



9-ASIS
LIETUVOS
JAUNŲJŲ
MATEMATIKŲ
SUSITIKIMAS

Lokalizuoti konvoliuciniai neuroniniai tinklai

Mantas Lukoševičius
<https://mantas.info/>
mantas.lukosevicius@ktu.edu
2020.12.28



Turinys

- Idėja/motyvacija
- Ankstesni panašūs sprendimai
- Realizacija
- Rezultatai

Straipsnis

Open Access

Article

Localized Convolutional Neural Networks for Geospatial Wind Forecasting

by  **Arnas Uselis** * ,  **Mantas Lukoševičius** *   and  **Lukas Stasytis**

Faculty of Informatics, Kaunas University of Technology, LT-51368 Kaunas, Lithuania

* Authors to whom correspondence should be addressed.

Energies **2020**, *13*(13), 3440; <https://doi.org/10.3390/en13133440> <https://doi.org/10.3390/en13133440>

Received: 19 May 2020 / Revised: 19 June 2020 / Accepted: 24 June 2020 / Published: 3 July 2020

(This article belongs to the Special Issue **Machine Learning and Deep Learning for Energy Systems**)

View Full-Text

Download PDF

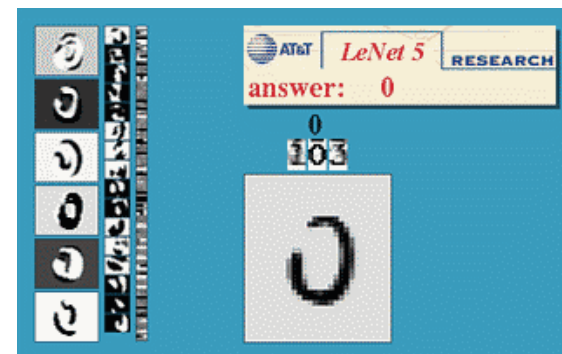
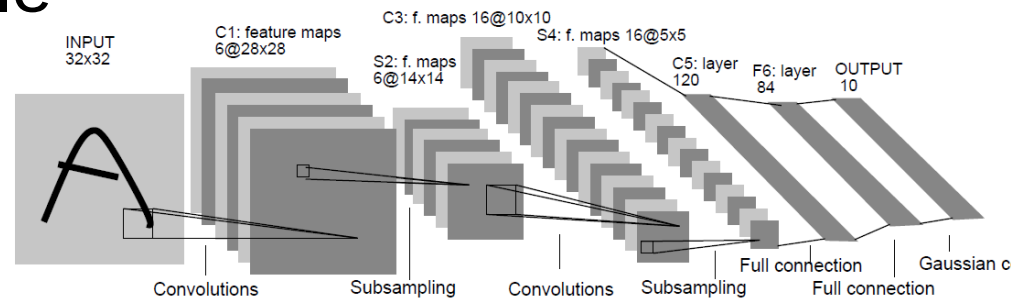
Browse Figures

Review Reports

Cite This Paper

Convoliucinis neuroninis tinklas (CNT)

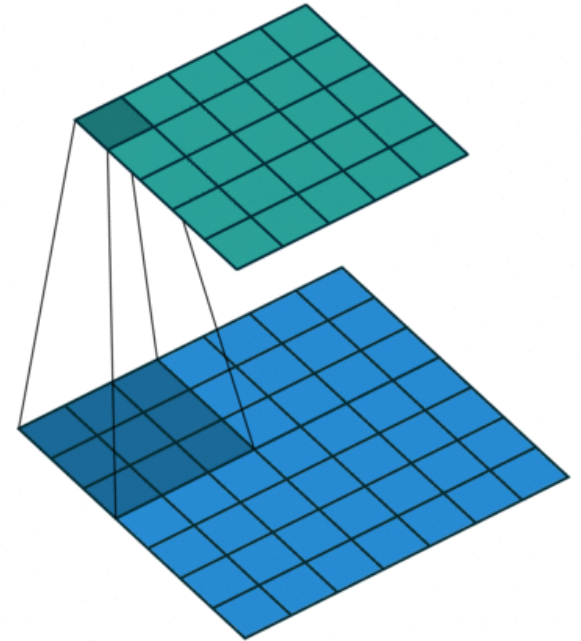
- Vaizdams, bei kitiems gardele organizuotiems duomenims
- Pradžiai: Yann LeCun et al, 1998
- Perversmas kompiuterinėj regoj
- Giliojo mokymosi klasika
- Svarbiausias elementas – konvoliucinis sluoksnis



<http://yann.lecun.com/exdb/lenet/>

Konvoliucinis sluoksnis

- 2D konvoliucijos operacija
- Neuronas / „filtras“
 - Svorinė suma ir netiesiškumas
 - Mato tik mažą plotelį
 - Slenkamas per visą įvestį
 - Alternatyviai: neuronai dalinasi tuos pačius svorius

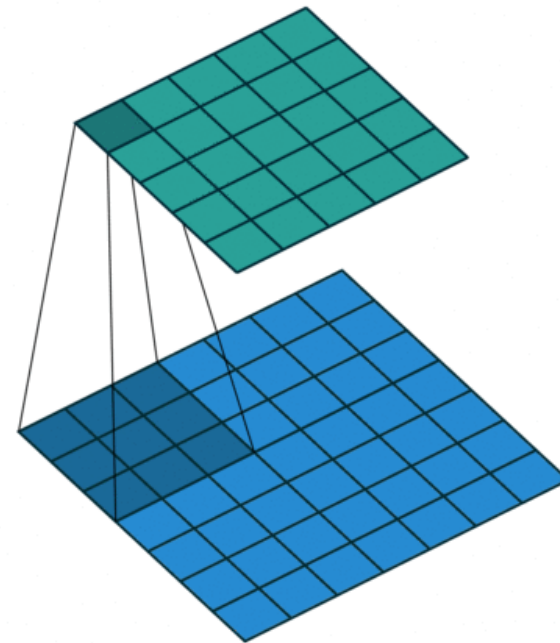


iš https://github.com/vdumoulin/conv_arithmetic

Konvoliucinio sluoksnio privalumai

Lyginant su pilnai sujungtu

- Daug mažiau apmokomų svorių
- Kiekvienas filtras mokomas ant kiekvieno plotelio
- Neprištas prie įvesties dydžio
- Atpažinimas nepriklauso nuo vietos vaizde



iš https://github.com/vdumoulin/conv_arithmetic

Vieta kartais svarbi!

- Konvoliucinis filtras negali išmokti nieko susieto su vieta paveikslėlyje
 - Nežino kurioj vietoj yra, negali nieko išmokto užsaugoti susieto su vieta
- Visiškas nepriklausymas nuo vietos yra geras, kai paveikslėlis gali būti bet kaip kadruotas
- Tačiau kartais nėra optimalus, kai kadravimas yra fiksuotas
 - Geoerdviniams taikymams: kiekvienas taškas turi savo ypatybių
 - Kai gardelė nėra tobula: netikslios lokacijos, defektai
 - Statiškai kadruotiems paveikslėliams: CCTV, vaizdai iš prietaisų, ...
 - Turintiems statinių artefaktų: titrai, logotipai, lęšio iškraipymai, mirę pikseliai, ...

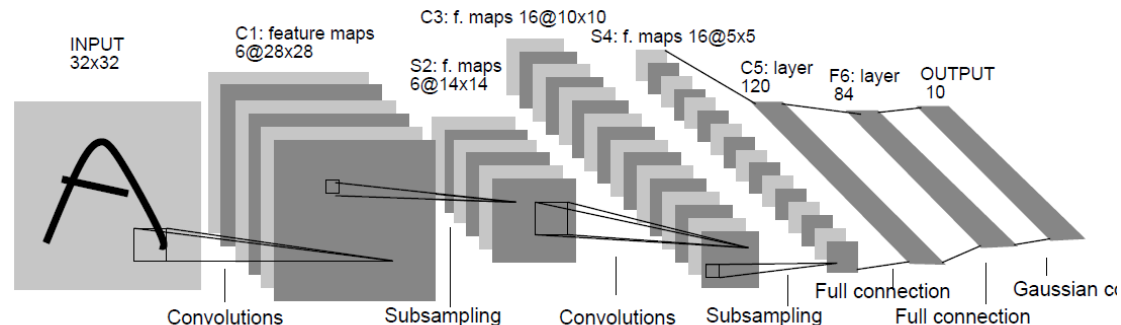
Balansas

- Norime, kad būtų atsižvelgiama į vietą
- Tačiau, taip pat norime neprarasti CNT gerųjų savybių tiek kiek galima ar reikia
 - Negrįžtame prie pilnai sujungto daugiasluoksnio perceptrono

Ankstesni panašūs sprendimai

Ankstesni panašūs sprendimai

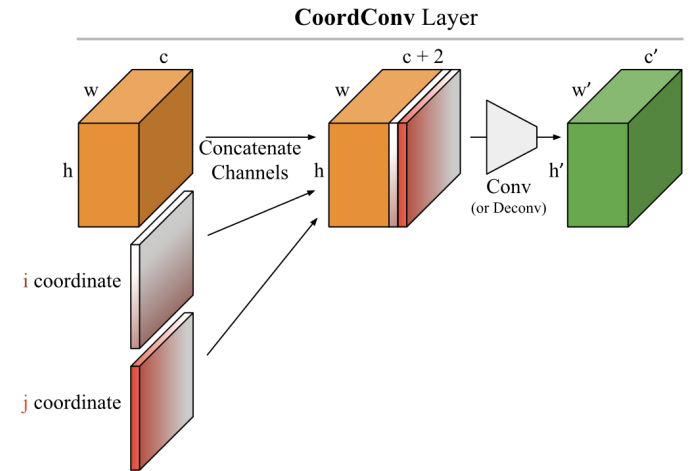
- Vieta svarbi aukštesniuose CNT sluoksniuose



- Klasika vaizdų atpažinimo architektūrose
- Nesimaišo su konvoliucijomis, eina po jų
- CoordConv
- ...

CoordConv

- Tiesiog pridėti x ir y koordinates kaip 2 papildomus įvesties kanalus
 - Turėjome omeny kaip vieną iš paprasčiausių lokalizacijų, bet ji jau buvo realizuota
- Uber tyrėjai pastebėjo panašias CNT problemas, savo sprendimą paskelbė NeurIPS 2018 konferencijoje
 - „An intriguing failing of convolutional neural networks and the CoordConv solution“ Liu et al, 2018
- Rezultatai pagerėjo praktiškai visur, kur vieta yra svarbi



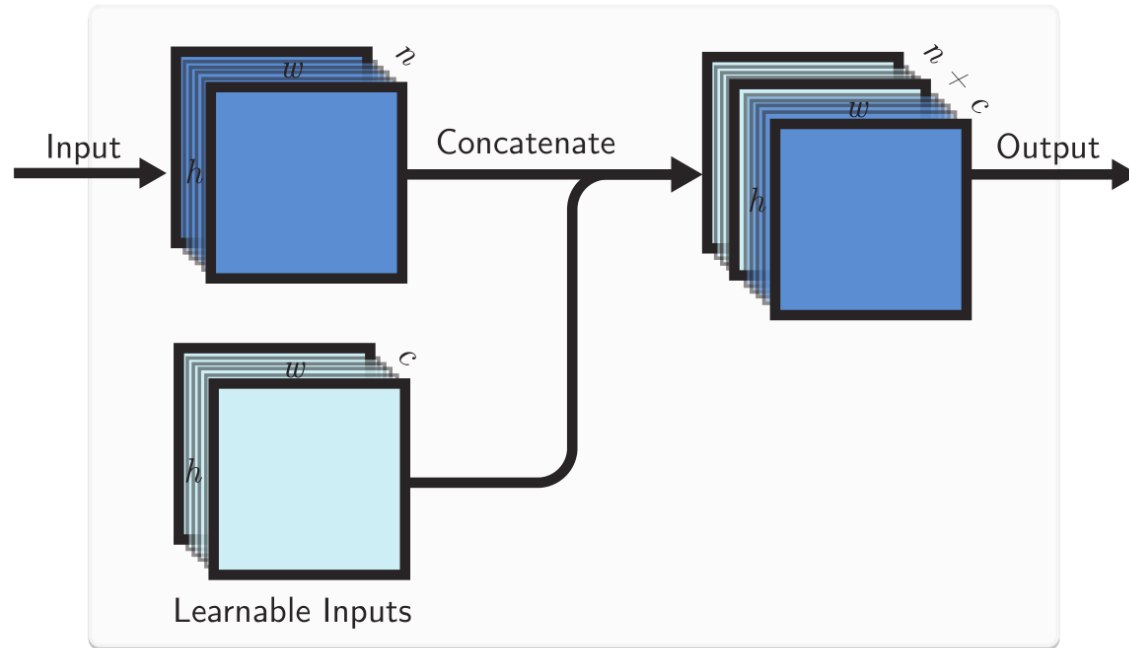
<https://eng.uber.com/coordconv/>

<https://papers.nips.cc/paper/8169-an-intriguing-failing-of-convolutional-neural-networks-and-the-coordconv-solution.pdf>

Mūsų sprendimas –
keli skirtingi lokalizuojantys sluoksniai
prieš konvoliucinį

Išmokstamos įvestys

- Leidžia išmokti lokalias savybes

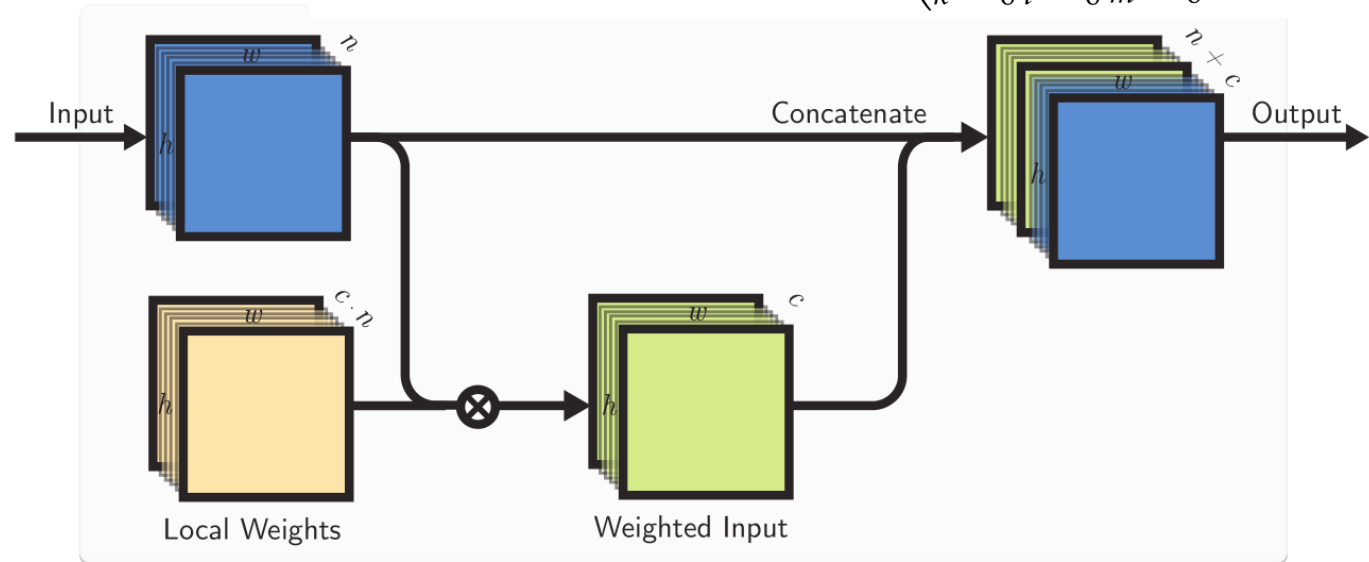


Lokalūs svoriai

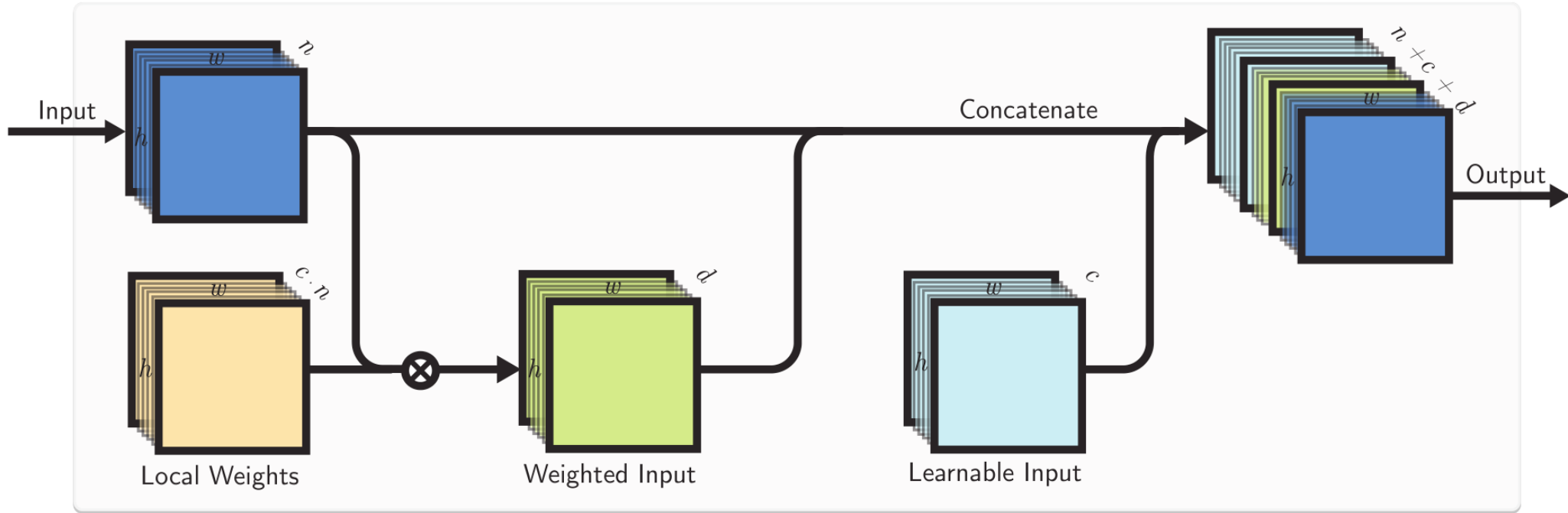
- Leidžia išmokti lokalias įvesties transformacijas

– / vietos „svarba“

$$I_{ijo} = f(X \otimes_{(A,B)} M)_{ijo} = f \left(\sum_{k=0}^n \sum_{l=0}^A \sum_{m=0}^B X_{i+l,j+m,k} \cdot M_{ijklmo} \right)$$

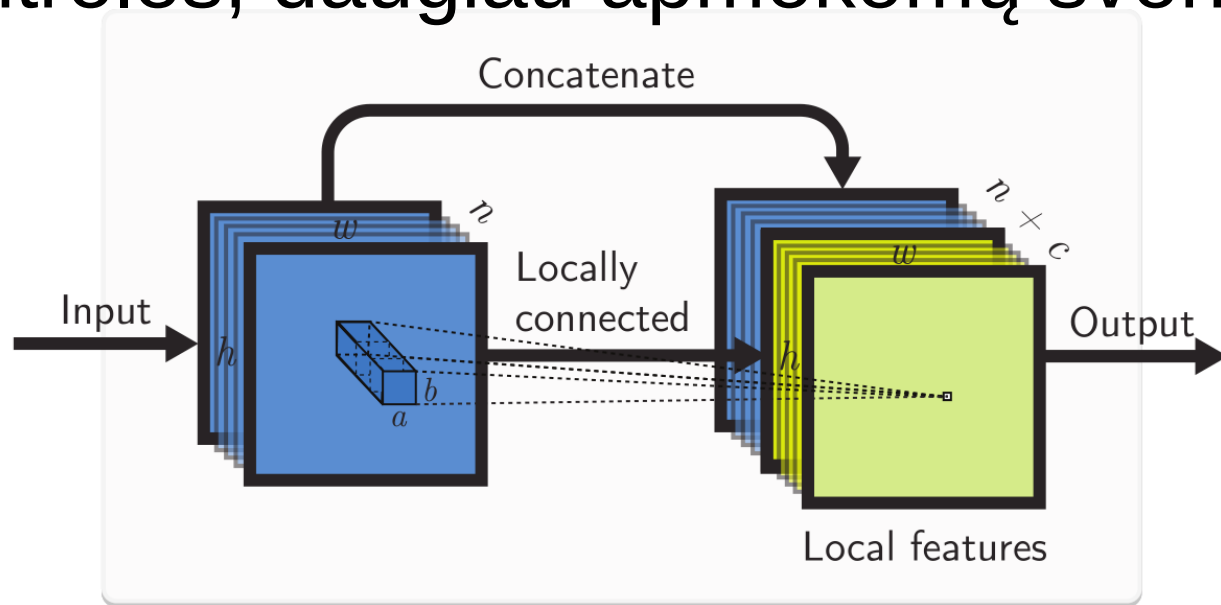


Abu kartu



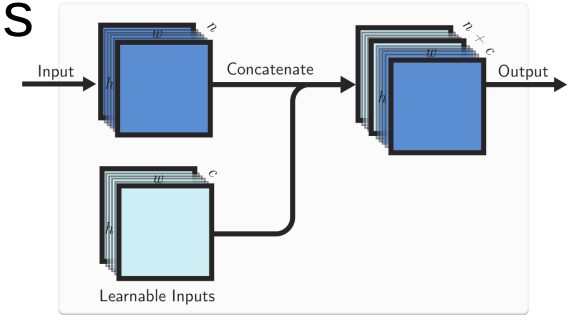
~ Lokaliai sujungtas sluoksnis

- Su šališkumo (angl. *bias*) reikšmėm
- Mažiau kontrolės, daugiau apmokomų svorių



Realizacija TensorFlow (TF)

- Sluoksniai realizuoti standartiniais TF elementais
 - Kartais ne visai pagal paskirtį
 - TF karkaso pilnai apmokomi
- Pvz., išmokstamos įvestys:

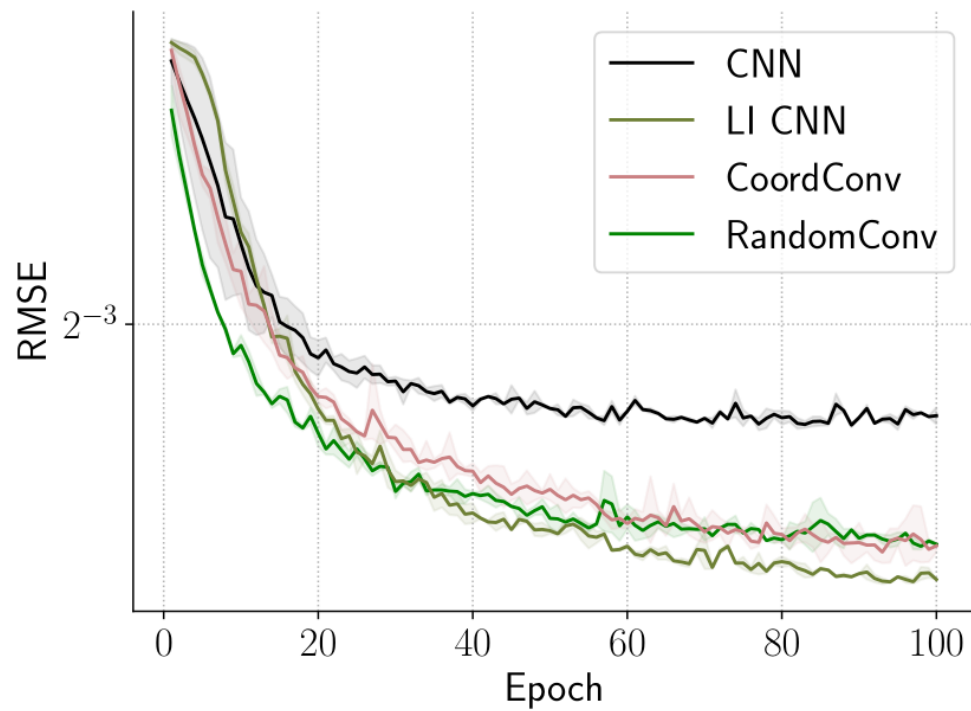
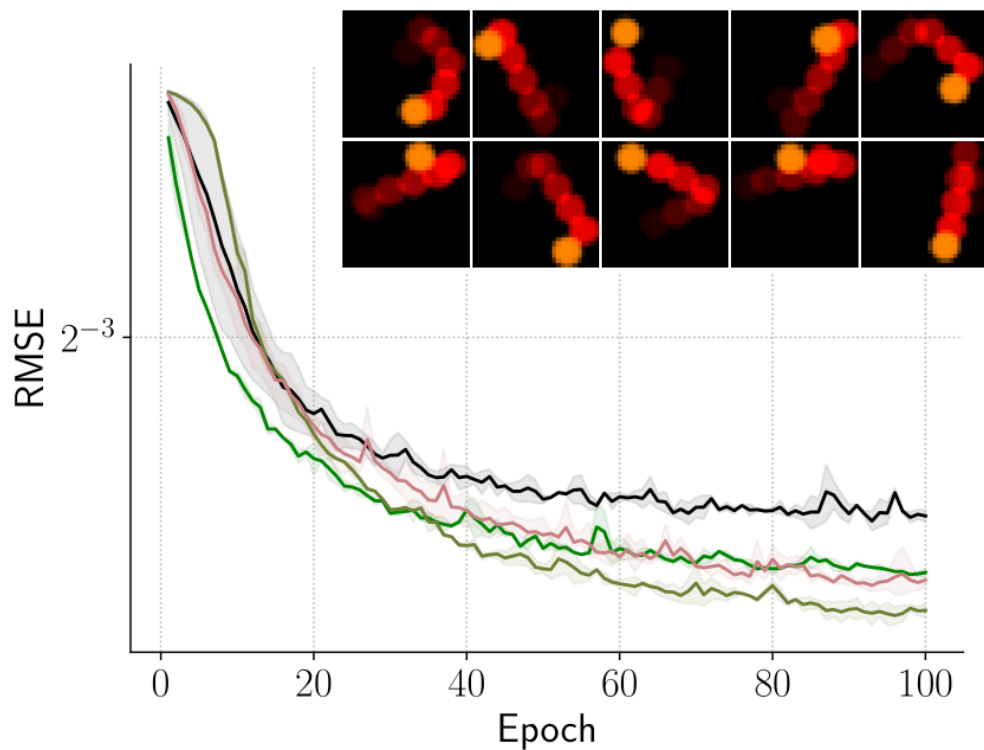


```
...
# To implement learnable inputs, we use auxillary input...
I2 = Input(shape=(self.input_dim[0], self.input_dim[1], 1))
# ...that receives a matrix of 1s and implements locally connected layer without biases with a 1x1 kernel.
C1 = LocallyConnected2D(self.learnable_inputs_count, kernel_size=(1, 1), input_shape=(self.input_dim[0],
self.input_dim[1], 1), bias_initializer='zeros', use_bias=False)(I2)
# Finally, we concatenate real input with the learnable input.
Merged1 = Concatenate(-1)([I1, C1])
...
```

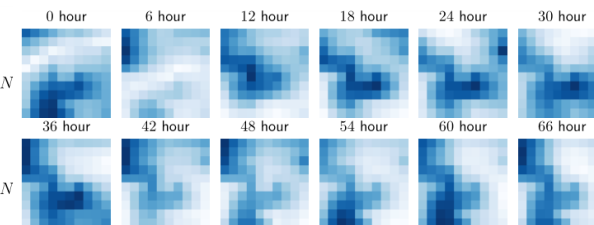
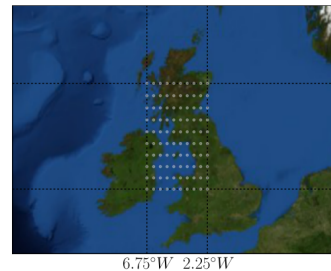
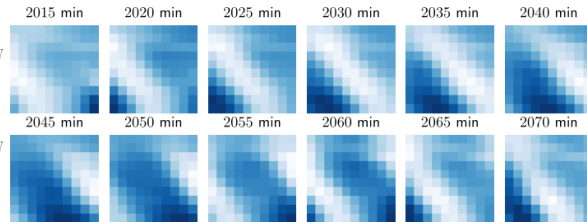
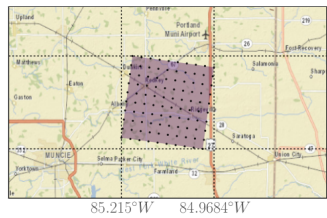
<https://github.com/oshapio/Localized-CNNs-for-Geospatial-Wind-Forecasting>

Rezultatai

Atšokantis kamuoliukas



Vėjo greičio prognozavimas



- 2D gardelėje, 3 skirtingos duomenų aibės
- Ankstesnės *state-of-the-art* architektūros kombinuoja laikinius ir erdvinius aspektus
 - erdviniai realizuojami konvoliuciniais sluoksniais
→ galima taikyti mūsų lokalizacijas
 - kitus hiperparametrus keičiame taip, kad bendras apmokomų svorių kiekis nebūtų didesnis

Vėjo greičio prognozavimo rezultatai

- Trumpesnėm prognozėm svarbesnis erdvinis komponentas, ilgesnėm – laikinis
- Praktiškai visoms architektūroms, kur svarbus erdvinis komponentas, mūsų lokalizacijos pagerino prognozes
- Jei gardelė netobula, kartais verta naudoti tik mūsų lokaliai transformuotas įvestis, bet išmesti originalias

Rezultatų pavyzdys

Model	Direct Embedding		Optimized Embedding	
	RMSE	MAE	RMSE	MAE
DL-STF (All nodes) * [33]	1.6200	1.1800	-	-
PR	1.791 ± 0.0000	1.238 ± 0.0000	1.791 ± 0.0000	1.238 ± 0.0000
CNN	1.527 ± 0.0059	1.140 ± 0.0057	1.509 ± 0.0084	1.119 ± 0.0066
MLP *	1.578 ± 0.0090	1.184 ± 0.0131	-	-
LSTM *	1.665 ± 0.0150	1.232 ± 0.0125	-	-
CoordConv [5]	1.524 ± 0.0064	1.134 ± 0.0066	1.519 ± 0.0135	1.124 ± 0.0085
ConvLSTM [7]	1.536 ± 0.0089	1.135 ± 0.0066	1.503 ± 0.0093	1.110 ± 0.0063
PSTN * [13]	1.724 ± 0.0228	1.293 ± 0.0149	1.716 ± 0.0113	1.271 ± 0.0152
PDCNN * [15]	1.696 ± 0.0173	1.277 ± 0.0151	1.696 ± 0.0120	1.265 ± 0.0100
E2E [16]	1.627 ± 0.0087	1.224 ± 0.0102	1.579 ± 0.0145	1.179 ± 0.0132
FC-CNN * [16]	1.676 ± 0.0210	1.255 ± 0.0182	1.676 ± 0.0145	1.251 ± 0.0076
LI CNN	1.518 ± 0.0088	1.133 ± 0.0074	1.505 ± 0.0094	1.118 ± 0.0048
LW CNN	1.516 ± 0.0062	1.132 ± 0.0044	1.508 ± 0.0091	1.120 ± 0.0044
LW111 CNN	1.511 ± 0.0038	1.129 ± 0.0053	1.506 ± 0.0055	1.122 ± 0.0043
LI + LW CNN	1.512 ± 0.0079	1.131 ± 0.0061	1.502 ± 0.0087	1.118 ± 0.0079
Persistent LI + LW CNN	1.507 ± 0.0072	1.124 ± 0.0062	1.501 ± 0.0077	1.111 ± 0.0050
LI + LW222 CNN	1.508 ± 0.0037	1.127 ± 0.0054	1.504 ± 0.0071	1.117 ± 0.0057
Persistent LI + LW222 CNN	1.507 ± 0.0044	1.126 ± 0.0066	1.499 ± 0.0072	1.116 ± 0.0057
LI + LW - I CNN	1.505 ± 0.0066	1.126 ± 0.0052	1.508 ± 0.0156	1.123 ± 0.0121
Persistent LI + LW - I CNN	1.496 ± 0.0046	1.116 ± 0.0047	1.492 ± 0.0095	1.106 ± 0.0083
LI + LW222 - I CNN	1.515 ± 0.0080	1.136 ± 0.0070	1.518 ± 0.0084	1.135 ± 0.0084
Persistent LI + LW222 - I CNN	1.492 ± 0.0058	1.113 ± 0.0058	1.496 ± 0.0061	1.109 ± 0.0082
LI + LW PDCNN *	1.677 ± 0.0054	1.254 ± 0.0086	1.675 ± 0.0138	1.251 ± 0.0115
LI + LW PSTN *	1.715 ± 0.0183	1.278 ± 0.0177	1.712 ± 0.0232	1.262 ± 0.0139

Ačiū už dėmesį!

Klausimai?

Daugiau detalių: <https://doi.org/10.3390/en13133440>
<https://mantas.info/>