Technical Report: Bilingual Transfer Learning with Progressive Neural Network

Jiasheng Zhang & Yiming Liao

May 31, 2017

1 Model Architecture

The neural network (NN) topology is adopted from progressive neural network model [2, 3]. As shown in right part of Fig. 1, the NN consists of base model, which is the left five layers, and additional block, which is the right two layers. In context of progressive NN, the base model is used to fit former task, while the additional block empowers the model to fit latter task.

As a language model, from bottom, the first layer (green) is input layer, which is a window of tokens before target token, represented as concatenated one-hot vector. The second layer (red) is embedding layer. Hence the weight matrix between these two is a look-up dictionary for embedding vectors. The next two layers (blue) are hidden layers and the top layer is softmax output of predicted target token. All these layers are densely connected (embedding-hidden1-hidden2-output). The two layers in additional block serve similar roles as two hidden layers as connecting embedding layer and output layer, and densely connected with each other. However, there are two differences. First, the upper layer in additional block is densely connected with the lower hidden layer in base model; second, the output layer is now connected with both the upper hidden layer in base model and upper layer of additional block. Here, both double connection is additional (input from two lower layers are added). Currently, the size of layers in additional block is the same with that of corresponding hidden layers, respectively.

Now we define normal transfer learning schema (**Normal**).

1. language (L1) (or in general first task) is fit to base model, without additional block as shown in left part of Fig. 1, until convergence (Performance in validation set stop improving).

2. A whole new NN with both base model and additional block (as shown in right part of Fig. 1) is built to learn second language.

3. The connection weights (including bias terms) between embedding layer and first hidden layer and those between first and second hidden layers

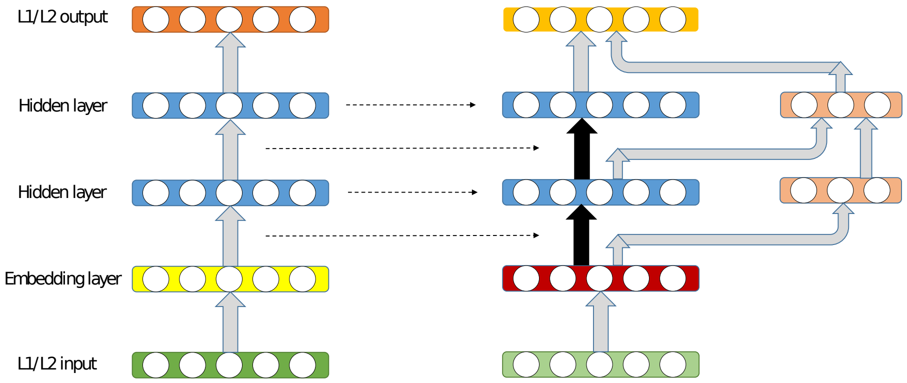

Figure 1: model architecture

are copied from the corresponding part of the base model which is fit to first language in first step.

4. The new NN obtained is fit to second language (L2), with weights (including bias terms) set in last step fixed (set to not-trainable).

To avoid possible confusion, we set two languages as La and Lb. In **Normal**, La is used as L1 and Lb as L2. Next, we define single language schema (**Single**). In **Single**, only base model is used to fit single language, which is exactly what we do in the first step of **Normal**. To show the impact of L1 learning on L2, we fit Lb in **Single** and compare the performance with that of Lb in **Normal**.

To test whether the results in **Normal** is related to two languages correlation, we define noise schema (**Noise**). It is similar to **Normal**, but the first step is removed and the copy operation in second step is replaced with setting those weights to random values (with same mean and variance with weights in **Normal**). In this way, the model represents learning some noise language as L1 and learning L2 later.

Finally, we define interleaving learning schema (**Interleaving**). Compared with **Normal**,

1. A small set of L1 (a small part of L1 corpus) is fed to a base model (NN-BASE), without additional block. Using stochastic gradient descent, only one epoch is run for that small set.

2. Same as Step 2 in **Normal** if no whole NN exists, otherwise skip this step. The whole NN is called (NN-WHOLE)

3. Same as Step 2 in **Normal**

4. The NN-WHOLE is fed with a small set of L2, similar to Step 1. Compared with Step 4 in **Normal**, those copied weights in Step 3 are now trainable.
5. Set weights (including bias) between embedding and first hidden layer and between two hidden layers in NN-BASE with the same values of corresponding part in NN-WHOLE.
6. Go back to step 1.

2 Experiment Result

2.1 data set

In this work, we use the English-Spanish parallel corpus in European Parliament Parallel corpora [1]. We set English as La and Spanish as Lb, with vocabulary size 26649 and 40403, respectively. From the corpus, we select 1 280 000 tokens as training set and 256 000 tokens as test set.

2.2 experiment setting

In learning each language, the training set, which is a stream of tokens, is divided into B blocks and each training epoch consists of a fixed random order of blocks (the same order for all epoch in individual training, different for different training). This setting contributes two factors to our goal of comparing the learning course of learning L2 based on model learning L1 and learning L2 from scratch. First, small blocks make it possible to follow the detail of early learning while the whole training set is still enough to for the model to master given language. Second, the course of learning, especially within each epoch, will be highly dependent on the order of training data. The random block setting alleviate such biases from dataset, and the final performance evolution over learning course will be based on training with different random order of blocks.

2.3 result

We record the perplexity of test set changing along learning courses, but take the lowest perplexity value as performance.

As shown in Fig.2, We compared Lb learning results from three schemes. Performance in **Single** and **Interleaving** is comparable while that in **Normal** is significantly worse.

As shown in Fig.3, we compared previous results with Lb learning in **Noise**. Though performance in **Noise** is better than that in **Normal**, it is still worse than that in **Single** [1].

As shown in Fig.4, we compared La learning results from **Single** and **Interleaving**. The performance of two schemes is comparable.

[1] Here **Noise** schema is tested only once

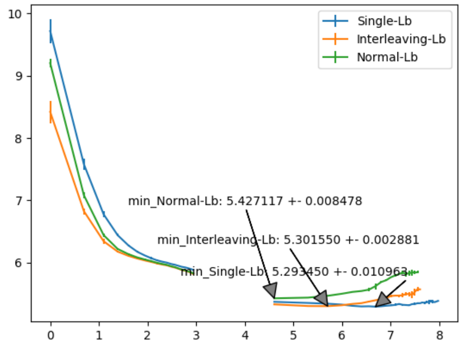

Figure 2: Experiment results of Lb learning. Here x axis is log-# epoch, y-axis is corresponding log-perplexity in validation set.

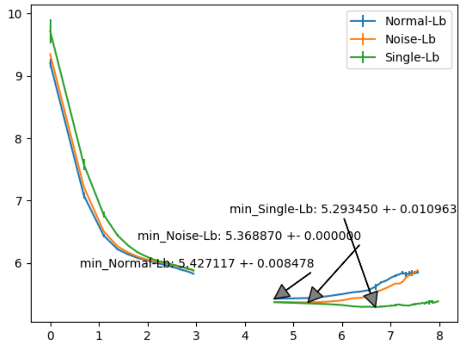

Figure 3: Experiment results of Lb learning compared with Noise schema.

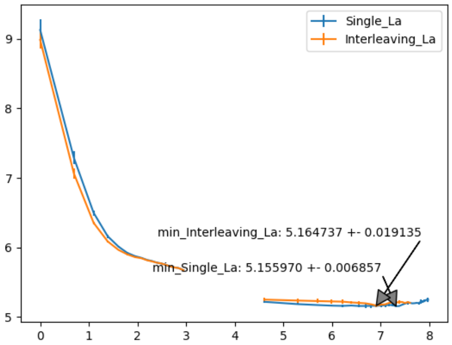

Figure 4: Experiment results of La learning.

3 Interpretation

3.1 human learner simulation

Schema **Normal** simulates sequential bilingual, who learn their first language early (step 1 in **Normal**) while second language late. In this schema, only connection between hidden layers and hidden and embedding layers are copied and fixed. Hence the input, embedding and output layers are language-aware, which means that the model knows which language it should use. In other words, we do not concern with switch function in human bilingual. Also the copy setting is based on the assumption that some of the deep structure of language processing in human brain is shared by two languages. And the artificial fixing weights setting in step 4 is based on the assumption that the deep structure of language processing in human brain shared by different languages is kept as long-term memory after learning first language.

Schema **Single** simulates monolingual behavior.

Schema **Interleaving** simulates native bilingual, who learn two languages in their early ages. Besides interleaving learning, the most distinguishing difference with **Normal** is that the copied connections are not fixed in Lb learning. This is based on the assumption that when the deep structure of language processing shared by different languages has not been well built by one language, it will be modified by concurrently learned the other language.

With above assumptions, the experiment results simulate that

- Sequential bilinguals achieve worse performance in their L2 compared with monolinguals.
- The native bilinguals achieve same performance in their both languages with monolinguals.

3.2 technical

The difference of Lb learning performance in **Normal** and **Single** is non-trivial but reasonable. On one hand, considering the NN-WHOLE model as shown in right part of Fig. 1, if we set weights between upper hidden layer in base model and output layer, and those between lower hidden layer in base model and upper layer in additional block, then as we set the layers in additional block the same size with two hidden layers in base model, the model reduces to that in **Single**. Hence there is at least one point in weights space that will make NN-WHOLE model (**Normal**) achieves at least the same performance in Lb learning with NN-BASE model (**Single**), so that the difference is non-trivial. On the other hand, This difference is because the copied weights lead fitting process into sub-optimal local minimum point in weights space. Thus it is also reasonable technically.

The same performance in both languages learned in **Interleaving** and **Single** is reasonable technically. The **Interleaving** schema represents interleaving multitask learning, when task complexity is under model limit, the performance should be similar to trained on single task.

4 Implication

As the simulation results described in Sec. 3.1, our model shows critical period effect on sequential bilinguals. As far as we know, this is the first work that simulates critical period effect with NN model. Though the essential part of critical period effect, the sudden drop of performance, is caused by the fixation of shared weights (see **Normal**), the assumption for that is reasonable. And our model successfully simulates native bilingual behavior in terms of performance compared with monolingual.

References

[1] Philipp Koehn et al. Europarl: A multilingual corpus for evaluation of machine translation, 2002.

[2] Guglielmo Montone, J. Kevin O'Regan, and Alexander V. Terekhov. The usefulness of past knowledge when learning a new task in Deep Neural Networks. *CEUR Workshop Proceedings*, 1583:1–9, 2015.

[3] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.