Cognition and computation:

- Francesco Gabriele
- student identification number: 2073224

NOTE: All the following refers to the script "Script of the assignment", except for point 7, which instead refers to the script "EMNIST-Letters restricted to ten classes".

1. INTRODUCTION

1.1 GOALS OF PAPER

The goal of the project is to analyze the efficiency of a Deep Belief Network and Feed-Forward Neural Network. Further examination on their resilience to noise in the data and to adversarial attacks will be also carried out.

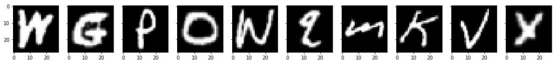
1.2 OVERVIEW OF THE DATASET

The EMNIST-letters consists of a set of images of capital and non-capital letters, made up of 28X28 pixels each.

The dataset is splitted into two subset:

- Training set: 124800 images
- Test set: 20800 images

When displayed at the screen they look like this:



1.3 PREPROCESSING:

Each single pixel value falls within [0,255] range where 0 represents "black" and 255 reprepresents "white". Since the model under examination is a DBN these values will be scaled down to the range [0,1].

2. UNSUPERVISED LEARNING:

2.1 BRIEF DESCRIPTION OF THE MODEL

Deep Belief Networks also referred to with the acronym D.B.N. are generative models and as such they build representations of the data they get as input in an unsupervised learning way.

A DBN is constituted by a stack of Restricted Boltzmann Machines (R.B.M.) which are stochastic recurrent networks; more specifically they are probabilistic graphical models where each neuron activation level represents a probability. The term "restricted" refers to the absence of intralayer connections between neurons.

Each RBM is trained aiming to maximize the likelihood function via an algorithm known as "Contrastive Divergence", consequently the whole DBN is trained in a greedy way.

The representations made by the model are used to classify the original images. This classification is carried out with a linear classifier that encodes the categories trained with supervised learning.

The linear classifier that is fully connected to the deepest hidden layer investigates the task performance whereas read-outs at previous hidden layers investigate how well each of them makes representations. An increasing accuracy across the linear read-outs indicates that the representations get progressively more disentangled.

Ideally, the intention would be to structure a DBN that makes representation of the data it has been fed on in an hierarchical fashion. Therefore the representations outputted by the first RBM are expected to be higher level than the ones produced by the last one.

To facilitate this, the DBN will be structured having an increasing number of neurons at each hidden layer of the RBMs.

2.2 SETTING UP THE PARAMETERS

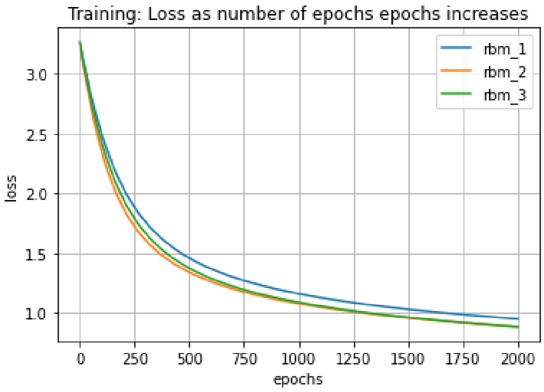
Provided that the DBN under analysis consist of a stack of three RBMs the main parameters to focus on are the:

- the number of neurons of the hidden layer of each RBM: having a network with too many neurons could require a while for the training to be completed and it could also lead to overfitting. Therefore, a reasonable number of neurons is to be chosen.

- the number of epochs the DBN is trained over
- the batch size: larger batch size may speed up the training but usually leads to worse model performance if compared to smaller ones.
- learning rate: if it is too small the training phase is faster but there is the risk of overshooting the minima, if it is too small instead the training is too slow.

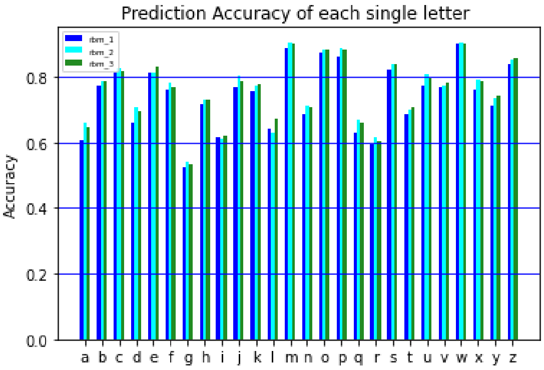
The performance of the model highly depends on the structure of the network, thus some reasoning about the matter is required. It can be proved that the same DBN structured with 400 neurons in the hidden layer of the first RBM,500 in the one of the second RBM and 800 neurons in the one of the third RBM reaches a good level of accuracy over the MNIST dataset.
The MNIST dataset, though, consists of ten classes, one for each digit whereas the EMNIST- letters one has more of the double of the classes, therefore a bigger number of neurons might be required to reach a similar performance.

number of neurons of the first hidden layer	number of neurons of the first hidden layer	number of neurons of the first hidden layer	batch size	epochs DBN	epochs linear read-outs	Accuracy of the first linear classifier	Accuracy of the second linear classifier	Accuracy of the third linear classifier
400	600	900	50	50	2000	74%	75%	76%



3 FURTHER INVESTIGATION OF THE MODEL

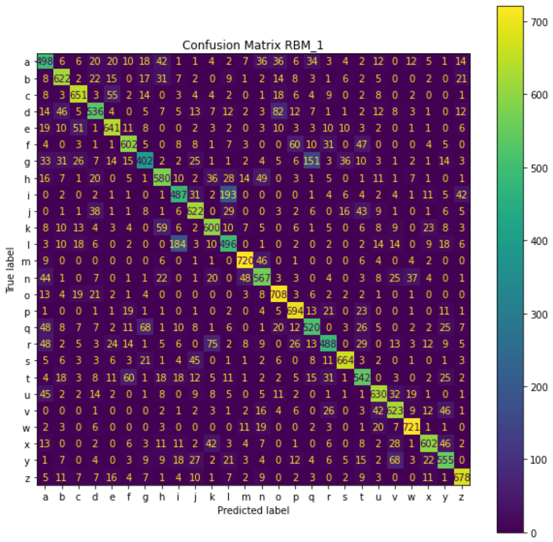
3.1 PREDICTION OF EACH LETTER

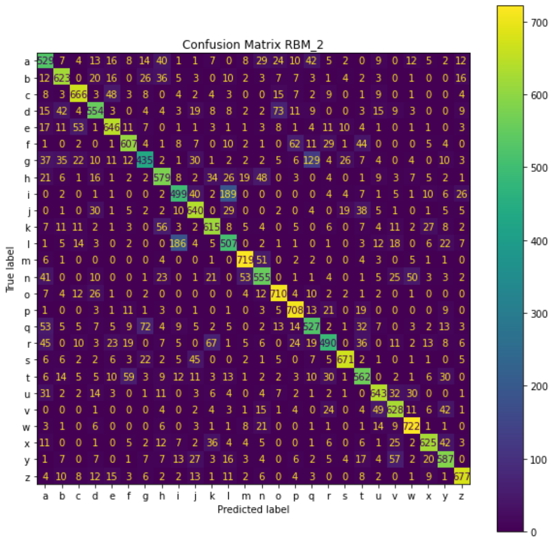


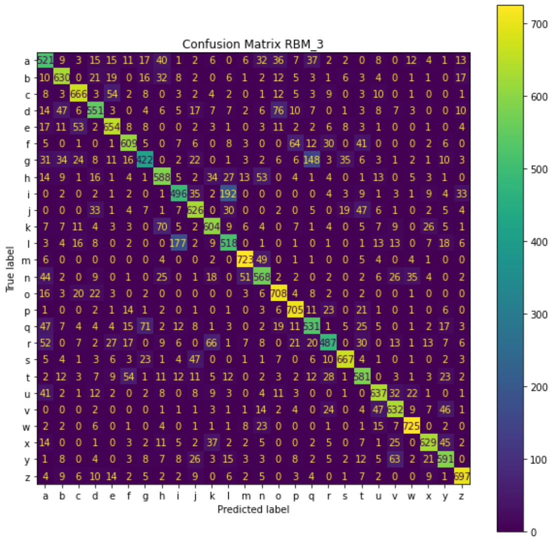
As it can be clearly seen from the bar chart above, the letter which the model struggles the most with is the “g”, while the one with it deals the best is the “w”.

3.2 CONFUSION MATRICES

An extensive description of the predictions made at test time can be represented in the form of confusion matrices.





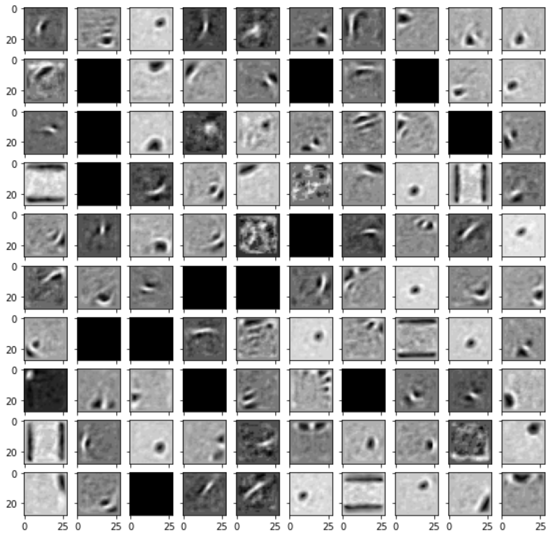


The matrices show that “i” and “l” are often mistaken for one another; the same thing happens for the letters “q” and “g”.

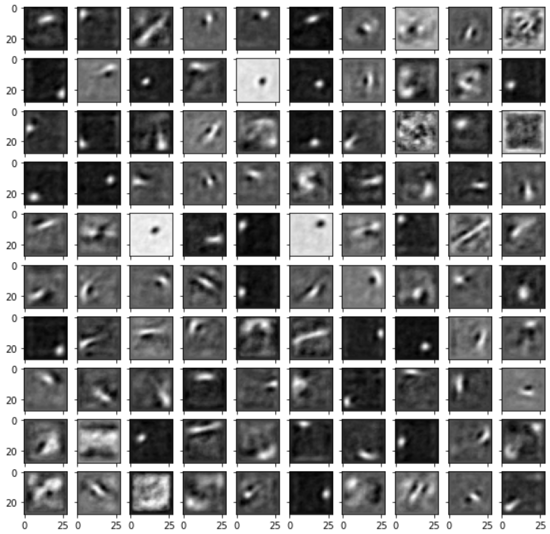
3.3 VISUALIZING RECEPTIVE FIELDS:

After the model has been trained it is possible to visualize the learned weights. In the images below some of the weights are plotted in cells of 28x28 pixels so that it will be possible to identify which parts of an image trigger the unit of a specific vector.

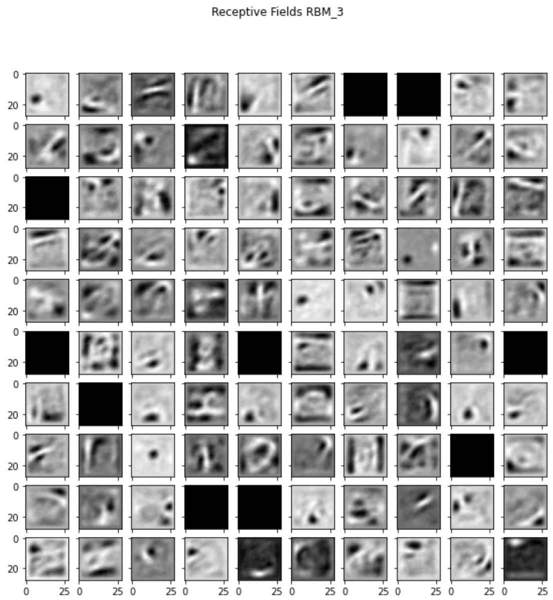
Receptive Fields RBM_1



Receptive Fields RBM_2



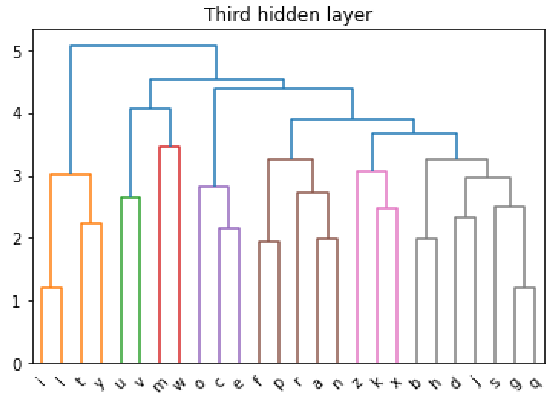
DISCUSSION:



In full accordance with what was mentioned before, the representations become better and better.

3.4 HIERARCHICAL CLUSTERING

Internal representation properties are explorable through plotting their dendrogram.



DISCUSSION:

The dendrogram above shows how similar the letters are to each other according to the DBN. The results seem to be pretty reasonable. For example the letter "o" differs from the "c" just for a small arch that would complete a "c" into an "o" that's why they are close in the dendrogram.

4. ROBUSTNESS TO NOISE

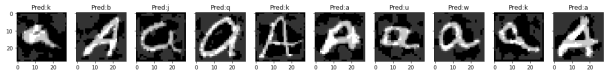
4.1 RANDOM GAUSSIAN NOISE

Let's now inject some noise in the images and then feed the models with them to examine how resilient the model is to noisy stimuli.

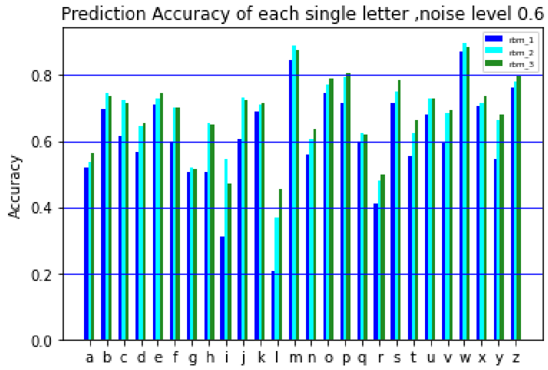
A tensor whose items come from a normal distribution of mean 0 and variance 1 is created, then it is multiplied by the noise level (ε) and finally the perturbed dataset is obtained by adding that tensor to the original data one.

4.2 IMPACT OF RANDOM NOISE ON THE IMAGES

In the following a string of letters affected by random gaussian noise (ε = 0.2) with the relative predictions of the DBN is displayed:



4.3 PREDICTION OF EACH LETTER

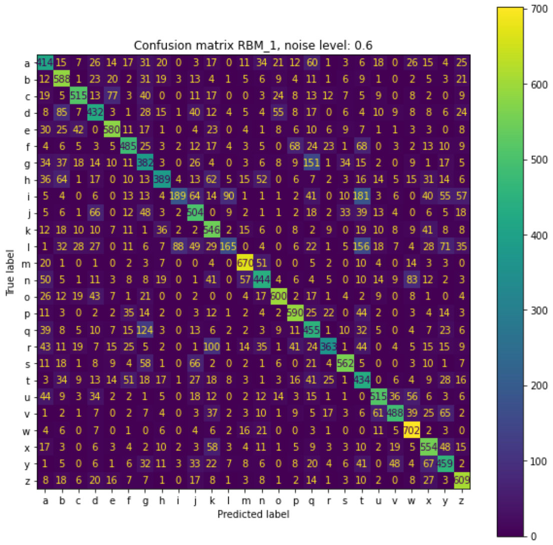


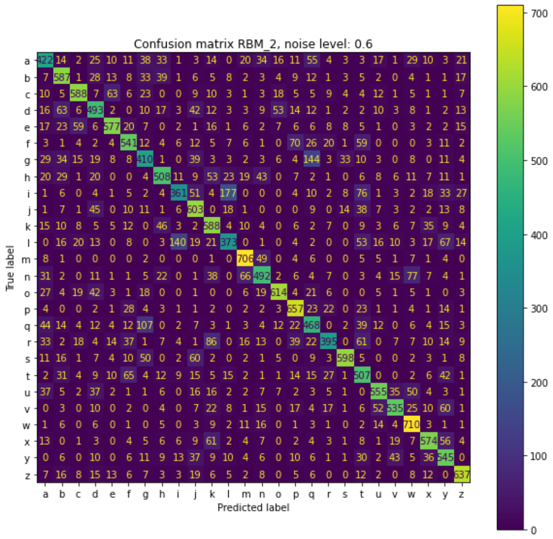
DISCUSSION

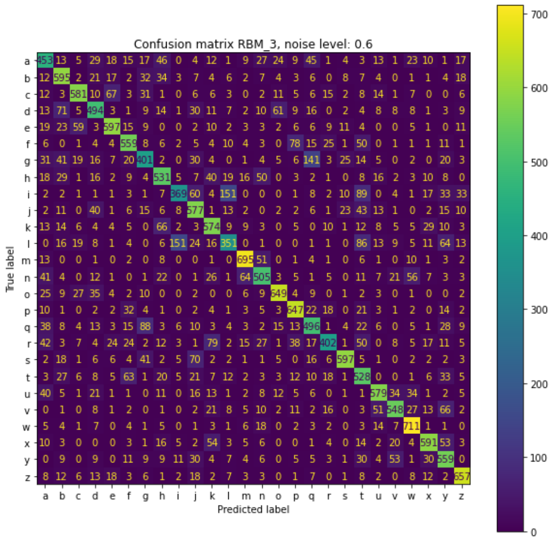
After the injection of the noise in the data the model really struggles with the letter "l" whereas "m" and "w" appear to be the ones it predicts more accurately, this might be related to the fact that they are closed in dendrogram.

In addition to that the bar chart highlights really well how the predictions of the first classifier are definitely more imprecise compared to the ones of the the remaining two classifiers, which instead are quite close in terms of right predictions.

4.4 CONFUSION MATRICES





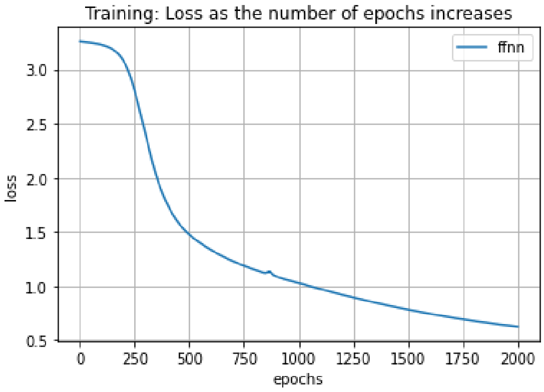


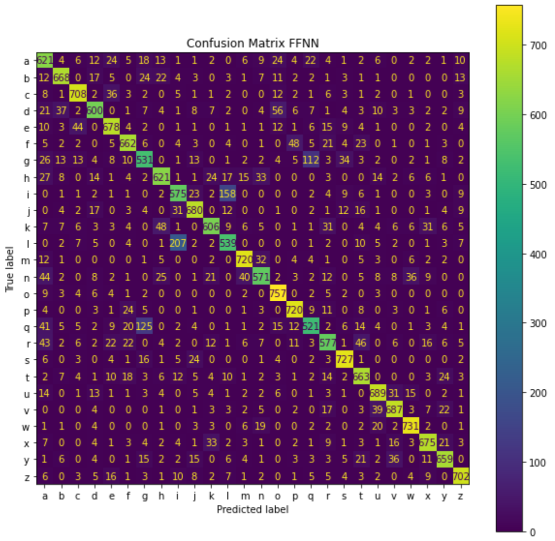
5. COMPARISON WITH A FEED-FORWARD NEURAL NETWORK

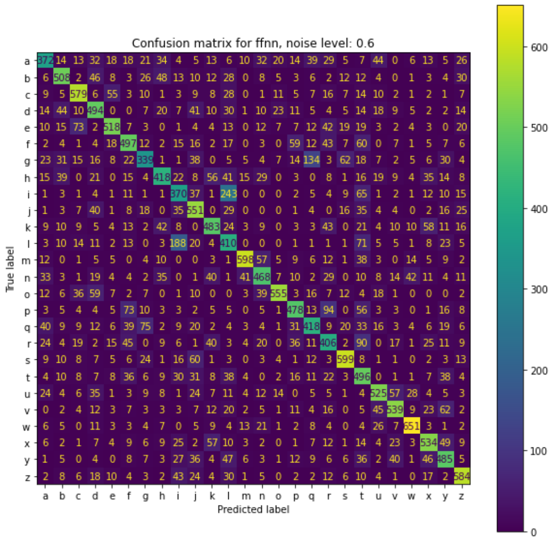
5.1 TRAINING OF FEED FORWARD NEURAL NETWORK

A feed-forward neural network is trained on the same dataset. To make a fair comparison with the DBN, the FFNN is trained with the same amount of epochs and the same size of hidden neurons. Coincely:

FEED FORWARD MODEL				
first hidden layer size	second hidden layer size	third hidden layer size	epochs	accuracy
400	600	900	2000	81%



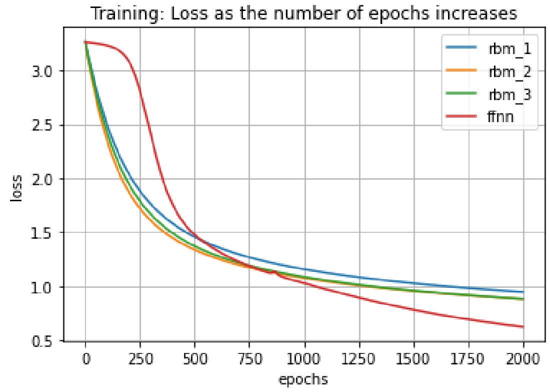




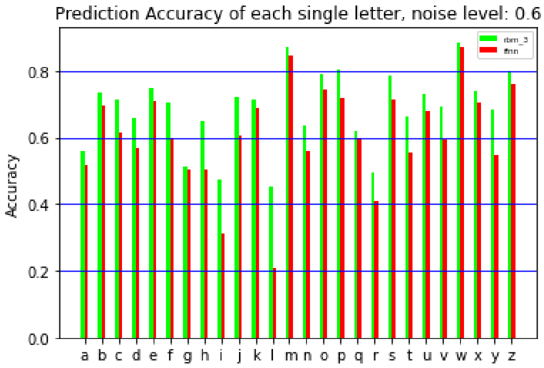
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	372	14	13	32	18	18	21	34	4	5	13	6	10	32	20	14	39	29	5	7	44	0	6	13	5	26
b	6	508	2	46	8	3	26	48	13	10	12	28	0	8	5	3	6	2	12	12	4	0	1	3	4	30
c	9	5	579	6	55	3	10	1	3	9	8	28	0	1	11	5	7	16	7	14	10	2	1	2	1	7
d	14	44	10	494	0	0	7	20	7	41	10	30	1	10	23	11	5	4	5	14	18	9	5	2	2	14
e	10	15	73	2	518	7	3	0	1	4	4	13	0	12	7	7	12	42	19	19	3	2	4	3	0	20
f	2	4	1	4	18	497	12	2	15	16	2	17	0	3	0	59	12	43	7	60	0	7	1	5	7	6
g	23	31	15	16	8	22	339	1	1	38	0	5	5	4	7	14	134	3	62	18	7	2	5	6	30	4
h	15	39	0	21	0	15	4	418	22	8	56	41	15	29	0	3	0	8	1	16	19	9	4	35	14	8
i	1	3	1	4	1	11	1	1	370	37	1	243	0	0	0	2	5	4	9	65	1	2	1	12	10	15
j	1	3	7	40	1	8	18	0	35	551	0	29	0	0	0	1	4	0	16	35	4	4	0	2	16	25
k	9	10	9	5	4	13	2	42	8	0	483	24	3	9	0	3	3	43	0	21	4	10	10	58	11	16
l	3	10	14	11	2	13	0	3	188	20	4	410	0	0	0	1	1	1	1	71	5	5	1	8	23	5
m	12	0	1	5	5	0	4	10	0	0	3	1	598	57	5	9	6	12	1	38	3	0	14	5	9	2
n	33	3	1	19	4	4	2	35	0	1	40	1	41	468	7	10	2	29	0	10	8	14	42	11	4	11
o	12	6	36	59	7	2	7	0	1	10	0	0	3	39	555	3	16	7	12	4	18	1	0	0	0	2
p	3	5	4	4	5	73	10	3	3	2	5	5	0	5	1	478	13	94	0	56	3	3	0	1	16	8
q	40	9	9	12	6	39	75	2	9	20	2	4	3	4	1	31	418	9	20	33	16	3	4	6	19	6
r	24	4	19	2	15	45	0	9	6	1	40	3	4	20	0	36	11	406	2	90	0	17	1	25	11	9
s	9	10	8	7	5	6	24	1	16	60	1	3	0	3	4	1	12	3	599	8	1	1	0	2	3	13
t	4	10	8	7	8	36	6	9	30	31	8	38	4	0	2	16	11	22	3	496	0	1	1	7	38	4
u	24	4	6	35	1	3	9	8	1	24	7	11	4	12	14	0	5	5	1	4	525	57	28	4	5	3
v	0	2	4	12	3	7	3	3	3	7	12	20	2	5	1	11	4	16	0	5	45	539	9	23	62	2
w	6	5	0	11	3	3	4	7	0	5	9	4	13	21	1	2	8	4	0	4	26	7	651	3	1	2
x	6	2	1	7	4	9	6	9	25	2	57	10	3	2	0	1	7	12	1	14	4	23	3	534	49	9
y	1	5	0	4	0	8	7	3	27	36	4	47	6	3	1	12	9	6	6	36	2	40	1	46	485	5
z	2	8	6	18	10	4	3	2	43	24	4	30	1	5	0	2	2	12	6	10	4	1	0	17	2	584

5.2 COMPARISON BETWEEN DEEPEST DBN READ OUT WITH THE FFNN

A five 5% gap in accuracy separates the FFNN and the DBN, but let's further investigate it.

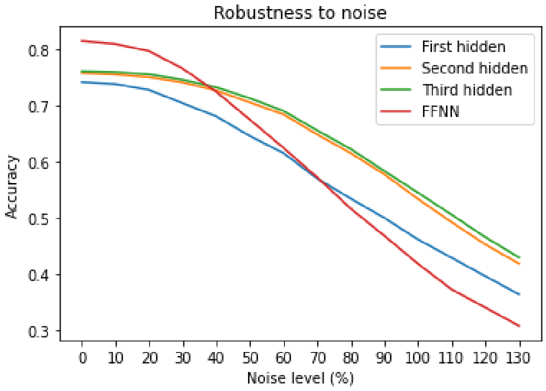


The loss relative to the FFNN stalls for the first two hundred epochs and then decreases abruptly until the five hundredth epoch is reached, after that the loss decreases more gradually. The loss curves relative to the DBN instead, go down more smoothly. However around the thousandth epoch, these loss curves start to diverge from the red one.



With a noise random gaussian with ε = 0.6 is injected in the model the DBN perform better than the FFNN, let's make the comparison on a wide range of values of ε
.

5.3 COMPARISON BETWEEN EACH RBN AND THE FFNN: PSYCHOMETRIC CURVE



The accuracy of the second and third RBMs remains almost the same for values of ε smaller than 0.4, that is also the value of epsilon around which the curve of the FFNN goes below the curves of the last two RBMs. For higher levels of noise the FFNN performance is even worse than the first RBM.

6. ADVERSARIAL ATTACKS

6.1 DEFINITION OF ADVERSARIAL ATTACK

The noise added to the images in adversarial attacks is specifically crafted to confuse the model.

Random gaussian noise is random by definition so it by no means depends on the gradient of the cost function. Conversely adversarial attacks aim to inject noise in the opposite direction of the descent of the gradient.

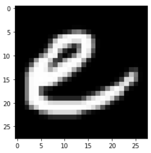
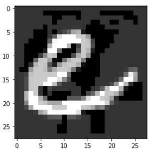
The attack is implemented according to the following formula, where “x” represents the data tensor.

$$x = x + \varepsilon\, sign(\nabla x(w, x, y))$$

Adversarial attacks are meant to pass undetected so usually ε is not very large.

6.2 VISUALIZE THE IMPACT OF ADVERSARIAL ATTACKS ON THE IMAGES

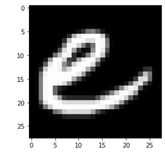
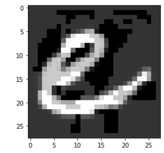
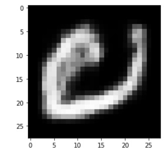
In the following table an image of the letter “e” is compared to its perturbed and reconstructed versions.

original image	perturbed image
	

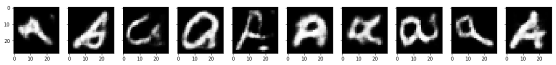
By taking a close look at the image in the middle it is possible to see some gray in the image due to the noise.

6.3 TOP DOWN RECONSTRUCTION AS A WAY TO MITIGATE THE EFFECT OF A.A.

Image reconstruction is a valid fix to adversarial attacks. In the following a top-down reconstruction will be performed.

original image	perturbed image	1 step reconstruction
		

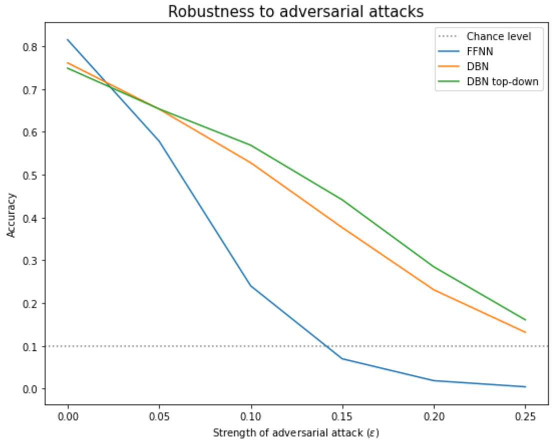
The gray in the third image is almost removed altogether even if there are still some blurry areas of the image, but the reconstruction is pretty good. Let’s how good is the reconstruction on the the letter “a”:



Apart from the first image the model appears to be quite good at denoising the data, in fact the reconstructions are fairly similar to the original images.

6.4 PSYCHOMETRICS CURVES

Let’s now explore the robustness of each model to different attack strengths.



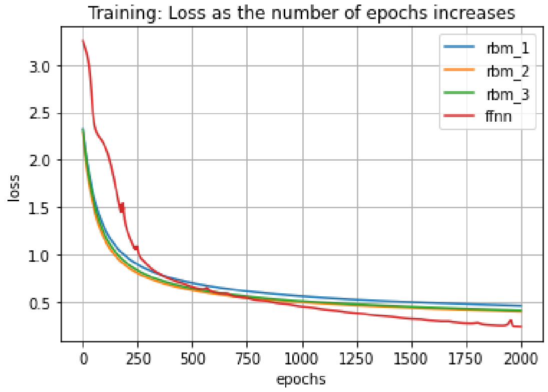
As the reconstructed images shown above suggest, the model denoise the images very well and this results in higher accuracy and more resilience to adversarial attacks.

Not only the DBN with top-down reconstruction outperforms the FFNN but it also outdoes the DBN without reconstructions.

7. ONE LAST NOTE ABOUT THE ACCURACY

It can be pointed out that the accuracy of the two models on the EMNIST-letters is not that great especially if compared to the model performance on MNIST dataset where the accuracy is found to be well above the 90%.
To make a fair comparison let's take into consideration a subset of the classes of the EMNIST- letters dataset. More specifically, let's restrict the classes to the first ten letters of the alphabet and see what happens. All the other parameters are left unaltered.

EMNIST-letters classes restricted to the first ten letters of the alphabet	
First read-out accuracy	86%
Second read-out accuracy	87%
Third read-out accuracy	88%
ffnn read out	91%



As expected the levels of accuracy of the model working on a smaller number of classes are comparable to the one recorded with MNIST dataset.
Therefore the low level of accuracy seems to be related to the number of classes.

REFERENCE PAPERS

- Training restricted Boltzmann machines: An introduction - ScienceDirect
- Frontiers | Deep Unsupervised Learning on a Desktop PC: A Primer for Cognitive Scientists (frontiersin.org)
- Reducing the Dimensionality of Data with Neural Networks | Science
- guideTR.pdf (toronto.edu)