

Meeting 6 - ResNet DenseNet arhitektūras

Code review

1. Centsties neizmantot [[]]

```
def get_confusion_matrix(expected, predicted) -> np.ndarray:
    matrix = np.zeros(shape=(3, 3), dtype=np.int)

    for expected_item, predicted_item in zip(expected, predicted):
        matrix[expected_item][predicted_item] += 1

    return matrix
```

2. Bug ar Kernel vienādojumu (nevar batch dimensiju iekļ)

```
for j in range(0, x_padded_size - self.kernel_size, self.stride):
    x_part = x_padded[:, :, i:i+self.kernel_size, j:j+self.kernel_size]
    # -1 means self.kernel_size*self.kernel_size*self.in_channels
    x_part = x_part.reshape(batch_size, -1)

    out_part = (K.t() @ x_part.unsqueeze(dim=-1)).squeeze(dim=-1)
    out[:, :, i_out, j_out] = out_part

    j_out += 1

    i_out += 1
```

3. Pārtaisīt train / test dataset split

```
def main():
    data = fetch_lfw_people(data_home='../data', color=False, resize=0.112, slice=None)
    idx_split = int(len(data.images) * 0.8)
    feature_count = max(data.target) + 1

    model = Model(feature_count)
    model.to(DEVICE)
    optimizer = torch.optim.Adam(model.parameters(), lr=LEARNING_RATE)
    metrics = {}

    for stage in ['train', 'test']:
        for metric in ['loss', 'acc']:
            metrics[f'{stage}_{metric}'] = []

    data_loader_train = torch.utils.data.DataLoader(
        dataset=DatasetLFWPeople(data.images[:idx_split], data.target[:idx_split], feature_count),
        batch_size=BATCH_SIZE,
        shuffle=True
    )

    data_loader_test = torch.utils.data.DataLoader(
        dataset=DatasetLFWPeople(data.images[idx_split:], data.target[idx_split:], feature_count),
        batch_size=BATCH_SIZE,
        shuffle=False
    )
```

**dataset_full = DatasetLFW() # pašā constructor
lejupielādē visu dataset**

**dataset_train, dataset_test =
torch.utils.data.random_split(dataset_full, (idxes_train,
idxes_test))**

4. Min faces per person, lai nebūtu tā, ka tikai 3 paraugi sejai

```
def main():
    data = fetch_lfw_people(data_home='../data', color=False, resize=0.112, slice_=None, min_faces_per_person=100)
    idx_split = int(len(data.images) * 0.8)
    feature_count = max(data.target) + 1
```

5. Neizmantoj set_default_dtype! float32 !

```
from sklearn.datasets import fetch_lfw_people
from torch.nn import ReLU, Linear, Parameter

#torch.set_default_dtype(torch.float32)
plt.rcParams['figure.figsize'] = (10, 10)

has_cuda = torch.cuda.is_available()
```

6. Implementēt datu standartizāciju un izveidot modeli, kurš strādā ar pilna izmēra

```
def main():
    data = fetch_lfw_people(data_home='../data', color=False, resize=0.112, slice_=None, min_faces_per_person=100)
    images = data.images
    images = (images - np.mean(images, axis=0)) / np.std(images, axis=0)
    idx_split = int(len(images) * 0.8)
    feature_count = max(data.target) + 1

    model = Model(feature_count)
    model.to(DEVICE)
```

TODO

Video:

<https://youtu.be/y9CcSpH1Db4>

Jamboard:

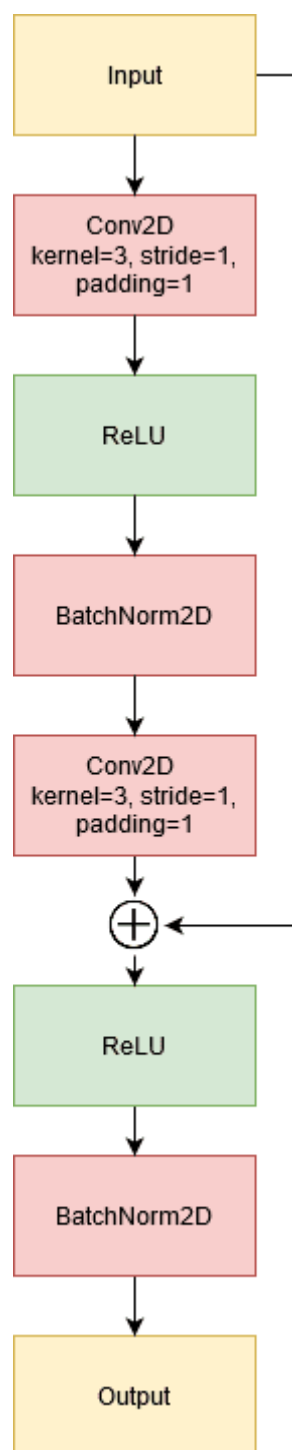
https://jamboard.google.com/d/1T0k_rnWjj5Flyd0BzJ3-At0QyOth78lfTSkKMdS1Zck/edit?usp=sharing

1. Uzdevums - Implementēt ResBlock pēc shēmas

Template:

http://share.yellowrobot.xyz/1630528570-intro-course-2021-q4/8_2_resnet_densenet_template.py

Schema:

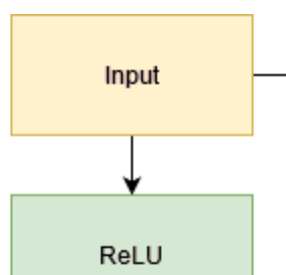


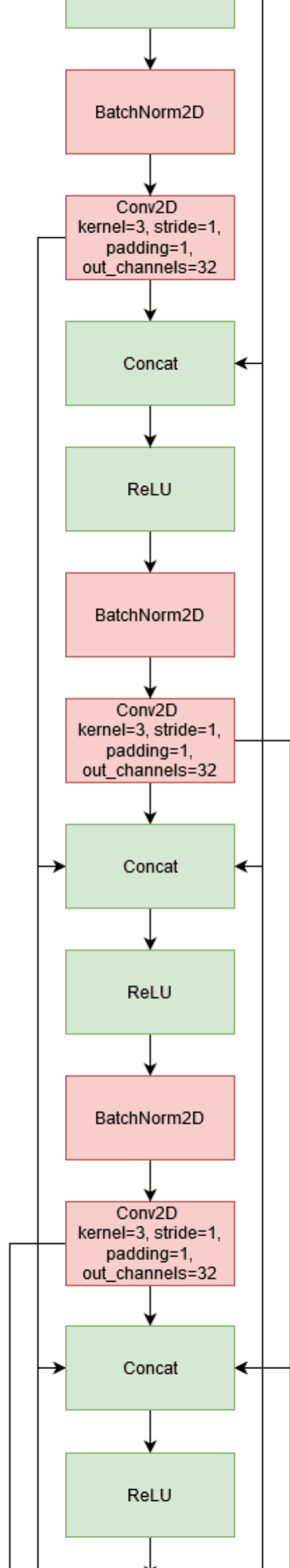
2. Uzdevums - Implementēt DenseBlock pēc shēmas

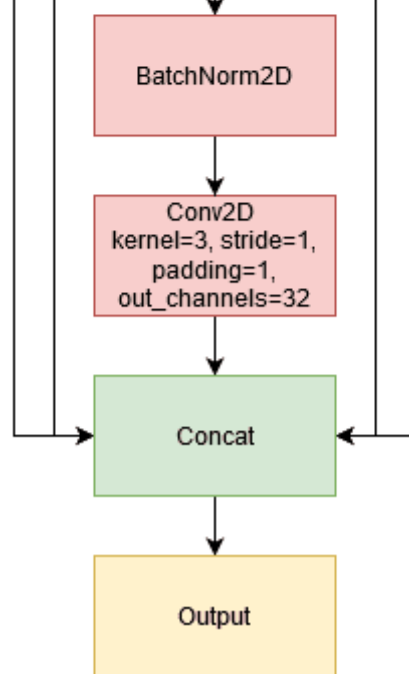
<http://share.yellowrobot.xyz/1628158950-vea-rtu-course-2020-q1/DenseBlock.png>

concat torch pa channel dimensiju:

```
out = torch.cat([x, conv1, conv2], dim=1)
```

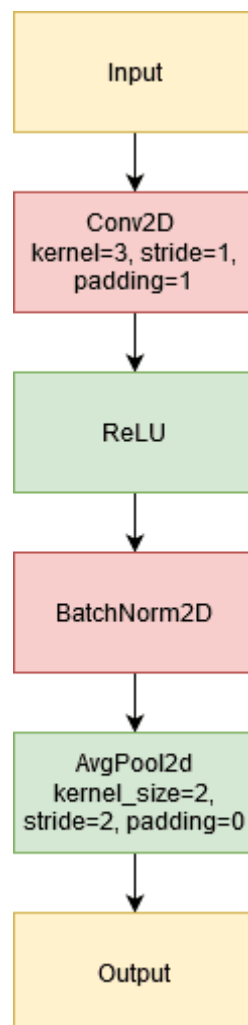






3. Uzdevums - Implementēt Transition Layer pēc shēmas

http://share.yellowrobot.xyz/1628158950-vea-rtu-course-2020-q1/TransitionLayer_DenseNet.png



4. Implementēt InceptionNet pēc shēmas

1. Implementēt InceptionBlockA pēc shēmas

<http://share.yellowrobot.xyz/1628158950-vea-rtu-course-2020-q1/InceptionBlockA.png>

2. Implementēt InceptionNet pēc shēmas

<http://share.yellowrobot.xyz/1628158950-vea-rtu-course-2020-q1/InceptionNet.png>

Gan InceptionBlockA, gan InceptionNet modeļu shēmas nav norādīts channel count katram slānim, to var brīvi izvēlēties. InceptionBlockA visiem zariem channel count nav jāsakrīt, var eksperimentēt katram liekot savu vērtību.

Papildus uzdevumi:

1. Izveidot InceptionBlock bottleneck versiju, lai var starp blokiem neizmantot maxpool layer.

2. Implementēt cita veida InceptionNet blokus:

<https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202>

<https://arxiv.org/pdf/1409.4842.pdf>

<https://arxiv.org/pdf/1512.00567.pdf>

Template:

http://share.yellowrobot.xyz/1630528570-intro-course-2021-q4/8_3_homework_inceptionnet_template.py

5. Implementēt savu dataset , izmantojot numpy mmap

Šim te dataset: <https://www.kaggle.com/hsankesara/flickr-image-dataset>

Sākotnēji, lai pre-processor uztaiša scale-and-crop, lai visi attēli vienādi, bet teorētiski varam arī padding un JSON glabāt arī katra attēla izmērus

1. Izveidot data pre-processor, kurš izveido mmap attēliem (N, 3, W, H) formātā un JSON failu kurā ir shape un labels (y) saglabāti <https://numpy.org/doc/stable/reference/generated/numpy.memmap.html>

2. DataSet klasē ielādēt šos datus, BET strādāt tikai ar mmap fp to pārvērst par tensor tikai uz getitem, lai netiktu nekas ierakstīts heap atmiņā

TODO rakstu darbi

⚠ Pievienot skaidrojumus un References no grāmatām (iedevu jau link)

BCE

Atpakaļizplatīšanās algoritms

Šis algoritms cenšas mainīt tīkla svarus un nobīdes tā, lai kļūda būtu 0.

$$\theta_0 := \theta_0 - \alpha \cdot \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$b := b - \alpha \cdot \frac{\partial}{\partial b} J(b, W)$$

$$\theta_1 := \theta_1 - \alpha \cdot \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

$$W := W - \alpha \cdot \frac{\partial}{\partial W} J(b, W)$$

**vienādojums modeļa
piemēram**

y_prim = lin(relu(lin(x)))

Kur, piemēram, $J(\theta_0, \theta_1) = L_{MSE} = \frac{1}{N} \sum_{i=0}^N (h_{\theta}(x_i) - y_i)^2$ ir MSE kļūdas funkcija un α ir apmācības

koeficients, kurš noteic, cik strauji svārs W un nobīde b tiek mainīti. Kļūdas funkcijas atvasinājums noteic vai parametru ir jāpalielina, vai jāsamazina un uz kādu lielumu.

