A Parameter Initialization Derivation

Following the derivation of Glorot and Kaiming initialization (Glorot and Bengio, 2010; He et al., 2015), we assume the values of each input vector $\mathbf{x}_t$ are i.i.d with zero mean and a small variance:

$$\mathrm{E}[x_{t,i}] = 0, \quad \mathrm{Var}[x_{t,i}] \ll 1 \quad \forall 1 \leq i \leq d \ .$$

We initialize each weight matrix with zero mean and a variance of $1/d$. After a matrix multiplication $\mathbf{y} = \mathbf{W}\mathbf{x}_t$, each value y_i would have

$$\mathrm{E}[y_i] = \mathrm{E}[\sum_j w_{i,j}\mathbf{x}_{t,j}] = 0$$
$$\mathrm{Var}[y_i] = \sum_j \mathrm{Var}[w_{i,j}] \cdot \mathrm{Var}[x_{t,j}] = \mathrm{Var}[x] \ ,$$

which means the scale of the values after matrix multiplication remains the same.

A.1 Computing Var[$\mathbf{c}_t$]

Let $f_{t,i}$ be the i-th entry of the forget gate $\mathbf{f}_t$:

$$f_{t,i} = \sigma(\mathbf{w}_{f,i}^\top \mathbf{x}_t + v_{f,i}\, c_{t-1,i} + b_{f,i}) \ .$$

The pre-activation value will be sufficiently close to 0 because the parameters are initialized with zero mean and small variance and the bias value is initially 0. As a result,

$$E[f_{t,i}] = \sigma(0) = 0.5 \ .$$

The state value $c_{t,i}$ is computed according to

$$c_{t,i} = f_{t,i} \cdot c_{t-1,i} + (1 - f_{t,i}) \cdot (\mathbf{w}_i^\top \mathbf{x}_t) \ .$$

Substituting the expectation of $f_{t,i}$ in, we get:[6]

$$c_{t,i} = \mathbf{w}_i^\top \left(\frac{\mathbf{x}_t}{2} + \frac{\mathbf{x}_{t-1}}{4} + \frac{\mathbf{x}_{t-2}}{8} + \cdots \right) \ .$$

Therefore, $\mathrm{E}[c_{t,i}] = 0$ as $\mathrm{E}[\mathbf{w}^\top \mathbf{x}] = 0$. The variance of $c_{t,i}$ however depends on the correlation between input vectors. When the input vectors are independent:

$$\mathrm{Var}[c_{t,i}] = \mathrm{Var}[\mathbf{w}_i^\top \mathbf{x}] \left(\frac{1}{2^2} + \frac{1}{4^2} + \frac{1}{8^2} + \cdots \right)$$
$$\approx \mathrm{Var}[\mathbf{w}_i^\top \mathbf{x}] \cdot \frac{1}{3} = \mathrm{Var}[x]/3 \ .$$

However, the two vectors in the input sequence, for instance $\mathbf{x}_t$ and $\mathbf{x}'_t$, are not necessarily independent, for example because two words in an input sentence are often correlated. When the input vectors are perfectly correlated $\mathbf{x}_t = \mathbf{x}_{t-1} = \cdots = \mathbf{x}$, on the other hand,

$$\mathrm{Var}[c_{t,i}] = \mathrm{Var}[\mathbf{w}_i^\top \mathbf{x}] = \mathrm{Var}[x] \ .$$

In practice, multiple SRU layers are stacked to construct a deep network. The internal state $\mathbf{c}_t$ and $\mathbf{h}_t$ would be a weighted combination of inputs $\{\mathbf{x}_1 \cdots \mathbf{x}_t\}$, which will increase the correlation of the state vectors at different steps. These state vectors are again fed into the next layer, and keep increasing the correlation. As a result, we expect the actual ratio between the variance of $\mathbf{c}_t$ and that of the input of the current layer $\mathbf{x}_t$ lies between the two derived values,

$$\frac{1}{3} \leq \frac{\mathrm{Var}[c]}{\mathrm{Var}[x]} \leq 1 \ , \tag{5}$$

and would finally converge to the upper bound value of 1. Figure 6 confirms our expectation by computing the empirical value of $\mathrm{Var}[c]/\mathrm{Var}[x]$ in deep SRU networks.

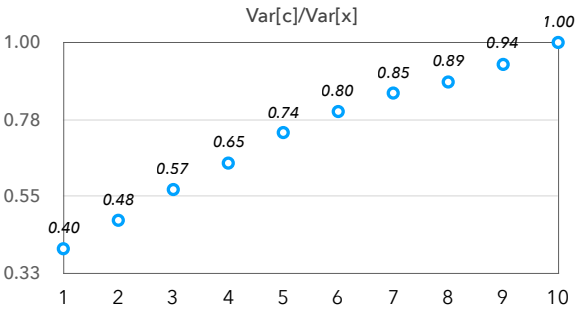

Figure 6: Empirical estimation of the variance ratio $\mathrm{Var}[\mathbf{c}_t]/\mathrm{Var}[\mathbf{x}_t]$ at each layer in a randomly initialized SRU model. We use the pre-trained word2vec embeddings as input, resulting an initial ratio slightly higher than $1/3$. As expected, the ratio increases to 1 in deep layers.

A.2 Computing Var[$\mathbf{h}_t$]

Given the result in Equation (5), we proceed to compute $\mathrm{Var}[\mathbf{h}_t]$. The i-th entry of $\mathbf{h}_t$ is similarly computed as

$$h_{t,i} = r_{t,i} \cdot c_{t,i} + (1 - r_{t,i}) \cdot x_{t,i}$$
$$\text{where} \quad r_{t,i} = \sigma(\mathbf{w}_{r,i}^\top \mathbf{x}_t + v_{r,i} c_{t-1,i} + b_{r,i}) \ .$$

The highway reset gate is not necessarily initialized with a zero bias. Let the initial bias be b and $u = \mathbf{w}_{r,i}^\top \mathbf{x}_t + v_{r,i} c_{t-1,i}$ denote the rest of terms

[6] We are ignoring the correlation between $\mathbf{f}_{t,i}$ and $\mathbf{f}_{t,i'}$ here because their variance is small.

in the sigmoid function. We have $\mathrm{E}[u] = 0$ and $\mathrm{Var}[u] \ll 1$ because $\mathbf{x}_t$ and $\mathbf{c}_{t-1}$ have small variance.

We approximate the value of $r_{t,i}$ using its Taylor expansion at $u = 0$:

$$\begin{aligned} r_{t,i} &= \sigma(u+b) \\ &\approx \frac{e^b}{e^b+1} + \frac{e^b \cdot u}{(e^b+1)^2} \\ \mathrm{E}[r_{t,i}^2] &\approx \frac{e^{2b}}{(e^b+1)^2} + \frac{e^{2b} \cdot u^2}{(e^b+1)^4} \quad . \end{aligned}$$

We can ignore the term with u^2 since $\mathrm{Var}[u] \ll 1$, which gives us

$$\mathrm{E}[r_{t,i}^2] \approx \frac{e^{2b}}{(e^b+1)^2} \quad .$$

Substituting this result in $\mathrm{Var}[h_{t,i}]$,

$$\begin{aligned} \mathrm{Var}[h_{t,i}] &= \mathrm{E}\left[r_{t,i}^2 \mathbf{c}_{t,i}^2 + (1-r_{t,i})^2 x_{t,i}^2\right] \\ &= \frac{e^{2b} \cdot \mathrm{Var}[c]}{(e^b+1)^2} + \frac{\mathrm{Var}[x]}{(e^b+1)^2} \end{aligned} \tag{6}$$

Since from (5) we have $\mathrm{Var}[x]/3 \leq \mathrm{Var}[c] \leq \mathrm{Var}[x]$, we get the bound of $\mathrm{Var}[h_{t,i}]$

$$\frac{e^{2b}+3}{3(e^b+1)^2} \leq \frac{\mathrm{Var}[h]}{\mathrm{Var}[x]} \leq \frac{e^{2b}+1}{(e^b+1)^2}$$

which is equivalent to

$$\frac{1}{3} \leq \frac{\mathrm{Var}[h]}{\mathrm{Var}[x]} \leq \frac{1}{2}$$

when $b = 0$.

A.3 Computing the Scaling Constant α

Finally, we compute the scaling constant α (Section 3.2). Using the result in Equation (6), when α is introduced we get:

$$\begin{aligned} \mathrm{Var}[h_{t,i}] &= \frac{e^{2b} \cdot \mathrm{Var}[c]}{(e^b+1)^2} + \frac{\alpha^2 \cdot \mathrm{Var}[x]}{(e^b+1)^2} \\ &\approx \frac{e^{2b}+\alpha^2}{(e^b+1)^2} \cdot \mathrm{Var}[x] \quad , \end{aligned}$$

as $\mathrm{Var}[c] \to \mathrm{Var}[x]$ according to Equation (5) and the empirical evaluation (Figure 6). This implies $e^{2b} + \alpha = (1+e^b)^2$ if we want $\mathrm{Var}[h] \approx \mathrm{Var}[x]$. By solving for α we have

$$\alpha = \sqrt{1 + 2 \cdot e^b} \quad ,$$

and $\alpha = \sqrt{3}$ when $b = 0$.

B Experimental Details

We include additional experimental setup and results in this section.

B.1 Classification

The data and pre-processing code are obtained from the code repository of Harvard NLP.[7]

We use a batch size of 32 and a dropout probability of 0.5 for all models. In addition, we increment the dropout to 0.55 or 0.6 for the 8-layer SRU model. Following the implementation of (Kim, 2014), out-of-vocabulary words that are not in the pre-trained embeddings are initialized with random vectors with values from $[-0.25, 0.25]$.

B.2 Question Answering

We use a word embedding dropout of 0.5 and a recurrent dropout of 0.2. In the setup of Chen et al. (2017a), the bi-LSTM models concatenates the output of each layer and feed it to subsequent layers. This helps the gradient propagation and improves the final performance. With highway connection, this is no longer necessary. In SRU and Q-RNN (with highway), only the output of the last layer is given to subsequent layers.

B.3 Machine Translation

We use the OpenNMT PyTorch implementation for the translation experiments.Table 6 shows the list of configuration options used for training. For evaluation, we use beam size 5 and length penalty 0.6.

-layers	4 to 6	-share_embedding	
-rnn_size	512	-position_encoding	
-word_vec_size	512	-param_init 0	
-batch_type	tokens	-max_grad_norm 0	
-normalization	tokens	-dropout 0.1	
-batch_size	5120	-label_smoothing 0.1	
-accum_count	5	-epoch 40	
-optim	adam	-param_init_glorot	
-learning_rate	2		
-adam_beta2	0.998		
-decay_method	noam		
-warmup_steps	16000		

Table 6: Translation training configuration.

[7] https://github.com/harvardnlp/sent-conv-torch

Epoch	Transformer base		w/ SRU (4 layer)		w/ SRU (5 layer)	
	Valid	Test	Valid	Test	Valid	Test
20	26.1	27.3	26.2	27.6	26.6	27.9
21	26.2	27.3	26.3	27.7	26.6	28.1
22	26.1	27.4	26.3	27.8	26.7	28.0
23	26.2	27.4	26.4	27.7	26.8	28.1
24	26.2	27.4	26.4	27.8	26.7	28.0
25	26.3	27.4	26.4	27.7	26.6	28.1
26	26.5	27.5	26.5	27.7	26.7	28.1
27	26.4	27.6	26.4	27.6	26.8	28.1
28	26.4	27.6	26.4	27.7	26.7	28.2
29	26.4	27.5	26.4	27.8	26.8	28.2
30	26.5	27.7	26.4	27.8	26.9	28.1
31	26.4	27.6	26.6	27.7	26.9	28.3
32	26.5	27.5	26.5	27.8	26.9	28.3
33	26.5	27.5	26.5	27.8	27.1	28.3
34	26.4	27.6	26.5	27.9	26.9	28.2
35	26.4	27.6	26.5	27.9	26.9	28.2
36	26.5	27.6	26.5	27.8	26.9	28.3
37	26.5	27.5	26.5	27.8	26.9	28.2
38	26.5	27.6	26.5	28.0	27.0	28.2
39	26.5	27.6	26.7	27.8	27.0	28.2
40	26.6	27.6	26.6	27.9	27.0	28.2

Table 7: Average BLEU scores after each epoch.

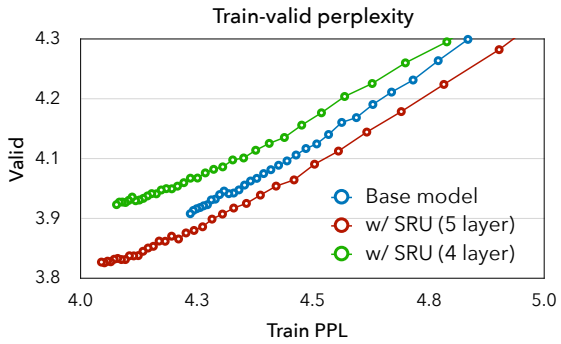

Figure 7: Training and validation perplexity curves of the base model and two SRU models.

Table 7 shows the averaged BLEU score of each model from 20th to 40th epoch. The improvement over the Transformer base model is consistent across different epochs.

Figure 7 plots the training and validation perplexity of three models. With a higher dropout (0.2) used for the SRU, the 5-layer model gets consistent lower validation perplexity over the base model and the 4-layer model. We also see that models with SRU exhibit much faster training progress with much lower training perplexity, suggesting the models could be tuned better with further training regularization.

B.4 Character-level Language Modeling

We train all models using a weight decay of 10^{-7} and a gradient clipping of 0.3. We set the learning rate factor of Noam scheduling to 3 and the warmup steps to $32,000$ ($8,000$ for models without projection). We tune the dropout probability from $\{0.2, 0.3\}$.

The projection (bottleneck) trick is implemented as follows. Recall that the batched multiplication of SRU is computed as

$$\begin{pmatrix} \mathbf{W} \\ \mathbf{W}_f \\ \mathbf{W}_r \end{pmatrix} [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L] \ .$$

The stacked parameter matrices on the left is reparameterized by a low-rank factorization,

$$\begin{pmatrix} \mathbf{W} \\ \mathbf{W}_f \\ \mathbf{W}_r \end{pmatrix} = \mathbf{P}^\top \mathbf{Q} \ ,$$

where $\mathbf{Q} \in \mathbb{R}^{d_{in} \times d'}$ and $\mathbf{P} \in \mathbb{R}^{3d_{out} \times d'}$ are two new parameter matrices to be learned, and d' is the projection dimension that is much smaller than the input and output dimension of the SRU.