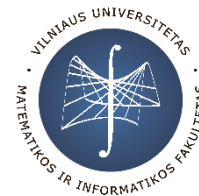




Vilniaus universitetas
Matematikos ir informatikos fakultetas
Informatikos katedra



Kapsuliniai neuroniniai tinklai

prof. dr. Olga Kurasova
Olga.Kurasova@mii.vu.lt

2018

Kapsuliniai neuroniniai tinklai

- **Kapsuliniai neuroniniai tinklai** (*Capsule Neural Network*, **CapsNet**) dar vadinami tinklų kapsulėmis.
- Teigiama, kad jų veikimas **panašesnis** į biologinių neuronų mechanizmus.
- Pasiūlyti **prieš metus** (2017 m.)

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in Neural Information Processing Systems* (pp. 3856-3866). <https://arxiv.org/abs/1710.09829>

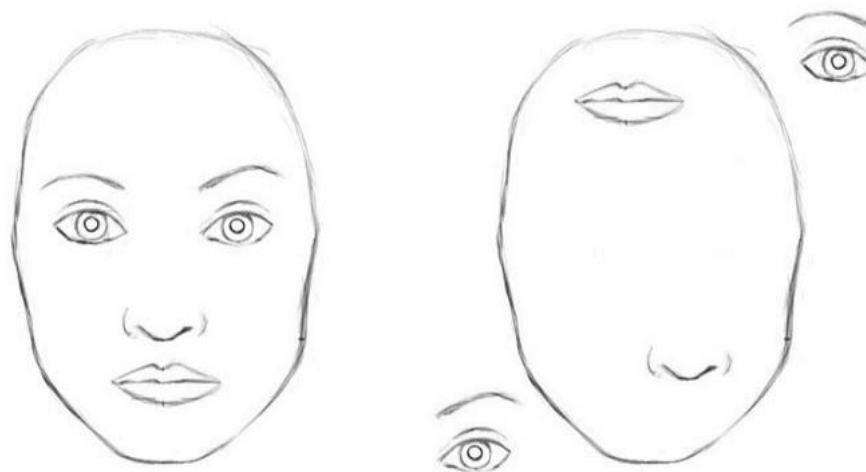
Citatos iš straipsnio:

- **Human vision ignores irrelevant details** by using a carefully determined sequence of fixation points to ensure that only a tiny fraction of the optic array is ever processed at the highest resolution.
- We show that a discrimininatively trained, multi-layer capsule system achieves state-of-the-art performance on MNIST and **is considerably better** than a convolutional net at recognizing highly overlapping digits.

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in Neural Information Processing Systems* (pp. 3856-3866). <https://arxiv.org/abs/1710.09829>

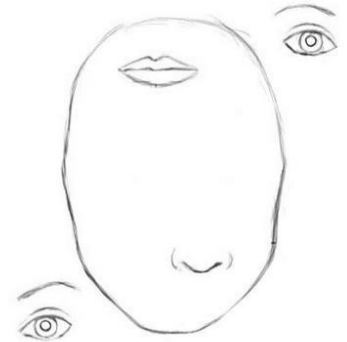
CNN trūkumai

- Nagrinėjime **paprastą pavyzdį**. Iš kokių komponentų sudarytas veidas? Veidas yra ovalios formos, yra dvi akys, nosis ir burna.
- CNN tinklui tai labai svarbios komponentų savybės, tačiau visai **nesvarbi jų padėtis** vieno kito atžvilgiu.

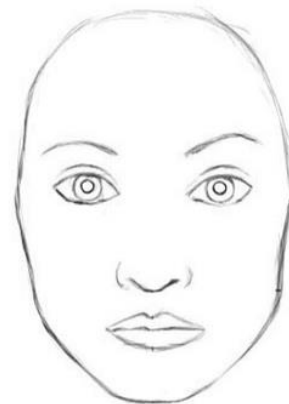


Kas geriau?

```
if (2 eyes && 1 nose && 1 mouth) {  
    It's a face!  
}
```

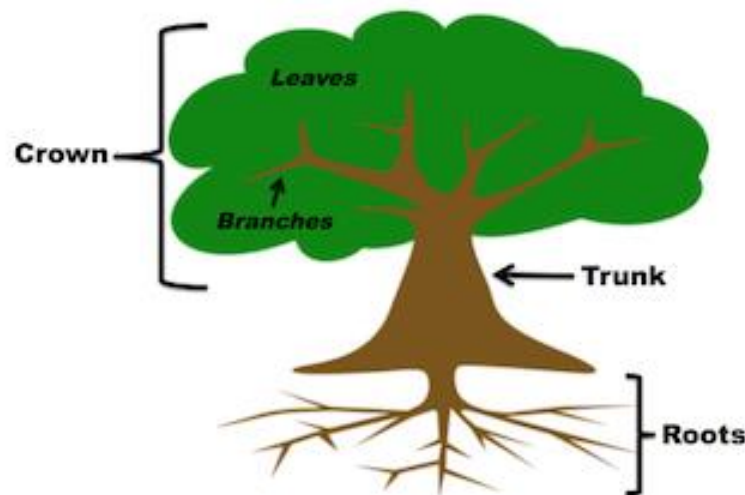


```
if (2 adjacent eyes && nose under eyes && mouth under nose) {  
    It's a face!  
}
```



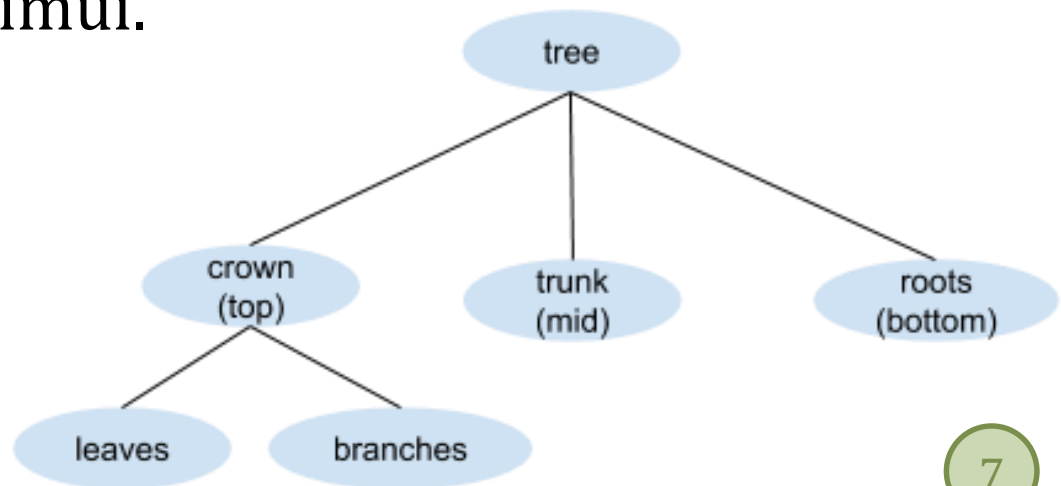
Žmogaus rega

- Objektai **sudaryti iš komponentų** (sudedamųjų dalių). Pvz., medis sudarytas iš kamieno, lapų vainiko ir šaknų.
- Įprastai dalys turi tam tikrą **hierarchiją**. Pvz., lapų vainiką sudaro šakos, šakos turi lapus ir pan.



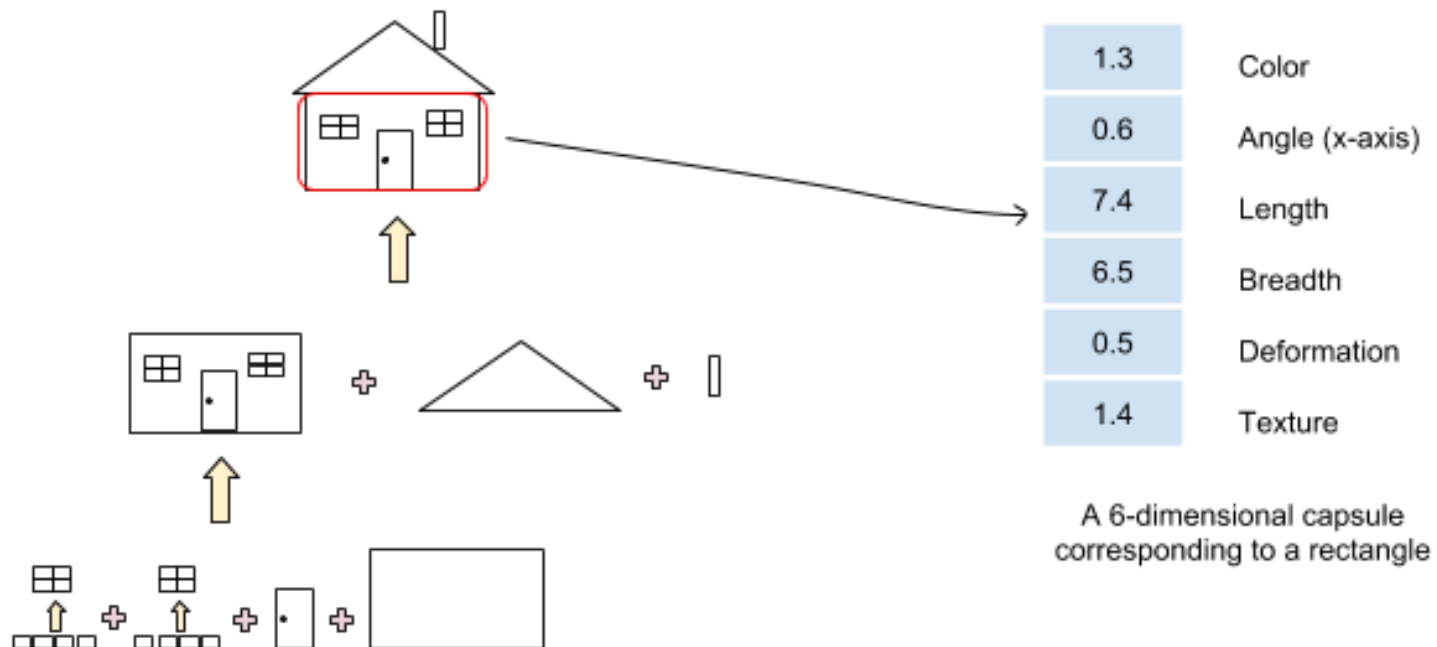
Žmogaus rega

- Kai mes matome tam tikrą objektą, mūsų akys **fiksuoja** tam tikrus taškus, o jų pozicija ir prigimtis **padedą smegenims atpažinti** šį objektą.
- Smegenys **ne analizuoja** kiekvieną komponentą tiek detaliai. Sudaroma tam tikra **hierarchija**, kuri padeda atpažinimui.



Pavyzdys

- Nagrinėkime, iš ko sudarytas **namas**. Tegu elementariausias objektas būna **stačiakampis**.
- Sudarykime 6 kapsules.



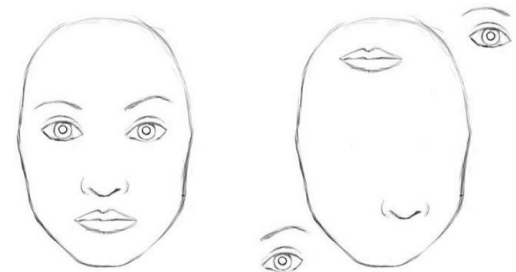
CapsNet vs CNN

- Nustatyta, kad **žmogus**, gavęs vaizdinę informaciją, dekonstruoja hierarchinį aplinkos reprezentavimą ir stengiasi jį sutapatinti su jau **išmoktais šablonais** ir smegenyse saugomus ryšius.
- Objektų reprezentavimas smegenyse nepriklauso nuo juos sudarančių briaunų, kaip yra **CNN paradigmoje**.
- CapsNet tinkle svarbus **objektų ryšys**. Kitais žodžiais – 3D rekonstrukcija.



CNN trūkumai

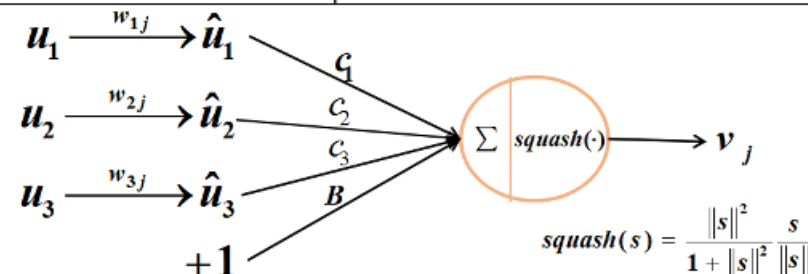
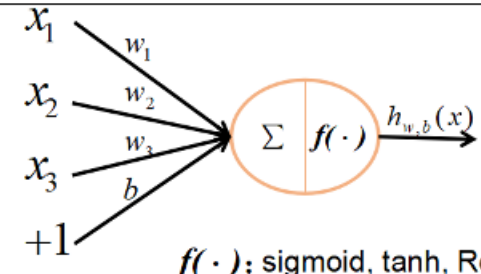
- „Klaidos skleidimo atgal“ algoritmas reikalauja **didelio kiekio duomenų** tinklui išmokyti.
- Pakeitus objekto vietą, neuronai **kitaip aktyvuojami**. Nors CNN tinkluose tai sprendžiama taikant duomenų didinimą (*augmentation*), tačiau tai lieka ne iki galo išspręsta problema.
- Naudojant **sujungimo** (*pooling*) sluoksnį prarandamas ryšys objektų.



Kapsulės

- **Kapsulės** – tai neuronų grupės.
- Neuronai sugrupuojami kiekviename sluoksnyje į kapsules, siekiant atlikti **vidinį skaičiavimą**, o išėjimuose gaunama „kompaktiška“ informacija.
- Kapsulės išėjimuose gaunami du dalykai:
 - **Tikimybė**, kad objektas priklauso kažkuriai klasei.
 - Apibendrinta **objekto poza – rekonstrukcija** (vieta, orientacija, mastelis, deformacija, spalva) ir kt.

Kapsuliniai neuroniniai tinklai

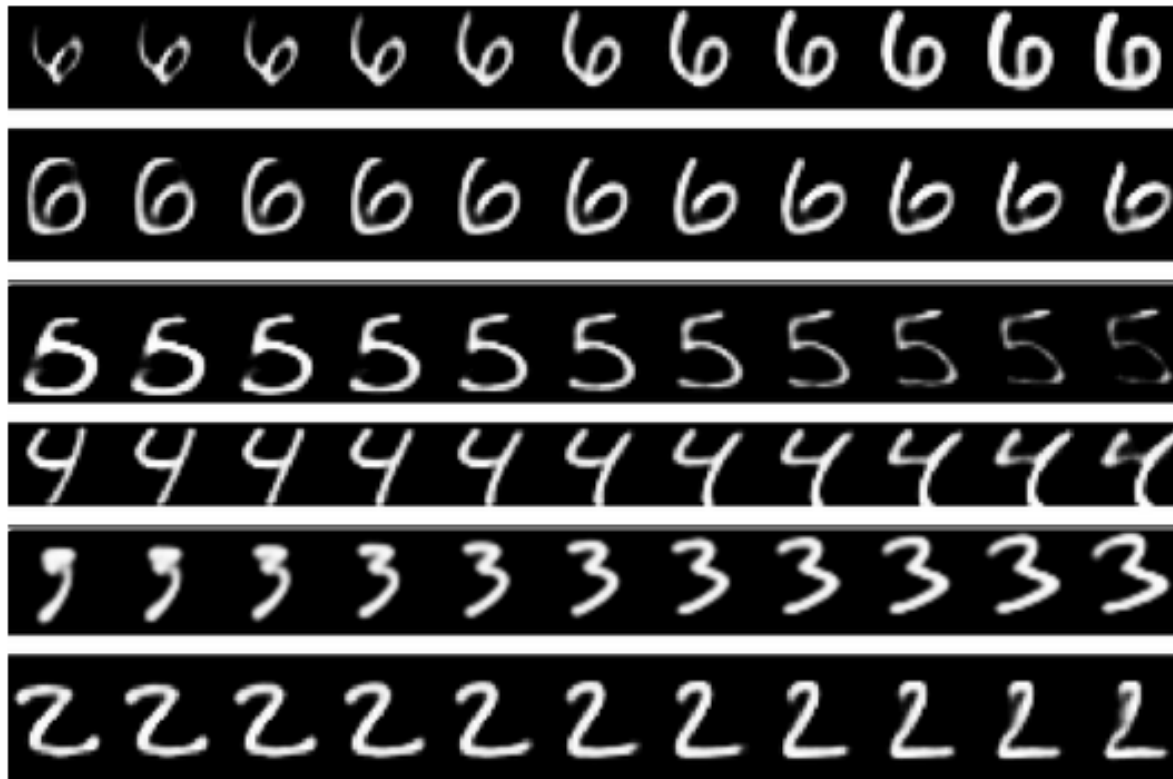
capsule		VS.	traditional neuron
Input from low-level neurons/capsules		$\text{vector}(u_i)$	$\text{scalar}(x_i)$
Operations	Linear/Affine Transformation	$\hat{u}_{ji} = W_{ij} u_i + B_j$ (Eq. 2)	$a_{ji} = w_{ij} x_i + b_j$
	Weighting	$s_j = \sum_i c_{ij} \hat{u}_{ji}$ (Eq. 2)	$z_j = \sum_{i=1}^3 1 \cdot a_{ji}$
	Summation		
	Non-linearity activation	$v_j = \text{squash}(s_j)$ (Eq. 1)	$h_{w,b}(x) = f(z_j)$
output		$\text{vector}(v_j)$	$\text{scalar}(h)$
<div></div>		<div></div>	

Capsule = New Version Neuron!
vector in, vector out VS. scalar in, scalar out

<https://github.com/naturomics/CapsNet-Tensorflow>

CapsNet išėjimai

- CapsNet tinklai ne tik **atpažįsta klase**, bet ir generuoja „vidinius“ **paveiksliukus**.



Kapsulių sluoksniai

- **Kapsulės** apjungia savyje neuronų grupes.
- Kapsules sudaro **sluoksnius**.
- **Žemesniame** sluoksnyje kapsulės atitinka paprastus objektus (stačiakampis, trikampis, apskritimas). **Aukštesniame** – sudėtingesnius objektus (durys, langai).
- Būtinas **apjungimo** mechanizmas. Autoriai jį vadina „Dynamic routing between capsules“

Routing algorithm

Procedure 1 Routing algorithm.

```
1: procedure ROUTING( $\hat{u}_{j|i}, r, l$ )
2:   for all capsule  $i$  in layer  $l$  and capsule  $j$  in layer  $(l + 1)$ :  $b_{ij} \leftarrow 0$ .
3:   for  $r$  iterations do
4:     for all capsule  $i$  in layer  $l$ :  $c_i \leftarrow \text{softmax}(b_i)$  ▷ softmax computes Eq. 3
5:     for all capsule  $j$  in layer  $(l + 1)$ :  $s_j \leftarrow \sum_i c_{ij} \hat{u}_{j|i}$ 
6:     for all capsule  $j$  in layer  $(l + 1)$ :  $v_j \leftarrow \text{squash}(s_j)$  ▷ squash computes Eq. 1
7:     for all capsule  $i$  in layer  $l$  and capsule  $j$  in layer  $(l + 1)$ :  $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j$ 
   return  $v_j$ 
```

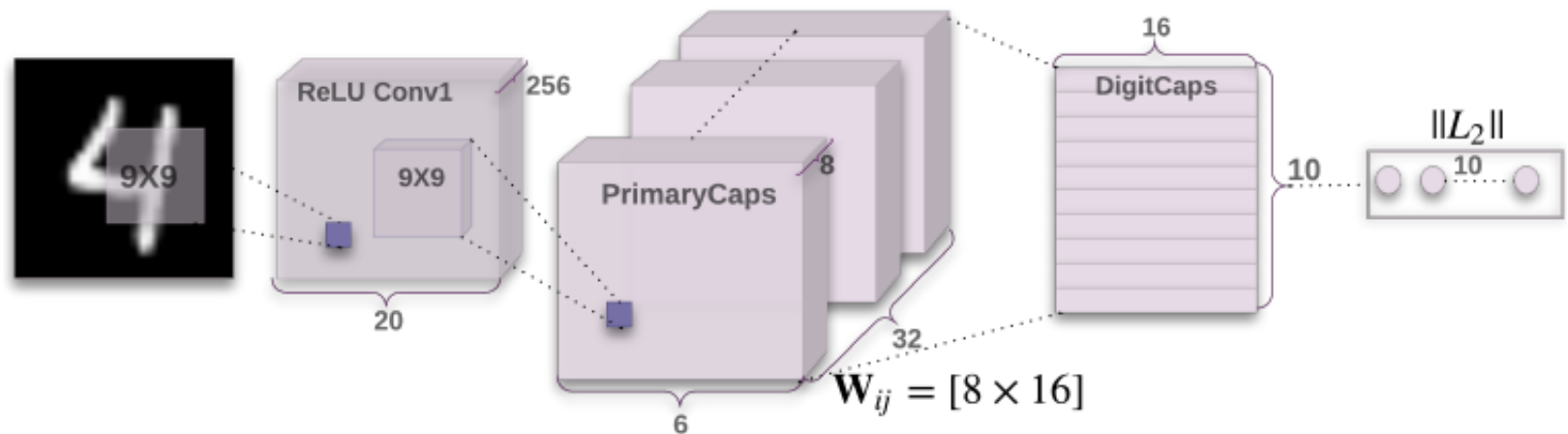
$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (1)$$

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \quad (3)$$

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in Neural Information Processing Systems* (pp. 3856-3866). <https://arxiv.org/abs/1710.09829>

CapsNet architektūra

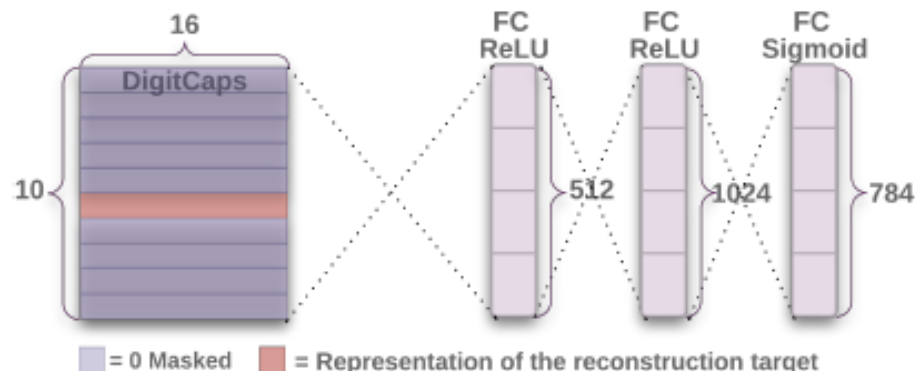
Figure 1: A simple CapsNet with 3 layers. This model gives comparable results to deep convolutional networks (such as Chang and Chen [2015]). The length of the activity vector of each capsule in DigitCaps layer indicates presence of an instance of each class and is used to calculate the classification loss. W_{ij} is a weight matrix between each $u_i, i \in (1, 32 \times 6 \times 6)$ in PrimaryCapsules and $v_j, j \in (1, 10)$.



Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in Neural Information Processing Systems* (pp. 3856-3866). <https://arxiv.org/abs/1710.09829>


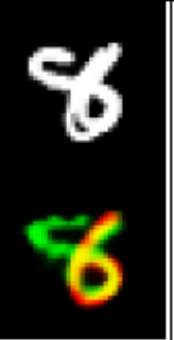
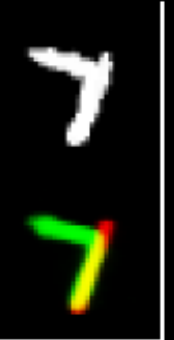


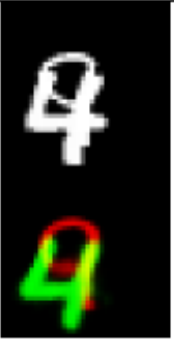
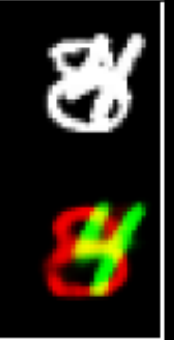


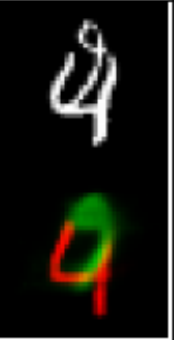
CapsNet architektūra

Figure 2: Decoder structure to reconstruct a digit from the DigitCaps layer representation. The euclidean distance between the image and the output of the Sigmoid layer is minimized during training. We use the true label as reconstruction target during training.



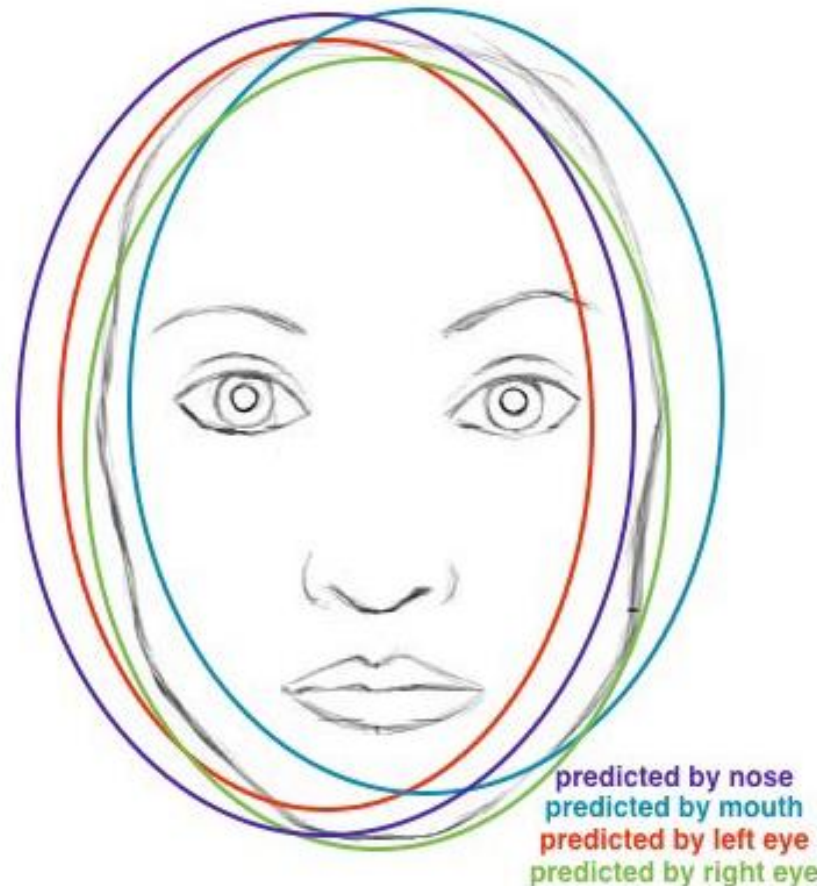
Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in Neural Information Processing Systems* (pp. 3856-3866). <https://arxiv.org/abs/1710.09829>

CapsNet rezultatai

R: (2, 7) L: (2, 7)	R: (6, 0) L: (6, 0)	R: (6, 8) L: (6, 8)	R: (7, 1) L: (7, 1)	*R: (5, 7) L: (5, 0)	*R: (2, 3) L: (4, 3)	R: (2, 8) L: (2, 8)	R:P: (2, 7) L: (2, 8)
							
R: (8, 7) L: (8, 7)	R: (9, 4) L: (9, 4)	R: (9, 5) L: (9, 5)	R: (8, 4) L: (8, 4)	*R: (0, 8) L: (1, 8)	*R: (1, 6) L: (7, 6)	R: (4, 9) L: (4, 9)	R:P: (4, 0) L: (4, 9)
							

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Advances in Neural Information Processing Systems* (pp. 3856-3866). <https://arxiv.org/abs/1710.09829>

CapsNet



<https://medium.com/ai%C2%B3-theory-practice-business/understanding-hintons-capsule-networks-part-ii-how-capsules-work-153b6ade9f66>

CapsNet

- **Pirmoji** CapsNet dalis yra tradicinis **konvoliucinis sluoksnis** ir **ReLU** sluoksnis. Tikslas – išskirti **pagrindinius požymius**, tokius kaip briaunos.
- **Antroji** dalis – **PrimaryCaps**. Čia pradedama nuo paprastų požymių (*instantiation parameters*), baigiama – sudėtingais. Naudojama **Squashing** funkcija (nepakeisčiant vektoriaus krypties, o tik jo ilgį, rezultatas intervale $(0,1)$).
- **Trečioji** dalis – **Rooting** procedūra (vietoje MaxPooling).
- **Ketvirtoji** dalis – **DigitCaps** (Class Capsules)
- **Penktoji** dalis – **rekonstrukcija**.

CapsNet privalumai

- Pasiekiamas **aukštesnis** tikslumas (nei CNN).
- Reikalauja **mažiau** mokymo duomenų.
- Tinkamesni vaizdų **segmentavimui** ir objektų **atpažinimui**.
- Atpažįsta **persidengiančius** objektus.

Dar kartą apie CapsNet

- <https://www.analyticsvidhya.com/blog/2018/04/essentials-of-deep-learning-getting-to-know-capsulenets/>
- <https://software.intel.com/en-us/articles/understanding-capsule-network-architecture>
-

CapsNet taikymai

Nors jie sukurti visai neseniai, bet jau pradėti intensyviai **taikyti įvairiems uždaviniams spręsti:**

- Kim, Y., Wang, P., Zhu, Y., & Mihaylova, L. (2018, October). A Capsule Network for Traffic Speed Prediction in Complex Road Networks. In 2018 Sensor Data Fusion: Trends, Solutions, Applications (SDF) (pp. 1-6). IEEE.
- Zhu, Z., Peng, G., Chen, Y., & Gao, H. (2019). A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis. Neurocomputing, 323, 62-75.
- Afshar, P., Mohammadi, A., & Plataniotis, K. N. (2018). Brain tumor type classification via capsule networks. arXiv preprint arXiv:1802.10200.
- Daugiau straipsnių:
<http://principlesofdeeplearning.com/index.php/resources/list-of-research-papers-on-capsule-networks/>