This repository contains implementation of Siamese Neural Networks in Tensorflow built based on 3 different and major deep learning architectures:

Convolutional Neural Networks

Recurrent Neural Networks

Multihead Attention Networks

The main reason of creating this repository is to compare well-known implementaions of Siamese Neural Networks available on GitHub mainly built upon CNN and RNN architectures with Siamese Neural Network built based on multihead attention mechanism originally proposed in Transformer model from [Attention is all you need](https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf) paper.

**Supported datasets**

Current version of pipeline supports working with **3** datasets:

[The Stanford Natural Language Inference (SNLI) Corpus](https://nlp.stanford.edu/projects/snli/)

[Quora Question Pairs](https://www.kaggle.com/c/quora-question-pairs)

🆕 Adversarial Natural Language Inference (ANLI) benchmark: [GitHub](https://github.com/facebookresearch/anli/), [arXiv](https://arxiv.org/pdf/1910.14599.pdf)

**Installation**

**Data preparation**

In order to download data, execute the following commands (this process can take a while depending on your network throughput):

cd bin

chmod a+x prepare\_data.sh

./prepare\_data.sh

As result of executing above script, **corpora** directory will be created with **QQP**, **SNLI** and **ANLI** data.

**Dependency installation**

This project was developed in and has been tested on **Python 3.6**. The package requirements are stored in **requirements** folder.

To install the requirements, execute the following command:

For **GPU** usage, execute:

pip install -r requirements/requirements-gpu.txt

and for **CPU** usage:

pip install -r requirements/requirements-cpu.txt

**Training models**

To train model run the following command:

python3 run.py train SELECTED\_MODEL SELECTED\_DATASET --experiment\_name NAME --gpu GPU\_NUMBER

where **SELECTED\_MODEL** represents one of the selected model among:

cnn

rnn

multihead

and **SELECTED\_DATASET** is represented by:

SNLI

QQP

ANLI

**--experiment\_name** is an optional argument used for indicating experiment name. Default value **{SELECTED\_MODEL}\_{EMBEDDING\_SIZE}**.

**--gpu** is an optional argument, use it in order to indicate specific GPU on your machine (the default value is '0').

Example (GPU usage): Run the following command to train Siamese Neural Network based on CNN and trained on SNLI corpus:

python3 run.py train cnn SNLI --gpu 1

Example (CPU usage): Run the following command to train Siamese Neural Network based on CNN:

python3 run.py train cnn SNLI

**Training configuration**

This repository contains main configuration training file placed in **'config/main.ini'**.

[TRAINING]

num\_epochs = 10

batch\_size = 512

eval\_every = 20

learning\_rate = 0.001

checkpoints\_to\_keep = 5

save\_every = 100

log\_device\_placement = false

[DATA]

logs\_path = logs

model\_dir = model\_dir

[PARAMS]

embedding\_size = 64

loss\_function = mse

**Model configuration**

Additionally, each model contains its own specific configuration file in which changing hyperparameters is possible.

**Multihead Attention Network configuration file**

[PARAMS]

num\_blocks = 2

num\_heads = 8

use\_residual = False

dropout\_rate = 0.0

**Convolutional Neural Network configuration file**

[PARAMS]

num\_filters = 50,50,50

filter\_sizes = 2,3,4

dropout\_rate = 0.0

**Recurrent Neural Network configuration file**

[PARAMS]

hidden\_size = 128

cell\_type = GRU

bidirectional = True

**Training models with GPU support on Google Colaboratory**

If you don't have an access to workstation with GPU, you can use the below exemplary Google Colaboratory notebook for training your models (CNN, RNN or Multihead) on SNLI or QQP datasets with usage of **NVIDIA Tesla T4 16GB GPU** available within Google Colaboratory backend: [Multihead Siamese Nets in Google Colab](https://colab.research.google.com/drive/1FUEBV1JkQpF2iwFSDW338nAUhzPVZWAa)

**Testing models**

Download pretrained models from the following link: [pretrained Siamese Nets models](https://drive.google.com/file/d/1STgv1hIxdVpKLQ6-EZK7J3C4ZtfZgbkS/view?usp=sharing), unzip and put them into **./model\_dir** directory. After that, you can test models either using predict mode of pipeline:

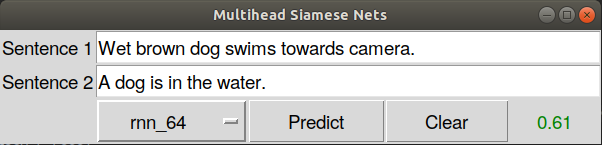
python3 run.py predict cnn

or using GUI demo:

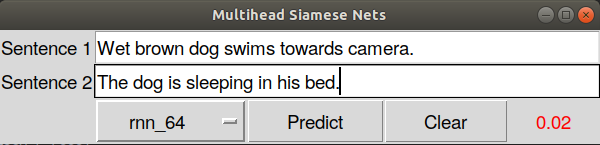
python3 gui\_demo.py

The below pictures presents Multihead Siamese Nets GUI for:

Positive example:

[](https://github.com/tlatkowski/multihead-siamese-nets/blob/master/pics/positive_sample.png)

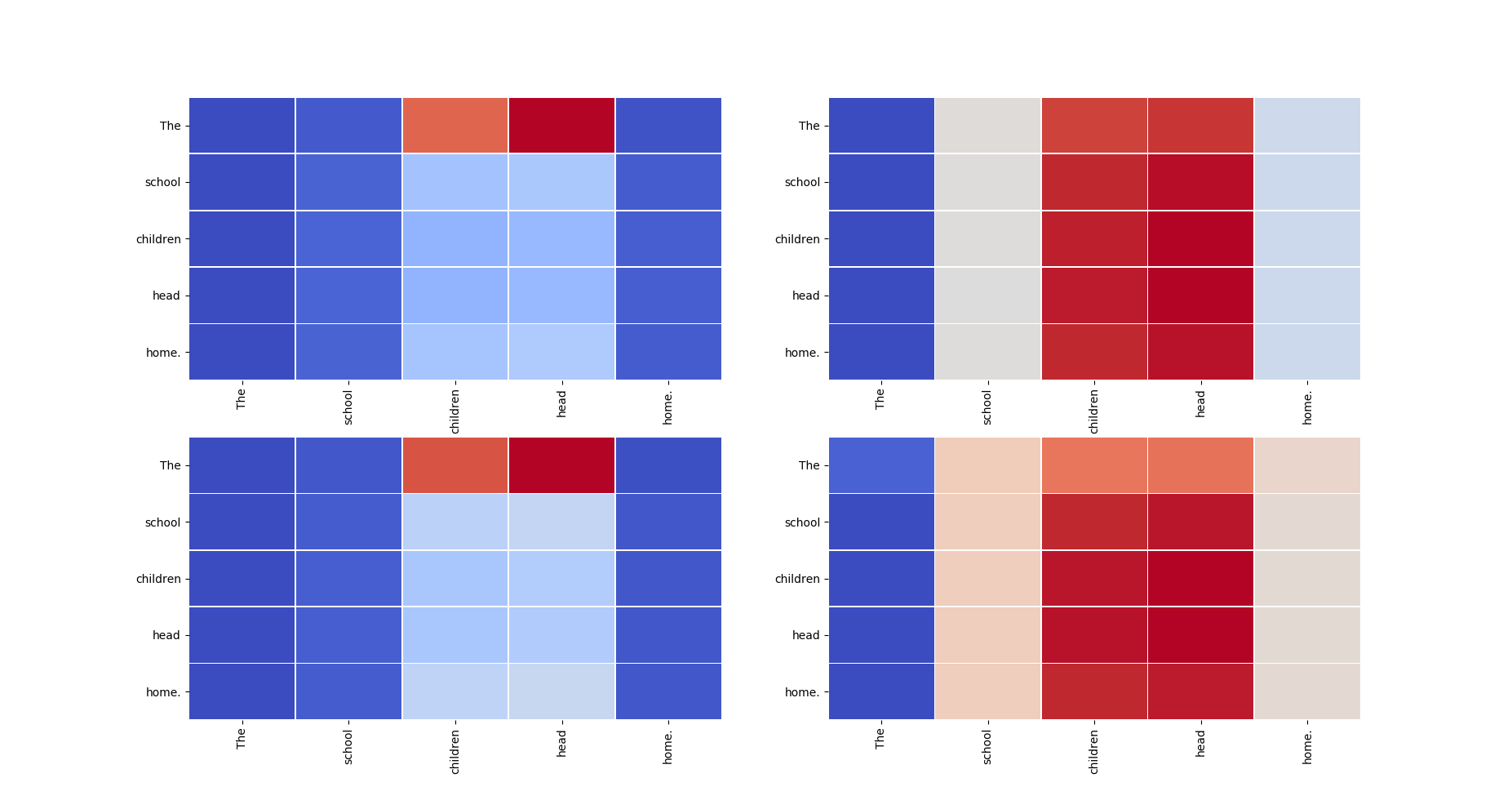
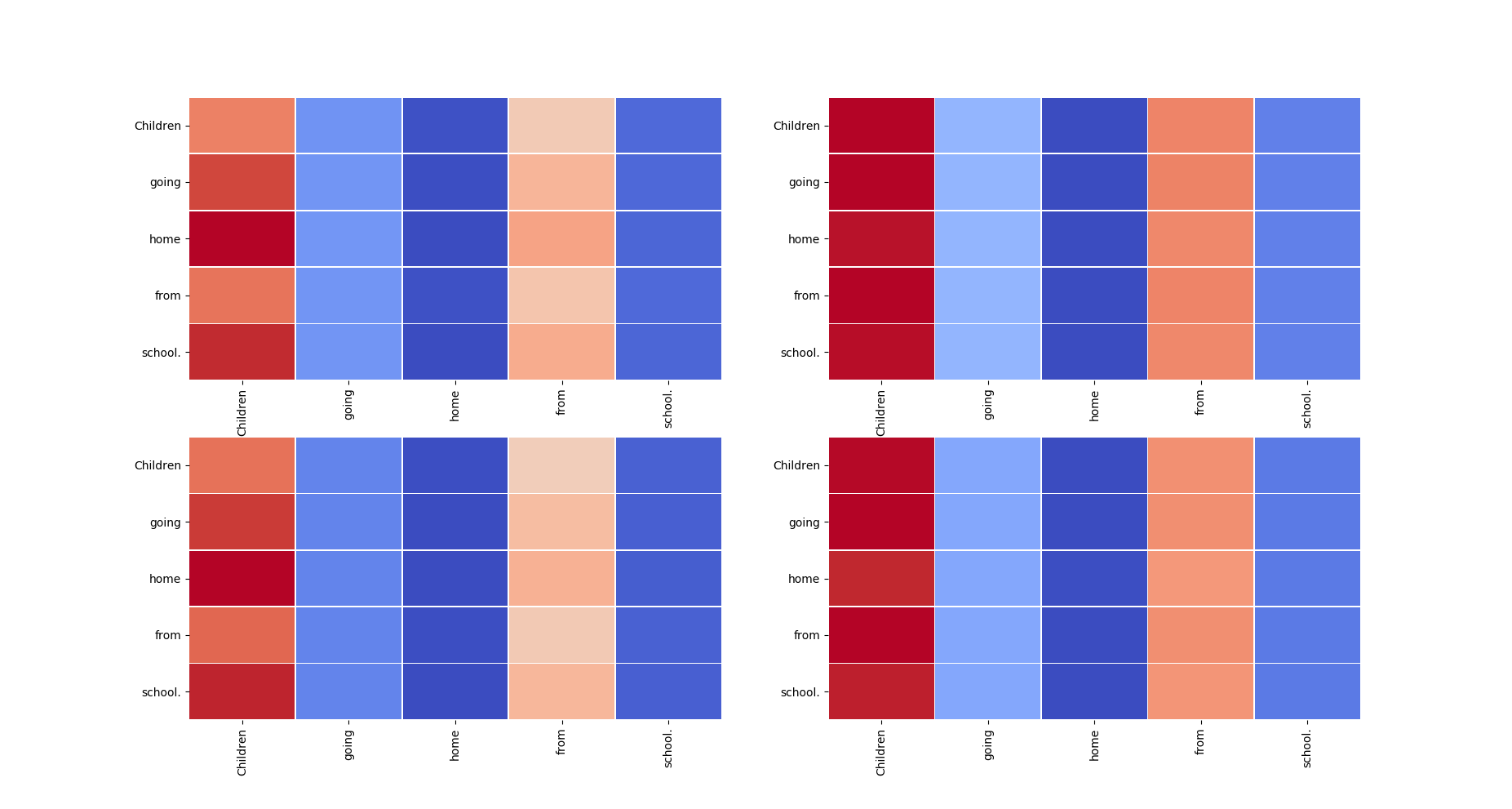
Negative example:

[](https://github.com/tlatkowski/multihead-siamese-nets/blob/master/pics/negative_sample.png)

**Attention weights visualization**

In order to visualize multihead attention weights for compared sentences use GUI demo - check 'Visualize attention weights' checkbox which is visible after choosing model based on multihead attention mechanism.

The example of attention weights visualization looks as follows (4 attention heads):

[](https://github.com/tlatkowski/multihead-siamese-nets/blob/master/pics/attention1.png) [](https://github.com/tlatkowski/multihead-siamese-nets/blob/master/pics/attention2.png)

**Comparison of models**

Experiments performed on GPU **Nvidia GeForce GTX 1080Ti**.

**> SNLI dataset.**

Experiment parameters:

Number of epochs : 10

Batch size : 512

Learning rate : 0.001

Number of training instances : 326959

Number of dev instances : 3674

Number of test instances : 36736

Embedding size : 64

Loss function: mean squared error (MSE)

Specific hyperparameters of models:

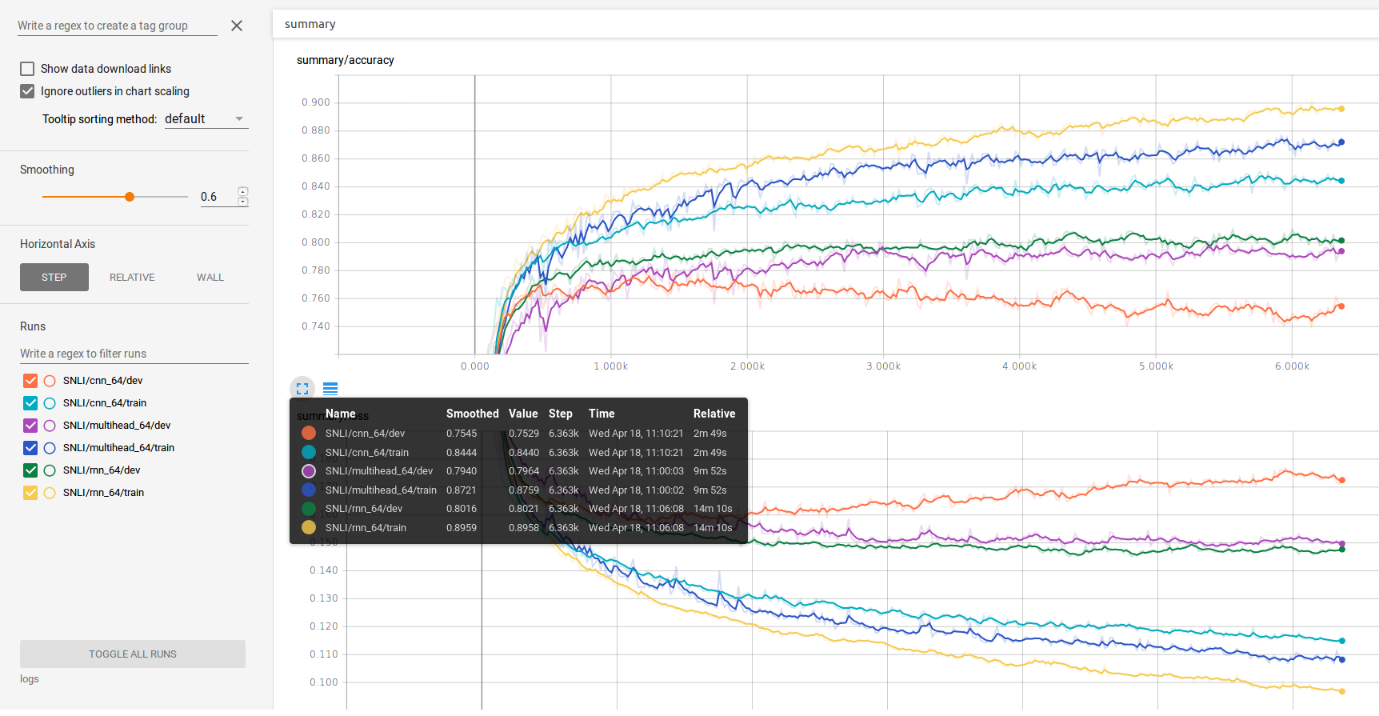
| **CNN** | **RNN** | **Multihead** |
| --- | --- | --- |
| num\_filters = 50,50,50 | hidden\_size = 128 | num\_blocks = 2 |
| filter\_sizes = 2,3,4 | cell\_type = GRU | num\_heads = 8 |
|  | bidirectional = True | use\_residual = False |
|  |  | layers\_normalization = False |

Evaluation results:

| **Model** | **Mean-Dev-Acc\*** | **Last-Dev-Acc\*\*** | **Test-Acc** | **Epoch Time** |
| --- | --- | --- | --- | --- |
| CNN | 76.51 | 75.08 | 75.40 | 15.97s |
| bi-RNN | 79.36 | 79.52 | 79.56 | 1 min 22.95s |
| Multihead | 78.52 | 79.61 | 78.29 | 1 min 00.24s |

\*Mean-Dev-Acc: the mean development set accuaracy over all epochs.

\*\*Last-Dev-Acc: the development set accuaracy for the last epoch.

Training curves (Accuracy & Loss): [](https://github.com/tlatkowski/multihead-siamese-nets/blob/master/pics/snli_train_curves.png)

**> QQP dataset.**

Experiment parameters:

Number of epochs : 10

Batch size : 512

Learning rate : 0.001

Number of training instances : 362646

Number of dev instances : 1213

Number of test instances : 40428

Embedding size : 64

Loss function: mean squared error (MSE)

Specific hyperparameters of models:

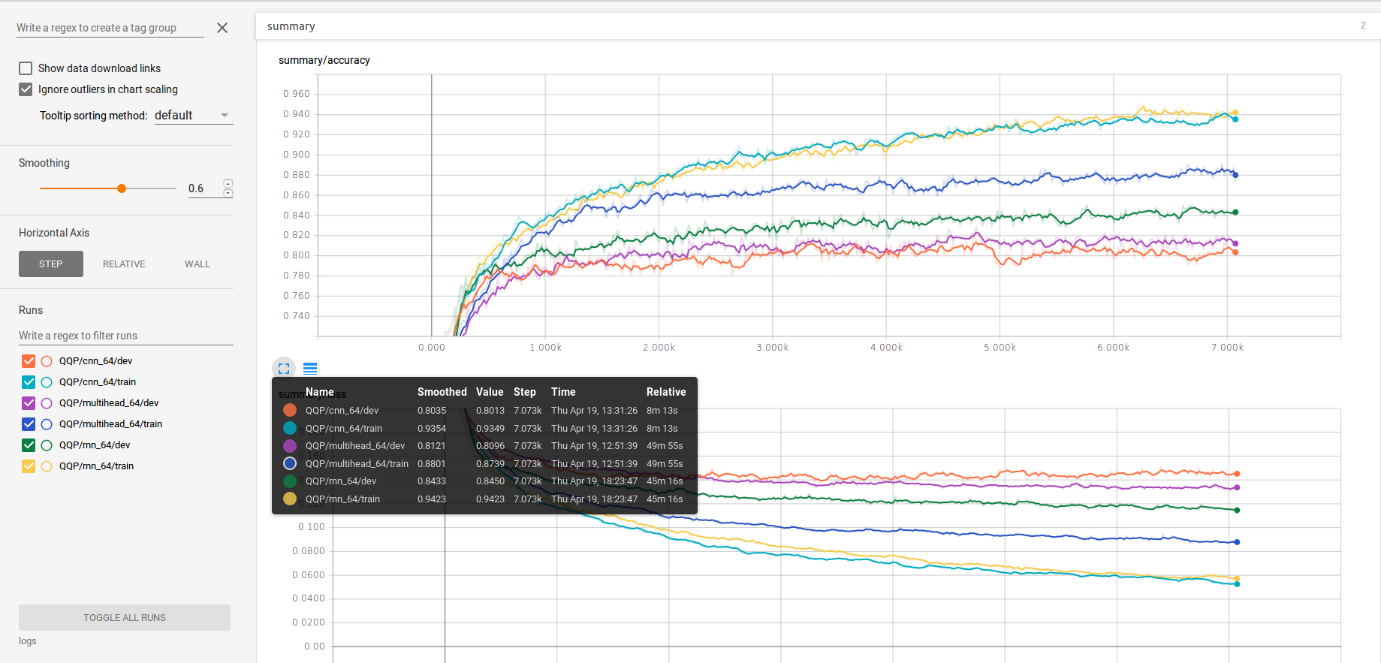
| **CNN** | **RNN** | **Multihead** |
| --- | --- | --- |
| num\_filters = 50,50,50 | hidden\_size = 128 | num\_blocks = 2 |
| filter\_sizes = 2,3,4 | cell\_type = GRU | num\_heads = 8 |
|  | bidirectional = True | use\_residual = False |
|  |  | layers\_normalization = False |

Evaluation results:

| **Model** | **Mean-Dev-Acc\*** | **Last-Dev-Acc\*\*** | **Test-Acc** | **Epoch Time** |
| --- | --- | --- | --- | --- |
| CNN | 79.74 | 80.83 | 80.90 | 49.84s |
| bi-RNN | 82.68 | 83.66 | 83.30 | 4 min 26.91s |
| Multihead | 80.75 | 81.74 | 80.99 | 4 min 58.58s |

\*Mean-Dev-Acc: the mean development set accuracy over all epochs.

\*\*Last-Dev-Acc: the development set accuracy for the last epoch.

Training curves (Accuracy & Loss): [](https://github.com/tlatkowski/multihead-siamese-nets/blob/master/pics/qqp_train_curves.png)