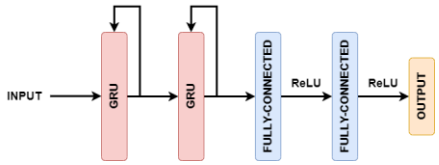
Assignment 3: Recurrent neural networks *(approximate training run-time on a CPU: 1 hour)*

Implement a recurrent neural network that solves the speech classification task. This neural network must have two gated recurrent unit (GRU) layers with 64-dimensional hidden states followed by two fully-connected layers with 128 neurons each. Apply a rectified linear unit (ReLU) activation function at the output of each fully-connected layer. Furthermore, bear in mind that each speech feature matrix represents a temporal sequence of T=101 frames. Hence, for classification, you only have to pass the hidden state of the second GRU layer at time T to the first fully-connected layer.



Once again, consider a minibatch size of 1,024 samples and, as the optimizer, employ Adam with default parameters. Train your model for 50 epochs. Use "torch.manual_seed(0)" to make your results reproducible and be patient… backpropagation through time (BPTT) can be computationally expensive!

1) Plot the training and validation losses, as well as the training and validation accuracies, as a function of the training iteration.
2) What is the test accuracy?
3) Calculate, by hand, the total number of parameters of the model, and indicate, step by step, how you reached your solution.
4) Very briefly, compare, in terms of performance and computational complexity, this model with the models that you implemented in the previous assignments.

Hints:

- Do not forget to normalize your speech feature matrices, e.g., `X_train ← (X_train - np.mean(X_train)) / np.std(X_train)`
- Shuffle your training data before model training.
- Accuracy is defined as the ratio between the number of correct classifications and the total number of classifications.
- Training on a CPU takes time. To check more quickly that your implementation is correct, you may want to train your model for 5 epochs. In this case, test accuracy should be *around* 64%.
- To check that you correctly calculated by hand the number of parameters of your model, you may want to use the following snip of code:

```
def get_no_params(model):
    nop = 0
    for param in list(model.parameters()):
        nn = 1
        for s in list(param.size()):
            nn = nn * s
        nop += nn
    return nop
```