An exploration of dropout with LSTMs

Gaofeng Cheng\textsuperscript{1,3}, Vijayaditya Peddinti\textsuperscript{4,5}, Daniel Povey\textsuperscript{4,5}, Vimal Manohar\textsuperscript{4,5}, Sanjeev Khudanpur\textsuperscript{4,5}, Yonghong Yan\textsuperscript{1,2,3}

Abstract

Long Short-Term Memory networks (LSTMs) are a component of many state-of-the-art DNN-based speech recognition systems. Dropout is a popular method to improve generalization in DNN training. In this paper we describe extensive experiments in which we investigated the best way to combine dropout with LSTMs—specifically, projected LSTMs (LSTMp). We investigated various locations in the LSTM to place the dropout (and various combinations of locations), and a variety of dropout schedules. Our optimized recipe gives consistent improvements in WER across a range of datasets, including Switchboard, TED-LIUM and AMI.

Index Terms: speech recognition, LSTM, DNN, dropout, lattice-free MMI

1. Introduction

The Long Short-Term Memory (LSTM) network\textsuperscript{[1, 2]} is used in many state-of-the-art ASR systems\textsuperscript{[3]}, often in the popular 'projected' variant\textsuperscript{[4]}.

Dropout\textsuperscript{[5]} is a mechanism to improve generalization of neural nets. It consists of multiplying neural net activations by random zero-one masks during training (random masks are not used in test time, but are approximated by a fixed scale). The dropout probability $p$ determines what proportion of the mask values are one; the original paper suggested that $p = 0.5$ works well for a range of tasks. However, the experience of the speech community is that dropout implemented in this way doesn’t usually work for speech recognition tasks, and at the very least requires careful selection of the dropout probability and the use of a time-varying schedule (as is typical for negative results, it is hard to supply a citation for this).

This paper describes the outcome of extensive experiments in which we investigated the best way to use dropout with LSTMs. The issues we investigated are:

- Various locations for dropout (e.g. different LSTM gates or combinations thereof).

The experiments reported in this paper were performed with systems based on lattice-free MMI (LF-MMI)\textsuperscript{[6]}. These systems use a frame shift of 30ms, as opposed to the 10ms of typical systems.

\textbf{1.1. Prior work on dropout for ASR}

A few publications have used dropout for large-scale Automatic Speech Recognition (ASR) tasks. In\textsuperscript{[7]}, dropout is applied to a network based on Maxout nonlinearities. The dropout probability started at 0.5 and linearly decreased to 0.0 after 8 epochs, after which no dropout was used. In\textsuperscript{[8]}, dropout was applied to LSTMs at the point where the input comes from the previous layer (this is equivalent to our "Location 2" below). In that case, no dropout schedules were used. Improvements in frame accuracy were shown on a Google-internal dataset of Icelandic speech, but no WER results.

\textbf{2. Prior work}

\textbf{2.1. Prior work on dropout for ASR}

A few publications have used dropout for large-scale Automatic Speech Recognition (ASR) tasks. In\textsuperscript{[7]}, dropout is applied to a network based on Maxout nonlinearities. The dropout probability started at 0.5 and linearly decreased to 0.0 after 8 epochs, after which no dropout was used. In\textsuperscript{[8]}, dropout was applied to LSTMs at the point where the input comes from the previous layer (this is equivalent to our "Location 2" below). In that case, no dropout schedules were used. Improvements in frame accuracy were shown on a Google-internal dataset of Icelandic speech, but no WER results.

\textbf{2.2. Projected LSTMs}

Projected LSTMs (LSTMps)\textsuperscript{[4]} are an important component of our baseline system, and to provide context for our explanation of dropout we will repeat the equations for them; here $x_t$ is the input to the layer and $y_t$ is the corresponding output.

$$i_t = \sigma(W_{ix}x_t + W_{ir}r_{t-1} + W_{ic}c_{t-1} + b_i)$$

$$f_t = \sigma(W_{fx}x_t + W_{fr}r_{t-1} + W_{fc}c_{t-1} + b_f)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{cx}x_t + W_{cr}r_{t-1} + b_c)$$

$$o_t = \sigma(W_{ox}x_t + W_{or}r_{t-1} + W_{oc}c_t + b_o)$$

$$m_t = o_t \odot \tanh(c_t)$$

$$p_t = W_{p_m}m_t$$

$$r_t = W_{r_m}m_t$$

$$y_t = (p_t, r_t)$$
where $\odot$ stands for element-wise multiplication; $\sigma$ (the sigmoid function) and $\tanh$ are also applied element-wise. In all experiments reported here, $p_t$ and $r_t$ are one quarter of the cell dimension: for example, the cell dimension might be 1024, so $p_t$ and $r_t$ would have dimension 256 and the output $y_t$ would have dimension 512.

### 3. Methods proposed

In this section we introduce some terminology to describe the various forms of dropout that we experimented with.

#### 3.1. Per-frame versus per-element dropout

One of the modifications to dropout that we tried is per-frame dropout. Unlike per-element dropout in which each element of the dropout mask is chosen independently, in per-frame dropout the entire vector is set to either zero or one. However, this is done separately for each individual part of the network, e.g. if dropout is used for multiple layers or gates then each would have an independently chosen mask.

#### 3.2. Dropout schedules

We experimented with various dropout schedules, and settled on schedules in which the dropout probability $p$ is zero for a time at the start of training and at the end, and rises to a peak somewhere closer to the start of training. Bear in mind that in the Kaldi met3-based recipes we used here, the number of epochs is determined in advance (early stopping is not used). Therefore we are able to specify dropout schedules that are expressed relative to the total number of epochs we will use. We express the dropout schedule as a piecewise linear function on the interval $[0, 1]$, where $f(0)$ gives the dropout proportion at the start of training and $f(1)$ gives the dropout proportion after seeing all the data. As an example of our notation, the schedule $0@0.2@0.4,0@1$ means a function $f(x)$ that is linearly interpolated between the points $f(0) = 0$, $f(0.4) = 0.2$, and $f(1) = 0$. We write this as $0, 0.2@0.4,1$ as a shorthand, because the first point is always for $x = 0$ and the last is always for $x = 1$.

After experimenting with various schedules, we generally recommend schedules of the form $0, p@0.2, p@0.5,0'$, where $p$ is a constant (e.g. 0.3, 0.5 or 0.7) that is different in different models. Because these schedules end at zero, there is no need to do the normal trick of rescaling the parameters before using the models for inference. The optimal $p$ for BLSTMPs is 0.15~0.10, we use 0.10 here.

#### 3.3. LSTMP with dropout

Some of the locations in the LSTMP layer where we tried putting dropout are as follows (see Figure 1 for illustration):

- **Location 1** (dropout before the projection): replace equation (5) with:
  \[ m_t = (o_t \odot \tanh(c_t)) \odot m_{drop}^{(m_t)} \]

- **Location 2** (dropout on the output of the layer): replace equation (8) with:
  \[ y_t = (p_t \odot r_t) \odot m_{drop}^{(y_t)} \]

- **Location 3** (dropout on each half of the projection output): replace equation (6), (7) with:
  \[ p_t = (W_{ym}m_t) \odot m_{drop}^{(p_t)} \]
  \[ r_t = (W_{ym}m_t) \odot m_{drop}^{(r_t)} \]

As alternatives to Location 4, we also tried dropout on all possible subsets of the $i, f, o$ gates (experiments not shown), and found that dropping out all 3 was the best. We didn’t try dropping out the output of either of the $\tanh$ gates, because it would have the same effect as dropping out the sigmoid gate that each one is multiplied by. In the rest of the paper, when we talk about per-frame dropout, we refer to the per-frame dropout location 4.

### 4. Experimental Setup

In this paper, HMM-DNN hybrid neural network acoustic models are used. The met3 toolkit in Kaldi speech recognition toolkit [9] is used to perform neural network training. The distributed neural network training algorithm is described in [10]. LF-MMI [6] is the training criterion.

40-dimensional Mel-frequency cepstral coefficients (MFCCs) without cepstral truncation are used as the input into the neural network [10][11]. These 40-dimensional features are spliced across $\pm n$ ($n$ may be 1 or 2) frames of context, then appended with a 100-dimensional i-vectors [12] to perform instantaneous adaptation of the neural network [13]. These i-vectors contain information about the mean offset of the speaker’s data, so cepstral normalization is not necessary. In all cases we use the speed-perturbation method described in [14] to augment the data 3-fold. Our AMI setup is similar to that described in [15]; one notable feature is that alignments

![Diagram of LSTMP block](image-url)
obtained using the individual headset microphone (HIM) data are used to train the single distant microphone (SDM) system.

4.1. TDNN-LSTMPs

A common trend in ASR modeling is to combine different types of layers [16] [17]. We have recently found that a combination of TDNN and LSTM layers can outperform BLSTMs. See Figure 2 for the illustration of a TDNN-LSTM configuration. Adding per-frame dropout to the LSTM units in TDNN-LSTMPs reduces the WER significantly. (Fast LSTMPs are efficient implementation of the LSTMPs in Kaldi, in which the recurrence would be of dimension 256 and the output $y_t$ would be of dimension 512.)

4.2. Neural network configuration

We report results on two different model structures: BLSTMP and TDNN-LSTMP. Regarding the projection in the LSTM layers, the dimensions of $p_t$ and the recurrence $r_t$ are always one-quarter the cell dimension, so for instance if the cell-dim is 1024, the recurrence would be of dimension 256 and the output $y_t$ would be of dimension 512.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Switchboard/Tedlium</th>
<th>AMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[-2,-1,0,1,2] TDNN</td>
<td>[-1,0,1] TDNN</td>
</tr>
<tr>
<td>2</td>
<td>[-1,0,1] TDNN</td>
<td>[-1,0,1] TDNN</td>
</tr>
<tr>
<td>3</td>
<td>[-1,0,1] TDNN</td>
<td>[-1,0,1] TDNN</td>
</tr>
<tr>
<td>4</td>
<td>[0] LSTMP</td>
<td>[0] LSTMP</td>
</tr>
<tr>
<td>5</td>
<td>[-3,0,3] TDNN</td>
<td>[-3,0,3] TDNN</td>
</tr>
<tr>
<td>6</td>
<td>[-3,0,3] TDNN</td>
<td>[-3,0,3] TDNN</td>
</tr>
<tr>
<td>7</td>
<td>[0] LSTMP</td>
<td>[0] LSTMP</td>
</tr>
<tr>
<td>8</td>
<td>[-3,0,3] TDNN</td>
<td>[-3,0,3] TDNN</td>
</tr>
<tr>
<td>9</td>
<td>[-3,0,3] TDNN</td>
<td>[-3,0,3] TDNN</td>
</tr>
<tr>
<td>10</td>
<td>[0] LSTMP</td>
<td>[0] LSTMP</td>
</tr>
<tr>
<td>11</td>
<td>-</td>
<td>[-3,0,3] TDNN</td>
</tr>
<tr>
<td>12</td>
<td>-</td>
<td>[0] LSTMP</td>
</tr>
</tbody>
</table>

Table 2: Dropout results on AMI IHM, dropout schedule is '0,0@0.20,p@0.5,0@0.75,0'.
Table 4: Per-frame dropout on AMI LVCSR, dropout schedule is ‘0.0@0.20,p@0.5’

<table>
<thead>
<tr>
<th>Model</th>
<th>Fast LSTM</th>
<th>Dropout Location</th>
<th>Dropout prob. p</th>
<th>Epoch</th>
<th>SDM</th>
<th>IHM</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLSTM</td>
<td>No</td>
<td>None</td>
<td>None</td>
<td>6</td>
<td>39.8</td>
<td>42.9</td>
</tr>
<tr>
<td>BLSTM</td>
<td>No</td>
<td>None</td>
<td>None 0.1</td>
<td>6</td>
<td>38.3</td>
<td>41.6</td>
</tr>
<tr>
<td>TDNN-LSTM</td>
<td>No</td>
<td>None</td>
<td>Location4</td>
<td>4</td>
<td>37.4</td>
<td>40.8</td>
</tr>
<tr>
<td>TDNN-LSTM</td>
<td>No</td>
<td>None</td>
<td>Location4 0.3</td>
<td>4</td>
<td>36.3</td>
<td>39.7</td>
</tr>
<tr>
<td>TDNN-LSTM</td>
<td>No</td>
<td>None</td>
<td>Location4 0.3</td>
<td>5</td>
<td>37.6</td>
<td>40.7</td>
</tr>
<tr>
<td>TDNN-LSTM</td>
<td>Yes</td>
<td>None</td>
<td>Location4</td>
<td>5</td>
<td>35.6</td>
<td>39.6</td>
</tr>
<tr>
<td>TDNN-LSTM</td>
<td>Yes</td>
<td>None</td>
<td>Location4 0.3</td>
<td>5</td>
<td>36.1</td>
<td>39.6</td>
</tr>
</tbody>
</table>

Table 5: Per-frame dropout on SWBD and TED-LIUM, dropout schedule is ‘0.0@0.20,p@0.5’

<table>
<thead>
<tr>
<th>Model</th>
<th>Fast LSTM</th>
<th>Dropout Location</th>
<th>Dropout prob. p</th>
<th>Epoch</th>
<th>SWBD</th>
<th>TED-LIUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLSTM</td>
<td>No</td>
<td>None</td>
<td>None</td>
<td>4</td>
<td>14.2</td>
<td>13.9</td>
</tr>
<tr>
<td>BLSTM</td>
<td>No</td>
<td>None</td>
<td>Location4 0.1</td>
<td>4</td>
<td>13.6</td>
<td>12.1</td>
</tr>
<tr>
<td>BLSTM</td>
<td>No</td>
<td>None</td>
<td>Location4 0.1</td>
<td>5</td>
<td>13.8</td>
<td>11.8</td>
</tr>
<tr>
<td>TDNN-LSTM</td>
<td>No</td>
<td>None</td>
<td>Location4</td>
<td>4</td>
<td>14.0</td>
<td>11.9</td>
</tr>
<tr>
<td>TDNN-LSTM</td>
<td>No</td>
<td>None</td>
<td>Location4 0.3</td>
<td>4</td>
<td>13.5</td>
<td>11.5</td>
</tr>
<tr>
<td>TDNN-LSTM</td>
<td>No</td>
<td>None</td>
<td>Location4 0.3</td>
<td>5</td>
<td>13.2</td>
<td>11.4</td>
</tr>
<tr>
<td>TDNN-LSTM</td>
<td>Yes</td>
<td>None</td>
<td>Location4</td>
<td>4</td>
<td>13.7</td>
<td>11.6</td>
</tr>
<tr>
<td>TDNN-LSTM</td>
<td>Yes</td>
<td>None</td>
<td>Location4 0.3</td>
<td>5</td>
<td>13.8</td>
<td>11.6</td>
</tr>
</tbody>
</table>

Figure 2: Diagram of our TDNN-LSTMP configuration. The input context is \{-2, -1, 0, 1\}, and the splicing indexes for layer2, layer4 and layer5 are \{-1, 1\}, \{-3, 0, 3\}, \{-3, 0, 3\}; LSTM time delay is 3.

some very minor differences in how the diagonal matrices of the LSTM are trained. The general conclusion that we draw is that our recipe works consistently using both LSTM implementations and a range of datasets. Of course we cannot guarantee that this would work in a setup that was very different from ours. For instance, our LSTMs generally operate with a recurrence that spans 3 time steps instead of 1.

We have not shown our experiments where we tuned the dropout schedule. We did find that it was important to leave the dropout proportion at zero for a certain amount of time at the beginning of training.

6. Conclusions

In this paper, we proposed per-frame dropout and investigated the best way to combine dropout with LSTMs. Our findings are:

- Per-frame, rather than per-element, dropout seemed to work best for us.
- We got the best results from applying dropout to the LSTM’s o, f and i gates, with separate (per-frame) dropout masks for each.
- A dropout schedule that starts and ends at zero, and has a maximum value of 0.3 or less, worked well for us.
- For TDNN-LSTM combinations, we favor dropout only for LSTM layers (and not for TDNN layers).

With these configurations we observed relative WER reductions of between 3% and 7% on a range of LVCSR datasets. Our conclusions are a little different from [8], which recommends dropout only for the forward (non-recurrent) connections, like our Location 2; but we should point out that [8] did not report any experimental comparisons with alternative ways of doing dropout.

7. References


