**RWTH Aachen University**
Institute for Data Science in
Mechanical Engineering

# Bonus Points Assignment 2 Changelog

**Prof. Sebastian Trimpe**

Winter 2021/22

Version 2.1.1

# Version 2.1.0

## 1 Topic PDF File

### 1.1 Section 1.2

**Wrong** Remember our convention that $x$ is of shape $(B, w, h, F)$...

**Right** Remember our convention that $x$ is of shape $(B, h, w, F)$...

## 2 Notebook "CNN with NumPy"

### 2.1 Unit Tests

Add unit tests for most functions and classes. These tests are *different* than the ones that will be used for grading.

### 2.2 Documentations

Add documentation for the functions `add_padding`, `convolve`, and `dilate`.

### 2.3 Class "Layer"

Correct the documentation.

**Wrong** Shape is (batch_size, num_inputs) or Shape is (batch_size, num_outputs).

**Right** Shape is (batch_size, h_in, w_in, F_in) or Shape is (batch_size, h_out, w_out, F_out), respectively.

### 2.4 Class "MaxPooling"

Change $K < S$ into $S < K$ in the exception.

**Wrong** if $K < S$: and This implementation of pooling only supports disjoint pooling windows ($K \geq S$) for backpropagation.

**Right** if $S < K$: and This implementation of pooling only supports disjoint pooling windows ($S \geq K$) for backpropagation, respectively.

### 2.5 Class "FullyConnected"

Add an attribute `self.input_shape` in the constructor, and store the shape of `x` in it in the `forward` method.

### 2.6 Class "Sigmoid"

Fix the class definition and implement the `update` method.

**Wrong** def Sigmoid(Layer) and def __forward__(self, x).

**Right** class Sigmoid(Layer) and def forward(self, x), respectively.

### 2.7 Class "FeedForwardNet"

Fix the `training_step` method by dividing the gradient with the batch size instead of an undefined `N`, and reshape the target with the prediction's shape instead of the input's

**Wrong** target = target.reshape(x.shape) and gradient /= N

**Right** target = target.reshape(pred.shape) and gradient /= x.shape[0]

## 3 Notebook "Casting Classification"

### 3.1 PyTorch version

This notebook was developed using an outdated version of PyTorch and TorchVision; namely, PyTorch 1.7.0 and TorchVision 0.8.1. You need to use these two versions for the assignment. We recommend three options, depending on your individual case:

1. We provide a `yaml` file alongside the release 2.1.0 of the assignment. You can create an environment with the correct versions using this file with the command

   ```
   conda create --name [NAME] --file [YAML FILE]
   ```

   **This file was created for MacOS; it may not work as expected on other platforms**;

2. If the yaml file does not work for you, you can create a virtual environment and install the required version with

   ```
   conda install pytorch==1.7.0 torchvision==0.8.1 -c pytorch
   ```

3. On the RWTH JupyterHub, the profile "[CSME2] Computer Science in Mechanical Engineering II" has the correct versions installed; you can use that to run the notebooks without any installations.

### 3.2 Title

Change the title.

### 3.3 Imports Cell

Added and fixed imports, and set default data type to `float32`.

### 3.4 Function "train"

#### Fix typos

**Wrong** criterion = toch.nn.BCELoss() and optimizer = optim.Adam(net.parameters(), lr=learning_rate).

**Right** criterion = torch.nn.BCELoss() and optimizer = torch.optim.Adam(net.parameters(), lr=learning_rate).

#### Tensor casting and prediction shape

**Wrong** outputs = net(inputs) and loss = criterion(outputs, labels).

**Right** outputs = net(inputs.to(torch.float32)), outputs = outputs.reshape(labels.shape), and loss = criterion(outputs, labels.to(torch.float32)).

#### Number of parameters

Compute and display the number of trainable parameters.

# Version 2.1.1

## 4 Notebook "CNN with NumPy"

### 4.1 Class "Conv2D"

Fix unit test of backward pass. In particular, the gradient returned by `backward` should have the same shape as the input to the layer. In the previous version of the test, the gradient had size $(B, h_{\text{in}} + 2P, w_{\text{in}} + 2P, F_i n)$, because it considered the padding neurons (which it should not).
Add unit test for `update` function.

## 5 Notebook "Casting Classification"

### 5.1 Function "train"

Add indications that the labels should be transformed to be understandable by the loss function.