earth's surface[5]. There are 135 samples. Each sample corresponds to a pixel (hyperspectral) in an optical satellite image. Experts have provided the ground truth chlorophyll content for these 135 samples (corresponding to physical locations on the Earth). This is shown in Fig. 6.

The chlorophyll data is available in

```
Xchl_tr.mat          ychl_tr.mat
```

Problem 2: Chlorophyll Estimation by Regression

Even though regression hasn't been a primary focus in our course, regression (function estimation) can be performed by several of the methods we have looked at with either exactly the same procedure, or by small modifications, and is very important in pattern recognition.

(*) The Norwegian government is very interested in finding the best way to estimate chlorophyll from space. You must help by performing some analysis. Base the analysis on a so-called leave-one-out procedure. This means that you loop through the data set 135 times, leave out one data point at the

[4] I.e. mean and standard deviation of the accuracy.

[5] This is a slightly modified version (to not violate intellectual properties) of real data J. Verrelst, L. Alonso, J. Caicedo, J. Moreno and G. Camps-Valls, "Gaussian Process Retrieval of Chlorophyll Content from Imaging Spectroscopy Data," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 6(2), 867-874, 2013

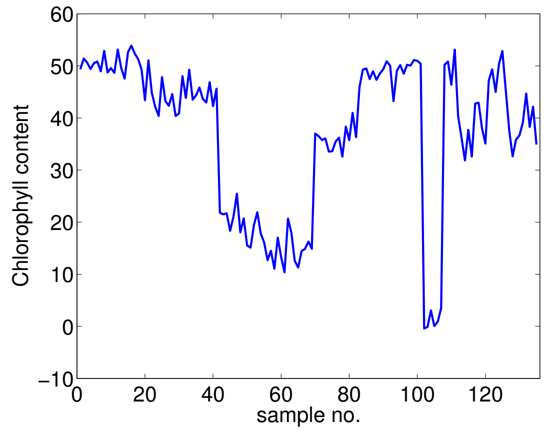

Figure 6: Plot of ground truth chlorophyll content.

time and train on the remaining 134 points. Then test on the left out point. Visualize results and/or provide some numerical measure of how well the methods are working.

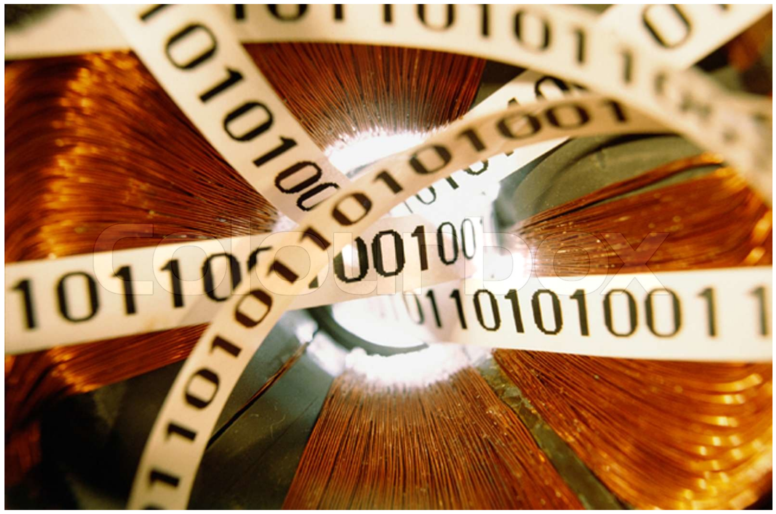

Figure 7: We will develop a bit-based communication system using handwritten digit recognition machines (figure from colourbox.com).

Binary Digit Recognition

The start of deep learning as we know it today and the use of convolutional neural networks can to a large degree be traced back to recognition of handwritten letters. Since letters (capital and lower-case), spaces, punctuation marks and so on define many different classes, and since people write so differently, this is extremely difficult (imagine also chinese letter recognition, with thousands of classes).

Digit recognition is a related ten-class problem, for example for automated reading of checks[6].

If we only need to distinguish between two digits, then we have a binary problem which is easier since it is a two-class problem. The Norwegian government has a great idea! All people should write in ASCII code (see table in Appendix A). This means that each character is represented by a 7-bit code. Hence, if a person writes for example: 1000001 by hand, this is the letter "A". Especially in our cold Arctic climate, the government thinks this may simplify life, since it is easier to just have to write ones and zeros with gloves on if your hands and fingers are cold, AND, the recognition system only needs to recognize two different handwritten symbols. The system will therefore be *binary*, with only two classes "1" and "0".

Note that all digital communication systems are binary, however bits are

[6]LeCun, Yann; Lon Bottou; Yoshua Bengio; Patrick Haffner (1998). "Gradient-based learning applied to document recognition". Proceedings of the IEEE. 86 (11): 2278?2324. doi:10.1109/5.726791

represented in different ways. All such systems will in any case need classifiers at the receiving end, just like our system in this problem. The binary digit recognition system we will study now is based on raw pixels as inputs and not convolutional networks.

We have available two different training sets. One is a set of 10 (16×16) handwritten ones and zeros in

`Xtr_digits.mat` `ytr_digits.mat`

Each digit (image) is represented as a 256-dimensional vector. See Appendix C for more information on this real data set, obtained from the US postal service. A plot of the 10 training digits at our disposal is shown in Fig. 8.

The other training set is larger, and can be found in

`Xtr_digits_larger.mat` `ytr_digits_larger.mat`

Figure 8: Plot of the training set for the "1" and "0" digits.

In Fronter, there is a Matlab-function

`ascii.m`

which takes as input a vector of labels $\in \{-1, 1\}$, separated by spaces, and outputs the *text* corresponding to the labels using the ASCII code. Only lower case letters are currently implemented, in addition to "!" and "." If a 7-bit string doesn't correspond to any symbol "a-z" or "!" and ".", the symbol "#" is output. The message you will try to decode in order to test our binary handwritten digit recognition system, is available in

`Xte_digits_2017.mat`

Problem 3: Binary digits in bits-based communication

(*) Design binary digit recognition systems and run `ascii.m` on classification results. You may need to find good parameters by experimenting a little bit, or by other means. You are free to use all methods you would like to use, and to test out data transformation and dimensionality reductions methods. Enlighten the world on any differences etc. with respect to the two different training sets.

Appendix A: Data Format

It seems that for some people Matlab data stored in Fronter is downloaded in the format struct. For example, if you use the command "whos" and you see something like this:

```
>> whos Xtest
  Name       Size              Bytes  Class     Attributes

  Xtest      1x1             1935536  struct
```

then "Xtest" is a struct variable. You can check the content of the variable just by typing it's name:

```
>> Xtest

Xtest =

    Xtest: [256x945 double]
```

So here, the struct variable "Xtest" contains the data in double format, and you need to write in this particular case "Xtest.Xtest" in the Matlab command window to access the actual data. Please ask if you experience problems, or if you find find this confusing.

Appendix B: ASCII Table

b_7 → b_6 → b_5 →					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Bits	b_4 ↓	b_3 ↓	b_2 ↓	b_1 ↓	Column → Row ↓	0	1	2	3	4	5	6	7
	0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
	0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
	0	0	1	0	2	STX	DC2	"	2	B	R	b	r
	0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
	0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
	0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
	0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
	0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
	1	0	0	0	8	BS	CAN	(	8	H	X	h	x
	1	0	0	1	9	HT	EM	)	9	I	Y	i	y
	1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
	1	0	1	1	11	VT	ESC	+	;	K	[	k	{
	1	1	0	0	12	FF	FC	,	<	L	\	l	|
	1	1	0	1	13	CR	GS	-	=	M	]	m	}
	1	1	1	0	14	SO	RS	.	>	N	^	n	~
	1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Figure 9: ASCII Table. The letter "A", for example, has a 7-bit code 1000001.

Appendix C: About the Digits Data

Taken from http://www-stat.stanford.edu/~tibs/ElemStatLearn/

```
Normalized handwritten digits, automatically
scanned from envelopes by the U.S. Postal Service. The original
scanned digits are binary and of different sizes and orientations; the
images  here have been deslanted and size normalized, resulting
in 16 x 16 grayscale images (Le Cun et al., 1990).
```