Klasifikacija slika kućnih brojeva dubokim konvolucijskim modelima

Ivan Šego

4. srpnja 2018.

Sadržaj

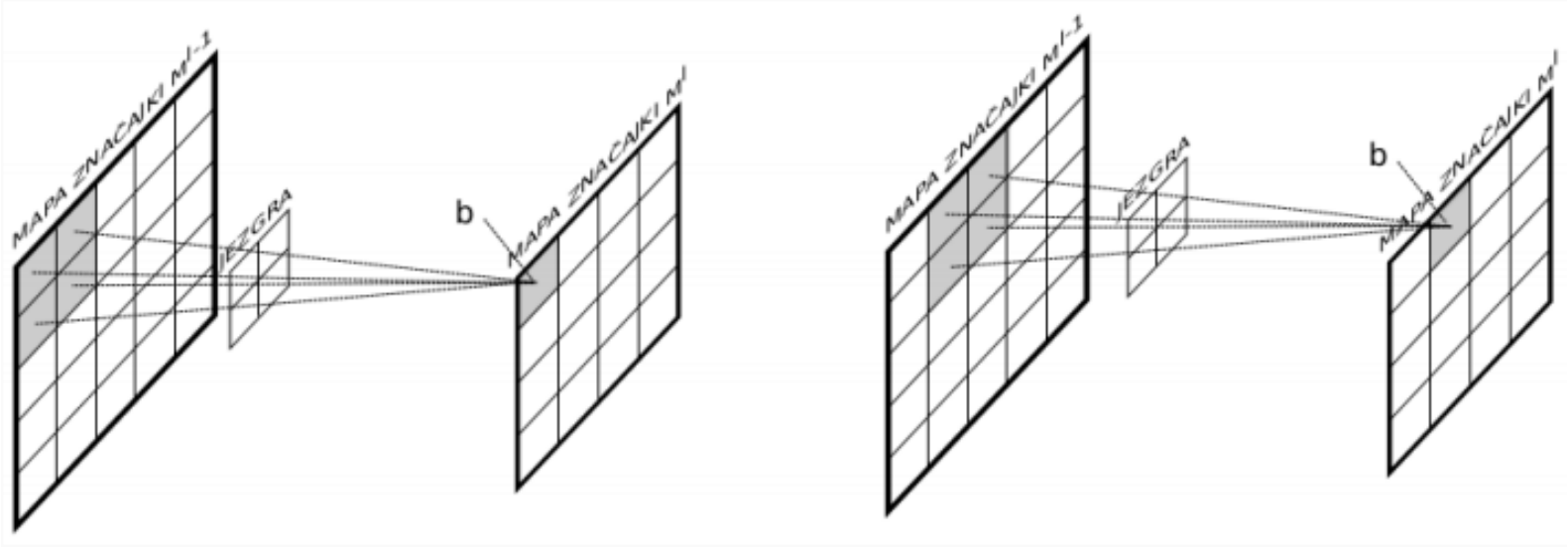
Duboko učenje

- Klasifikacija
- ImageNet 2012.
- Hipoteza o dubini
- Razvoj GPU

Sadržaj

Konvolucijski sloj

- Filteri (engl. kernel) koji uče težine
- Određene prostorne značajke

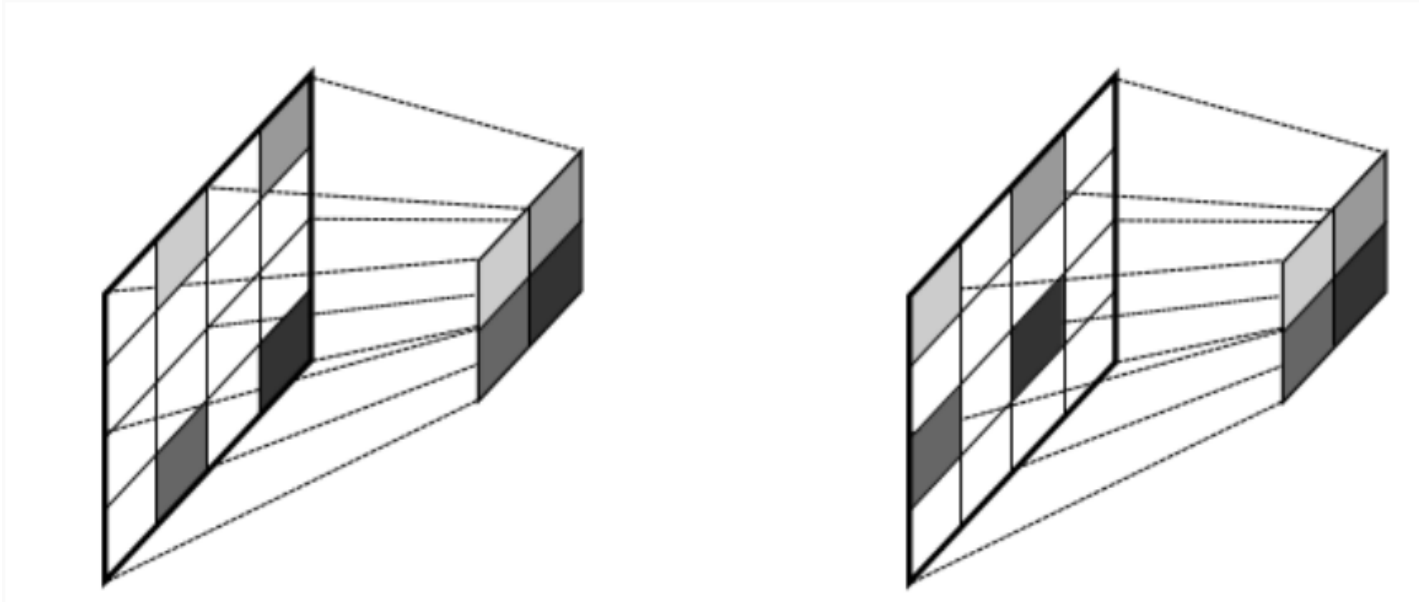


Slika: Dva koraka konvolucijskog sloja

Sadržaj

Sloj sažimanja

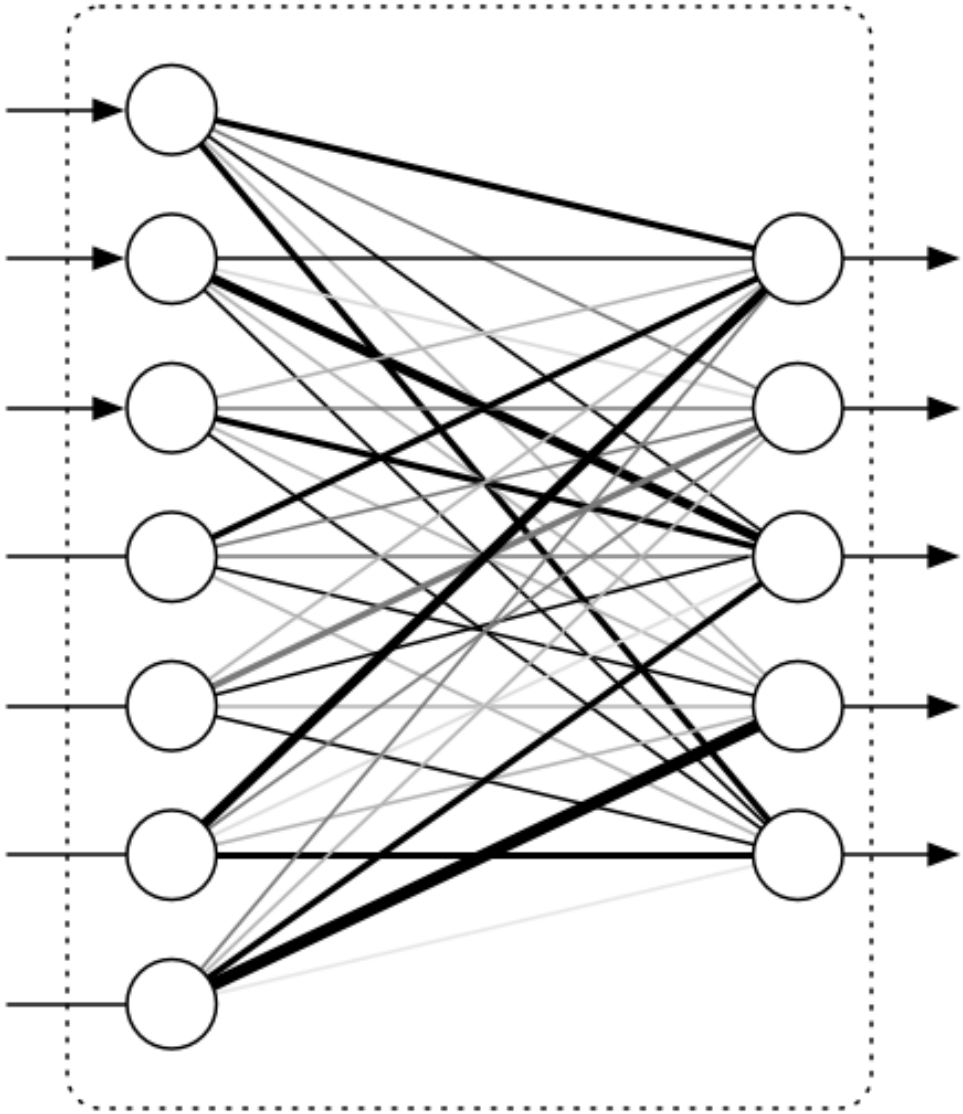
- Smanjivanje dimenzionalnosti
- Mapiranje prostorno bliskih značajki (max, aritmetička sredina)



Slika: Sloj sažimanja

Sadržaj

Potpuno povezani sloj

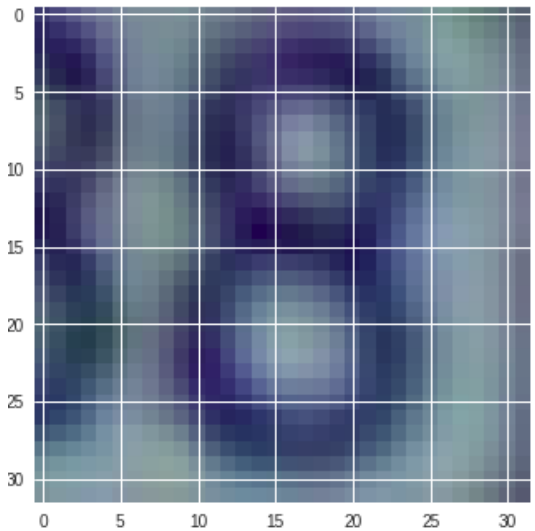


Slika: Potpuno povezani sloj

Sadržaj

Općenito

- The Street View House Numbers (SVHN) Dataset
- Podrezani skup: slika 32×32 piksela, RGB
- Trening skup: 73257 znamenaka, 80% učenje, 20% validacija
- Test skup: 26032 znamenaka
- Učestalost u skupu za treniranje redom brojeva od 0-9: 4433, 12443, 9519, 7634, 6650, 6195, 5215, 5046, 4582, 4215

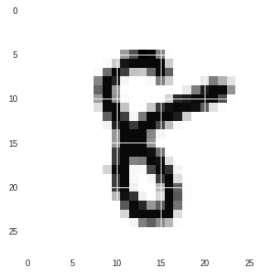


Slika: Kućni broj 8

Sadržaj

Općenito

- The Modified National Institute of Standards and Technology database
- Originalni skup: slika 28×28 piksela, crno - bijela
- Trening skup: 60000 znamenaka, 80% učenje, 20% validacija
- Test skup: 10000 znamenaka
- Distribucija znamenaka u skupu za treniranje jednolika



Slika: Ručno napisan broj 8

Sadržaj

Proces treniranja

- Aktivacijska funkcija: zglobnica (engl. ReLU)
- Gradijentni spust: Adam
- Funkcija gubitka: unakrsna entropija
- Stopa učenja: 0.001
- Regularizacija: normalizacija grupe (engl. batch normalization)
- Broj epoha: 100
- Prosječno trajanje po epohi: 3.6 min
- Programski okvir: PyTorch 0.4.0

Sadržaj

Arhitektura

1. ulazni sloj - slika $32 \times 32 \times 3$
2. Konvolucijski sloj - 32 filtra 5×5, batch-norm, ReLU, nadopunjavanje 4.
3. Sloj sažimanja - filter 2×2.
4. Konvolucijski sloj - 64 filtra 3×3, batch-norm, ReLU, nadopunjavanje 2.
5. Sloj sažimanja - filter 2×2.
6. Konvolucijski sloj - 128 filtra 3×3, batch-norm, ReLU, nadopunjavanje 2
7. Sloj sažimanja - filter 2×2.
8. Potpuno povezani sloj - 4608 neurona ulaz, izlaz 256.
9. Potpuno povezani sloj - 256 ulaz, 128 izlaz
10. Potpuno povezani sloj - 128 ulaz, 10 izlaz

Sadržaj

Točnost

- Najveća točnost na validacijskom skupu: 92.67%, epoha:38
- Točnost na skupu za testiranje 91.80%

Slika: Prikaz točnosti kroz epohe na skupu za treniranje i validacijskom skupu

State of the art

Result	Method	Venue	Details
1.69%	Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree	AISTATS 2016	Details
1.76%	Competitive Multi-scale Convolution	arXiv 2015	
1.77%	Recurrent Convolutional Neural Network for Object Recognition	CVPR 2015	Details
1.81%	Batch-normalized Maxout Network in Network	arXiv 2015	Details
1.92%	Deeply-Supervised Nets	arXiv 2014	
1.92%	Multi-Loss Regularized Deep Neural Network	CSVT 2015	Details
1.94%	Regularization of Neural Networks using DropConnect	ICML 2013	
1.97%	On the Importance of Normalisation Layers in Deep Learning with Piecewise Linear Activation Units	arXiv 2015	
2%	Estimated human performance	NIPS 2011	Details
2.15%	BinaryConnect: Training Deep Neural Networks with binary weights during propagations	NIPS 2015	
2.16%	Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks	ICLR 2014	Details
2.35%	Network in Network	ICLR 2014	Details

Slika:

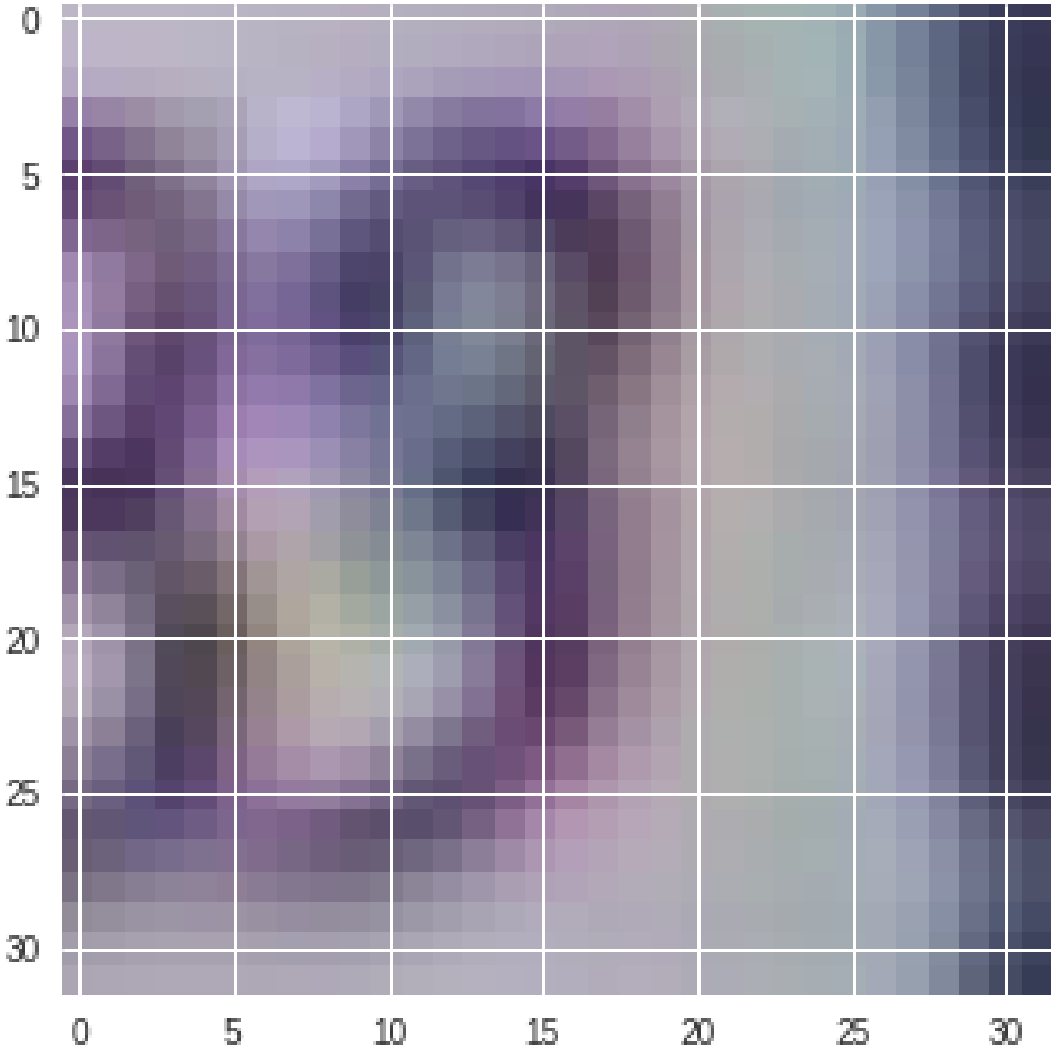
http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.h

Matrica zabune

Matrica zabune										
Klase	0	1	2	3	4	5	6	7	8	9
0	**93.12**	0.69	0.46	0.34	0.40	0.17	1.55	0.52	0.46	2.29
1	0.92	**93.78**	0.88	0.55	1.82	0.10	0.31	1.22	0.22	0.20
2	0.34	0.72	**94.38**	1.04	0.89	0.31	0.39	0.94	0.14	0.84
3	0.42	2.08	1.49	**86.09**	0.45	3.71	0.59	0.49	0.87	3.82
4	0.55	2.26	0.87	0.4	**93.74**	0.36	0.52	0.36	0.28	0.67
5	0.17	0.5	0.42	1.93	0.63	**91.99**	2.64	0.17	0.34	1.22
6	1.77	0.71	0.1	0.76	0.61	2.23	**91.1**	0.35	1.57	0.81
7	0.25	4.26	1.93	0.74	0.25	0.15	0.1	**91.58**	0.0	0.74
8	1.14	0.96	0.48	1.45	0.6	1.45	5.48	0.12	**84.46**	3.86
9	1.76	0.56	1.5	0.44	0.5	0.63	0.63	0.25	0.69	**93.04**

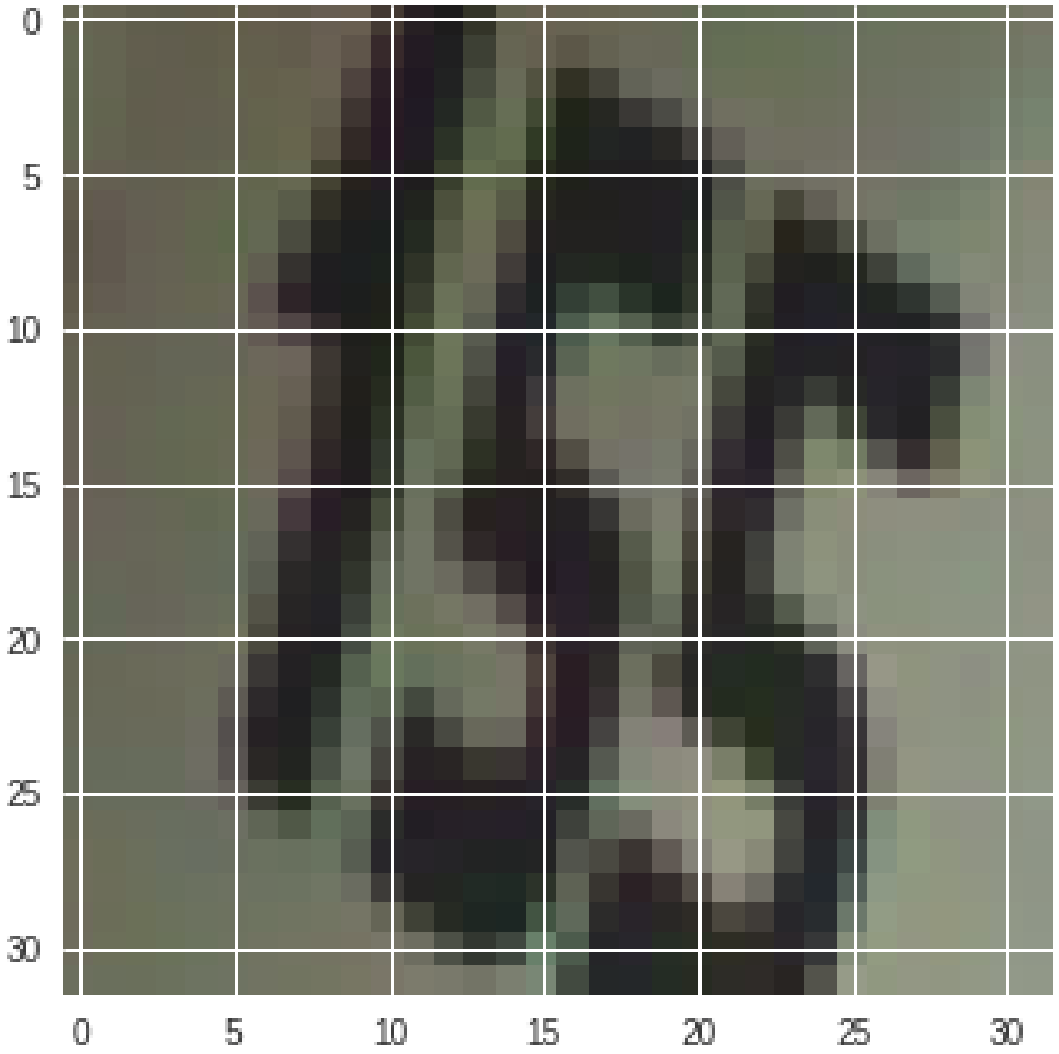
Slika: Prikaz matrice zabune u postotcima

Netočno klasificirane slike(1)



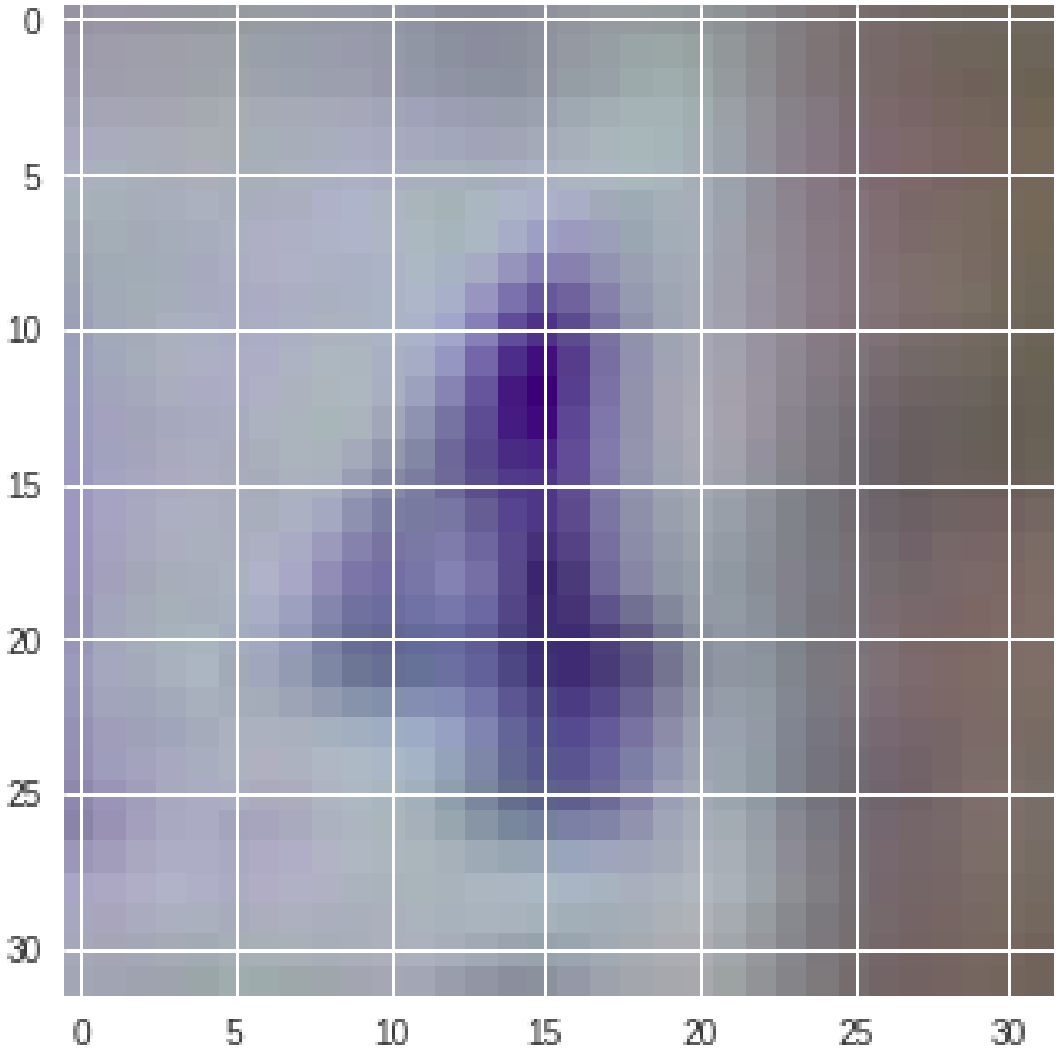
Slika: 9 s vjerojatnosti 1.0, trebao 3

Netočno klasificirane slike(2)



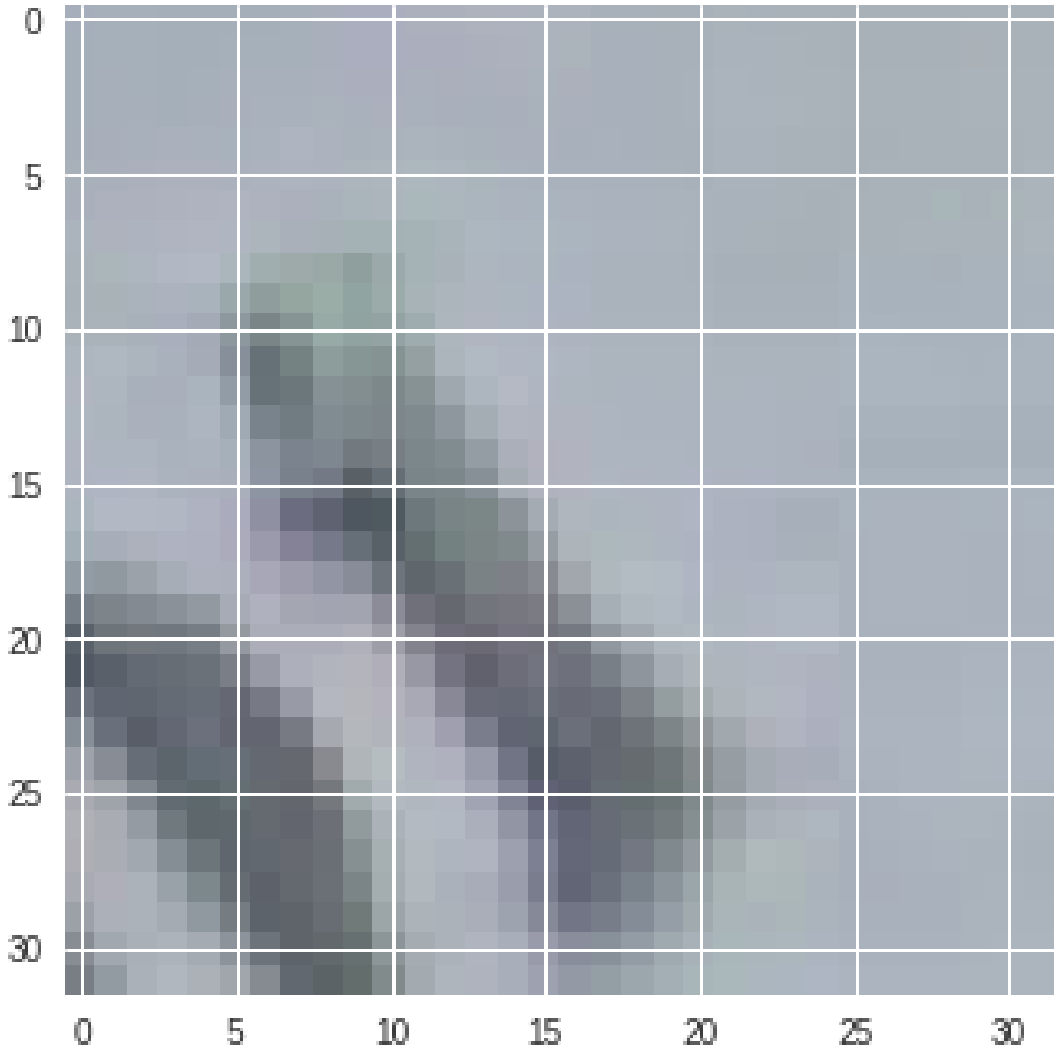
Slika: 3 s vjerojatnosti 1.0, trebao 5

Netočno klasificirane slike(3)



Slika: 3 s vjerojatnosti 0.98, trebao 4

Netočno klasificirane slike(4)

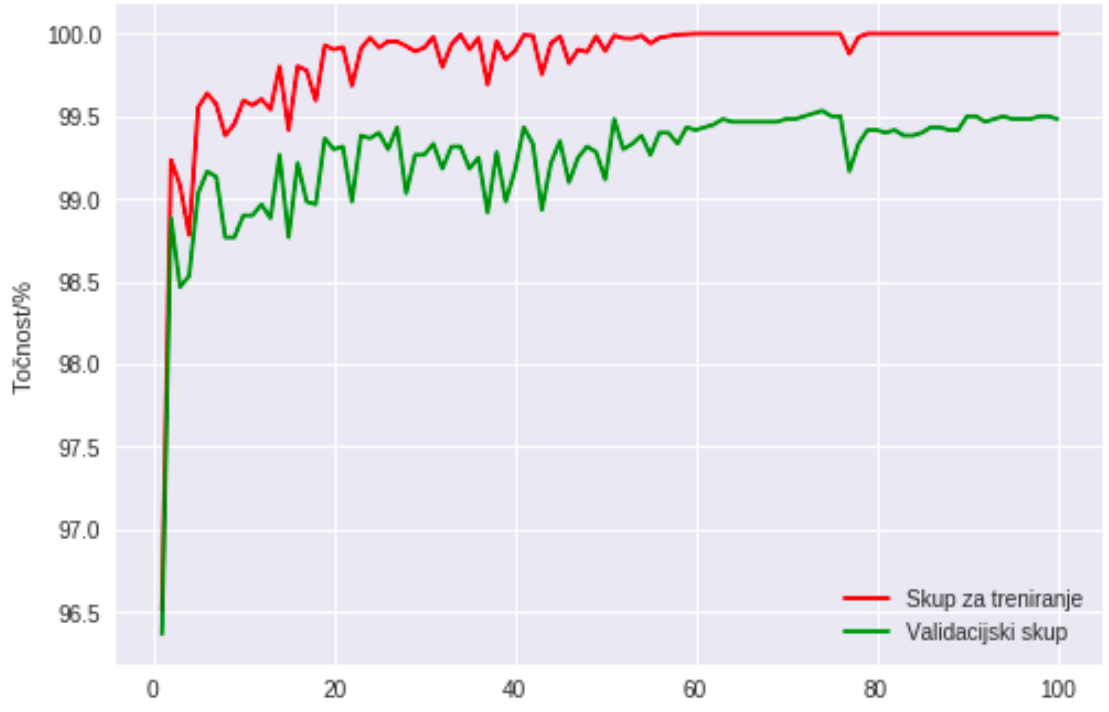


Slika: 0 s vjerojatnosti 0.48, trebao 1

Sadržaj

Treniranje

- Najveća točnost na validacijskom skupu: 99.53%, epoha:74
- Točnost na skupu za testiranje 99.52%



Slika: Prikaz točnosti kroz epohe na skupu za treniranje i validacijskom skupu

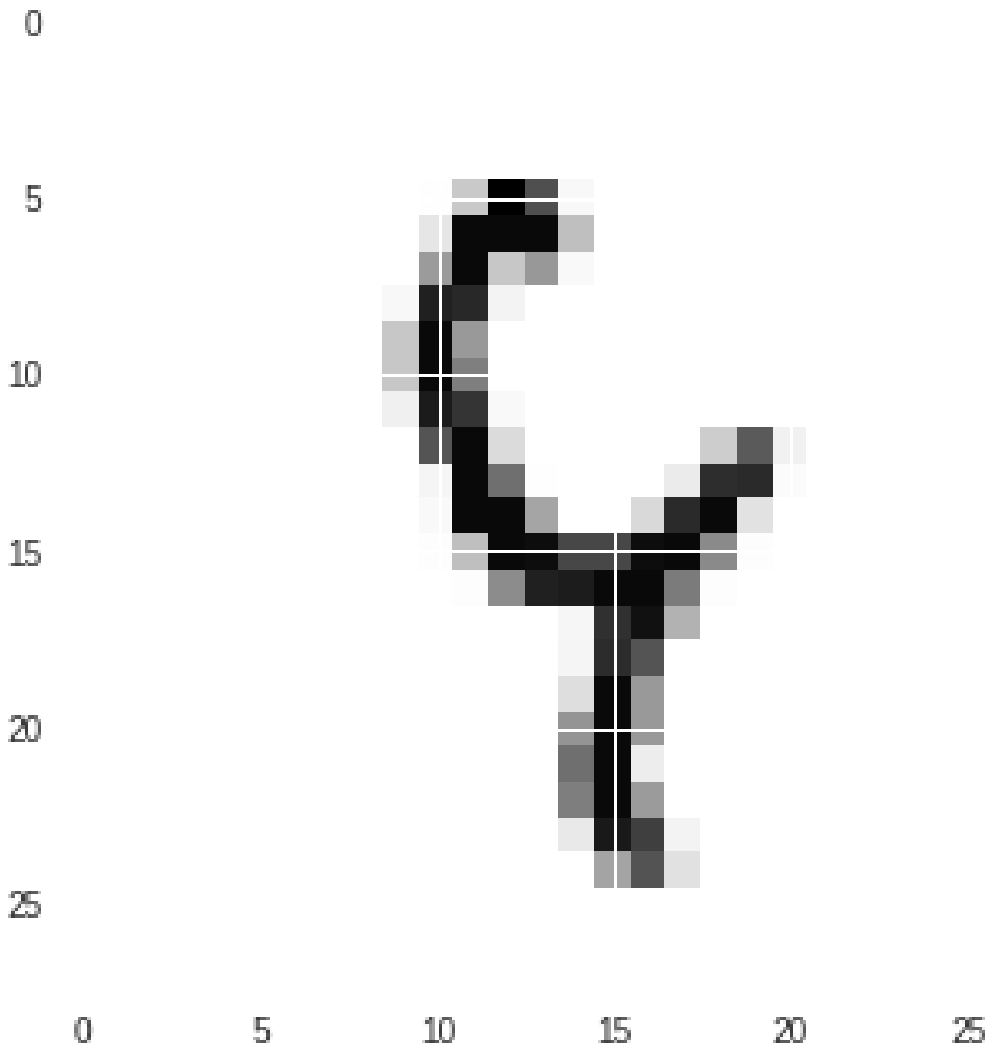
State of the art

Result	Method	Venue	Details
0.21%	Regularization of Neural Networks using DropConnect	ICML 2013	
0.23%	Multi-column Deep Neural Networks for Image Classification	CVPR 2012	
0.23%	APAC: Augmented PAttern Classification with Neural Networks	arXiv 2015	
0.24%	Batch-normalized Maxout Network in Network	arXiv 2015	Details
0.29%	Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree	AISTATS 2016	Details
0.31%	Recurrent Convolutional Neural Network for Object Recognition	CVPR 2015	
0.31%	On the Importance of Normalisation Layers in Deep Learning with Piecewise Linear Activation Units	arXiv 2015	
0.32%	Fractional Max-Pooling	arXiv 2015	Details
0.33%	Competitive Multi-scale Convolution	arXiv 2015	
0.35%	Deep Big Simple Neural Nets Excel on Handwritten Digit Recognition	Neural Computation 2010	Details

Slika:

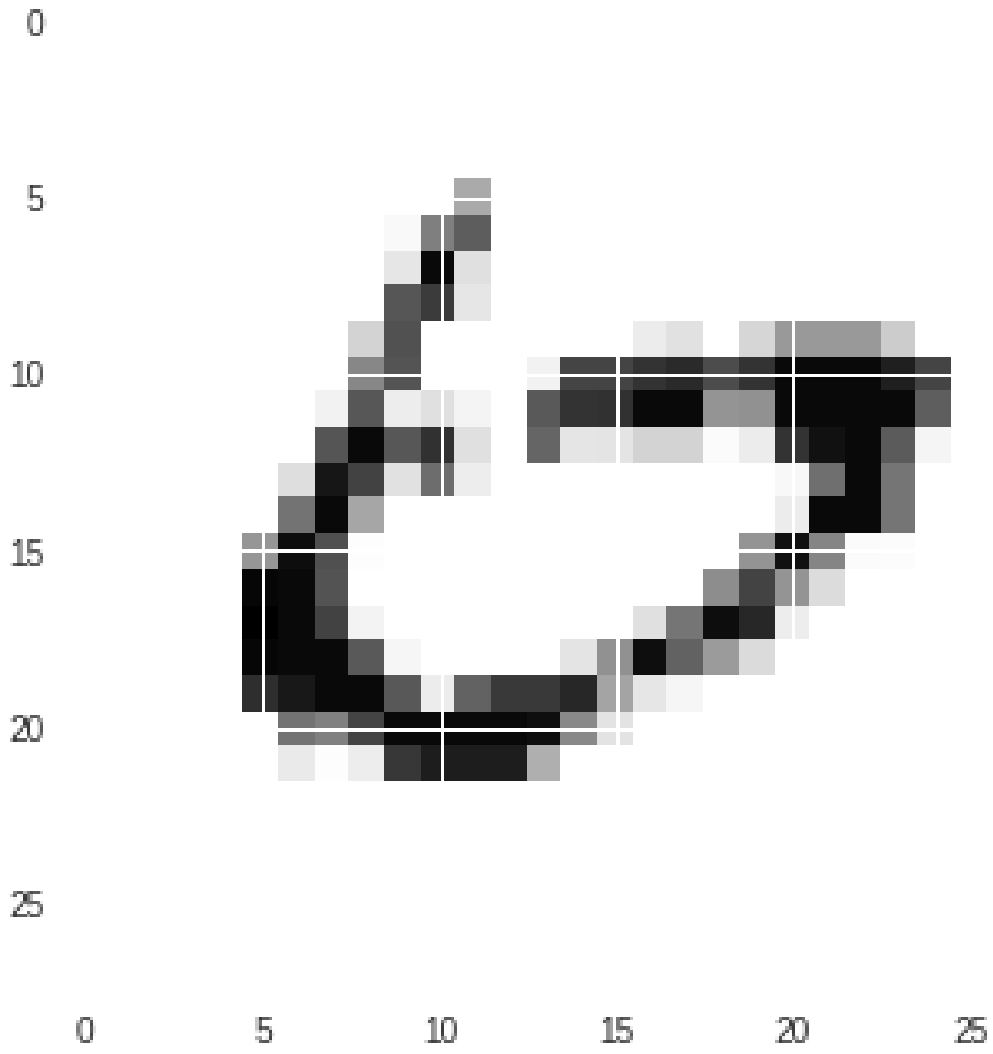
http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.h

Netočno klasificirane slike(1)



Slika: 4 s vjerojatnosti 1.0, trebao 9

Netočno klasificirane slike(2)

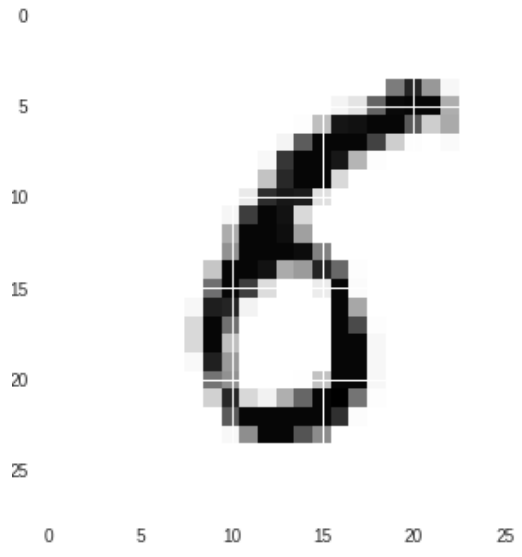


Slika: 0 s vjerojatnosti 0.99, trebao 6

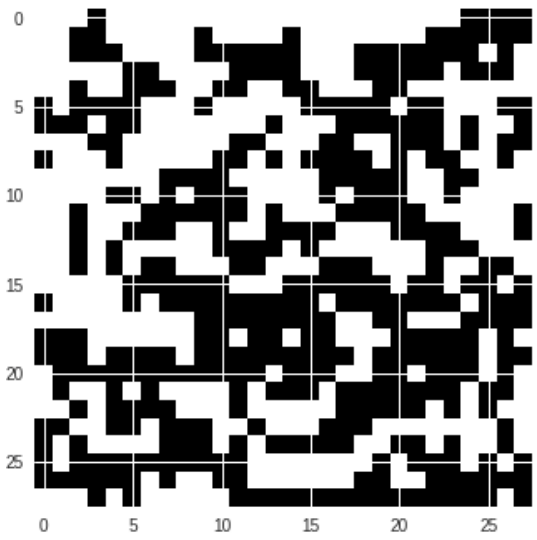
Neprijateljski primjeri

- Perturbiranje predznakom gradijenta(engl. Fast Gradient Sign Method)

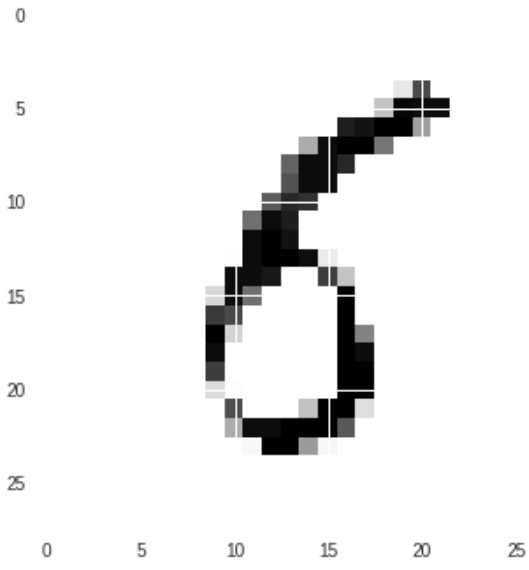
$$\eta = \epsilon sign(\nabla_{\boldsymbol{x}} J(\theta, \boldsymbol{x}, y)) \tag{1}$$



Slika: Predividio 6 s vjerojatnosti 0.98



Slika: Perturbacija η



Slika: Predividio 5 s 0.99

Neprijateljski primjeri - PyTorch

```
#x - ulazna slika, requires_grad = True
output = model(x)
loss = nn.CrossEntropyLoss(output, x_label)
loss.backward()
x_grad = torch.sign(x.grad.data)
x_adv = torch.clamp(x.data + epsilon*x_grad, 0, 1)
```

Zaključak

- SVHN: 91.80%, MNIST: 99.52%
- Duboki modeli poboljšavaju točnost
- Neprijateljski primjeri predstavljaju sigurnosni problem
- Budući rad: suparničko učenje, prepoznavanje više uzastopnih znamenki?